© 2014 Shi Zhi

INTEGRATING MULTIPLE CONFLICTING SOURCES BY TRUTH DISCOVERY AND SOURCE QUALITY ESTIMATION

BY

SHI ZHI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Professor Jiawei Han

ABSTRACT

Multiple descriptions about the same entity from different sources will inevitably result in data or information inconsistency. Among conflicting pieces of information, which one is the most trustworthy? How to detect the fraudulence of a rumor? Obviously, it is unrealistic to curate and validate the trustworthiness of every piece of information because of the high cost of human labeling and lack of experts. To find the truth of each entity, much research work has shown that considering the quality of information providers can improve the performance of data integration. Due to different quality of data sources, it is hard to find a general solution that works for every case. Therefore, we start from a general setting of truth analysis at first and narrow down to two basic problems in data integration. We first propose a general framework to deal with numerical data with flexibility of defining loss function. Source quality is represented by a vector to model the source credibility in different error interval. Then we propose a new method called **No Truth Truth Model(NTTM)** to deal with truth existence problem in low-quality data. Preliminary experiments on real stock data and slot filling data show promising results.

To my parents, for their love and support.

Acknowledgments

I would like to give special thanks to my advisor, Dr. Jiawei Han, for his continuous support on my research, inspiring advice on my topic, endless patience to instruct me, and his advice towards my professional and personal life. His support will guide me through the my whole PhD life in the upcoming years.

I would like to thank to Dr. Jing Gao, who is the assistant professor from University of Buffalo, and Dr. Bo Zhao, who is the researcher from Microsoft Research, Yanglei Song, for their contribution of ideas and practical advice on the collaboration of my research topic.

I would like to thank to Dr. Heng Ji, who is the associate professor from Rensselaer Polytechnic Institute and Dian Yu, for their helpful advice and expertise from natural language processing perspective, and their contribution on the ideas of the application of truth analysis in slot filling task.

I would also like to thank other colleagues, Hao Luo, Chi Wang, Jingjing Wang, Xiao Yu, Ahmed Elky, for their advice and helpful discussion on my research topic.

Table of Contents

Chapter 1

Introduction

The websites offer information for us every day, becoming an essential part of information sources for us. Not only the agencies with high reputation publish the news and reviews on their websites, but also informal websites and other social interaction media (social networks, discussion forums, blogs). Are all the information resources trustworthy? The answer probably is no. There is a lot of incorrect information on the websites. It is hard for one reader to identify whether a claim is true or not. Also, it is intractable for the information providers to make sure that every piece of news is credible. Besides, there are bad sources who spread rumors and intentionally modify the truth. A report by [1] mentions that consumers in US have low trust in information on the websites. Erroneous information and rumors would propagate on the web according to [2]. Thus, verifying the truth is an important and challenging problem.

On the other hand, data integration [3, 4, 5] have drawn people's attention for many years in all aspects of life. For example, when a patient is registered in many hospitals, integration of the information of the patient may be very helpful to the diagnosis of illness if treatment history is mapped correctly to the same person. Recently, the automatic construction of knowledge base have been paid more and more attention. Given a set of information resources, integrating conflicting descriptions of entities is very challenging.

The most straightforward method is *majority voting*. It collects all the facts describing one object, calculates the frequency of each type of facts, and chooses the fact with the maximum number of votes as the truth. However, this method fails to consider the trustworthiness of sources. It tends to provide false conclusions when multiple sources provide wrong facts about the target object, while only a few high-quality sources give the correct in-

formation. Therefore, majority voting, though simple and intuitive, often results in bad performance of accuracy and is intractable in many applications.

A better way to truth finding is to consider source quality. One basic intuition is that if an answer is provided by more reliable sources, it is more likely to be true. If one source is supported by many trustworthy answers, it is prone to be reliable. Therefore, it is natural to infer the truth and source quality together. It enables us to find truth and estimate source quality in an unsupervised, or semi-supervised way, which is the most common situation in truth finding problems.

Motivated by the need of obtaining high-quality information, truth finding [6, 7, 8, 9, 10, 11, 12, 13, 14, 15] have been studied for a long time in database community. Most of the previous work are focused on integrating data in categorical value[16, 17, 18, 19]. But numerical data exists commonly in data integration tasks, such as weather, stock price, flight price, age of a person, etc. Recently [20] propose a *Gaussian Truth Model (GTM)* which deals with real-valued data integration. In GTM, each claim is assumed to follow a Gaussian distribution centered as latent truth with variance of each source. Thus, source quality is represented by a single score–the variance estimated from the inferred truth. But in reality, this assumption may not hold. Therefore, we propose to define source quality as a vector. Each dimension of the vector identifies the accuracy of the source in the corresponding error interval. Thus, it helps to increase the granularity of the representation of source quality. Experiment on real stock data set shows the effectiveness of our model.

Another problem we are interested in is that when the overall quality of data is extremely low, how we can have a relatively good integration result. If we still use a single source quality score to measure each source, such as precision, integration output must be erroneous. For example, there are 18 related websites that provide birth place of famous people. There is one person whose birth place is actually unavailable in public, but there are still 2 systems providing an answer to it. When people make a query about the birth place of this person, the ideal output is to keep silent. We can calculate a confidence score of each answer and set a threshold to decide whether we

want to give out an answer or not. But the problems may be (1) It's hard to find a suitable threshold without supervision. (2)Even with weak supervision, it's too bold to have a strong belief on that because of the bias of limited training data. This problem becomes harder when correct answers are only provided by a small portion of sources. Say if precision is the only measure used to represent the source quality, and a large number of sources provide answers for both people whose birth place is commonly known, and the people whose birth place is unavailable, the judgment will be either to keep silent to all queries, or to give out answer for every question. In result, the overall performance is either with high precision, low recall or with high recall, low precision. In both cases, F1 measure is harmed seriously. Therefore, we propose to have more types of source quality to measure the performance of systems of different aspects. Error rate, miss rate and recall for each system are introduced later. We fit the data into a probabilistic graphical model and jointly estimate the source quality and the truth. Special techniques to initialize the priors are discussed, too. Experiments on real slot filling data set demonstrate the effectiveness of our method.

To summarize, our major contributions in this thesis are:

- Propose a general framework for numerical data integration and define a vector representation for source quality which could effectively represent the performance of sources in different error intervals;
- To the best of our knowledge, we are the first to propose a method to deal with such low-quality sources, whose F1 measure is around 30% at best.
- Our method can automatically decide whether or not to integrate answer instead of providing an integrated answer consistently across all questions.

In the following chapters, we first present a literature review on truth analysis problems. In Chapter 3, we introduce a general truth finding framework to integrate numerical data. In Chapter 4, we raise a new problem in truth finding and propose a new model for low-quality data integration. In Chapter 5, we implement our new models and test the effectiveness on two real

data sets. Finally we make a conclusion of the thesis.

Chapter 2

Literature Review

In this chapter, we make an overview of existing methods for the truth finding problem. First in Section 2.1, we introduce the basic truth finding framework and discuss their advantages as well as disadvantages. In Section 2.2, some extensions under the same framework are introduced. Then in Section 2.3, we present existing probabilistic models, which are more adaptable and have fewer parameters. Next, we discuss the copy detection and group detection problem in Section 2.5 and Section 2.4, respectively. Finally, some applications of truth finding methods are introduced in Section 2.6.

2.1 Basic Truth Finding Framework

In this section, we first introduce the basic truth finding framework. The network of sources, facts and objects is represented as Figure 2.1. One source provides multiple facts about multiple objects. One object is supported by multiple facts. Some facts are true but some are not. Table 2.1 is an example of this framework, where several websites provide the cast information for different movies. Netflix[1] asserts that Daniel Radcliffe and Emma Watson are the actors of Harry Potter, which is true, but BadResources[2] claims that Brad Pitt plays a part in Harry Potter, which is false. Based on such conflicting information, the goal of truth finding is to iteratively find out whether one claim for the object is true and also infer the quality of the fact.

Let s denote a source, f denote a fact and o denote an object. In the following, we give several definitions before introducing the truth finding methods. We first define the trustworthiness of a source, denoted by $t(s)$, as the confidence of the facts provided by s. We then define the confidence

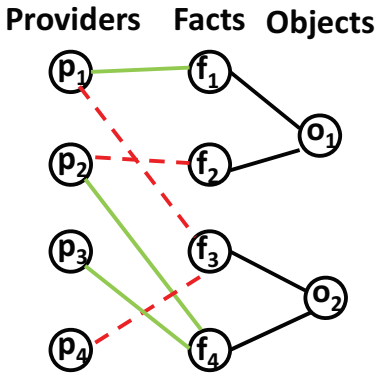
[1] www.netflix.com/

[2] www.badresources.com

Table 2.1: An example of data conflicts between sources: Websites, Cast and Movies

Source (Website)	Fact (Cast)	Object (Movie)
Netflix	Daniel Radcliffe, Emma Watson	Harry Potter
Netflix	Johnny Depp	Pirates 4
Hulu	Daniel Radcliffe, Emma Watson	Harry Potter
Hulu	Johnny Depp	Pirates 4
Badresources.com	Brad Pitt Harry	Potter
...	...	...

Figure 2.1: Basic Truth Finding Model summarized by [21].



Name	*Description*
M	Number of web sites
N	Number of facts
w	A web site
$t(w)$	The trustworthiness of w
$\tau(w)$	The trustworthiness score of w
$F(w)$	The set of facts provided by w
f	A fact
$s(f)$	The confidence of f
$\sigma(f)$	The confidence score of f
$\sigma^*(f)$	The adjusted confidence score of f
$W(f)$	The set of web sites providing f
$o(f)$	The object that f is about
$imp(f_j \rightarrow f_k)$	Implication from f_j to f_k
ρ	Weight of objects about the same object
γ	Dampening factor
δ	Max difference between two iterations

Figure 2.2: Definitions of notations in TruthFinder

of a fact, denoted by $t(f)$, is the probability that f is true. Given a raw database of tuples *(source, fact, object)*, our goal is to obtain truth value of each object, and trustworthiness of each source.

Most of the existing truth finding models are unsupervised, due to the limitation to obtain ground truth labels. A common practice is to iteratively compute the trustworthiness of sources and the confidence of facts. Next, we will introduce the intuition behind it.

2.1.1 TruthFinder

[16] propose an algorithm called TruthFinder. It originates from the idea that if one fact is provided by many trustworthy sources, it is likely to be true; however, if one fact is conflicting with other facts given by many trustworthy sources, it tends to be false. On the other hand, if one source provides facts with high confidence, it is highly credible; otherwise it is not. By iteratively estimating the trustworthiness of sources and the confidence of facts, Truth Finder can model the source quality to better infer the facts about the target objects.

To introduce the details of Truth Finder, we first give the necessary notations in Figure 2.2, and provide the definitions of *Confidence of facts* and *Trustworthiness of websites.*

Definition 1 (Confidence of facts) *The confidence of a fact f (denoted by $s(f)$) is the probability of f being correct.*

Definition 2 (Trustworthiness of websites) *The trustworthiness of a web site w (denoted by $t(w)$) is the expected confidence of the facts provided by w.*

In Truth Finder, the connection between the above two definitions is given by the following equation.

$$t(w) = \frac{\sum_{f \in F(w)} s(f)}{|F(w)|} \tag{2.1}$$

$$s(f) = 1 - \prod_{w \in W(f)} (1 - t(w)) \tag{2.2}$$

Equation 2.1 has an interpretation that the trustworthiness of a website is the average confidence of the facts it provides; and Equation 2.2 is the probability of a fact being true. These two equations build the connection between $t(w)$ and $s(f)$, which can be solved in an iterative way.

Different facts of one object may be conflicting with each other, while they could also support each other. Thus it brings another definition called *implication*. The implication from fact f_1 to f_2 is denoted by $imp(f_1 \rightarrow f_2)$. It is f_1's influence on f_2's confidence. $imp(f_1 \rightarrow f_2)$ is a value between -1 and 1. A positive value indicates f_1 and f_2 are positively correlated. While a negative value means if f_1 is correct, f_2 is likely to be wrong.

