Corroboration de vues discordantes fondée sur la confiance*

Alban Galland† Serge Abiteboul† Amélie Marian‡
Pierre Senellart§

Résumé. Cet article traite de la corroboration d'informations, dans le contexte de vues exprimant des opinions sur des faits de façon éventuellement contradictoire. Il s'agit de prédire si un fait est vrai ou faux. Des méthodes d'agrégation simples comme le vote donnent déjà de bons résultats, mais nous présentons dans cet article des algorithmes qui tiennent compte de la confiance dans les vues pour améliorer les prédictions. Les trois algorithmes proposés sont des algorithmes de point fixe correspondant à différents niveaux de complexité du modèle probabiliste sous-jacent. Ils estiment à la fois la valeur de vérité des faits et la confiance dans les vues. Cet article présente une étude expérimentale sur des données synthétiques et réelles. Ces expériences montrent dans quelle mesure et dans quel contexte nos algorithmes peuvent améliorer les résultats par rapport au vote. La corroboration apparaît naturellement dans un grand nombre d'applications, comme la sélection de sources dans le Web sémantique, les tests de qualité de données ou le nettoyage d'annotations sémantiques dans les réseaux sociaux. Ce travail pose donc les bases de techniques plus complexes pour traiter les problèmes précédents.

Mots-clefs : corroboration, vue, confiance, modèle probabiliste, point fixe, contradiction, espérance-maximisation

*This work has been partially funded by the Advanced European Research Council grant Webdam `http://webdam.inria.fr/`

†INRIA Saclay – Île-de-France, 4 rue J. Monod, 91893 Orsay Cedex, France, first.last@inria.fr.

‡Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854-8019, United States, amelie@cs.rutgers.edu

§Institut Télécom, Télécom ParisTech, CNRS LTCI, 46 rue Barrault, 75634 Paris Cedex 13, France, pierre.senellart@telecom-paristech.fr.

1 Introduction

The Web provides an interface to access a wide variety of information and viewpoints from individual Web sources that have different degree of trustworthiness based on their origin or bias. The most daunting problem when trying to answer a question seems not to be *where* to find an answer, but *which* answer to trust among the ones reported by different Web sources. This happens not only when no true answer exists, because of some opinion or context differences, but also when one or more true answers are expected. Such conflicting answers can arise from disagreement, outdated information, or simple errors.

Simple questions often yield disagreeing answers from different sources. As an example, the birth date of Napoleon Bonaparte, a contentious topic of importance to historians as it determines whether Napoleon was born French or Italian, is reported as August 15, 1769 or as January 7, 1768 depending on the sources. A more familiar everyday example is a simple professional contact information search: contact information is time-dependent; yet because of the nature of Web sources, many sources will continue to list outdated information if a person has switched jobs. For instance, as of the writing of this paper, a Google search for "Mor Naaman" lists three possible affiliations in the first ten results: Stanford University, Yahoo! Research Berkeley, and SCILS, Rutgers University. The correct current affiliation, SCILS, does not appear in first position. In addition, sources may identify the object incorrectly; in the case of a contact search this can happen in the presence of homonyms (the first page of Google results for "Mor Naaman Facebook" returns two separate Facebook profiles), misspellings or name changes.

We consider each Web source as a separate view over the data. To accurately answer a question in the presence of conflicting information, a natural approach is to simply count the number of occurrences of each answer, i.e., the number of views reporting each answer. This simple voting strategy performs well in many scenarios but is easily misguided in a Web environment where many sources can either malignantly collude to propagate false information, or naively replicate outdated or wrong data. The quality of the views should then be taken into account when corroborating answers to identify the best answer to a query. Without *a priori* knowledge on the quality, or trustworthiness, of views, or on the correctness of answers, we are left with a recursive definition: a correct answer is returned by many trusted views and a trustworthy view returns many correct answers. In this paper, we propose fixpoint computation techniques that derive estimates of the truth value of facts reported by a set of views, as well as estimates of the quality of the views.

We believe that data corroboration can improve data quality in a wide range of domains, including source selection in the semantic Web [14], semantic annotation cleaning in social networks, and information extraction. For instance, information extraction tools [5] typically return one or more answers to an information extraction task; using several different tools might lead to different answers. By corroborating answers from different tools over a set of tasks, we can not only identify the most likely answer, but also assess the quality (trust) of each extraction tool. Our corroborative approach can also be useful for collaborative tagging systems in social networks [8, 13]. In such systems many

independent users assign tags to objects; the tags are aggregated to create a description, or categorization, of the object. By including not only frequency information but also user trustworthiness or expertise in the aggregation process, we can improve the quality of the collaborative filtering system.

We first introduce a probabilistic data model for corroboration that takes into account the uncertainty associated to facts reported by the views, as well as the limited coverage of the views. Our main contribution consists in three algorithms, namely COSINE, 2-ESTIMATES and 3-ESTIMATES, that estimate the truth values of facts and the trust in sources. They all refine these estimates iteratively until a fixpoint is reached. Their particularities are as follows:

- COSINE is based on the cosine similarity measure that is popular in Information Retrieval [12].
- 2-ESTIMATES uses two estimators for the truth of facts and the error of views that are proved to be perfect in some statistical sense.
- 3-ESTIMATES refines 2-ESTIMATES by also estimating how hard each fact is, i.e. the propensity of sources to be wrong on this fact.

We present an experimental evaluation of the algorithms with respect to two baseline algorithms, VOTING and COUNTING, over both synthetic and real-world data. Our results show that our three algorithms are able to predict correct truth values better than the baseline algorithms in cases where views have various degrees of trustworthiness. Furthermore, we show that in general, 3-ESTIMATES provides better estimates than the other two, which demonstrates the interest of taking into account the hardness of facts.

The paper is organized as follows. The probabilistic data model is described in Section 2. Our three algorithms as well as the base algorithms are presented in Section 3. Experiments are discussed in Section 4. We discuss some related work and conclude in Section 5.

2 Model

The opinion of sources can be seen as views over the real world W. Views report beliefs that are of positive or negative statements. Based on these beliefs, the problem is to "guess" what the real world actually is.

Let $\mathcal{F}$ be a set $\{f_1 \dots f_n\}$ of *facts*. A *view* (over $\mathcal{F}$) is a (partial) mapping from $\mathcal{F}$ to the set $\{T, F\}$ (T stands for *true*, and F for *false*). We have a set of views $\mathcal{V} = \{V_1 \dots V_m\}$ and from them we try to estimate the real world W, a total mapping from $\mathcal{F}$ to the set $\{T, F\}$. From a mathematical viewpoint, based on some probabilistic model, we want to estimate the most likely W given the views.

For instance, W may state that the fact "Paris is the capital city of France" holds. Some views may agree with W on this fact while other views may believe that "Lyon is the capital city of France". A particular case is when views only believe in positive facts,

as is often the case on the Web. Nevertheless, negative facts can still be introduced by functional dependencies. Suppose we know that France has exactly one capital city. If a source states "Paris is the capital city of France", then it also states implicitly that it does not believe "Lyon is the capital city of France". We explain the relationship between functional dependencies and negative statements in more detail further.

