Truth Finding on the Deep Web: Is the Problem Solved?

Xian Li
SUNY at Binghamton
xianli@cs.binghamton.edu

Xin Luna Dong
AT&T Labs-Research
lunadong@research.att.com

Kenneth Lyons
AT&T Labs-Research
kbl@research.att.com

Weiyi Meng
SUNY at Binghamton
meng@cs.binghamton.edu

Divesh Srivastava
AT&T Labs-Research
divesh@research.att.com

ABSTRACT

The amount of useful information available on the Web has been growing at a dramatic pace in recent years and people rely more and more on the Web to fulfill their information needs. In this paper, we study truthfulness of Deep Web data in two domains where we believed data are fairly clean and data quality is important to people's lives: *Stock* and *Flight*. To our surprise, we observed a large amount of inconsistency on data from different sources and also some sources with quite low accuracy. We further applied on these two data sets state-of-the-art *data fusion* methods that aim at resolving conflicts and finding the truth, analyzed their strengths and limitations, and suggested promising research directions. We wish our study can increase awareness of the seriousness of conflicting data on the Web and in turn inspire more research in our community to tackle this problem.

1. INTRODUCTION

The Web has been changing our lives enormously. The amount of useful information available on the Web has been growing at a dramatic pace in recent years. In a variety of domains, such as science, business, technology, arts, entertainment, government, sports, and tourism, people rely on the Web to fulfill their information needs. Compared with traditional media, information on the Web can be published fast, but with fewer guarantees on quality and credibility. While conflicting information is observed frequently on the Web, typical users still trust Web data. In this paper we try to understand the truthfulness of Web data and how well existing techniques can resolve conflicts from multiple Web sources.

This paper focuses on Deep Web data, where data are stored in underlying databases and queried using Web forms. We considered two domains, *Stock* and *Flight*, where we believed data are fairly clean because incorrect values can have a big (unpleasant) effect on people's lives. As we shall show soon, data for these two domains also show many different features.

We first answer the following questions. Are the data consistent? Are correct data provided by the majority of the sources? Are the sources highly accurate? Is there an authoritative source that we can trust and ignore all other sources? Are sources sharing data with or copying from each other?

Our observations are quite surprising. Even for these domains that most people consider as highly reliable, we observed a large amount of inconsistency: for 70% data items more than one value is provided. Among them, nearly 50% are caused by various kinds of ambiguity, although we have tried our best to resolve heterogeneity over attributes and instances; 20% are caused by out-of-date data; and 30% seem to be caused purely by mistakes. Only 70% correct values are provided by the majority of the sources (over half of the sources); and over 10% of them are not even provided by more sources than their alternative values are. Although well-known authoritative sources, such as *Google Finance* for stock and *Orbitz* for flight, often have fairly high accuracy, they are not perfect and often do not have full coverage, so it is hard to recommend one as the "only" source that users need to care about. Meanwhile, there are many sources with low and unstable quality. Finally, we did observe data sharing between sources, and often on low-quality data, making it even harder to find the truths on the Web.

Recently, many *data fusion* techniques have been proposed to resolve conflicts and find the truth [2, 3, 6, 7, 8, 10, 13, 14, 16, 17, 18, 19, 20]. We next investigate how they perform on our data sets and answer the following questions. Are these techniques effective? Which technique among the many performs the best? How much do the best achievable results improve over trusting data from a single source? Is there a need and is there space for improvement?

Our investigation shows both strengths and limitations of the current state-of-the-art fusion techniques. On one hand, these techniques perform quite well in general, finding correct values for 96% data items on average. On the other hand, we observed a lot of instability among the methods and we did not find one method that is consistently better than others. While it appears that considering trustworthiness of sources, copying or data sharing between sources, similarity and formatting of data are helpful in improving accuracy, it is essential that accurate information on source trustworthiness and copying between sources is used; otherwise, fusion accuracy can even be harmed. According to our observations, we identify the problem areas that need further improvement.

Related work: Dalvi et al. [4] studied redundancy of structured data on the Web but did not consider the consistency aspect. Existing works on data fusion ([3, 8] as surveys and [10, 13, 14, 17, 19, 20] as recent works) have experimented on data collected from the Web in domains such as *book, restaurant* and *sports*. Our work is different in three aspects. First, we are the first to quantify and study consistency of Deep Web data. Second, we are the first to compare all fusion methods proposed up to date empirically. Finally, we focus on two domains where we believed data should be quite clean and correct values are more critical. We wish our study on these two domains can increase awareness of the seriousness of

Proceedings of the VLDB Endowment, Vol. 6, No. 2
Copyright 2012 VLDB Endowment 2150-8097/12/12... $ 10.00.

Table 1: Overview of data collections

	Srcs	Period	Objects	Local attrs	Global attrs	Considered items
Stock	55	July 2011	1000*21	333	153	16000*21
Flight	38	Dec 2011	1200*31	43	15	7200*31

Table 2: Examined attributes for *Stock*.

Last price	Open price	Today's change (%)	Today's change($)
Market cap	Volume	Today's high price	Today's low price
Dividend	Yield	52-week high price	52-week low price
EPS	P/E	Shares outstanding	Previous close

conflicting data on the Web and inspire more research in our community to tackle this problem.

In the rest of the paper, Section 2 describes the data we considered, Section 3 describes our observations on data quality, Section 4 compares results of various fusion methods, Section 5 discusses future research challenges, and Section 6 concludes.

2. PROBLEM DEFINITION AND DATA SETS

We start with defining how we model data from the Deep Web and describing our data collections[1].

