Combining User Reputation and Provenance Analysis for Trust Assessment

Davide Ceolin, VU University Amsterdam
Paul Groth, Elsevier B.V.
Valentina Maccatrozzo, VU University Amsterdam
Wan Fokkink, VU University Amsterdam
Willem Robert van Hage, Netherlands eScience Center
Archana Nottamkandath, VU University Amsterdam

Trust is a broad concept which, in many systems, is often reduced to user reputation alone. However, user reputation is just one way to determine trust. The estimation of trust can be tackled from other perspectives as well, including by looking at provenance.

Here, we present a complete pipeline for estimating the trustworthiness of artifacts given their provenance and a set of sample evaluations. The pipeline is composed of a series of algorithms for: (1) extracting relevant provenance features, (2) generating stereotypes of user behavior from provenance features, (3) estimating the reputation of both stereotypes and users, (4) using a combination of user and stereotype reputations to estimate the trustworthiness of artifacts and, (5) selecting sets of artifacts to trust. These algorithms rely on the W3C PROV recommendations for provenance and on evidential reasoning by means of subjective logic.

We evaluate the pipeline over two tagging datasets: tags and evaluations from the Netherlands Institute for Sound and Vision's *Waisda?* video tagging platform; and crowdsourced annotations from the *Steve.Museum* project. The approach achieves up to 85% precision when predicting tag trustworthiness. Perhaps more importantly, the pipeline provides satisfactory results using relatively little evidence through the use of provenance.

ACM Reference Format:
Davide Ceolin and Paul Groth and Valentina Maccatrozzo and Wan Fokkink and Willem Robert van Hage and Archana Nottamkandath, 2015. Combining User Reputation and Provenance Analysis for Trust Assessment *ACM J. Data Inform. Quality* V, N, Article A (January YYYY), 29 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

From deciding the next book to read to selecting the best movie, we often use the reputation of an author as proxy for trusting that the item in question will be interesting, relevant or good. Reputation is an important mechanism in our set of strategies to determine trust. However, we may base our assessment on a variety of other factors as well, including prior performance (if a user has been trustworthy recently, then we may believe that he is), content on its own (we may decide that an artifact is trustworthy based on some characteristics of its content), a guarantee (some artifacts are

Author's addresses: D. Ceolin, V. Maccatrozzo, W. Fokkink, A. Nottamkandath, Computer Science Department, VU University Amsterdam, de Boelelaan, 1081a, 1081HV Amsterdam, The Netherlands; email: {d.ceolin, v.maccatrozzo, w.j.fokkink, a.nottamkandath}@vu.nl; P. Groth, Elsevier B.V., Radarweg 29, 1043 NX, Amsterdam, The Netherlands; email: p.groth@elsevier.com; W. R. van Hage, Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands; email: w.vanhage@esciencecenter.nl.

DOI: http://dx.doi.org/10.1145/0000000.0000000

certified by third party trusted authorities), or knowledge of how something was produced (we may decide to trust an artifact if we know that it is the result of a renowned and trusted process). Nevertheless, many systems, especially on the Web, choose to reduce trust evaluation to user reputation analysis alone [Jøsang et al. 2007]. As others have noted [De Alfaro et al. 2011], this is of concern when the identity of the user is unknown or may not be of particular importance.

In general, this work tackles the problem of assessing the trust in Web artifacts. Specifically, we focus on cultural heritage and media crowdsourced annotations, and in particular on untyped and non-hierarchical annotations or tags, that is, on annotations that aim at generically describing the content of an object without further specifying their type (for instance, we do not consider annotations that explicitly aim at representing the author or the genre of an object). These annotations are qualified as crowdsourced because these are annotations collected by organisations that use crowds of Web users to annotate their large collections. In this manner, these organizations cope with the high workload needed to annotate their own collections. However, since their quality standards are stringent, and the identity and trustworthiness of Web users is often unknown, a trust issue arises. On the one hand, accepting all the crowdsourced annotations without first checking their quality may affect the organizations' processes based on the annotations themselves, in case these are of low quality. On the other hand, the workload needed to manually check all the annotations would be unfeasible for the limited workforce at disposal of these organizations. So, the main challenge addressed here divides into two sub-challenges that are possibly conflicting with each other. On the one hand, we need to provide annotation evaluations that are close to how the organization would have evaluated them. On the other hand, these assessments have to be produced semi-automatically, to reduce the workload for the organization. Therefore, we take a multi-faceted approach. We look at trust assessment using a combination of reputation and provenance. We adopt the definition of provenance from the W3C Provenance Working Group: "Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness." [Moreau et al. 2013]. In particular, we look at how to use the provenance of multiple artifacts, i.e., of objects made or shaped by some agent or intelligence, to compute a proxy for a set of users. We term such a proxy a *stereotype*. For example, if there are several artifacts that are produced by users between 5:00 and 6:00 am, we could generate a stereotype of an "early morning user". Just like we can calculate the reputation of a user, we can also calculate the reputation of a stereotype (e.g., early morning users produce high quality content). The reputation of a stereotype can then be used as a stand-in for user reputation when none exists or to augment existing user reputations to provide more input when calculating trust. Importantly, our approach for creating stereotypes leverages the W3C PROV recommendations [Groth and Moreau (eds.) 2013] for provenance interchange. This allows the approach to be applied in a generic fashion to systems that use this standard. So, while the use of the user reputation aims at providing a high quality and user-tailored trust prediction, the use of provenance stereotypes aims at reusing the evaluation available, thus improving the estimations made while reducing the size of the set of evaluations needed by our system (and, consequently, reducing the organization's workload in terms of such evaluations).

Specifically, this article makes three contributions:

(1) A feature selection algorithm for extracting relevant behavioral features from provenance graphs and associating them with users.
(2) An approach for constructing stereotypes using the selected features.

(3) A trust calculation algorithm that combines user and stereotype reputations to determine trust for an artifact.

We evaluate our approach using data from a video tagging game platform (i.e., an online platform where users challenge each other in annotating videos) and from a cultural heritage annotation project. We show that by combining provenance-based stereotypes with user reputation we are able to accurately estimate trust (up to 85% precision). More importantly, we show that even with relatively small amounts of training data, by leveraging provenance, our approach can still obtain good performance.

This article extends the work presented in a previous paper of ours [Ceolin et al. 2012a]. It adds the aforementioned feature selection algorithm and introduces the notion of provenance-based stereotype. Moreover, we improve our trust estimation algorithm by means of merging information stemming from provenance and reputation. Finally, we also have refined and expanded our evaluation.

The rest of this article is organized as follows. We discuss related work in trust with a particular focus on reputation estimation and provenance analysis in Section 2. This is followed by a description of the approach chosen in Section 3. The approach is evaluated in Section 4, and Section 5 proposes an analysis of the robustness of the framework. Finally, we discuss future work and conclude in Section 6.

2. RELATED WORK

Trust is a widely explored topic within a variety of computer science areas. Here, we focus on those works directly touching upon the intersection of trust, reputation and the Web. We refer the reader to the work of Sabater and Sierra [Sabater and Sierra 2005], Artz and Gil [Artz and Gil 2007], and Golbeck [Golbeck 2006] for comprehensive reviews about trust in respectively artificial intelligence, Semantic Web and Web. Trust has also been widely addressed in the agent systems community. Pinyol and Sabater-Mir provide an up-to-date review of the literature in this area [Pinyol and Sabater-Mir 2013]. Prasad et al. [Krishnaprasad et al. 2014] provide instead a comprehensive overview of Bayesian trust management systems that we use in this paper as a basis for benchmarking the system that we propose.

Part of our work focuses on reputation estimation and is inspired by the works collected by Masum and Tovey [Masum and Tovey 2012]. Similar to our work, those of Pantola et al. [Pantola et al. 2010] and of Javanmardi et al. [Javanmardi et al. 2010] present reputation systems that measure the overall reputation of the authors based on the quality of their contribution and on user behavior in a wiki environment. Our work differs in that it aims to provide a generic framework and is evaluated within a tagging environment, i.e., by evaluating the trustworthiness of crowdsourced tags.

Another part of our work focuses on the usage of provenance information for estimating trust. In the work of Bizer and Cyganiak [Bizer and Cyganiak 2009], Hartig and Zhao [Hartig and Zhao 2009] and Zaihrayeu et al. [Zaihrayeu et al. 2005], provenance and background information expressed as annotated or named graphs [Carroll et al. 2005] are used to calculate trust values. Our work is different in that we use this provenance information to build a model of behavior. This is a similar approach to that presented in the work of Rajbhandari et al. [Rajbhandari et al. 2006; Rajbhandari et al. 2008], where they quantify the trustworthiness of scientific workflows by means of probabilistic and fuzzy models. Our work differs in that we focus on user behavior and not on structured workflows.

Provenance has also been used for data verification in crowdsourced environments by Ebden et al. [Ebden et al. 2012]. They introduced provenance tracking into their online CollabMap application (used to crowdsource evacuation maps), and in this way

collected approximately 5,000 provenance graphs. Based on this large corpus they learned useful features for determining an artifact trustworthiness based purely on the provenance graph topology. Here, the graphs available are much more limited, so we cannot rely on the graph topology. Instead, we use provenance graphs to enable the extraction of useful features about classes of users.

Provenance mechanisms have also been used to understand and study workflows in collaborative environments as discussed in Altintas et al. [Altintas et al. 2010]. Our work is complimentary as it could enable trust assessment in this sort of environments.

This work extends also another prior work of ours [Ceolin et al. 2010], where we determined the trustworthiness of event descriptions by applying subjective logic [Jøsang 2001] to provenance traces for those descriptions. In the current paper, we apply a similar approach to a new domain. We still represent trust values by means of subjective opinions, but trust assessments are computed by merging user reputations with provenance-based estimates. Additionally, we also investigated approaches for determining trust of annotations based on user reputation [Ceolin et al. 2014]. That work relied on semantic similarity measures to weigh evidence to make trust assessments and rank annotations. Here, the ranking is obtained through the combination of user reputations and provenance stereotypes, i.e., provenance-based representations of user behaviors. Importantly, this work looks at how to combine provenance and reputation to make trust assessments.