The underlying probabilistic model we assume is described by Equation (1):

$$\begin{cases} \mathbb{P}(V_i(f_j) \text{ is undefined}) = \varphi(V_i)\varphi(f_j) \\ \mathbb{P}(V_i(f_j) = \neg W(f_j)) = (1 - \varphi(V_i)\varphi(f_j))\varepsilon(V_i)\varepsilon(f_j) \\ \mathbb{P}(V_i(f_j) = W(f_j)) = (1 - \varphi(V_i)\varphi(f_j))(1 - \varepsilon(V_i)\varepsilon(f_j)) \end{cases} \tag{1}$$

In this model, views ignore some facts and make errors. First, with some probability $\varphi(V_i)\varphi(f_j)$, view V_i ignores fact f_j, i.e., $V_i(f_j)$ is undefined. Now, when $V_i(f_j)$ is defined, V_i makes an error on f_j (with respect to W) with probability $\varepsilon(V_i)\varepsilon(f_j)$. The functions φ, ε define the *ignorance* and *error factors* respectively. Besides estimating W, we are interested in estimating these factors as well. Note that while $\varepsilon(V_i)$ and $\varepsilon(f_j)$ represent the error factors for views and facts, they cannot be interpreted as probabilities without normalization, although their product is a probability (and similarly for $\varphi(V_i)$ and $\varphi(f_j)$).

In most scenarios, views only make positive statements, typically giving, for some query, the answer they have the most confidence in, but not giving the list of all possible false answers (which can be of unbounded size). For instance, it is unlikely that a view would return a list of all cities of France (or of the world) that are not the correct answer to the query "what is the capital city of France?" Nevertheless, we focus on the situation where we have both positive and negative statements and use functional dependency information, if available, to infer possibly omitted negative facts. In particular, we consider functional dependencies of the form "there is one and only one true answer to this question". More formally, we define a set of queries $\mathcal{Q}$ and each fact is associated with a reference query $ref(f_j) \in \mathcal{Q}$. Then for each query $q \in \mathcal{Q}$, we impose the following *functional dependency constraints*:

$$\begin{cases} \exists f_j \in \mathcal{F}, ref(f_j) = q \wedge W(f_j) = T \\ \forall f \in \mathcal{F} - \{f_j\}, ref(f) = q \Rightarrow W(f) = F \end{cases} \tag{2}$$

These constraints express that each query has exactly one answer. We show in Section 3 how we use Equation (2) to transform a problem with functional dependencies into a related problem with positive and negative statements.

3 Algorithms

This section presents three algorithms to estimate the real world W and error factors $\varepsilon(f_j)$ and $\varepsilon(V_i)$. In the model previously presented, the ignorance factors $\varphi(f_j)$ and $\varphi(V_i)$ are independent of these parameters and their estimation is relatively straightforward given the *structure* of the views, $\mathcal{S} = \{(V_i, f_j) \in \mathcal{V} \times \mathcal{F} \mid V_i(f_j) \text{ is defined}\}$. In the following,

$\Theta(\cdot)$ denotes the estimates (given by each algorithm) of the different parameters (notably, error factors and truth values).

Baseline Algorithms. We will compare our algorithm to the following VOTING baseline:

$$\Theta(W(f_j)) = \begin{cases} T & \text{if } \dfrac{|\{V_i : V_i(f_j) = T\}|}{|\{V_i : (V_i, f_j) \in \mathcal{S}\}|} \geqslant 0.5 \\ F & \text{otherwise} \end{cases}$$

This algorithm corresponds to choosing the assessment of the majority about the fact. Note that the estimated truth of a fact only depends on the views stating something about it. A straightforward estimate of the error factor of each view would then make use of the estimated truth value for each fact (say, by assigning as error factor of view i the percentage of estimated true assertions of this view). It is natural to use in turn this estimated error factor to improve the precision of the estimated truth values of facts. This corroboration process is the basis of the 2-ESTIMATES method presented further.

In some cases, we have no mapping to F, for example because the views only give positive statements, in a context where no functional dependencies are assumed. Obviously, the VOTING baseline maps all facts to T in this particular case, which is not helpful. Another baseline is more adapted to this case, namely COUNTING. The method ignores the negative links. More precisely,

$$\Theta(W(f_j)) = \begin{cases} T & \text{if } \dfrac{|\{V_i : V_i(f_j) = T\}|}{\max_f |\{V_i : V_i(f) = T\}|} \geqslant \eta \\ F & \text{otherwise} \end{cases}$$

where η is a fixed threshold. It is typically difficult to set such a threshold that should depend on the data distribution. In our experiments, we fix it to 0.5. This basically consists in assigning T to *popular* facts, in other words facts that are often asserted.

Remark. This popularity notion is reminiscent of the PageRank [4] popularity score for pages of the World Wide Web or, more generally, for nodes in a graph. This suggests using PageRank on the positive votes. PageRank is actually (up to the addition of random jumps, that mostly serve to guarantee the convergence of the algorithm) the equilibrium measure of the random walk in the graph. Observe that, when there is no mapping to F, $\mathcal{V}$ can be seen as a bipartite undirected graph G between views and facts: there is an edge between view V_i and fact f_j if $V_i(f_j) = T$. Importance scores for views and facts can then be computed as the PageRank scores in the view-view and fact-fact graphs obtained by considering all paths of length 2 in G. However, since these two graphs are undirected (G itself is an undirected graph), it can be shown that the equilibrium measure of the random walk is proportional to the degree of the nodes in the graph [9]. Let us restate this result: in the case of an undirected graph, such as those we obtain by considering views that assert the same facts, or facts asserted by the same views, PageRank amounts to the same as our COUNTING baseline. This is

actually only true if the *damping factor* is close to 1, that is, if the probability of random jumps is small. At the limit where a random jump is done at each step of the PageRank computation, the equilibrium measure is obviously uniform and therefore uncorrelated with the degree. We experimented with a typical value for the damping factor (0.85, i.e., 15% probability of performing a random jump) and obtained results very similar to COUNTING.

There is no obvious extension to PageRank with negative links. Our fixpoint methods can be seen as an extension of the random walk interpretation of PageRank to a case with positive and negative links. We also considered an extension based on the cash flow interpretation of PageRank developed in [1] and the algorithm it suggests. We obtained improvements over the baseline methods. However, we chose not to present that algorithm because our other techniques outperform it.

Estimation of Two Series of Parameters. We present in this section two different algorithms that aim to estimate two series of parameters: the truth of facts, and the trustworthiness of views.

