Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources

Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang
Wilko Horn, Camillo Lugaresi, Shaohua Sun, Wei Zhang
Google Inc.
{lunadong|gabr|kpmurphy|vandang|wilko|camillol|sunsh|weizh}@google.com

Abstract

The quality of web sources has been traditionally evaluated using exogenous *signals such as the hyperlink structure of the graph. We propose a new approach that relies on* endogenous *signals such as the correctness of factual information provided by the source: a source that has few false facts is considered to be trustworthy. The facts are automatically extracted from each source by information extraction methods commonly used to construct knowledge bases. We propose a way to distinguish errors made in the extraction process from factual errors in the web source per se, by using joint inference in a novel multi-layer probabilistic model. We call the trustworthiness score we computed* Knowledge-Based Trust (KBT). *We apply our method to a database of 2.8B facts extracted from the web, and thereby estimate the trustworthiness of 119M webpages. Manual evaluation of a subset of the results confirms the effectiveness of the method.*

1 Introduction

"Learning to trust is one of life's most difficult tasks." – Isaac Watts.

Quality assessment for web sources (specific webpages, such as `wiki.com/page1`, or whole websites, such as `wiki.com`) is of tremendous importance in web search. It has been traditionally evaluated using exogenous signals such as hyperlinks and browsing history. However, such signals mostly capture how popular a web source is. For example, the gossip websites listed in [16] mostly have high PageRank scores [4], but would not generally be considered trustworthy. Conversely, some less popular websites nevertheless have very trustworthy information.

In this paper, we address the fundamental question of estimating how trustworthy a given web source is. Informally, we define the trustworthiness or *accuracy* of a web source as the probability that it provides the correct value for a fact (such as Barack Obama's nationality), assuming that it mentions any value for that fact. (There can be other endogenous signals such as the *completeness* and *freshness* of a web source; they are orthogonal to the accuracy measure and out of the scope of this paper.)

We propose using *Knowledge-Based Trust (KBT)* to estimate source trustworthiness as follows. We extract a plurality of facts from many pages using information extraction techniques. We then jointly estimate the

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

correctness of these facts and the accuracy of the sources using inference in a probabilistic model. Inference is an iterative process, since we believe a source is trustworthy if its facts are correct, and we believe the facts are correct if they are extracted from a trustworthy source. We leverage the redundancy of information on the web to break the symmetry. Furthermore, we show how to initialize our estimate of the accuracy of sources based on authoritative information, in order to ensure that this iterative process converges to a good solution.

The fact extraction process we use is based on the *Knowledge Vault* (KV) project [9]. KV uses 16 different information extraction systems to extract (subject, predicate, object) *knowledge triples* from webpages. An example of such a triple is *(Barack Obama, nationality, USA)*. A subject represents a real-world entity, identified by an ID such as *mid*s in *Freebase* [2]; a predicate is pre-defined in *Freebase*, describing a particular attribute of an entity; an object can be an entity, a string, a numerical value, or a date.

The facts extracted by automatic methods such as KV may be wrong. There are two main sources of errors: incorrect facts on a page, and incorrect extractions made by an extraction system. As shown in [10], extraction errors are far more prevalent than source errors. Ignoring this distinction can cause us to incorrectly distrust a web source.

The KBT method introduced in this paper uses a novel multi-layer probabilistic model to distinguish errors made in the extraction process from factual errors in the web source per se. This provides a much more accurate estimate of the source reliability. We propose an efficient, scalable algorithm for performing inference and parameter estimation in the proposed probabilistic model (Section 3). We applied our model to 2.8 billion triples extracted from the web, and were thus able to reliably predict the trustworthiness of 119 million webpages and 5.6 million websites (Section 4).

We note that source trustworthiness provides an additional signal for evaluating the quality of a web source. We discuss new research opportunities for improving it and using it in conjunction with existing signals such as PageRank (Section 4.4). Also, we note that although we present our methods in the context of knowledge extraction, the general approach we propose can be applied to many other tasks that involve data integration and data cleaning. This paper is a summary of the paper [11].

2 Problem Definition and Overview

This section formally defines the notion of *Knowledge-Based Trust* (KBT) and gives an overview of our approach.

