

Inferring Reputation on the Semantic Web

Jennifer Golbeck

University of Maryland, College Park
A. V. Williams Building
College Park, Maryland 20742
301-498-7476
golbeck@cs.umd.edu

James Hendler

University of Maryland, College Park
MIND Lab, 8400 Baltimore Ave
College Park, Maryland 20740
301-314-6642
hendler@cs.umd.edu

ABSTRACT

The so-called "Web of Trust" is one of the ultimate goals of the Semantic Web. Research on the topic of trust in this domain has focused largely on digital signatures, certificates, and authentication. More social notions of trust which are reputation based are beginning to gain some attention in their own right, but have been traditionally overlooked. In this paper, we describe an algorithm for generating locally-calculated reputation ratings from a semantic network. We present mathematical and experimental results that show the effectiveness of this algorithm to accurately infer the reputation of a node. We then describe TrustMail, an application that uses the network for rating relevant emails

CATEGORIES AND SUBJECT DESCRIPTORS

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – *semantic networks*.

GENERAL TERMS

Reliability, Human Factors, Verification.

KEYWORDS

Social Networks, Trust, Reputation, Semantic Web

1. INTRODUCTION

On the hypertext web, any person is allowed to make any statement with no requirements about its accuracy or truthfulness. When reading a web page, humans make many judgments based on the appearance of the page and the source of the information. Although someone could lie about their sources, it is relatively easy to generate at least some information about the source. On the Semantic Web, content is a series of statements that cannot be judged by appearance or professionalism. Since the underlying philosophy of the Semantic Web is to allow a computer take

distributed statements about the same resource and aggregate them, the source of information becomes removed one step from the presentation. The word "Trust" has come to have several definitions on the Semantic Web. Much research has focused on authentication of resources, including work on digital signatures and public keys. This provides confidence in the source or author of a statement, which is very important, but trust in this sense ignores the credibility issue. Confirming the source of a statement does not have any explicit implication about the quality of the statement.

Reputation is more social notion of trust. In our lives we each maintain a set of reputations for people we know. When we need to work with a new, unknown person, we can ask people with whom we already have relationships for information. Based on the information we gather, we form an opinion about the reputation of the new person. This system works well, even though there are a lot of people in the world, because communities tend to be highly interconnected, and the number of steps between any two people tends to be rather small. This is known as the Small World effect, and it has been shown to be true for a variety of social and web based systems [1,4,20,21].

If a reputation based system can be used in a Semantic Web context, it can be used to automatically generate ratings about other people on the semantic web, as well as the reliability of their statements. One issue that should be considered here is that of context-specific reputation. One person may trust another very highly with respect to physics, but not much at all with respect to auto repair. Maintaining context for reputation is important, but it does not affect the methods used to aggregate reputation ratings. For a specific topic, a sub-graph of the entire social network is used which only contains reputation ratings with respect to the given topic. From there, analysis can proceed just as it would for a more general rating.

In this paper, we present a voting based algorithm for aggregating reputation ratings on the Semantic Web. We show mathematically and through experiments that this algorithm gives very accurate inferences, even in the face of high uncertainty, for networks with many attackers. In section 5, we discuss applications

that we have developed using this reputation system, and then present more areas where such an analysis can be useful.

2. RELATED WORK

Yolanda Gil and Varun Ratnakar addressed the issue of trusting content and information sources [12]. 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 P2P context, the EigenTrust system [14] effectively computes global trust values for peers, based on their previous behavior. Their system was shown to be highly resistant to attack. The system could be translated to the semantic context that we are interested in, but one of the premises of this work is to compute local, rather than global reputations. Raph Levin's Advogato project [19] also calculates a global reputation for individuals in the network, but from the perspective of designated seeds. His metric uses group assertions for determining membership within a group. The Advogato [3] website, for example, certifies users at three levels. 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 [13] and Richardson, et al. [27]. 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.