We first present a heuristic approach for estimating the truth values of facts and the trustworthiness of views. It is based on the classical cosine similarity measure that is popular in information retrieval [12], hence the name COSINE for this method. We use an alternative representation where these variables have values -1 (false facts, systematically wrong views), 0 (indeterminate facts, views with random statements) or 1 (true facts, perfect views). The idea is then to compute, for each view V_i, given a set of truth values for facts, the similarity between the statements of V_i, viewed as a set of ± 1 statements on facts, and the predicted real world. The technique is precisely described in Algorithm 1. Observe that to improve the stability of the method, we set the new value of the estimation to be a linear combination of the old value and the predicted cosine similarity. As for the estimate of the truth value of facts given the trustworthiness of views, we use a simple averaging, except that we give more weight to predictable views, that is views with high $\Theta(\varepsilon(V_i))^2$ (consistently often correct, or consistently often wrong). We also experimented with a weighting of $|\Theta(\varepsilon(V_i))|$, with similar results. In the initialization phase, estimates are set as if all facts were true. The alternative representation (trustworthiness and truth values between -1 and 1) can easily be mapped to that of Section 2: trustworthiness of the views are estimated as $\frac{\Theta(\varepsilon(V_i))+1}{2}$ and facts are predicted true when $\Theta(W(V_i)) > 0$.

Our second algorithm is more closely related to our probabilistic model. As with COSINE, it focuses on the estimation of $W(f_j)$ (or, more precisely, the probability that $W(f_j) = T$) for each fact f_j, and $\varepsilon(V_i)$ for each view V_i. To simplify, we assume, for this algorithm, that error factors are independent of facts, that is, $\varepsilon(f_j) = 1$ for all f_j. The idea is to iteratively find a good estimate of the $\varepsilon(V_i)$ given $\mathbb{P}(W(f_j) = T)$, and conversely, using a fixpoint computation. As described in Algorithm 2, we first initialize the parameters as if all the views where true about W, then successively estimate one set of parameters given the other one and the views, until convergence. We proved that the estimates that are used in 2-ESTIMATES are valid when $\mathcal{S}$ is given, in the sense that

Algorithm 1 COSINE

Require: $\mathcal{F}, \mathcal{V}, \mathcal{S}$

Ensure: an estimate of $\varepsilon(V_i)$ for each view, an estimate of $W(f_i)$ for each fact

for all $V_i \in \mathcal{V}$ **do** {*Initialization*}

$\Theta(\varepsilon(V_i)) \leftarrow \frac{|\{f_j \,|\, V_i(f_j)=T\}|-|\{f_j \,|\, V_i(f_j)=F\}|}{|\{f_j \,|\, V_i(f_j)\in\mathcal{S}\}|}$

end for

for all $f_j \in \mathcal{F}$ **do**

$\Theta(W(f_j)) \leftarrow 1$

end for

repeat {*Core of the algorithm*}

for all $V_i \in \mathcal{V}$ **do** {η *is a constant (e.g.,* $\eta = 0.2$*)*}

$posFacts \leftarrow \sum_{\substack{f_j\in\mathcal{F} \\ V_i(f_j)=T}} \Theta(W(f_j))$

$negFacts \leftarrow \sum_{\substack{f_j\in\mathcal{F} \\ V_i(f_j)=F}} \Theta(W(f_j))$

$norm \leftarrow \sqrt{|\{f_j \in \mathcal{F} \,|\, V_i(f_j) \in \mathcal{S}\}| \times \sum_{\substack{f_j\in\mathcal{F} \\ V_i(f_j)\in\mathcal{S}}} \Theta(W(f_j))^2}$

$\Theta(\varepsilon(V_i)) \leftarrow (1-\eta) \times \Theta(\varepsilon(V_i)) + \eta \times \frac{posFacts-negFacts}{norm}$

end for

for all $f_j \in \mathcal{F}$ **do**

$posViews \leftarrow \sum_{\substack{V_i\in\mathcal{V} \\ V_i(f_j)=T}} \Theta(\varepsilon(V_i))^3$

$negViews \leftarrow \sum_{\substack{V_i\in\mathcal{V} \\ V_i(f_j)=F}} \Theta(\varepsilon(V_i))^3$

$norm \leftarrow \sum_{\substack{V_i\in\mathcal{V} \\ V_i(f_j)\in\mathcal{S}}} \Theta(\varepsilon(V_i))^3$

$\Theta(W(f_j)) \leftarrow \frac{posViews - negViews}{norm}$

end for

until convergence

return Θ.

Algorithm 2 2-ESTIMATES

Require: $\mathcal{F}, \mathcal{V}, \mathcal{S}$

Ensure: an estimate of $\varepsilon(V_i)$ for each view, an estimate of $W(f_i)$ for each fact

for all V_i **do** {*Initialization*}

$\Theta(\varepsilon(V_i)) \leftarrow 0$

end for

repeat {*Core of the algorithm*}

for all $f_j \in \mathcal{F}$ **do**

$posViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=T}} 1 - \Theta(\varepsilon(V_i))$

$negViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=F}} \Theta(\varepsilon(V_i))$

$nbViews \leftarrow |\{V_i \in \mathcal{V} \,|\, (V_i, f_j) \in \mathcal{S}\}|$

$\Theta(W(f_j)) \leftarrow \dfrac{posViews + negView}{nbViews}$

end for

for all $V_i \in \mathcal{V}$ **do**

$posFacts \leftarrow \sum_{\substack{f_j \in \mathcal{F} \\ V_i(f_j)=T}} 1 - \Theta(W(f_j))$

$negFacts \leftarrow \sum_{\substack{f_j \in \mathcal{F} \\ V_i(f_j)=F}} \Theta(W(f_j))$

$nbFacts \leftarrow |\{f_j \in \mathcal{F} \,|\, (V_i, f_j) \in \mathcal{S}\}|$

$\Theta(\varepsilon(V_i)) \leftarrow \dfrac{posFacts + negFacts}{nbFacts}$

end for

until convergence

return Θ.

the expectation of $\Theta(W(f_j))$ given the correct set of $\varepsilon(V_i)$'s and the views, is indeed the expectation of $\mathbb{P}(W(f_j) = T)$; similarly for $\Theta(\varepsilon(V_i))$ given the correct set of $W(f_j)$'s and the views. Due to space limitations, this proof is omitted.

Although based on valid estimates, the whole algorithm needs to be tuned to avoid convergence on local optima. Actually, it is relatively easy to see that one of the local optima is a solution where $\forall f_j \in \mathcal{F}$, $\Theta(W(f_j)) = 0.5$, which means that the truth values of the facts are undetermined, and where $\forall V_i \in \mathcal{F}$, $\Theta(W(V_i)) = 0.5$, which means that the views decide randomly. To avoid it, we normalize $\Theta(W(f_i))$ to the closest value in $\{0, 1\}$, which constrains W to map each fact to either T or F, and $\Theta(\varepsilon(V_i))$ to the whole range $[0, 1]$. This is still not satisfactory because the estimation becomes then quite unstable. We fixed the problem using a linear combination between the non-normalized value and the normalized value, as described in Algorithm 3 for the truth values of facts (a similar normalization is applied to the trustworthiness of views). We use a weight λ progressively (and linearly) decreasing from 1 to 0. Experiments show that this technique brings to a good solution in a stable manner. Lastly, a remaining issue with 2-Estimates is that, for one set of views, a given distribution of estimates is always as likely its dual one, where W is replaced by its negation and each error factor $\varepsilon(V_i)$ is replaced by $1 - \varepsilon(V_i)$. We decided to keep the optimistic model, where the average of error factors is assumed to be less than 0.5.