Input: We are given a set of web sources $\mathcal{W}$ and a set of extractors $\mathcal{E}$. An extractor is a method for extracting `(subject, predicate, object)` triples from a webpage. For example, one extractor may look for the *pattern "*$A*, the president of* $B*, ..."*, from which it can extract the triple *(*$A*, nationality,* $B*)*. Certainly, this is not always correct (*e.g.*, if A is the president of a company, not a country). In addition, an extractor reconciles the string representations of entities into entity identifiers such as Freebase mids, and sometimes this fails too. The different extractors can apply different extraction techniques on different types of data (*e.g.*, Web texts, DOM trees, and Webtables), and each of them may use a large number of different patterns; details of the extractors we used in KV can be found in [10]. In the rest of the paper, we represent such triples as (data item, value) pairs, where the data item is in the form of `(subject, predicate)`, describing a particular aspect of an entity, and the `object` serves as a value for the data item.

We define an observation variable X_{ewdv}. We set $X_{ewdv} = 1$ if extractor e extracted value v for data item d on web source w; if it did not extract such a value, we set $X_{ewdv} = 0$. We use matrix $X = \{X_{ewdv}\}$ to denote all the data. We can represent X as a (sparse) "data cube", as shown in Figure 1, where each cell gives the values extracted by an extractor from a web source on a particular data item. Table 1 shows an example of a single horizontal "slice" of this cube for the case where the data item is d^* *= (Barack Obama, nationality).* We discuss this example in more detail next.

Table 1: Obama's nationality extracted by 5 extractors from 8 webpages. Column 2 (Value) shows the nationality truly provided by each source; Columns 3-7 show the nationality extracted by each extractor. Wrong extractions are shown in italics.

	Value	E_1	E_2	E_3	E_4	E_5
W_1	USA	USA	USA	USA	USA	*Kenya*
W_2	USA	USA	USA	USA	*N.Amer.*	
W_3	USA	USA		USA	*N. Amer.*	
W_4	USA	USA		USA	*Kenya*	
W_5	Kenya	Kenya	Kenya	Kenya	Kenya	Kenya
W_6	Kenya	Kenya		Kenya	*USA*	
W_7	-			*Kenya*		*Kenya*
W_8	-					*Kenya*

Example 1: Suppose we have 8 webpages, $W_1 - W_8$, and suppose we are interested in the data item *(Obama, nationality)*. It is widely believed that Obama has nationality USA, not any other country.

The value stated for this data item by each of the webpages is shown in the left hand column of Table 1. We see that $W_1 - W_4$ provide *USA* as the nationality of Obama, whereas $W_5 - W_6$ provide *Kenya* (a false value)[1]. Pages $W_7 - W_8$ do not provide any information regarding Obama's nationality.

Now suppose we have 5 different extractors of varying reliability. The values they extract for this data item from each of the 8 webpages are shown in the table. Extractor E_1 extracts all the provided triples correctly. Extractor E_2 misses some of the provided triples (false negatives), but all of its extractions are correct. Extractor E_3 extracts all the provided triples, but also wrongly extracts the value *Kenya* from W_7, even though W_7 does not provide this value (a false positive)[2]. Extractor E_4 and E_5 both have poor quality, missing a lot of provided triples and making numerous mistakes.

Knowledge-based trust (KBT): For each web source $w \in \mathcal{W}$, we define its *accuracy*, denoted by A_w, as the probability that a value it provides for a fact is correct (*i.e.*, consistent with the real world). We use $A = \{A_w\}$ for the set of all accuracy parameters. We now formally define the problem of KBT estimation.

Definition 1 (KBT Estimation): The *Knowledge-Based Trust (KBT) estimation task* is to estimate the web source accuracies $A = \{A_w\}$ given the observation matrix $X = \{X_{ewdv}\}$ of extracted triples.

Our solution: To precisely estimate KBT, it is critical to distinguish extraction errors from source errors. Simply assuming all extracted values are actually provided by the source obviously would not work. In our example, we may wrongly infer that W_1 is a bad source because of the extracted *Kenya* value, although this is an extraction error. Instead, we wish to distinguish correctly extracted true triples (*e.g.*, *USA* from $W_1 - W_4$), correctly extracted false triples (*e.g.*, *Kenya* from $W_5 - W_6$), wrongly extracted true triples (*e.g.*, *USA* from W_6), and wrongly extracted false triples (*e.g.*, *Kenya* from $W_1, W_4, W_7 - W_8$).