3. THE REPUTATION INFERENCE ALGORITHM

3.1 Terms and Background

Reputation networks are made up of ratings given from one node to another. A directed edge from node A to node B indicates B's *reputation rating* for A. Different systems have used various ranges for ratings, including a three tiered system [19], a 1-9 rating system [13], and a continuous scale in the range [0,1] [27]. This work uses a {0,1} scale where 1 indicates a

good reputation, and 0 is a *bad* reputation. If A has no opinion of B, then there is no edge between the two nodes; the system does not have a neutral value.

If two nodes, say node A and node C, do not have a direct edge connecting them, the network can be used generate an *inferred reputation rating*. If node A knows node B, and node B knows node C, then A can use the path to compose the inferred rating for C. The algorithm for inferring this value forms the basis of any trust or reputation system.

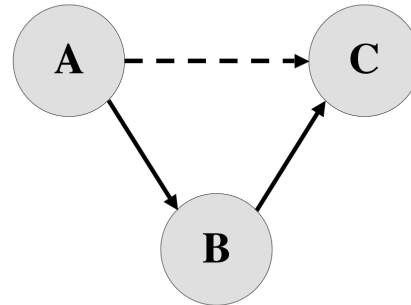


Figure 1. To produce a reputation from node A to node C, an inference can be made following the path through node B.

In real situations, people make evaluations of others based on local views of the world. To form an opinion about a known person, we turn to people we know and ask about the unknown person's reputation; in most cases, we do not rely on some centralized authority that determines the reputation each person deserves. One of the fundamental characteristics of this algorithm is that it is a purely local analysis. To infer a reputation rating starting at node A, the algorithm begins with A's neighbors and expands out. This means that node A may infer that node C has a very good reputation, while node D infers a poor reputation for node C. The difference will arise from the reputation ratings given along the paths from A to C and from D to C.

3.2 The Reputation Inference Algorithm

To make the inference from node A (the *source*) to node C (the *sink*), the algorithm begins with node A and works out. A polls each of its neighbors to which it has given a good reputation rating. Neighbors with bad ratings are ignored, since their reputation means that they give unreliable information. Each of A's good neighbors will return their rating for node C. Node A will then average these ratings and round the final value. A value between 0.5 and 1 rounds to a 1, while anything under 0.5 rounds down to 0. This rounded value is the inferred reputation rating from A to C.

```

currentColoring = color all nodes white

getRating(source, sink)
    numberOfNeighborsWithRatings = 0;
    sumOfRatings=0;

    colorGray (source)
    if sink is a direct neighbor of source
        source's rating of sink= rating(source, sink)
    else
        for each n adjacent to source

            if n is white
                if n has no ratingOfSink
                    c = currentColoring
                    inferredRating = getRating(n,sink)
                    if inferredRating >= 0
                        sumOfRatings +=inferredRating
                        n's rating of sink = inferredRating
                        numberOfNeighborsWithRatings++
                        currentColoring = c
                    else
                        if n's rating of sink >= 0
                            sumOfRatings+=n's rating of sink;

                colorGray (n)

            if numberOfNeighborsWithRatings > 0
                source's rating of sink =
                    Math.round(sumOfRatings/numberOfNeighborsWithRatings);
                return source's rating of sink

        else
            source's rating of sink = -1
            return source's rating of sink

```

Each of node A's neighbors will use this same process to come up with their reputation ratings for node C – if there is a direct edge connecting them to node C, the value of that edge is used; otherwise, the value is inferred. If a node is encountered in two paths from source to sink, it is considered in each path. For computational efficiency, results are cached at each node so the rating must only be inferred the first time the node is encountered.

Our system uses only 0 and 1 as reputation values. A value in the range [0,1] could be used, but the rounding in the algorithm will transform all of the values to 0 or 1 one iteration up from the nodes adjacent to the sink. Because of this, when a node polls its neighbors, it is essentially taking a majority vote. If half or more neighbors say that the sink has a good reputation, the average works out to be half or more, which rounds up 1.

