A Probabilistic Trust Model for Semantic Peer to Peer Systems

Gia Hien Nguyen, Philippe Chatalic, Marie-Christine Rousset

A Probabilistic Trust Model for Semantic Peer to Peer Systems

G.H. Nguyen
LIG, Univ. Grenoble 1
681 rue de la passerelle
BP 72
St Martin d'Heres, F-38402
gia-hien.nguyen@imag.fr

P. Chatalic
LRI, Univ. Paris-Sud
CNRS, UMR 8623 / INRIA Saclay
4, rue Jacques Monod
Orsay F-91893
philippe.chatalic@lri.fr

M.C. Rousset
LIG, Univ. Grenoble 1
681 rue de la passerelle
BP 72
St Martin d'Heres, F-38402
marie-christine.rousset@imag.fr

ABSTRACT

Semantic peer to peer (P2P) systems are fully decentralized overlay networks of people or machines (called peers) sharing and searching varied resources (documents, videos, photos, data, services) based on their semantic annotations using ontologies. They provide a support for the emergence of open and decentralized electronic social networks, in which no central or external authority can control the reliability of the peers participating to the network. This lack of control may however cause some of the results provided by some peers to be unsatisfactory, because of inadequate or obsolete annotations.

In this paper, we propose a probabilistic model to handle trust in a P2P setting. It supports a local computation and a simple form of propagation of the trust of peers into classes of other peers. We claim that it is well appropriate to the dynamics of P2P networks and to the freedom of each peer within the network to have different viewpoints towards the peers with which it interacts.

1. INTRODUCTION

Recently, P2P systems have received considerable attention because their underlying infrastructure is appropriate to scalable and flexible distributed applications over Internet. In P2P systems, there is no centralized control or hierarchical organization: each peer is equivalent in functionality and cooperates with other peers in order to solve a collective task. P2P systems have evolved from simple keyword-based file sharing systems like Napster [3] and Gnutella [2] to semantic data management systems like Edutella [20], Piazza [16], SomeWhere [8], or semantic social networks like FOAF [4].

In this paper, by *semantic peer to peer systems* we refer to fully decentralized overlay networks of people or machines (called peers) sharing and searching varied resources (documents, videos, photos, data, services) based on their semantic annotations using ontologies. Indexing resources based on the terms of an ontology enables more accurate information retrieval and query answering than indexing by keywords of a textual annotation.

In a P2P view, each peer is free to use its own ontology to annotate the resources that it stores locally and that it agrees to share with others. For example, SomeWhere [8], an infrastructure for deploying semantic P2P systems, is based on simple personalized ontologies (e.g., taxonomies of atomic classes).

In such semantic P2P systems, no user imposes to others his own ontology but logical mappings between ontologies make possible the creation of a network of people in which personalized semantic marking up of data cohabits nicely with a collaborative exchange of data. The mappings are exploited during information retrieval or query answering for query reformulation between peers.

Semantic P2P systems provide a support for the emergence of open and decentralized electronic social networks, in which no central or external authority can control the *reliability* of the peers participating to the network. Some of the peers may have (accidentally or deliberately) annotated some (or all) of the resources they have in an inappropriate way or resource content may evolve over time in such a way that prior annotations are not suitable anymore.

As a consequence, a user of such semantic P2P systems is not always satisfied with the answers returned to his queries. After a while, when having received enough answers, he may naturally be inclined to trust/distrust further answers obtained by those sources that have contributed to obtain previous good/bad results. The proposal of an adequate model to assess the level of confidence that a peer may have in a given answer is thus an important issue.

Trust is now widely acknowledged as an important factor when considering networks of autonomous interacting entities and notably in the context of the semantic web. When referring to the notion of trust, T. Berners-Lee advocates with his "Oh Yeah" button [10] for a user to be able to check for reasons why he/she could be confident with a returned answer. He suggests that such justifications would be presented under the form of partial logical proofs.