The algorithm is shown in Algorithm 2.3. In line 3, the confidence of fact is calculated as the multiplication of all the trustworthiness of sources providing this fact. In line 6, it takes the impact between facts into consideration, weighing the implication by parameter ρ. In line 9, the confidence of fact is compensated by a parameter γ, which represents the compensation for duplication between facts. In line 14, the trustworthiness of source is computed by all the confidence of facts it provides. The algorithm continues to calculate $t(s)$ and $s(f)$ until it converges, which is based on a threshold δ.

TruthFinder has several advantages over the naive Majority Voting strategy, because it considers the source quality and the source quality with the confidence of facts together. In empirical test, it has better performance over majority voting in the book-author experiment. Besides, in terms of the name abbreviation problem (i.e. Graeme Witt and G. Witt), it gives the

Figure 2.3: The basic algorithm proposed by [16].

Input: 1. Facts f provided by different providers related to objects $o \in O$. 2. Implications matrix imp.
Initialize $t(p)$ to a value v $\forall p$, where $0 \leq v \leq 1$.
while $|normalize(t^t) - normalize(t^{t-1})| \geq \delta$ **do**
  for every fact f **do**
    $\sigma(f) = log(\prod_{p \in P(f)}(1 - t(p)))$
  end for
  for every fact f **do**
    $\sigma^*(f) = \sigma(f) + \rho \sum_{o(f')=o(f)} \sigma(f') imp(f' \rightarrow f)$
  end for
  for every fact f **do**
    $s(f) = \frac{1}{1+e^{\gamma \sigma^*(f)}}$
  end for
  for every provider p **do**
    $t(p) = \frac{\sum_{f \in F(p)} s(f)}{|F(p)|}$
  end for
end while
return $t(p)$ and $s(f)$ for every f and p

abbreviation less score than the full name. Therefore it defines the fact in different granularities. However, the effectiveness of [16] requires the following four assumptions.

- Usually there is only one true fact for a property of an object.
- This true fact appears to be the same or similar on different sources.
- The false facts on different sources are less likely to be the same or similar.
- In a certain domain, a source that provides mostly true facts for many objects will likely provide true facts for other objects.

Because of the above assumptions, TruthFinder has the following disadvantages.

- **Multiple-value fact** It assumes that an object has only one true fact instead of multiple facts. In fact, many objects have multiple facts. For example, one movie could have 100 actors, while in TruthFinder, we could only assign k actors (i.e. three principle actors) as the true value. If a fact claims two actors or four actors, it would be masked as a false fact since most of the trustworthy sources assert three actors for this movie.

- **Copy detection** It assumes that the false facts on different sources are less likely to be the same. However, copy among facts from different sources happens frequently in the real world. Also in the later calculation of the confidence of facts, it uses the parameter γ to compensate the source dependency. But for different facts, the intensity of copy is likely to be different. Some sources are likely to copy from others while the others tend to be the original authors.

- **Domain Expert** It assumes that a source providing true facts for many objects will likely provide truth for other objects as well. But in reality, sources excel in particular domains. Thus, clustering is not applied in this scheme.

- **Rumors** It works only if most trustworthy sources provide the truth. But in certain domains no expert exists, and rumors can spread across the network. Thus, truth finding in heterogeneous network may help to split out rumors.

- **Parameters** There are three parameters in the Truth Finder method, which can be hard to tune in many truth finding applications.

2.2 Extensions of Basic TruthFinder

2.2.1 Alternatives of Propagation Functions

Since the basic truth finding algorithms have shortcomings, other fact finding models, aiming to improve the basic algorithms, have been introduced. [17] introduce several fact finders. They are in the same framework of Truth Finder but different in the way of calculating the confidence of facts and the trustworthiness of sources.

Sums Inspired by Hubs and Authorities[22], the authors in [17] treat sources as hubs and claims as authorities. We change their notations to make them more consistent with our earlier ones.

$$t^i(w) = \sum_{f \in F(w)} s^{i-1}(f) \qquad s^i(f) = \sum_{w \in W(f)} t^i(w) \tag{2.3}$$

Here, the authors normalize $s^i(f)$ and $t^i(w)$ to prevent overflow. The same normalization trick is also applied to the following alternatives proposed by the same authors.

Average·Log In TruthFinder method [16], the trustworthiness for a website is the average of the confidence of facts it provides. The authors in [17] claim these will overemphasize those sources with relatively few claims. So they use the log of the number of claims provided by a particular source to modulate the average.

$$t^i(w) = \log(|F(w)|)\frac{\sum_{f\in F(w)} s^{i-1}(f)}{|F(w)|} \tag{2.4}$$

Investment In this algorithm, the sources distribute their trustworthiness equally among their claims, and the total belief in a claim grows according to a nonlinear function. After each iteration, the sources are paid back proportional to their investment.

$$s^i(f) = G\left(\sum_{w\in W(f)} \frac{t^i(w)}{|W(f)|}\right) \tag{2.5}$$

$$t^{i+1}(w) = \sum_{f\in F(w)} s^i(f)\frac{\frac{t^i(w)}{|F(w)|}}{\sum_{r\in W(f)}\frac{t^i(r)}{|F(r)|}} \tag{2.6}$$

PooledInvestment The "investing" and "harvesting" processes remain the same as Investment. Now we hope the total belief of facts for a particular object remains the same, so we perform addition normalization procedure. Denote $H^i(f) = \sum_{w\in W(f)} \frac{t^i(w)}{|W(f)|}$, then

$$s^i(f) = H^i(f) * \frac{G(H^i(f))}{\sum_{d\in M_f} H^i(d)} \tag{2.7}$$

where M_f represents the facts concerning $o(f)$.

2.2.2 Hardness of Facts

In [9], the authors introduce three ways to estimate the uncertainty as well as the limited coverage of the claims. *Cosine* is based on the cosine similarity measure which measures the similarity of the truth value and the value given by one source. To estimate the truth value, they use simple averaging method. *2-Estimates* is a heuristic approach to estimate the truth values of facts the error of sources together. It computes the trust score of truth value and the probability of one source making errors iteratively with proper normalization. *3-Estimate* is an extension of *2-Estimate*, which calculates the difficulty of facts, i.e. the propensity of sources to be wrong on this fact. Intuitively, one source will earn more credits when it correctly answers a difficult question than answering something trivial. It estimates truth of facts, trust score of sources and hardness of facts iteratively.

2.2.3 Semi-supervised Fact Finding

Traditional fact finders are proposed in an unsupervised way. If false claims spread by copying among sources and false information takes the majority part, unsupervised methods are often ineffective. Thus, including some level of supervision can help with the iterative process of calculating truth values and source quality. [19] propose a semi-supervised approach called Semi-Supervised Truth Finder(SSTF) to find the true values. The intuition of SSTF is based on three principles: claims of the same source should have similar confidence score; similar claims should have similar confidence score; if two claims are conflicting, they cannot be both true. The claims and their relationships are encoded into a graph. Each claim is modeled as the node in the graph and the similarity between claims is models as the edge. If two claims are provided by the same source, or they support each other, the weight should be positive. If two claims are conflicting, the similarity score should be negative. The weight is normalized to $[-1, 1]$. Then, truth finding is equivalent to assign score to each fact that are consistent with the relationships between nodes indicated by the graph edges.

2.2.4 Generalized Fact Finding

In [18, 23] propose a generalized fact-finding framework, which allows to incorporate prior knowledge such as prior confidence of sources or the attributes of sources. Classical truth finders do not consider uncertainty in the information extraction process. In their paper, they separately calculate two types of uncertainty: w_u for uncertainty in information extraction, w_p for source uncertainty. w_σ is for the implicit assertion of similar claims of a source and w_g is for the implicit assertion of claims for the same object of a source. Since these factors are orthogonal, they combine them together to calculate final assertion weight:

$$w(w, f) = w_u(w, f) * w_p(w, f) + w_\sigma(w, f) + w_g(w, f) \tag{2.8}$$

And in the final step, they incorporate the assertion weight into the iteration equation, e.g. Sums, Average·Log and Invest etc.

More generally, they propose a layered model which allows to add new layers of groups or attributes to the existing bipartite graph of sources and claims. The additional layer directly connects to the sources to form a 3-layered graph.

2.3 Probabilistic Models for Truth Finding

In the previous section, we indicate that the basic truth finding algorithms have some drawbacks. One of the shortcomings is that it has a large number of parameters, which are tuned manually based on different dataset. Probabilistic models help to solve this problem. Incorporating the prior probability, these models succeed in automatically modeling the posterior probability based on all observations. Because it makes fewer assumptions than basic truth finding framework, its performance is often more stable and accurate.

2.3.1 Latent Truth Model

[15] introduce a probabilistic model called Latent Truth Model (LTM). The principle of LTM is that, by considering the truth as a latent random variable,

it is feasible to model the source quality and the complete spectrum of errors in a probabilistic framework. To illustrate LTM, we introduce the following definitions.

Definition 3 *$DB = \{row_1, row_2,' row_N\}$ is the raw database. Each row is a tuple (e, a, c), where e denotes the entity (object), a is the attribute, and c is the source.*

Definition 4 *$F = \{f_1, f_2,' f_F\}$ is the set of distinct facts selected from DB. Each fact has an unique id, forming a tuple (id_f, e_f, a_f).*

Definition 5 *$C = \{c_1, c_2,' .c_C\}$ is the set of claims selected from DB. Each claim is in the format of (id_{f_c}, s_c, o_c), where o_c is the observation of the claim, taking True or False*

Definition 6 *$T = \{t_1, t_2,' t_T\}$ is the set of truths, which takes a Boolean value of True or False. Each fact has a truth value, thus we denote the truth associated with f as t_f*

In the basic framework model, the trustworthiness of a source is under the assumption that there is one single truth for each entity. In fact, multiple values can be true. For example, one movie could have 100 actors. Multiple values of facts allow each actor to be considered as one truth. Even though one source provides one fact, it is considered to be true if this fact provides the correct actor for this movie. Also, the source quality providing one correct actor could be evaluated by partly offering the correct fact.

Another drawback of the basic framework is that the evaluation of source quality is based on a single parameter: whether one fact is true or not. But in practice, sources may differ in its preference to provide information. Under the assumption that one source could provide multiple facts for one entity, sources behave differently. Some sources are prone to provide more facts, while others are more conservative, likely to provide fewer correct facts. This intuition brings about the idea of evaluating source quality by two-sided measures. Table 2.2 shows the confusion matrix of source s.

Four derivative quality measures of source s are shown in Table 2.2. These measures have their own advantages and disadvantages. Precision only considers positive claims while ignoring negative claims. Accuracy takes both the positive and negative claims, but it ignores the difference between two

Table 2.2: Confusion Matrix of Source s

	t=True	t=False
o=True	True Positives(TP)	False Positive(FP)
o=False	False Negative(FN)	True Negative(TN)

different types of errors: false positive and false negative. Sensitivity and Specificity consider the two different types of errors and could distinguish the conservative sources from non-conservative sources, but using either of them alone still cannot represent all the characteristics of a source. Thus, Bo et al. propose a method to model the source quality by two-sided measures: sensitivity and specificity.

Table 2.3: Four derivative quality measures of source s

Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{TP+TN+FN+FP}$
Sensitivity or Recall	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{FP+TN}$

Figure 2.4 shows the structure of conditional dependence of LTM. The following is the algorithm of LTM.