3. APPROACH

In this section, we describe the approach taken to address the issue of estimating the trustworthiness of cultural heritage annotations. First, we provide background on the formal framework and definitions we use. Then, we give an overview of how the various algorithmic components introduced in this paper fit together within an overall pipeline. In this context, we introduce the terminology used in the remainder of the paper. Lastly, we explain in detail each component of the framework.

3.1. Formal Framework: Subjective Logic

Subjective logic [Jøsang 2001] is a type of probabilistic logic characterized by the representation of the uncertainty of the estimates and the recording of argument sources. Subjective logic is compatible with both binary and probabilistic logics. Thus, it allows for the representation of the truth value of propositions both in terms of boolean values and probabilities. In addition, the logic allows one to account for uncertainty when a truth value is estimated on the basis of a limited set of evidence. The basic element of subjective logic is called an opinion. A subjective opinion (represented by means of the symbol ω) represents the belief (b), disbelief (d), uncertainty (u) and prior (or base rate, a) owned by source $source$ with respect to the proposition x. These values are computed based on an evidence set observed by $source$. Belief and disbelief are computed based on the ratio between positive and negative pieces of evidence in such set. Uncertainty is determined based on the size of the set: the larger the set of evidence, the lower the uncertainty, and vice-versa. See Equation (1) for a formal representation of subjective opinions. For brevity, the source can be omitted.

$$\omega_x^{source} = (b, d, u, a). \tag{1}$$

The probability of a statement x can be claimed by $source$ with no uncertainty. In that case, we talk about "dogmatic opinions" ($P(x) = b$). However, in many cases, the probability of x being true is determined from evidence. For instance, the probability that the proposition x (defined in Equation (2)) is true can be determined from a set of

items of evidence about Davide's performance as an annotator.

$$x = \text{Davide is a trustworthy annotator.} \tag{2}$$

Such a set can be formed of p positive pieces of evidence and n negative ones. We can then infer the values of ω_x as described in Equation (3).

$$b = \frac{p}{p+n+2}, \quad d = \frac{n}{p+n+2}, \quad u = \frac{2}{p+n+2}, \quad a = \frac{1}{2}. \tag{3}$$

Some important facts can be derived from this inference. First, $b + d + u = 1$. b and d represent the probability mass assigned in the belief of x being true and $false$ respectively. u represents some probability mass that is not assigned to one of the two values (true, false) on the basis of the evidence observed, since this evidence is limited and thus, possibly fallacious. Such probability mass will be assigned to either true or false when the expected value E will be computed, on the basis of the prior value a. The default value for the smoothing factor in the denominator of u is equal to the cardinality of the set of possible outcomes of the distribution (true, false), that is, 2. $a \in [0, 1]$ and a is set equal to 0.5 as to indicate that our prior is "neutral". A priori, we have no bias neither against nor pro x.

We said before that not all the probability mass of the truth value of x is assigned on the basis of the evidence observed. Thus, how can we compute the expected truth of x? Equation (4) provides a formula for computing the expected truth value.

$$E = b + a \times u. \tag{4}$$

The expected truth of x is in fact the expected value of a probability distribution, namely of a Beta probability distribution. See Fig. 3.1 for an example of Beta probability distribution. Subjective opinions are built in such a manner that they are equivalent to Beta probability distribution. The reason for this choice is that propositions take a Boolean value (we restrict ourselves to the Boolean case; subjective logic can also deal with propositions taking one over more than two values). So, if we have the probability for a proposition to be true, we could use a Binomial probability distribution to model the truth of x. However, we do not have such a probability distribution, but we estimate it, based on the evidence at our disposal. This evidence is used to build a Beta distribution that describes the probability for all the values in the $[0, 1]$ interval to be the truth value of x (i.e., to represent the right parameter p of the Binomial distribution). The choice of the Beta distribution is due to the fact that the Beta is the so-called "conjugate prior" of the Binomial distribution. This means that the two distributions belong to the same family and, by using the Beta as a prior of the Binomial, we simplify the computation of the posterior distribution, that is, of the distribution that we obtain by considering the evidence observed. In fact, in our case, given a $Beta(\alpha, \beta)$, after having observed p positive items of evidence and n negative ones, the posterior distribution will be $Beta(\alpha + p, \beta + n)$. By making use of Equation (3), we can rewrite Equation (4) as Equation (5).

$$E = \frac{p+1}{p+n+2}. \tag{5}$$

that is the expected value of $Beta(\alpha + p, \beta + n)$, where $\alpha = 1$ and $\beta = 1$, i.e., when the Beta's prior is uninformative. When no additional information is available, the Beta's prior is set to $\alpha = 1$ and $\beta = 1$ so as not to privilege positive or negative outcomes. In subjective logic, this is represented by $a = \frac{1}{2}$ in Equation (4). One last important remark about the Beta distribution is the fact that its shape depends on the ratio of the sample at our disposal (positive vs. negative evidence), but also on its size. The larger the sample, the lower the variance of the distribution and, in subjective logic, this

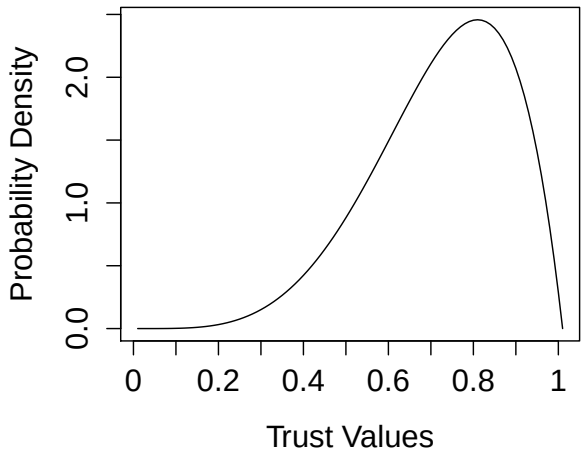

Fig. 1. Beta distribution based on 4 positive pieces of evidence, 1 negative and uninformative prior (Beta(4+1,1+1) = Beta(5,1)). The expected value of this distribution is 0.8, and the distribution is equivalent to an opinion $\omega\left(\frac{4}{7}, \frac{1}{7}, \frac{2}{7}\right)$.

fact is captured by a corresponding lower uncertainty (u in Equation (3)). Because of this mechanism, we do not employ cross-validation in our pipeline, since uncertainty is modelled exactly to reduce the probability of overfitting. Hence, thanks to the adoption of subjective logic, our model can incrementally incorporate new knowledge every time that new evidence is acquired, instead of having to rerun cross validation over the new, extended training set.

Equation (4) represents one of the crucial reasons why we choose subjective logic for managing our trust estimates. In fact, the expected probability is determined by averaging our prior knowledge (a) and our belief, that is determined on the basis of the evidence observed. This average is weighed on the basis of the uncertainty (u), and the uncertainty is computed in such a manner that it decreases as long as the evidence set grows. In other words, when the evidence set is small, the weigh of the belief is higher, and vice versa, when the evidence set is large, the weight of the base rate is small (and the weight of the belief is high). Later, we discuss how we estimate the base rate from an analysis of the provenance of the artifacts we analyze. Likewise, we discuss how belief is determined by analyzing the reputation of the creators of artifacts.

Subjective logic also offers a wide variety of operators for combining opinions. These include, for instance, operators for weighing opinions using semantic similarity measures that we developed in the past [Ceolin et al. 2012b]. We do not make use of such operators here. However, this capability may be useful for further extensions to this work.

3.2. System Overview

Figure 2 presents a schematic diagram of how the individual components fit together and of the various pieces of data they operate on. Before describing this diagram, we begin by introducing the terminology we use. There are four core input data types used within the system.

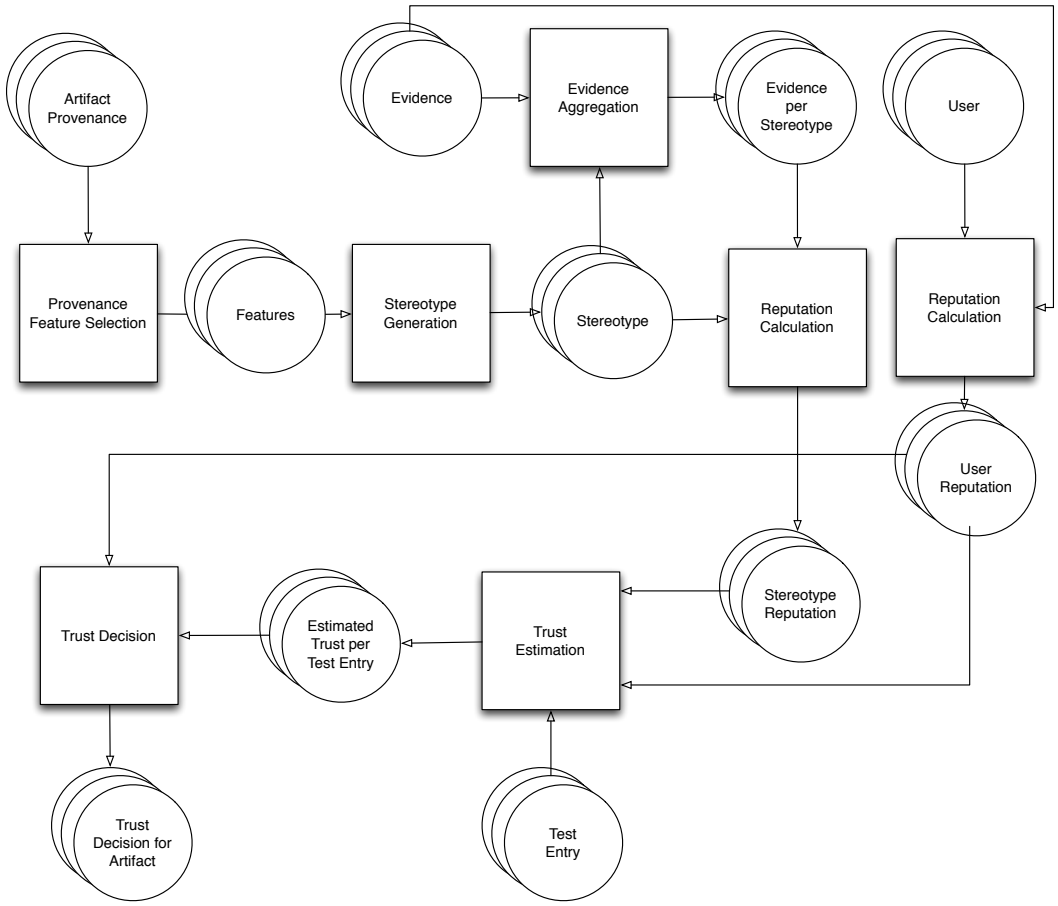

Fig. 2. Overview of our trust evaluation pipeline. Circles denote data, stacked circles denote data sets, and squares denote processing components.