2.1 Data model

We consider Deep Web sources in a particular *domain*, such as flights. For each domain, we consider *objects* of the same type, each corresponding to a real-world entity. For example, an object in the flight domain can be a particular flight on a particular day. Each object can be described by a set of *attributes*. For example, a particular flight can be described by scheduled departure time, actual departure time, etc. We call a particular attribute of a particular object a *data item*. We assume that each data item is associated with a single *true value* that reflects the real world. For example, the true value for the actual departure time of a flight is the minute that the airplane leaves the gate on the specific day.

Each data source can provide a subset of objects in a particular domain and can provide values of a subset of attributes for each object. Data sources have heterogeneity at three levels. First, at the schema level, they may structure the data differently and name an attribute differently. Second, at the instance level, they may represent an object differently. This is less of a problem for some domains where each object has a unique ID, such as stock ticker symbol, but more of a problem for other domains such as business listings, where a business is identified by its name, address, phone number, business category, etc. Third, at the value level, some of the provided values might be exactly the true values, some might be very close to (or different representations of) the true values, but some might be very different from the true values. In this paper, we manually resolve heterogeneity at the schema level and instance level whenever possible, and focus on heterogeneity at the value level, such as variety and correctness of provided values.

2.2 Data collections

We consider two data collections from *stock* and *flight* domains where we believed data are fairly clean and we deem data quality very important. Table 1 shows some statistics of the data.

Stock data: The first data set contains 55 sources in the *Stock* domain. We chose these sources as follows. We searched "stock price quotes" and "AAPL quotes" on *Google* and *Yahoo*, and collected the deep-web sources from the top 200 returned results. There were 89 such sources in total. Among them, 76 use the `GET` method (*i.e.*, the form data are encoded in the URL) and 13 use the `POST` method (*i.e.*, the form data appear in a message body). We focused on the former 76 sources, for which data extraction poses fewer problems. Among them, 17 use Javascript to dynamically generate data and 4 rejected our crawling queries. So we focused on the remaining 55 sources. These sources include some popular financial aggregators such as *Yahoo! Finance*, *Google Finance*, and *MSN Money*, official stock-market websites such as *NASDAQ*, and financial-news websites such as *Bloomberg* and *MarketWatch*.

We focused on 1000 stocks, including the 30 symbols from Dow Jones Index, the 100 symbols from NASDAQ Index (3 symbols appear in both Dow Jones and NASDAQ), and randomly chosen 873 symbols from the other symbols in Russell 3000. Every weekday in July 2011 we searched each stock symbol on each data source, downloaded the returned web pages, and parsed the DOM trees to extract the attribute-value pairs. We collected data one hour after the stock market closes on each day to minimize the difference caused by different crawling times. Thus, each object is a particular stock on a particular day.

We observe very different attributes from different sources about the stocks: the number of attributes provided by a source ranges from 3 to 71, and there are in total 333 attributes. Some of the attributes have the same semantics but are named differently. After we matched them manually, there are 153 attributes. We call attributes before the manual matching *local attributes* and those after the matching *global attributes*. Figure 1 shows the number of providers for each global attribute. The distribution observes *Zipf's law*; that is, only a small portion of attributes have a high coverage and most of the "tail" attributes have a low coverage. In fact, 21 attributes (13.7%) are provided by at least one third of the sources and over 86% are provided by less than 25% of the sources. Among the 21 attributes, the values of 5 attributes can keep changing after market close due to after-hours trading. In our analysis we focus on the remaining 16 attributes, listed in Table 2. For each attribute, we normalized values to the same format (*e.g.*, "6.7M", "6,700,000", and "6700000" are considered as the same value).

For purposes of evaluation we generated a gold standard for the 100 NASDAQ symbols and another 100 randomly selected symbols. We took the voting results from 5 popular financial websites: *NASDAQ*, *Yahoo! Finance*, *Google Finance*, *MSN Money*, and *Bloomberg*; we voted only on data items provided by at least three sources. The values in the gold standard are also normalized.

Flight data: The second data set contains 38 sources from the flight domain. We chose the sources in a similar way as in the stock domain and the keyword query we used is "flight status". The sources we selected include 3 airline websites (*AA, UA, Continental*), 8 airport websites (such as *SFO, DEN*), and 27 third-party websites, including *Orbitz, Travelocity*, etc.

We focused on 1200 flights departing from or arriving at the *hub* airports of the three airlines (*AA, UA,* and *Continental*). We grouped the flights into batches according to their scheduled arrival time, collected data for each batch one hour after the latest scheduled arrival time every day in Dec 2011. Thus, each object is a particular flight on a particular day. We extracted data and normalized the values in the same way as in the *Stock* domain.

There are 43 local attributes and 15 global attributes (distribution shown in Figure 1). Each source covers 4 to 15 attributes. The distribution of the attributes also observes *Zipf's law*: 6 global attributes (40%) are provided by more than half of the sources while 53% of the attributes are provided by less than 25% sources. We focus on the 6 popular attributes in our analysis, including *scheduled departure/arrival time, actual departure/arrival time*, and *departure/arrival gate*. We took the data provided by the three airline websites on 100 randomly selected flights as the gold standard.

[1] Our data are available at http://lunadong.com/fusionDataSets.htm.

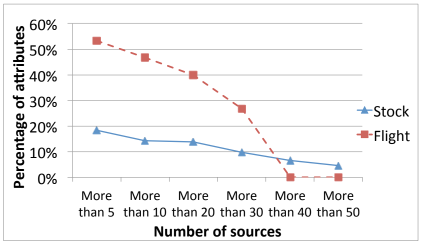


Figure 1: Attribute coverage.

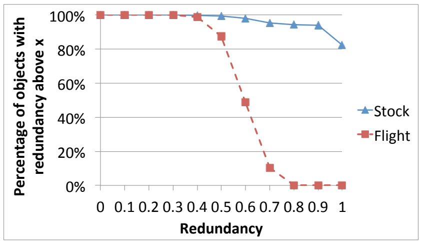


Figure 2: Object redundancy.

