Knowing What to Believe (when you already know something)

Jeff Pasternack **Dan Roth**
University of Illinois, Urbana-Champaign
{jpaster2, danr}@uiuc.edu

Abstract

Although much work in NLP has focused on simply determining what a document *means*, we also must know whether or not to *believe* it. Fact-finding algorithms attempt to identify the "truth" among competing claims in a corpus, but fail to take advantage of the user's prior knowledge and presume that truth itself is universal and objective rather than subjective. We introduce a framework for incorporating prior knowledge into *any* fact-finding algorithm, expressing both general "common-sense" reasoning and specific facts already known to the user as first-order logic and translating this into a tractable linear program. As our results show, this approach scales well to even large problems, both reducing error and allowing the system to determine truth respective to the user rather than the majority. Additionally, we introduce three new fact-finding algorithms capable of outperforming existing fact-finders in many of our experiments.

1 Introduction

Although establishing the trustworthiness of the information presented to us has always been a challenge, the advent of the Information Age and the Internet has made it more critical. Blogs, wikis, message boards and other collaborative media have eliminated the high entry barrier–and, with it, the enforced journalistic standards–of older, established media such as newspapers and television, and even these sometimes loosen their fact-checking in the face of increased competitive pressure. Consequently, we find that corpora derived from these sources now offer far more numerous views of far more questionable veracity. If one author claims Mumbai is the largest city in the world, and another claims it is Seoul, who do 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?) 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. Note that the user may be another computational system rather than a human (e.g. building a knowledge base of city sizes for question answering), and often neither the user's nor the information source's perspective will be explicit (e.g. an author will not fully elaborate "the largest city by metropolitan area bounded by...") but will instead be 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).

The most basic approach is to take a vote: if multiple claims are mutually exclusive of each other, select the one asserted by the most sources. In our experiments, sources will be the authors of the document containing the claim, but other sources could be publishers/websites (when no authorship is given), an algorithm that outputs claims, etc. Although sometimes competitive, we found voting to be generally lackluster. A class of algorithms called fact-finders are often a dramatic improvement, but are incapable of taking advantage of the user's prior knowledge. Our framework translates prior knowledge (expressed as first-order logic) into a linear program that constrains the claim beliefs produced by a fact-finder, ensuring that our belief state is consistent with both common sense ("cities usually grow") and known facts ("Los Angeles is more populous than Wichita"). While in the past first-order logic has been translated to NP-hard integer linear programs, we use polynomial-time-solvable linear programs, al-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 877–885,
Beijing, August 2010

lowing us to readily scale to large problems with extensive prior knowledge, as demonstrated by our experiments.

We next discuss related work, followed by a more in-depth description of the fact-finding algorithms used in our experiments, including three novel, high-performing algorithms: Average·Log, Investment, and PooledInvestment. We then present the framework's mechanics and the translation of first-order logic into a linear program. Finally, we present our experimental setup and results over three domains chosen to illustrate different aspects of the framework, demonstrating that both our new fact-finders and our framework offer performance improvements over the current state of the art.

2 Related Work

The broader field of trust can be split into three areas of interest[1]: theoretical, reputation-based, and information-based.

2.1 Theoretical

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 et al., 1999), an extension of propositional logic permitting [0,1] belief over propositions. Yu and 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 et al. (2006) uses the related subjective logic (Josang, 1997). While our belief in a claim is decidedly Bayesian (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.

2.2 Reputation-based

Reputation-based systems determine an entity's trust or standing among peers via transitive recommendations, as PageRank (Brin and Page, 1998) does among web pages, Advogato (Levien, 2008) does among people, and Eigentrust (Kamvar et al., 2003) does among peers in a network. Some, such as Hubs and Authorities (Kleinberg, 1999), are readily adapted to fact-finding, as demonstrated later.

2.3 Information-Based

Information-based approaches utilize content (rather than peer recommendations) to compute trust, and are often specialized for a particular domain. For example, (Zeng et al., 2006) and Wikitrust (Adler and de Alfaro, 2007) determine trust in a wiki's text passages from sequences of revisions but lack the claim-level granularity and general applicability of fact-finders.