In this paper, we present a new probabilistic model that can estimate the accuracy of each web source, factoring out the noise introduced by the extractors. We consider two sets of latent variables, one set representing the true value of each data item, and the other representing whether each extraction was correct or not; this allows us to distinguish extraction errors and source data errors. In addition, we define a set of parameters for the accuracy of the web sources (A), and a set of parameters for the quality of the extractors (we formally define them in Section 3.1); this allows us to separate the quality of the sources from that of the extractors. We call the new model the *multi-layer model*, because it contains two layers of latent variables and parameters.

[1]Real example can be found at http://beforeitsnews.com/obama-birthplace-controversy/2013/04/alabama-supreme-court-chief-justice-roy-moore-to-preside-over-obama-eligibility-case-2458624.html.

[2]As an example, KV made such a wrong extraction from webpage http://www.monitor.co.ug/News/National/US+will+respect+winner+of+Kenya+election++Obama+says/-/688334/1685814/-/ksxagx/-/index.html.

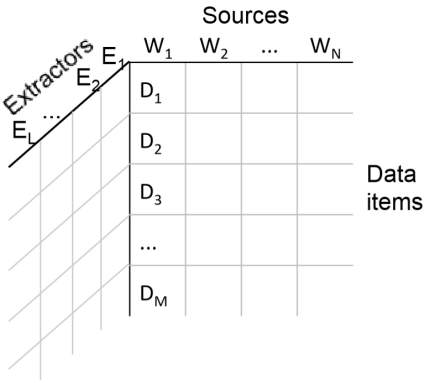

Figure 1: Form of the input data.

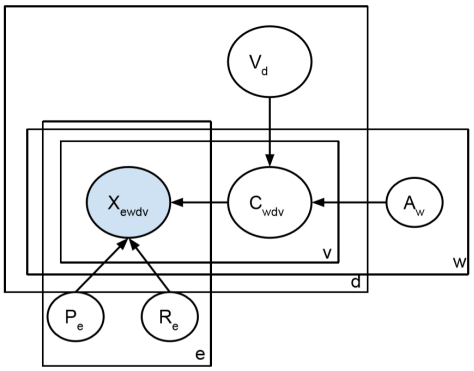

Figure 2: A representation of the multi-layer model using graphical model plate notation.

3 Multi-Layer Model

In this section, we describe in detail how we compute $A = \{A_w\}$ from our observation matrix $X = \{X_{ewdv}\}$ using a multi-layer model.

3.1 The multi-layer model

We assume that each data item can only have a single true value. This assumption holds for functional predicates, such as *date-of-birth*, but is not technically valid for set-valued predicates, such as *child*. Nevertheless, [10] showed empirically that this "single truth" assumption works well in practice even for non-functional predicates, especially when the input data contain a lot of noises; thus, we adopt it in this work for simplicity and we can extend it for multi-valued attributes by applying approaches in [21, 27, 33]. Based on the single-truth assumption, we define a latent variable $V_d \in \text{dom}(d)$ for each data item d to present the true value for d, where $\text{dom}(d)$ is the domain (set of possible values) for data item d.

We introduce the binary latent variables C_{wdv}, which represent whether web source w actually provides triple (d, v) or not. These variables depend on the true values V_d and the accuracy of each of the web sources A_w as follows:

$$p(C_{wdv} = 1 | V_d = v^*, A_w) = \begin{cases} A_w & \text{if } v = v^* \\ \frac{1-A_w}{n} & \text{if } v \neq v^* \end{cases} \tag{7}$$

where v^* is the true value, and n is the number of false values for this domain (*i.e.*, we assume $|\text{dom}(d)| = n+1$). The model says that the probability for w to provide a true value v^* for d is its accuracy, whereas the probability for it to provide one of the n false values is $1 - A_w$ divided by n (as in [7], we assume uniform distribution of the false values).

The likelihood of an observed extraction depends on how likely the extractor extracts a truly provided triple and how likely it extracts an unprovided triple. Following [27, 33], we use a two-parameter noise model for the observed data, as follows:

$$p(X_{ewdv} = 1 | C_{wdv} = c, Q_e, R_e) = \begin{cases} R_e & \text{if } c = 1 \\ Q_e & \text{if } c = 0 \end{cases} \tag{8}$$

Here R_e is the *recall* of the extractor; that is, the probability of extracting a truly provided triple. And Q_e is 1 minus the *specificity*; that is, the probability of extracting an unprovided triple. Parameter Q_e is related to the

Table 2: Extraction correctness and value truthfulness for the data in Table 1. Columns 2-4 show $p(C_{wdv} = 1|X_{wdv})$, and the last row shows $p(V_d|\hat{C}_d)$ (note that this distribution does not sum to 1.0, since not all of the values are shown in the table).