Several proposals have already been done that do not all

DAMAP'08, March 25th, 2008, Nantes, France.

share the same meaning for the notion of trust (see [23] and [9] for surveys). Many of them are user/agent/peer centered and rely on the (sometimes implicit) assumption that all (or communities of) peers share similar implicit goals. Trust is then closely related to the notion of reputation in a community. This is the case in the context of E-commerce, auction or recommendation systems. In full P2P architectures, computing a global reputation of peers is a difficult task that leads to sophisticated machineries for propagating trust values within a network of peers. In many cases such approaches proceed in an ad'hoc way and lack a clear mathematical semantics.

In semantic P2P systems, the result of a search or of a query is a set of resources (e.g., documents) that have been annotated by one or several peers. We consider that such semantic annotations constitute some form of *logical justifications*. Correlating such justifications with evaluations of previously returned answers can gradually give the user some evidence for estimating in which respect she may be confident with such annotations.

In the context of semantic P2P systems, peers may correspond to different points of view. For this reason, we rather promote a way to compute trust values based on direct experiences between peers. However, when this information is not available or not sufficient, we can use a simple mechanism of trust propagation in order for a peer to take advantage of the experiences of other peers.

We also argue for a finer grained context sensitive approach to trust in order to take into account the fact that for answers annotated by a same peer, the trust into these answers may vary according to their annotations. For instance, a peer may have high quality annotations on film genres but poor annotations on music genres.

We propose a probabilistic model to handle trust in a P2P setting. The trust in resources justified by some annotation L, from the viewpoint of a peer P', is defined as the probability that a resource annotated by L satisfies the peer P'. It can be estimated using the Bayesian approach to statistics from the evolving number of resources annotated by L and observed by the peer P', that satisfy or do not satisfy the peer P'. Our model supports a local computation of the trust of peers into classes of other peers. We claim that our model is well appropriate to the dynamic of P2P networks and to the freedom of each peer within the network to have different viewpoints towards the peers with whom he/she interacts.

The paper is organized as follows. In section 2 we present the intuition behind the notion of trust that we propose and illustrate with an example how direct feedbacks on past answers can be involved. In section 3, we give a probabilistic definition of the trust of a peer into a resource annotated by classes of other peers. We show how it is possible to estimate it by means of the Bayesian approach to statistics, which consists in taking into account feedbacks on past experiences. In section 4 we propose a simple form of trust propagation, that can be used when direct feedbacks is missing or insufficient to take advantage of other peers' experience. The section 5 is an extensive comparison of our approach with related work on the handling of trust in P2P or ad-hoc networks. We conclude in section 6.

2. PRELIMINARIES AND ILLUSTRATIVE EXAMPLE

In the following we consider a network of semantic peers $\mathcal{P} = (P_i)_{i=1..n}$. Each peer P_i uses its own ontology, expressed on its own vocabulary V_i, for describing and structuring its knowledge as well as for annotating its resources. A class $C \in V_i$ of a peer P_i is referred by $P_i{:}C$ or simply by C when no confusion is possible. Peers are connected each other by means of *mappings*, corresponding to logical constraints linking classes of different peers. Users ask queries to one of the peers, using the vocabulary of this peer. When processing a query, the reasoning propagates from one peer to other peers thanks to those mappings. The mappings are exploited during information retrieval or query answering for query reformulation between peers.

For example, let us consider a semantic P2P system sharing movies based on semantic annotations, where P_1 organizes his video resources according to their genres (*Suspense*, *Action*, *Animation*), and P_2 organizes his films based on the actors playing in the movies (Bruce Willis, Jolie). While having different views for classifying movies, P_1 and P_2 can establish some mappings between their two classifications. For example, they can agree that the class *BruceWillis* of P_2 (denoted by $P_2{:}BruceWillis$) is more specific than the class *Action* of P_1 (denoted by $P_1{:}Action$). It will result into the mapping $P_2{:}BruceWillis \sqsubseteq P_1{:}Action$. Similarly, P_1 and another peer P_3 can have established the mapping $P_1{:}Action \sqcap P_1{:}Suspense \sqsubseteq P_3{:}Thriller$ between their two classifications, in order to state that the category named *Thriller* by P_3 is more general than what P_1 classifies as both *Action* and *Suspense*. As a result, the movies that are classified by P_1 as *Suspense* and by P_2 as *BruceWillis* are returned as answers to the query *Thriller* asked by the user at the peer P_3.