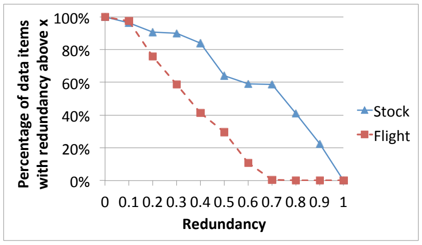


Figure 3: Data-item redundancy.

Summary and comparison: In both data collections objects are easily distinguishable from each other: a stock object can be identified by date and stock symbol, and a flight object can be identified by date, flight number, and departure city (different flights departing from different cities may have the same flight number). On the other hand, we observe a lot of heterogeneity for attributes and value formatting; we have tried our best to resolve the heterogeneity manually. In both domains we observe that the distributions of the attributes observe *Zipf's Law* and only a small percentage of attributes are popular among all sources. The *Stock* data set is larger than the *Flight* data set with respect to both the number of sources and the number of data items we consider.

Note that generating gold standards is challenging when we cannot observe the real world in person but have to trust some particular sources. Since every source can make mistakes, we do voting on authority sources when appropriate.

3. WEB DATA QUALITY

We first ask ourselves the following four questions about Deep Web data and answer them in this section.

1. *Are there a lot of redundant data on the Web?* In other words, are there many different sources providing data on the same data item?
2. *Are the data consistent?* In other words, are the data provided by different sources on the same data item the same and if not, are the values provided by the majority of the sources the true values?
3. *Does each source provide data of high quality in terms of correctness and is the quality consistent over time?* In other words, how consistent are the data of a source compared with a gold standard? And how does this change over time?
4. *Is there any copying?* In other words, is there any copying among the sources and if we remove them, are the majority values from the remaining sources true?

We report detailed results on a randomly chosen data set for each domain: the data of 7/7/2011 for *Stock* and the data of 12/8/2011 for *Flight*. In addition, we report the trend on all collected data (collected on different days).

3.1 Data redundancy

We first examine redundancy of the data. The *object (resp., data-item) redundancy* is defined as the percentage of sources that provide a particular object (resp., data item). Figure 2 and Figure 3 show the redundancy on the objects and data items that we examined; note that the overall redundancy can be much lower.

For the *Stock* domain, we observe a very high redundancy at the object level: about 16% of the sources provide all 1000 stocks and all sources provide over 90% of the stocks; on the other hand, almost all stocks have a redundancy over 50%, and 83% of the stocks have a full redundancy (*i.e.*, provided by all sources). The redundancy at the data-item level is much lower because different sources can provide different sets of attributes. We observe that 80% of the sources cover over half of the data items, while 64% of the data items have a redundancy of over 50%.

For the *Flight* domain, we observe a lower redundancy. At the object level, 36% of the sources cover 90% of the flights and 60% of the sources cover more than half of the flights; on the other hand, 87% of the flights have a redundancy of over 50%, and each flight has a redundancy of over 30%. At the data-item level, only 28% of the sources provide more than half of the data items, and only 29% of the data items have a redundancy of over 50%. This low redundancy is because an airline or airport web site provides information only on flights related to the particular airline or airport.

Summary and comparison: Overall we observe a large redundancy over various domains: on average each data item has a redundancy of 66% for *Stock* and 32% for *Flight*. The redundancy neither is uniform across different data items, nor observes *Zipf's Law*: very small portions of data items have very high redundancy, very small portions have very low redundancy, and most fall in between (for different domains, "high" and "low" can mean slightly different numbers).

3.2 Data consistency

We next examine consistency of the data. We start with measuring inconsistency of the values provided on each data item and consider the following three measures. Specifically, we consider data item d and we denote by $\bar{V}(d)$ the set of values provided by various sources on d.

- *Number of values:* We report the number of different values provided on d; that is, we report $|\bar{V}(d)|$, the size of $\bar{V}(d)$.
- *Entropy:* We quantify the *distribution* of the various values by *entropy* [15]; intuitively, the higher the inconsistency, the higher the entropy. If we denote by $\bar{S}(d)$ the set of sources that provide item d, and by $\bar{S}(d, v)$ the set of sources that provide value v on d, we compute the entropy on d as
$$E(d) = -\sum_{v \in \bar{V}(d)} \frac{|\bar{S}(d,v)|}{|\bar{S}(d)|} \log \frac{|\bar{S}(d,v)|}{|\bar{S}(d)|}. \quad (1)$$
- *Deviation:* For data items with conflicting numerical values we additionally measure the difference of the values by deviation. Among different values for d, we choose the *dominant value* v_0 as the one with the largest number of providers; that is, $v_0 = \arg\max_{v \in \bar{V}(d)} |\bar{S}(d, v)|$. We compute the deviation for d as the relative deviation w.r.t. v_0:
$$D(d) = \sqrt{\frac{1}{|\bar{V}(d)|} \sum_{v \in \bar{V}(d)} (\frac{v - v_0}{v_0})^2}. \quad (2)$$
We measure deviation for time similarly but use absolute difference by minute, since the scale is not a concern there.

We have just defined dominant values, denoted by v_0. Regarding them, we also consider the following two measures.

Table 3: Value inconsistency on attributes. The numbers in parentheses are those when we exclude source *StockSmart*.

