Judging the Veracity of Claims and Reliability of Sources with Fact-Finders

Jeff Pasternack JPASTER2@UIUC.EDU
Dan Roth DANR@UIUC.EDU
University of Illinois,
Urbana, Illinois

1. Introduction

The Information Age has made publishing, distributing and collecting information easier, resulting in the exponential growth of information available to us. Databases were once ledgers written by hand by a single person; today they can be vast stores of data agglomerated from a myriad of disparate sources. The mass media, formerly limited to newspapers and television programs held to strict journalistic standards, has expanded to include collaborative content such as blogs, wikis, and message boards. Documents covering nearly every topic abound on the Internet, but the authors are often anonymous and the accuracy uncertain.

To cope with this new abundance, we employ information retrieval to suggest documents, and information extraction to tell us what they say, but how can we determine what we should actually *believe*? Not all information sources are equally trustworthy, and simply accepting the majority view often leads to errors: a Google search for "water runs downhill" returns 17.5K documents, while "water runs uphill" yields 116K.

When we consider a collection of data with various authorship, we may view it as a set of information *sources* each making one or more *claims*. Sources often make claims that are contradictory ("Shakespeare was born on April 26th, 1564" and "Shakespeare was born on April 23rd, 1564") and, even in the absence of contradiction, we have no guarantee that the sole presented claim is true. How, then, can we know which claims to believe, and which sources to trust? The typical approach is simple: take a vote and choose the claim made by the largest number of sources. However, this implicitly (and implausibly) assumes that all sources are equally trustworthy and, moreover, ignores the wealth of other claims being made by both these and other sources that could inform our belief in the particular claim at hand. For example, if we can ascertain that John's other claims of birthdays for historic figures were correct, his claim about Shakespeare should (ceteris paribus) carry more weight.

A diverse class of algorithms collectively known as *fact-finders* does just this, using the full network of sources and claims to jointly estimate both the trustworthiness of the sources and the believability of the claims. This is useful not just in judging the assertions made by authors in articles, but also in areas such as sensor networks and crowdsourcing. Crowdsourcing of information–where information is polled from a wider population–can be done via direct voting, the most famous being reCaptcha (Von Ahn, Maurer, McMillen, Abraham, & Blum, 2008), which uses humans to solve difficult OCR problems as a Turing test and accepts the text for a candidate word image once it has accrued enough votes. Similarly, the ESP Game (Von Ahn & Dabbish, 2004) obtains image labelings, presenting the task as a game. In both cases, the annotators are presented with examples

where the labels are already known in an effort to verify their correctness. However, such a coarse approach, where an annotator's trustworthiness is binary, is economically inefficient in systems such as Amazon's Mechanical Turk marketplace where there is a cost for soliciting each vote. (Dawid & Skene, 1979) presents a more sophisticated model where each source has a unique, latent confusion matrix that probabilistically determines the likelihood of each possible response to a question given the truth, with inference performed via Expectation Maximization; this inference can be cast as a fact-finder, though the model is only applicable when the responses (claims) are homogenous across all the questions. Fact-finders have also been directly applied to sensor networks, such as Apollo (Le, Pasternack, Ahmadi, Gupta, Sun, Abdelzaher, Han, Roth, Szymanski, & Adali, 2011) which received its sensor data via Twitter and inferred the effective speed limit of a road or the true statement about a topic in the varied domains of driving and breaking news, respectively, using an abstract distance metric to cluster similar observations prior to running a standard fact-finder. Such a system is especially invaluable in an emergency, when reports from unreliable first-person sources relayed through social media must be collated and analyzed to discover the moment-to-moment reality.

The basis of each fact-finding algorithm is a two layer bipartite graph, consisting of a layer of information sources (such as the New York Times, Twitter user @truthteller, or your friend Bob) with edges connecting them to another layer that consists of the claims they make (Shakespeare's birthday, the height of Everest, etc.) Fact-finders then iteratively calculate the belief in each claim as a function of the trustworthiness of the sources asserting it and the trustworthiness of a source as a function of the belief in the claims it makes. After the stopping criteria is met, the claim taken to be true is simply the one with the highest belief score versus contradicting, mutually exclusive claims (e.g. water runs either uphill or downhill, but not both).

There have been many algorithms for discovering the truth that fit this pattern (e.g. (Yin, Yu, & Han, 2008; Pasternack & Roth, 2010; Galland, Abiteboul, Marian, & Senellart, 2010)), while others, though intended to solve different problems, have been adapted for this task with minor changes (e.g. Hubs and Authorities (Kleinberg, 1999), originally used for determining the importance of documents). Fact-finders generally have the advantages of being both highly tractable (typically linear time in the number of claims and sources) and easy to describe and implement. Combined with the multitude of extant methods, tuning to find a performant fact-finder on a given domain is straightforward; this is fortunate since, while clear update rules describe how belief and trustworthiness is determined in each particular iteration, fact-finders are (usually) opaque on the whole—after running to completion there is no concise, precise explanation why a particular claim is to be believed over its alternatives. A few fact-finders, such as SimpleLCA (Pasternack & Roth, 2013), are transparent, generative models with explicit semantics, although these are the minority, and, in practice, several opaque fact-finders have, perhaps surprisingly, been shown to work well across a broad range of problems, even without tuning.

Still, since they consider only a bipartite graph of sources and claims, traditional fact-finders do ignore the wealth of background and auxiliary knowledge that are frequently available, including common-sense and specific knowledge about claims, attributes of the sources, and the quality of the information extraction. In this chapter we also present a framework that addresses this by both generalizing the fact-finding algorithms to admit more informative inputs and enforcing declarative constraints over the iterative process (Pasternack & Roth, 2010, 2011a). Each of these two (orthogonal) additions can yield substantially more accurate trust decisions than standard fact-finders on real-world data, both individually and in conjunction. Furthermore, by modeling the user's prior

knowledge and beliefs, they allow for *subjective*, rather than objective, truth, often with dramatic practical benefit; it is thus possible to, for example, model the truth of statements such as ”travel to Great Britain does not require a visa” relative to the nationality of the user.

2. Related Work

An overview of computational trust can be found in (Artz & Gil, 2007) and (Sabater & Sierra, 2005); we next discuss in more detail the foundations of trust for a broader contextualization as well as more concrete work relating to generalized fact-finding and similar problems.

2.1 Foundations of Trust

(Marsh, 1994) observes that trust can be global (e.g. eBay's feedback scores), personal (each person has their own trust values), or situational (personal and specific to a context). Fact-finding algorithms are based on global trust, while our framework establishes personal trust by exploiting the user's individual prior knowledge.

Probabilistic logics have been explored as an alternate method of reasoning about trust. (Manchala, 1998) utilizes fuzzy logic (Novak, Perfilieva, & Mockof, 1999), an extension of propositional logic permitting [0,1] belief over propositions. (Yu & Singh, 2003) employs Dempster-Shafer theory (Shafer, 1976), with belief triples (mass, belief, and plausibility) over *sets* of possibilities to permit the modeling of ignorance, while (Josang, Marsh, & Pope, 2006) uses the related subjective logic (Josang, 1997). While our belief in a claim is decidedly Bayesian (corresponding to the probability that the claim is true), "unknowns" (discussed later) allow us to reason about ignorance as subjective logic and Dempster-Shafer do, but with less complexity.

Of course, a user's perception of trust cannot be captured entirely by such abstract formalisms. A large amount of work from fields such as human-computer interaction, economics, psychology and other social sciences looks at how trust is created and used by people, which has direct implications in developing an automated system that can take into account the full breadth of relevant information to calculate trust. Much of this research as it pertains to human-computer interaction specifically has been done by Fogg's Persuasive Technologies Lab (Tseng & Fogg, 1999; Bailey, Gurak, & Konstan, 2001; Fogg, Marshall, Kameda, Solomon, Rangnekar, Boyd, & Brown, 2001a; Fogg & Tseng, 1999; Fogg, Swani, Treinen, Marshall, Laraki, Osipovich, Varma, Fang, Paul, Rangnekar, & Others, 2001b), demonstrating that, besides factors traditionally considered by trust systems such as recommendations and past reliability, humans are superficial, e.g. placing more faith in a site with a ".org" domain name or a modern, sophisticated design. We note that, though stereotypes, these factors also make useful priors, as ".org" websites are often non-profits that may be more truthful than a website trying to sell a product, and a sophisticated design implies a high degree of investment by the website creator who presumably have much more to lose if he violates his visitors' trust than someone whose website appears to have been prepared in less than an hour. (Gil & Artz, 2007) similarly identifies 19 factors that influence trust in the context of the semantic web, including user expertise (prior knowledge), the popularity of a resource (the wisdom of crowds), appearance (superficial features) and recency (on the basis that more recent information is more likely to be up-to-date and correct). These hitherto exotic factors in trust decisions help motivate our trust framework's ability to incorporate a broad spectrum of prior knowledge, thus permitting them to be leveraged in the context of fact-finding.

2.2 Consistency in Information Extraction

Some information extraction systems have confronted the problem of "noise" in their extractions, using various mechanisms to increase the consistency and accuracy of the results. Here the veracity of the documents is not questioned (explicitly or at all), but rather errors (presumably made by the information extraction itself) are to be corrected and the content of the documents is to be accurately parsed. Such systems can be divided into those that are "local", viewing each document independently, and those who attempt to maintain a global consistency across documents.

2.2.1 Local Consistency

Constrained conditional models (CCMs) (Chang, Ratinov, & Roth, 2012) enforce (soft) constraints across the extracted fields. For example, one constraint might specify that an apartment listing contains only a single price. These constraints do not cross documents and therefore cannot leverage redundancy, as they might to identify and resolve inconsistency in the reported prices in two listings for the same apartment, but they can enforce rules within a document document, e.g. the prior knowledge that a one-bedroom apartment in San Francisco costs at least $1000.

2.2.2 Global Consistency

In (Poon & Domingos, 2007), a Markov Logic Network joint model for both extracting citation fields and identifying coreferential citations is proposed; here, consistency in the information extraction is enforced as a consequence of the (weighted) first-order logic rules; e.g. if a trigram seems likely to start a field in one citation, it probably also starts a field in another citation held to be coreferential. We can consider such constraints to be global is the sense that they cross documents (citations), but such cross-document information could be more accurately regarded as non-binding hints rather than constraints, and there is no attempt to identify, e.g., the true authors of a given work or its canonical title.