3.3 Accuracy of Inferences

If all of the nodes correctly rated the reputation of each of their neighbors without error, there would be no need to do this type of search. We could simply choose a single path and trace it to get an accurate

rating of the sink. There are two places that errors can occur in real systems. Good nodes may make errors in ratings. When a good node, say A, inaccurately rates another good node, say B, as bad, the impact is very small. If B is not the sink, it will merely be dropped from the path; when calculating reputation ratings, neighbors with bad ratings are ignored. If B is the sink, chances are good that the incorrect rating will eventually be overcome by the accurate ratings of the other good nodes in the network. If a good node incorrectly gives a *bad* node, say C, a good rating, the situation is much more serious. Since bad nodes are intentionally incorrect 100% of the time, all bad neighbors between C and the sink will be considered in the network, and all good nodes will be ignored. Since it is much more common for any given node to be in the path between the source and sink and to be the sink itself, the effect of a misclassified bad node is much larger. The infiltration of bad nodes is the most serious threat to accuracy in the network.

At what points will results deteriorate? We will consider the worst case where “bad” nodes are always incorrect with their reputation ratings – they will say

every good node is bad, and that every bad node is good.

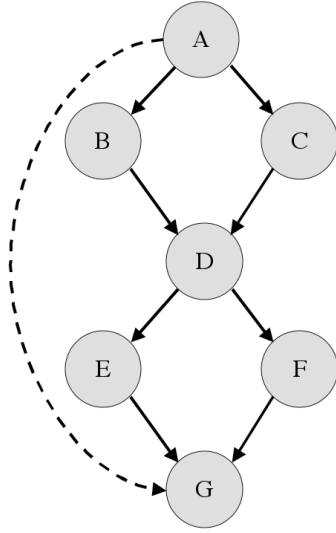


Figure 2. To infer a reputation from A to G, the rating from D will be used by both B and C. When a reputation rating from D is first requested, D will average the ratings from E and F. The value is then cached at D, so that the second time D's reputation rating is needed, no calculations are necessary.

With this algorithm, the inference is accurate if a majority of the nodes return correct ratings. Since the bad nodes are always incorrect, the accuracy of the good nodes must compensate. Let b be the percentage of bad nodes in the network. Then g , the percentage of good nodes, is equal to $1-b$. Let p_a be the accuracy of the good nodes – how often the reputation ratings they give are correct. To obtain a correct inference from a majority vote, we need

$$g * p_a \geq 0.5$$

Let $a = g * p_a$. For a given graph with n nodes, the probability that the majority of the nodes will correctly rate the sink is given by the following formula:

$$\sum_{i=\lceil \frac{n}{2} \rceil}^n \binom{n}{i} a^i (1-a)^{n-i}$$

When a is at or above the necessary threshold, the probability that our inference is correct increases as the graph size increases.

$$\lim_{n \rightarrow \infty} \sum_{i=\lceil \frac{n}{2} \rceil}^n \binom{n}{i} a^i (1-a)^{n-i}$$

The central limit theorem shows that for $a > 0.5$, the limit of the sum goes to 1 as the number of nodes goes to infinity. This is a critical point. As long as $g * p_a$ is greater than half, we can expect to have a highly accurate inference.

The formula above only gives the probability for the base level: nodes directly adjacent to the sink. As the algorithm moves up from the immediate neighbors of the sink toward the source, a will vary from node to node, but it will increase at each level because of the central limit theorem result. Figure 1 illustrates this point where the network starts with all good nodes, accurate in 70% of their classifications. After moving up three levels from the sink, the accuracy of the inference will be approximately 96%.

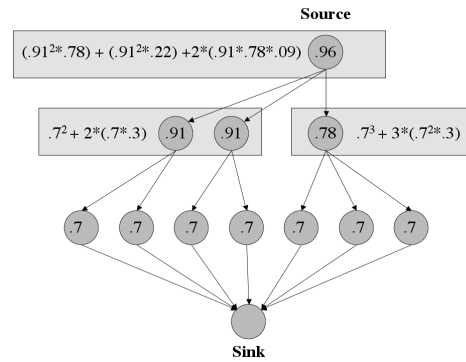


Figure 3. This figure shows a simple network and demonstrates the increasing probability of accuracy. Beginning with a uniform accuracy of 0.7, the probability of properly classifying the sink increases as we move up the search tree from sink to source. After only three levels, this source has a 96% chance of properly classifying the sink.