(1) An *artifact* is the content that is under consideration (e.g., a tag).
(2) *Artifact provenance* is how some artifact is produced (e.g., the provenance graph associated with an artifact).
(3) A *user* is a unique identity of a given user.
(4) An *evidence item* is a mapping from a user/stereotype, artifact tuple to an evaluation. (e.g., whether a tag generated by a user is rated as good or not).

These elements are assembled into two different structures:

(1) *Evidence set* (or *evidence* for the sake of brevity) is a set of evidence items.
(2) An *Entry* is a tuple that associates an artifact, provenance, and the user who creates the artifact together.[1]

Thus, our system has a set of known entries and the corresponding evidence set (i.e., the training set). These are divided into the input data shown in Figure 2: a set of provenance information for each known artifact, the evidence set, and a set of users.

[1]We note that the user is part of the provenance of the artifact. However, for ease of explanation we explicitly identify it.

The aim is to determine for a set of *test entries* whether the artifacts identified within those entries should be trusted or not.

The pipeline begins by first extracting *features* from the provenance of the training data (e.g., the video a tag is added to). These features are then used to compute multiple *stereotypes*. Stereotypes are a representation of groups of users based on their behavior. In this case, the behavior is extracted from the provenance of the artifacts. Once stereotypes have been generated, we aggregate the input evidence set per stereotype. We now have the input data necessary to calculate the reputation not only for each user but also for each stereotype.

Our reputation algorithm learns a model for the reputation of each user or stereotype based on an evidence set about past performance of that user/stereotype. These reputation models are used together within the trust estimation component. Importantly, the stereotype reputation acts as prior to ensure that we can deal with artifacts produced by users where there is little or no prior experience. For instance, when a new user tags a video. This component then generates the probability whether a given set of artifacts should be trusted, based on the supplied test entries and the background information about reputation. An estimate is of the form of a likelihood that the given artifact should be trusted. For example, there may be a 0.8 probability that this tag is trustworthy. To translate this probability into an actual decision about whether to go ahead and trust (i.e., accept) a given artifact, we introduce a trust decision component. This component uses the reputation of the user in order to rank the given entries and then select the number of artifacts to be trusted.

Running Example. Suppose that a user, *Davide*, provided fifteen annotations of paintings to a crowdsourcing platform that collects annotations on behalf of a museum. Our algorithm estimates which annotations to trust and which not, in a semi-automated fashion, as follows:

(1) it computes a reputation for *Davide* by asking the museum to evaluate a fixed number of his annotations, let us say five;
(2) it takes the provenance of all the evaluated annotations in the system;
(3) it groups the annotations based on the stereotype each annotation belongs to (e.g., "weekday between 00:00 and 08:00"), and it computes a reputation per stereotype;
(4) for each new annotation provided by *Davide*, it computes a trust value by merging *Davide's* reputation with the reputation of the provenance stereotype each of *Davide*'s annotations belongs to;
(5) it ranks *Davide*'s annotations based on their trust values;
(6) it accepts the first $E\%$ of *Davide*'s ordered annotations, where $E\%$ is *Davide*'s reputation, as computed in step 1.

We now investigate each of these components in turn, discussing the underlying algorithms and approach. We also develop further the example above in order to explain each component in depth.

3.3. Provenance Feature Selection

One of our underlying assumptions is that if one trusts artifacts produced in a particular fashion, then one is likely to trust new artifacts generated in a similar fashion. Here the similarity refers to both the processes employed as well as the inputs taken. Thus, a requirement of our pipeline is to obtain information about how artifacts have been produced. Provenance provides a structured description of how a given artifact was produced. From this structured information, we should be able to extract important features for use by subsequent learning algorithms in the creation of models for determining trust.

The W3C PROV recommendation [Moreau et al. 2013] provides a standard model for provenance information. By leveraging the model, we are able to develop an algorithm that is domain independent and generic. Thus, we can obtain features for trust calculation only based on provenance specific semantics with minimal knowledge of the domain. As the use of PROV increases, we can easily adapt our pipeline to new domains. We refer the reader to the PROV Primer [Gil et al. 2013] for PROV specific terminology .

The algorithm for selecting the relevant features from provenance is defined by means of the recursive function described in Function ExtractProvenanceFeatures. Given a starting PROV entity, that is, the artifact for which we want to obtain provenance, the function:

(1) retrieves all the PROV activities that generated that entity;
(2) from each of these activities, extracts their identifiers and temporal information;
(3) from each of these activities, retrieves the PROV agents that are associated with them and extracts the agents's identifiers;
(4) from each of the activities, retrieves all the entities used as input, and recursively applies this algorithm to them.

Thus, we explore the entire provenance graph and extract the relevant features along the way. We assume that the graph is acyclic and finite, therefore we do not need to make explicit a fixed point that assures the termination of the algorithm in a finite amount of time. This algorithm is applied to all artifacts in a given training set. Its importance relies on its ability to provide us with all the provenance features that are to some extent related to a given artifact (even if not always directly, but through related activities or entities).

Function ExtractProvenanceFeatures(artifact)

Input: An artifact: entity(e,-)
Output: A finite set of provenance features $Result = \{feature_1\}$

```
1  res ← ∅
2  forall the a : wasGeneratedBy(e,a) do
3      activity(a, [starttime = t1, endtime = t2 ...])
4      res ← res ∪ {t1, t2}
5      forall the ag : wasControlledBy(a,ag) do
6          res ← res ∪ ag
7      forall the e1 : used(a,e1) do
8          res ← res ∪ e1
9          res ← Extract_provenance_features(e1)
10     return res
```

Running Example. We now start to analyze in depth the example outlined above. Suppose that the crowdsourcing platform through which annotations are collected records, for each annotation, a timestamp, the typing duration, the browser used, the vocabulary from which the annotation is selected, and the author of the vocabulary item. This information is representable in PROV, and the algorithm here described allows us to traverse this graph and to select all this information to be used as features to define provenance stereotypes.

However, for the sake of simplicity, suppose that the crowdsourcing platform records only the timestamp of the annotations, encoded as the timestamp of the *activity* from

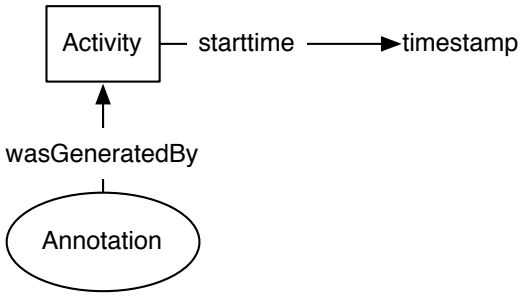

Fig. 3. Provenance graph of the simple annotation described in the example.

which each annotation *wasGeneratedBy* (again, for simplicity, we let the starting and ending time be coincident). See Fig. 3. In this, extremely simple, case we can use the algorithm above to extract this information and use it as a feature of the annotation.

3.4. Stereotype Generation

Once the provenance features have been extracted, we proceed to the generation of the stereotypes. Stereotypes are a representation of groups of users based on their behavior. These stereotypes are built based on the features extracted from the provenance of the artifacts created by multiple users. We build stereotypes using all the provenance at our disposal, and considering the distribution of the evidence we use. The reason why we do not discriminate among the different kinds of provenance information that we have at our disposal is that we do not know which are more correlated with trustworthiness. So, for instance, we hypothesize that the time when an artifact was created might affect the ability of its creator and thus the artifact's trustworthiness (e.g., late at night people might be sleepy and thus, less attentive). Following this reasoning, we include all the information at our disposal, but we group them in order to be able to gather enough evidence for each stereotype thus identified. In this manner, we can compute a reputation per stereotype, and use it in our framework.

In order to create these groups, we adopt the following strategy. First, we discretize the feature values as follows:

— Categorical data are kept as they are.
— Temporal features are discretized as follows: we extract the *day of the week* and the *hour of the day* from the timestamps at our disposal. Then, we group each of these two features in n in order to have the artifacts evenly distributed in each of the groups thus obtained (each group is thus the nth quantile). For instance, in the *Waisda?* dataset we split the *hours of the day* in the following groups: {0-13, 13-17, 17-21, 21-24} because this guarantees us that $\frac{1}{4}$ of the artifacts falls into each group. Each group is identified by a progressive number (from zero to three).
— Continuous features (e.g., typing duration) are, again, discretized using the quantiles method adopted for the temporal features.

These values are then concatenated to obtain a stereotype. Function CreateStereotype shows the function to create a stereotype. The *discretize* function employed in Function CreateStereotype discretizes the features according to the quantiles that were previously determined. Temporal and continuous features are grouped in order to obtain even distributions because in this manner we guarantee the availability of evidence for the corresponding stereotypes. Alternative approaches are possible as well and will be investigated in the future.

Function CreateStereotype(features)

Input: A set of provenance features
Output: A stereotype identifier

```
forall the feature do
    if feature is categorical then
        id ← concatenate(id,feature)
    else
        quantile_id ← discretize(feature)
        id ← concatenate(id,quantile_id)
return id
```

Function CollectEvidenceStereotype uses Function CreateStereotype to group the evidence related to that stereotype. The function returns a map having the stereotype as a key and a set of artifacts as a value.