We assume that each resource r returned as an answer to some query is associated with a label $L(r) = C_{i1} \dots C_{iL}$ corresponding to its logical justification. $L(r)$ is a set of classes of the vocabularies of (possibly different) peers known to annotate the resource r and supposed to characterize a *sufficient condition* for r to be an answer. Any other resource annotated in the same way is thus equally supposed to be an alternative answer to the query. We also assume that the classes used in labels are *independent* in the sense that for any two classes of a justification, none of them is a subclass of the other. This important assumption means that for a returned answer, the only classes that appear in its justifications are those corresponding to most specific classes of the network.

Finally we assume that the user, when querying a peer P_i, is randomly asked to evaluate some of the returned answers as *satisfying* or *not satisfying* and to store the result of this evaluation in a local *observation database* $\mathcal{O}_i$. Each evaluation is recorded into $\mathcal{O}_i$ as a pair $S.L$ or $\overline{S}.L$, where S (resp. $\overline{S}$) denotes the user satisfaction (resp. unsatisfaction) and L is the label of the evaluated resource. Note that, because labels correspond to *sufficient* conditions, any observation with a label L' such that $L \subseteq L'$ is also an observation that satisfies the condition L and therefore may be considered as relevant for L.

Label (L)	$O_1^+(L)$	$O_1^-(L)$
P_2:$MyActionFilms$	30	6
P_2:$MyCartoons$	3	15
P_4:$ScienceFiction$	14	14
P_5:$Italian$ P_5:$Western$	0	6
P_6:$AnimalsDocum$	8	2
P_7:$JeanRenoir$	22	11
P_8:$Bollywood$	6	35

Table 1: Summary of Peter's observations at P_1

DEFINITION 1 (OBSERVATION RELEVANT TO A LABEL L). *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label. An observation of $\mathcal{O}_i$ is said to be* relevant to L *if and only if its label contains all classes of L. The number of satisfying and unsatisfying observations of P_i that are relevant to L are respectively denoted by:*

$$O_i^+(L) = |\{S.L' \in \mathcal{O}_i / L \subseteq L'\}|$$

$$O_i^-(L) = |\{\overline{S}.L' \in \mathcal{O}_i / L \subseteq L'\}|$$

These two numbers summarize the past experience of the peer P_i relevant to the label L, i.e. of the evaluated resources justified by at least the classes of L.

For instance, suppose that Peter is the user querying the peer P_1. After a number of answers have been evaluated, Peter's past experience may be summarized as in table 1.

Among all the resources evaluated by Peter and annotated with the class $MyActionFilms$ of the peer P_2, 30 have been considered as satisfactory and 6 as not satisfactory. For the same peer P_2, only 3 out of 18 evaluated resources tagged by $MyCartoons$ were positive. Similarly all evaluated resources annotated with both $Italian$ and $Western$ by P_5, obtained negative feedbacks.

One can clearly understand that for the next queries, given this past experience, Peter would intuitively prefer (or more trust) answers classified as P_2:$MyActionFilms$ to those annotated by P_2:$MyCartoons$ or by P_8:$Bollywood$. This is because the proportion of positive feedbacks obtained so far is higher for P_2:$MyActionFilms$ than for the others.

Each peer P_i can progressively update its observation database $\mathcal{O}_i$, as new answers are evaluated, and refine the trust it has towards answers justified by the different observed labels. The level of trust can vary according to the justification. In the following, we present a Bayesian model of trust which makes possible to estimate the trust towards a label from the observation database.