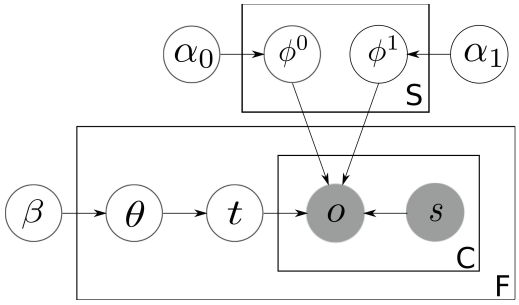
1. **Modeling Priors** For each source k, model its false positive rate $\phi_k^0 \sim Beta(\alpha_{0,1}, \alpha_{0,0})$, where $\alpha_{0,1}$ is the prior FP, and $\alpha_{0,0}$ is the prior TP. Model its false negative rate $\phi_k^1 \sim Beta(\alpha_{\{1,1\}}, \alpha_{1,0})$. For each fact f, model its **prior truth probability** $\theta_f \sim Beta(\beta_1, \beta_0)$, where β_1 is the prior true count, and β_0 is the prior false count. β determines how likely each fact is to be true. We can use a uniform prior if there is no prior knowledge to the facts. Model its **truth label** $t_f \sim Bernoulli(\theta_f)$. The prior probability that t_f is true is θ_f. For each claim c, model its **observation** $o_c \sim Bernoulli(\theta_{s_c}^{t_f})$

2. **MAP** The complete likelihood of all observations, latent variables and

unknown parameters is $p(o, s, t, \theta, \phi^0, \phi^1 | \alpha_0, \alpha_1, \beta)$. The best estimation is to get the maximum a posterior (MAP) for t:

$$\hat{t}_{MAP} = \arg\max_t \iiint p(o, s, t, \theta, \phi^0, \phi^1, \alpha_0, \alpha_1, \beta) d\theta d\phi^0 d\phi^1$$

Figure 2.4: The structure of conditional dependence of LTM proposed by [15]. S denotes source, F denotes fact and C denotes claim.



The Latent Truth Model makes the following contributions:

- **Granularity of source quality** It models the two-sided source quality, which makes LTM naturally support multiple values of facts for one object and outperform other models which can only support one truth for one object.
- **Prior domain knowledge** LTM incorporates the prior knowledge into its modeling, thus is more flexible for different datasets.

2.3.2 Gaussian Truth Model

Algorithms discussed above all deal with categorical data, and the GTM model in [20] is the first in the literature to deal with numerical truth.

The authors claim that due to the continuous nature, numerical data has a more common and severe issue in data quality. Examples include presidential election polls, census, and economic statistics to stock price predictions and weather forecasts[20].

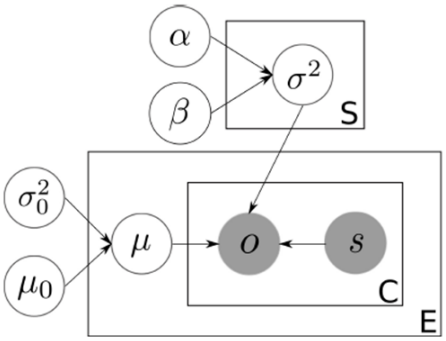


Figure 2.5: Graphical model for GTM

The difference between numerical and categorical data brings new challenges. First, unlike categorical data, numerical data have inherent distance or similarity. For example, if 50, 90 and 100 are observed for an object, it is reasonable to believe the truth lie close to 90 or 100, rather than 50. Second, in the numerical case, claims are not exactly right or wrong, but should get credits according the distance from the truth. Of course, closer claim will get more credits. Third, the consensus level among claims for each entity should be a factor in estimating truth and source quality. Fourth, numerical data can often have outliers. Some outliers may deviate very much from the truth, which will great impact the truth finding.

To tackle these challenges, authors in [20] propose GTM model, which is shown in figure 2.5. Next we will briefly introduce the generative story for the GTM.

First, they use deviation σ_s^2 to model the quality of sources. Intuitively, the smaller the deviation, the better the quality. In the paper, authors assume σ_s^2 is drawn from a inverse Gamma distribution:

$$\begin{aligned} & \sigma_s^2 \sim Inv - Gamma(\alpha, \beta) \\ \Rightarrow \quad & p(\sigma_s^2) = (\sigma_s^2)^{-\alpha-1} \exp\{-\frac{\beta}{\sigma_s^2}\} \end{aligned}$$

The choice of the inverse Gamma distribution is because it's the conjugate prior for Gaussian distribution, and the MAP inference is much more effi-

cient. If no addition information is available, the same parameter α, β can be used for all sources.

Second, the latent truth μ is drawn from a Gaussian distribution with mean μ_0 and variance σ_0^2:

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2) \tag{2.9}$$

In the original paper [20], the authors first remove outliers and then do normalization of the values provided for the same object. Since the focus their work is on trust analysis, they simply use z-score to remove outliers. After normalization, they set $\mu_0 = 0$ and $\sigma_0^2 = 1$.

Finally, the value provided by s_c for entity e is draw from a Gaussian distribution with the μ_e as the mean and $\sigma_{s_c}^2$ as the variance:

$$o_c \sim \mathcal{N}(\mu_e, \sigma_{s_c}^2) \tag{2.10}$$

So the joint distribution for all random variables is given as follows:

$$\begin{aligned} &p(o, \mu, \sigma | \alpha, \beta, \mu_0, \sigma_0) = \\ &\prod_{s \in S} p(\sigma_s^2 | \alpha, \beta) \prod_{e \in E} \left(p(\mu_e | \mu_0, \sigma_0^2) \prod_{c \in C_e} p(o_c | \mu_e, \sigma_e^2) \right) \end{aligned} \tag{2.11}$$

For inference on GTM, the authors use EM algorithm to compute the truth μ and the variance σ^2 in an iterative manner. By merit of Bayesian approach, the authors claim their methods can work in incremental mode.

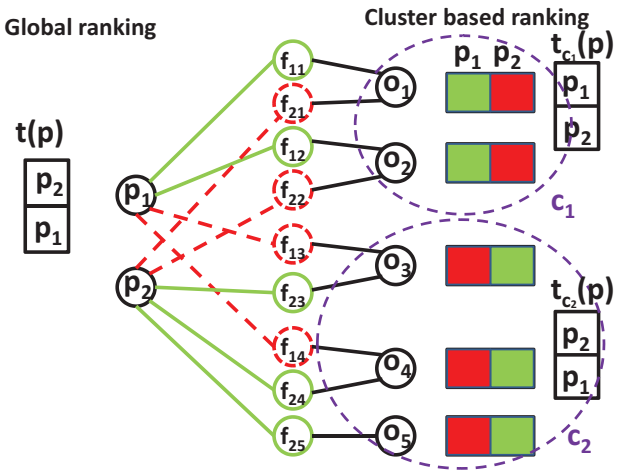
2.4 Truth Finding with Group Detection

2.4.1 Clustering-based Truth Finding

In the basic framework of truth finding, it does not consider the group effect of objects. Each resource may excel in certain domains, or for certain group of objects. Thus, group information helps to justify the truth of facts provided by different sources. [24] propose a clustering-based truth finding algorithm.

It assumes that objects can be clustered based on the trustworthiness of sources, and thus performs truth finding in a way personalized to a particular group of objects. Figure 2.6 shows the framework of the method.

Figure 2.6: Framework of clustering-based truth finding



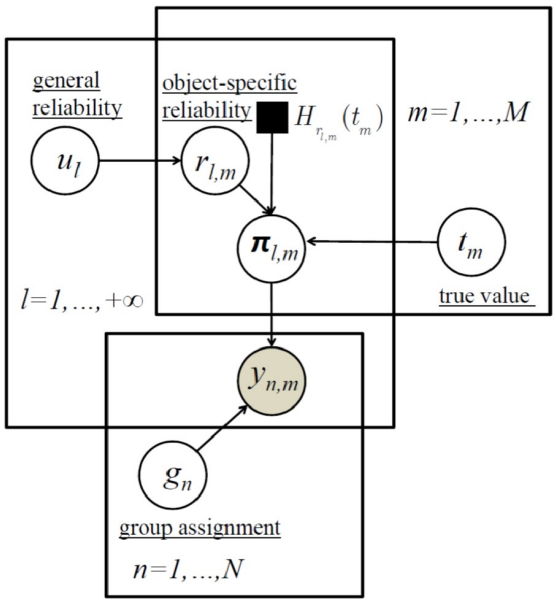
To compute the clustering of the objects, one can compute the trustworthiness of source for each object. Then the algorithm clusters the objects using the object-conditional trust (t_o(p), is computed as the confidence of the fact offered by p for object o) vectors.

2.4.2 Generative Model of Multi-source Sensing

Though Latent Truth Model considers the two-sided source quality and supports multi-value truth, it does not consider the source dependency, which may influence the sensitivity and specificity evaluation. [25] propose an algorithm to infer the source dependency by detecting latent group labels among sources together with source quality estimation and truth discovery. Dependent sources are grouped and their credibility is dependent on different group level. Figure 2.7 is the graphical model of the generative process of multi-source sensing problem. For each source S_n, they draw its group assignment g_n from a stick breaking process. For each group G_l, they draw it group reliability u_l from a beta distribution. For each object O_m, they draw its true value t_m from a uniform distribution. A object-specific group reliability indicator $r_{l,m}$ is drawn from a Bernoulli distribution parameterized by group

reliability u_l. The model parameter $\pi_{l,m}$ of each group on a particular object is drawn from the conjugate prior $H_{r_{l,m}}(t_m)$ to the distribution of observed claims $F(\pi_{g_n,m})$, which is dependent on the true value t_m and object-specific group reliability $r_{l,m}$. Then, given the group assignment g_n, each source generates its claim $y_{n,m}$ based on the corresponding distribution of observations $F(\pi_{g_n,m})$. Latent group assignment, truth value and group reliability are estimated by maximizing the observation likelihood. Variational inference is applied to approximate the parameters.

Figure 2.7: Graphic model for multi-source sensing



2.5 Truth Finding with Copy Detection

Web technologies have enabled data sharing but also simplified illegal copying. The model of copying can be complex, one article can copy several sources; some sources are copied several times by different articles; some articles do not directly copy but rephrase paraphrase the words in certain articles. There are growing needs for understanding these copying relationships for business and legal uses. For the data management applications, it is also important for data integration systems to track the copying relationships.

Otherwise, it is difficult to resolve the conflicting data sources. The conflicts can be between data sources, or between data and the real-world entity.

[7] is the first to consider the copy detection problem. The simple assumption is that the truth provided by the majority of sources is more likely to be true. However, simple voting methods are not enough. For example, when a false value is spread through copying, the voting method became venerable. They consider how to find true values from conflicting information when there are a large number of sources, among which some may copy from others. They present a novel approach that considers dependence between data sources in truth discovery. They assume that if two data sources provide a large number of common values, but many of which are rarely provided by other sources, it is very likely that one copies from the other and these values are wrong. They take the Bayesian analysis approach to decide the dependence of copying. They show with experiment that they can discover most of both copy-from-non-authority and copy-from-authority cases.

[8] propose a method of modeling copying between data in a dynamically changing world to invalidate out-of-date data. Straightforward ways to resolve the conflicts created by time-varying information sources may lead to noisy results. They propose a method to detect the true values and determining the copying relationship between sources when the update history of the sources is known. The change of sources are considered when evaluating the quality of the sources. Other metrics such as coverage, exactness and freshness mentioned by previous work [7] are also taken into account. They utilize the Hidden Markov Model to describe the copying behavior, including moment it happens. Then they use the Bayesian model on the aggregated data to decide the real value for a data item. Finally, they build the evolution path of the true values.

[26] propose a method of detecting not only copying between a pair of sources, but also the techniques in the presence of complex copying relationships. That is to say, the model of copying is not limited to one-to-one copying. They propose techniques that discover global copying relationships between a set of structured sources. First, they propose a detection method that can identify the real sources between co-copying and transitive copying.

Then they improve the techniques in [7] for detecting the copying direction by taking different types of evidence and correlations between different data items into account.

2.6 Applications of Truth Finding Algorithms

In this section, we will introduce some real applications of truth finding algorithms to resolve conflicts between claims of information providers.

2.6.1 Ranking System

Truth finding can be used to merge several query results from different sources in case of there is conflict between individual queries. For example, [11] propose a system to provide a result of web search for uses rather than letting the user to analyze the truth themselves. Similar systems are also used in question answering website like Quora[3] for recommending the right answers [27]. In crowd sourcing systems like Mechanical Turk[4], truth finding and copy detection are also useful for picking the most relevant work to user queries [28, 29]. Truth finding systems take the importance and similarity into account to rank the results. In the system [11] propose, the importance of the answer within the web sources which contain the answers are considered. It not only recommends the best matched answer, it also reduces the duplicated answers. It uses the frequency of answers as the metric. More specifically, multiple answers on a same page decreases the credibility, multiple page with the same answer increases the credibility. Then the ranking system ranks the independent answers as well as pointing out their web page sources. The positions of the answer in the corresponding web page and the their duplications are listed along the results. [11] uses the query results from the major search engine. They only consider top-k pages, where k is relatively small. However, the accuracy of the processed result is relatively accurate, since the distribution of of the relevance decreases fast with the duplicated results are eliminated and the relevant answers merged. Another explanation is that the score of the lowly ranked results will be trivial for the

[3] www.quora.com/
[4] www.mturk.com/

overall results.

