Open Rating Systems

R. Guha
IBM Research, Almaden
guha@cs.stanford.edu

ABSTRACT

In the offline world, we look to the people we trust and those they trust for reliable information. In this paper, we present a computational model of this phenomenon and show how it can be used to identify high quality content in an Open Rating System, i.e., a system in which any user can rate content. We present a case study (Epinions.com) of a system based on this model and describe a new platform called PeopleNet for harnessing this phenomenon in an open distributed fashion.

1. INTRODUCTION

Before the advent of the Internet and the World Wide Web, most large corpuses of content were Closed Publishing Systems. In a closed system there is a central authority who acts as a gatekeeper for publishing into the system. The gatekeeper typically verifies the quality of the content, and vouches for it. Such systems, by virtue of their central administration, tend to exhibit uniformity and predictability in the quality and growth rate of content.

In contrast, the Web is an Open Publishing System. In an Open Publishing System, anyone can publish content into the system without going through a central gatekeeper. Such systems have the capacity to exhibit very sharp growth phases during which the amount of content available increases very rapidly. Lacking a central quality control mechanism, they also exhibit a wide variation in the quality of the content available. Such systems inevitably run into the problem that a large amount of the content available on them is of a low quality. This problem is encountered in every widely adopted open publishing system, ranging from the Usenet and bulletin boards to the World Wide Web.

Consequently, filtering and ranking the content based on its quality and reliability becomes very important for these systems to remain usable. The most striking example of this is Usenet, which has grown dramatically but has become almost useless because of decreasing average quality. Other communication systems such as email are now facing the same problem.

The common approach to filtering and ranking is to to rely on groups of people who rate the content based on its quality and reliability. There are two main variations of this, based on whether the system for recording and publishing the ratings itself is open or closed.

Closed Rating Systems: In such systems, a group of "editors" are pre-qualified so that their ratings are known to be of an adequate quality. The Yahoo! [30] and LookSmart [25] directories are examples of such systems. The Open Directory Project [5] though open in the sense of making its data freely available, is also a closed rating system because its editors have to be pre-qualified before they can contribute. The biggest problem faced by such systems is that of scaling. If the amount of underlying content increases dramatically, as it did on the Web, such systems are unable to cover any significant portion of it. This is seen even in the context of ODP, where the number of sites reviewed has not increased in proportion to the growth of the Web. The problem is not just that of being able to pay an ever increasing number of editors (ODP does not have this problem), but also that the complexity of centralized coordination soon gets out of hand.

Open Rating Systems: One solution to the problem of getting ratings on a corpus of rapidly growing content is to make the system for publishing these ratings itself open. Not only can anyone publish content, they can also publish ratings on this content. The systems for publishing the content and the ratings can the same or different.

Over the last few years, a number of popular systems, such as Epinions [3] and Slashdot [6] which use the latter approach have emerged on the Web. Google [20] , by virtue of PageRank [27] interpreting links as ratings, also arguably shares this philosophy. In the next section, we describe the two problems that Open Rating Systems have to solve in order to effectively filter and rank content.

2. OPEN RATING SYSTEMS

Open rating systems tackle the problem of getting ratings on a large and rapidly growing body of content by using an open system for the expression of these ratings, i.e., anyone can publish ratings. Two problems need to be solved in the context of such systems:

Aggregation: We need a mechanism for aggregating the ratings of many sources into a single ranking. Various techniques such as PageRank [27] and Rank Aggregation [16] have been proposed for aggregating ratings and rankings.

Meta-Ranking: Given the open nature of the rating publishing system, often, there is a wide variation in the quality of the ratings themselves. Like the content being rated, many ratings turn out to be of poor quality. Consequently, some amount of ranking and filtering needs to be performed on the ratings

themselves. As a result, we are back to the problem of ranking, only this time, we are ranking ratings. Since each piece of content can have many ratings, this is arguably harder than the original problem of ranking the content.

Systems such as Epinions [3], Amazon reviews [2] and SlashDot [6]) have tackled the second problem by taking the open system philosophy to its natural next step. They allow ratings to be stated not just about the content, but also about the raters themselves. In practice, since a small, vocal minority of people state most of the ratings, and since there is considerable uniformity in the quality of ratings stated by a particular person, this step ameliorates the problem. These systems do have to avoid infinite regress, i.e., avoid getting into the question of whose ratings of raters we trust, and so on. This is typically done by a combination of mechanisms:

- The hierarchy of rating statements is allowed to bottom out at some level. e.g., explicit ratings on ratings on raters are not allowed.
- Some raters are apriori deemed trust-worthy and these are treated as the seed or root from which the trust of all others is computed. Advogato [1] and SlashDot[6] use this approach.
- The fact that many of the users of the system are themselves providers of ratings on raters can be exploited to give each user a view of the overall system which is maximally consistent with their ratings. This also has the added benefit of giving each user a personalized view of the system.

Systems such as Epinions, SlashDot and Amazon reviews seem to work. That is, they support an Open Rating System, incorporating mechanisms for rating not just content but also other users, and using the combination of these two kinds of ratings they are able to do a good enough job of ranking that these sites have remained useful.

In this paper, we provide a general model of such systems. We describe some lessons learnt from one of these systems (Epinions) and then describe a new system, PeopleNet, which provides an open distributed platform for open ratings. We also describe some applications of PeopleNet that are being built.

2.1 Context & Related Work

These systems often draw their inspiration from the phenomenon of "word of mouth" or "Web of Trust" as it occurs in the real world. In the real world, people's beliefs are strongly affected by who they trust. Information flows are significantly mediated by the networks of people that the information flows through [19]. Consequently, a better understanding of this phenomenon and the ability to capture it in computer networks will help us better exploit this phenomenon in coping with the flood of information.