3. BAYESIAN MODEL AND ESTIMATION OF TRUST

Given a label L, let X_{iL} be the binary random variable defined on the set of resources annotated by L as follows:

$$X_{iL}(r) = \begin{cases} 1 & \text{if the resource } r \text{ is satisfying for } P_i \\ 0 & \text{otherwise} \end{cases}$$

We define the trust of a peer P_i towards a label L as the probability that the random variable X_{iL} is equal to 1, given the observations resulting from the past experiences of P_i.

DEFINITION 2 (TRUST OF A PEER TOWARDS A LABEL L). *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label, the trust $T(P_i, L)$ of P_i towards L is defined as follows:*

$$T(P_i, L) = Pr(X_{iL} = 1 | \mathcal{O}_i)$$

The following theorem provides a way to estimate the trust $T(P_i, L)$ of a peer P_i towards a label L, and the associated error of estimation.

THEOREM 1. *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label. After $O_i^+(L)$ satisfying and $O_i^-(L)$ unsatisfying observations relevant to L have been performed, $T(P_i, L)$ can be estimated to*

$$\frac{1 + O_i^+(L)}{2 + O_i^+(L) + O_i^-(L)}$$

with a standard deviation of

$$\sqrt{\frac{(1 + O_i^+(L)) \times (1 + O_i^-(L))}{(2 + O_i^+(L) + O_i^-(L))^2 \times (3 + O_i^+(L) + O_i^-(L))}}$$

Sketch of the proof: It follows from a well known result in probabilities of the application of the Bayes rule to random variables following a Bernoulli distribution the parameter of which is unknown. The Bernoulli distribution is the simplest discrete distribution having two possible outcomes labeled by 1 and 0 in which 1 ("success") occurs with probability p and 0 ("failure") occurs with probability $1-p$, where p is called the parameter of the distribution.

In our setting, it is quite natural to consider that the random variable X_{iL} follows a Bernoulli distribution the parameter of which is precisely the probability of satisfaction of P_i by answers annotated with L, which is unknown and requires to be estimated.

In the Bayesian approach to statistics such unknown parameter is modeled as a random variable. It can be shown [17](page 336) that if its prior distribution is a Beta distribution of parameters α and β, then its posterior distribution, after n observations $o_1, \ldots, o_n$ (each o_i is either 1 or 0) of a random sample of the random variable X_{iL}, is also a Beta distribution, the parameters of which are $\alpha + \sum_{i=1}^{n} o_i$ and $\beta + (n - \sum_{i=1}^{n} o_i)$.

By setting the prior distribution of X_{iL} to be the uniform distribution, which corresponds to the Beta distribution of parameters 1 and 1, we obtain that the posterior distribution of X_{iL}, given the observations $\mathcal{O}_i$ follows a Beta distribution of parameters $1 + O_i^+(L)$ and $1 + O_i^-(L)$, since $O_i^+(L) = \sum_{i=1}^{n} o_i$ and $O_i^-(L) = n - \sum_{i=1}^{n} o_i$.

In probability, it is usual to estimate the value of a random variable by its mean and the precision of the estimate with the standard deviation around this value. For a Beta distribution this leads to the above expression. □

By applying Theorem 1 to the Peter's observations summarized in Table 1, we may characterize the estimations for the trust of the peer P_1 towards the labels observed so far by P_1. Table 2 describes the corresponding estimations with their associated standard deviation. It is important to notice that this standard deviation is very small.

Label (L)	Estimated trust of P_1 towards L	Standard deviation
$P_2 : MyActionFilms$	0.815	0.062
$P_2 : MyCartoons$	0.2	0.087
$P_4 : ScienceFiction$	0.5	0.089
$P_5 : Italian$ $P_5 : Western$	0.125	0.11
$P_6 : AnimalsDocum$	0.75	0.12
$P_7 : JeanRenoir$	0.657	0.079
$P_8 : Bollywood$	0.162	0.055

Table 2: Estimated trust of P_1 towards the labels of Table 1

4. PROPAGATION OF TRUST

The Bayesian model presented in the section 3 relies on direct interactions between peers in order to compute trust. The numbers of resources evaluated as good/bad are sufficient to compute the trust of a peer P_i towards resources annotated with a label L. In addition to the simplicity of the formula of Theorem 1 for computing the trust, the advantage of the model is that it provides also a simple way to compute the minimum number of experiences that are required for guaranteeing a good precision of the estimation of the trust. A direct application of the formula of Theorem 1, that characterizes the standard deviation of the trust estimation shows that the evaluation of a sample of 22 answers annotated with a given label is enough to have an estimation of the trust of a peer towards that label with a low estimation error (less than 0.1) and this, without using any kind of trust propagation.