[30] propose another truth finding method for web search. They take the credibility of each web site. The basic assumption is that it is very likely the result from the credible web site is credible. The iterative model of truth finding is employed to find the correct answer. Their system also considers only the top-k results from the major search engines and compute the scores of the results. The facts considered by [11] is also considered here, i.e., the duplication of the website and the data. It take these facts into accounts and iteratively compute the best score.

[31] propose an algorithm to rank the websites under certain keywords according to their popularity and influence. Similar to the the approach of [11] and [30], it is based on the assumption that if certain piece of content appears on more web pages, it is likely to have higher importance. Their system further improves the assumption by considering the score of certain sources in two parts. The first part is significance of the website itself, e.g., the number of visitors, the number of reports in the same field. The second part is the score of the the specific page, i.e, the influences of this page among all the pages. This assumption avoids the case where a credible website produces a report with less credibility. For example, a political review article from a sport website is not considered important in this case. In the iterative scheme, the initial values are set to the the significance of the source website. Then in every step, the popularity of the pages are taken into consideration. Then the score of the website is updated by the popularity. The result is supposed to that the website with more popular pages are assigned more significance, while the page on the more significant websites are assigned more popularity. The convergence of this iterative method is proved in their work.

2.6.2 Data Validation

In sensor networks, the results collected are not always consistent with each other. For example, due to the fault of sensors, the data collected may be contaminated. Simple statistical methods such as averaging are not applica-

ble here since it cannot identify the faulty sensor. [32] propose a fact finder in participatory sensing networks. It is the first truth finding systems designed for participatory sensors network data. Apollo extract shared participatory sensing data from Twitter. Then it convert the source data into a common representation of sources and claims. The clustering algorithm is performed on the inputs according to the similarity of observations. The number of claims are then reduced by a large portion. After clustering, it takes the iterative procedure to evaluate the credibility of each sensing data. They report that this validation procedure reduces significant amount of contaminated data.

[33] propose a data validation system for Twitter[5]. The basic assumption is that most of the tweets can be trusted, while sometimes contaminated data such as rumors and misinformation are propagated. Their system is based on the detailed analysis to the credibility of the news propagated. Then they propose a procedure for estimating the credibility of a certain tweet give a group of tweets on the same topic. They evaluate the tweets data related to certain topics and label the credibility results based on features extracted from them. The features include user behavior, text features of the tweets, and citation from outside resources. They try to assess the level of credibility of the social network information based on these features. Their procedure relies on the sociology concepts: the reactions of users from certain message and the emotion conveyed by users, the level of certainty of users, the external citations, and characteristics of the users propagating the information. They assume that tweets with strong emotional terms are highly related to non-credible information. Positive sentiment is more likely related to credible information. In the contrast, tweets with question marks or reference to another user are less incredible. Other useful features include depth of the re-tweets tree, presence of URLs, number of tweets by the corresponding users, number of friends of the users.

[5] www.twitter.com

2.6.3 Data Fusion

Data fusion aims to merge several data source into one in order to accelerate the data management procedure. Applications such as enterprise data managing [34], community data management [35] and scientific data sharing [36]. However, conflicting data poses challenges for data fusion applications. For example, the same category of data from different sources may be completely different because of errors, incompleteness or out-of-date. In this case, classification between the right and wrong data as well as correcting techniques are needed. [37] propose a novel method for this problem. They come up with classification techniques with several conflict resolving algorithms: conflict handling strategies, without resolution methods; conflict avoiding strategies, group-level resolution; conflict resolving strategies, individual-level resolution. They take several conflict resolution algorithms into account: the source accuracy method proposed by [16, 7]; the source freshness method by [8]; the source dependency method by [2].

2.6.4 Recommendation System

Truth finding can be applied to recommending the most original and significant news about a certain topic. For example, given the topic "US Election", there are millions of pages on different websites. However, some of them are rumors, and some of them are just irrelevant. Truth finding systems can pick the news which are the most relevant and closest to the truth. [38] propose a new truth finding model called Topic-oriented Website Evaluation Model (TWEM). TWEM mainly consider the interdependency between different websites and news articles. The dependency includes copying, citation and mutual support between news articles. TWEM also uses the popularity measured by traditional methods as the reference, for example, the Alexa Rank[6]. It also provide merging operation of two articles, i.e., as long as the similarity of two articles exceeds certain threshold, they are merged into one article. One of the two articles are considered super-article. Influence of the super article is computed as the portion of the common parts between the two articles. This is called merge-TWEM model.

[6] www.alexa.com/

Another system is called Corroboration Trust [39]. Corroboration Trust verify the credibility of certain news source by seeking more more than one source. That is to say, it is based on evidence. The evidence includes but are not limited to the news articles themselves. Other evidence information, such as person, location, time and keywords, which are all extracted from the news articles are considered. A news article is considered trustworthy if and only if the evidences extracted from it are all trustworthy. Corroboration Trust system extract the evidences of the articles using entity recognition techniques [40]. Then the news articles are grouped together using topic section and tracking [41]. In the case of paraphrasing certain words, it uses dependency tree analysis [42] on the entities already discovered. After the preprocessing steps, it then uses the extracted evidence and the clustered articles to compute the score of credibility using the iterative scheme. The basic assumption behind this model is that the articles are all dependent on each other, especially in the way that they tell the same stories, but with different forms. In this model, the credibility of an article is measured by its currency, availability, information-to-noise ratio, authority, popularity and cohesiveness.

Chapter 3

A General Framework to Integrate Numerical Data

In this chapter, we introduce a general framework to jointly estimate truth for numerical data and source quality. We give an iterative solution, and conduct experiments on real stock data.

3.1 Motivation

Different from categorical data, we need to consider the distances between claims. For example, if two sources provide 9,895 and 10,000 respectively, these values could both be true considering rounding. These two values support each other to some extent, and thus we have a higher confidence to say the latent truth is around 10,000 than observing only one of them. Also, for numerical data, we cannot directly use the absolute distance. For example, 9,895 and 10,000 seem to be closer than 1 and 5, even though they have a larger absolute distance. This implies the necessity of normalization.

3.2 Problem Formulation

We assume that there are S sources denoted by $\{s_j\}, j = 1, ..., N$, and Q questions $\{q_k\}, k = 1, ..., Q$ each with a unique truth $\{\mu_k\}, k = 1, ..., Q$. Denote by $o_{j,k}$ the answer provided by source s_j for question q_k. Here the sources are not required to answer all questions, and we do not allow for multiple truths. Further, we denote all sources providing answers for question q_k as $Src(q_k)$, and the collection of questions answered by source s_j as $Q(s_j)$. With these notations, we can formally define trust analysis problem as follows.

Definition 7 *Given all the the claims, denoted as O, provided by S sources*

for Q questions(objects), our goal is to find the truth $\{\mu_k\}, k = 1, ..., Q$.

Here the sources can be websites or human workers, and questions can be a particular attribute of an object. For example, websites may provide the author information for a particular book, or the value of a particular stock; a human may be asked whether an image contains sky. The claims can be categorical or numerical, while previous work mainly focus on categorical case.

3.3 A General Framework

It is commonly recognized that beyond majority voting, which treats every source equally, we should consider source quality when utilizing collective wisdom. Some sources may be more reliable than others, so they should have a larger weight in deciding the latent truth.

We denote the source quality of s_j as w_j. We formalize our problem as trying to find source quality $\{w_j\}$ and latent truth $\{\mu_k\}$ which maximize the probability of all claims O. In addition, we assume the questions are independent and the sources make decisions independently. Then we formulate the model as follows:

$$\max_{\{w_j\},\{\mu_k\}} \prod_{j=1}^{S} \prod_{k \in Q(s_j)} P(o_{j,k}|w_j, \mu_k). \tag{3.1}$$

We still need to specify the form of the conditional probability. Intuitively, if a source is more reliable, the probability that its claim is to the latent truth is high. We capture this intuition by first partitioning the distance range between the claim and the latent truth into L intervals $\{I_l\}, l = 1, ..L$, and define the source quality as

$$w_{s,l} = P(dist(o_s, \mu) \in I_l), \sum_{l=1}^{L} w_{s,l} = 1 \tag{3.2}$$

$$\max_{\{w_j\},\{\mu_k\}} \prod_{j=1}^{S} \prod_{k \in Q(s_j)} \prod_{l=1}^{L} w_{s,l}^{\mathbb{1}\{dist(o_{j,k},\mu_k \in I_l)\}}$$

$$s.t. \sum_{l=1}^{L} w_{j,l} = 1 \tag{3.3}$$

Thus, we give source quality an explicit meaning. The source quality $\mathbf{w_j}$ is a probability measure, which specifies the distribution of the distance between claims of source and latent truth. If a source is more reliable, then this distance tend to fall into intervals indicating small errors.

3.4 Iterative Solution

Once we define the distance function and intervals, our task is to find source quality and truth, which maximize the likelihood of claims, namely maximum likelihood estimation. We adopt the block coordinate descent method. In the first step, we fix the truth and maximize Equation 3.3 with respect to source quality. By simple calculus, we have

$$w_{j,l} = \frac{\sum_{k \in Q(s_j)} \mathbb{1}\{dist(o_{j,k}, \mu_k)\} \in I_l}{\sum_{k \in Q(s_j)} \sum_{i=1}^{L} \mathbb{1}\{dist(o_{j,k}, \mu_k)\} \in I_i} \tag{3.4}$$

Here, $w_{j,l}$ is the probability of $dist(o_{j,k}, \mu_k)$ falling into I_l, which is empirically the number of occurrence that the distance between provided claim and the latent truth falling into interval I_l divided by the number of questions s answers.

In the second step, we fix the source quality and maximize the objective w.r.t the truth. Given source quality, each question decouples. So for each question, we try find $\{\mu_k\}$ such that

$$\max \prod_{j \in Src(q_j)} \prod_{l=1}^{L} w_{j,l}^{\mathbb{1}} \{dist(o_{j,k}, \mu_k)\} \in I_l \tag{3.5}$$

3.5 Truth Finding for Numerical Data

Most of the previous work focuses on categorical data, and Gaussian Truth Model [20] is the only method in literature trying to integrate numerical data. In this section, we start with analyzing special characteristics of numerical data, and apply the framework to the numerical data. For example, if two sources provide 9,895 and 10,000 respectively, these values could both be true considering rounding. As aforementioned, these two values support each other to some extent, and we would have a higher confidence to say the latent truth is around 10,000 than only observing one of them. Also, for numerical data, we cannot directly use the absolute distance. For example, 9,895 and 10000 seem closer than 1 and 5, even though they have a larger absolute distance. This implies the necessity of normalization.

We propose a loss function as follows

$$L(\hat{\mu}, \mu) = \begin{cases} \frac{|\hat{\mu}-\mu|}{\alpha*\mu} & if|\hat{\mu}-\mu| < \alpha * \mu \\ 1 & otherwise \end{cases} \tag{3.6}$$

Here μ is the estimated truth, which is the output of trust analysis algorithm, and μ is the latent truth. We use this kind of evaluation function instead of absolute distance because we do not want to over-penalize. For example, suppose a source provides 99 correct answers for 100 questions, however, for the remaining last one question, its answer deviates infinitely from the truth. If absolute distance is used, this source will receive infinite penalty, which is not what we want. So in the proposed evaluation function, if the estimated truth is beyond a certain interval of the latent truth $(\mu-\alpha*\mu, \mu+\alpha*\mu)$, then the loss will be 1. α is a user-defined parameter, indicating the tolerance for a value to be considered close to the latent truth.

To apply the framework in Section 3.4, we need to define the interval partitioning for source quality. One natural way of defining the distance function is to use Equation 3.6. We can first partition R into 5 intervals: $I_1 = 0$, $I_2 = (0, 0.1]$, $I_3 = (0.1, 0.3]$, $I_4 = (0.3, 0.6]$, $I_5 = (0.6, 1]$.