Given a large set of sources making conflicting claims, fact-finders determine "the truth" by iteratively updating their parameters, calculating belief in facts based on the trust in their sources, and the trust in sources based on belief in their facts. TruthFinder (Yin et al., 2008) is a straightforward implementation of this idea. AccuVote (Dong et al., 2009a; Dong et al., 2009b) improves on this by using calculated source dependence (where one source derives its information from another) to give higher credibility to independent sources. (Galland et al., 2010)'s 3-Estimates algorithm incorporates the estimated "hardness" of a fact, such that knowing the answer to an easy question earns less trust than to a hard one. Except for AccuVote (whose model of repeated source-to-source copying is inapplicable to our experimental domains) we experimented over all of these algorithms.

3 Fact-Finding

We have a set of sources S each asserting a set of claims C_s, with $C = \bigcup_{s \in S} C_s$. Each claim $c \in C$ belongs to a *mutual exclusion set* $M_c \subseteq C$, a set of claims (including c) that are mutually exclusive with one another; for example, "John was born in 1960" and "John was born in 1965" are mutually exclusive because a person cannot be born in more than one year. If c is not mutually exclusive

[1] Following the division proposed by Artz and Gil (2007); see also (Sabater and Sierra, 2005) for a survey from a different perspective.

to any other claims, then $M_c = \{c\}$. Assuming there exists exactly one true claim $\overline{c}$ in each mutual exclusion set M, our goal is to predict $\overline{c}$ for each M, with accuracy measured by the number of successful predictions divided by the number of mutual exclusion sets, ignoring trivially correct claims that are the sole members of their mutual exclusion set. To this end, fact-finding algorithms iterate to find the trustworthiness of each source $T^i(s)$ at iteration i in terms of the belief in its claims in the previous iteration $B^{i-1}(C_s)$, and belief in each claim $B^i(c)$ in terms of $T^i(S_c)$, where $S_c = \{s : s \in S, c \in C_s\}$ is the set of all sources asserting c. Note that "trustworthiness" and "belief" as used within a fact-finding algorithm typically do not have meaningful semantics (i.e. they are not $[0, 1]$ Bayesian probabilities). Iteration continues until convergence or some predefined stop criteria.

3.1 Priors

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

3.2 Algorithms

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) and claims as authorities (with 0 hub score):

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

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 (discussed next); 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 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, 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. We used $\mathcal{G}(x) = x^g$ with $g = 1.2$ in our experiments, 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, 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 our experiments. Given $H^i(c) = \sum_{s\in S_c} \frac{T^i(s)}{|C_s|}$, we have:

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

3.3 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.

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

3.3.1 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 The Framework

To apply prior knowledge to a fact-finding algorithm, we translate the user's prior knowledge into a linear program. 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" beliefs $B^i(C)$ with the LP

4.1 Propositional Linear Programming

To translate prior knowledge into a linear program, we first propositionalize our first-order formulae into propositional logic (Russell and 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 our system is 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 $P(c)$.[2] 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 solve a global inference problem, e.g. (Roth and Yih, 2004; Roth and Yih, 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 $y_v \geq 0.5$ where

[2] 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.

$y_v \geq x_v \geq 1 - y_v$. 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, ..., c^n can be compactly written as $c_v^1 + c_v^2 + \ldots + c_v^n = 1$.

4.2 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 pseudo-probabilistic [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 $\omega_{idn}(x) = x$.

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 so 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:

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

Thus, the belief distribution found by our LP will 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 $x \geq f$.

4.3 From Values to Votes to Belief

Solving the LP gives us $[0, 1]$ values for each variable c_v, but we need to calculate an updated belief $B(c)$. We propose 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: $\omega_{idn}^{-1}(x) = x$ and $\omega_{inv}^{-1}(x) = 1 - (1 + y)^{-1}$. 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 just slightly better on average and used it for all reported results.

4.4 "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 $Age(John, 35) \Rightarrow Age(Tom, 40) \lor Age(Tom, Unknown)$. 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.

5 Experiments

Experiments were conducted over three domains (city population, basic biographies, and American vs. British spelling) with four datasets, all using the VoteDistance cost function and Vote Loss vote redistribution. We fixed the number of iterations of the framework (calculating $T^i(S)$, $B^i(S)$ and then solving the LP) at 20, which was found sufficient for all fact-finders. To evaluate accuracy, after the final iteration we look at each mutual exclusion set M and predict the highest-belief claim $c \in M$ (or, if u_M had the highest belief, the second-highest claim), breaking ties randomly, and check that it is the true claim t_M. We omit any M that does not contain a true claim (all known claims are false) and any M that is trivially correct (containing only one claim other than u_M). All results are shown in Table 1. **Vote** is the baseline, choosing either the claim occurring in the most Wikipedia revisions (in the Pop dataset) or claimed by the most sources (for all other datasets). **Sum** is Sums (Hubs and Authorities), **3Est** is 3-Estimates, $\mathbf{TF^s}$ is simplified TruthFinder, $\mathbf{TF^c}$ is "full" TruthFinder, **A·L** is Average·Log, $\mathbf{Inv}^{1.2}$ is Investment with $g = 1.2$, and $\mathbf{Pool}^{1.4}$ is PooledInvestment with $g = 1.4$.