When the observation database contains only a few observations relevant to a label, several choice are possibles. One may either keep a strict point of view, and trust only labels on the basis of the direct experience. Note that in the case where there is strictly no relevant observation, the estimate is 1/2, which reflects a neutral point of view. One may also, as often in real life, take some advice from more experienced peers. In that case, solicited peers' observations may prove useful to compensate for the lack of local relevant observations, provided that they use similar evaluation criteria. Instead of propagating trust between peers, our approach consists in *propagating the pairs of numbers used for computing trust*. Propagating two numbers instead of one does not represent a significant overhead. Yet, it has the significant advantage of providing a well-founded way to compute a joint trust using the same Bayesian model as the one presented in section 3. Instead of using an ad-hoc aggregation function for combining local coefficients of trust, the numbers $O^+_{i1}(L) \dots O^+_{il}(L)$ (respectively $O^-_{i1}(L) \dots O^-_{il}(L)$) coming from solicited peers $P_{i1} \dots P_{il}$ are cumulated to compute the joint trust of the subset $P_{i1} \dots P_{il}$ towards L, by applying the formula of Theorem 1. In the absence of any local relevant observations, this comes down to using observations of other peers as an information to set the prior distribution to a Beta distribution which is not uniform.

Different strategies are possible to gather on the querying peer the relevant information from the solicited peer's observations.

- The *lazy* strategy consists in waiting for getting some answer justified by a label L and then asking one or several trusted neighbors for their direct feedbacks about the label L. Since it applies after the obtention of answers, such a strategy can be used as a post-precessing and does not require to change the query evaluation mechanism itself. As a consequence it can be applied to different kinds of semantic P2P systems, provided they are able to justify answers by means of such labels (e.g. sets of independant semantic annotations).

- The *greedy* strategy consists in collecting the direct feedbacks likely to be relevant (i.e., concerning the classes in the annotation being built) during the query processing. It thus requires some adaptation of the query answering algorithm. In a system like SomeWhere [7], the DeCa algorithm [8] is first used to infer, from the ontologies and mappings, all the possible reformulations (i.e. rewritings) of the initial query into conjunctions of extensional classes (i.e. containers of instances) $C_1, \dots, C_n$. Each instance in $C_1 \sqcap \dots \sqcap C_n$ is then produced as an answer, $C_1 \dots C_n$ being the semantic annotation justifying it. The DeCa algorithm can be slightly modified in order to convey, when transmitting back rewritings from a queried peer P to the querying peer P', those feedbacks likely to be relevant. When a rewriting $C_j \sqcap \dots \sqcap C_m$ is transmitted from P to P' within a message, P uses that message to convey its direct observations $(O^+(L), O^-(L))$ for all labels L containing the classes of the rewriting. By construction, those classes will be part of the annotation of an answer. Therefore, observations relevant to these classes may be relevant for computing (if needed) the joint trust towards the labels annotating answers returned to the peer the initial query is issued from. Note that this strategy leads to combining feedbacks from the very peers that have contributed to obtain an answer. Those peers may thus be considered as naturally relevant for obtaining appropriate feedbacks. However, such sets of peers are determined at query time and may vary according to the query and the returned answer.

5. RELATED WORK

Modeling and handling reliability of agents is a key issue which has been studied in many electronic applications involving communities of agents. Reputation systems have been proposed to address this problem. They use (satisfaction or complaint) feedbacks on past transactions between agents. They differ on (i) the way they envision reliability (as a local versus a global notion) and (ii) the manner they aggregate feedbacks for computing it.