An alternative distance function is introduced here. The evaluation function is symmetric at the latent truth μ, but we may want to distinguish

those sources who tend to provide higher value than actual truth from those constantly providing lower value. To this end, we propose an alternative distance function together with a possible interval partition.

$$dist(o_s, \mu) = \frac{2}{\pi} * atan(\frac{o_s - \mu}{\beta * \mu}) \tag{3.7}$$

This distance function changes faster around the latent truth, thus equal partition of the range of distance function will lead to finer resolution around the provided value.

Chapter 4

Integrating Multiple Low-Quality Sources to Discover Truth for Data Integration

4.1 Motivation

Previous work mostly focused on data in rather high quality. In the application of Truth Finder[16] on book-author data set, the accuracy of sources is around 90%. In the experiment of LTM[15] on book-author data set, the false positive rate is very low (under 10%), and there are some high-quality sources whose false positive rate is below 2%. Also, the recall of sources ranges from 50% to 90%. Under this condition, most of the claims provided by sources are correct, and source quality is a strong indicator of the credibility of provided claims. But in many tasks like the slot filling in natural language processing, the performance of every single system is quite low. For example, in the latest TAC-KBP slot filling task, the F1-measure of every single system is about 30% with either low precision or low recall. Moreover, there are many cases that truth value does not exist in any responses from claims of all sources. For example, if a system is asked the question like "What is the death date of Bill Gates?", whose truth does not exist. Some systems tend to provide an untrustworthy answer but some are prone to keep silent. The ideal output of our truth finding model is to disagree with any non-empty answers. In such circumstances, current algorithms cannot figure out the truth correctly. The main reason is that previous methods overlook the distinction between has-truth question with no-truth question.

Example 1 Table 4.1 shows a sample integrated raw data of slot filling task. Each column represents the answers to a certain question provided by 13 systems. If an element is blank, it means the system does not provide any answer to this question. There are totally 6 questions. For the first 4 questions, there are correct answers among all the candidates. For the last

4, there is no correct answer to this question. In other words, we cannot obtain the truth from current available claims provided by all systems. It is natural in the slot filling task, because it is possible that there is no such information in the corpus, or the useful information is stated so obscure that no system could successfully extract it. In both cases, the ideal output of our integration process should be correct answer if there is correct answer in the candidate answer list, and no output when no system can extract the answer.

Strategy 1

Majority voting easily fails in this example. If we only consider the first 4 questions, majority voting will have a correct judgment of question 1,3,4 and have a random guess of question 2 because 'Pakistan' and 'Afganistan' both have 2 voters. But if we move to the last 4 questions, majority voting will randomly choose a candidate answer instead of refusing to answer it. Truth Finder also suffers from this problem. Each candidate answer is given a score normalized by all the candidate answers to one question. If we consider the 5th question, system 16 will be evaluated as a better system than system 10 because system 16 provides 2 correct answers while system 10 provide no correct ones. According to the mechanism of Truth Finder, 'Netherlands' will be estimated as truth. But actually neither of these two is correct.

Strategy 2

An instant way to deal with the last 4 cases is to treat no-answer as an empty claim equally as the other candidates. If a system refuses to provide an answer, we can consider it providing an empty answer. Both majority voting and source-quality-based method can be applied again. Both methods can succeed in the last 4 questions, but will fail in the 2nd question, because the majority claim of it will be 'empty'.

To summarize, previous methods will fail in the case that both recall and precision of each system is low. It results in a large number of empty responses as well as incorrect candidate answers. Thus, we propose a new model called **No-Truth Truth Model (NTTM)** that can leverage the empty response and erroneous answers is proposed in the next section.

Table 4.1: An example of raw data of slot filling task

system	1:age	1:country	1:state_of_birth	2:age	2:country	3:age	3:country_of_birth	3:state_of_birth
2	43			50				
4	:59:11			:51:31			uk	
5	520627	pakistan		50				
6		afganistan	ghazni					
7	43			50				
8	43		ghazni	50				
9	43	afganistan	ghazni	50				
10	american	khost		marks	LONDON	marks	russian	hollywood
12	7/25/13	pakistan	pakistan			actress	spencer	spencer
13			kabul					
14	43			50				
15		afganistan	ghazni					
16	9		ghazni	50	Netherlands			
Truth	43	afganistan	ghazni	50	Empty	Empty	Empty	Empty

4.2 Problem Formulation

We now provide the details of our data model, and formally define the truth mining problem with low-quality sources.

4.2.1 Data Model

Suppose for each question $q_i, i = \{1, 2, ..., M\}$, each system $s_j, j = \{1, 2, ..., N\}$ will return only one answer or refuse to answer this question. If a system does not return one answer, we treat it as an "empty" answer. So for each question q_i, there are N answers $\{a_{ij}, j = 1, .., N\}$ returned by N systems, either a non-empty or an empty answer. There are N_i distinct non-empty candidate answers $\{d_{in}, n = 1, 2, ..., N_i\}$ $(N_i \leq N)$. We use E to represent an empty answer,and represent the truth of question q_i as t_i. There is only one candidate answer to be the true answer to question q_i. Thus, the input data is in the format of triples *(question, system, answer)* where question serves as a key entity we explore, system identifies from where the data originates and answer is a piece of information extracted by the source as a solution to the question.

Definition 8 *Let E to denote an empty answer when a system refuses to provide an answer to a question.*

Definition 9 *Let $Q = \{q_1, ..., q_M\}$ be the set of distinct questions where M is the total number of questions. Each question either has no truth or single truth.*

Definition 10 *Let $S = \{s_1, ..., s_N\}$ be the set of sources where N is the total number of sources.*

Definition 11 *Let $D_i = \{d_{i1}, ..., d_{iN_i}\}$ be the set of distinct non-empty candidate answers to question q_i where N_i is the number of distinct candidate answers to question q_i.*

Definition 12 *Let $T = \{t_1, ..., t_M\}$ be the set of truth where each t_i associated with a question q_i can either takes one of the candidate answers provided by sources or an empty value E.*

Definition 13 *Let $A = \{a_{11}, ..., a_{MN}\}$ be the set of candidate answers provided by all sources. Each candidate answer a_{ij} associated with q_i and s_j. Each system s_j will provide only one candidate answer to question q_i. The response can take either an empty or non-empty value.*

4.2.2 Problem Definition

Now we define the problem of interest in this thesis. Given a set of candidate answers a_{ij} for M questions provided by N systems, the goal is to (1) infer the true answer t_i to each question q_i, and (2) estimate the quality of each system.

The inference of truth is not independent of source quality. Source quality indicates how reliable each system is for the questions. It can be used to decide whether or to what extent to believe the claims given by systems. Also, the correctness of a claim can help to determine the source quality.

In the next section, we will define the source quality in the No-Truth Truth Model and explain why previous work are inadequate in the case of interest in our thesis.

4.3 Source Quality

In last section, we have already discussed the limitation of precision, which is the source quality score used in Truth Finder. It fails in the case when none of the systems provide a correct claim. In this section, we will define three types of quality scores and show that our algorithm can effectively estimate the truth existence.

We have shown in Table 4.1 that previous methods fail when they overlook the distinction between has-truth questions and no-truth questions. Thus, a natural approach is try to estimate the truth existence and distinguish the different performance of system in no-truth cases.

Table 4.2 shows the different system behaviors in has-truth and no-truth cases. When the truth exists in candidate answers, a system may have three types of behaviors: provide a candidate answer which is the truth (true

positive); provide a candidate answer which is not the truth (false positive); fails to provide a candidate answer (false negative). When truth of a question does not exist, a system may have two types of behaviors: a. provide a candidate answer (false positive); keep silence (true negative).

Table 4.2: System behaviors and corresponding source quality

Truth existence	System behavior	Category of behavior	Source quality
Has-truth	Empty answer	False negative	$\phi^{(1)}$
	Wrong answer	False positive	$\phi^{(2)}$
	Correct answer	True positive	$\phi^{(3)}$
No-truth	Non-empty answer	False positive	$\phi^{(2)}$
	Empty answer	True negative	$\phi^{(1)} + \phi^{(3)}$

To describe the different behaviors of system, we need to define multiple source quality. Here we introduce three types of source quality to measure the performance of each system s_j. We assume the system performance is consistent across all questions and all candidate answers. Later we will see the necessity of these definitions in Table 4.1.

- Miss rate, $\phi_j^{(1)} = p(a_{ij} = E|t_i = d_{in})$, is the possibility of not providing an answer when a question has a true answer.
- Error rate is $\phi_j^{(2)} = p(a_{ij} \neq d_{in}|t_i = d_{in}) = p(a_{ij} \neq E|t_i = E)$. Here we assume the error rate of each system on has-truth questions and no-truth questions is consistent. It is a reasonable assumption since each system will not distinguish whether a question has truth or not when it gives out an answer.
- Recall, $\phi_j^{(3)} = p(a_{ij} = d_{in}|t_i = d_{in})$, is the ability to provide a trustworthy answer when a question has truth.

The relationship between these three source quality is that miss_rate, error_rate and recall sums up to 1, i.e.,

$$\phi_j^{(1)} + \phi_j^{(2)} + \phi_j^{(3)} = 1. \tag{4.1}$$

Then we may derive the probability that a system does not provide an answer when true answer does not exist in all candidate answers, i.e. the sum of recall and miss rate.

$$
\begin{aligned}
p(a_i j = E | t_i = E) =& 1 - p(a_i j \neq E | t_i = E) \\
&= \phi_j^{(1)} + \phi_j^{(3)}
\end{aligned} \tag{4.2}
$$

The intuition behind Equation 4.2 is that when a system refuses to provide an answer to a question, it may result from two reasons. One is that it successfully makes the judgment that there is no true answer to be provided, which is associated recall. The other reason is that all the system fails to extract the correct answer, which is corresponding to the probability to make mistakes, i.e. the error rate.

For a system with high miss rate, the system is prone to keep silent to most of the questions, neither provide correct answers nor wrong answers. For a system with high error rate, the system is likely to provide answer to every question but few of them are correct. For a system with high recall, the system is reliable and provides trustworthy answers. We can see that our definition of source quality can model different behaviors or preference of systems.

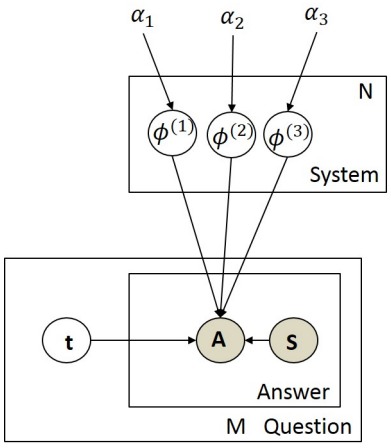
In Example 1, we can easily infer that system 9 is of high recall, system 10 and 12 are of high error rate and system 13 is of high miss rate.

Note that the ground truth labels are not given in the input. Instead, we must infer the hidden truths by our new model and estimate the source quality based on the inferred truth. But to evaluate the effectiveness of our method, human annotated labels are used in the experiment.

4.4 Truth Model with Low-Quality Sources

We tackle the truth existence problem using Bayesian network framework. Figure 4.1 is the graphical structure of our probabilistic model. Each node represents a random variable. The shaded ones indicate the variables are observed.

Figure 4.1: Probability Graphical Model of the New Model



4.4.1 Prior Settings

Prior of source quality

For each source $s_j \in S$, three types of quality miss rate $\phi_j^{(1)}$, error rate $\phi_j^{(2)}$ and recall $\phi_j^{(3)}$ is generated from a Dirichlet distribution with hyper-parameter $\alpha = (\alpha_1, \alpha_2, \alpha_3)$. Later we will see that α serves as the pseudo count of miss count, error count and correct count when estimating the corresponding source quality.

$$(\phi_j^{(1)}, \phi_j^{(2)}, \phi_j^{(3)}) \sim Dirichlet(\alpha_1, \alpha_2, \alpha_3) \tag{4.3}$$

Prior of candidate answers

For each question q_i, the probability that the truth is empty answer or each candidate answer should sum up to 1.

$$p(t_i = E) + \sum_{n=1}^{N_i} p(t_i = d_{in}) = 1 \tag{4.4}$$

For $p(t_i = E)$ and $p(t_i = d_{in})$, which denote the probability that question q_i has no truth, and the probability that the truth of question q_i equals to the candidate answer d_{in}, respectively, we will discuss the initialization of them in depth in Section 4.6.