Never Ending Language Learning (NELL) (Carlson, Betteridge, Kisiel, Settles, Hruschka, & Mitchell, 2010) is a system for large-scale, ongoing open information extraction, combining a number of subcomponents that infer categories and relations for noun phrases. NELL also induces rules to, together with previously inferences, broaden and (hopefully) improve future extractions (in practice, without human intervention, accuracy declines with time, as might be expected since the "easiest" facts tend to be extracted first). Consistency is achieved by (ontology-based) mutual exclusion and relation type- checking; this allows NELL to know, for example, that an entity cannot be both an actor and a city. Once there exists a preponderance of evidence for a particular candidate fact (i.e. the subcomponents collectively agree or one subcomponent strongly believes *and* the candidate does not conflict with already-believed knowledge) it is permanently added to the knowledge base: NELL does not (yet) revisit previous decisions.

SOFIE (Suchanek, Sozio, & Weikum, 2009) enforces consistency with rules which are grounded as propositional logic, and then approximately maximize the (weighted) number of satisfied clauses. A rule might specify, for example, that one must die within 100 years of birth (note that, as the clauses are weighted, violations are penalized but not prohibited). This is similar to the constraints used by constrained conditional models except that they enforce the consistency of knowledge across documents.

2.3 Source Dependence

AccuVote (Dong, Berti-equille, & Srivastava, 2009a; Dong, Berti-Equille, & Srivastava, 2009b) is a fact-finding variant that is particularly interesting as it attempts to compute source dependence (where one source copies another) and gives greater credence to more "independent" sources. The idea is that independent confirmation, where multiple sources reach the same conclusion on their own, is much more compelling than a set of dependent sources parroting each other's assertions. Ideally in such cases the copying source would verify the assertions of the copied source before repeating them, but this is often not done in practice. On the benign end of the spectrum, a blog might repost a news story while presuming its veracity with no further consideration. More maliciously, a concerted action across multiple sources is sometimes used to spread misinformation online, e.g. spreading takeover rumors to drive up the price of a stock.

2.3.1 Comparison to Credibility Analysis

Credibility analysis models the trustworthiness of each source, whereas information extraction consistency checking effectively treats every source as equally reliable; even if a document overwhelmingly consists of "facts" already disbelieved by the IE system, it will still give equal credence to any others therein. The goals are also different: information extraction seeks to determine what things a document *says*, whereas credibility analysis seeks to determine whether those things should be *believed*. Even in systems that enforce world consistency such as SOFIE, there is a presumption that the documents are truthful and that any violation of the constraints is due to a shortcoming of the model (or the constraints themselves). If two sets of documents simply disagree on a claim, however, the results will be similar to voting (whichever option is asserted most will be believed, or, in the case of NELL, whichever is asserted first!).

2.4 Comparison to Other Trust Mechanisms

Reputation-based systems and trust metrics determine trust among peers, with each peer providing recommendations (or disapproval) for other peers; this may be implicit as in PageRank (Brin & Page, 1998), where the "recommendation" is in the form of a link, or explicitly, as in Advogato (Levien, 2008). Reputation algorithms thus tend to focus on the transitivity of these recommendations, whereas fact-finders specify the relationship between sources and claims and derive their graph structure from corpora. Wikitrust (Adler & de Alfaro, 2007; Adler, Chatterjee, Alfaro, Faella, Pye, & Raman, 2008) and (Zeng, Alhossaini, Ding, Fikes, & McGuinness, 2006) are similarly "content-based" and corpus-driven, but these approaches are specialized to wikis and lack the broader applicability of fact-finders. Lastly, data fusion systems address conflicting claims within a database (e.g. (Bertino, Dai, Lim, & Lin, 2008) and (Dai, Lin, Bertino, & Kantarcioglu, 2008b, 2008a)) by examining the provenance (chain of custody) of the data–data that has passed through the hands of trusted agents is more believable than data that has been filtered through one or more less trustworthy agents; in most domains, however, we only know the immediate source of a claim (who said it to us) and not the full provenance, limiting the utility of these approaches.

3. Fact-Finding

Before we discuss generalized fact-finding, we'll first formalize the standard fact-finding model. We have a set of sources, S, a set of claims C, the claims C_s asserted by each source $s \in S$,

Table 1: Symbols and their descriptions

Symbol	Meaning
s	An information source
c	A claim
S	The set of all sources
C_s	The set of claims asserted by $s \in S$
S_c	The set of sources asserting $c \in C$
$C = \bigcup_{s \in S} C_s$	The set of all claims
$M_c \subseteq C$	The *mutual exclusion set* of c
$T^i(s)$	Trustworthiness of source s in iteration i
$B^i(c)$	Belief in claim c in iteration i

and the set of sources S_c asserting each claim $c \in C$. The sources and claims can be viewed as a bipartite graph, where an edge exists between each s and c if $c \in C_s$. In each iteration i, we estimate the trustworthiness $T^i(s)$ of each source s given $B^{i-1}(C_s)$, the belief in the claims it asserts, and estimates the belief $B^i(c)$ in each claim c given $T^i(S_c)$, the trustworthiness of the sources asserting it, continuing until convergence or a stop condition. Note that "trustworthiness" and "belief" as used within a fact-finding algorithm typically do not have well-defined semantics (e.g. they are not $[0, 1]$ probabilities). An initial set of beliefs, $B^0(C)$, serve as priors for each algorithm; these are detailed in the next section. Notice that *any* fact-finder can be specified with just three things: a trustworthiness function $T(s)$, a belief function $B(c)$, and the set of priors $B^0(C)$.

The *mutual exclusion set* $M_c \subseteq C$ is the set of claims that are mutually exclusive to one another (e.g. putative Obama birthplaces) to which c belongs; if c is not mutually exclusive to any other claims, $M_c = \{c\}$. For each mutual exclusion set M containing true claim $\bar{c}$, the goal of the fact-finder is to ensure $\text{argmax}_{c \in M_c} B^f(c) = \bar{c}$ at the final iteration f; the reported accuracies in the empirical results presented later are thus the percentage of mutual exclusion sets we correctly predict over, discounting cases where this is trivial ($|M| = 1$) or no correct claim is present ($\bar{c} \notin M$).

3.1 Priors

Except for the 3-Estimates algorithm (where the priors are dictated by the algorithm itself), every fact-finder requires priors for $B^0(C)$. We'll draw from these three: $B^0_{voted}(c) = |S_c| / \sum_{d \in M_c} |S_d|$, $B^0_{uniform}(c) = 1/|M_c|$, and $B^0_{fixed}(c) = 0.5$.

3.2 Fact-Finding Algorithms

The fact-finding algorithms we will specifically consider in the remainder of this chapter are Sums (Hubs and Authorities), TruthFinder, 3-Estimates, Average-Log, Investment, and PooledInvestment.

3.2.1 Sums (Hubs and Authorities)

Hubs and Authorities (Kleinberg, 1999) gives each page a hub score and an authority score, where its hub score is the sum of the authority of linked pages and its authority is the sum of the hub scores of pages linking to it. This is adapted to fact-finding by viewing sources as hubs (with 0 authority)

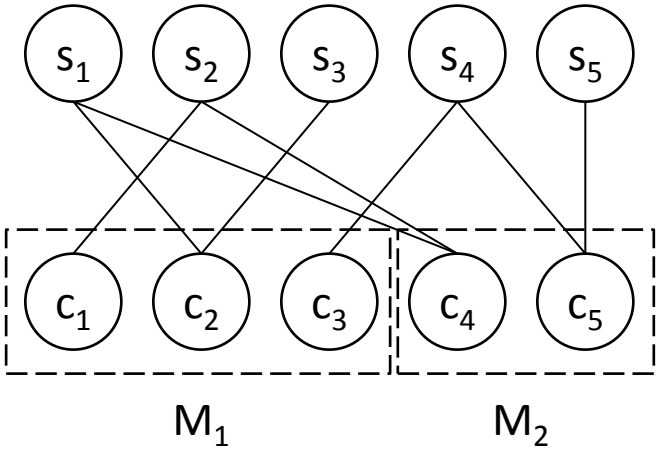

Figure 1: A small fact-finding problem with five sources, and five claims in two mutual exclusion sets, M_1 and M_2. Edges link each source to the claims it asserts. The fact-finding algorithm alternates between updating the trustworthiness of each source given the belief in the claims it asserts and the belief in each claim given the trustworthiness of the sources asserting it.

and claims as authorities (with 0 hub score):

$$T^i(s) = \sum_{c \in C_s} B^{i-1}(c) \qquad\qquad B^i(c) = \sum_{s \in S_c} T^i(s)$$

We normalize to prevent $T^i(s)$ and $B^i(c)$ from growing unbounded (dividing by $\max_s T^i(s)$ and $\max_c B^i(c)$, respectively), a technique also used with the Investment and Average·Log algorithms; this avoids numerical overflow. B^0_{fixed} priors are used.