Algorithm 3 NormalizeWFacts

Require: $\mathcal{F}, \Theta, \lambda$
Ensure: a normalized value of Θ
  $maxW \leftarrow \max_{f_j \in \mathcal{F}} \Theta(W(f_j))$
  $minW \leftarrow \min_{f_j \in \mathcal{F}} \Theta(W(f_j))$
  for all $f_j \in \mathcal{F}$ **do**
    $value_1 \leftarrow \frac{\Theta(W(f_j)) - minW}{maxW - minW}$
    $value_2 \leftarrow \text{round}(\Theta(W(f_j)))$
    $\Theta(W(f_j)) \leftarrow \lambda \times value_1 + (1 - \lambda) \times value_2$
  end for
  return Θ.

Though Cosine is a heuristic algorithm that cannot easily be linked to our probabilistic data model, we will show in Section 4 that it is usually more precise and stable than 2-Estimates. In order to overcome the limitations of 2-Estimates, we introduce next an algorithm with an additional series of parameters, namely, the error factor of facts.

Estimation of Three Series of Parameters. Our third algorithm, 3-Estimates, is founded on the full data model described by Equation (1) in Section 2. The algorithm estimates $W(f_j)$ ($f_j \in \mathcal{F}$), $\varepsilon(f_j)$ ($f_j \in \mathcal{F}$) and $\varepsilon(V_i)$ ($V_i \in \mathcal{V}$). We present 3-Estimates in Algorithm 4. As an initialization, we assume that the errors of the views are null and that all the facts are easy to guess. Then we successively estimate one parameter given the other two (and the views). We iterate until convergence with a fixpoint computation very similar to 2-Estimates. Here again, $\Theta(W(f_j))$ is more precisely given a numerical

Algorithm 4 3-ESTIMATES

Require: $\mathcal{F}, \mathcal{V}, \mathcal{S}$

Ensure: an estimate of ε for each view and fact, an estimate of $W(f_i)$ for each fact

for all V_i **do** {*Initialization*}

$\Theta(\varepsilon(V_i)) \leftarrow 0$

end for

for all f_j **do**

$\Theta(\varepsilon(f_j)) \leftarrow 0.1$

end for

repeat {*Core of the algorithm*}

for all $f_j \in \mathcal{F}$ **do**

$posViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=T}} 1 - \Theta(\varepsilon(V_i))\Theta(\varepsilon(f_j))$

$negViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=F}} \Theta(\varepsilon(V_i))\Theta(\varepsilon(f_j))$

$nbViews \leftarrow |\{V_i \in \mathcal{V} \,|\, (V_i, f_j) \in \mathcal{S}\}|$

$\Theta(W(f_j)) \leftarrow \dfrac{posViews + negView}{nbViews}$

end for

for all $f_j \in \mathcal{F}$ **do**

$posViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=T,\, \Theta(\varepsilon(V_i))\neq 0}} \frac{1-\Theta(W(f_j))}{\Theta(\varepsilon(V_i))}$

$negViews \leftarrow \sum_{\substack{V_i \in \mathcal{V} \\ V_i(f_j)=F,\, \Theta(\varepsilon(V_i))\neq 0}} \frac{\Theta(W(f_j))}{\Theta(\varepsilon(V_i))}$

$nbViews \leftarrow |\{V_i \in \mathcal{V} \,|\, (V_i, f_j) \in \mathcal{S},\ \Theta(\varepsilon(V_i)) \neq 0\}|$

$\Theta(\varepsilon(f_j)) \leftarrow \dfrac{posViews + negViews}{nbViews}$

end for

for all $V_i \in \mathcal{V}$ **do**

$posFacts \leftarrow \sum_{\substack{f_j \in \mathcal{F} \\ V_i(f_j)=T,\, \Theta(\varepsilon(f_j))\neq 0}} \frac{1-\Theta(W(f_j))}{\Theta(\varepsilon(f_j))}$

$negFacts \leftarrow \sum_{\substack{f_j \in \mathcal{F} \\ V_i(f_j)=F,\, \Theta(\varepsilon(f_j))\neq 0}} \frac{\Theta(W(f_j))}{\Theta(\varepsilon(f_j))}$

$nbFacts \leftarrow |\{f_j \in \mathcal{F} \,|\, (V_i, f_j) \in \mathcal{S},\ \Theta(\varepsilon(f_j)) \neq 0\}|$

$\Theta(\varepsilon(V_i)) \leftarrow \dfrac{posFacts + negFacts}{nbFacts}$

end for

until convergence

return Θ.

value that is an estimation of $\mathbb{P}(W(f_j) = T)$. Again, as for 2-ESTIMATES, we proved that the three estimators used in 3-ESTIMATES are valid given the other correct sets of parameters.

As was the case with 2-ESTIMATES, we need to apply additionally a normalization procedure for $\varepsilon(f_j)$, similar to those already presented in the previous section. With the ensured condition $\max_{f_j \in \mathcal{F}} \varepsilon(f_j) = 1$, it can be shown that the $\varepsilon(V_i)$'s and $\varepsilon(f_j)$'s are uniquely identified from the set of all products $\varepsilon(V_i)\varepsilon(f_j)$.

Dealing with Functional Dependencies. We explained in Section 2 how a model with both positive and negative assertions is relevant when only positive statements are made, in the presence of functional dependencies. Specifically, given a set of views $\mathcal{V} = \{V_1, \ldots, V_m\}$ with no negative statements, and a set of queries $\mathcal{Q}$ verifying the constraints of Equation (2), we apply the algorithms described in the previous sections to a modified set of views $\mathcal{V}' = \{V_1', \ldots, V_m'\}$, obtained as follows:

$$\begin{cases} \forall f_j \in \mathcal{F}, V_i(f_j) = T \Rightarrow V_i'(f_j) = T \\ \forall f_j \in \mathcal{F}, (V_i(f_j) \text{ undefined} \wedge \exists f \in \mathcal{F}, \\ \qquad\qquad (ref(f) = ref(f_j) \wedge V_i'(f) = T)) \Rightarrow V_i'(f_j) = F \end{cases}$$

In other words, positive statements are kept, and negative statements are added for every unstated facts that refer to a query for which a positive statement has been made. When a view contradicts a functional dependency using more than one positive statement for the same query, we keep all its positive statements, even if they are inconsistent in such a case.