Function CollectEvidenceStereotype(stereotype)

Input: A set {entities, provenance} pairs and a set of stereotypes
Output: A set of stereotypes

```
forall the entity do
    stereotype ← create_stereotype(extract_features(provenance))
    evidence_stereotype [stereotype ] ← entity
    return evidence_stereotype
```

Running Example. Following the example started above, suppose that the annotations are not distributed evenly during the week. Following the discretization indications defined above, from the timestamps we extract the day of the week and the hour of the day. Each of these is a separate feature of our annotations. For the sake of simplicity, we split each feature in two quantiles, so that we obtain four stereotypes in total, by combining these two pairs of quantiles. As we said, the annotations are not distributed evenly during the week, so the stereotypes we obtain are:

— day-weekday;
— night-weekday;
— day-weekend;
— night-weekend.

For each of these, Function CollectEvidenceStereotype collects the available pieces of evidence.

3.5. Computing Reputation

Reputation is an estimation we make about the reliability of a user or of its stereotype. This estimation is based on a limited set of observations about the user (or the stereotype) performance. We measure reliability in terms of the percentage of positive artifacts (positively evaluated artifacts that correspond, for instance, to correct annotations) over the total amount of artifacts provided by a specific user or stereotype (see

Equation (6)).

$$P(user) = P(\text{artifacts created by user are trustworthy}) = \frac{\#\text{positively evaluated artifacts created by user}}{\#\text{artifacts created by user}}. \quad (6)$$

We assume that the reputation we compute represents a useful approximation of reliability, and in the evaluation presented in Section 4 we test whether this is actually the case.

$$\omega_{user} \approx P(user). \quad (7)$$

We use subjective opinions (ω_{user}) to represent reputations because these allow us to take into account both the ratio of positive and negative observations in our samples and the size of the samples. We propose two procedures for computing the reputations, one for computing user reputations and the other for computing stereotype reputations.

3.5.1. User Reputation. The user reputation is computed by means of the expected value of a subjective opinion (ω_{user}), which in turn is computed using an evidence set (of size $size$) about this user. Function UserReputation illustrates the function for creating a user reputation given the user identifier and the size of the evidence set used. Recall that a subjective opinion is composed by three values: belief, disbelief and uncertainty. To compute the expected value of the subjective opinion, we need only two of these, namely belief and uncertainty, which are the two values computed by Function UserReputation. To compute the expected probability of the subjective opinion (i.e., the user reputation), we need to combine them as in the following formula:

$$E(\omega_{user}) = b_{user} + a_{user} \times u_{user}. \quad (8)$$

The formula above makes use also of the user base rate a_{user}, that is, the trust we have in the user before observing any evidence item about him or her. When we evaluate the trustworthiness of an artifact, we combine the belief in the artifact's creator (i.e., the user) with a base rate determined by the reputation of the stereotype that the artifact belongs to. The artifact belongs to a stereotype if its provenance matches that of the behavior represented in the stereotype. In other words, a stereotype models how this artifact has been generated. So, when we estimate the trustworthiness of an artifact, we base part of our estimate on the evidence about the artifact's creator (belief and uncertainty), and part of it on the stereotype reputation, which we use as prior knowledge. In case no evidence about the user is available, we fully rely on the stereotype reputation. If evidence about the user is available, then the weight of this evidence on the trustworthiness estimate is proportional to the size of the evidence set. We implement this in Section 3.6.1.

On the other hand, to compute the overall reputation of a given user we assign the value 0.5 to a_{user} (base rate) because, in general, we have no prior bias against or in favor of a given user and 0.5 represents a neutral value for the base rate. We do this in the algorithm presented in Section 3.6.2. For this reason, our function returns a pair $\{belief, uncertainty\}$ instead of a reputation E for a given user. We do not explicitly output the disbelief in $user$, both because it is not necessary for the computation of E and because it is easily computable from Equation (4).

3.5.2. Stereotype Reputation. The reputation of a stereotype is computed in a similar fashion to that of a user. In fact, the function reported in Function StereotypeReputation behaves like Function UserReputation, except that Function StereotypeReputation already assumes the base rate to be equal to 0.5. This is due to the fact that we

Function UserReputation(user,size)

Input: A user and a number $size$
Output: A user reputation, expressed as a subjective opinion based on the evidence observed

```
1 evidence ← evidence_selection (user,size)
  /* evidence: a set of p positive and n negative items of evidence (p+n =
  size)                                                                  */
2 belief ← p / (size + 2)
3 uncertainty ← 2 / (size + 2)
4 return {belief, uncertainty }
```

do not have any prior knowledge about the expected trustworthiness of the artifacts belonging to a given stereotype. Rather, we use the expected probability of the stereotype (that is, its reputation) as a base rate for the computation of the trust values of the artifacts we analyze.

Function StereotypeReputation(stereotype,size)

Input: A stereotype and a number $size$
Output: A stereotype reputation, expressed as a subjective opinion based on the evidence observed

```
1 evidence ← evidence_selection (stereotype,size)
  /* evidence: a set of p positive and n negative items of evidence (p+n =
  size)                                                                  */
2 belief ← p / (size + 2)
3 uncertainty ← 2 / (size + 2)
4 reputation ← belief + uncertainty/2
5 return reputation
```

Running Example. Let us continue with the example introduced above. Recall from Section 3 that user *Davide* provided fifteen annotations in total, and five of these are used to compute his reputation. Suppose that one of these is rejected and four annotations are accepted. Then, using Equation (3) and (4), his reputation is:

$$E^{Davide}_{museum} = \frac{4}{7} + \frac{2}{7} \times \frac{1}{2} = 0.71. \tag{9}$$

Likewise, we compute the reputations also for the four stereotypes identified above. For example, if the stereotype *weekend-night* has five positive and five negative pieces of evidence, then its reputation is:

$$E^{weekend-night}_{museum} = \frac{5}{12} + \frac{2}{12} \times \frac{1}{2} = 0.5. \tag{10}$$

3.6. Calculating Trust

Following the trust theory of O'Hara [O'Hara 2012], our trust calculation algorithm consists of two steps: trustworthiness estimation and a trust decision.

3.6.1. Trustworthiness Estimation. The trust estimation procedure computes the trust level for an artifact based on a combination of the reputation of the user that created the artifact and of the provenance stereotype that the artifact belongs to:

$$E(\omega_{artifact}) = b_{user} + u_{user} \times E(\omega_{stereotype}). \quad (11)$$

The rationale behind this formula is the following. The user reputation is the probability of the artifact produced by a given user to be trustworthy. To estimate the trust value of an artifact, i.e., its probability to be trustworthy, we can employ evidence about the prior performance of its creator. We represent this value by means of b_{user}. However, the same user may be more or less reliable depending on when, how or where he or she created the artifact. In fact, we can build a reputation for the stereotype the artifact belongs to (represented as $\omega_{stereotype}$), which represents the probability for an artifact belonging to this stereotype to be trustworthy. The actual estimation of the trustworthiness of the artifact is based on a combination of the reputation of its author and of the reputation of its provenance stereotype. The two reputations are combined by averaging them, weighed on the uncertainty parameter (u_{user}): the more evidence we own about the user, the more the user reputation will weigh on the trust value, because if we own a large set of evidence about the user performance, we can reasonably rely on the user reputation we derive from it. If such an evidence set is limited, then the uncertainty will be higher, so more weight will be given to the stereotype reputation.

Function TrustEstimation(artifact,user,provenance)

Input: A user and a number $size$
Output: A user reputation, expressed as a subjective opinion based on the evidence observed

```
features ← extract_provenance_features (artifact,provenance)
stereotype ← compute_provenance_stereotype (features)
stereotype_reputation ← reputation (stereotype,size)
user_belief, user_uncertainty ← reputation (user,size)
return user_belief + user_uncertainty × stereotype_reputation
```

3.6.2. Trust Decision. Our trust selection algorithm relies on the assumption that the user reputation represents a reliable approximation of the user reliability. Since the user reputation is represented by means of an expected probability, we can use that probability by selecting as "accepted" only the best p% artifacts created by a given user, where p% is the user reputation, expressed in a percentage. We can then rank the artifacts produced by the same user thanks to the fact that not all the artifacts belong to the same stereotype. This is because in Equation (11) we assign a slightly different trust value to artifacts created by the same user but belonging to different user stereotypes.

Running Example. Continuing with our example, consider an annotation, *a1*, that *Davide* created during the weekend, at night. Using Equation (11), its trust value is:

$$E_{a1} = 0.57 + 0.5 \times 0.5 = 0.82. \quad (12)$$

Now, recall that *Davide* contributed fifteen annotations, but five of these are used as training set, so ten have to be evaluated. Since his reputation is 0.57, 57% of the 10 annotations to be evaluated (6) are accepted, and the remaining 43% (4) are rejected.

Function TrustSelection(artifacts, reputation)

Input: A set of artifacts and the reputation of user that created them
Output: A set of selected artifacts

```
evaluated_artifacts ← ∅
forall the artifact do
    evaluated_artifacts ← evaluated_artifacts ∪ trust_estimation (artifact)
ranked_artifacts ← rank (evaluated_artifacts)
selected_artifacts ← select_artifacts(ranked_artifacts,user_reputation)
return selected_artifacts
```

To decide which annotations to accept, our algorithm computes a trust value per annotation, based on *Davide*'s reputation and on the annotation stereotype. Then, the annotations are ranked on the basis of their trust value and the first six annotations are accepted. Thus, *a1* is accepted only if at most five other annotations have a trust value higher than 0.82. Otherwise *a1* is rejected.

4. EVALUATION

We evaluate our trust assessment pipeline using two tagging datasets, the *Waisda?* and the *Steve.Museum* datasets. The algorithms discussed above and corresponding evaluation analysis have been implemented in Python and are available online.[2] Only the *Steve.Museum* dataset is available publicly.[3]

Before presenting the datasets and the results obtained on them, we define a baseline to assist our understanding of such results.