3.2.2 AVERAGE·LOG

Computing $T(s)$ as an average of belief in its claims overestimates the trustworthiness of a source with relatively few claims; certainly a source with 90% accuracy over a hundred examples is more trustworthy than a source with 90% accuracy over ten. However, summing the belief in claims allows a source with 10% accuracy to obtain a high trustworthiness score by simply making many claims. Average·Log (Pasternack & Roth, 2010) attempts a compromise, while still using Sums' B^i update rule and B^0_{fixed} priors.

$$T^i(s) = \log |C_s| \cdot \frac{\sum_{c \in C_s} B^{i-1}(c)}{|C_s|}$$

3.2.3 INVESTMENT

In the Investment algorithm (Pasternack & Roth, 2010), sources "invest" their trustworthiness uniformly among their claims. The belief in each claim then grows according to a non-linear function $\mathcal{G}$, and a source's trustworthiness is calculated as the sum of the beliefs in their claims, weighted by the proportion of trust previously contributed to each (relative to the other investors). Since claims with higher-trust sources get higher belief, these claims become relatively more believed and their

sources become more trusted. The experimental results we will present use $\mathcal{G}(x) = x^g$ with $g = 1.2$, together with B^0_{voted} priors.

$$T^i(s) = \sum_{c \in C_s} B^{i-1}(c) \cdot \frac{T^{i-1}(s)}{|C_s| \cdot \sum_{r \in S_c} \frac{T^{i-1}(r)}{|C_r|}}$$
$$B^i(c) = \mathcal{G}\left(\sum_{s \in S_c} \frac{T^i(s)}{|C_s|}\right)$$

3.2.4 POOLEDINVESTMENT

Like Investment, in PooledInvestment (Pasternack & Roth, 2010) sources uniformly invest their trustworthiness in claims and obtain corresponding returns, so $T^i(s)$ remains the same, but now after the belief in the claims of mutual exclusion set M have grown according to $\mathcal{G}$, they are linearly scaled such that the total belief of the claims in M remains the same as it was before applying $\mathcal{G}(x) = x^g$, with $g = 1.4$ and $B^0_{uniform}$ priors used in the experiments. Given $H^i(c) = \sum_{s \in S_c} \frac{T^i(s)}{|C_s|}$, we have:

$$B^i(c) = H^i(c) \cdot \frac{\mathcal{G}(H^i(c))}{\sum_{d \in M_c} \mathcal{G}(H^i(d))}$$

3.2.5 TRUTHFINDER

TruthFinder (Yin et al., 2008) is pseudoprobabilistic: the basic version of the algorithm below calculates the "probability" of a claim by assuming that each source's trustworthiness is the probability of it being correct and then averages claim beliefs to obtain trustworthiness scores. We also used the "full", more complex TruthFinder, omitted here for brevity. $B^0_{uniform}$ priors are used for both.

$$T^i(s) = \frac{\sum_{c \in C_s} B^{i-1}(c)}{|C_s|}$$
$$B^i(c) = 1 - \prod_{s \in S_c} \left(1 - T^i(s)\right)$$

3.2.6 3-ESTIMATES

3-Estimates (Galland et al., 2010), also omitted for brevity, differs from the other fact-finders by adding a third set of parameters to capture the "difficulty" of a claim, such that correctly asserting a difficult claim confers more trustworthiness than asserting an easy one; knowing the exact population of a city is harder than knowing the population of Mars (presumably 0) and we should not trust a source merely because they provide what is already common knowledge.

4. Generalized Constrained Fact-Finding

If one author claims that Mumbai is the largest city in the world, and another claims that it is Seoul, who should we believe? One or both authors could be intentionally lying, honestly mistaken or, alternatively, of different viewpoints of what constitutes a "city" (the city proper? The metropolitan area?) Even here, truth is not objective: there may be many valid definitions of "city", but we should

believe the claim that accords with our *user's* viewpoint. Rarely is the user's or author's perspective explicit (e.g. an author will not fully elaborate "the largest city by metropolitan area bounded by...") but it is often implied (e.g. a user's statement that "I already know the population of city A is X, city B is Y..." implies that his definition of a city accords with these figures). A standard fact-finder, however, knows nothing about the user's prior belief and viewpoint and presumes instead to find the (frequently non-existent) objective truth that holds for everyone.

Of course, domain knowledge is not limited to specific statements such as "Los Angeles is more populous than Wichita", but also includes common-sense reasoning such as "cities usually grow over time". We may also know something about the information sources (e.g. "John works for the U.S. census"), the source's own certainty in their claim ("I am 60% certain that..."), the information extraction system's certainty in the claim ("it is 70% certain that John claimed he was 60% certain that..."), and the similarity between mutually exclusive claims (if John thinks the population of a city is 1,000, he disagrees less with a claim that the population is 1,200 than a claim that it is 2,000).

Motivated by this, we can synthesize a framework that both *generalizes* fact-finding to incorporate source and similarity information (Pasternack & Roth, 2011b) and *constrains* it to enforce the user's common-sense and specific declarative knowledge about the claims (Pasternack & Roth, 2010). As these aspects are orthogonal and complementary, we introduce them separately, experimentally demonstrating their individual contribution to performance before combining them into a single system able to leverage a very broad range of relevant information for making trust decision while still building upon the diversity and tractability of existing state-of-the-art fact-finding algorithms.

5. Generalized Fact-Finding

The key technical idea behind generalized fact-finding is that we can quite elegantly encode the relevant background knowledge and contextual detail by replacing the bipartite graph of standard fact-finders with a new weighted k-partite graph, transitioning from binary assertions to weighted ones ("source s claims c with weight x"), rewriting the fact-finders to take advantage of these weights (discussed next), and adding additional "layers" of nodes to the graph to represent source groups and attributes (as discussed later).

5.1 Rewriting Fact-Finders for Assertion Weights

Generalized fact-finders use weighted assertions, where each source s asserts a claim c with weight $\omega(s,c) = [0,1]$. A surprisingly large amount of information, including the uncertainty of the information extractor, the uncertainty of the source itself, the similarity of claims, and the group membership and attributes of sources, can be encoded into this weight (we will present the details and formula for calculating $\omega(s,c)$ in the next section). After calculating the weights $\omega(s,c)$ for all $s \in S$ and $c \in C$, we need to rewrite each fact-finder's $T(s)$, $B(c)$ and $B^0(c)$ functions to use these weights in the generalized fact-finding process by qualifying previously "whole" assertions as "partial", weighted assertions. We start by redefining S_c as $\{s : s \in S, \omega(s,c) > 0\}$, and C_s as $\{c : c \in C, \omega(s,c) > 0\}$. The basic rewriting rules are:

- Replace $|S_c|$ with $\sum_{s \in S_c} \omega(s,c)$
- Replace $|C_s|$ with $\sum_{c \in C_s} \omega(s,c)$
- In $T^i(s)$, replace $B^{i-1}(c)$ with $\omega(s,c)B^{i-1}(c)$

- In $B^i(c)$, replace $T^i(s)$ with $\omega(s,c)T^i(s)$

These rules suffice for all the linear fact-finders we encountered; one, TruthFinder, is instead log-linear, so an exponent rather than a coefficient is applied, but such exceptions are straightforward.

5.1.1 GENERALIZED SUMS (HUBS AND AUTHORITIES)

$$T^i(s) = \sum_{c \in C_s} \omega(s,c) B^{i-1}(c) \qquad\qquad B^i(c) = \sum_{s \in S_c} \omega(s,c) T^i(s)$$

5.1.2 GENERALIZED AVERAGE·LOG

Average·Log employs the same belief function as Sums, so we list only the trustworthiness function:

$$T^i(s) = \log\left(\sum_{c \in C_s} \omega(s,c)\right) \cdot \frac{\sum_{c \in C_s} \omega(s,c) B^{i-1}(c)}{\sum_{c \in C_s} \omega(s,c)}$$

5.1.3 GENERALIZED INVESTMENT

The Investment algorithm requires sources to "invest" their trust uniformly in their claims; we rewrite this such that these investments are weighted by ω.

$$T^i(s) = \sum_{c \in C_s} \frac{\omega(s,c) B^{i-1}(c) T^{i-1}(s)}{\sum_{c \in C_s} \omega(s,c) \cdot \sum_{r \in S_c} \frac{\omega(r,c) T^{i-1}(r)}{\sum_{b \in C_r} \omega(r,b)}}$$

$$B^i(c) = \mathcal{G}\left(\sum_{s \in S_c} \frac{\omega(s,c) T^i(s)}{\sum_{c \in C_s} \omega(s,c)}\right)$$

5.1.4 GENERALIZED POOLEDINVESTMENT

PooledInvestment utilizes the same trustworthiness function as Investment, and instead alters the belief function, which we generalize below.

$$H^i(c) = \sum_{s \in S_c} \frac{\omega(s,c) T^i(s)}{\sum_{c \in C_s} \omega(s,c)}$$

$$B^i(c) = H^i(c) \cdot \frac{\mathcal{G}(H^i(c))}{\sum_{d \in M_c} \mathcal{G}(H^i(d))}$$

5.1.5 GENERALIZED TRUTHFINDER

TruthFinder (Yin et al., 2008) has both a "simple" and "complete" version, with the latter making a number of adjustments to the former. We specify only the simple version below, as the modifications to the complete variant are similar. Both models calculate claim belief non-linearly, and in either case we have the option of using logarithms to obtain a log-linear function. This is what we do in practice, since it avoids underflow in the floating-point variables; for clarity, however, we present the "multiplicative" version below. Note that using $\omega(s,c)$ as an exponent here is equivalent to its

use as a coefficient in the log-linear function.

$$T^i(s) = \frac{\sum_{c \in C_s} \omega(s,c) B^{i-1}(c)}{\sum_{c \in C_s} \omega(s,c)}$$
$$B^i(c) = 1 - \prod_{s \in S_c} \left(1 - T^i(s)\right)^{\omega(s,c)}$$

5.1.6 GENERALIZED 3-ESTIMATES

3-Estimates (Galland et al., 2010) incorporates an additional set of parameters to model the "hardness" of each claim (referred to as $\varepsilon(\mathcal{F})$) that can be incorporated into the B and T functions to fit our common model. We omit the full algorithm here for brevity, but generalizing it is quite straightforward–when calculating a summation over sources for a given claim or a summation over claims for a given source, we simply weight each element of the sum by the relevant assertion weight between the particular source and claim in question.

5.2 Encoding Information in Weighted Assertions

As previously mentioned, weighted assertions allow us to encode a variety of factors into the model:

- Uncertainty in information extraction: we have a $[0, 1]$ probability that source s asserted claim c.
- Uncertainty of the source: a source may qualify his assertion ("I'm 90% certain that...")
- Similarity between claims: a source asserting one claim also implicitly asserts (to a degree) similar claims.
- Group membership: the other members of the groups to which a source belongs implicitly support (to a degree) his claims.

We separately calculate ω_u for uncertainty in information in extraction, ω_p for uncertainty expressed by the source, ω_σ for the source's implicit assertion of similar claims, and ω_g for a source's implicit assertion of claims made by the other members of the groups to which he belongs. These are orthogonal, allowing us to calculate the final assertion weight $\omega(s, c)$ as: $\omega_u(s, c) \times \omega_p(s, c) + \omega_\sigma(s, c) + \omega_g(s, c)$. Here, $\omega_u(s, c) \times \omega_p(s, c)$ can be seen as our expectation of the $[0, 1]$ belief the source s has in claim c given the possibility of an error in information extraction, while $\omega_\sigma(s, c)$ and $\omega_g(s, c)$ redistribute weight based on claim similarity and source group membership, respectively.