5.1 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 et al., 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.

5.2 Wikipedia Infoboxes

To focus on the performance of the framework, we (like previous fact-finding work) naively assume that our data are accurately extracted, but we also require large corpora. Wikipedia Infoboxes (Wu and Weld, 2007) are a semi-structured source covering many domains with readily available authorship, and we produced our city population and basic biographic datasets from the most recent full-history dump of the English Wikipedia (taken January 2008). However, attribution is difficult: if an author edits the page but not the claim within the infobox, is the author implicitly agreeing with (and asserting) the claim? The best performance was achieved by being strict for City Population data, counting only the direct editing of a claim, and lax for Biography data, counting any edit. We hypothesize this is because editors may lack specific knowledge about a city's population (and thus fail to correct an erroneous value) but incorrect birth or death dates are more noticeable.

5.3 Results

5.3.1 City Population

We collected infoboxes for settlements (Geobox, Infobox Settlement, Infobox City, etc.) to obtain 44,761 populations claims qualified by year (e.g. $pop(Denver, 598707, 2008)$), with 4,107 authors total. We took as our "truth" U.S. census data, which gave us 308 non-trivial true facts to test against. Our "common sense" knowledge is that population grows

Table 1: Experimental Results ($\emptyset$ indicates no prior knowledge; all values are percent accuracy) Some results are omitted here (see text). A·L, Inv$^{1.2}$, Pool$^{1.4}$ are our novel algorithms

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

over time ("Growth" in table 1); 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.

5.3.2 Synthetic City Population

What if attribution were certain and the data more dense? To this end we created a synthetic dataset. 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 tried 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.

5.3.3 Basic Biographies

We scanned infoboxes to find 129,847 claimed birth dates, 34,201 death dates, 10,418 parent-child pairs, and 9,792 spouses. 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. 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.

5.3.4 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 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$^{200}_{IBT}$).

6 Conclusion

We have introduced a new framework for incorporating prior knowledge into a fact-finding system, along with several new high-performing fact-finding algorithms (Investment, PooledInvestment, and Average·Log). While the benefits of prior knowledge were most dramatic in the Spelling domain, we saw gains from both "common sense" and specific knowledge in all experiments–even the difficult Biography domain saw faster convergence with common sense alone and notably higher results when specific knowledge was added. We find that while prior knowledge is helpful in reducing error, when the user's viewpoint disagrees with the norm it becomes absolutely essential and, formulated as a linear program, it need not be the computational burden that might otherwise be expected.

Acknowledgements

This research was partly sponsored by the Army Research Laboratory (ARL) (accomplished under Cooperative Agreement Number W911NF-09-2-0053). Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the ARL.

References

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

Artz, D and Y Gil. 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, June.

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

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

Dong, X.L., L. Berti-Equille, and Divesh Srivastava. 2009b. Truth discovery and copying detection in a dynamic world. *VLDB*, 2(1):562–573.

Galland, Alban, Serge Abiteboul, A. Marian, and Pierre Senellart. 2010. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 131–140. ACM.

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

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

Kamvar, S, M Schlosser, and H Garcia-molina. 2003. The Eigentrust algorithm for reputation management in P2P networks. *WWW '03*.

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.

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

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

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

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

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

Roth, Dan and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8.

Roth, D and W Yih. 2007. Global Inference for Entity and Relation Identification via a Linear Programming Formulation. In Getoor, Lise and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Russell, Stuart and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition.

Sabater, Jordi and Carles Sierra. 2005. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60, September.

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

Wu, Fei and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 41.

Yin, Xiaoxin, Philip S. Yu, and Jiawei Han. 2008. Truth Discovery with Multiple Conflicting Information Providers on the Web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808.

Yu, Bin and Munindar P. Singh. 2003. Detecting deception in reputation management. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS '03*, page 73.

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