4.1. Baseline Definition

Following the classification described by Krishnaprasad et al. [Krishnaprasad et al. 2014], the system that we propose is partly a direct trust management system and partly an indirect one:

— The author reputation is a form of direct trust: we observe the user performance for a given number of executions (i.e., we evaluate a fixed amount of artifacts he produces) and we estimate his reputation based on such an evaluation. So, this is a form of direct trust.
— The stereotype reputation is used to tailor the user reputation with respect to a given artifact. Such an adjustment is made by taking into account how other users performed when they operated in similar conditions (i.e., artifacts produced in the morning, at night, etc.). Therefore, this is a form of indirect trust.

Also, unlike the trust management systems described by Krishnaprasad et al., we do not make use of third party recommendations, since we do not have them at our disposal and their management is outside the current scope of our system. We could use those systems to make trust estimates, but: (1) they all reduce to the same system, where a user's reputation is computed by estimating a Beta distribution based on a set of positive and negative observations; (2) those systems allow us only to compute user reputations, while the system that we propose here provides us also with a final decision about which artifacts to use and which not, based on their specific trust

[2]We divided our code based on dataset into two iPython notebooks. The *Waisda?* analysis can be viewed at http://nbviewer.ipython.org/gist/davideceolin/9003776 and downloaded at https://gist.github.com/davideceolin/9003776. The *Steve.Museum* analysis is viewable at http://nbviewer.ipython.org/gist/davideceolin/9010127 and downloadable at https://gist.github.com/davideceolin/9010127.
[3]The *Steve.Museum* dataset is downloadable at http://verne.steve.museum/steve-data-release.zip.

value. The three models described by Krishnaprasad et al. (i.e., Denko-Sun's approach for mobile and ad-hoc networks [Denko and Sun 2008], Ganeriwal approach for sensor networks [Ganeriwal et al. 2008] and Sun et al.'s approach for mobile and ad-hoc networks [Sun et al. 2006]) yield the same value for the user reputation, that is the following value, based on Equations 3 and 4:

$$reputation = \frac{p}{p+n+2} + \frac{2}{p+n+2} \times \frac{1}{2}.$$

This formula is the same as the one we use in our system to compute the user reputation. However, those systems do not allow us to determine how to use such a value to decide which artifacts to trust and which not. Therefore, we propose two naïve methods to take such a decision and we use them as baseline to compare against the one we propose.

4.1.1. Baseline Decision Method 1: Fixed Threshold. This first decision method works as follows: we set a threshold value and we accept only the artifacts whose trust value are higher or equal to the threshold. Since this method is not able to discriminate among the artifacts created by a given author, all the artifacts are accepted if the author's reputation is higher or equal to the threshold, and rejected otherwise. Function FixedThresholdDecisionStrategy shows the algorithm.

Function FixedThresholdDecisionStrategy

Input: An artifact: entity(e,-); a threshold *threshold*
Output: A decision d $\in$ {`accept`, `reject`}

```
1 a1 ← {a : wasGeneratedBy(e,a)}
2 ag1 ← {ag : wasControlledBy(a,ag)}
3 reputation ← get_reputation (ag1)
4 if reputation≥threshold then
5     return accept
6 else
7     return reject
```

Running Example. Suppose that we set the threshold to 0.9. Then since *Davide*'s reputation is 0.71, all the fifteen artifacts he created are rejected by this decision method. Viceversa, if we set the threshold to 0.7, the decision method lets us accept all fifteen contributions.

4.1.2. Baseline Decision Method 2: Random Selection. The previous baseline method has two main limitations:

— It requires the setting of a threshold, that has to be set a priori and is valid for all the annotations of all the authors, without the possibility to tune it.
— The use of a threshold neglects the probabilistic nature of trust values. Assuming that the reputation that we estimate is representative of the real user reputation, even if we set the threshold at 0.95, on average we accept one wrong annotation out of twenty. And the number of correct annotations discarded is even higher.

We propose a second baseline decision method that tries to tackle the second criticism mentioned above, and thus, it takes into account the fact that if a user has reputation p, then he has $p\%$ probability to be right and *1-p%* to be wrong. We accept $p\%$ of

his annotations and we reject *1-p%*, randomly chosen. Function RandomSelectionDecisionStrategy describes the algorithm.

Function RandomSelectionDecisionStrategy

Input: A list of artifacts: s = set(entity(e,-)) made by an agent: Agent(ag)
Output: A list of decisions set(d), d ∈ {`accept`, `reject`}

```
reputation ← get_reputation (ag1)
l ← len(s)
index_accept = random_select (l,reputation)
index_reject = random_select (l,(1-reputation))
result ← []
for i in index_accept do
    result[i] ← accept
for i in index_reject do
    result[i] ← reject
return result
```

Running Example. Recall that *Davide*'s reputation is 0.71 and he contributed fifteen annotations: therefore, eleven of them are accepted and four rejected. The selection of these eleven and four is made randomly.

4.2. *Waisda?*

4.2.1. Dataset Description. Waisda?[4] [Hildebrand et al. 2013] is a video tagging gaming platform launched by the Netherlands Institute for Sound and Vision in collaboration with the public Dutch broadcaster KRO. The game's logic is simple: users watch videos and tag the content. Whenever two or more players insert the same tag about the same video in the same time frame (10 sec.), they are both rewarded. The number of matches[5] for a tag is used as an estimate of its trustworthiness. We note that when a tag is not matched, this does not imply that the tag is untrustworthy. For example, a tag can refer to an element of a video that has not yet been identified, or the tag could belong to a niche vocabulary.

We validate our trust assessment approach by using matches as proxy for the trustworthiness of tag entries produced within the game. Our total corpus contains 122,997 tag entries, with contributions from 265 registered users. We treat the matched tags as positive items of evidence, and the unmatched tags as negative ones. We choose not to focus on the number of matches that a given tag receives. Rather, following the game's logic, we estimate the reputation of users based on their ability to produce tags that agree with other users' tags.

Applying our pipeline, the feature selection algorithm produced the following feature types:

video:. the video that is annotated in the game;
series:. the series the video belongs to;
source:. the broadcasting source for the video;
typing duration:. time taken to type a tag;

[4]http://www.waisda.nl
[5]Matches must be exact on normalized tags (i.e., tag matching is case insensitive, etc.). Typos or synonymities are not taken into consideration.

creation time:. server side timestamp indicating when the tag has been created.[6]

These features are used to produce between 364 to 3453 stereotypes, depending on the size of the training set and how we discretize the features. An example stereotype is "early morning fast taggers annotating a video that was broadcasted by a *source* and belongs to a *series*". During stereotype generation, both *creation time* and the *typing duration* are discretized as follows:

typing duration:. typing duration is grouped in order to evenly distribute the tags into n groups.
creation time:. we extract the day of the week and the hour of the day. Each of the two features is treated independently, and grouped in order to evenly distribute the tags in n groups.

For the results reported below $n = 1, 2, 3, 4, 5, 6, 7$. We tested a number of variations in the number of quantiles used, but these did not impact results noticeably (within a decimal place of the reported result). This is because in Equation (4), the impact of the stereotype-based estimations on the overall trust estimate is low. Nevertheless, it is crucial to allow the possibility to rank the artifacts (tags, in this case) and then select the most trustworthy ones. Also, with respect to discretization, the distribution of tags over the quantiles is computed with respect to the training set. It may be the case that the tag distribution in the test set differs from that in the training set. Our assumption is that the training set and the test set are similar and the quantiles are large enough so that they cover most cases. To cover any outliers, we also add a buffer of an additional quantile when looking at the training set.

4.2.2. Results. We run our system by using 5, 10, 15 tags per user reputation and then predicting the trustworthiness of the remaining tags per user. Table I reports statistics about the distribution of the resulting training and test sets.

Table II reports the results obtained by comparing the predicted trustworthiness of a set of tag entries and their actual trustworthiness.[7] We calculate the results on a per user basis and then aggregate. Thus, we select five, ten, and fifteen tags per user and then predict the trustworthiness of the remaining tags for each user. Not all the users contributed at least five, ten or fifteen annotations each. For each case, the users that contributed fewer annotations were discarded: actually 68% of the users contributed at least five annotations, 53% at least ten and 45% at least fifteen. The percentage of annotations actually used in the three cases is 99%, 98%, and 97%. Of these, respectively 4%, 6% and 8% of the data was used as training set.

The table reports also the results obtained with the two baseline decision strategies. For the first decision strategy (fixed threshold) we report the results with three different thresholds.

At each coverage level, the aggregated precision, recall and accuracy are reported. Precision means the percentage of tags that were correctly predicted to be trustworthy:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}.$$

[6]We note that time is collected on the server side. Therefore, in principle, there may be tags that are provided from different time zones. However, given that game is in Dutch and only covers Dutch programs, we make the assumption that all players are located in the same time zone and thus that the use of server side time is justifiable.
[7]Again, we use the number of matches of a tag as a proxy for trustworthiness.

Recall means the percentage of tags that were predicted to be trustworthy of the total amount of trustworthy tags:

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}.$$

Accuracy means the percentage of correct estimates over the total amount of estimates performed:

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}.$$

F-measure is the harmonic mean of precision and recall:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

Table I. Statistics of the Waisda? dataset. # of annotations per reputation is the number of annotations (i.e., of pieces of evidence) used to build a user reputation. The training set-test set ratio is computed as $\frac{trainingset}{trainingset+testset}$. Only some users provided a number of annotations equal or higher than the # of annotations per reputation, hence only those users are considered in each round of computation. % of users considered shows the percentage of users selected. % of annotations considered indicates the annotations therefore selected.

# annotations per reputation	Training set - test set ratio	% of users considered	% of annotations considered
5	4%	68%	99%
10	6%	53%	98%
15	8%	45%	97%

Table II. Results obtained on the *Waisda?* dataset. We report the results for our system (Stereotype-based system) followed by the results for the two baselines. For the first baseline (fixed threshold), we report the results with three different thresholds (thld.).