	Attribute w. low incons.	Number	Attribute w. high incons.	Number
Stock	Previous close	1.14 (1.14)	Volume	7.42 (6.55)
	Today's high	1.98 (1.18)	P/E	6.89 (6.89)
	Today's low	1.98 (1.18)	Market cap	6.39 (6.39)
	Last price	2.21 (1.33)	EPS	5.43 (5.43)
	Open price	2.29 (1.29)	Yield	4.85 (4.12)
Flight	Scheduled depart	1.1	Actual depart	1.98
	Arrival gate	1.18	Scheduled arrival	1.65
	Depart gate	1.19	Actual arrival	1.6
	Low-var attr	Entropy	High-var attr	Entropy
Stock	Previous close	0.04 (0.04)	P/E	1.49 (1.49)
	Today's high	0.13 (0.05)	Market cap	1.39 (1.39)
	Today's low	0.13 (0.05)	EPS	1.17 (1.17)
	Last price	0.15 (0.07)	Volume	1.02 (0.94)
	Open price	0.19 (0.09)	Yield	0.90 (0.90)
Flight	Scheduled depart	0.05	Actual depart	0.60
	Depart gate	0.10	Actual arrival	0.31
	Arrival gate	0.11	Scheduled arrival	0.26
	Low-var attr	Deviation	High-var attr	Deviation
Stock	Last price	0.03 (0.02)	Volume	2.96 (2.96)
	Yield	0.18 (0.18)	52wk low price	1.88 (1.88)
	Change %	0.19 (0.19)	Dividend	1.22 (1.22)
	Today's high	0.33 (0.32)	EPS	0.81 (0.81)
	Today's low	0.35 (0.33)	P/E	0.73 (0.73)
Flight	Schedule depart	9.35 min	Actual depart	15.14 min
	Schedule arrival	12.76 min	Actual arrival	14.96 min

- *Dominance factor:* The percentage of the sources that provide v_0 among all providers of d; that is, $F(d) = \frac{|\bar{S}(d,v_0)|}{|\bar{S}(d)|}$.
- *Precision of dominant values:* The percentage of data items on which the dominant value is *true* (*i.e.*, the same as the value in the gold standard).

Before describing our results, we first clarify two issues regarding data processing.

- *Tolerance:* We wish to be fairly tolerant to slightly different values. For time we are tolerant to 10-minute difference. For numerical values, we consider all values that are provided for each particular attribute A, denoted by $\bar{V}(A)$, and take the median; we are tolerant to a difference of

$$\tau(A) = \alpha * \text{Median}(\bar{V}(A)), \qquad (3)$$

  where α is a predefined *tolerance factor* and set to .01 by default.
- *Bucketing:* When we measure value distribution, we group values whose difference falls in our tolerance. Given numerical data item d of attribute A, we start with the dominant value v_0, and have the following buckets: $\dots, (v_0 - \frac{3\tau(A)}{2}, v_0 - \frac{\tau(A)}{2}], (v_0 - \frac{\tau(A)}{2}, v_0 + \frac{\tau(A)}{2}], (v_0 + \frac{\tau(A)}{2}, v_0 + \frac{3\tau(A)}{2}], \dots$.

Inconsistency of values: Figure 4 shows the distributions of inconsistency by different measures for different domains and Table 3 lists the attributes with the highest or lowest inconsistency.

Stock: For the *Stock* domain, even with bucketing, the number of different values for a data item ranges from 1 to 13, where the average is 3.7. There are only 17% of the data items that have a single value, the largest percentage of items (30%) have two values, and 39% have more than three values. However, we observe one source (*StockSmart*) that stopped refreshing data after June 1st, 2011; if we exclude its data, 37% data items have a single value, 16% have two, and 36% have more than three. The entropy shows that even though there are often multiple values, very often one of them is dominant among others. In fact, while we observe inconsistency on 83% items, there are 42% items whose entropy is less than .2 and 76% items whose entropy is less than 1 (recall that the maximum entropy for two values, happening under uniform distribution, is 1). After we exclude *StockSmart*, entropy on some attributes is even lower. Finally, we observe that for 64% of the numerical data items the deviation is within .1; however, for 14% of the items the deviation is above .5, indicating a big discrepancy.

The lists of highest- and lowest-inconsistency attributes are consistent w.r.t. number-of-values and entropy, with slight changes on the ordering. The lists w.r.t. deviation are less consistent with the other lists. For some attributes such as Dividend and 52-week low price, although there are not that many different values, the provided values can differ a lot in the magnitude. Indeed, different sources can apply different semantics for these two attributes: Dividend can be computed for different periods–year, half-year, quarter, etc; 52-week low price may or may not include the price of the current day. For Volume, the high deviation is caused by 10 symbols that have terminated–some sources map these symbols to other symbols; for example, after termination of "SYBASE", symbol "SY" is mapped to "SALVEPAR" by a few sources. When we remove these 10 symbols, the deviation drops to only .28. Interestingly, Yield has high entropy but low deviation, because its values are typically quite small and the difference is also very small. We observe that real-time values often have a lower inconsistency than statistical values, because there is often more semantics ambiguity for statistical values.

Flight: Value inconsistency is much lower for the *Flight* domain. The number of different values ranges from 1 to 5 and the average is 1.45. For 61% of the data items there is a single value after bucketing and for 93% of the data items there are at most two values. There are 96% of the items whose entropy is less than 1.0. However, when different times are provided for departure or arrival, they can differ a lot: 46% of the data items have a deviation above 5 minutes, while 20% have a deviation above 10 minutes.

Among different attributes, the scheduled departure time and gate information have the lowest inconsistency, and as expected, the actual departure/arrival time have the highest inconsistency. The average deviations for actual departure and arrival time are as large as 15 minutes.

Reasons for inconsistency: To understand inconsistency of values, for each domain we randomly chose 20 data items and in addition considered the 5 data items with the largest number-of-values, and manually checked each of them to find the possible reasons. Figure 6 shows the various reasons for different domains.