4. EXPERIMENTS

The literature has demonstrated that both social and web based systems exhibit small world behavior [4,6,7,17,21,23,28]. Because semantic reputation networks are essentially social networks that should have the small world properties, we used this model to automatically generate sample networks that were representative of what will occur in this domain. We used the β -graph model [30] to generate small world graphs. Our main results are on graphs with 400 nodes, but similar results were achieved for graphs with 1000 nodes.

To test the effectiveness of this algorithm, we generated a series of graphs where each node was given an *absolute rating of good or bad*. Accuracy of the inference was determined by whether or not it matched the absolute rating. For each edge in the graph, a reputation rating was assigned. To make this a worst case analysis, every bad node incorrectly rates its neighbors. For the good nodes, ratings are correctly generated with probability p_a .

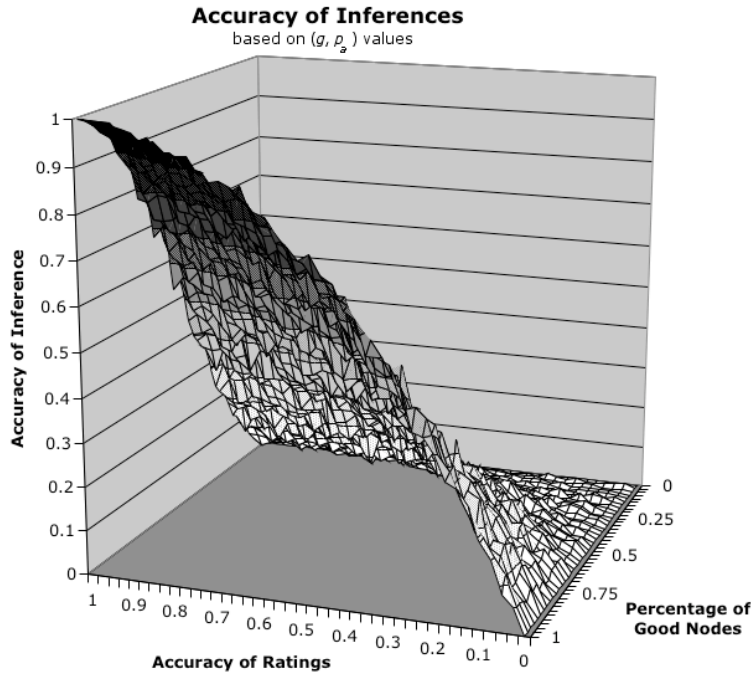


Figure 4. A surface map illustration how the accuracy of inferences changes with respect to changes in g and p_a .

The two variables in the network – percentage of good nodes, g , and the accuracy, p_a , of good nodes – each range from 0 to 1. We did not include the 0 value for either variable in our experiments, since it would always produce an inference accuracy of 0%. For the other values, using increments of 0.025, there are

As figure 4 shows, the accuracy of inferences remains very high, even as the accuracy of the initial ratings decreases. With no malicious nodes, inferences are 90% accurate for p_a as low as 65%. When attackers make up 10% of the population, inferences still remain over 90% accurate with p_a down to 0.75. With a population made up of 90% or more good nodes, the inferences actually remain more than 60% accurate, even when the initial ratings are below 40% accurate. Though initially this seems to contradict the result presented in the previous section – the inaccuracy should actually grow as it propagates up the network – the rounding gives us one additional benefit here. When looking at a network that comprises primarily good nodes, any ties in the voting will round up to a good rating. In a random network, rounding up from 0.5 would be correct only 50% of the time, but since most nodes are good, rounding up is correct nearly all of the time. That means that after one level of inference, the probability of a correct inference actually increases to more than 50%. This probability will, in turn, increase in subsequent layers as described above.

As bad nodes are introduced into the network, the accuracy of inference drops off. For example, the inverse of the previous example – when 67% of the nodes are good, but the good nodes are accurate 100% of the time – does not translate to a high inference. Though the inferences are accurate 75% of the time in this case, it is nowhere close to the accuracy seen with the high number of good nodes.

1,600 pairs (g, p_a) . For each pair we generated 1,000 small-world graphs with 400 nodes, and one inference was made and checked for accuracy on each graph. This created a smooth picture of the surface of the space.