5.2.1 UNCERTAINTY IN INFORMATION EXTRACTION

The information extractor may be uncertain whether an assertion occurs in a document due to intrinsic ambiguities in the document or error from the information extractor itself (e.g. an optical character recognition mistake, an unknown verb, etc.); in either case, the weight is given by the probability $\omega_u(s, c) = P(s \to c)$.

5.2.2 UNCERTAINTY OF THE SOURCE

Alternatively, the source himself may be unsure. This may be specific ("I am 60% certain that Obama was born in...") or vague ("I am pretty sure that..."); in the latter case, we assume that the information extractor will assign a numerical certainty for us, so that in either event we have $\omega_p(s,c) = P_s(c)$, where $P_s(c)$ is the estimate provided by source s of the probability of claim c.

5.2.3 SIMILARITY BETWEEN CLAIMS

Oftentimes a meaningful similarity function exists among the claims in a mutual exclusion set. For example, when comparing two possible birthdays for Obama, we can calculate their similarity as the inverse of the time between them, e.g. $|days(date1) - days(date2)|^{-1}$ (where $days$ measures the number of days relative to an arbitrary reference date). A source claiming $date1$ then also claims $date2$ with a weight proportional to this degree of similarity, the idea being that while $date2$ is not what he claimed, he will prefer it over other dates that are even *more* dissimilar. Given a $[0,1]$ similarity function $\sigma(c_1, c_2)$, we can calculate:

$$\omega_\sigma(s,c) = \sum_{d \in M_c, d \neq c} \omega_u(s,d)\omega_p(s,d)\sigma(c,d)$$

Notice that a self-consistent source will not assert multiple claims in mutual exclusion set M with $\sum_{c \in M} \omega_u(s,c)\omega_p(s,c) > 1$, so the addition of $\omega_\sigma(s,c)$ to our formula for $\omega(s,c)$ will never result in $\omega(s,c) > 1$; it is possible, however, that $\sum_{c \in M} \omega(s,c) > 1$ for a given source s. One way to avoid this is to redistribute (smooth) weight rather than add it; we introduce the parameter α to control the degree of smoothing and obtain:

$$\omega_\sigma^\alpha(s,c) = \sum_{d \in M_c, d \neq c} \left(\frac{\alpha\omega_u(s,d)\omega_p(s,d)\sigma(c,d)}{\sum_{e \in M_d, e \neq d} \sigma(d,e)} \right) - \alpha\omega_u(s,c)\omega_p(s,c)$$

This function ensures that only a portion α of the source's expected belief in the claim, $\omega_u(s,c)\omega_p(s,c)$, is redistributed among other claims in M_c (proportional to their similarity with c), at a cost of $\alpha\omega_u(s,c)\omega_p(s,c)$.

(Yin et al., 2008) previously used a form of additive similarity as "Implication" functions in TruthFinder; however, the formalization presented here generalizes this idea and allows us to apply it to other fact-finders as well.

5.2.4 GROUP MEMBERSHIP VIA WEIGHTED ASSERTIONS

Oftentimes a source belongs to one or more groups; for example, a journalist may be a member of professional associations and an employee of one or more publishers. Our assumption is that these groups are *meaningful*, that is, sources belonging to the same group tend to have similar degrees of trustworthiness. A prestigious, well-known group (e.g. the group of administrators in Wikipedia) will presumably have more trustworthy members (in general) than a discredited group (e.g. the group of blocked Wikipedia editors). The approach discussed in this section encodes these groups using ω_g; a more flexible approach, discussed later, alternatively adds additional "layers" of groups and attributes to create a k-partite rather than bipartite graph.

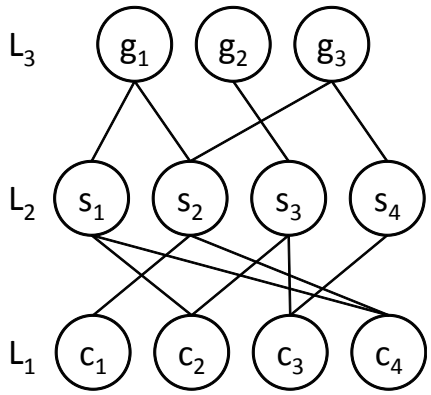

Figure 2: A fact-finding problem with a single group layer. Edges between sources and groups denote membership.

Let G_s be the set of groups to which a source s belongs. If a source s and source u are both members of the same group g, we interpret this as an implicit assertion by u in C_s, and by s in C_u—that is, group members mutually assert each others' claims to a degree. We use a redistribution parameter β such that the original weight of a member's assertion is split between the member (proportional to $1-\beta$) and the other members of the groups to which he belongs (proportional to β). This gives us:

$$\omega_g^\beta(s,c) = \beta \sum_{g \in G_s} \sum_{u \in g} \frac{\omega_u(u,c)\omega_p(u,c) + \omega_\sigma(u,c)}{|G_u| \cdot |G_s| \cdot \sum_{v \in g} |G_v|^{-1}} - \beta(\omega_u(s,c)\omega_p(s,c) + \omega_\sigma(s,c))$$

$\sum_{v \in g} |G_v|^{-1}$ in the denominator gives greater credence to "small" groups (where members belonging to many other groups weigh less heavily), with the intuition that smaller groups have more similar members. Note that in the worst case (where all sources belong to a single group and each assert a unique set of k claims) this can effectively create as many as $(k \cdot |S|)^2 - k \cdot |S|$ new assertions, with a corresponding increase in computational cost when running the fact-finder.

5.3 Encoding Groups and Attributes as Layers of Graph Nodes

Instead of using weighted assertions, we can add additional "layers" to represent groups and attributes directly. Each node in these layers will represent a group or attribute, with edges linking to its adjoining layers (either the sources or other groups/attributes), creating a k-partite graph (with $k > 3$ used to encode meta-groups and meta-attributes.) An standard fact-finder iteratively alternates between calculating the first layer (the claims) and the second layer (the sources), using the B and T functions, respectively. Now we replace these with generic "up" and "down" functions for each layer. For a k-partite graph with layers $L_{1...k}$, we define $U_j^i(L_j)$ over $j = 2...k$ and $D_j^i(L_j)$ over $j = 1...k-1$, with special cases $U_1^i(L_1) = D_1^{i-1}(L_1)$ and $D_k^i(L_k) = U_k^i(L_k)$. The U_j and D_j functions may differ for each layer j, or they may be the same over all layers. In each iteration i, we calculate the values $U_j^i(L_j)$ for layers $j = 2$ to k, and then calculate $D_j^i(L_j)$ for layers $j = k-1$ to

1. For example, to extend Sums to k layers, we calculate $U_j(e)$ and $D_j(e)$ as follows for $e \in L_j$:

$$U_j^i(e) = \sum_{f \in L_{j-1}} \omega(e,f) U_{j-1}^i(f)$$
$$D_j^i(e) = \sum_{f \in L_{j+1}} \omega(e,f) D_{j+1}^i(f)$$

Where $\omega(e,f) = \omega(f,e)$ is the edge weight between nodes e and f; if e or f is a group or attribute, $\omega(e,f)$ is 1 if e has attribute or group f or vice-versa, and 0 otherwise. In many cases, though, we may benefit from using an existing fact-finder over the claim and source layers, while using a different set of functions to mediate the interaction between the source and group/attribute layers. In particular, an information bottleneck often exists when calculating trustworthiness of a source in the "down phase", as it will be wholly dependent upon the trustworthiness of the groups to which it belongs: a source belonging to one overall-mediocre group may make many correct claims, but still be assigned a low trustworthiness score by the D function because of its group membership. This type of problem can be resolved by incorporating both the layer below *and* the layer above in each calculation; for example, for a given $D_j(e)$, we can define $\omega_{children} = \sum_{f \in L_{j-1}} \omega(e,f)$ and $D_j^{smooth}(e) = (1 + \omega_{children})^{-1} D_j(e) + \omega_{children}(1 + \omega_{children})^{-1} U_j(e)$, which returns a mixture of the value derived from e's ancestors, $D_j(e)$ and the value derived from e's descendants, $U_j(e)$, according to the (weighted) number of children e possesses, the intuition being that with more children the trustworthiness computed by $U_j(e)$ is more certain and should be weighted more highly, whereas with fewer children we should depend more upon broad inferences made from the ancestor groups and attributes. We will use $D_j^{smooth}(e)$ in our experiments.

5.3.1 Source Domain Expertise

The idea of incorporating the domain expertise of the source into the trust decision has been around at least as far back as Marsh's 1994 thesis (Marsh, 1994) and, in the generalized fact-finding framework, we can model it using the same techniques we used to model groups. For example, we expect a plant biologist to be more trustworthy on topics such as photosynthesis and genetic engineering, but less reliable on topics outside his expertise, such as fusion and computational complexity. Still, these aspects are not entirely separate: if we have two plant biologists A and B, and A gives accurate information about plant biology while B gives inaccurate information, we will tend to assign greater credence to A with respect to other domains (such as physics) as well–that is, we assume A is more "generally trustworthy" overall.

To implement this, we may create additional layers to represent our trustworthiness in a source for various domains. In Figure 3, we see that source s_1 has made claims in four different domains: cellular biology, ecology, astronomy, and classical mechanics. Each node shown in layers 2 and 3 is specific to s_1, representing his trustworthiness in that particular field or subfield, such that the sources are actually s_1's "Cellular", "Ecology", "Astro" and "Classic" nodes, which belong to two groups corresponding to s_1's biology and physics trustworthiness, which themselves belong to a metagroup corresponding to s_1's general trustworthiness.