The global vision of reliability leads to defining the *reputation* of an agent, judged trustworthy or not according to the feedbacks returned by the agents which have been interacting with it. In contrast, in some settings, it is reasonable to envision the *trust* of an agent towards another one as a local and relative notion.

The computation of both reputation and trust relies on aggregating past experiences. We also distinguish the existing approaches depending on whether the aggregation is

based on a probabilistic model or not.

5.1 Non probabilistic models of reputation

One of the most illustrative example of reputation systems is Ebay [1], the largest online auctioning system. After each transaction, the buyer and the seller file a (positive, negative or neutral) feedback about each other. Feedbacks concerning the users are centralized into a server to compute 2 notes for each user. The first note is the difference between the number of positive feedbacks and the number of negative feedbacks. The second note is the percentage of positive feedbacks w.r.t. the total feedbacks about this user.

While the method used in Ebay relies on a central server, [6] is one of the first approaches for computing reputation of peers in a P2P setting. It is based on a distributed storage of feedbacks and an ad-hoc aggregation of the number of feedbacks. When two peers P and Q interact, each of them files a feedback about the other. The type of feedbacks are complaints (negative feedbacks). Let $c(P,Q)$ be a complaint made by P about Q, Q can also make a complaint about P, $c(P,Q)$. Because of this, it is not possible for a third peer R to know which from P or Q is trustworthy. To solve this, the feedbacks are stored by a so-called P-Grid structure [5] with a number of replicas for each complaint. Then, when a peer R assesses the global trustworthiness of a peer P, noted as $T(P)$, it searches the P-Grid structure to retrieve all the complaints concerning P. $T(P)$ is defined as the product of the number of complaints made by P with the number of complaints made about P. High values of $T(P)$ mean that P is not trustworthy. In this model, the threshold defining that $T(P)$ is high must be set in an ad-hoc way. In addition, the interpretation of $T(P)$ equal to 0 is not clear.

5.2 Probabilistic models of reputation

The PageRank algorithm [21, 11] of Google is based on a probabilistic model for computing the global reputation of web pages. A page has high rank if it is referred by other pages that themselves are highly ranked. The rank $R(p)$ of a page p is defined by the formula: $R(p) = d.\frac{1}{T} + (1-d)\sum_{i=1}^{k}\frac{R(p_i)}{C(p_i)}$ where $d \in$ [0,1], T is the total number of pages, $p_1, ...p_k$ are the pages referring to p, each has rank $R(p_i)$ and $C(p_i)$ is the number of links out of p_i. The probabilistic interpretation of $R(p)$ is based on the meaning of the *random walk model*. Let us consider a Web surfer who wandering the Web. Suppose first that he is on the page p. At each step, the surfer may jump to one of the page referred by p with probability $(1-d)$ or may jump to any random page on the Web with probability d. The probability that the random surfer visits a page is thus equal to its rank. This means that pages with high rank are more likely to be visited than pages with low rank.

The formula used in the PageRank algorithm is recursive as well as the formula used in [18] to compute the global reputation of peers participating in a P2P system. The global reputation of a peer P_j is given by aggregating the local reputation values c_{ij} assigned to P_j by other peers P_i, weighted by the global reputations of the P_is. Each c_{ij}, computed by P_i, is the difference between the number of direct positive feedbacks and the number of direct negative feedbacks. In this model, all the peers participate simultaneously into the computation until the convergence of the global reputation value for each peer.

[15] has proposed a different probabilistic way to compute the reputation of peers, based on maximum likelihood estimation. The reputation of a peer P_j is its innate probability μ_j of performing honestly in its interactions with others. The technique consists in maximizing the likelihood function, i.e. the function of μ_j given the value of the observations (feedbacks). The feedbacks concerning a peer are binary [positive, negative], and are stored by a P-Grid structure. Note that each peer may lie with probability k when making feedbacks about others. When a peer P_i computes the reputation of a peer P_j, it explores the P-Grid structure to retrieve feedbacks about P_j. For each feedback received by P_i, the probability that this feedback is a truthful one is a function f of μ_j and k. The likelihood function of μ_j, given all the retrieved feedbacks, is thus the product of the functions f. Then, while supposing that k is fixed, the maximum likelihood estimation procedure amounts to find the value of μ_j that maximizes this formula, i.e. to find the most likely value of μ_j.