Many different fields have looked at how people's come to hold beliefs and make decisions based on their relations with other people and organizations. Tversky and Kahneman [22] were the amongst the first to study some of these phenomenon in the context of decision making. There is also a substantial body of work on understanding trust in fields like political science ([26], [13], [29]). We draw a number of useful lessons from these fields, especially in assigning semantics to trust statements, but unfortunately, that work is not computational in nature.

There has been considerable work concerning trust in computer science, most of it focused in the area of security. Formal logical models ([12], [18]) have been used to in the context of cryptography and authentication. PGP ([17]) was one of first popular systems to explicitly use the term "Web of Trust", though it was not in the context of search or information flows. We believe that the *same kind* of trust relations between agents can be used not just for belief in statements about identity, but also for statements pertaining to the quality of content. However, the logical models proposed in the context of security are not appropriate for aggregating ratings for the purpose of ranking content.

Formal models ([23], [27], [21]) have been proposed for aggregating statements of quality for the purpose of ranking content. However, these models do not cover systems in which ratings can be stated not just about pieces of content, but also about authors and raters.

3. MODEL

In this section, we present a model which can be used for aggregating statements of quality based on trust relations between agents.

We have

1. a set of objects O: $\{O_1, O_2, O_3, ...\}$. These correspond to the objects that are being rated. In this paper, we are interested in the case where the objects are pieces of content. However, this model applies equally well to the case where the objects are products that are being rated or propositions whose truth is being judged.
2. a set of agents A: $\{A_1, A_2, A_3, ...\}$. These are the people who are either authors of the content and/or are stating ratings (raters) about the objects.
3. a set of possible values for ratings of objects D: $\{D_1, D_2, ...\}$.

   We assume that if A_i is the author of a piece of content, she rates it positively. This makes it possible to build a system exclusively on such implicit ratings.
4. a set of possible values for ratings of agents by other agents T: $\{T_1, T_2, ...\}$. For the sake of simplicity, in the rest of this paper we will assume that T = {Positive, Negative}.
5. a partial function $R : A \text{ x } O \rightarrow D$. This corresponds to the ratings given by various agents to various objects. For typical systems, consisting of a large number of agents and objects, most agents will not have had an opportunity to rate most objects and hence this function will be very sparse.
6. a partial function $W : A \text{ x } A \rightarrow T$. This corresponds to the ratings given by various agents to other agents. For typical systems, consisting of a large number of agents, most agents will not have had an opportunity to rate most other agents and hence this function too will be very sparse.

3.1 Discussion

W defines a directed labeled graph in which the nodes correspond to the agents and the edges, labelled with one of the values from D, correspond to the trust relations between agents.

The graph defined by W is often referred to as the "Web of Trust". We will refer to the case of an agent A_i having a Positive rating of

another agent A_j as A_i trusts A_j and to the case of A_i having a Negative rating of another agent A_j as A_i distrusts A_j.

Since we allow for the statement of not just positive ratings, but also negative ratings, both about agents and objects, we can get seemingly irrational/inconsistent trusts and ratings such as A_i trusting A_j even though they disagree on the rating for O_n. I.e., $R(A_i, A_j)$ is Positive, $R(A_j, O_n)$ is Negative but $W(A_i, A_j)$ is Positive.

3.2 Special Cases

There are a number of special cases of the above model which are interesting.

- Restrictions on D: We can restrict the set of possible ratings for objects to a small set of values. In particular, we can restrict D = {Positive, Negative} or even to just {Positive}.

  In the case where D = {Positive}, R too defines a simple relation corresponding to the agents who have rated objects positively. We will refer to this relation as R_P.

  In this paper, we will mostly focus our attention on the case where D= {Positive, Negative}.

- Restrictions on T: Similarly we can restrict the set of possible ratings for agents. In particular, we can restrict D = {Positive, Negative} or even to just {Positive}.

  In the special case where T = {Positive}, i.e., there is no distrust, we can drop the labels and get a simple directed graph. We will refer to the graph we obtain by ignoring the distrust relations as W_T.

  In this paper, we will mostly focus our attention on the case where T= {Positive, Negative}.

- Restrictions on T & D:

  In the special case of ignoring distrust and negative ratings, we define a graph G = $R_P \cup W_T$. This is a graph whose nodes are agents and objects, with arcs going from agents to agents when the first agent trusts the second or from agents to objects when the agent has rated the object positively.

3.3 Richer Formulations

There are many richer formulations which try to capture more real world phenomena. For example, an agent might trust another only for certain topics and not for others. Further, these topics can be arranged in a hierarchy with some form of inheritance of trust between nodes in the hierarchy. Both trust and ratings need not be just Positive and Negative, but can take on values in an interval. These more advanced formulations are beyond the scope of this paper.

The labelled graph defined by W, the Web of Trust, can either be obtained by each user explicitly making statements of trust (or distrust) or by mining corpuses such as email, newsgroup postings and bibliographic databases (for citations and co-authorship). This model does not make any assumptions about how the Web of Trust is created. However, how the graph is obtained does impact what it *means* and hence what mechanisms can be meaningfully carried out on it.

3.4 Rating & Ranking

In the context of the above model, the task of identifying high quality content maps into two distinct problems: that of Rating and Ranking.

Rating: The rating problem is to complete R using W, i.e., assuming that Jane[1] has not rated O_j, predict what her rating would be, if she were to rate it.