Approach	# annotations per reputation	Precision	Recall	Accuracy	F-measure
Stereotype-based	5	70%	61%	56%	65%
Stereotype-based	10	71%	63%	57%	67%
Stereotype-based	15	71%	64%	57%	67%
Baseline 1 (thld. 0.7)	5	76%	61%	61%	68%
Baseline 1 (thld. 0.7)	10	77%	54%	57%	63%
Baseline 1 (thld. 0.7)	15	76%	62%	61%	68%
Baseline 1 (thld. 0.8)	5	82%	33%	50%	47%
Baseline 1 (thld. 0.8)	10	82%	34%	50%	48%
Baseline 1 (thld. 0.8)	15	81%	39%	52%	53%
Baseline 1 (thld. 0.9)	5	0%	0%	33%	0%
Baseline 1 (thld. 0.9)	10	86%	15%	40%	39%
Baseline 1 (thld. 0.9)	15	86%	8%	36%	15%
Baseline 2	5	60%	60%	41%	60%
Baseline 2	10	60%	60%	40%	60%
Baseline 2	15	60%	60%	39%	60%

4.3. *Steve.Museum*

4.3.1. Dataset description. Steve.Museum [US Institute of Museum and Library Services 2012] is a project involving museums and professionals in the cultural heritage domain. Part of the project focuses on understanding the various effects of crowdsourcing cultural heritage artifact annotations. Within the context of this project, experiments were performed using external annotators annotating museum collections (i.e., crowdsourcing museum annotations). A subset of these annotations were evaluated for trustworthiness. In total, 89,671 artifacts from 21 participating museums were tagged by 4,588 users using 480,617 tags. 45,860 of the crowdsourced tags were manually evaluated by professionals at these museums. We use this dataset for our evaluation here.

Each tag was classified by the professional as being part of one of the following classes:

— Todo,
— Judgement-negative, Judgement-positive,
— Problematic-foreign, Problematic-huh, Problematic-misperception, Problematic-misspelling, Problematic-no_consensus, Problematic-personal,
— Usefulness-not_useful, Usefulness-useful

These classes are grouped into three main categories: judgement (a personal judgement by the annotator about the picture), problematic (for several, different reasons) and usefulness (stating whether the annotation is useful or not). Here, we consider only "usefulness-useful" as a positive evaluations (i.e., trustworthy), all the others are considered negative (i.e., untrustworthy). The tags classified as "todo" are discarded, since their evaluation has not been performed yet.

The *Steve.Museum* dataset is provided as a MySQL database and consists of several tables. Those most important for us are: "steve_term", that contains information like the identifiers for the artifact annotated and the words associated with them (tags); "steve_session" that reports information about when the tags are provided and by whom, and "steve_term_review" that contain information about the tag evaluations. We process this information into the necessary inputs for our pipeline - tag entries, evidence and users.

After preprocessing, we run the provenance features selection algorithm, which retrieves the following features:

creation time:. Server-side timestamp[8] indicating when the tag was created.

Based on this, we created stereotypes using the following discretization method:

Day of the week:. extracted from the creation time and grouped according to a predefined number of quantiles. We could have kept the number of weekdays as seven, but we tried to both balance the amount of evidence in each group and find a generic rule to apply in the stereotype creation.
Hour of the day:. extracted from the creation time and grouped according to a predefined number of quantiles.

We used 1, 2, 3, 4, 5, 6, 7 quantiles, resulting in a number of stereotypes ranging from 4 to 42, depending on the number of quantiles chosen.

[8]Like the *Waisda?* case, we make the assumption that the time zones are within one country, in this case, the United States. Our justification is that the participating museums are located in the US and the percentage of foreign language tags is negligible (1.9%). While the United States has multiple time zones, the selected quintiles are not fine-grained enough to be impacted significantly by this variation.

Table III. Statistics of the *Steve.Museum* dataset. # of annotations per reputation is the number of annotations (i.e., of pieces of evidence) used to build a user reputation. The training set-test set ratio is computed as $\frac{trainingset}{trainingset+testset}$. Only some users provided a number of annotations equal or higher than the # of annotations per reputation, hence only those users are considered in each round of computation. % of users considered shows the percentage of users selected. % of annotations considered indicates the annotations therefore selected.

# annotations per reputation	Training set - test set ratio	% of users considered	% of annotations considered
5	13%	57%	96%
10	19%	39%	91%
15	22%	30%	86%

Table IV. Results obtained on the *Steve.Museum* dataset. We report the results for our system (Stereotype-based system) followed by the results for the two baselines. For the first baseline (fixed threshold), we report the results with three different thresholds.

Approach	# annotations per reputation	Precision	Recall	Accuracy	F-measure
Stereotype-based	5	84%	74%	68%	79%
Stereotype-based	10	85%	78%	71%	81%
Stereotype-based	15	85%	86%	76%	85%
Baseline 1 (thld. 0.7)	5	84%	79%	71%	81%
Baseline 1 (thld. 0.7)	10	85%	76%	70%	80%
Baseline 1 (thld. 0.7)	15	85%	87%	77%	86%
Baseline 1 (thld. 0.8)	5	85%	51%	53%	64%
Baseline 1 (thld. 0.8)	10	86%	56%	56%	68%
Baseline 1 (thld. 0.8)	15	86%	61%	60%	71%
Baseline 1 (thld. 0.9)	5	0%	0%	19%	0%
Baseline 1 (thld. 0.9)	10	86%	34%	42%	49%
Baseline 1 (thld. 0.9)	15	87%	21%	33%	34%
Baseline 2	5	77%	26%	33%	39%
Baseline 2	10	75%	21%	30%	33%
Baseline 2	15	75%	19%	29%	30%

4.3.2. Results. We run our system by using five, ten and fifteen tags per user reputation and then predicting the trustworthiness of the remaining tags per user. Table III reports statistics about the distribution of the resulting training and test sets. Table IV reports the results obtained. Again, we are comparing the predicted trustworthiness of a set of tag entries and their actual trustworthiness. In this case, trustworthiness is measured by the rating of an expert. We calculate the results on a per user basis and then aggregate. Thus, we select five, ten, and fifteen tags per user and then predict the trustworthiness of the remaining tags for each user. Not all the users contributed at least five, ten or fifteen annotations each. For each case, the users that contributed fewer annotations were discarded: actually 57% of the users contributed at least five annotations, 39% at least 10 and 30% at least fifteen. The percentage of annotations actually used in the three cases is 96%, 91%, and 86%. Of these, respectively 13%, 19% and 22% of the data was used as training set. The table reports also the results obtained with the two baseline decision strategies. For the first decision strategy (fixed threshold) we report the results with three different thresholds.

At each coverage level, the aggregated precision and recall are reported.

4.4. Discussion

For both test cases, we observe above 70% precision in terms of predicting whether a tag is trustworthy and we are able to select a large majority of trustworthy tags. In both cases, a larger training set guarantees an equal or slightly better performance, although the improvement is higher in the *Waisda?* case. However, with little evidence our approach can still obtain satisfactory results (in the case of *Steve.Museum*, 84% precision with only 13% of the data used for training, 70% precision with 4% of the data used for training in the *Waisda?* case). This is possible thanks to the smoothing of subjective logic that prevents over-reliance on too little evidence as well as the provenance-based generation of stereotypes. These stereotypes provide a fall-back position when there is only a small amount of evidence on a per user basis. This is confirmed also by the fact that our system almost always outperforms both baselines.

In particular, compared to our system, the first baseline lacks robustness, since its performance depends much on the parameter setting. Moreover, in the scenarios represented through our case studies, these parameters (i.e., the thresholds) are likely to be imposed by some policy. In fact, in the cultural heritage and media domain, crowdsourced annotations are used to index and retrieve items. Therefore, it is preferable to reject correct annotations (and thus, waste part of the annotators work) than to accept wrong annotations (thus avoiding to incorrectly index the items annotated). Consequently, the threshold is likely to be set high: since the trust value represents the probability of the annotation to be correct, the higher the threshold, the lower the probability to accept wrong annotations. However, with the highest threshold (0.9), we obtained the poorest performance. In particular, with five items of evidence per reputation and a threshold set to 0.9, no annotation is classified as useful since the highest reputation computable in this case (when all five pieces of evidence are positive) is $\frac{5+1}{5+2} = 0.85714285$. Consequently, in all these cases, precision and recall are equal to zero because the set of annotations estimated to be useful is empty. Precision and recall indicate in fact whether such a set is precise and complete, respectively.

The second baseline is more robust than the first one, but its performance is still worse than that of the system that we propose. This is due to the fact that a random selection is not able to correctly judge the annotations, even though the selection follows the proportion indicated by the reputation.

In order to provide an overall comparison between our system (stereotype-based) and the different baselines (baseline 1 with thresholds 0.7, 0.8, 0.9 and baseline 2) we compared the performance of the system with the performance of each baseline pairwise. The comparison is made by means of a Wilcoxon signed-rank test at 90% confidence level. The test compares the performance (precision, recall, accuracy and F-measure) of our system and of the baseline strategy with 5, 10 and 15 tags per reputation (i.e., the performance indicated in Tables II and IV). If the two systems compared are significantly different, then the system with highest average performance outperforms the other system.

For instance, on the *Steve.Museum* dataset, our system gets 84%, 85% and 85% precision with 5, 10 and 15 annotations respectively, while baseline 2 gets 75%, 77% and 77%. A Wilcoxon signed-rank test at 90% tells us that the two are significantly different, and since the average precision of our system is higher than the average precision of baseline 2, we consider that our system outperforms baseline 2 in this case. Consider that in this case the p-value is 0.0722. In fact, although the two samples are quite different, the test considers the possibility that the two samples actually belong to the same population, but look different because of their size. Thus, we set our confidence level at 90% to tackle the uncertainty due to the small sample size.

Table V. Comparison between approaches on the *Waisda?* dataset. We ran a Wilcoxon signed-rank test at 90% confidence level to check if the results obtained with the baseline decision strategies are significantly different from the results obtained with our stereotype-based system. Each row compares one baseline with our system. The corresponding cells contain the name of the system with the highest performance (if any), or the symbol '=' if the two are not significantly different.