These results also support the analysis described in the previous section and in Figure 3. Without that improvement, we would expect the probability of accurate inferences to be equal to the probability of an accurate initial rating – the variable a as presented above. As Figure 5 shows, for $p_a > 0.5$, the inferred value is always greater than a .

This system is *not* intended as a security system. Clearly, the error rates in this system would be unacceptable when sensitive information needs to be protected. However, its use as a system to make recommendations about the reputation of an information source is much better than most other recommendation systems [29]. In the recommendation context, our inferences are very accurate even in the face of exceptionally high uncertainty. Similarly, even when 10% of the population is made up of malicious entities – a very high attack rate for almost any system – this algorithm still produces good numbers. When properly applied to semantic contexts, reputation can be a useful and accurate metric for determining the integrity of a source.

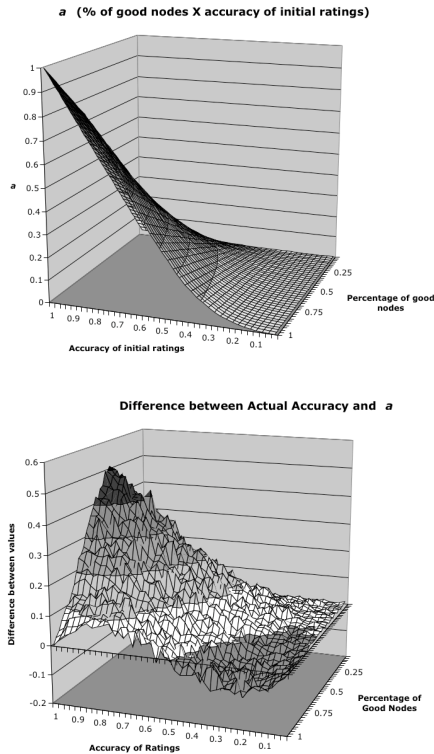


Figure 5. The top surface chart shows the value of a , and the lower surface shows the difference between the inferred accuracy and the value predicted by a . Positive values indicate when the inferred value was higher than a , and negative values when it was lower.

5. APPLICATIONS

To implement this algorithm, we have developed an OWL ontology¹ for assigning reputation. It extends the FOAF ontology [25] and adds the ability to specify reputation in several ways. The simplest method lets a `foaf:Person` rate the reputation of another `foaf:Person` with a value that should be 0 or 1. A `foaf:Person` can have a `reputationRating` property which has a range of a `Reputation` object. In the simple case, this object has two properties: the target of the reputation rating, and the rating itself. To accommodate topic specific ratings, the ontology also provides an `onTopic` property on the `Reputation` object that can take any URI as a value. This effectively restricts the reputation rating to that specified topic. With an ontology, the process of building a reputation network is only a matter of developing a large enough user community. In this work, we were able to assemble a network with several hundred nodes based on a mapping from the trust network ontology used in [13].

¹ Available at <http://www.mindswap.org/2003/reputation.owl>

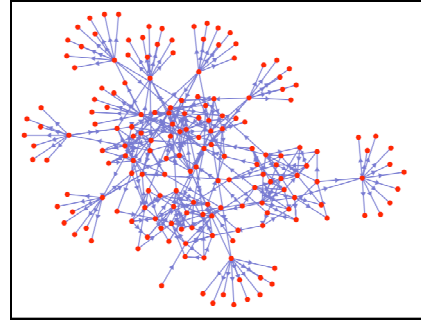


Figure 6. The reputation network examined in this work

Several interfaces to this network are available online including a web service, CGI interface, IRC Bot named TrustBot running on `freenode.net`², and TrustMail.

5.1 Reputation in Email – TrustMail

An ongoing project that has used these methods of reputation analysis is TrustMail [13]. TrustMail is an email client that looks up the mail sender in the reputation network and provides an inline rating for each email message. TrustMail can be configured to show trust levels for the mail sender either on a general level or with respect to a certain topic. Unlike spam filters that indicate which messages to ignore, reputation ratings in TrustMail can also tell users which messages are important to read.