	USA	Kenya	N.Amer.
W_1	1	0	-
W_2	1	-	0
W_3	1	-	0
W_4	1	0	-
W_5	-	1	-
W_6	0	1	-
W_7	-	.07	-
W_8	-	0	-
$p(V_d\|\hat{C}_d)$	.995	.004	0

recall (R_e) and precision (P_e) as follows:

$$Q_e = \frac{\gamma}{1-\gamma} \cdot \frac{1-P_e}{P_e} \cdot R_e \tag{9}$$

where $\gamma = p(C_{wdv} = 1)$ is the prior probability for any $v \in \text{dom}(d)$, as explained in [27].

To complete the specification of the model, we must specify the prior probability of the various model parameters:

$$\theta_1 = \{A_w\}_{w=1}^{W}, \theta_2 = (\{P_e\}_{e=1}^{E}, \{R_e\}_{e=1}^{E}), \theta = (\theta_1, \theta_2) \tag{10}$$

For simplicity, we use uniform priors on the parameters. By default, we set $A_w = 0.8$, $R_e = 0.8$, and $Q_e = 0.2$. In Section 4, we discuss an alternative way to estimate the initial value of A_w, based on the fraction of correct triples that have been extracted from this source, using an external estimate of correctness (based on *Freebase* [2]).

Let $V = \{V_d\}$, $C = \{C_{wdv}\}$, and $Z = (V, C)$ be all the latent variables. Our model defines the following joint distribution:

$$p(X, Z, \theta) = p(\theta)p(V)p(C|V, \theta_1)p(X|C, \theta_2) \tag{11}$$

We can represent the conditional independence assumptions we are making using a graphical model, as shown in Figure 2. The shaded node is an observed variable, representing the data; the unshaded nodes are hidden variables or parameters. The arrows indicate the dependence between the variables and parameters. The boxes are known as "plates" and represent repetition of the enclosed variables; for example, the box of e repeats for every extractor $e \in \mathcal{E}$.

As an example, Table 2 shows the probabilities computed for the latent variables V and C. The multi-layer model is able to decide that *USA* is likely to be true and is likely to be provided by $W_1 - W_4$, contributing positively to their trustworthiness. On the other hand, *Kenya* is likely to be false and is likely to be provided by $W_5 - W_6$, contributing negatively to their trustworthiness. We describe next how we may compute these probabilities.

3.2 Inference

Recall that estimating KBT essentially requires us to compute the posterior over the parameters of interest, $p(A|X)$. Doing this exactly is computationally intractable, because of the presence of the latent variables Z. One approach is to use a Monte Carlo approximation, such as Gibbs sampling, as in [32]. However, this can be

Algorithm 1 MULTILAYER(X, t_{max})

Input: X: all extracted data;
Output: Estimates of Z and θ.

```
Initialize θ to default values;
for t ∈ [1, t_max] do
    Estimate C according to Eq.(8);
    Estimate V according to Eq.(7);
    Estimate θ_1 by Eq.(15);
    Estimate θ_2 by Eqs.(12-13);
    if Z, θ converge then
        break;
    end if
end for
return Z, θ;
```

slow and is hard to implement in a Map-Reduce framework, which is required for the scale of data we use in this paper.

A faster alternative is to use EM, which will return a point estimate of all the parameters, $\hat{\theta} = \text{argmax} p(\theta|X)$. Since we are using a uniform prior, this is equivalent to the maximum likelihood estimate $\hat{\theta} = \text{argmax} p(X|\theta)$. From this, we can derive $\hat{A}$.

As pointed out in [26], an exact EM algorithm has a quadratic complexity even for a single-layer model, so is unaffordable for data of web scale. Instead, we use an iterative "EM like" estimation procedure, where we initialize the parameters as described previously, and then alternate between estimating Z and then estimating θ, until we converge.

We next give an overview of this EM-like algorithm. Algorithm 1 gives a summary of the pseudo code; the details can be found in [11].

In our case, Z consists of two "layers" of variables. We update them sequentially, as follows. First, let $X_{wdv} = \{X_{ewdv}\}$ denote all extractions from web source w about a particular triple $t = (d, v)$. We compute by Bayesian analysis the extraction correctness $p(C_{wdv}|X_{wdv}, \theta_2^t)$, which is our guess about the "true contents" of each web source. This can be done in parallel over d, w, v.