5.3.2 Additional Layers versus Weighted Edges

Relative to adding edges to represent groups, expanding our model with additional layers increases the complexity of the algorithm, but prevents the quadratic expansion of the number of edges and

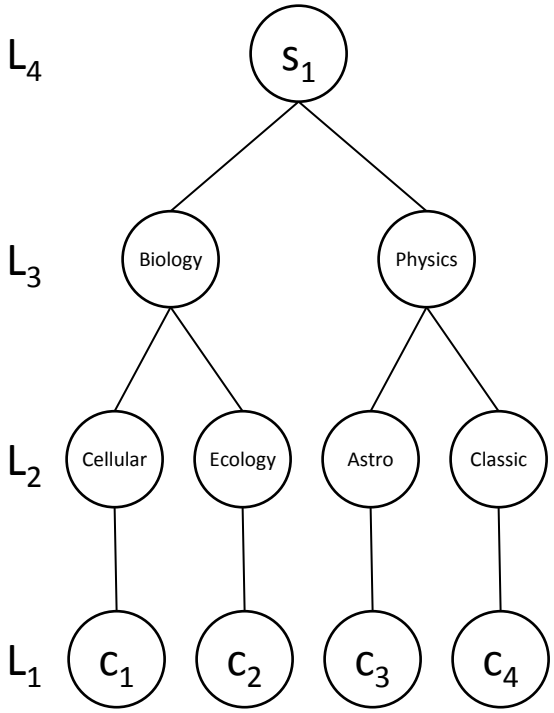

Figure 3: Use of additional layers to model specialization of the source s_1 into two categories, biology and physics, and four subcategories (cellular biology, ecology, astronomy and classical mechanics), each containing a single claim made by the source. The dotted lines connect to the same subcategories of other sources (not shown) making the same claims. See text for details.

corresponding increase in time complexity. More importantly, though, the flexibility in specifying the U and D functions for the higher layers representing groups and attributes allows us to augment an existing fact-finder to take advantage of them in a highly flexible way.

6. Constrained Fact-Finding

In constrained fact-finding, we incorporate rules and constraints to fact-finding. Concretely, to apply the user's specific and common-sense prior knowledge of claims to a fact-finding algorithm, we translate it into a linear program (LP). We then iterate the following until convergence or other stopping criteria:

1. Compute $T^i(s)$ for all $s \in S$
2. Compute $B^i(c)$ for all $c \in C$
3. "Correct" the beliefs $B^i(C)$ by applying the linear program

6.1 Propositional Linear Programming

To translate prior knowledge into a linear program, we first propositionalize our first-order formulae into propositional logic (Russell & Norvig, 2003). For example, assume we know that Tom is older than John and a person has exactly one age ($\exists_{x,y} Age(Tom, x) \wedge Age(John, y) \wedge x >$

$y) \wedge (\forall_{x,y,z} Age(x,y) \wedge y \neq z \Rightarrow \neg Age(x,z))$, and we are considering the following claims: $Age(Tom,30)$, $Age(Tom,40)$, $Age(John,25)$, $Age(John,35)$. Our propositional clauses (after removing redundancies) are then $Age(Tom,30) \Rightarrow Age(John,25) \wedge (Age(Tom,30) \oplus Age(Tom,40)) \wedge (Age(John,25) \oplus Age(John,35))$.

Each claim c will be represented by a proposition, and ultimately a $[0,1]$ variable in the linear program corresponding informally to $P(c)$.[1] Propositionalized constraints have previously been used with *integer* linear programming (ILP) using binary $\{0,1\}$ values corresponding to $\{false, true\}$ to find an (exact) consistent truth assignment minimizing some cost and solving a global inference problem, e.g. (Roth & Yih, 2004, 2007). However, propositional linear programming has two significant advantages:

1. ILP is "winner take all", shifting all belief to one claim in each mutual exclusion set (even when other claims are nearly as plausible) and finding the single most believable consistent *binary assignment*; we instead wish to find a *distribution* of belief over the claims that is consistent with our prior knowledge and as close as possible to the distribution produced by the fact-finder.
2. Linear programs can be solved in polynomial time (e.g. by interior point methods (Karmarkar, 1984)), but ILP is NP-hard.

To create our constraints, we first convert our propositional formula into conjunctive normal form. Then, for each disjunctive clause consisting of a set P of positive literals (claims) and a set N of negations of literals, we add the constraint $\sum_{c \in P} c_v + \sum_{c \in N} (1 - c_v) \geq 1$, where c_v denotes the $[0,1]$ variable corresponding to each c. The left-hand side is the union bound of at least one of the claims being true (or false, in the case of negated literals); if this bound is at least 1, the constraint is satisfied. This optimism can dilute the strength of our constraints by ignoring potential dependence among claims: $x \Rightarrow y$, $x \vee y$ implies y is true, but since we demand only $y_v \geq x_v$ and $x_v + y_v \geq 1$ we accept any value of y_v such that $y_v \geq x_v \geq 1 - y_v$. However, when the claims are mutually exclusive, the union bound is exact; a common constraint is of the form $q \Rightarrow r^1 \vee r^2 \vee \ldots$, where the r literals are mutually exclusive, which translates exactly to $r_v^1 + r_v^2 + \ldots \geq q_v$. Finally, observe that mutual exclusion amongst n claims $c^1, c^2, \ldots, c^n$ can be compactly written as $c_v^1 + c_v^2 + \ldots + c_v^n = 1$.

6.2 The Cost Function

Having seen how first-order logic can be converted to linear constraints, we now consider the cost function, a distance between the new distribution of belief satisfying our constraints and the original distribution produced by the fact-finder.

First we determine the number of "votes" received by each claim c, computed as $\omega_c = \omega(B(c))$, which should scale linearly with the certainty of the fact-finder's belief in c. Recall that the semantics of the belief score are particular to the fact-finder, so different fact-finders require different vote functions. TruthFinder has pseudoprobabilistic [0,1] beliefs, so we use $\omega_{inv}(x) = \min((1-x)^{-1}, m_{inv})$ with $m_{inv} = 10^{10}$ limiting the maximum number of votes possible; we assume $1/0 = \infty$. ω_{inv} intuitively scales with "error": a belief of 0.99 receives ten times the votes of 0.9 and has a tenth the error (0.01 vs. 0.1). For the remainder of the fact-finders whose beliefs are already "linear", we use the identity function $\omega_{idn}(x) = x$.

1. This is a slight mischaracterization, since our linear constraints only *approximate* intersections and unions of events (where each event is "claim c is true"), and we will be satisfying them subject to a linear cost function.

The most obvious choice for the cost function might be to minimize "frustrated votes": $\sum_{c \in C} \omega_c(1-c_v)$. Unfortunately, this results in the linear solver generally assigning 1 to the variable in each mutual exclusion set with the most votes and 0 to all others (except when constraints prevent this), shifting all belief to the highest-vote claim and yielding poor performance. Instead, we wish to satisfy the constraints while keeping each c_v close to ω_c/ω_{M_c}, where $\omega_{M_c} = \sum_{d \in M_c} \omega_d$, and thus shift belief among claims as little as possible. We use a weighted Manhattan distance called **VoteDistance**, where the cost for increasing the belief in a claim is proportional to the number of votes against it, and the cost for decreasing belief is proportional to the number of votes for it:

$$\sum_{c \in C} \max \begin{pmatrix} (\omega_{M_c} - \omega_c) \cdot (c_v - \omega_c/\omega_{M_c}), \\ \omega_c \cdot (\omega_c/\omega_{M_c} - c_v) \end{pmatrix}$$

The belief distribution found by our linear program will thus be the one that satisfies the constraints while simultaneously minimizing the number of votes frustrated by the change from the original distribution. Note that for any linear expressions e and f we can implement $\max(e, f)$ in the objective function by replacing it with a new $[-\infty, \infty]$ helper variable x and adding the linear constraints $x \geq e$ and $x \geq f$.

6.3 Values → Votes → Belief

Solving the linear program gives us $[0, 1]$ values for each variable c_v, but we need to calculate an updated belief $B(c)$. (Pasternack & Roth, 2010) proposes two methods for this:

Vote Conservation: $B(c) = \omega^{-1}(c_v \cdot \omega_{M_c})$
Vote Loss: $B(c) = \omega^{-1}(\min(\omega_c,\ c_v \cdot \omega_{M_c}))$

ω^{-1} is an inverse of the vote function: $\omega^{-1}_{idn}(x) = x$ and $\omega^{-1}_{inv}(x) = 1 - (1 + y)^{-1}$. Vote Conservation reallocates votes such that the total number of votes in each mutual exclusion set, ω_M, remains the same after the redistribution. However, if the constraints force c to lose votes, should we believe the other claims in M_c more? Under Vote Loss, a claim can *only* lose votes, ensuring that if other claims in M_c become less believable, c does not itself become more believable relative to claims in other mutual exclusion sets. We found Vote Loss slightly better on average and used it for all reported results.

6.4 LP Decomposition

Frequently our linear programs can be *decomposed* into smaller problems that can be solved independently. If there exists a subset of linear constraints $L' \subset L$ that contain a set of variables $V' \subset V$ such that $\forall_{v \in V', l \in L/L'}\ v \notin l$, then L' together with the terms in the cost function containing the variables V' can be solved as a separate LP.

We can also reduce running time by observing that, for any such "sub-LP", it is easy to set each variable c_v to ω_c/ω_{M_c} (yielding the minimum possible cost of 0) and check if the constraints are satisfied—if they are, the optimal solution is found without invoking the linear solver. Together, these techniques allowed us to solve most LPs one or two orders of magnitude faster in our experiments (almost always within a matter of seconds), taking more than a minute to solve on a modest desktop machine only when the presence of tens of thousands of constraints prevented meaningful decomposition.

6.5 Tie Breaking

We must also address "ties" between claims with the same number of votes. If the linear solver is allowed to break these arbitrarily, the results may vary from solver to solver. This is of particular concern when using a chain of solvers (our experiments used Microsoft Solver Foundation (MSF) simplex → lp_solve simplex → MSF interior point) to enable "fallback" when one solver fails and consistent behavior is required. To handle this we identify pairs of claims with the same number of votes in each decomposed LP, multiplying the votes of one by $1 + 10^{-10}$ and repeating until no pair of claims is tied. Which claim gets slightly boosted depends upon a "precedence" that is assigned randomly at the start of the experiment.