Compared approaches	Precision	Recall	Accuracy	F-measure
Baseline 1 (thld. 0.7) *vs.* Stereotype-based	*Baseline 1 (thld. 0.7)*	=	=	=
Baseline 1 (thld. 0.8) *vs.* Stereotype-based	*Baseline 1 (thld. 0.8)*	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*
Baseline 1 (thld. 0.9) *vs.* Stereotype-based	=	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*
Baseline 2 *vs.* Stereotype-based	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*

Table VI. Comparison between approaches on the *Steve.Museum* dataset. We ran a Wilcoxon signed-rank test at 90% confidence level to check if the results obtained with the baseline decision strategies are significantly different from the results obtained with our stereotype-based system. Each row compares one baseline with our system. The corresponding cells contain the name of the system with the highest performance (if any), or the symbol '=' if the two are not significantly different.

Approach	Precision	Recall	Accuracy	F-measure
Baseline 1 (thld. 0.7) *vs.* Stereotype-based	=	=	=	=
Baseline 1 (thld. 0.8) *vs.* Stereotype-based	=	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*
Baseline 1 (thld. 0.9) *vs.* Stereotype-based	=	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*
Baseline 2 *vs.* Stereotype-based	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*	*Stereotype-based*

Tables V and VI report the results of this comparison for the *Waisda?* and *Steve.Museum* respectively. It is interesting to note that only in two cases the baseline outperforms our system, only on the *Waisda?* dataset. However, this outperformance regards only the precision, and the recall of our system is always equal or higher than the recall of the baselines. As a result, the overall performance (accuracy and f-measure) of our system are always equal or higher than the overall performance of the baselines: in other words, on average, our system provides more accurate results than the baseline.

4.5. Improvements Over Our Prior Work

We provide here a comparison with previous work that we proposed to tackle this same problem. Each of the frameworks and algorithms previously introduced were evaluated on a variety of datasets that include at least one of those used here.

4.5.1. User-centric System, Ranking Using Semantic Similarity. The previous system estimated the user reputation using a set of evaluated annotations [Ceolin et al. 2013]. Annotations could obtain only a positive or negative evaluation. For each new annotation provided by the same user, that system computed the semantic similarity with the already evaluated annotations available and used this value as a weight for the corresponding positive or negative piece of evidence. Thus, annotations were ranked, and then evaluated using the annotator reputation, hence accepting only the first p% annotations, where p is the user reputation. We evaluated that system on the *Steve.Museum* dataset, with results that are similar to those presented in this paper. However:

— in the previous work we obtained slightly lower performance;
— the previous work was computationally less efficient, due to the fact that, per user, each new annotation is compared against each piece of evidence. We proposed also

a computationally efficient version of such a system based on clustering, which improves the computational effort while not compromising the performance sensibly, but still requires a quite heavy preprocessing step.

4.5.2. Provenance-centric System, Ranking Using Semantic Similarity. The previous system works as the user-centric system, but instead of using users, focuses on stereotypes [Ceolin et al. 2014]. Thus, it computed stereotype reputations, ranked the annotations that belong to a given stereotype using a semantic similarity measure, and then accepted or rejected them based on the stereotype reputation. The performance of that system has also been evaluated on the *Steve.Museum* dataset, with results that are more accurate than those obtained with the system presented in this paper. Provenance stereotypes were computed by grouping and discretizing the provenance available for the annotations similar to what we do in this paper. However, for each stereotype, we collected a fixed amount of evidence, and we used it to compute a stereotype reputation. Then, the remaining annotations were evaluated. In that case, the stereotypes are created a priori: therefore the annotations are unevenly distributed among them, and despite the system proposed in this paper, some stereotypes are not used in the evaluation since they do not have enough evidence. The system that we propose here, instead, provides a more detailed set of trust estimations, since it creates stereotypes such that the evidence at our disposal is evenly distributed among them. In this manner, a reputation is computed for each of the stereotypes in the training set, thus increasing the possibility to compute a trust assessment for each of the tags in the test set (in case these belong to the same stereotypes of the tags in the training set).

4.5.3. User Reputation Combined with Provenance-based Trust Estimates Computed using Support Vector Machines. That system computed a user reputation similar to the user-centric system recalled above. When not enough pieces of evidence about a given user were available, trust was estimated by running a support vector machine system that predicts annotation trust based on provenance features [Ceolin et al. 2012a]. One of the limitations of that system is the fact that to run it, one had to arbitrarily set a threshold for the minimum number of pieces of evidence accepted for a user reputation and the minimum trust level that an accepted annotation needed to have. These were both arbitrary parameters. The accuracy achieved by that system is very high, but at the cost of accepting or rejecting almost all annotations, which is a risky policy: when it is too restrictive, it might discourage users, and when it is too permissive, it may accept wrong annotations. So, the choice of the threshold value is not trivial, and the fact that the system we propose in the current paper does not need it is an important advantage.

In the system that we propose in the current paper we overcome such a limitation, since it is no longer necessary to set a minimum trust threshold (this is determined by the user reputation), and the number of items considered to build a user reputation does not affect the merge of user- and provenance-based trust estimates. We evaluated that system on the *Waisda?* dataset. Also, the accuracy obtained with this system, although leaves room for improvement, is higher than the accuracy obtained with several thresholds carefully chosen to evaluate that system.

5. ROBUSTNESS ANALYSIS

In this section we propose a qualitative analysis of the robustness of the trust management framework that we propose. We refer to the attacks enlisted by Krishnaprasad et al. [Krishnaprasad et al. 2014] and we analyze in which situations the framework is able to resist them.

5.1. Ballot-stuffing Attack and Bad-mouthing Attack

> *"In ballot-stuffing attack, majority of the recommenders collude to unfairly promote the trustworthiness of an undeserving trustee. In bad-mouthing attack, majority of the recommenders collude to unfairly denigrate the trustworthiness of a victim."*

Since we do not make use of recommendations in our system, users cannot collude to directly influence the trustworthiness of other trustees. Users might collude only to unfairly promote a particular stereotype, but this will affect all the artifacts created with that stereotype, not necessarily the trustworthiness of a user.

— The fact that the artifacts contributed by a given user belong to a given stereotype does not affect the user reputation.
— The artifact stereotype affects only its ranking and, thus, its probability to be accepted.

For these reasons, such a user collusion could be used to make sure (or to increase the chance) that some artifacts are accepted or rejected, but it cannot promote or denigrate the trustworthiness of a user.

Running Example. Referring to the example introduced before, suppose that *Archana* and *Willem*, two platform users, collude to either promote or denigrate *Davide*. However, given that *Davide*'s reputation is based only on museum evaluations, they cannot modify it. The only thing they can do is to modify the reputation of one or more of the stereotypes which Davide's annotations belong to, by providing many trustworthy or untrustworthy annotations (respectively to promote or denigrate it), but this would result in:

— the annotations provided by *Davide* that belong to "attacked stereotype" being decreased or increased of rank, depending of whether the attack is intended to promote or denigrate *Davide*. Consequently, correct annotations could be denigrated and rejected and incorrect annotations could be promoted and accepted.
— the same effect caused to other annotations belonging to the same stereotype and provided by other users.

Therefore, these attacks do not provide the intended result because there is no way to affect the user reputation by means of recommendations. The user reputation is based only on the evaluation from the museum (or equivalent institution). However, this attack could affect the overall performance of the system by inducing acceptance of wrong annotations and the rejection of correct annotations.

5.2. Newcomer and Sybil Attack

> *"In newcomer attack, a malicious trustee creates new identity to avoid detection by a trust system that tracks history of interactions. In Sybil attack, a malicious trustee creates multiple fake identities to exert undue adverse influence."*

These attacks are prevented by collecting a minimum amount of evidence before deciding about the contribution made by a given user: newcomers' contributions are necessarily accepted or rejected based on their history, and if the history is not long enough, then other evidence is collected before a decision is taken.

In case of sybil attack, the effect that users may exert regards single artifacts, like in the case of the ballot-stuffing and bad-mouthing attacks.

Running Example. In our example, the newcomer attack is avoided since *Davide*'s reputation is computed by the museum after having collected a fixed amount of evidence. The sybil attack could affect the stereotypes reputations if the museum would consider good or wrong annotations belonging to specific stereotypes added by the fake identities.

5.3. Sleeper and On-off Attack

> *"A malicious trustee acquires high reputation/trust by behaving well for long durations and then behaving bad intermittently. The sleeper attack is also called betrayal attack, and on-off attack is also called inconsistency attack."*

Our framework is susceptible to these attacks. In fact, if a user behaves well for the initial period (i.e., while the evidence for computing his reputation is collected) and then betrays the system, then we would not be able to prevent such an attack. This could be circumvented by periodically checking the reliability of the computed user reputation. These checks have to be thought through in order to balance both the computational effort and the risk to suffer an attack (so, for instance, their periodicity should be randomized). Future research will be devoted to this problem.

Running Example. In our example, the museum computes *Davide*'s reputation using the first five annotations that he provides: four good ones and one bad one, and he obtains a reputation of 0.71. If he then decides to contribute only one good annotation out of every ten, the system is unable to detect this change. Likewise if he does this with long pauses between contributions. To solve this, we would need to periodically check the validity of the reputation.

5.4. Conflicting Behavior Attack

> *"In conflicting behavior attack, the attacker uses "divide and conquer" strategy by providing conflicting recommendations on a trustee to multiple trustworthy sources. When a victim seeks recommendations from these trustworthy sources, which faithfully transmit the attackers views, the victim ends up getting conflicting recommendations on the trustee, thereby causing it to incorrectly reduce its trust in a subset of trustworthy sources (recommenders). This hampers the overall "morale"."*

This attack is also avoided by the fact that we do not make use of recommendations. A user might decide to provide conflicting annotations belonging to different stereotypes, but this, again, would affect only single items, and not the reputation of specific users. Thus, this specific attack has a limited effect on our system, similar to the ballot-stuffing and the bad-mouthing attacks.

Running Example. As said, this attack is not possible in our system because we do not make use of recommendations. Again, colluders can induce consequences similar to those described in Section 5.1 if they intentionally provide conflicting annotations for a given stereotype.