In the presence of functional dependencies, an optional post-filtering step that can be used is to impose that no two facts referring to the same query are predicted true, since we know that such a constraint holds in the real world. In this case, we redefine the estimates of the truth values of facts, after all computations are performed, as:

$$\begin{cases} \Theta(W(f_j)) \leftarrow \min(0.49, E(W(f_j))) & \text{if some other } f \\ \qquad \text{with } ref(f) = ref(f_j) \text{ has a better estimate } \Theta(W(f)) \\ \\ \Theta(W(f_j)) \leftarrow \max(0.51, E(W(f_j))) & \text{otherwise} \end{cases}$$

Only one fact per query can then be estimated true (except when two facts have exactly the same score), and the new estimate of the confidence is corrected to be at least slightly positive for the best fact and at least slightly negative for the other facts. Note that we assume that the views contain the correct answer for each query; this is not always the case in practice. We discuss this issue further in Section 5.

Remark. Even though 2-ESTIMATES and 3-ESTIMATES are based on valid estimates, we do not know whether the fixpoint computation is guaranteed to converge to the best (in mathematical terms) estimates of the dataset and the errors. In a more classical manner, we have been collaborating intensively with a team of statisticians, to study an

Expectation-Maximization (EM) algorithm [6] to the corroboration problem. From our current understanding, the situation is as follows. EM or refinements like ECM suffer from an exponential blowup. The reasons are the discreteness of the decision (true/false) and the non-linearity of the model. A linear model is not well adapted to the situations of interest. We have carried out the formal computation of the expectation of truth values of facts and trustworthiness of sources, with respect to the observations of the model. Our conclusion were that for the system of equations we obtained, classical gradient-like or simulated annealing methods are not really adapted, especially because of the discreteness of the parameters. The best hope would be to use probabilistic estimations based on biased Monte Carlo techniques. A main issue that we found is that of choosing the right bias avoiding the standard risk of overfitting. This work is on-going. In any case, these techniques would probably be more costly that the algorithms we presented and that already produce good results.

4 Experiments

We conducted experiments to test the precision of the algorithms for corroboration presented in the previous section on two kinds of datasets: different instances of a highly configurable synthetic dataset, and a variety of real-world datasets. This variety of datasets demonstrates the improvements we obtain over the Voting and Counting baselines when using our fixpoint algorithms, and in which context these improvements occur.

The algorithms presented in Section 3 and the synthetic data generator discussed in Section 4 have been implemented in Java. All datasets used in this paper, as well as the implementation of the various methods, are freely available from `http://datacorrob.gforge.inria.fr/`.

Measures. We use a number of different quality measures to compare the prediction of the different algorithms. A first measure is the *global precision* of prediction, i.e., the ratio of facts wrongly predicted among all facts. Though interesting to get quickly a general idea of the quality of our methods, this measure does not give a full view of the nature of the differences between methods.

The estimated truth values of facts by most of our methods is given through a score $\Theta(W(f_j))$, which can be seen as the confidence we have in the prediction that the fact is true. To show the differences between methods in this respect, we can plot (in the case of a synthetic dataset where we have this information) this confidence against the *correctness* of the fact, that is, $1 - \varepsilon(f_j) \times \text{avg}_{V_i}\, \varepsilon(V_i)$.

Similarly, the estimated trustworthiness of views $\Theta(\varepsilon(V_i))$ can be compared to the actual *correctness* of the view, computed as $1 - \varepsilon(V_i) \times \text{avg}_{f_j}\, \varepsilon(f_j)$.

Finally, an interesting way to plot the quality of the prediction is through a precision-recall graph, as done when evaluating search engine results in information retrieval [12]. Specifically, we plot the recall-at-k (ratio of true facts among all true facts in the k facts

with the highest estimated truth value) against the precision-at-k (ratio of true facts among the k facts with the highest estimated truth value).

Synthetic Dataset. Our initial experiments were carried out on a synthetic dataset, in order to test our algorithms on a broad scale of situations, with a precise hold on the parameters. We use the following procedure to generate the synthetic dataset, extending the probabilistic data model mentioned in Section 2.

We define two sets $\mathcal{F} = \{f_1 \dots f_n\}$ and $\mathcal{V} = \{V_1 \dots V_m\}$ and we fix the following parameters:

- α, the ratio of true facts among all facts;
- $\varepsilon : \mathcal{F} \cup \mathcal{V} \to [0, 1]$, the error factor for facts and sources;
- $\varphi^+, \mathcal{F} \cup \mathcal{V} \to [0, 1]$ and $\varphi^- : \mathcal{F} \cup \mathcal{V} \to [0, 1]$, the ignorance factors for positive and negative statements, respectively.

We then randomly select for each fact $W(f) = T$ or $W(f) = F$ with probability α and $(1 - \alpha)$ respectively. The view V_i (ignoring some facts and making errors) is obtained as follows:

error For each fact f_j, we randomly set $b(V_i, f_j) = W(f_j)$ with probability $(1-\varepsilon(V_i)\varepsilon(f_j))$ and we make a mistake, i.e., set $b(V_i, f_j) = \neg W(f_j)$, with probability $\varepsilon(V_i)\varepsilon(f_j)$.

ignorance Then we possibly ignore this information, i.e., we set $V_i(f_j)$ to undetermined:

- with probability $\varphi^+(V_i)\varphi^+(f_j)$ if $b(V_i, f_j) = T$.
- with probability $\varphi^-(V_i)\varphi^-(f_j)$ if $b(V_i, f_j) = F$.

Otherwise $V_i(f_j)$ is set to $b(V_i, f_j)$

We ran some experiments on some large synthetic dataset (up to 10,000 facts, 10,000 sources, 5,000,000 statements). As expected, our algorithms are roughly linear in the number of statements. In such conditions, the execution time on a desktop PC is of the order of seconds. The main limitation comes from memory usage, because the current version of our program stores the full set of views in memory. It could easily be adapted to work on disk. Besides, the computations are highly parallelizable. Observe also that, in general, each estimation of parameters for views or facts uses only a small subset of the full set of statements.

We next report on smaller-scale experiments obtained for a synthetic dataset of 1,000 facts and 1,000 sources to analyze the behavior of the algorithms in more details. We use a distribution (see Figure 1) of the probability of errors for facts and sources in three groups for facts (easy, medium and hard) and three for sources (expert, medium, random). Note that the probability of errors for facts is obtained by multiplying the error factor of a fact by the average error factor of sources, and reciprocally for the probability of errors for sources. The average probability of ignorance for a source is of 70%; it ranges between 60 and 80%.