6.6 "Unknown" Augmentation

Augmenting our data with "Unknown" claims ensures that every LP is feasible and can be used to model our ignorance given a lack of sufficient information or conflicting constraints. An Unknown claim U_M is added to every mutual exclusion set M (but invisible to the fact-finder) and represents our belief that *none* of the claims in M are sufficiently supported. Now we can write the mutual exclusion constraint for M as $U_M + \sum_{c \in M} c_v = 1$. When propositionalizing FOL, if a disjunctive clause contains a non-negated literal for a claim c, then we add $\vee\, U_{M_c}$ to the clause. For example, $Age(John, 35) \Rightarrow Age(Tom, 40)$ becomes $Age(John, 35) \Rightarrow Age(Tom, 40) \vee Age(Tom, Unknown)$. The only exception is when the clause contains claims from only one mutual exclusion set (e.g. "I know Sam is 50 or 60"), and so the LP can only be infeasible if the user directly asserts a contradiction (e.g. "Sam is 50 *and* Sam is 60"). The Unknown itself has a fixed number of votes that cannot be lost; this effectively "smooths" our belief in the claims and imposes a floor for believability. If $Age(Kim, 30)$ has 5 votes, $Age(Kim, 35)$ has 3 votes, and $Age(Kim, Unknown)$ is fixed at 6 votes, we hold that Kim's age is unknown due to lack of evidence. The number of votes that should be given to each Unknown for this purpose depends, of course, on the particular fact-finder and ω function used; in our experiments, we are not concerned with establishing ignorance and thus assign 0 votes.

7. Experimental Results

To evaluate our extensions to fact-finding, both the generalization of the fact-finders themselves and the application of constraints to encode prior knowledge, our experiments apply a number of state-of-the-art fact-finding algorithms to both real-world and semi-synthetic datasets. We considered both extensions separately, finding each was independently able to improve the accuracy of trust decisions by incorporating different types of background knowledge into the fact-finding process, and then combined these orthogonal components into a joint model able to achieve significantly better results than were possible using either alone.

7.1 Data

A number of real-world datasets are used, including two (Population and Biography) extracted from Wikipedia infoboxes (Wu & Weld, 2007) (semi-structured tables with various fields within Wikipedia articles). An example of an infobox for the city of Laguna Beach is shown in Figure 4.

City of Laguna Beach	
— City —	
Country	United States
State	California
County	Orange
Area	
- Total	9.7 sq mi (25.2 km^2)
- Land	8.8 sq mi (22.9 km^2)
- Water	0.9 sq mi (2.3 km^2)
Population (2000)	
- Total	23,727
- Density	2,683.5/sq mi (1,036.1/km^2)

Figure 4: Example of a Wikipedia Infobox

7.1.1 POPULATION

(Pasternack & Roth, 2010) collected Wikipedia infoboxes for settlements (Geobox, Infobox Settlement, Infobox City, etc.) to obtain 44,761 population claims qualified by year (e.g. triples such as (Denver, 598707, 2008)) from 171,171 sources ("editors", in Wikipedia parlance), with a test set of 308 "true" claims taken from U.S. census data (omitting the many cases where editors did not contest the population, or where all claims in Wikipedia were wrong). To allow for a direct comparison between generalized fact-finding and declarative prior knowledge, we use the population dataset across both sets of experiments and for the combined, joint model as well.

7.1.2 BOOKS

For generalized fact-finding, we also have (Yin et al., 2008)'s Books dataset, extracted from online bookstore websites. The Books dataset is a collection of 14,287 claims of the authorship of various books by 894 websites, where a website asserts that a person was an author of a book (e.g. (Bronte, "Jane Eyre")) explicitly by including them in the list of authors, or implicitly asserts a person was *not* an author (e.g. (¬Bronte, "Jane Eyre")) by omitting them from the list (when at least one other website lists that person as an author of the book—if nobody lists a person as an author, his non-authorship is not disputed and can be ignored). The test set is 605 true claims collected by examining the books' covers.

7.1.3 BIOGRAPHY

For our declarative prior knowledge experiments, (Pasternack & Roth, 2010) created the Biography dataset by scanning Wikipedia infoboxes to find 129,847 claimed birth dates, 34,201 death dates, 10,418 parent-child pairs, and 9,792 spouses as reported by 1,167,494 editors. To get "true" birth and death dates, we extracted data from several online repositories (after satisfying ourselves that they were independent and not derived from Wikipedia!), eliminating any date these sources disagreed upon, and ultimately obtained a total of 2,685 dates to test against.

Data	Weights	Vote	Sum	3Est	TF^c	A·L	$Inv^{1.2}$	$Pool^{1.4}$
Pop	Unweighted	81.49	81.82	81.49	84.42	80.84	87.99	80.19
Pop	Tuned	81.90	82.90	82.20	87.20	83.90	90.00	80.60
Pop	Best	81.82	83.44	82.47	87.66	86.04	90.26	81.49

Table 2: Experimental Results for Tuned Assertion Certainty. All values are percent accuracy.

7.1.4 AMERICAN VS. BRITISH SPELLING

Finally, we examined a domain where the truth was plainly subjective and thus the user's prior knowledge is essential: identifying the "correct" spelling of words given 209,189 articles from the British National Corpus, The Washington Post and Reuters written by 9,387 distinct authors (Pasternack & Roth, 2010).

7.2 Experimental Setup

For our experiments we used a number of state-of-the-art fact-finding algorithms: Sums / Hubs and Authorities (**Sum**), 3-Estimates (**3Est**), simplified TruthFinder (**TF^s**), "full" TruthFinder (**TF^c**), Average·Log (**A·L**), Investment with $g = 1.2$ (**$Inv^{1.2}$**), and PooledInvestment with $g = 1.4$ (**$Pool^{1.4}$**). The voting baseline (**Vote**) simply chooses the claim asserted by the most sources. The number of iterations used for each fact-finder was fixed at 20. To evaluate accuracy, after the final iteration we looked at each mutual exclusion set M and predicted the highest-belief claim $c \in M$ (other than u_M, if applicable), breaking ties randomly, and checked if it was the true claim t_M. We omitted any M that did not contain a true claim (all known claims are false) and any M that was trivially correct (containing only one claim [other than u_M, if applicable]).

7.3 Generalized Fact-Finding

7.3.1 TUNED ASSERTION CERTAINTY

A user modifying a field of interest in an infobox (e.g. the *population_total* field) is clearly asserting the corresponding claim ("population $= x$"), but what if he edits another part of the infobox, or somewhere else on the page? Did he also read and approve the fields containing the claims we are interested in, implicitly asserting them? We can simply consider only direct edits of a field containing a claim to be an assertion of that claim, but this ignores the large number of potential assertions that may be implicit in an editor's decision to *not* change the field.

This may be considered either uncertainty in information extraction (since we are not able to extract the author's true intent) or uncertainty on the part of the authors (an editor leaves a field unaltered because he believes it is "probably" true). In either case, we can weight the assertions to model this uncertainty in the generalized fact-finder. The information extractor provides a list of all edits and their type (editing the field of interest, another field in the infobox, or elsewhere in the document), and each type of edit implies a different certainty (a user editing another field in the infobox is more likely to have read and approved the neighboring field of interest than a user editing a different portion of the document), although we do not know what those levels of certainty are. These can be discovered by tuning with a subset of the test set and evaluating on the remainder, varying the relative weights of the "infobox", "elsewhere", and "field of interest" assertions. The results are shown in Table 2. In the "unweighted" case only direct edits to the "field of interest"

Data	Assertions	Vote	Sum	3Est	TF^c	A·L	Inv$^{1.2}$	Pool$^{1.4}$
Pop	Unweighted	71.10	77.92	71.10	78.57	76.95	78.25	74.35
Pop	Generalized (Weighted)	**76.95**	**78.25**	**76.95**	**80.19**	**78.90**	**84.09**	**78.25**
Books	Unweighted	80.63	77.93	80.74	80.56	79.21	77.83	81.20
Books	Generalized (Weighted)	**81.88**	**81.13**	**81.88**	**82.90**	**81.96**	**80.50**	**81.93**

Table 3: Experimental Results for Uncertainty in Information Extraction

β	Vote	Sum	3Est	TF^c	A·L	Inv$^{1.2}$	Pool$^{1.4}$
(No groups)	81.49	81.82	81.49	84.42	80.84	87.99	80.19
0.7	**84.09**	**84.09**	**84.42**	**85.71**	**84.74**	84.74	**83.44**
0.5	83.77	**84.09**	**84.42**	85.06	84.09	87.01	82.79
0.3	82.47	83.77	83.77	84.74	83.77	87.01	82.79
0.00001	83.44	82.14	83.44	84.42	81.49	**88.96**	80.51

Table 4: Experimental Results for Groups using Weighted Assertions.

are considered, and "infobox" and "elsewhere" edits are ignored (giving all edits equal weight fares much worse).

We tuned over 208 randomly-chosen examples and evaluated on the remaining 100, repeating the experiment ten times. We also tuned (and tested) with all 308 labeled examples to get the "Best" results, only slightly better than those from legitimate tuning. As expected, assigning a smaller weight to the "infobox" assertions (relative to the "field of interest") and a much lower weight to the "elsewhere" assertions yielded the greatest results, confirming our common-sense assumption that edits close to a field of interest confer more supervision and implicit approval than those elsewhere on the page. We find a significant gain across all fact-finders, notably improving the top Investment result to 90.00%, demonstrating that generalized fact-finders can dramatically increase performance.

7.3.2 Uncertainty in Information Extraction

We next consider the case where the information extractor is uncertain about the putative claims, but provides an (accurate) estimate of $\omega_u(s, c) = P(s \rightarrow c)$, the probability that source s made a given claim c.

For the Population dataset, we augment each mutual exclusion set M with an additional (incorrect) claim, ensuring $|M| \geq 2$. For each assertion $s \rightarrow c$ we select another $c' \in M_c$, and draw a p from a Beta(4,1) distribution ($\mathbb{E}(p) = 0.8 \Rightarrow 20\%$ chance of error). We then set $\omega_u(s, c) = p$ and $\omega_u(s, c') = 1 - p$. In the unweighted case (where edge weights must be 0 or 1), we keep the edge between s and c if $p \geq 0.5$, and replace that edge with one between s and c' if $p < 0.5$.