The core intuition is that there is a strong relation between R and W, i.e., if many of the agents rated highly by Jane rate an object O_j highly, then Jane is more likely to herself rate the O_i highly, if she were to rate it.

Ranking: Often, we don't need explicit ratings of various pieces of content. We are given a subset of O (typically corresponding to the results of a query process such as search) and need to rank the items in this subset from the perspective of Jane.

Further, in many cases, the subset of O identified by the query is quite large and we don't need to rank all the items in the subset of O. All we have to do is pick the top N (where N is typically 10 or less) items of the subset. In this paper, our primary focus is on this problem, which we will refer to as the Top-N Ranking problem.

4. APPROACHES TO RANKING

There have been different approaches to the ranking problem, some of which are more ad-hoc than others.

SlashDot, a popular site amongst users and followers of open source software has dealt with an increasing volume of postings by using a moderation system based on a open rating system with a rudimentary Web of Trust. Each posting has a rating (or score) which is computed as follows. Regular readers are selected on an adhoc basis each morning (by the administrator of the system) to become temporary moderators. The temporary moderator has the power to increment or decrement the scores of other readers' posts. Readers of SlashDot can choose to read only posts whose score have been moderated to above some threshold. Users whose posts are consistently moderated up gain bonus points for their future posts. Similarly, the administrator occasionally, when he has the time, takes away the the moderation/rating of those who have been judged by him as not moderating well. Users can als select "friends" and "foes" which in turn affects what they can see through an unpublicized mechanism. While it is true that SlashDot seems to work and has its following, we find it rather adhoc. [7] proposes a more systematic, though complex mechanism based on concepts like context and experience, but as it points out, that approach too requires several adhoc weights per user.

We believe that for such systems to become more widely useful, the ranking mechanism has to be systematic (i.e., not adhoc). Further, we would like it to be well founded, i.e., respect the semantics of trust statements.

PageRank provides a systematic approach to the problem of determining the importance of pages on the Web. So, we first consider a adaptation of PageRank for our purposes.

4.1 PageRank Adaptation

The PageRank algorithm [27] provides a way of computing the importance of a page based on the number of other pages that link to and their importance.

[1]For the rest of this paper, we will use Jane to refer to the user for whom the system has to generate the rating or ranking and Fred to someone who she trusts or distrusts

If we ignore distrust and negative ratings, i.e., we only consider statements of trust and positive ratings, we can easily adapt PageRank as follows. We focus our attention on the graph G defined in the previous section, whose nodes are agents and objects with arcs going from agents to agents when the first agent trusts the second or from agents to objects when the agent has rated the object positively. G is analogous to the Web graph whose nodes are web pages with arcs corresponding to hyperlinks. We apply the PageRank algorithm to G to get a rank (AORank) for each agent and object. U sing AORank, we can easily solve the ranking problem.

This simple formulation does not cover distrust and negative ratings. We cover negative ratings as follows. We first compute the AORank for every content object based on just trust and positive ratings. We then adjust the AORank of every content object as follows:

Let B_v be the set of agents who have rated an object O_i Negative. Then, the Modified PageRank of O_i is :

$$ModifiedPageRank(O_i) =$$

$$AORank(O_i) - \sum_{v \in B_v} AORank(v)/N_v$$

where AORank(x) (both, for agents and objects) is the rank of x as computed on G using the PageRank algorithm and N_v is the normalization for the number of negative ratings made by v. Note that unlike PageRank or AORank, ModifiedPageRank can be negative. Intuitively, the more trustworthy a person is, the more her negative rating should count for.[2]

This simple approach does not make use of user's Web of Trust. By adapting the approach for topic specific biasing PageRank ([21]) by (implicitly) assuming that every agent trusts the user, we can boost the importance of the user's Web of Trust and generate a set of AORanks for each user, i.e., corresponding to each user there will be an AORank for each other agent and object. Unfortunately, while this works in principle, in any system with more than a handful of users, it is not really practical to compute as many sets of ranks as users.

4.2 Discussion

Apart from the practical difficulty of computing a user-specific AORank, we see another problem in simply adapting PageRank for the problem of ranking. We argue that the intuitions behind PageRank differ from those of Webs of Trust in at least a few important ways.

The activity level of different users varies substantially. Some users do not like to make trust statements or rate content. Others are prolific. Some users may have been using the system for a long time while others may be fairly new. With the PageRank approach, when an agent makes the $N+1^{st}$ statement or trust (or rating), s/he reduced the import of the first N statements. This however does not correspond to the real world in which a statement of trust (or rating) does *not* decrease in value when the user trusts one more person or rates one more object. Therefore, using a uniform approach to normalizing their statements (based on the number of statements they have made) is not appropriate.

Another aspect of trust not captured by PageRank is its tendency to decay. PageRank has a decay effect by virtue of its normalization. With trust, even in a case where each person trusted only one other (in which case PageRank would not show any decay), if we were to traverse a sequence of N trust links starting with Jane, she typically does not trust the N^{th} person as much as she trusts the first.

On the web, there is no explicit, machine understandable mechanism for the author of one page to say that he thinks that another page or site is of high quality. PageRank *interprets* a link from one page to another as such a statement. In our case, we have users explicitly making statements of trust and rating objects. Any mechanism which uses these statements imposes a *semantics*, either explicitly or implicitly, on such statements. The relative merits of different mechanisms for ranking can be judged only in the context of a specified semantics for statements of trust. It would therefore be very useful to have a semantics for trust. In the next section, we define a semantics for trust.