For the *Stock* domain, we observe five reasons. (1) In many cases (46%) the inconsistency is due to *semantics ambiguity*. We consider semantics ambiguity as the reason if ambiguity is possible for the particular attribute and we observe inconsistency between values provided by the source and the dominant values on a large fraction of items of that attribute; we have given examples of ambiguity for Dividend and 52-week low price earlier. (2) The reason can also be *instance ambiguity* (6%), where a source interprets one stock symbol differently from the majority of sources; this happens mainly for stock symbols that terminated at some point. Recall that instance ambiguity results in the high deviation on Volume. (3) Another major reason is *out-of-date data* (34%): at the point when we collected data, the data were not up-to-date; for two thirds of the cases the data were updated hours ago, and for one third of the cases the data had not been refreshed for days. (4) There is one error on data unit: the majority reported 76M while one source re-

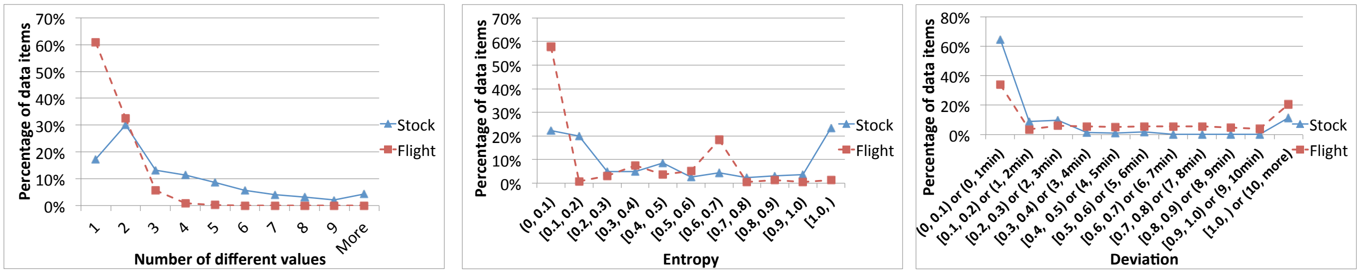
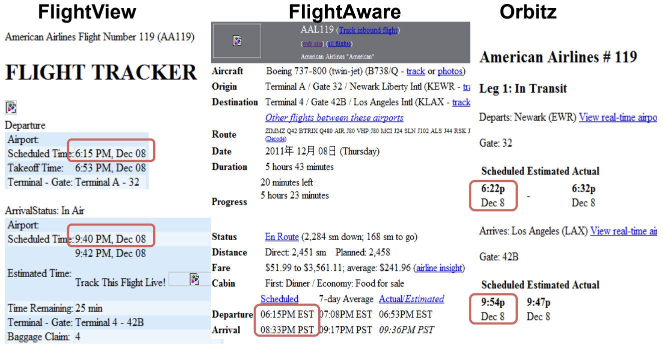


Figure 4: Value inconsistency: distribution of number of values, entropy of values, and deviation of numerical values.



FlightView

American Airlines Flight Number 119 (AA119)

FLIGHT TRACKER

Departure

Airport:
Scheduled Time: 6:15 PM, Dec 08
Takeoff Time: 6:53 PM, Dec 08
Terminal - Gate: Terminal A - 32

ArrivalStatus: In Air

Airport:
Scheduled Time: 9:40 PM, Dec 08
9:42 PM, Dec 08
Estimated Time:
Track This Flight Live!
Time Remaining: 25 min
Terminal - Gate: Terminal 4 - 42B
Baggage Claim: 4

FlightAware

AAL119 (Track inbound flight)
(web site) (all flights)
American Airlines "American"

Aircraft Boeing 737-800 (twin-jet) (B738/Q - track or photos)
Origin Terminal A / Gate 32 / Newark Liberty Intl (KEWR - tra
Destination Terminal 4 / Gate 42B / Los Angeles Intl (KLAX - track
Other flights between these airports
Route ZIMMZ Q42 BTRIX Q480 AIR J80 VHP J80 MCI J24 SLN J102 ALS J44 RSK J (Decode)
Date 2011年 12月 08日 (Thursday)
Duration 5 hours 43 minutes
Progress 20 minutes left
5 hours 23 minutes
Status En Route (2,284 sm down; 168 sm to go)
Distance Direct: 2,451 sm Planned: 2,458
Fare $51.99 to $3,561.11; average: $241.96 (airline insight)
Cabin First: Dinner / Economy: Food for sale

	Scheduled	7-day Average	Actual/*Estimated*
Departure	06:15PM EST	07:08PM EST	06:53PM EST
Arrival	08:33PM PST	09:17PM PST	*09:36PM PST*

Orbitz

American Airlines # 119

Leg 1: In Transit

Departs: Newark (EWR) View real-time airpo

Gate: 32

Scheduled	Estimated	Actual
6:22p Dec 8	-	6:32p Dec 8

Arrives: Los Angeles (LAX) View real-time ai

Gate: 42B

Scheduled	Estimated	Actual
9:54p Dec 8	9:47p Dec 8	

Figure 5: Screenshots of three flight sources.

ported 76B. (5) Finally, there are four cases (11%) where we could not determine the reason and it seems to be *purely erroneous data*.

For the *Flight* domain, we observe only three reasons. (1) *Semantics ambiguity* causes 33% of inconsistency: some source may report takeoff time as departure time and landing time as arrival time, while most sources report the time of leaving the gate or arriving at the gate. (2) *Out-of-date data* causes 11% of the inconsistency; for example, even when a flight is already canceled, a source might still report its actual departure and arrival time (the latter is marked as "estimated"). (3) *Pure errors* seem to cause most of the inconsistency (56%). For example, Figure 5 shows three sources providing different scheduled departure time and arrival time for Flight AA119 on 12/8/2011; according to the airline website, the real scheduled time is 6:15pm for departure and 9:40pm for arrival. For scheduled departure time, *FlightView* and *FlightAware* provide the correct time while *Orbitz* provides a wrong one. For scheduled arrival time, all three sources provide different times; *FlightView* again provides the correct one, while the time provided by *FlightAware* is unreasonable (it typically takes around 6 hours to fly from the east coast to the west coast in the US). Indeed, we found that *FlightAware* often gives wrong scheduled arrival time; if we remove it, the average number of values for *Scheduled arrival* drops from 1.65 to 1.31.