4.4.2 Model Description

To infer the truth of each question, we can formulate it as a Maximize Likelihood Estimation (MLE) problem, i.e. to maximize the joint probability of source quality and observations by Equation 4.5. A set of latent variables $\{t_i\}$ are introduced. Each observation is parameterized by latent truth t_i and generated by N systems. It is originated from the simple Bayesian rule and is formulated as the combination of two mixing components: the has-truth part and no-truth part. The likelihood function and its expansion given latent truth is:

$$\begin{aligned}
&\max_{\boldsymbol{\phi}} p(\mathbf{A}, \mathbf{s}, \boldsymbol{\phi}|\boldsymbol{\alpha}) \\
=&\prod_{i=1}^{M} p(\mathbf{A}_{\mathbf{i},\cdot}, \mathbf{s}, \boldsymbol{\phi}|\boldsymbol{\alpha}) \\
=&\prod_{i=1}^{M} \left[p(\mathbf{A}_{\mathbf{i},\cdot}, \mathbf{s}|\boldsymbol{\phi}, t_i \neq E)p(t_i \neq E|\boldsymbol{\phi}) + p(\mathbf{A}_{\mathbf{i},\cdot}, \mathbf{s}|\boldsymbol{\phi}, t_i = E)p(t_i = E|\boldsymbol{\phi})\right] p(\boldsymbol{\phi}|\boldsymbol{\alpha})
\end{aligned} \tag{4.5}$$

where the first mixing component is shown in Equation 4.6 and the second one is shown in Equation 4.7. We assume the questions are independent and the sources make decisions independently. So systems and sources are decoupled with each other. Given the latent truth t_i, the likelihood of the observations is the multiplication of the source quality corresponds to the answer provided by each system. $\mathbb{1}\{\cdot\}$ is an indicator function that serves as a selector of corresponding source quality.

$$
\begin{aligned}
& p(\mathbf{A_{i,.}}, \mathbf{s} | t_i \neq E, \boldsymbol{\phi}) \\
=& \sum_{n=1}^{N_i} \left(\prod_{j=1}^{N} {\phi_j^{(1)}}^{\mathbb{1}\{a_{ij}=E\}} {\phi_j^{(2)}}^{\mathbb{1}\{a_{ij}\neq d_{in}, a_{ij}\neq E\}} (1-\phi_j^{(1)}-\phi_j^{(2)})^{\mathbb{1}\{a_{ij}=d_{in}, a_{ij}\neq E\}} \right) \\
& * p(t_i = d_{in} | t_i \neq E) \qquad (4.6) \\
& p(\mathbf{A_{i,.}}, \mathbf{s} | t_i = E, \boldsymbol{\phi}) \\
=& \prod_{j=1}^{N} {\phi_j^{(2)}}^{\mathbb{1}\{a_{ij}\neq E\}} (1-\phi_j^{(2)})^{\mathbb{1}\{a_{ij}=E\}} \qquad (4.7) \\
& p(\boldsymbol{\phi} | \boldsymbol{\alpha}) \\
=& C \prod_{j=1}^{N} {\phi_j^{(1)}}^{\alpha_1 - 1} {\phi_j^{(2)}}^{\alpha_2 - 1} (1-\phi_j^{(1)}-\phi_j^{(2)})^{\alpha_3 - 1} \qquad (4.8)
\end{aligned}
$$

Then our problem becomes: given a set of candidate answers $\mathbf{A}$, candidate answer list $\mathbf{d}$, prior $p(t_j = d_{in})$, $p(t_j = E)$ and conjugate prior for $\boldsymbol{\phi}^{(1)}, \boldsymbol{\phi}^{(2)}, \boldsymbol{\phi}^{(3)}$, we want to infer the parameters $\theta = \{\boldsymbol{\phi}^{(1)}, \boldsymbol{\phi}^{(2)}, \boldsymbol{\phi}^{(3)}\}$ and estimate the posterior of true answers $\mathbf{t}$.

4.5 Inference

Expectation-maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori estimates of parameters in statistical models. EM algorithm starts with randomly assigning values to all the parameters to be estimated. It then iteratively alternates between two steps, called the expectation step (E-step) and the maximization step (M-step), respectively. In the E-step, it computes the expected likelihood for the complete data(Q-function) where the expectation is taken. In the M-step, it re-estimates all the parameters by maximizing the Q-function. This process continues until the likelihood converges, i.e., reaching a local maximal.

Given the observed data, we use the Expectation-Maximization (EM) algorithm to infer the parameters $\theta = \{\boldsymbol{\phi}^{(1)}, \boldsymbol{\phi}^{(2)}, \boldsymbol{\phi}^{(3)}\}$ and estimate the posterior of true answers $\mathbf{t}$. $Q(\theta|\theta^{(k)})$ is the expectation of complete-data log-likelihood.

$$Q(\theta|\theta^{(k)}) = \mathbb{E}_{\theta^{(k)}}\left[l_{cd}(\theta)|\mathbf{A}, \mathbf{s}\right] \tag{4.9}$$

E-step: For each question q_i and each of its distinct candidate answer d_{in}, we calculate the posterior probability of the question not having a truth $\gamma_i^{(k)}$ and the probability that the truth of the question equals to the candidate answer d_{in},$\gamma_{in}^{(k)}$, given observations and current estimation of system quality

$$\begin{aligned}
\gamma_i^{(k)} =& p(t_i = E|\boldsymbol{\phi}^{(k)}, \mathbf{A}, \mathbf{s}) \\
=& \frac{\prod_{j=1}^{N} {\phi_j^{(2)}}^{\mathbb{1}\{a_{ij}\neq E\}}(1-\phi_j^{(2)})^{\mathbb{1}\{a_{ij}=E\}} p(t_i = E)}{p(\mathbf{A}, \mathbf{s}|\boldsymbol{\phi})} \qquad (4.10) \\
\gamma_{in}^{(k)} =& p(t_i = d_{in}, t_i \neq E|\boldsymbol{\phi}^{(k)}, \mathbf{A}, \mathbf{s}) \\
=& \frac{\prod_{j=1}^{N} {\phi_j^{(1)}}^{\mathbb{1}\{a_{ij}=E\}} {\phi_j^{(2)}}^{\mathbb{1}\{a_{ij}\neq d_{in}, a_{ij}\neq E\}}(1-\phi_j^{(1)}-\phi_j^{(2)})^{\mathbb{1}\{a_{ij}=d_{in}, a_{ij}\neq E\}}}{p(\mathbf{A}, \mathbf{s}|\boldsymbol{\phi})} \\
& * p(t_i = d_{in}, t_i \neq E) \qquad (4.11)
\end{aligned}$$

M-step: for every source j, we re-estimate the system quality $(\phi_j^{(1)}, \phi_j^{(2)}, \phi_j^{(3)})$ by maximizing the expectation of the complete-data likelihood $Q(\theta|\theta^{(k)})$. Take derivatives of Equation 4.9 with respect to $\phi_j^{(1)}$ and $\phi_j^{(2)}$, we can get:

$$0 = \frac{\partial Q(\theta|\theta^{(k)})}{\partial \phi_j^{(1)}} = \frac{a_j}{\phi_j^{(1)}} - \frac{b_j}{1-\phi_j^{(1)}-\phi_j^{(2)}} \tag{4.12}$$

$$0 = \frac{\partial Q(\theta|\theta^{(k)})}{\partial \phi_j^{(2)}} = \frac{c_j}{\phi_j^{(2)}} - \frac{d_j}{1-\phi_j^{(2)}} - \frac{b_j}{1-\phi_j^{(1)}-\phi_j^{(2)}} \tag{4.13}$$

where

$$a_j = \sum_{i=1}^{M} (1 - \gamma_i^{(k)}) \mathbb{1}\{a_{ij} = E\} + \alpha_1 - 1 \tag{4.14}$$

$$b_j = \sum_{i=1}^{M} \sum_{n=1}^{N_i} \gamma_{in}^{(k)} \mathbb{1}\{a_{ij} = d_{in}, a_{ij} \neq E\} + \alpha_3 - 1 \tag{4.15}$$

$$c_j = \sum_{i=1}^{M} \left(\gamma_i^{(k)} \mathbb{1}\{a_{ij} \neq E\} + \sum_{n=1}^{N_i} \gamma_{in}^{(k)} \mathbb{1}\{a_{ij} \neq d_{in}, a_{ij} \neq E\} \right) + \alpha_2 - 1 \tag{4.16}$$

$$d_j = \sum_{i=1}^{M} \gamma_i^{(k)} \mathbb{1}\{a_{ij} = E\} \tag{4.17}$$

$$\Rightarrow \phi_j^{(1)} = \frac{a_j}{a_j + b_j} (1 - \phi_j^{(2)}) \tag{4.18}$$

$$\phi_j^{(2)} = \frac{c_j}{a_j + b_j + c_j + d_j} \tag{4.19}$$

$$\phi_j^{(3)} = 1 - \phi_j^{(1)} - \phi_j^{(2)} = \frac{b_j}{a_j + b_j} (1 - \phi_j^{(2)}) \tag{4.20}$$

$$a_j + b_j + c_j + d_j = M + \alpha_1 + \alpha_2 + \alpha_3 - 3 \tag{4.21}$$

The estimation of source quality of system j is shown in Equation 4.18 to 4.21. Equations 4.14 to 4.17 are the empirical counts of each cases weighted by the probability of each case being true. Equation 4.14 is the weighted count of errors that judge a has-truth question as no truth. Equation 4.15 is the weighted count of cases providing a correct answer. Equation 4.16 is the weighted count of making two types of errors: providing an answer when there is no truth and giving an incorrect answer. Equation 4.17 is the weighted count of making correct judgment for no-truth questions. These weighted counts are added by corresponding pseudo counts originated from prior of source quality. Thus, the prior of source quality serves as a smoothing factor for source quality.

The estimation of posterior of correct answers are in Equation 4.10 and Equation 4.11. The intuition behind these two equations is very clear. Equation 4.10 indicates that if a system with high error rate does not provide an answer, or a system with low error rate keeps silent, this question is prone to having no truth. Equation 4.11 shows that if a system with high recall

provides this answer, or a system with high error rate does not claim it true, then we know that this answer is likely to be the truth.

Equation 4.21 is the total number of questions plus the pseudo counts. The error rate is the proportion of questions that a system make mistakes. Note that the estimation of miss rate and recall is not the proportion of its type of counts, but they separate the weighted count of Equation 4.17 proportional to their own counts. It makes sense because when a system does not provide an answer to a question which is at last proved to have no truth, it may come out of two reasons. One is that a high-recall system searches all contents in the corpus and is very confident that there is no correct answer to the question. The other is that a system of high miss rate fails to recognize the answer in the corpus. We can hardly distinguish these two cases, neither can we obtain the ground truth for it. Thus, making a fair separation of this part of counts on miss rate and recall is a reasonable solution in our case.

4.6 Prior Initialization

Truth Existence Initialization

We initialize the prior of truth existence($p(t_i = E)$) by features of a question. For each question, there are two features that are useful to indicate the truth existence: (1) The number of claims for each question (2) The number of the majority claim. For example, for question 1, the claims are listed in Table 4.3. In this example, the feature of question 1 is (9,4). The intuition behind this initialization is that if a large number of systems provide an answer to a certain question and the answers reach an agreement, for example, the votes of answers have peaks rather than uniform distributed, then more likely the question has correct answer within its candidate answers.