5. THE SEMANTICS OF TRUST

We assign the following intuitive meaning to the statement that Jane trusts Fred. If Jane trusts Fred, all other things being equal, she assigns a much higher likelihood to a statement being true if Fred believes it than if he didn't. Note that she might not be convinced (i.e., assign it a probability of 1) that the statement is true. She simply assigns it a much higher likelihood. Or more generally, all other things being equal, the likelihood she assigns a statement strongly correlates with that assigned by Fred. We formalize this intuition as:

$$trust(A, B) \Longrightarrow (P_A(s) \ll P_A(s|believes(B, s))$$

where $trust(A, B)$ means that A trusts B, $P_A(s)$ refers to the apriori probability assigned by the agent A to the statement s, $believes(B, s)$ means that the agent B believes that s is true and $P_A(s|believes(B, s))$ refers to the probability assigned by A to s, conditioned on B stating that s is true. This says that B believing s causes a substantial change in the likelihood assigned by A to s. It does not say by how much. In order to capture the intuition about decay, we need to be more precise.

$$trust(A, B) \Longrightarrow (P_A(s) \leq \alpha \times P_B(s))$$

where $P_B(s)$ is the probability assigned by the agent B to the statement s and α is a decay factor such that $\alpha^m \approx 0$, where m is the number steps in the Web of Trust by which we want the trust to decay out.

The semantics of distrust is less straightforward than that of trust. A simple interpretation of distrust statements would be to interpret them as the negation of trust statements. That is, if Jane distrusts Fred, the likelihood she assigns a statement strongly correlates negatively to that assigned by Fred. Unfortunately, distrust is not exactly like negation. Unlike negation, where two negations cancel out each other, most often two distrusts don't cancel out each other,

[2]Distrust is a more tricky concept to incorporate, which we shall come to later.

i.e., it is *not* true that

$$distrust(x, y) \wedge distrust(y, z) \Longrightarrow trust(x, z)$$

A more conservative meaning of distrust would be that if Jane distrusts Fred, then the likelihood she assigns to statements is *not* correlated to that assigned by Fred. In terms of probability, this says that the likelihood assigned by Jane is independent of that assigned by Fred.

But in a world where W is very sparse, to perform any computation, we have to assume that those not in Jane's Web of Trust don't affect her beliefs. In such a system, why do we need explicit distrust statements? As our experience in Epinions showed, distrust statements are very useful for users to "debug" their Web of Trust. Jane might trust Fred who trusts Joe who trusts Jack, but Jane might herself not trust Jack. By stating that she distrusts Jack she can tell the system that she does not want what she sees to be affected by Jack's ratings, even though she trusts Fred and Fred trusts Joe. In the rest of this paper, we will use this interpretation of distrust.

For the problem of ranking, we do not need to assign actual probability values. The formulas given above allow us to take a qualitative approach to solving the ranking problem.

The semantics described above does not make any assumptions about the causal origins of the different ratings or the apriori probabilities of ratings. In practice, any system which computes rankings must take these into account.

5.1 Assumptions

A system which computes rankings will have to make some assumptions about apriori ratings and independence.

Apriori Ratings: Both W and R tend to be very sparse, i.e., most agents don't know each other and most agents haven't rated most of the content. The value of knowing a trust relation or a rating is a function of the apriori rating of of a piece of content. If there is very little variation in the quality of the content, or if a rating from a trusted source provides little new information, the rating system as a whole contributes little. We assume that the variation in the quality of content is sufficiently high that knowing a rating by a trusted agent substantially increases our confidence in the quality of the content (positively or negatively). Furthermore, we assume that the difference in confidence level is sufficiently high that in the presence of a rating by a trusted agent, we can ignore the apriori bias.

Independence: There are two independence assumptions we make.

- **Causality:** If Jane trusts Jim and Mary and Mary trusts Jim, it is quite possible that Jane trusts Jim because she trusts Mary. In such cases, different trust statements and ratings are not independent. The conditional probabilities of various trust and rating statements should in principle be taken into account by the ranking system. However, determining this is usually beyond the scope of the ranking system. Consequently most systems, including those described in this paper, assume that the different trust statements and ratings are all independent.
- **Apriori Trust:** We assume that unknown agents, i.e., agents not in Jane's Web of Trust, do not affect her beliefs. In other words, not only is the apriori ranking of an unknown piece of content very low, the apriori trust in an unknown agent is also very low.

In the next section, we describe an approach which solves the ranking problem with respect to the above interpretation of trust.

6. RANKING

Given the probabilistic interpretation of the previous section, we can create a belief network, compute likelihood's of various beliefs and use that for ranking. The structure of the belief network is derived from the Web of Trust. However, this approach may be computationally hard in practice in a system with millions of users supporting millions of queries every day. Further, it does not exploit either the structure of the semantics, i.e., that trust statements make *substantial* changes to likelihood estimates, the independence assumptions or the real-world observation that trust decays out quite fast, over three or at most four steps. Here is an approximate algorithm which has proven to work quite well in the context of Epinions.

Given a user A_u who trusts $\{A_{ut1}, A_{ut2}, ...\}$, distrusts $\{A_{ud1}, A_{ud2}, ...\}$ and a set of objects O_s $\{O_{s1}, O_{s2}, ...\}$, where at least some of $\{A_{ut1}, A_{ut2}, ..., A_{ud1}, A_{ud2}, ...\}$ have stated ratings on the objects, we need to compute the top N rated objects in O_s. Under the belief interpretation assigned to trust statements, we have to select the N objects with the highest probabilities of having Positive ratings.