Dominant values: We now focus on the dominant values, those with the largest number of providers for a given data item. Similarly, we can define the *second dominant value*, etc. Figure 7 plots the distribution of the dominance factors and the precision of the dominant values with respect to different dominance factors.

For the *Stock* domain, we observe that on 42% of the data items the dominant values are supported by over 90% of the sources, and on 73% of the data items the dominant values are supported by over half of the sources. For these 73% data items, 98% of the dominant values are consistent with the gold standard. However, when the dominance factor drops, the precision is also much lower. For 9% of the data items with dominance factor of .4, the consistency already drops to 84%. For 7% of the data items where the dominance factor is .1, the precision for the dominant value, the second dominant value, and the third dominant value is .43, .33, and .12 respectively (meaning that for 12% of the data items none of the top-3 values is true).

For the *Flight* domain, more data items have a higher dominance factor–42% data items have a dominance factor of over .9, and 82% have a dominance factor of over .5. However, for these 82% items the dominant values have a lower precision: only 88% are consistent with the gold standard. Actually for the 11% data items whose dominance factor falls in $[.5, .6)$, the precision is only 50% for the dominant value. As we show later, this is because some wrong values are copied between sources and become dominant.

Summary and comparison: Overall we observe a fairly high inconsistency of values on the same data item: for *Stock* and *Flight* the average entropy is .58 and .24, and the average deviation is 13.4 and 13.1 respectively. The inconsistency can vary from attributes to attributes. There are different reasons for the inconsistency, including ambiguity, out-of-date data, and pure errors. For the *Stock* domain, half of the inconsistency is because of ambiguity, one third is because of out-of-date data, and the rest is because of erroneous data. For the *Flight* domain, 56% of the inconsistency is because of erroneous data.

If we choose dominant values as the true value (this is essentially the VOTE strategy, as we explain in Section 4), we can obtain a precision of 0.908 for *Stock* and 0.864 for *Flight*. We observe that dominant values with a high dominance factor are typically correct, but the precision can quickly drop when this factor decreases. Interestingly, the *Flight* domain has a lower inconsistency but meanwhile a lower precision for dominant values, mainly because of copying on wrong values, as we show later.

3.3 Source accuracy

Next, we examine the accuracy of the sources over time. Given a source S, we consider the following two measures.

- *Source accuracy:* We compute accuracy of S as the percentage of its provided true values among all its data items appearing in the gold standard.
- *Accuracy deviation:* We compute the standard deviation of the accuracy of S over a period of time. We denote by $\bar{T}$ the time points in a period, by $A(t)$ the accuracy of S at time $t \in \bar{T}$, and by $\hat{A}$ the mean accuracy over $\bar{T}$. The variety is computed by $\sqrt{\frac{1}{|\bar{T}|} \sum_{t \in \bar{T}} (A(t) - \hat{A})^2}$.

Source accuracy: Figure 8(a) shows the distribution of source accuracy in different domains. Table 4 lists the accuracy and item-level coverage of some authoritative sources.

In the *Stock* domain, the accuracy varies from .54 to .97 (except *StockSmart*, which has accuracy .06), with an average of .86. Only 35% sources have an accuracy above .9, and 3 sources (5%) have an accuracy below .7, which is quite low. Among the five popular

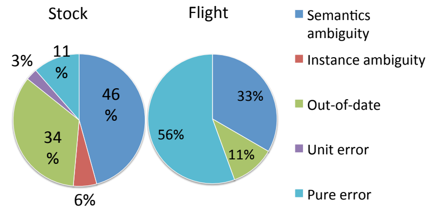


Figure 6: Reasons for value inconsistency.

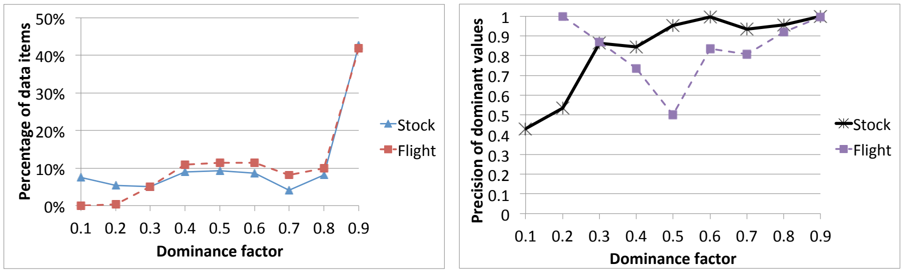
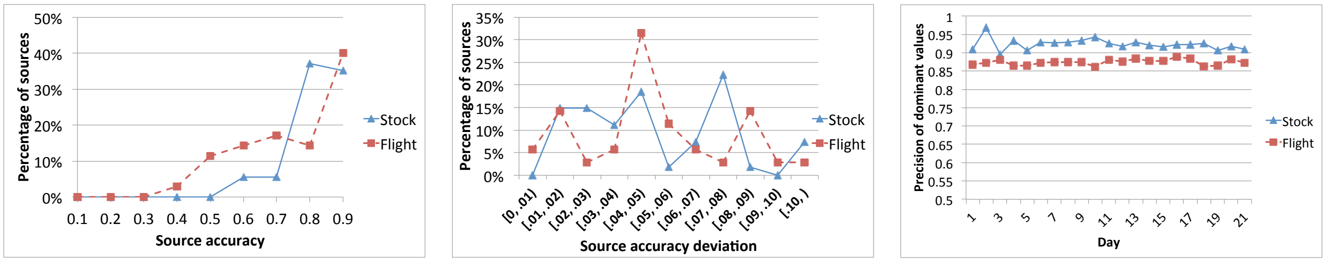


Figure 7: Dominant values.