6. CONCLUSION

This paper introduces a pipeline for estimating the trustworthiness of artifacts based on the analysis of their provenance information. The pipeline is composed of several algorithms, namely:

— an algorithm for feature extraction from provenance graphs;
— an algorithm for stereotype creation based on provenance and one for collecting stereotype-related evidence;

— an algorithm for reputation computation, with two variants, one for computing the reputation of users and another one for computing the reputation of stereotypes;
— an algorithm for artifact trust estimation;
— an algorithm for selecting artifacts on the basis of their estimated trustworthiness.

Each of these algorithms can be applied individually. However, the entire pipeline provides a powerful means to analyze the trustworthiness of artifacts.

Importantly, we have demonstrated that the use of provenance to augment reputation in trust estimation provides two key benefits: first, it provides a mechanism for ranking and selecting artifacts created by the same user; and second, it allows an artifact's trustworthiness to be determined when little evidence about the user that produced the artifact is available. In principle, these two benefits might have been achieved by using another information class instead of provenance. However, such an information class would need to present these two characteristics in order to effectively replace provenance: (1) correlate with artifacts trustworthiness and (2) be an attribute of all the artifacts.

The pipeline has been evaluated over two tagging corpora. Precision ranges from 67% to 85% depending on the dataset and training size used. Furthermore, recall can reach up to 82%. Importantly, performance is satisfactory when only a minimum amount of evidence is employed (up to 83% precision and 79% recall when using 4% of the data for training in the Steven.Museum case). This shows that we have satisfactorily addressed both sub-challenges outlined in Section 1: providing reliable trust estimates while reducing the workload of the organization providing sample artifacts evaluations.

In the future, we plan to use provenance in order to be able to assess the trustworthiness of artifacts created by completely unknown or anonymous users. Likewise, we plan to introduce a mechanism to periodically check and eventually revise the user reputations computed. This will provide us protection against sleeper and on-off attacks. Moreover, we are investigating the deeper use of the semantics of PROV to expand the number and types of features that we can extract from provenance graphs. Also, we aim at integrating external sources of evidence (e.g., from the Web) to increase the accuracy and recall of our trust evaluations. Each of these extensions will be embedded in the pipeline presented in this paper, possibly by utilizing additional subjective logic operators. For instance, semantic similarity weighing and partial evidence opinions [Ceolin et al. 2012b] will be used to incorporate Social Web data and contextualize them. Lastly, we will explore alternative approaches for building provenance stereotypes in standarized manner.

7. ACKNOWLEDGEMENTS

This publication was supported by the Dutch national program COMMIT.

REFERENCES

I. Altintas, M. K. Anand, D. Crawl, S. Bowers, A. Belloum, P. Missier, B. Ludäscher, C. A. Goble, and P. M. A. Sloot. 2010. Understanding Collaborative Studies through Interoperable Workflow Provenance. In *Provenance and Annotation of Data and Processes - Third International Provenance and Annotation Workshop (IPAW 2010)*. Springer, Troy, NY, USA, 42–58.

D. Artz and Y. Gil. 2007. A Survey of Trust in Computer Science and the Semantic Web. *Journal of Web Semantics* 5, 2 (2007), 131–197.

C. Bizer and R. Cyganiak. 2009. Quality-driven information filtering using the WIQA policy framework. *Journal of Web Semantics* 7, 1 (Jan. 2009), 1–10.

J.J. Carroll, C. Bizer, P. Hayes, and P. Stickler. 2005. Named graphs, provenance and trust. In *Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*. ACM, Chiba, Japan, 613–622.

D. Ceolin, P. Groth, and W. R. Van Hage. 2010. Calculating the Trust of Event Descriptions using Provenance. In *Proceedings of the Second International Workshop on the role of Semantic Web in Provenance Management (SWPM 2010), co-located with the 9th International Semantic Web Conference (ISWC 2010)*. CEUR-WS.org, Shanghai, China, 11–16.

D. Ceolin, P. Groth, W. R. van Hage, A. Nottamkandath, and W. Fokkink. 2012a. Trust Evaluation through User Reputation and Provenance Analysis. In *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2012) at the 11th International Semantic Web Conference (ISWC 2012)*. CEUR-WS.org, Boston, MA, USA, 15–26.

Davide Ceolin, Archana Nottamkandath, and Wan Fokkink. 2012b. Subjective Logic Extensions for the Semantic Web. In *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2012), co-located with the 11th International Semantic Web Conference (ISWC 2012) (CEUR Workshop Proceedings)*, Vol. 900. CEUR-WS.org, Boston, MA, USA, 27–38.

D. Ceolin, A. Nottamkandath, and W. Fokkink. 2013. Semi-automated Assessment of Annotation Trustworthiness. In *Proceedings of the Eleventh Annual International Conference on Privacy, Security and Trust (PST 2013)*. IEEE Computer Society, Tarragona, Catalonia, 325–332.

D. Ceolin, A. Nottamkandath, and W. Fokkink. 2014. Efficient Semi-automated Assessment of Annotation Trustworthiness. *Journal of Trust Management* 1 (May 2014), 1–31. Issue 1.

L. De Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. 2011. Reputation systems for open collaboration. *Communications of the ACM* 54, 8 (2011), 81–87.

M. K. Denko and T. Sun. 2008. Probabilistic Trust Management in Pervasive Computing. In *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008 (EUC '08)*. IEEE Computer Society, Shanghai, China, 610–615.

M. Ebden, T. D. Huynh, L. Moreau, S. Ramchurn, and S. Roberts. 2012. Network analysis on provenance graphs from a crowdsourcing application. In *Provenance and Annotation of Data and Processes - 4th International Provenance and Annotation Workshop (IPAW 2012)*. Springer-Verlag, Santa Barbara, California, 168–182.

S. Ganeriwal, L. Balzano, and M. B Srivastava. 2008. Reputation-based Framework for High Integrity Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* 4 (June 2008), 1–37. Issue 3.

Y. Gil, S. Miles (eds.), K. Belhajjame, H. Deus, D. Garijo, G. Klyne, P. Missier, S. Soiland-Reyes, and S. Zednik. 2013. *PROV Model Primer*. W3C Working Group Note NOTE-prov-primer-20130430. World Wide Web Consortium. http://www.w3.org/TR/prov-primer/

J. Golbeck. 2006. Trust on the World Wide Web: A Survey. *Foundations and Trends in Web Science* 1, 2 (2006), 131–197.

P. Groth and L. Moreau (eds.). 2013. *PROV-Overview. An Overview of the PROV Family of Documents*. W3C Working Group Note NOTE-prov-overview-20130430. World Wide Web Consortium. http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

O. Hartig and J. Zhao. 2009. Using web data provenance for quality assessment. In *Proceedings of the First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009), co-located with the 8th International Semantic Web Conference (ISWC 2009)*. CEUR-WS.org, Washington D.C., USA, 1–6.

M. Hildebrand, M. Brinkerink, R. Gligorov, M. van Steenbergen, J. Huijkman, and J. Oomen. 2013. Waisda?: Video Labeling Game. In *Proceedings of the 21st ACM International Conference on Multimedia (MM '13)*. ACM, Barcelona, Spain, 823–826.

S. Javanmardi, C. Lopes, and P. Baldi. 2010. Modeling user reputation in wikis. *Statistical Analysis Data Mining* 3, 2 (April 2010), 126–139.

A. Jøsang. 2001. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9, 3 (2001), 279–212.

A. Jøsang, R. Ismail, and C. Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43, 2 (2007), 618 – 644.

T. Krishnaprasad, P. Anantharam, C. A. Henson, and A. P. Sheth. 2014. Comparative Trust Management with Applications: Bayesian Approaches Emphasis. *Future Generation Computer Systems* 31 (2014), 182–199.

H. Masum and M. Tovey (Eds.). 2012. *The Reputation Society*. MIT Press, Boston, MA, USA.

L. Moreau, P. Missier (eds.), K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, and C. Tilmes. 2013. *PROV-DM: The PROV Data Model*. W3C Recommendation REC-prov-dm-20130430. World Wide Web Consortium. http://www.w3.org/TR/2013/REC-prov-dm-20130430/

K. O'Hara. 2012. *A General Definition of Trust*. Technical Report. University of Southampton.

A. V. Pantola, S. Pancho-Festin, and F. Salvador. 2010. Rating the raters: a reputation system for wiki-like domains. In *Proceedings of the 3rd International Conference on Security of Information and Networks (SIN 2010)*. ACM, Taganrog, Rostov-on-Don, Russian Federation, 71–80.

I. Pinyol and J. Sabater-Mir. 2013. Computational Trust and Reputation Models for Open Multi-agent Systems: A Review. *Artificial Intellelligence Review* 40, 1 (June 2013), 1–25.

S. Rajbhandari, O. F. Rana, and I. Wootten. 2008. A fuzzy model for calculating workflow trust using provenance data. In *Proceedings of the 15th ACM Mardi Gras Conference: From lightweight mash-ups to lambda grids: Understanding the spectrum of distributed computing requirements, applications, tools, infrastructures, interoperability, and the incremental adoption of key capabilities (MG 2008)*. ACM, Baton Rouge, LA, USA, 1–8.

S. Rajbhandari, I. Wootten, A. S. Ali, and O. F. Rana. 2006. Evaluating Provenance-based Trust for Scientific Workflows. In *Proceeding of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2006)*, Vol. 1. IEEE Computer Society, Singapore, 365–372.

J. Sabater and C. Sierra. 2005. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review* 24 (2005), 33–60.

Y. Sun, W. Yu, Z. Han, and K.J.R Liu. 2006. A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*. IEEE Computer Society, Barcelona, Spain, 230–236.

US Institute of Museum and Library Services. 2012. Steve Social Tagging Project. (June 2012).

I. Zaihrayeu, P.P. da Silva, and D. L. McGuinness. 2005. IWTrust: Improving User Trust in Answers from the Web, In Proceedings of the 3rd International Conference on Trust Management (iTrust2005). *Trust Management* 3477 (2005), 384–392.