We do an iterative deepening (up to a pre-specified number of levels, typically 3) traversal of the Web of Trust graph. We first traverse all the agents directly trusted by A_u, then the agents they trust, who are not distrusted by A_u, and so on. At each level of the iteration we collect all the objects (in the set O_s) which have been rated along with their ratings and aggregate the ratings into a cumulative score for each object. Every Positive rating by a trusted agent adds a point to the score, every Negative rating by a trusted agent deletes a point from the score. In each pass we pick the objects with a score more than a preset threshold of scores. Typically, this threshold is just 1, i.e., we just need one of the agents in Jane's Web of Trust to certify that a piece of content is good. We stop when we have N objects.

It is easy to see that under the independence assumption, in any single step of the iteration, the object with the highest score is the one most likely to be Positive.

This approach assumes that the effect of even one agent directly trusted by A_u is going to be more than the cumulative effect of those who are one level out in A_u's Web of Trust. Or more generally, the effect of level N is always *much more* than the effect of level N+1. It is easy to construct examples where this is not true, i.e., in which a large number of the agents in level N+1 disagree with that of level N. In practice, given that some of the agents in level N trust some of those in level N+1, this rarely occurs.

7. GLOBAL TRUSTWORTHINESS

The above approach by itself works well for those who have a rich Web of Trust. However, in most real world systems a substantial number of users are anonymous to the system. They are either new to the system or have for some other reason, have expressed few or no statements of trust. The ranking mechanism not only has

to behave reasonably for users who have made their statements of trust, but also for these anonymous users. Even those with rich Web of Trusts may sometimes find that no one in their Web of Trust has rated any of the objects they are interested in in ranking (e.g., when they venture out into new topics).

This phenomenon of having to look beyond one's Web of Trust occurs often in the real-world as well. In such cases, we look to established, globally trusted sources for our opinions. Therefore, we need a way of determining the global trustworthiness of agents. This problem requires a global analysis. Since the solution applies equally well to all anonymous users, this analysis can be performed statically in batch mode.

We model the process by which the user (who is looking for someone to trust in some new context) decides who to trust as "asking around", i.e., picking someone at random, asking that person who she trusts, asking those people in turn and so on. Those whose names come up frequently (as being trustworthy) get trusted. This process is very similar to the random walk motivating PageRank. And hence, PageRank is a good basis for computing a global *TrustRank*.

PageRank [27] computes a level of quality for web page based on how many other pages link to it. We can use a similar concept to compute the *TrustRank* of an agent. One complicating factor is that of distrust for which there is no analog in PageRank. There are two candidate approaches to distrust. One is to combine trust and distrust to come up with a single measure that combines both. Such a measure, could be calculated by an iteration in which the TrustRank in iteration N+1 is computed from the TrustRanks in iteration N as follows:

$$TrustRank_{N+1}(A_u) = \sum_{v \in T_v} TrustRank_N(v)/N_v - \sum_{u \in D_u} TrustRank_N(u)/N_u$$

It is easy to see that in a Web of Trust dominated by distrust, this iteration might not converge. We can deal with this by increasing N_u in every iteration. This approach treats distrust as being analogous to negation, i.e., if Jane distrusts Fred who distrusts Jack, Jane trusts Jack. However, this is not supported by the semantics of section 5.

Further, this approach, does not distinguish between an unknown, i.e., someone who almost no one trusts or distrusts and someone conterversial who inspires strong positive and negative feelings. One way of getting around these two problems is to use two separate measures, one for trust and another for distrust.

We first compute the TrustRank by looking only at the trust relations. i.e.,

$$TrustRank_{N+1}(A_u) = \sum_{v \in T_v} TrustRank_N(v)/N_v$$

We define a *DistrustRank*, the global distrust level, of an agent A_u as follows. Let B_v be the set of agents who distrust A_u. Then, the distrust rank of A_u is

$$DistrustRank(A_u) = \sum_{v \in B_v} TrustRank(v)/N_v$$

where DistrustRank(A_u) is the distrust rank of A_u, TrustRank(v) is the trust rank of v and N_v is the normalization for the number of people distrusted by v. Note that unlike the first calculation, this is not an iteration. This is done only once. Essentially, what this says is that if highly trusted agents distrust someone, that person has a higher distrust level.

Once we have these TrustRanks, we can pick the N most trusted agents, assume that the anonymous user trusts these agents and apply the ranking process of the previous section. Depending on the application, we can use just the TrustRank or a combination of the Trust and Distrust Ranks. We can either only look only at high trust ranks (i.e., ignore distrust ranks) or look at high trust ranks but avoid those who also have high distrust ranks or combine the trust and distrust ranks.

There are several other enhancements that can be made in the computation of the trust ranks, some of which are discussed in the context of the case study in the next section.

7.1 Combining Local and Global Trusts

Often, though a user might trust some small set of other agents, the ratings stated by these agents (and those they trust, etc.) might not be enough to pick out the top N objects from the given set of objects. In such cases, we can use a hybrid approach wherein we first look to the user's trusted agents to select the objects and if not enough objects are found, turn to globally trusted agents to pick the rest. This approach of augmenting every user's Web of Trust with the globally trusted agents not only provides the system with more predictable behavior in the absence of a significant number of ratings, but also has interesting privacy protection features as explained in the next section.

8. A CASE STUDY: EPINIONS

In this section we briefly describe a large scale system[3] which uses the model described in the last section to automatically rank user generated content.

Epinions [3] is a website where users can write reviews about a variety of different things, ranging from consumer durables (such as cars and toasters) to media objects (such as music and movies) to colleges to vacation spots. Given the large number of users (on the order of millions) and the high rate of new reviews (on the order of thousands a day), it is very important to have an automated mechanism for selecting the best reviews for any given topic. A complicating factor in many areas such as movies, music and wines, where tastes are subjective, is that what counts as a good review for one user might not be a useful review for another person.