(a) Distribution of source accuracy.

(b) Accuracy deviation over time.

(c) Dominant values over time.

Figure 8: Source accuracy and deviation over time.

Table 4: Accuracy and coverage of authoritative sources.

	Source	Accuracy	Coverage
Stock	*Google Finance*	.94	.82
	Yahoo! Finance	.93	.81
	NASDAQ	.92	.84
	MSN Money	.91	.89
	Bloomberg	.83	.81
Flight	*Orbitz*	.98	.87
	Travelocity	.95	.71
	Airport average	.94	.03

financial sources, four have an accuracy above .9, but *Bloomberg* has an accuracy of only .83 because it may apply different semantics on some statistical attributes such as EPS, P/E and Yield. All authoritative sources have a coverage between .8 and .9.

In the *Flight* domain, we consider sources excluding the three official airline websites (their data are used as gold standard). The accuracy varies from .43 to .99, with an average of .80. There are 40% of the sources with an accuracy above .9, but 10 sources (29%) have an accuracy below .7. The average accuracy of airport sources is .94, but their average coverage is only .03. Authoritative sources like *Orbitz* and *Travelocity* all have quite high accuracy (above .9), but *Travelocity* has low coverage (.71).

Accuracy deviation: Figure 8(b) shows the accuracy deviation of the sources in a one-month period, and Figure 8(c) shows the precision of the dominant values over time.

In the *Stock* domain, we observe that for 4 sources the accuracy varies tremendously (standard deviation over .1) and the highest standard deviation is as high as .33. For 59% of the sources the accuracy is quite steady (standard deviation below .05). We did not observe any common peaks or dips on particular days. The precision of the dominant values ranges from .9 to .97, and the average is .92. The day-by-day precision is also fairly smooth, with some exceptions on a few days.

In the *Flight* domain, we observe that for 1 source the accuracy varies tremendously (deviation .11), and for 60% sources the accuracy is quite steady (deviation below .05). The precision of the dominant values ranges from .86 to .89, and the average is .87.

Summary and comparison: We observe that the accuracy of the sources can vary a lot. On average the accuracy is not too high: .86 for *Stock* and .80 for *Flight*. Even authoritative sources may not have very high accuracy. We also observe that the accuracy is fairly steady in general. On average the standard deviation is 0.06 for *Stock* and 0.05 for *Flight*, and for about half of the sources the deviation is below .05 over time.

3.4 Potential copying

Just as copying is common between webpage texts, blogs, etc., we also observe copying between deep-web sources; that is, one source obtains some or all of its data from another source, while possibly adding some new data independently. We next report the potential copying we found in our data collections (Table 5) and study how that would affect precision of the dominant values. For each group $\bar{S}$ of sources with copying, we compute the following measures.

- *Schema commonality:* We measure schema commonality as the average Jaccard similarity between the sets of provided attributes on each pair of sources. If we denote by $\bar{A}(S)$ the set of global attributes that S provides, we compute schema commonality of $\bar{S}$ as $Avg_{S,S' \in \bar{S}, S \neq S'} \frac{|\bar{A}(S) \cap \bar{A}(S')|}{|\bar{A}(S) \cup \bar{A}(S')|}$.
- *Object commonality:* Object commonality is also measured by average Jaccard similarity but between the sets of provided objects.
- *Value commonality:* The average percentage of common values over all shared data items between each source pair.
- *Average accuracy:* The average source accuracy.

On the *Stock* domain, we found two groups of sources with potential copying. The first group contains 11 sources, with exactly the same webpage layout, schema, and highly similar data. These sources all derive their data from *Financial Content*, a market data service company, and their data are quite accurate (.92 accuracy). The second group contains 2 sources, also with exactly the same schema and data; the two websites are indeed claimed to be merged in 2009. However, their data have an accuracy of only .75. For each group, we keep only one randomly selected source and remove the rest of the sources; this would increase the precision of dominant values from .908 to .923.

Table 5: Potential copying between sources.

	Remarks	Size	Schema sim	Object sim	Value sim	Avg accu
Stock	Depen claimed	11	1	.99	.99	.92
	Depen claimed	2	1	1	.99	.75
Flight	Depen claimed	5	0.80	1	1	.71
	Query redirection	4	0.83	1	1	.53
	Depen claimed	3	1	1	1	.92
	Embedded interface	2	1	1	1	.93
	Embedded interface	2	1	1	1	.61

On the *Flight* domain, we found five groups of sources with potential copying. Among them, two directly claim partnership by including the logo of other sources; one re-directs its queries; and two embed the query interface of other sources. Sources in the largest two groups provide a little bit different sets of attributes, but exactly the same flights, and the same data for all overlapping data items. Sources in other groups provide almost the same schema and data. Accuracy of sources in these groups vary from .53 to .93. After we removed the copiers and kept only one randomly selected source in each group, the precision of dominant values is increased significantly, from .864 to .927.

Summary and comparison: We do observe copying between deep-web sources in each domain. In some cases the copying is claimed explicitly, and in other cases it is detected by observing embedded interface or query redirection. For the copying that we have observed, while the sources may provide slightly different schemas, they provide almost the same objects and the same values. The accuracy of the original sources may not be high, ranging from .75 to .92 for *Stock*, and from .53 to .93 for *Flight*. Because the *Flight* domain contains more low-accuracy sources with copying, removing the copied sources improves the precision of the dominant values more significantly than in the *Stock* domain.