Let $\hat{C}_d = \hat{C}_{wdv}$ denote all the estimated values for d across the different web sources. We then compute by Bayesian analysis $p(V_d|\hat{C}_d, \theta_1^t)$, which is our guess about the "true value" of each data item. This can be done in parallel over d.

Having estimated the latent variables, we then estimate θ^{t+1}. This parameter update also consists of two steps (but can be done in parallel): estimating the source accuracies $\{A_w\}$ and the extractor reliabilities $\{P_e, R_e\}$, as explained next.

3.3 Estimating the quality parameters

For reasons explained in [27], it is much more reliable to estimate P_e and R_e from data, and then compute Q_e using Equation (9), rather than trying to estimate Q_e directly. According to the definition of precision and recall, we can estimate them as follows:

$$\hat{P}_e = \frac{\sum_{wdv:X_{ewdv}=1} p(C_{wdv}=1|X)}{\sum_{wdv:X_{ewdv}=1} 1} \quad (12)$$

$$\hat{R}_e = \frac{\sum_{wdv:X_{ewdv}=1} p(C_{wdv}=1|X)}{\sum_{wdv} p(C_{wdv}=1|X)} \quad (13)$$

Following [7], we estimate the accuracy of a source by computing the average probability of its provided values being true:

$$\hat{A}_w = \frac{\sum_{dv:\hat{C}_{wdv}=1} p(V_d = v|X)}{\sum_{dv:\hat{C}_{wdv}=1} 1} \tag{14}$$

We can take uncertainty of $\hat{C}$ into account as follows:

$$\hat{A}_w = \frac{\sum_{dv:\hat{C}_{wdv}>0} p(C_{wdv} = 1|X)p(V_d = v|X)}{\sum_{dv:\hat{C}_{wdv}>0} p(C_{wdv} = 1|X)} \tag{15}$$

Eq. (15) is the key equation behind Knowledge-based Trust estimation: it estimates the accuracy of a web source as the weighted average of the probability of the facts that it contains (provides), where the weights are the probability that these facts are indeed contained in that source.

4 Experimental Results

This section describes our experimental results on large-scale real-world data. We show that (1) our algorithm can effectively estimate the correctness of extractions and the truthfulness of triples; and (2) KBT provides a valuable additional signal for web source quality.

We implemented the multi-layer model described in Section 3, called MULTILAYER. We initialized the source quality according to a gold standard, as we shall describe shortly. We note that such quality initialization is not required for MULTILAYER, but did significantly improve the predictions (see [11] for detailed comparison).

4.1 Data set

We experimented with knowledge triples collected by Knowledge Vault [9] on 7/24/2014; for simplicity we call this data set *KV*. There are 2.8B triples extracted from 2B+ webpages by 16 extractors, involving 40M extraction patterns (many extractors each learns and applies a large number of extraction patterns). Implementation details for KV are described in [9].

Figure 3 shows the distribution of the number of distinct extracted triples per URL (*i.e.*, webpage) and per extraction pattern. On the one hand, we observe some huge sources and extractors: 26 URLs each contributes over 50K triples (a lot due to extraction mistakes), 15 websites each contributes over 100M triples, and 43 extraction patterns each extracts over 1M triples. On the other hand, we observe long tails: 74% URLs each contributes fewer than 5 triples, and 48% extraction patterns each extracts fewer than 5 triples.

To determine whether these triples are true or not (gold standard labels), we use two methods. The first method is called the *Local-Closed World Assumption (LCWA)* [9, 10, 15] and works as follows. A triple (s, p, o) is considered as `true` if it appears in the Freebase KB. If the triple is missing from the KB but (s, p) appears for any other value o', we assume the KB is locally complete (for (s, p)), and we label the (s, p, o) triple as `false`. We label the rest of the triples (where (s, p) is missing) as `unknown` and remove them from the evaluation set. In this way we can decide truthfulness of 0.74B triples (26% in *KV*), of which 20% are true (in Freebase).

Second, we apply type checking to find incorrect extractions. We found that in the following cases a triple (s, p, o) is often due to an extraction error: 1) $s = o$; 2) the type of s or o is incompatible with what is required by the predicate; or 3) o is outside the expected range (*e.g.*, the weight of an athlete is over 1000 pounds). We discovered 0.56B triples (20% in KV) that violate such rules and consider them both as `false` triples and as extraction mistakes.