The clustering process can be conducted in either supervised or unsupervised way. If we select a small number of questions and acquire the labels of their truth existence, we may train a classifier on defined features to classify questions left. But in most real cases, the labeling information is not known in advance, or is expensive to obtain. Here we introduce an unsupervised method to coarsely estimate the clustering. By using a Gaussian Mixture

Question	Claim	Number of supporting systems
1	Afghanistan	4
1	Pakistan	2
1	Khost	1
1	Ghanzi	1
1	Iran	1

Table 4.3: Example of extracting features from claims

model (GMM), we can softly cluster questions into two groups: has-truth cluster and no-truth cluster. We may use the posterior probability of each question belonging to one cluster as the initialization score of truth existence. Since GMM is an unsupervised method, we can consider it as a "relative clustering", where the clustering of questions is affected by the behaviors of the other questions. In the experiment of next section, we can see the first initialization step can reach an accuracy of 0.81 in predicting cluster labels. Note that the only requirement here is that we need one labeling question to indicate which cluster is the has-truth cluster. The simplest way is to assume that the question with most systems and largest number of votes belongs to the has-truth cluster.

Figure 4.2 shows the ground truth of truth existence, where the red dots indicate no-truth questions, while the blue dots represent has-truth questions. Clustering centers and variance are shown by ellipsoids. From the result we can see that the dots around clustering centers have a high accuracy of being correctly labeled. For the ones that are given a biased prior of truth existence, our mechanism in the new model can effectively rectify the wrong judgment of truth existence. Figure 4.3 shows the accuracy of clustering using a threshold of 0.5, where the blue dots suggest the questions are correctly labeled while the red ones indicate wrong clustering.

Smoothing Factor. One problem with the initialization step is that the posterior probability of clustering may be very small, which is close to 0, or very large, which is close to 1. In this case, the prior judgment of truth existence of a question may be too bold. For example, if we set the prior for $p(t_i = E) = 10^{-5}$, then it is almost impossible for our new model to rectify the prior judgment. So we introduce a smoothing factor δ to compensate this bold judgment. Say we use $posterior + \delta$ as truth existence prior of posterior

Figure 4.2: Truth Existence Initialization

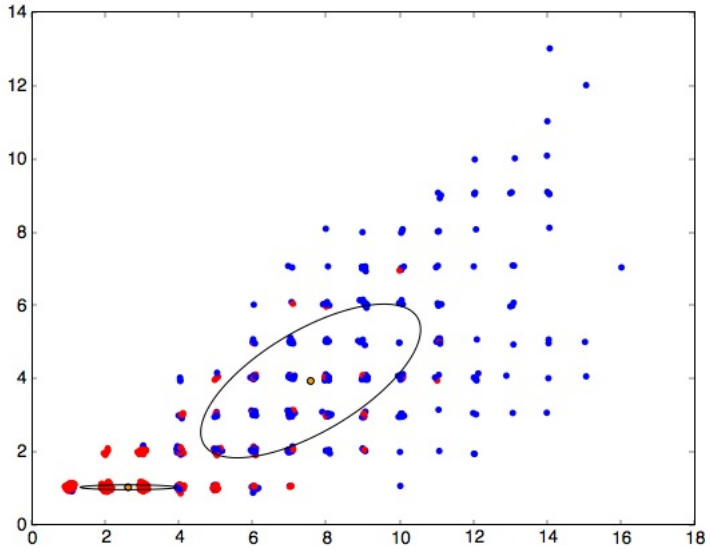
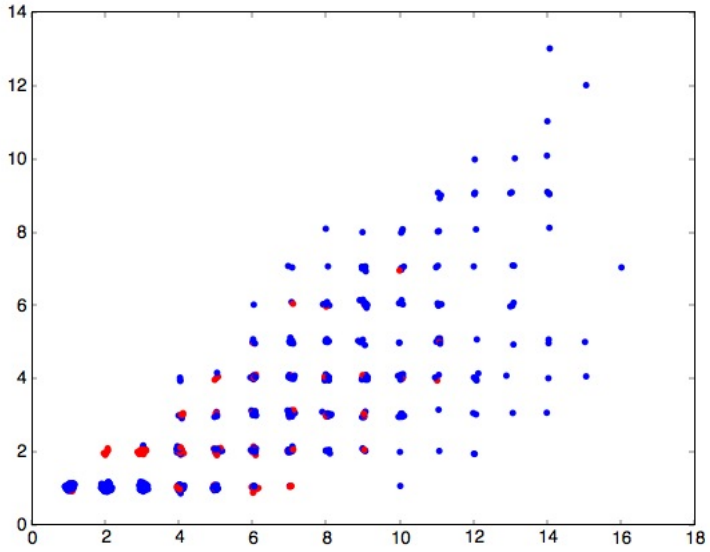


Figure 4.3: Accuracy of Initialization by GMM



of GMM $< 0.5 - \delta$ and $posterior - \delta$ if posterior $> 0.5 + \delta$.

For the non-empty candidate answers, we use the state-of-art method to initialize the prior for each answer. We set the prior for each candidate answer proportionally to the number of occurrence of the answers.

Algorithm 1 presents the pseudo-code for implementing the inference algorithm for NTTM.

Algorithm 1 EM Algorithm for NTTM inference

```
{Initialization of priors}
for all q_i ∈ Q do
    calculate feature vectors
Cluster q_i by GMM
for all t_i ∈ T do
    p(t_i = E) = p(q_i = E|GMM) ± δ
for all q_i ∈ Q do
    for all n = 1 → n_i do
        p(t_i = d_in) = (1 − p(t_i = E))/|N_i|
{Initialization of sources}
for all s_j ∈ S do
    initialize (φ_j^1)^(0), (φ_j^2)^(0), (φ_j^3)^(0)

{EM Algorithm}
for k = 1 → K do
    k ← k + 1
    {E-Step}
    for all q_i ∈ Q do
        for all n = 1 → n_i do
            compute γ_i^(k) and γ_in^(k)
    {M-Step}
    for s_j ∈ S do
        compute (φ_j^1)^(0), (φ_j^2)^(0), (φ_j^3)^(0)
        if sqrt(Σ_{l=1}^{3} ((φ_j^l)^(k) − (φ_j^l)^(k−1))^2) < ε then
            Stop EM loop
```

Result on toy example We run our algorithm on Example 1 and our model can correctly find all the correct answers. The source quality is listed in Table 4.4. In intuition, claims provided by high-recall systems such as system 9 and system 8 in Table 4.1 are usually correct. Empty claims that provided

by systems of low error rate are prone to be correct of no-truth question. Such as the empty claim of to the 5th question. System 2,4,5,6,7,8,9,12,13,14,15 all provide empty claim. Only 4 and 12 are of high error rate, while the others are of low error rate. Thus the 5th question will be judged as no truth question by our model.

Table 4.4: Inferenced Source Quality of New Model

system	miss rate	error rate	recall
SFV2013_02	0.5	0	0.5
SFV2013_04	0.63	0.38	0
SFV2013_05	0.38	0.25	0.38
SFV2013_06	0.5	0	0.5
SFV2013_07	0.5	0	0.5
SFV2013_08	0.25	0	0.75
SFV2013_09	0	0	1
SFV2013_10	0.13	0.88	0
SFV2013_12	0	0.88	0.13
SFV2013_13	0.88	0.13	0
SFV2013_14	0.5	0	0.5
SFV2013_15	0.5	0	0.5
SFV2013_16	0.25	0.25	0.5

Chapter 5

Experiment

In this section we demonstrate the effectiveness of our model on real-world data sets and compare it with state-of-art algorithms.

5.1 Experimental Setup

5.1.1 Data Set

We use the following real-world data sets to test the effectiveness of the two proposed models: VQTM and NTTM.

Real Stock Price Data Set We use the stock dataset in [43], wherein detailed description can be found. It contains 21 days stock data from 55 sources. We focus on Nasdaq-100 stock (100 largest stocks) each with 16 attributes, and we use the value provided by nasdaq.com as the latent truth. So the task becomes that based on the information provided the other 54 sources, we try to find the truth for these stocks. This data set is used to test our new model.

To measure the effectiveness of algorithms, we adopt the evaluation function given by Equation 3.6, and set $\alpha = 0.1$.

Slot Filling Data Set This data is got from TAC-KPB 2013 slot filling validation (SFV) task. In this task, each participating team is given a set of questions and is supposed to return the answers(slot fillers) and evidence sentences of the queries. This data set contains responses provided by 18 teams with 52 runs in all. There are 100 queries: 50 are about person and

50 are about organization. Because our model is focused on the questions that has only one correct answer, we select a subset of queries whose answer type is single. Our data set contains 3913 claims from 18 slot filling systems. There are 774 questions in total, within which 340 have a correct answer in the claims provided by 18 systems. This data set is used to test our NTTM model.

To evaluate the effectiveness of our model, we make use of the assessment results of the data set by TAC-KBP conference. The original labeling considers the correctness of both slot filler and evidence sentence. But in the construction of knowledge base, we are more interested in slot filler itself rather evidence sentence. Thus, we merge the conflicting judgment of the assessment by TAC-KBP in this way: we label one answer as correct when there is one assessor evaluates it as correct.

5.1.2 Environment

All the experiments presented were conducted on a laptop with 4 GB RAM, 2.4 GHz Intel Core i7 CPU, and OS X 10.8.5. Algorithms were implemented in Python 2.7.

5.2 Performance

We compare the effectiveness of our new model with previous state-of-art methods on the aforementioned data set. We briefly introduce them as follows, and provides the original paper here for reference.

5.2.1 Real Stock Data Set

- **Median** We treat the median of all claims as the latent truth.
- **Gaussian Truth Model** [20] This is the first work in the literature dealing with numerical data. We choose the parameter which has the best performance on the test part.

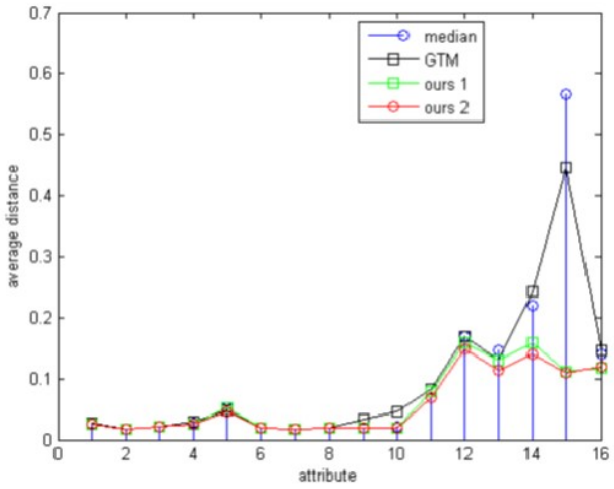


Figure 5.1: Average loss per attribute

- **New method 1** We use evaluation function as distance function 3.6, and adopt unequal partitioning. We set $\alpha = 0.1$ as in the evaluation function.

- **New method 2** We use an alternative distance function, and adopt equal partition. We set $\beta = 0.02$, which is an approximation of the evaluation function.

Table 5.1: Overall Loss of Stock Data

	Median	GTM	New 1	New 2
Overall Loss	1387.6	1392.1	1066.7	1000.6

To check whether our representation of source quality is reasonable, we can split data into training and testing part. For the training part, we reveal the truth, and thus we can accurately estimate the proposed source quality. Then we use the estimated source quality to find the truth on the testing part. If the representation is good, then the algorithm will perform reasonably well on the testing data. For the experiment on the stock data set, we use the first 8 days' data as training set, and the remain 13 days as testing set.

The overall penalty calculated by the evaluation Equation 3.6 is shown in Table 5.1. And the average loss for each attribute is plotted in Figure 5.1. We can see that our two methods can outperform the baseline (median) and

GTM in terms of overall loss. For some attributes (e.g first 10 attributes in Figure 5.1), the data quality is high, and all methods behave similarly. But for the latter attributes, our methods can actually improve the performance. These results indicate the effectiveness of our representation of source quality.

In our experiment, we split the dataset into training and testing part in order to check whether the representation of source quality is good. In many practical scenarios, golden standard is not available, or expensive to get. So we need to address the unsupervised setting. Secondly, we have shown that treating different attributes independently is better than putting all attributes together. It might be interesting to explore the relationship among source quality for different attributes.

5.2.2 Slot Filling Data Set

- **Majority Voting** For each question, we calculate the number of occurrence of each candidate answer provided by all sources. Here empty claims are ignored in majority voting. The majority among candidate answers to a certain question is considered as the estimated truth. In tie cases, we randomly picked up one answer when several candidate answers get the same count.