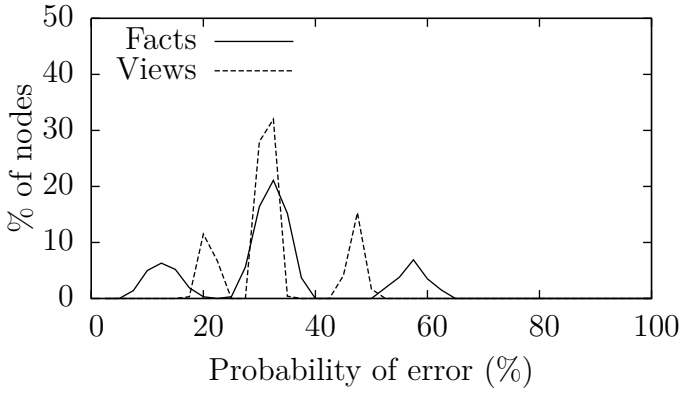

Figure 1: Distribution of errors on synthetic dataset

Table 1: Global precision on the synthetic dataset

	Precision (%) (typical)	Precision (%) (no ignorance)
Voting	84.5	80.2
Counting	84.6	83.3
2-Estimates	88.1	85.1
Cosine	88.2	85.5
3-Estimates	91.5	99.9

The results are shown in Table 1 and Figures 2 to 4. They are fairly typical of the results obtained by varying the parameters. The first data column of Table 1 shows the global precision of the various methods for this dataset. Observe first that the two baselines already perform quite well, with a precision of 85%. Despite this, we can see a significant improvement using 2-Estimates and Cosine, and a larger improvement still with 3-Estimates (observe that the number of errors is divided by two), with a global precision of 91%. The second data column of Table 1 shows what happens when the ignorance factor is set to 0, meaning that each source expresses an opinion on each fact (all other parameters kept unchanged). Many more relevant items of information are present, but this also means much more noise. The performance of the methods does not change much, except for 3-Estimates, which is nearly perfect in this case. In the following, we only consider the case of a non-zero ignorance factor.

Figure 2 shows the confidence on the prediction that the fact is true for the facts according to their correctness on this dataset. For this figure, we randomly sample a subset of the facts to improve readability of the point cloud. The first graph concerns false facts, while the second one is about true facts. On the former, every point in the

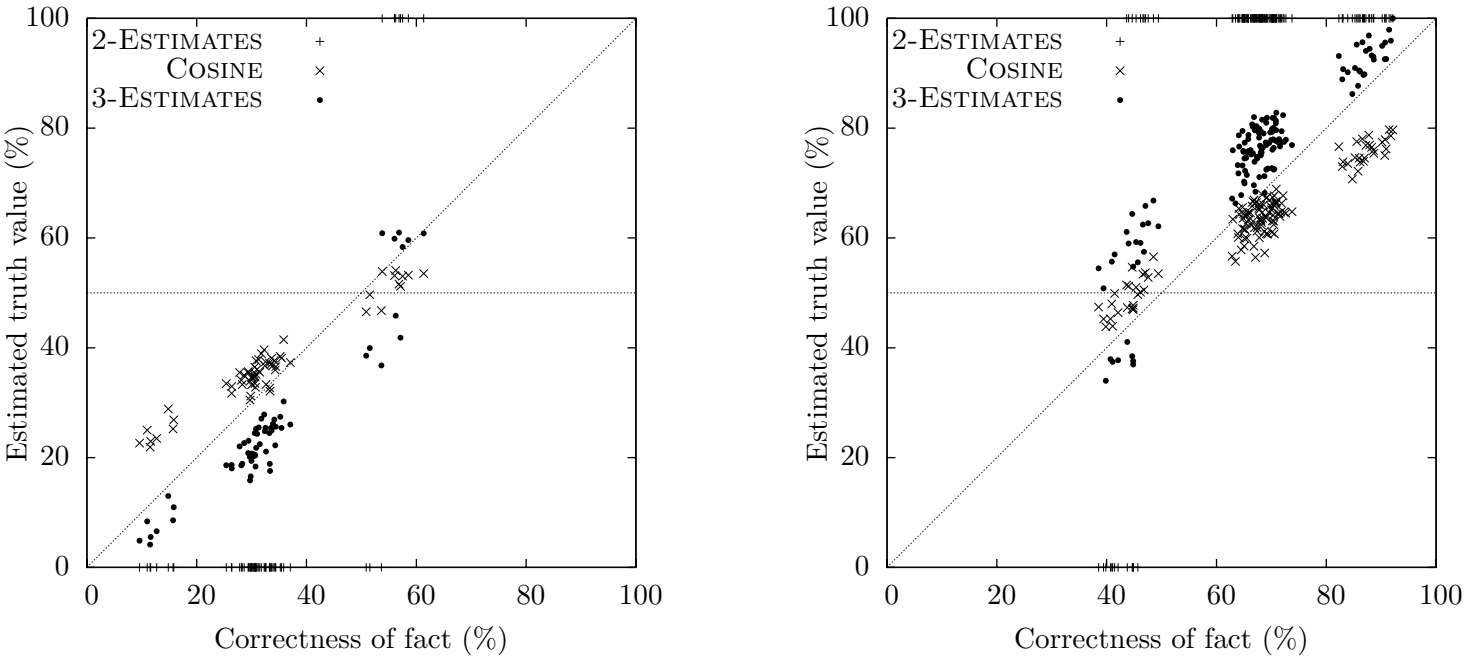

Figure 2: Confidence that the fact is true (left: false facts, right: true facts) with respect to correctness, for the synthetic dataset

upper region of the graph corresponds to a prediction error; on the latter, every point in the lower region does. Thus, the better a method is, the lower the points are in the top graph, and the higher they are in the bottom one. Baseline methods are not plotted on these graphs for readability, but their estimations basically lie on the $y = x$ line: their predictions basically match the correctness, which means that they perform well only if the probability of error for a given fact is lower than 0.5.

We can observe three bags of points from left to right, corresponding respectively to easy, medium or hard false facts in the first graph, and hard, medium or easy true facts in the second one. We clearly see different behaviors for our three non-baseline methods. 2-Estimates is limited to predict 1 or 0, because of its partly *ad hoc* normalization. All the points are consequently on the topmost and bottommost lines of the graph. All the errors occur on the hard facts. Cosine and 3-Estimates perform both reasonably well, but 3-Estimates clearly separates better false facts from true facts. The estimations indeed follow the correctness, since the easy true facts (right on the second graph) get a high probability to be true and the easy false facts (left on the first graph) a low probability to be true, i.e., a high probability to be false. All the errors are once again made on the hard facts, but the estimations of the probabilities to be true are close to 0.5, showing that the methods assign a higher uncertainty to these facts.

Figure 3 shows the estimation of the trustworthiness of views, with respect to their actual correctness. We recognize three bags of points from left to right : random, medium and expert views. As expected, the estimations follow the correctness of views, since the expert views get higher trustworthiness than the random ones, whatever the method.