5.3 Non probabilistic models of trust

[22] has proposed a reputation model for gregarious societies, i.e. where people gather into groups. The trust for an agent b belonging to a group B, when computed by an agent a belonging to a group A, is the result of an intuitive aggregation of all the knowledge of A about B. This implies that, if an agent a' belongs to a different group A', the trust of b computed by a is different to the one computed by a'. The aggregation of the knowledge of A about B combines (i) the local trust of a for b, $R(a,b)$, (ii) the trust of a for B, $R(a,B)$, (iii) the trust of A for b, $R(A,b)$ and (iv) the trust of A for B, $R(A,B)$. $R(a,b)$ is the sum of each a's feedback about b (in [-1,1]), weighted by a time-dependent function that gives higher values for more recent feedbacks. $R(a,B)$ is the sum of local trust of a towards each agent b_i of B. $R(A,b)$ and $R(A,B)$ are respectively the sum of local trust of each a_i of A for b and for each b_i of B, weighted by the importance of each a_i in A. Finally the trust of b when computed by a is equal to $R(a,b)\ \xi(a,b) + R(a,B)\ \xi(a,B) + R(A,b)\ \xi(A,b) + R(A,B)\ \xi(A,B)$, where the $\xi(.,.)$ are weights reflecting the importance of each source of opinion and are chosen such that $\xi(a,b) + \xi(a,B) + \xi(A,b) + \xi(A,B) = 1$.

[14] has proposed an approach based on votes for a peer to choose resources to download in P2P networks. Each peer keeps a binary feedback database towards resources, and a binary feedback database towards the peers having provided him resources. For estimating the quality of a given resource, a peer asks all its neighbors to vote. Before downloading the resource, P does the same procedure to choose the best peer among those offering this resource.

5.4 Probabilistic models of trust

The Bayesian estimation approach to statistics has been adopted in [19, 12]. In these works, the Beta distribution has been used as a prior distribution of the unknown quantity. When a peer P_i computes its trust towards a peer P_j, it needs to collect the feedbacks about P_j from a set S of other peers. In [19], S is the set of peers on the path from P_i to P_j. In [12], S is the set of neighbors of P_i. In those cases, the set S is dependent on the peer P_i computing its trust for P_j.

	Probabilistic	Non-probabilistic
`Reputation`	[21, 11, 18, 15]	[6, 1]
`Trust`	[19, 12], our model	[22, 14]

Table 3: Comparison of our model with some related work

Our approach is also based on a Bayesian estimation of trust. Compared to [19, 12], it gives a priority to direct experiences. It is only when this information is not available or not sufficient that we collect observations from a set S of other peers. In the proposed *greedy* strategy, the set S depends on the query and the answer. The observations are collected *during* the query rewriting procedure as opposed to the usually used *lazy* strategy in which the observations are collected *after* the answers have been obtained. An other difference with [19, 12] is that we propose a finer grained definition of trust that allows to distinguish trust towards different annotations.

5.5 Summary

We summarize how our approach and the existing ones can be classified and compared to each other by the Table 3. Our model of trust is based on a probabilistic interpretation. The trust of a peer into an answer is defined as the probability that this answer satisfy this peer given its label. This trust is relative to the peer computing it, even in the case where the computing peer needs to collect observations from other peers.