For the Books dataset, each mutual exclusion set had exactly two claims (a person is either an author of a book or he is not) and thus did not require augmentation. Here we drew p from a Beta(2,1) distribution ($\mathbb{E}(p) = 2/3$), corresponding to a greater (33%) chance of error. Our results are shown in Table 3; on both datasets, generalized fact-finders easily outperform their standard counterparts.

Description	Sum	TF^c	A·L	Inv$^{1.2}$	Inv$^{1.2}$/Avg	Pool$^{1.4}$/Avg
No Groups	81.82	**84.42**	80.84	87.99	87.99	80.19
Group Layer	83.77	83.44	**84.42**	83.44	88.64	64.94
Group Layer with D_2^{smooth}	**84.74**	84.09	82.79	**88.96**	**89.61**	**84.74**
Tuned + Group Layer	**86.10**	83.30	**87.00**	**88.50**	**90.00**	77.90
Tuned + Group Layer with D_2^{smooth}	83.20	**85.30**	84.20	87.40	**90.00**	**83.50**

Table 5: Experimental Results for Groups as an Additional Layer.

7.3.3 Groups as Weighted Assertions

Using the Population data we considered three groups of editors: administrators, blocked users, and regular users with at least one template on their user page (intended to capture more serious editors). To keep things simple, we allowed each user to belong to at most one of these groups—if an administrator had been blocked, he nonetheless belonged to the administrator group; if an otherwise "regular" user were blocked, he (of course) belonged to the blocked group. Given that administrators are promoted to that position by being trusted by other Wikipedia editors, and that blocked users are blocked by trusted administrators for (presumable) misbehavior, we expected that administrators will be relatively trustworthy on the whole, while blocked users will be more untrustworthy, with serious editors somewhere in between. We then encoded these groups as weighted assertions, using ω_g with arbitrarily chosen β parameters, as shown in Table 4. We see improved performance with all β values tested, with the exception of the Investment algorithm, which requires a much lower β; we can conclude from this that β should be tuned independently on each fact-finder for best results.

7.3.4 Groups as Additional Layers

We next took the same three groupings of editors (administrators, blocked users, and regular users) and added them as a third layer in our generalized fact-finders, continuing to use the same Population dataset as before. For most fact-finders, we can directly adapt the T and B functions as U and D functions, respectively, though this excludes PooledInvestment (which depends on mutual exclusion sets) and 3-Estimates (whose "claim difficulty" parameters are not readily extended to groups). In the former case, we can calculate the trustworthiness of the groups in the third layer as a weighted average of the trustworthiness of its members, giving us $U_3^i(g) = \sum_{s \in g} U_2^i(s)/|g|$, where g is a group and $|g|$ is the number of sources it contains. Likewise, we can calculate the trustworthiness a source inherits from its groups as the weighted average of the groups' trustworthiness, giving $D_2^i(s) = \sum_{g \in G_s} D_3^i(g)/|G_s|$, where G_s is the set of groups to which source s belongs (recall that, since there are three layers, $D_3^i(g) = U_3^i(g)$). We can use these new U_3 and D_2 functions to handle the interaction between the group layer and the source layer, while continuing to use an existing fact-finder to mediate the interaction between the source layer and claim layer. We apply this hybrid approach to two fact-finders, giving us Inv$^{1.2}$/Avg, and Pool$^{1.4}$/Avg. Finally, note that regardless of the choice of D_2, we are discarding the trustworthiness of each source as established by its claims in favor of the collective trustworthiness of its groups, an information bottleneck. When we have ample claims for a source, its group membership is less important; however, when there are few claims, group membership becomes much more important due to the lack of other "evidence". The previously described D_j^{smooth} captures this idea by scaling the impact of groups on a source by

Dataset	Prior Knowledge	Vote	Sum	3Est	TF^s	TF^c	A·L	Inv$^{1.2}$	Pool$^{1.4}$
Pop	$\emptyset$	81.49	81.82	81.49	82.79	84.42	80.84	**87.99**	80.19
Pop	Growth$_{IBT}$	82.79	79.87	77.92	82.79	**86.36**	80.52	85.39	79.87
Pop	Growth$_{L+I}$	82.79	79.55	77.92	83.44	85.39	80.52	**89.29**	80.84
Pop	Larger$^{2500}_{IBT}$	85.39	85.06	80.52	86.04	87.34	84.74	**89.29**	84.09
Pop	Larger$^{2500}_{L+I}$	85.39	85.06	80.52	86.69	86.69	84.42	**89.94**	84.09
SynPop	$\emptyset$	73.45	87.76	84.87	56.12	87.07	**90.23**	89.41	90.00
SynPop	Pop$\pm 8\%_{IBT}$	88.31	95.46	92.16	**96.42**	95.46	96.15	95.46	**96.42**
SynPop	Pop$\pm 8\%_{L+I}$	88.31	94.77	92.43	82.39	95.32	95.59	**96.29**	96.01
Bio	$\emptyset$	89.80	89.53	89.80	73.04	**90.09**	89.24	88.34	90.01
Bio	CS$_{IBT}$	89.20	89.61	89.20	72.44	89.91	89.35	88.60	**90.20**
Bio	CS$_{L+I}$	89.20	89.61	89.20	57.10	90.09	89.35	88.49	**90.24**
Bio	CS+Decades$_{IBT}$	90.58	90.88	90.58	80.30	91.25	90.91	90.02	**91.32**
Bio	CS+Decades$_{L+I}$	90.58	90.91	90.58	69.27	90.95	90.91	90.09	**91.17**
Spell	$\emptyset$	13.54	9.37	11.96	**41.93**	7.93	10.23	9.36	9.65
Spell	Words$^{100}_{IBT}$	13.69	9.02	12.72	**44.28**	8.05	9.98	11.11	8.86
Spell	Words$^{100}_{L+I}$	13.69	8.86	12.08	**46.54**	8.05	9.98	9.34	7.89
Spell	CS+Words$^{100}_{IBT}$	35.10	31.88	35.10	56.52	29.79	32.85	73.59	**80.68**
Spell	CS+Words$^{100}_{L+I}$	35.10	31.72	34.62	**55.39**	22.06	32.21	30.92	29.95

Table 6: Constrained Fact-Finding Results ($\emptyset$ indicates no prior knowledge)

the (weighted) number of claims made by that source. We show results both with and without this smoothing in Table 5.

Except for TruthFinder, group information always improves the results, although "smoothing" may be required. We also tuned the assertion certainty as we did in Table 2 in conjunction with the use of groups; here we find no relative improvement for Investment or TruthFinder, but gain over both tuning and groups alone for all other fact-finders.

7.4 Constrained Fact-Finding

7.4.1 IBT vs. L+I

We can enforce our prior knowledge against the beliefs produced by the fact-finder in each iteration, or we can apply these constraints just once, after running the fact-finder for 20 iterations without interference. By analogy to (Punyakanok, Roth, Yih, & Zimak, 2005), we refer to these approaches as inference based training (IBT) and learning + inference (L+I), respectively. Our results show that while L+I does better when prior knowledge is not entirely correct (e.g. "Growth" in the city population domain), generally performance is comparable when the effect of the constraints is mild, but IBT can outperform when prior knowledge is vital (as in the spelling domain) by allowing the fact-finder to learn from the provided corrections.

7.4.2 City Population

Our "common sense" knowledge is that population grows over time ("Growth" in Table 6); therefore, $\forall_{v,w,x,y,z} pop(v,w,y) \wedge pop(v,x,z) \wedge y < z \Rightarrow x > w$. Of course, this often does not hold

true: cities can shrink, but performance was nevertheless superior to no prior knowledge whatsoever. The L+I approach does appreciably better because it avoids forcing these sometimes-incorrect constraints onto the claim beliefs while the fact-finder iterates (which would propagate the resulting mistakes), instead applying them only at the end where they can correct more errors than they create. The sparsity of the data plays a role—only a fraction of cities have population claims for multiple years, and those that do are typically larger cities where the correct claim is asserted by an overwhelming majority, greatly limiting the potential benefit of our Growth constraints. We also considered prior knowledge of the relative sizes of some cities, randomly selecting 2500 pairs of them (a, b), where a was more populous than b in year t, asserting $\forall_{x,y} pop(a, x, t) \wedge pop(b, y, t) \Rightarrow x > y$. This "Larger" prior knowledge proved more effective than our oft-mistaken Growth constraint, with modest improvement to the highest-performing Investment fact-finder, and Investment$_{L+I}$ reaches **90.91%** with 10,000 such pairs.

7.4.3 Synthetic City Population

As our real-world data was sparse, we created a synthetic dataset to determine how effective common-sense knowledge would be in the presence of "dense" data. We chose 100 random (real) cities and created 100 authors whose individual accuracy a was drawn uniformly from $[0, 1]$. Between 1 and 10 claims (also determined uniformly) were made about each city in each year from 2000 to 2008 by randomly-selected authors. For each city with true population p and year, four incorrect claims were created with populations selected uniformly from $[0.5p, 1.5p]$, each author claiming p with probability a and otherwise asserting one of the four incorrect claims. Our common-sense knowledge was that population did not change by more than 8% per year (also tested on the Wikipedia dataset but with virtually no effect). Like "Growth", "Pop$\pm 8\%$" does not always hold, but a change of more than 8% is much rarer than a shrinking city. These constraints greatly improved results, although we note this would diminish if inaccurate claims had less variance around the true population.

7.4.4 Basic Biographies

Our common sense ("CS") knowledge was: nobody dies before they are born, people are infertile before the age of 7, nobody lives past 125, all spouses have overlapping lifetimes, no child is born more than a year after a parent's (father's) death, nobody has more than two parents, and nobody is born or dies after 2008 (the "present day", the year of the Wikipedia dump). Applying this knowledge roughly halved convergence times, but had little effect on the results due to data sparsity similar to that seen in the population data—while we know many birthdays and some death dates, relatively few biographies had parent-child and spouse claims. To this we also added knowledge of the decade (but not the exact date) in which 15,145 people were born ("CS+Decades"). Although common sense alone does not notably improve results, it does very well in conjunction with specific knowledge.