The reviewable objects are arranged in a taxonomy with top level nodes corresponding to categories of objects (Electronics, Autos, Books, ...). Any user may contribute a review on any object. In addition to a human readable piece of text, each review also typically contains two to five rankings, on various axes (e.g., usability, reliability, etc.) of the object, typically on a scale of one to five. These axes are a function of the kind of object. So, reliability may be an axis for cameras but not for universities. Finally, the user also has to provide an overall rank on a scale of 1-5 for the object.

[3]The system described here, which was designed by the author, was operational circa April 2000. This paper might not accurately reflect on the system currently being used at Epinions.

In addition to writing reviews, a user can also rate reviews of other users on a scale of four ratings, ranging from very useful to useless. Finally, a user can also indicate that s/he "trusts" or "distrusts" another user. The Amazon and Slashdot websites also have similar concepts, though they use different terminologies.

Most objects accumulate more reviews than any user can read. Moreover, there is a wide variation in the quality of reviews. Most users are only looking for the top three to five reviews for any particular product. So, given a user and an object, the system needs to identify the top N reviews for that object, for that user. This is done using the approach described in sections 6 and 7.

Often, the user is not researching about a particular product (such as Fizko toaster model 4234) but is instead looking at the page corresponding to the product category (such as toasters or merlots under $10) and would like some recommendations on which products in that category he should look at. So, given a set of objects (each of which has a number of reviews) and a user, the system needs to identify the top N (typically 5) products to recommend to that user. A variation of this problem is one where we have to pick the top few products to warn the user about (i.e., identify the "lemons"). This problem is also solved by using the approach of sections 6 and 7, except, instead of using the ratings for reviews, we use the ratings of the products.

Getting one's reviews rated highly by a number of other users, especially if these users were highly trusted, resulted in these reviews getting more prominent positions. One complicating aspect at Epinions was that reviewers were paid royalties based on how many times their reviews were read. This motivated substantial efforts to game the system, i.e., introduce ratings and trust statements which did not reflect on either the content or the trustworthiness of the user.

In order to combat attempts to "game" the system, a small (a few hundred) "Top Editors" were selected, a few from each of the major topics on Epinions (Autos, Books, Electronics, ...) and given apriori high TrustRanks, i.e., were Globally Trusted for articles on those topics. This not only stabilized the system against attempts to game the global trust rankings, it also allowed us to use topic specific trust ranks. With topic specific trust ranks, depending on the major topic that the user was looking at, a different set of globally trusted agents would be used to augment the user's Web of Trust.

8.1 Discussion

Judging by the popularity of the site and the high quality of reviews that get selected, the approach described in sections 6 and 7 seem to work, at least in the context of Epinions. The Web of Trust at Epinions exhibits several interesting phenomenon. We highlight and discuss some of these below.

Cliques: There were a number of small groups of users (few dozen or fewer in each) many of whom trusted many others. Some of these groups corresponded to real-world social groups, i.e., a set of friends who did really trust each other. In other cases trust and ratings swapping cliques would emerge in an effort to boost the overall ratings of those involved in the cliques. In general, it is hard to distinguish between these two kinds of cliques, purely by looking at the graph structure of trust relations. However, a couple of heuristics turn out to be quite useful.

1. Rating swapping cliques are set up very fast. In contrast, real cliques tend to take time to form.
2. Rating swapping cliques are very insular. Almost no one outside the clique trusts any of the clique members.
3. Real cliques often have short paths leading from a Recognized Trusted User to one of the members of the clique.

Conflict of Interest: Users who had written a review of a particular product often rated other reviews of that product badly. A variation on this theme was that of "bad rating swapping" wherein two (or more) authors, who had written reviews of different products would collaborate to rate other reviews of those products badly. These are all different forms of conflict of interest which an ideal ranking system should guard against.

Privacy: Initially, all statements of trust were available for everyone to see. They were typically on the pages of the user. Some users do not want to reveal who they trust and distrust. Consequently it became important to give users the option of hiding who they trust/distrust. However, it is possible to guess who a user might trust/distrust by looking at the recommendations one gets by trusting that person. The Epinions solution to this problem is to augment every user's web of trust with the global web of trust. This way, it is not possible to precisely identify why a particular recommendation was made or not made. In general however, it is still possible to carefully craft situations which might reveal elements of a person's web of trust.

Power Law Distribution: From the beginning, how much how many people were trusted (i.e., the plot, as shown in figure 1, of number of people who were trusted by N others vs N) exhibited a power law distribution. This is not surprising given the self-reinforcing nature of the system, wherein reviews by the most trusted reviewers are prominently featured, causing them to be more frequently read, hence attracting more trusters. Further, because of this, those who rose to prominence early on tended to stay prominent. Both these "winner take all" phenomenon have been studied and explained by the literature on power law networks [9].

The down-side of this phenomenon, as experienced in Epinions, is the emergence of a small set of people who start having an undue influence. To combat this problem, and to enable new-comers to rise to the top, we needed a mechanism for providing exposure for new upcoming members of the community. This was done by time-weighting reviews, positive ratings of reviews and trust statements so that a new review that quickly accumulated even a few positive ratings was rated as highly as an older review which had more positive ratings.

9. PEOPLENET

Epinions, Slashdot, Amazon reviews and other such systems which record explicit relations between different users all have a very tightly coupled relation between the content system and the rating system. In other words, the Web of Trust is very tightly bound to a particular application. Furthermore, they all use highly centralized architectures, wherein all the content and all the trust relations are centrally stored.