6. CONCLUSION AND PERSPECTIVES

We have presented an approach for the modeling of trust in the context of semantic P2P systems. It allows each peer to evaluate the trust it has towards resources depending on their semantic annotations. Because it relies on few assumptions, it is applicable to a wide range of P2P systems. Like SomeWhere[8], many semantics P2P systems answer queries in two steps, first by reformulating the query in terms of conjunctive rewritings and then by evaluating those rewritings. Derived from the ontologies, the terms of rewritings, corresponding to classes, are semantic annotations. Our approach then provides a useful framework for ranking the different rewritings according to their trust level and to evaluate in the first place those for which the probability to obtain satisfying answers is the highest. The benefit may be high when the number of rewritings is important.

The granularity of the trust considered here is at the level of annotations. Defining a more abstract peer-based notion of trust is however straightforward since it would just amount, to aggregate the available information concerning all the classes of a same other peer.

In the current approach we assume each peer has no prior knowledge on other peers before interacting with them. When prior knowledge exists, we can exploit it to set an informed prior distribution. For instance, it has been shown in [13] how inconsistencies can be detected as new peers join the network. The information gathered during the detection of such inconsistencies can then be used as a basis to propose more informed prior distributions.

One of the objectives of reputation systems is the detection and handling of malicious agents in an electronic environment. In a P2P system, a peer can be malicious by providing to other peers virus-affected resources, or by simply lying when reporting its feedbacks about others. In our model, when a peer has enough direct experiences, it does not have to rely on other peers and thus avoid malicious peers. When it has to rely on observations of other peers for estimating its trust towards a label, it is reasonable to assume that the number of malicious peers is small. Therefore, it is possible to either increase the number of peers to solicit to get observations (in order to decrease the impact of wrong observations coming from few peers) or to discard the peers the observations of which change a lot the joint trust (they are likely to be malicious).

7. REFERENCES

[1] Ebay. http://www.ebay.com.

[2] Gnutella. http://gnutella.wego.com.

[3] Napster. http://www.napster.com.

[4] The Friend of a Friend (FOAF) project. http://www.foaf-project.org.

[5] K. Aberer. P-Grid: A self-organizing access structure for P2P information systems. *CoopIS 2001, Lecture Notes in Computer Science*, 2172:179–194, 2001.

[6] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In H. Paques, L. Liu, and D. Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 310–317. ACM Press, 2001.

[7] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, and L. Simon. Somewhere in the semantic web. In *PPSWR*, pages 1–16, 2005.

[8] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, and L. Simon. Distributed reasoning in a peer-to-peer setting: Application to the semantic web. *Journal of Artificial Intelligence Research*, 25:269–314, 2006.

[9] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics*, 5(2):58–71, 2007.

[10] T. Berners-Lee. Cleaning up the user interface. http://www.w3.org/DesignIssues/UI.html.

[11] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[12] S. Buchegger and J. L. Boudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks, 2003. In Proc. WiOpt'03 (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks).

[13] P. Chatalic, G. H. Nguyen, and M.-C. Rousset. Reasoning with inconsistencies in propositional peer-to-peer inference systems. In G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, editors, *Proc. of ECAI'06*, pages 352–356, Riva del Garda, Italy, Aug. 29 - Sept. 1 2006. IOS Press.

[14] E. Damiani, S. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation–based approach for choosing reliable resources in

peer–to–peer networks, 2002. In 9th ACM Conf. on Computer and Communications Security.

[15] Z. Despotovic and K. Aberer. Maximum likelihood estimation of peers' performance in p2p networks. In *The Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[16] A. Y. Halevy, Z. Ives, I. Tatarinov, and P. Mork. Piazza: data management infrastructure for semantic web applications. pages 556–567. ACM Press, 2003.

[17] M. H.DeGroot and M. J.Schervish. *Probability and Statistics*. Addison Wesley, 2002.

[18] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks, 2003. In Proceedings of the Twelfth International World Wide Web Conference.

[19] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. 35th Hawaii International Conference on System Science (HICSS), 2002.

[20] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, and al. Edutella: a p2p networking infrastructure based on rdf. pages 604–615. ACM, May 2002 2002.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[22] J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies, 2000. In Research Report. Institut d'Investigacio i Intelligencia Artificial.

[23] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligenc. Review*, 24(1):33–60, 2005.