7.4.5 American vs. British Spelling

Prior knowledge allows us to find a truth that conforms with the user's viewpoint, even if that viewpoint differs from the norm. After obtaining a list of words with spellings that differed between American and British English (e.g. "color" vs. "colour"), we examined the British National Corpus as well as Washington Post and Reuters news articles, taking the source's (the article author's) use

Prior Knowledge	Group Layer	Sum	TF^c	A·L	Inv$^{1.2}$	Inv$^{1.2}$/Avg	Pool$^{1.4}$/Avg
$\emptyset$	No Groups	81.82	84.42	80.84	87.99	87.99	80.19
$\emptyset$	Unsmoothed	83.77	83.44	84.42	83.44	88.64	64.94
$\emptyset$	D_2^{smooth}	84.74	84.09	82.79	88.96	89.61	84.74
Larger$_{IBT}^{2500}$	No Groups	85.06	87.34	84.74	89.29	89.29	84.09
Larger$_{L+I}^{2500}$	No Groups	85.06	86.69	84.42	89.94	89.94	84.09
Larger$_{IBT}^{2500}$	Unsmoothed	87.34	86.04	**86.69**	85.71	89.94	72.40
Larger$_{IBT}^{2500}$	D_2^{smooth}	**87.99**	**87.66**	**86.69**	**90.58**	**90.26**	**87.99**

Table 7: Experimental Results for the Joint Framework.

of a disputed word as a claim that his spelling was correct. Our goal was to find the "true" British spellings that conformed to a British viewpoint, but American spellings predominate by far. Consequently, without prior knowledge the fact-finders do very poorly against our test set of 694 British words, predicting American spelling instead in accordance with the great majority of authors (note that accuracy from an American perspective is $1-$"British" accuracy). Next we assumed that the user already knew the correct spelling of 100 random words (removing these from the test set, of course), but with little effect. Finally, we added our common sense ("CS") knowledge: if a spelling a is correct and of length ≥ 4, then if a is a substring of b, $a \Leftrightarrow b$ (e.g. colour $\Leftrightarrow$ colourful). Furthermore, while we do not know a priori whether a spelling is American or British, we do know if e and f are different spellings of the same word, and, if two such spellings have a chain of implication between them, we can break all links in this chain (while some American spellings will still be linked to British spellings, this removes most such errors). Interestingly, common sense alone actually *hurts* results (e.g. PooledInvestment (IBT) gets 6.2%), as it essentially makes the fact-finders more adept at finding the predominant American spellings! However, when some correct spellings are known, results improve greatly and demonstrate IBT's ability to spread strong prior knowledge, easily surpassing L+I. Results improve further with more known spellings (PooledInvestment gets **84.86%** with CS+Words$_{IBT}^{200}$).

7.5 The Joint Generalized Constrained Fact-Finding Framework

Our final experiments combine generalized and constrained fact-finding to create the full, joint framework, capable of leveraging a very broad set of background and domain knowledge in our trust decision. We again use the Population dataset, applying the Larger^{2500} declarative prior knowledge set to generalized fact-finders using the Wikipedia editor group information (administrator, normal user, blocked user) encoded as an additional layer. The results in Table 7 show a significant and consistent gain using the joint framework with D_2^{smooth} across all fact-finders (with IBT; L+I results [not shown] were only slightly lower). The top result from the Investment fact-finder rises to 90.58%, up from 89.61% using only group information, or 89.94% using only declarative prior knowledge, while even the very simple Sums fact-finder achieves a respectable 87.99% performance, up from 81.82% with no background knowledge of any kind.

8. Conclusion

Generalized Constrained Fact-Finding offers a framework for incorporating a broad range of knowledge into our trust decisions by augmenting fact-finding algorithms, both by generalizing the fact-finders themselves and by constraining them with declarative prior knowledge. Generalized fact-finding allows us to encode factors such as information extraction and source uncertainty, similarity between the claims, and source groupings and attributes, with substantial and consistent performance gains across a broad range of fact-finders. Simultaneously, declarative prior knowledge, expressed as constraints over the belief in the claims, proves vital when the user's subjective truth differs from the norm, as it did in the Spelling domain, and even in other experiments where the "truth" is less contested both common-sense and specific knowledge provided significant benefit; moreover, as the constraints are enforced by a linear program, the framework remains polynomial-time, an essential characteristic when dealing with real-world "web-scale" data. As both generalized and constrained fact-finding are orthogonal, they may be readily used together, achieving better results than were possible with either method alone and allowing the full breadth of our available information to be jointly leveraged in determining the oft-subjective truth in the presence of a morass of conflicting information.

References

Adler, B. T., & de Alfaro, L. (2007). A content-driven reputation system for the Wikipedia. *WWW '07*, *7*, 261–270.

Adler, B. T., Chatterjee, K., Alfaro, L. D., Faella, M., Pye, I., & Raman, V. (2008). Assigning Trust to Wikipedia Content. *Computer Engineering*.

Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2), 58–71.

Bailey, B. P., Gurak, L. J., & Konstan, J. A. (2001). An examination of trust production in computer-mediated exchange. *Proceedings of the 7th Conference on Human Factors and the Web*.

Bertino, E., Dai, C., Lim, H.-s., & Lin, D. (2008). High-Assurance Integrity Techniques for Databases. *Event (London)*, 244–256.

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

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E., & Mitchell, T. (2010). Toward an Architecture for Never-Ending Language Learning. *AAAI*.

Chang, M., Ratinov, L., & Roth, D. (2012). Structured Learning with Constrained Conditional Models. *Machine Learning*, *88*(3), 399–431.

Dai, C., Lin, D., Bertino, E., & Kantarcioglu, M. (2008a). An approach to evaluate data trustworthiness based on data provenance. *SDM*.

Dai, C., Lin, D., Bertino, E., & Kantarcioglu, M. (2008b). Trust evaluation of data provenance. Tech. rep., CERIAS Technical Report.

Dawid, A. P., & Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 20–28.

Dong, X., Berti-equille, L., & Srivastava, D. (2009a). Integrating conflicting data: the role of source dependence. *Technical report, AT&T Labs-Research, Florham Park, NJ*.

Dong, X., Berti-Equille, L., & Srivastava, D. (2009b). Truth discovery and copying detection in a dynamic world. *VLDB*.

Fogg, B. J., Marshall, J., Kameda, T., Solomon, J., Rangnekar, A., Boyd, J., & Brown, B. (2001a). Web credibility research: a method for online experiments and early study results. *Conference on Human Factors in Computing Systems*, 295–296.

Fogg, B. J., Swani, P., Treinen, M., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., Paul, J., Rangnekar, A., & Others (2001b). What makes Web sites credible?: a report on a large quantitative study. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 61–68.

Fogg, B. J., & Tseng, H. (1999). The elements of computer credibility. *CHI*, 80–87.

Galland, A., Abiteboul, S., Marian, A., & Senellart, P. (2010). Corroborating information from disagreeing views. In *WSDM*.

Gil, Y., & Artz, D. (2007). Towards content trust of web resources. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(4), 227–239.

Josang, A. (1997). Artificial reasoning with subjective logic. *2nd Australian Workshop on Commonsense Reasoning*.

Josang, A., Marsh, S., & Pope, S. (2006). Exploring different types of trust propagation. *Lecture Notes in Computer Science*, *3986*, 179.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, *4*(4), 373–395.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*(5), 604–632.

Le, H. K., Pasternack, J., Ahmadi, H., Gupta, M., Sun, Y., Abdelzaher, T., Han, J., Roth, D., Szymanski, B., & Adali, S. (2011). Apollo : Towards Factfinding in Participatory Sensing. *IPSN*.

Levien, R. (2008). Attack-resistant trust metrics. *Computing with Social Trust*, 121–132.

Manchala, D. (1998). Trust metrics, models and protocols for electronic commerce transactions. *Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183)*, 312–321.

Marsh, S. (1994). Formalising Trust as a Computational Concept. *PhD thesis, University of Stirling*.

Novak, V., Perfilieva, I., & Mockof, J. (1999). *Mathematical principles of fuzzy logic*. Kluwer Academic Publishers.

Pasternack, J., & Roth, D. (2010). Knowing What to Believe (when you already know something). In *COLING*.

Pasternack, J., & Roth, D. (2011a). Generalized Fact-Finding. In *WWW*.

Pasternack, J., & Roth, D. (2011b). Making Better Informed Trust Decisions with Generalized Fact-Finding. In *IJCAI*.

Pasternack, J., & Roth, D. (2013). Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1009–1020. International World Wide Web Conferences Steering Committee.

Poon, H., & Domingos, P. (2007). Joint inference in information extraction. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 22, p. 913. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Punyakanok, V., Roth, D., Yih, W., & Zimak, D. (2005). Learning and inference over constrained output. In *International Joint Conference on Artificial Intelligence*, Vol. 19.

Roth, D., & Yih, W. (2007). Global Inference for Entity and Relation Identification via a Linear Programming Formulation. In Getoor, L., & Taskar, B. (Eds.), *Introduction to Statistical Relational Learning*. MIT Press.

Roth, D., & Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In *CoNLL*, pp. 1–8.

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (Second edition). Prentice Hall.

Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, *24*(1), 33–60.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton University Press Princeton, NJ.

Suchanek, F., Sozio, M., & Weikum, G. (2009). SOFIE: a self-organizing framework for information extraction. *Proceedings of the 18th*, 631–640.

Tseng, S., & Fogg, B. J. (1999). Credibility and computing technology. *Communications of the ACM*, *42*(5), 39–44.

Von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 319–326. ACM.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). recaptcha: Human-based character recognition via web security measures. *Science*, *321*(5895), 1465–1468.

Wu, F., & Weld, D. S. (2007). Autonomously semantifying wikipedia. *CIKM*.

Yin, X., Yu, P. S., & Han, J. (2008). Truth Discovery with Multiple Conflicting Information Providers on the Web. *IEEE Transactions on Knowledge and Data Engineering*, *20*(6), 796–808.

Yu, B., & Singh, M. P. (2003). Detecting deception in reputation management. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS '03*, 73.

Zeng, H., Alhossaini, M., Ding, L., Fikes, R., & McGuinness, D. L. (2006). Computing trust from revision history. *Intl. Conf. on Privacy, Security and Trust*.