Consider the case of two research groups working on a project together. The professors that head each group know one another, and each professor knows the students in her own group. However, neither is familiar with the students from the other group. If, as part of the project, a student sends an email to the other group's professor, how will the professor know that the message is from someone worth paying attention to? Since the name is unfamiliar, the message is not distinguishable from other, not-so-important mail in the inbox. This scenario is exactly the type of situation that TrustMail improves upon. The professors need only to rate their own students and the other professor. Since the reputation algorithm looks for *paths* in the graph (and not just direct edges), there will be a path from professor to student. Thus, even though the student and professor have never met or exchanged correspondence, the student gets a high rating because of the intermediate relationship. If it turns out that one of the students is sending junk type messages, but the network is producing a high rating, the professor can simply add a direct rating for that sender, downgrading the reputation. That will not override anyone else's direct ratings, but will be factored into ratings where the professor is an intermediate step in the path.

² For more information about all applications, please visit <http://owl.mindswap.org>.

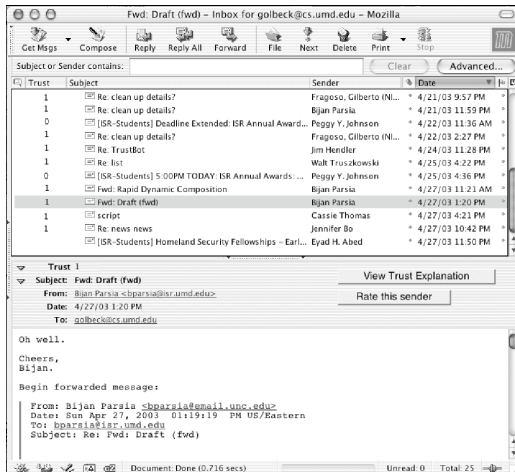


Figure 6. The TrustMail Interface

The ratings alongside messages are useful, not only for their value, but because they basically replicate the way trust relationships and reputations work in social settings. For example, today, it would be sensible and polite for a student emailing a professor she has never met to start her email with some indication of the relationships between the student and the two professors, e.g., "My advisor has collaborated with you on this topic in the past and she suggested I contact you." Upon receiving such a note, the professor might check with her colleague that the student's claims were correct, or just take those claims at face value, extending trust and attention to the student on the basis of the presumed relationship. The effort needed to verify the student by phone, email, or even walking down the hall weighed against the possible harm of taking the student seriously tends to make extending trust blindly worthwhile. In the context of mail, TrustMail lowers the cost of sharing trust judgments across widely dispersed and rarely interacting groups of people. It does so by gathering machine readably encoded assertions about people and their trustworthiness, reasoning about those assertions, and then presenting those augmented assertions in an end user friendly way.

5.2 Provenance

Reputation and trust on the semantic web have been gaining particular attention for their application to questions of provenance. Since any person can make assertions about any concept, it is important to be able to trace the source of an assertion. With all security measures in place and perfect performance at identifying the source of information, provenance alone does not give any information about whether the specified source should be trusted. Applying these reputation inference techniques for analysis of information sources can allow for automatic filtering of statements into reputable, unknown, and disreputable. As the semantic web stabilizes and develops a larger user community, provenance systems will become more common, and the automated sorting of statements will become a valuable tool.

6. FUTURE WORK

This paper addresses theory and applications for reputation – a semantic application that easily fits into the numeric format necessary for this type of analysis. We are currently in the process of doing more detailed network analysis using these techniques. To analyze networks with different types of connections between entities, we consider the implications of certain types of relationships in the specific domain. For example, when analyzing an organization, the strength of the connection between two individuals can be derived from the types of properties that connect them. We could expect that `people` `connected` `by` `a` `directSupervisorOf/directlySupervisedBy` property will have a stronger connection than two people connected with an `attendedTrainingSeminarWith` relationship. With a proper heuristic to assign numeric ratings to different types of connections based on the context of the analysis, the strength of a connection between two people can be inferred using methods similar to those described here.

In that context and others, our next step is to extend the analysis here to a continuous scale of trust ratings. The metric here succeeds, at least in part, because of its rounding and discrete scale. To understand how and why the metric will work with a range of [0,1] will allow the model to be extended into other domains where a polar system is not appropriate.