- **TruthFinder[16]** Consider only non-empty claim. For each non-empty candidate answer, we calculate the score of its being correct with precision of sources. The precision of sources are computed by the estimated score of each claim.

- **No-Truth Truth Model** For this data set, we run our new model and obtain the 3 source quality defined in the previous section together with estimated truth for each question.

Parameters for the Truth Finder is set as suggested in the original paper. The initial precision of sources is set to be 0.7. The dampen factor $\lambda = 0.3$. For our new method, we set the prior for source quality as $(\alpha_1, \alpha_2, \alpha_3) = (2.0, 2.0, 2.0)$. We didn't set a strong prior on source quality because extra knowledge of sources is not available in our current data set. But our model owns the capability to incorporate prior belief on the source

quality. The smoothing factor $\delta = 0.01$.

Table 5.2 shows the inference result of three methods. Result shows that our new model outperform existing methods by 20% and about 40% relative improvement in terms of F1 measure.

Table 5.2: Inference Result of Three Methods

Method	#correct	#provided	Precision	Recall	F1
Majority Voting	289	774	0.373	0.85	0.519
TruthFinder	303	774	0.391	0.891	0.544
NTTM	240	293	0.819	0.706	0.758

5.3 Case Study of Source Quality Prediction

For slot filling data set, we are interested in the quality of source quality prediction of NTTM. Here, Table 5.3 shows the MLE estimation and ground truth of the source quality of 18 extraction systems. The "ground truth" source quality is obtained in the following way. We feed ground truth labels of each question to the M-step to test the accuracy of our estimation of source quality. It shows that the optimal solution of likelihood function is very close to the ground truth.

5.4 Discussion

In the previous sections, we show some preliminary results. There remains many interesting issues that are worth attention.

Smoothing of priors One interesting issue is to figure out a more flexible way to smooth the initialization of truth existence. Current method is not flexible to different data set. One option is to treat truth existence as a latent random variable. It is assumed to be drawn from a prior and we would like to infer the truth existence together with the inference of truth value and source quality. And the output of GMM can be used as an initial value in

Table 5.3: Source Quality on Slot Filling Data

System	Our Model			Ground Truth		
	miss rate	error rate	recall	miss rate	error rate	recall
SFV2013_01	0.73	0.03	0.24	0.74	0.03	0.23
SFV2013_02	0.61	0.04	0.35	0.65	0.04	0.3
SFV2013_03	0.8	0.07	0.12	0.83	0.05	0.12
SFV2013_04	0.5	0.41	0.09	0.47	0.42	0.11
SFV2013_05	0.3	0.11	0.59	0.37	0.1	0.53
SFV2013_06	0.5	0.3	0.2	0.52	0.28	0.2
SFV2013_07	0.59	0.05	0.36	0.58	0.09	0.33
SFV2013_08	0.38	0.13	0.49	0.42	0.13	0.45
SFV2013_09	0.34	0.2	0.46	0.39	0.2	0.4
SFV2013_10	0.33	0.63	0.04	0.32	0.63	0.05
SFV2013_11	0.7	0.15	0.16	0.71	0.14	0.14
SFV2013_12	0.26	0.69	0.05	0.25	0.66	0.09
SFV2013_13	0.5	0.24	0.25	0.53	0.24	0.23
SFV2013_14	0.88	0.06	0.06	0.93	0.01	0.06
SFV2013_15	0.28	0.18	0.55	0.32	0.18	0.5
SFV2013_16	0.65	0.14	0.21	0.66	0.13	0.21
SFV2013_17	0.69	0.17	0.14	0.72	0.12	0.16
SFV2013_18	0.43	0.08	0.49	0.46	0.07	0.48

inference step.

Extension to Multiple truth Current model only focuses on single-truth data. To deal with multiple-truth data with low-quality sources is an open problem. We may extend our model to the multiple-truth cases.

Extension to Numerical truth Similar to multiple-truth data, we can try to extend our model to deal with numerical data such as age of a person.

Semi-supervised or Active Learning in initialization step By bringing a small amount of labels, we may have a better initial guess on truth existence. Or we may be interested in dynamically obtaining the labels of the question that on the boundary of the classifier.

From experiment perspective, more synthetic data sets can be created to test the effectiveness of our method. Empty claims and no-truth questions can be gradually added into the existing high-quality dataset. We can control the distribution of the data to test under what circumstances, our model work well and when it fails. Besides, convergence and scalability of our method can be explored on real data set further, though we use a standard EM process which has been proved efficiency and scalability in previous research.

Chapter 6

Summary

In this thesis, we first have a general overview of literature in truth analysis and crowd wisdom. Then we narrow down our problem to two challenging issues in truth finding. The first problem we deal with is to integrate numerical data. We propose a general optimization framework to allow free definition of source quality. Source quality and truth are dependent on each other. Then we define a vector representation of source quality for numerical data. Interval partitioning is conducted on the normalized loss axis. Each dimension of source quality measures the unique performance in each loss interval. We run our algorithm on real stock data set to evaluate the effectiveness. Result shows that our algorithm can outperform the state-of-art methods. The second problem we consider is to integrate multiple low-quality sources to discover truth. By defining three types of source quality: miss rate, error rate and recall, source behaviors are well captured in low-quality data. Theoretical analysis and experiments on real data set demonstrate the clear advantage of our method over any previous methods. We also list some remaining issues that may be interesting to explore.

References

[1] S. Cotten and S. Gupta, "Characteristics of online and offline health information seekers and factors that discriminate between them," *Social science and medicine*, vol. 59, no. 9, pp. 1795–1806, 2004. 1

[2] L. Berti-Equille, A. Sarma, A. Marian, D. Srivastava, *et al.*, "Sailing the information ocean with awareness of currents: Discovery and application of source dependence," *Fourth Biennial Conference on Innovative Data Systems Research (CIDR)*, 2009. 1, 25

[3] J. Bleiholder and F. Naumann, "Conflict handling strategies in an integrated information system," in *Proceedings of International Workshop on Information Integration on the Web (IIWeb)*, 2006. 1

[4] J. Bleiholder and F. Naumann, "Data fusion," *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 1, 2008. 1

[5] Z. Jiang, "A decision-theoretic framework for numerical attribute value reconciliation," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 7, pp. 1153–1169, 2012. 1

[6] "An approach to evaluate data trustworthiness based on data provenance," in *Secure Data Management* (C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, eds.), pp. 82–98, Springer, 2008. 2

[7] X. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 2, no. 1, pp. 550–561, 2009. 2, 21, 22, 25

[8] X. Dong, L. Berti-Equille, and D. Srivastava, "Truth discovery and copying detection in a dynamic world," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 2, no. 1, pp. 562–573, 2009. 2, 21, 25

[9] A. Galland, S. Abiteboul, A. Marian, and P. Senellart, "Corroborating information from disagreeing views," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 131–140, ACM, 2010. 2, 12

[10] G. Kasneci, J. V. Gael, D. Stern, and T. Graepel, "Cobayes: bayesian knowledge corroboration with assessors of unknown areas of expertise," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 465–474, ACM, 2011. 2

[11] M. Wu and A. Marian, "A framework for corroborating answers from multiple web sources," *Information Systems*, vol. 36, no. 2, pp. 431–449, 2011. 2, 22, 23

[12] V. Vydiswaran, C. Zhai, and D. Roth, "Content-driven trust propagation framework," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 974–982, ACM, 2011. 2

[13] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pp. 233–244, ACM, 2012. 2

[14] T. Wu, Y. Chen, and J. Han, "Re-examination of interestingness measures in pattern mining: a unified framework," *Data Mining and Knowledge Discovery*, vol. 21, no. 3, pp. 371–397, 2010. 2

[15] B. Zhao, B. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 5, no. 6, pp. 550–561, 2012. 2, 13, 16, 32

[16] X. Yin, J. Han, and P. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008. 2, 7, 9, 11, 25, 32, 52

[17] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 877–885, Association for Computational Linguistics, 2010. 2, 10, 11

[18] J. Pasternack and D. Roth, "Generalized fact-finding," in *Proceedings of the 20th international conference companion on World Wide Web*, pp. 99–100, ACM, 2011. 2, 13

[19] X. Yin and W. Tan, "Semi-supervised truth discovery," in *Proceedings of the 20th international conference on World Wide Web*, pp. 217–226, ACM, 2011. 2, 12

[20] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," *Proceedings of the International Workshop on Quality in Databases and Management of Uncertain Data (QDB)*, 2012. 2, 16, 17, 18, 30, 50

[21] M. Gupta and J. Han, "Heterogeneous network-based trust analysis: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 1, pp. 54–71, 2011. 6

[22] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999. 10

[23] J. Pasternack and D. Roth, "Making better informed trust decisions with generalized fact-finding," in *Proceedings of the 23nd international joint conference on Artificial Intelligence*, vol. 3, pp. 2324–2329, AAAI, 2011. 13

[24] M. Gupta, Y. Sun, and J. Han, "Trust analysis with clustering," in *Proceedings of the 20th international conference companion on World Wide Web*, pp. 53–54, ACM, 2011. 18

[25] G.-J. Qi, C. C. Aggarwal, J. Han, and T. Huang, "Mining collective intelligence in diverse groups," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1041–1052, 2013. 19

[26] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava, "Global detection of complex copying relationships between sources," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 3, no. 1-2, pp. 1358–1369, 2010. 21

[27] E. Voorhees, "Question answering in trec," in *Proceedings of the tenth international conference on Information and knowledge management*, pp. 535–537, ACM, 2001. 22

[28] P. Whitla, "Crowdsourcing and its application in marketing activities," *Contemporary Management Research*, vol. 5, no. 1, 2009. 22

[29] V. Raykar and S. Yu, "Ranking annotators for crowdsourced labeling tasks," *Advances in Neural Information Processing (NIPS)*, vol. 24, 2011. 22

[30] H. Dou, Q. Li, and Y. Zhang, "Find answers from web search results," in *Web Information Systems and Applications Conference (WISA), 2010 7th*, pp. 95–98, IEEE, 2010. 23

[31] S. Gao, Y. Miao, L. Yang, and C. Li, "Topic-based computing model for web page popularity and website influence," *AI 2009: Advances in Artificial Intelligence*, pp. 210–219, 2009. 23

[32] H. Le, J. Pasternack, H. Ahmadi, M. Gupta, Y. Sun, T. Abdelzaher, J. Han, D. Roth, B. Szymanski, and S. Adali, "Apollo: Towards factfinding in participatory sensing," in *10th International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 129–130, IEEE, 2011. 24

[33] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World Wide Web (WWW)*, pp. 675–684, ACM, 2011. 24

[34] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisiting the jdl data fusion model ii," tech. rep., DTIC Document, 2004. 25

[35] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets," *Journal of network and computer applications*, vol. 23, no. 3, pp. 187–200, 2000. 25

[36] G. Chin Jr and C. Lansing, "Capturing and supporting contexts for scientific data sharing via the biological sciences collaboratory," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 409–418, ACM, 2004. 25

[37] X. Dong and F. Naumann, "Data fusion: resolving data conflicts for integration," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 2, no. 2, pp. 1654–1655, 2009. 25

[38] Y. Miao, C. Li, L. Yang, L. Zhao, and M. Gu, "Evaluating importance of websites on news topics," *PRICAI 2010: Trends in Artificial Intelligence*, pp. 182–193, 2010. 25

[39] G. Zeng and W. Wang, "An evidence-based iterative content trust algorithm for the credibility of online news," *Concurrency and Computation: Practice and Experience*, vol. 21, no. 15, pp. 1857–1881, 2009. 26

[40] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the 7th conference on Natural language learning at HLT-NAACL*, pp. 188–191, Association for Computational Linguistics, 2003. 26

[41] J. Allan, "Introduction to topic detection and tracking," *Topic detection and tracking*, pp. 1–16, 2002. 26

[42] H. Yamada and Y. Matsumoto, "Statistical dependency analysis with support vector machines," in *Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT)*, vol. 3, 2003. 26

[43] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: is the problem solved?," *Proceedings of International Conference on Very Large Databases (VLDB)*, vol. 6, no. 2, pp. 97–108, 2012. 49