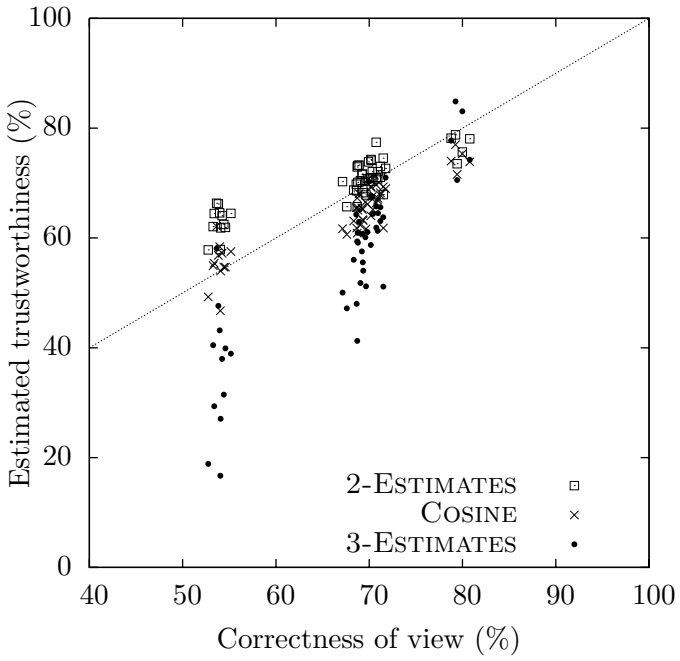

Figure 3: Trustworthiness of views with respect to correctness, for the synthetic dataset

3-Estimates has here an interesting behavior: The trustworthiness of the best views (dots on the right) is boosted further than with the other two methods (squares and crosses), while the trustworthiness of views of low quality (dots on the left) is lowered. This shows that 3-Estimates better assesses the quality of sources.

Finally, Figure 4 shows precision-recall curves for this dataset. These curves may be interpreted in two different ways. The first one is to compare individual points on the curves given a fixed recall/precision ratio, that is, a trade-off between these two conflicting measures (lines $y = \alpha x$). On these lines, the higher the point, the better the method. The other one is to compare the area above the curve: The smaller the area, the better the method. Given these two aspects, this figure confirms the good performance of Cosine and especially 3-Estimates with respect to the baselines. The relatively bad quality of 2-Estimates can be explained by the fact that the estimated truth values given by this method are restricted to 0 and 1, which prevent correctly ordering the best facts.

The previously described experiment is a fairly typical example of the behavior of the various methods on synthetic data, for a large range of values of the parameters. In the wide range of experiments we performed, we observed in particular the following features:

- Voting and Counting give quite good results already, with often some advantage for Counting.
- 2-Estimates generally yields good results (though as said above, it is not good at ordering facts), but is quite unstable and may perform worse than the baselines.

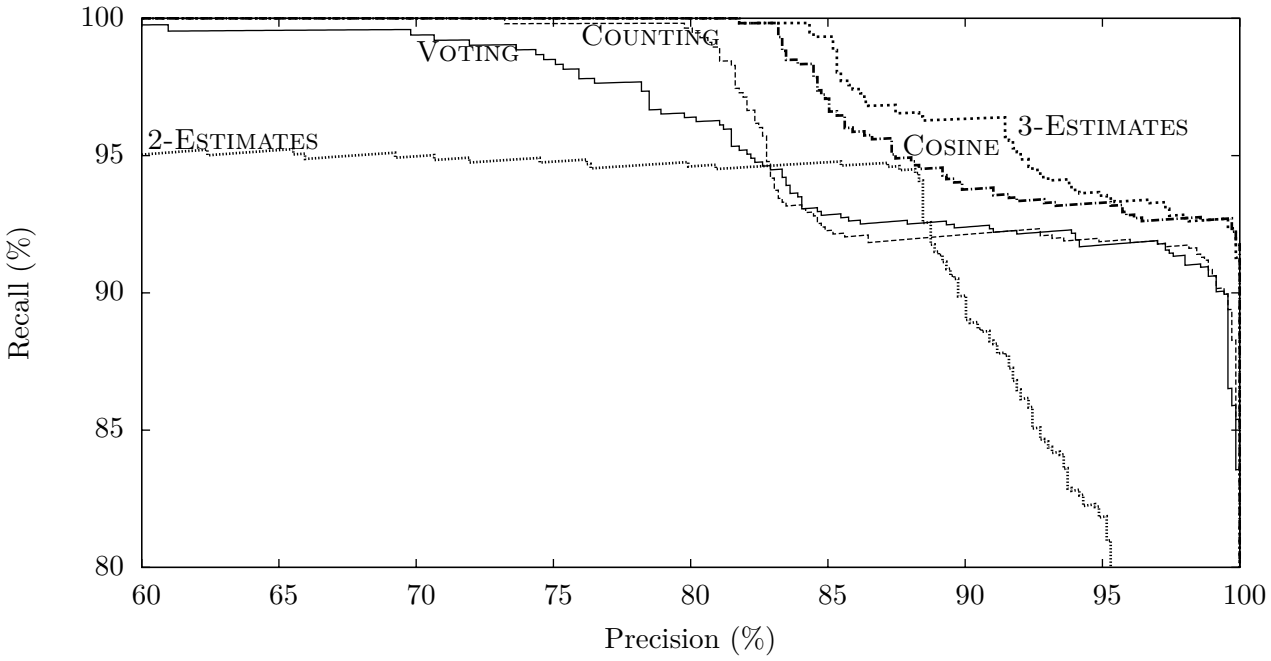

Figure 4: Precision-recall curve for the synthetic dataset

- Cosine is most of the time significantly better than the baselines.
- 3-Estimates consistently yields better results than Cosine.

We next report the results of our algorithms on real-world datasets.

General Knowledge Quiz. This real-world dataset consists of the results of an online general knowledge quiz. This (fairly complicated, and sometimes tricky) quiz is formed of 17 questions with topics ranging from literature to geography and history. For each question, there are between 4 and 14 possible answers, for a total number of 95 facts. There is only one correct answer per question, so we are in the presence of functional dependencies. This quiz was taken 601 times, which corresponds to 601 views. Some of these views are different trials of the same person. After applying the technique for dealing with functional dependencies presented in Section 3, we obtain a full set of 601 views with 37,170 statements. 18% of them are positive statements, and there are (only) $1 - \frac{37{,}170}{601 \times 95} \approx 35\%$ ignored facts (participants to the quiz could choose not to answer some questions).

Table 2 shows the total number of errors obtained by the various methods on this dataset, without and with the post-filtering step described in Section 3. Without post-filtering, all errors are false negatives, i.e., true facts predicted false because the confidence is not high enough. The post-filtering step guarantees that this does not happen. Note that 6 errors after the post-filtering step means only 3 questions with an erroneous answer, since both the false positive and the false negative facts are counted as errors for each of these questions.

Table 2: Number of errors on the first real dataset

	Number of errors (no post-filtering)	Number of errors (with post-filtering)
Voting	11	6
Counting	12	6
2-Estimates	6	6
Cosine	7	6
3-Estimates	9	0

Our three proposed methods systematically perform better or as good as the baselines. Besides, despite the large amount of available information, the baseline methods (as well as Cosine and 2-Estimates) are not able to determine all true facts correctly, whereas 3-Estimates (with post-filtering, which obviously makes the problem easier) is perfect on this dataset, which is a notable achievement.