In this section, we describe PeopleNet, a system which is based on the following premises:

- Many different applications can share a common Web of Trust, especially if the Web allows trust to be topic specific.
- The Web of Trust itself needs to be distributed, but available as a coherently unified whole to applications via a simple application programming interface. We draw our inspiration for this from the Domain Name System.

PeopleNet is a distributed system, consisting of a number hosts. Each host is the *home* for some number of users. Each PeopleNet user is identified by a URL on his/her home. For example, the author's PeopleNet ID is http://peoplenet.stanford.edu/user/guha. Peoplenet.stanford.edu is the first node on PeopleNet.

Each node contains a set of information about each user, including his relations to other users. In particular, some of these relations are trust/distrust relations. All the information about users is available as an RDF[24] graph in which each user is an RDF resource. Many of these relations will be between users with different homes. So, for example, http://peoplenet.stanford.edu/user/guha trusts http://mayhem.stanford.edu/robm, where mayhem.stanford.edu is a different PeopleNet node one of whose users is robm. Each user adds a javascript URL (Trust) to their bookmark list. When they are on the page corresponding to another user, they can add a trust/distrust relation to that other user by clicking on that bookmark. The trust/distrust can optionally be topic specific. The result is a topic specific Web of Trust which is distributed over a number of machines.

The web of trust can be programmatically accessed by a simple SOAP [11] interface called GetData [28] using which one can ask for values of an property of a user, including who he trusts/distrusts. Applications, such as those described below, can then use this Web of Trust. Many users will not want to make the list of who they trust/distrust public. So, a user can choose to make their trust/distrust list visible only to certain other users (or no one). Of course, if this list is not accessible by other programs, the user will not be helping those who trust him/her very much. To solve this, we distinguish between a trusted application, local to the home, accessing the users Web of Trust from a remote application accessing the Web of Trust. A local application, like the ones described below, can access the user's Web of Trust and make the results available to other, possibly remote instances of the application. This is explained in more detail in section 5.1

We are currently building two prototype applications on top of the PeopleNet infrastructure.

9.1 Recommended Links

In the early days of the Web, many users would surf the web looking for new and cool sites. With the Web having gotten so big, few people do this anymore. Instead, we rely on our friends sending us (usually via email) pointers to interesting sites. Occasionally, when we ourselves run across a site that is sufficiently interesting, we might send it to some of our friends. Unfortunately, since this process of sending email is so disruptive to the current activity (it requires picking a menu item, typing a set of email addresses, writing a cover letter, ...) that it happens quite rarely. The Recommended Links application is intended to make it easy to recommend a page and have this recommendation spread out to those who trust the recommender, and so on. The user interface to the application is a javascript URL (Recommend) which is part of the recommender's browser's bookmark list. When the recommender comes across a site s/he likes, s/he clicks on the bookmark, which pops up a menu from which s/he selects a topic under which s/he recommends it. A PeopleNet user can get a list of links recommended to her by those in her Web of Trust by visiting her page on her home node.

The Recommended Links (RL) application, which runs on each PeopleNet home node, periodically computes a list of recommended links for every user on that home, based on his/her Web of Trust, using the approach of sections 6 and 7. Particular nodes are free to either augment the approach of those sections or even use completely different approaches.

In a distributed setting, where the trust links are distributed over an arbitrary set of nodes, it is no longer possible to compute the global TrustRanks. So, each node computes its own list TrustRanks. In addition to trust relations between users, there can also be trust relations between nodes. Two nodes which trust each other can pool trust information to compute better global TrustRanks.

Many users might not be willing to expose the list of who they trust/distrust, which makes it difficult to use the approach of sections 6 and 7. So, we modify that approach as follows. As the RL application tries to traverse the user's Web of Trust, if it encounters someone (A_i) who is not willing to reveal who they trust, it asks the RL application running on the node (i.e., $A_i's$ home) for that users list of recommendations. It then makes the approximation that these recommendations are the items positively rated by A_i and that A_i does not trust anyone. A small modification of this approach would be to pick only the top N recommendations from A_i. Note that in the presence of distrust, a user hiding her trust can result in bad recommendations for those who trust her. E.g., Jane trusts Fred, who trusts Joe. But Jane distrusts Joe. Fred hides his Web of Trust from Jane and so Jane can only see which items Fred recommends, without knowing why he recommends those items. So, Fred might recommend an item because it is highly rated by Joe. This item will show up in Jane's recommendations. However, if Jane had known that Fred recommended this item because of Joe's rating, she might ignore that, keeping that item out of Jane's recommendations. This non-monotonicity is common in the real world. Often we take the recommendation of someone we trust at face value. However, when we come to know why s/he recommended it, we might decide to ignore that.

This kind of propagation of recommendations is quite similar to how many Peer to Peer ([15], [14], [8]) networks behave.

9.2 For the Machine Readable Web

Over the last few years, activities such as XML Web Services [11], the Grid [4] and the Semantic Web [10] have gained significant attention. Though these different activities have slightly different focii, with Web Services dealing with the invocation, relaying and composition of services, the Grid with distributing computation and the Semantic Web with the representation of data, they all share the goal of creating a web of machine-readable data.