7. ACKNOWLEDGEMENTS

This work was supported in part by grants from DARPA, the Air Force Research Laboratory, and the Navy Warfare Development Command. The Maryland Information and Network Dynamics Laboratory is supported by Industrial Affiliates including Fujitsu Laboratories of America, NTT, Lockheed Martin, and the Aerospace Corporation.

The applications described in this paper are available from the Maryland Information and Dynamics Laboratory, Semantic Web Agents Project (MIND SWAP) at <http://www.mindswap.org>.

8. REFERENCES

- [1] Adamic, L., "The Small World Web". *Proceedings of ECDL*, pages 443-- 452, 1999.
- [2] Adding SVG Paths to Co-Depiction RDF, <http://Jibbering.com/svg/codepiction.html>
- [3] The Advogato Website: <http://www.advogato.org>
- [4] Albert, R., Jeong, H. AND Barabasi, A.-L. "Diameter of the world-wide web." *Nature* **401**, 130–131, 1999
- [5] Bharat, K and M.R. Henzinger. "Improved algorithms for topic distillation in a hyperlinked environment," *Proc. ACM SIGIR*, 1998.
- [6] Brin, S and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Proc. 7th WWW Conf.*, 1998.

- [7] Broder, R Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph structure in the web." *Proc. 9th International World Wide Web Conference*, 2000.
- [8] Carriere, J and R. Kazman, "WebQuery: Searching and visualizing the Web through connectivity," *Proc. 6th WWW Conf.*, 1997.
- [9] Chakrabarti, S, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, "Automatic resource compilation by analyzing hyperlink structure and associated text," *Proc. 7th WWW Conf.*, 1998.
- [10] 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.
- [11] FOAFBot: IRC Community Support Agent: <http://usefulinc.com/foaf/foafbot>
- [12] 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.
- [13] Golbeck, Jennifer, Bijan Parsia, James Hendler, "Trust Networks on the Semantic Web," *Proceedings of Cooperative Intelligent Agents 2003*, August 27-29, Helsinki, Finland.
- [14] 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*, May 20-24, 2003, Budapest, Hungary.
- [15] Kleczkowski, A. and Grenfell, B. T. "Mean-field type equations for spread of epidemics: The 'small-world' model." *Physica A* **274**, 355–360, 1999.
- [16] Kleinberg, J, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, 1999.
- [17] 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.
- [18] Labalme, Fen, Kevin Burton, "Enhancing the Internet with Reputations: An Openprivacy Whitepaper," <http://www.openprivacy.org/papers/200103-white.html>, March 2001.
- [19] Levien, Raph and Alexander Aiken. "Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas, January 1998.
- [20] Milgram, S. "The small world problem." *Psychology Today* **2**, 60–67, 1967.
- [21] Moore, C. and Newman, M. E. J. "Epidemics and percolation in small-world networks." *Physical Review E* **61**, 5678–5682, 2000.
- [22] Mutton, Paul and Jennifer Golbeck, "Visualization of Semantic Metadata and Ontologies," *Proceedings of Information Visualization 2003*, July 16-18, 2003, London, UK.
- [23] Newman, Mark, "Models of the small world", *J. Stat. Phys.* 101, 819-841 (2000).
- [24] Open Privacy Initiative: <http://www.openprivacy.org/>
- [25] RDFWeb: FOAF: 'the friend of a friend vocabulary', <http://rdfweb.org/foaf/>
- [26] RDFWeb: Co-depiction Photo Meta Data: <http://rdfweb.org/2002/01/photo/>
- [27] Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. "Trust Management for the Semantic Web," *Proceedings of the Second International Semantic Web Conference*, 2003. Sanibel Island, Florida.
- [28] Spertus, E, "ParaSite: Mining structural information on the Web," *Proc. 6th WWW Conf.*, 1997.
- [29] Swearingen, Kristen and R. Sinha. "Beyond algorithms: An HCI perspective on recommender systems," *ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, LA, 2001
- [30] Watts, D. Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton, NJ: Princeton University Press, 1999.