Other Real-World Datasets. We finally briefly report on experiments conducted on two other real-world datasets, a sixth-grade biology test, and results from Web search engines. On the biology test, the results of the algorithms are very close, with or without functional dependencies. We think that our more complex methods do not perform better than the baselines because the distribution of the accuracy of students is hard to estimate, errors are correlated between students, and there are also correlations between facts. The Web search data aims to illustrate semantic Web applications. The data are a rough extraction of the summaries on the first-answer page of 13 web search engines for 50 keywords query. The algorithms again perform similarly to the baselines. An explanation is that search engines have very similar performance (for this task) and there is again a lot of correlation on the errors.

5 Conclusion

Previous works have considered corroborative evidence to improve trust in query results [3, 11, 7, 17, 18] in a variety of scenarios. Several Question Answering systems, such as [3, 11, 7] consider the frequency of an extracted answer as a measure of answer quality. However, these techniques rely mostly on redundancy of information and do not consider the trust associated with each extraction source to score extracted answers. Recent work has studied the impact of source trust in Web question answering [17, 18]; both projects provide ad hoc mechanisms to assess the trust associated with Web pages, and use this trust information to aggregate answer scores. TruthFinder [18] goes one step further by aiming to identify high-quality sources in addition to true facts. While the goal of TruthFinder is similar to ours, we use a complete probabilistic model for data corroboration that can be used for a variety of scenarios.

Several theoretical work have focused on estimating the probability of an event in

the presence of conflicting information. Osherson and Vardi [15] study the problem of inconsistent outcomes when aggregating logic statements from multiple sources. Their goal is to produce a logically coherent result. Work in subjective logic and trust management [10] consider the issue of trust propagation from one source to another, in a model where the sources are not independent.

Our work on data corroboration shares some interesting ties with work on uncertainty and lineage [2]. Lineage information could help improve the corroboration by giving information on possible correlation between sources. An interesting extension to our model would be to take into account uncertainty, i.e., the confidence each source itself has over the data it reports.

We are also interested in exploring the relationship between data corroboration and data prediction in a model where the true value of future facts is not known. By assigning trust values to sources based on past behavior we can weight predictions or beliefs given by the sources. This is tightly connected to work on prediction markets [16].

As previously mentioned, a goal of this paper was to set the bases for a systematic study of trust-based corroboration of inconsistent views. As we showed, using voting (or counting) for data corroboration works in general rather well. Our methods undoubtedly improve the precision of the results. Nevertheless, the previous discussion clearly points to different directions for further improvements. To conclude this section, we discuss some directions for future research.

First, when considering trust in a social network folksonomy, we may want to give *a priori* more credits to our friends beliefs than to others (but still evaluate how trustworthy they are). Similarly, one may want to specify beliefs in certain site such as the Nasa database for space information. It is easy to introduce bias in the trust of some views. Similarly, one may want to bias the trust we have in some facts. At the limits, we can take advantage of a database of verified facts. It is relatively straightforward to use it to bias trust assessment. Indeed, one could even consider using only these facts as a learning set to fully assess the quality of the sources. Such a standard machine learning technique would often be inappropriate in a Web setting where even if the database of known facts is available, it is very small compared to the size of the Web and does not cover all its facets. The use of a known database also suggests coupling corroboration techniques with knowledge bases (e.g., to avoid confusing a birth date with the date of an election as US president) or an ontology (to corroborate auburn and red hair).

Then, we showed that our technique is very well adapted to find an answer when we know there is exactly one. This should be improved in two directions. First, we should adapt it to the case of multiple answers, e.g., phone numbers. In such cases, we could use some a priori distribution of the number of answers. Also we have to make it robust when we know the question has an answer but this answer is missing from the dataset. In some contexts, forcing the dataset to contain a correct answer to a particular question introduces undesirable effects we would like to avoid.

Our technique is based on assessing the quality of sources in a global manner. However, in the same way that humans are typically experts in specific domains only, sources are specialized. It would be interesting to assess the quality of a source (error and ignorance) in specific domains. This will allow better selecting sources given a specific query. Note

that symmetrically (and less importantly), the same fact may have different truth values in different domains. For instance, "there are red jaguars" is true in the car domain but not in biology.

Another aspect of our technique is that it is assuming independence of the facts and of the sources. This assumption is typically defeated in practice, which may be a cause of a degradation of the quality of results.

Errors can then come from noisy duplication or delayed updates. An interesting direction to consider, especially when dealing with numerical values, is to consider distance between values. For instance, a fact that is stating that the age of a person is 40 is clearly contradicting that the person is 5, but to some extent, corroborating a source that says she is 39. This should be taken into consideration.

Changes in the real world also bring a challenge to corroboration since many sources may believe that an outdated information is correct. Since temporal data (e.g., timestamps of the facts) are rarely available, one could try to analyze the variations of the truth values in time and select a fact with a positive derivative rather than some contradicting fact that is apparently "more true" but has a negative derivative. This may also lead to evaluating a trust in the source that would depend on the time of the fact (if the fact is an event in time): one source (an encyclopedia) may be excellent historically and another one, best adapted to timely information (a newspaper).

Acknowledgement. We want to thank Eric Moulines and Francois Roueff for their help on the statistical model and the Expectation-Maximization method, as well as Yann Ollivier for his feedback.

References

[1] S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *Proc. WWW*, Budapest, Hungary, May 2003.

[2] O. Benjelloun, A. D. Sarma, A. Y. Halevy, M. Theobald, and J. Widom. Databases with uncertainty and lineage. *VLDB Journal*, 17(2), 2008.

[3] E. Brill, S. Dumais, and M. Banko. An analysis of the AskMSR question-answering system. In *Proc. EMNLP*, July 2002.

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

[5] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. A survey of Web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, Oct. 2006.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[7] D. Downey, O. Etzioni, and S. Soderland. A probabilistic model of redundancy in information extraction. In *Proc. IJCAI*, Edinburgh, United Kingdom, July 2005.

[8] S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[9] O. Häggström. *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, United Kingdom, 2002.

[10] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In *Proc. Trust Management*, Pisa, Italy, May 2006.

[11] C. C. T. Kwok, O. Etzioni, and D. S. Weld. Scaling question answering to the Web. In *Proc. WWW*, Hong Kong, China, May 2001.

[12] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[13] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proc. HYPERTEXT*, Odense, Denmark, Aug. 2006.

[14] G. A. Mihaila, L. Raschid, and M.-E. Vidal. Using quality of data metadata for source selection and ranking. In *Proc. WebDB*, Dallas, USA, May 2000.

[15] D. Osherson and M. Y. Vardi. Aggregating disparate estimates of chance. *Games and Economic Behavior*, 56(1):148–173, July 2006.

[16] J. Wolfers and E. Zitzewitz. Prediction markets. *The Journal of Economic Perspectives*, 18(2):107–126, 2004.

[17] M. Wu and A. Marian. Corroborating answers from multiple Web sources. In *Proc. WebDB*, Beijing, China, June 2007.

[18] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the Web. In *Proc. KDD*, San Jose, USA, Aug. 2007.