Our gold standard include triples from both labeling methods. It contains in total 1.3B triples, among which 11.5% are true.

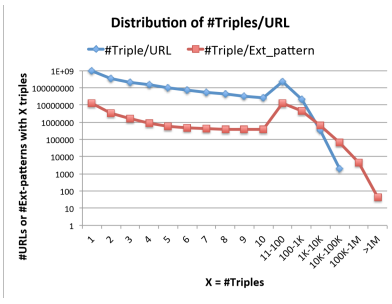

Figure 3: Distribution of #Triples per URL or extraction pattern shows a high variety of the sources and extractors.

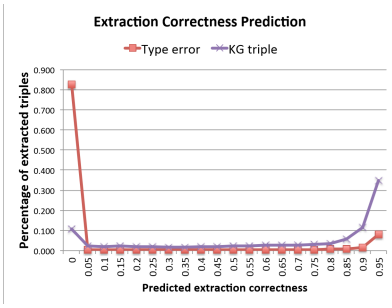

Figure 4: Distribution of predicted extraction correctness shows effectiveness of MULTILAYER.

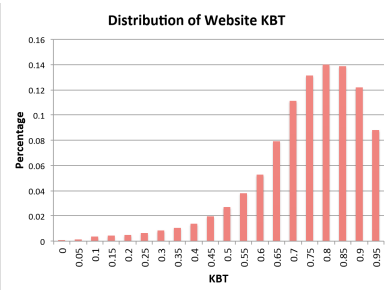

Figure 5: Distribution on KBT for websites with at least 5 extracted triples.

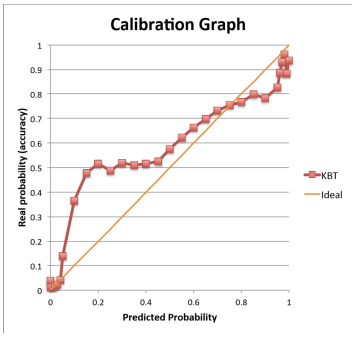

Figure 6: MULTILAYER predicts well-calibrated probabilities.

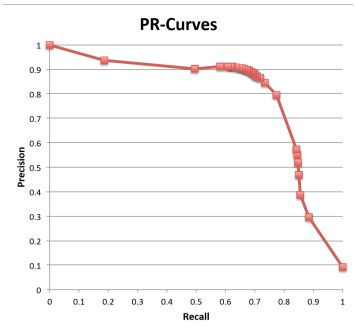

Figure 7: The PR-curve of MULTILAYER results is in good shape.

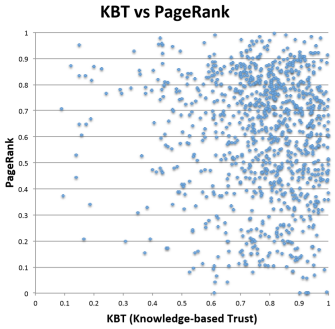

Figure 8: KBT and PageRank are orthogonal signals.

4.2 Correctness of triples and extractions

We divide our triples according to the predicted probabilities into buckets $[0, 0.01), \ldots, [0.04, 0.05), [0.05, 0.1), \ldots, [0.9, 0.95), [0.95, 0.96), \ldots, [0.99, 1), [1, 1]$ (most triples fall in $[0, 0.05)$ and $[0.95, 1]$, so we used a finer granularity there). For each bucket we compute the accuracy of the triples according to the gold standard, which can be considered as the real probability of the triples. Ideally, the predicted probabilities should be the same as the real probabilities. Figure 6 plots the calibration curve, showing that MULTILAYER computes well-calibrated probabilities for the truthfulness of triples. In addition, Figure 7 plots the PR-curve, where the X-axis represents the recall and the Y-axis represents the precision as we order triples according to the predicted probabilities. The PR-curve is also in good shape.

To examine the quality of our prediction on extraction correctness (recall that we lack a full gold standard), we plotted the distribution of the predictions on triples with type errors (ideally we wish to predict a probability of 0 for such extractions) and on correct triples (presumably a lot of them, though not all, would be correctly extracted and we shall predict a high probability for such extractions). Figure 4 shows the results by MULTILAYER. We observe that for the triples with type errors, MULTILAYER predicts a probability below 0.1 for 80% of them and a probability above 0.7 for only 8%; in contrast, for the correct triples in Freebase, MULTILAYER predicts a probability below 0.1 for 26% of them and a probability above 0.7 for 54%, showing effectiveness of our model.