An important aspect of this web of data is that different sites may contribute data about a particular resource. So, for example consider the cellist Yo-Yo Ma. Many different sources have data about Yo-Yo Ma. Amazon and CDNow have data about his albums, Ebay has data about auctions related to these albums, TicketMaster has data about his concert schedule, AllMusic has data about where he was born (Paris), and so on. Each of these sites can publish data about Yo-Yo Ma without getting permission from any centralized

authority, i.e., they can all extend the cumulative knowledge about any resource in a distributed fashion. This *distributed extensibility* is a very important aspect of this new web.

Of course, this feature leads to problems of its own. In a world where anyone can publish anything, a lot of what gets published cannot be trusted. On the HTML web, we, as humans, use our intelligence, invoking concepts of brand, who recommended what, etc. to decide whether to believe what a web site says. Programs, on the other hand, being relatively unintelligent, do not have recourse to all these facilities to decide whether to believe the data from a new site. This is an important problem that needs to be addressed.

We cannot expect programs to be able to make the kind of trust judgments (about sites) that we as humans make. Consequently, at some level, we have to create registries containing information which our programs can use, that specify which sites to trust about which kinds of data. One approach is to rely on centralized registries which ascertain the quality and trustworthiness of sites providing data. As our experience with the HTML web and centralized registries such as Yahoo [30] shows, such approaches don't scale.

Another approach, which complements centralized registries, is to rely on a network of local registries created by programmers, which share their entries through a web of trust *between registries*. In this model, a programmer adds some entries (on which sites should be queried for which kinds of data) to her local registry. In addition, she also specifies which other registries may be trusted.

When a query arrives from a program, the registry consults its local entries and if no match is found, forwards the query to its trusted registries. As a result, the work done by any of those in the programmers web of trust can be exploited by the program. This is the approach being taken by the TAP [28] system, a framework for building applications on the Semantic Web, for determining what information can be trusted.

10. CONCLUSIONS AND FUTURE WORK

The experience with Epinions and the success of systems like Slashdot suggest that Open Rating Systems together with the concept of "Web of Trust" can be very useful in locating high quality content. We also believe that the this concept will turn out to be very important in the context of the emerging machine readable web.

Our future work has two directions. On the implementation front, we are trying to create an open distributed platform around PeopleNet. In addition to the applications described above, it should be possible to build many other, as yet unforeseen applications. On the theoretical side, we are trying to get a better understanding of the different kinds of semantics associated with trust and distrust statements and their implications for algorithms which should be used. We are also trying to formulate mathematical models of the various social phenomenon which arise in real Webs of Trust.

Another line of work is in understanding the behavior of these kinds of systems when the are perturbed. In particular, we are interested in understanding how such systems may be perturbed to shake them out of the local maxima they can get stuck in due to the effects of the power law distributions that naturally occur in them.

Finally, it is important to realize that these are dynamic systems which can change quite rapidly. Understanding the time-dependent properties of such systems and exploiting these properties is another potentially useful line of inquiry.

11. ACKNOWLEDGEMENTS

I would like to thank the original Epinions team for the work in implementing the system described here. I would also like to thank Epinions for the permission to write about it and for providing me with the requisite data. I would like to thank IBM Almaden for support and help. I would also like to thank Rob McCool and other members of the Knowledge Systems Laboratory at Stanford, where PeopleNet is being built.

12. REFERENCES

[1] Advogato. `http://www.advogato.com/`.

[2] Amazon. `http://www.amazon.com/`.

[3] epinions. `http://www.epinions.com/`.

[4] Global grid forum. `http://www.gridforum.org/`.

[5] The open directory project. `http://www.dmoz.org/`.

[6] Slashdot. `http://www.slashdot.org/`.

[7] A. Abdul-Rahman and S. Hailes. Relying on trust to find reliable information.

[8] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman. Search in power law networks.

[9] A. Barab'asi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web, 1999.

[10] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2000.

[11] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winder. Simple Object Acceess Protocol. `http://www.w3.org/TR/SOAP/`, May 2000.

[12] M. Burrows, M. Abadi, and R. Needham. A logic of authentication, from proceedings of the royal society, volume 426, number 1871, 1989. In *William Stallings, Practical Cryptography for Data Internetworks, IEEE Computer Society Press, 1996*. 1996.

[13] J. Coleman. *Foundations of Social Theory*. Harvard University Press, Cambridge, Mass., 1990.

[14] B. F. Cooper and H. Garcia-Molina. Modeling and measuring scalable peer-to-peer search networks. `"http://dbpubs.stanford.edu/pub/2002-43"`.

[15] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. `"http://www-db.stanford.edu/ crespo/publications`

[16] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *World Wide Web*, pages 613–622, 2001.

[17] P. Feisthammel. Pgp web of trust. `http://www.rubin.ch/pgp/weboftrust`.

[18] U. Frendrup, H. Httel, and J. Nyholm. Modal logics for cryptographic processes.

[19] M. Gladwell. *The Tipping Point, How Little Things Can Make a Big Difference*. Little Brown, February 2000.

[20] Google. `http://www.google.com`.

[21] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, pages 517–526, 2002.

[22] D. Kahneman, P. S. P., and A. Tversky. *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press, April 1982.

[23] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 668–677, San Francisco, CA, 1998.

[24] O. Lassila and R. Swick. Resource description framework (rdf) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

[25] Looksmart. `http://www.looksmart.com`.

[26] B. Misztal. *Trust in Modern Societies: The search for the Bases of Social Order*. Polity Press, Cambridge MA, 1996.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Computer Science Department, Stanford University, 1998.

[28] R.Guha and R. McCool. Tap: Towards a web of data. `http://tap.stanford.edu/`.

[29] P. Sztompka. *Trust. A Sociological Theory*. Cambridge University Press, 1999.

[30] Yahoo. `http://www.yahoo.com`.