4.3 KBT vs PageRank

We now evaluate how well we estimate the trustworthiness of web sources. Our data set contains 2B+ webpages from 26M websites. Among them, our multi-layer model believes that we have correctly extracted at least 5 triples from about 119M webpages and 5.6M websites. Figure 5 shows the distribution of KBT scores: we observed that the peak is at 0.8 and 52% of the websites have a KBT over 0.8.

Since we do not have ground truth on web-source quality, we compare our method to PageRank. We compute PageRank for all websites on the web, and normalize the scores to $[0, 1]$. Figure 8 plots KBT and PageRank for 2000 randomly selected websites. As expected, the two signals are almost orthogonal. We next investigate the two cases where KBT differs significantly from PageRank.

Low PageRank but high KBT (bottom-right corner): To understand which sources may obtain high KBT, we randomly sampled 100 websites whose KBT is above 0.9. The number of extracted triples from each website varies from hundreds to millions. For each website we considered the top 3 predicates and randomly selected from these predicates 10 triples where the probability of the extraction being correct is above 0.8. We manually evaluated each website according to the following 4 criteria.

- *Triple correctness*: whether at least 9 triples are correct.
- *Extraction correctness*: whether at least 9 triples are correctly extracted (and hence we can evaluate the website according to what it really states).
- *Topic relevance*: we decide the major topics for the website according to the website name and the introduction in the "About us" page; we then decide whether at least 9 triples are relevant to these topics (*e.g.*, if the website is about business directories in South America but the extractions are about cities and countries in SA, we consider them as not topic relevant).
- *Non-trivialness*: we decide whether the sampled triples state non-trivial facts (*e.g.*, if most sampled triples from a Hindi movie website state that the language of the movie is Hindi, we consider it as trivial).

We consider a website as truly trustworthy if it satisfies all of the four criteria. Among the 100 websites, 85 are considered trustworthy; 2 are not topic relevant, 12 do not have enough non-trivial triples, and 2 have more than 1 extraction errors (one website has two issues). However, only 20 out of the 85 trustworthy sites have a PageRank over 0.5. This shows that KBT can identify sources with trustworthy data, even though they are tail sources with low PageRanks.

High PageRank but low KBT (top-left corner): We consider the 15 gossip websites listed in [16]. Among them, 14 have a PageRank among top 15% of the websites, since such websites are often popular. However, for all of them the KBT are in the bottom 50%; in other words, they are considered less trustworthy than half of the websites. Another kind of websites that often get low KBT are forum websites. For instance, we discovered that *answers.yahoo.com* says that *"Catherine Zeta-Jones is from New Zealand"* [3], although she was born in Wales according to *Wikipedia*[4].

4.4 Discussions

Although we have seen that KBT seems to provide a useful signal about trustworthiness, which is orthogonal to more traditional signals such as PageRank, our experiments also show places for further improvement as future work.

1. To avoid evaluating KBT on topic irrelevant triples, we need to identify the main topics of a web source, and filter triples whose entity or predicate is not relevant to these topics.

[3]https://answers.yahoo.com/question/index?qid=20070206090808AAC54nH.
[4]http://en.wikipedia.org/wiki/Catherine_Zeta-Jones.

2. To avoid evaluating KBT on trivial triples, we need to decide whether the information in a triple is trivial. One possibility is to consider a predicate with a very low variety of objects as less informative. Another possibility is to associate triples with an IDF (inverse document frequency), such that low-IDF triples get lower weight in KBT computation.
3. Our extractors (and most state-of-the-art extractors) still have limited extraction capabilities and this limits our ability to estimate KBT for all websites. We wish to increase our KBT coverage by extending our method to handle open-IE style information extraction techniques, which do not conform to a schema [14]. However, although these methods can extract more triples, they may introduce more noise.
4. Some websites scrape data from other websites. Identifying such websites requires techniques such as copy detection. Scaling up copy detection techniques, such as [6, 7], has been attempted in [23], but more work is required before these methods can be applied to analyzing extracted data from billions of web sources.
5. Finally, there have been many other signals such as PageRank, visit history, spaminess for evaluating web-source quality. Combining KBT with those signals would be important future work.

5 Related Work

There has been a lot of work studying how to assess quality of web sources. PageRank [4] and Authority-hub analysis [19] consider signals from link analysis (surveyed in [3]). EigenTrust [18] and TrustMe [28] consider signals from source behavior in a P2P network. Web topology [5], TrustRank [17], and AntiTrust [20] detect web spams. The knowledge-based trust we propose in this paper is different from all of them in that it considers an important *endogenous* signal–the correctness of the factual information provided by a web source.

KBT estimation is closely related to the *knowledge fusion* problem [10], where the goal is to decide the true (but latent) values for each of the data items, given the noisy extraction. It is also relevant to the body of work in *Data fusion* (surveyed in [1, 12, 23]), where the goal is to resolve conflicts from data provided by multiple sources, but assuming perfect knowledge on the values provided by each source (so no extraction error). Most of the recent work in this area considers trustworthiness of sources, measured by link-based measures [24, 25], IR-based measures [29], accuracy-based measures [7, 8, 13, 22, 27, 30], and graphical-model analysis [26, 31, 33, 32]. However, these papers do not model the concept of an extractor, and hence they cannot distinguish an untrustworthy source from a low-quality extractor.

Graphical models have been proposed to solve the data fusion problem [26, 31, 32, 33]. In particular, [26] considers single truth, [32] considers numerical values, [33] allows multiple truths, and [31] considers correlations between the sources. These prior works do not model the concept of an extractor, and hence they cannot capture the fact that sources and extractors introduce qualitatively different kinds of noise. In addition, the data sets used in the experiments of traditional data fusion works are typically 5-6 orders of magnitude smaller in scale than ours, and their inference algorithms are inherently slower than our algorithm.

6 Conclusions

This paper proposes a new metric for evaluating web-source quality–knowledge-based trust. We proposed a sophisticated probabilistic model that jointly estimates the correctness of extractions and source data, and the trustworthiness of sources. In addition, we presented an algorithm that dynamically decides the level of granularity for each source. Experimental results have shown both promise in evaluating web-source quality and improvement over existing techniques for knowledge fusion.

References

[1] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys*, 41(1):1–41, 2008.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[3] A. Borodin, G. Roberts, J. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *TOIT*, 5:231–297, 2005.

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

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *SIGIR*, 2007.

[6] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 2010.

[7] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1), 2009.

[8] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1), 2009.

[9] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.

[10] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *PVLDB*, 2014.

[11] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based trust: Estimating the trustworthiness of web sources. *PVLDB*, 2015.

[12] X. L. Dong and F. Naumann. Data fusion–resolving data conflicts for integration. *PVLDB*, 2009.

[13] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6, 2013.

[14] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: the second generation. In *IJCAI*, 2011.

[15] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422, 2013.

[16] Top 15 most popular celebrity gossip websites. http://www.ebizmba.com/articles/gossip-websites, 2014.

[17] Z. Gyngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *VLDB*, pages 576–587, 2014.

[18] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *WWW*, 2003.

[19] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, 1998.

[20] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *AIRWeb*, 2006.

[21] F. Li, X. L. Dong, A. Langen, and Y. Li. Knowledge verification for long tail verticals. Technical report, 2016. `www.comp.nus.edu.sg/~furongli/factCheck_report.pdf`.

[22] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD*, pages 1187–1198, 2014.

[23] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava. Scaling up copy detection. In *ICDE*, 2015.

[24] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *COLING*, pages 877–885, 2010.

[25] J. Pasternack and D. Roth. Making better informed trust decisions with generalized fact-finding. In *IJCAI*, pages 2324–2329, 2011.

[26] J. Pasternack and D. Roth. Latent credibility analysis. In *WWW*, 2013.

[27] R. Pochampally, A. D. Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *Sigmod*, 2014.

[28] A. Singh and L. Liu. TrustMe: anonymous management of trust relationshiops in decentralized P2P systems. In *IEEE Intl. Conf. on Peer-to-Peer Computing*, 2003.

[29] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *Proc. of the WebDB Workshop*, 2007.

[30] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.

[31] X. Yin and W. Tan. Semi-supervised truth discovery. In *WWW*, pages 217–226, 2011.

[32] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. In *QDB*, 2012.

[33] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian approach to discovering truth from conflicting sources for data integration. *PVLDB*, 5(6):550–561, 2012.