4. DATA FUSION

As we have shown in Section 3, deep-web data from different sources can vary significantly and there can be a lot of conflicts. *Data fusion* aims at resolving conflicts and finding the true values. A basic fusion strategy that considers the dominant value (*i.e.*, the value with the largest number of providers) as the truth works well when the dominant value is provided by a large percentage of sources (*i.e.*, a high dominance factor), but fails quite often otherwise. Recall that in the *Stock* domain, the precision of dominant values is 90.8%, meaning that on around 1500 data items we would conclude with wrong values. Recently many advanced fusion techniques have been proposed to improve the precision of truth discovery [2, 3, 6, 7, 8, 10, 13, 14, 16, 17, 18, 19, 20].

In this section we answer the following three questions.

1. *Are the advanced fusion techniques effective?* In other words, do they perform (significantly) better than simply taking the dominant values or taking all data provided by the best source (assuming we know which source it is).
2. *Which fusion method is the best?* In other words, is there a method that works better than others on all or most data sets?
3. *Which intuitions for fusion are effective?* In other words, does each intuition for fusion improve the results?

This section first presents an overview of the proposed fusion methods (Section 4.1) and then compares their performance on our data collections (Section 4.2).

4.1 Review of data-fusion methods

In our data collections each source provides at most one value on a data item and each data item is associated with a single true value. We next review existing fusion methods suitable for this context. Before we jump into descriptions of each method, we first enumerate the many insights that have been considered in fusion.

- *Number of providers:* A value that is provided by a large number of sources is considered more likely to be true.
- *Trustworthiness of providers:* A value that is provided by trustworthy sources is considered more likely to be true.
- *Difficulty of data items:* The error rate on each particular data item is also considered in the decision.
- *Similarity of values:* The provider of a value v is also considered as a partial provider of values similar to v.
- *Formatting of values:* The provider of a value v is also considered as a partial provider of a value that subsumes v. For example, if a source typically rounds to million and provides "8M", it is also considered as a partial provider of "7,528,396".
- *Popularity of values:* Popularity of wrong values is considered in the decision.
- *Copying relationships:* A copied value is ignored in the decision.

All fusion methods more or less take a voting approach; that is, accumulating votes from providers for each value on the same data item and choosing the value with the highest vote as the true one. The vote count of a source is often a function of the trustworthiness of the source. Since source trustworthiness is typically unknown *a priori*, they proceed in an iterative fashion: computing value vote and source trustworthiness in each round until the results converge. We now briefly describe given a data item d, how each fusion method computes the vote count of each value v on d and the trustworthiness of each source s. In [12] we summarized equations applied in each method.

VOTE: Voting takes the dominant value as the true value and is the simplest strategy; thus, its performance is the same as the precision of the dominant values. There is no need for iteration.

HUB [11]: Inspired by measuring web page authority based on analysis of Web links, in HUB the vote of a value is computed as the sum of the trustworthiness of its providers, while the trustworthiness of a source is computed as the sum of the votes of its provided values. Note that in this method the trustworthiness of a source is also affected by the number of its provided values. Normalization is performed to prevent source trustworthiness and value vote counts from growing in an unbounded manner.

AVGLOG [13]: This method is similar to HUB but decreases the effect of the number of provided values by taking average and logarithm. Again, normalization is required.

INVEST [13]: A source "invests" its trustworthiness uniformly among its provided values. The vote of a value grows non-linearly with respect to the sum of the invested trustworthiness from its providers. The trustworthiness of source s is computed by accumulating the vote of each provided value v weighted by s's contribution among all contributions to v. Again, normalization is required.

POOLEDINVEST [13]: This method is similar to INVEST but the vote count of each value on item d is then linearly scaled such that the total vote count on d equals the accumulated investment on d. With this linear scaling, normalization is not required any more.

COSINE [10]: This method considers the values as a vector: for value v of data item d, if source s provides v, the corresponding position has value 1; if s provides another value on d, the position has value -1; if s does not provide d, the position has value

Table 6: Summary of data-fusion methods. X indicates that the method considers the particular evidence.

Category	Method	#Providers	Source trustworthiness	Item trustworthiness	Value Popularity	Value similarity	Value formatting	Copying
Baseline	Vote	X						
Web-link based	HUB	X	X					
	AvgLog	X	X					
	Invest	X	X					
	PooledInvest	X	X					
IR based	2-Estimates	X	X					
	3-Estimates	X	X	X				
	Cosine	X	X					
Bayesian based	TruthFinder	X	X			X		
	AccuPr	X	X					
	PopAccu	X	X		X			
	AccuSim	X	X			X		
	AccuFormat	X	X			X	X	
Copying affected	AccuCopy	X	X			X	X	X

0. Similarly the vectors are defined for selected true values. Cosine computes the trustworthiness of a source as the cosine similarity between the vector of its provided values and the vector of the (probabilistically) selected values. To improve stability, it sets the new trustworthiness as a linear combination of the old trustworthiness and the newly computed one.

2-Estimates [10]: 2-Estimates also computes source trustworthiness by aggregating value votes. It differs from HUB in two ways. First, if source s provides value v on d, it considers that s votes against other values on d and applies a complement vote on those values. Second, it averages the vote counts instead of summing them up. This method requires a complex normalization for the vote counts and trustworthiness to the whole range of $[0, 1]$.

3-Estimates [10]: 3-Estimates improves over 2-Estimates by considering also *trustworthiness* on each value, representing the likelihood that a vote on this value being correct. This measure is computed iteratively together with source trustworthiness and value vote count and similar normalization is applied.

TruthFinder [18]: This method applies Bayesian analysis and computes the probability of a value being true conditioned on the observed providers. In addition, TruthFinder considers similarity between values and enhances the vote count of a value by those from its similar values weighted by the similarity.

AccuPr [6]: AccuPr also applies Bayesian analysis. It differs from TruthFinder in that it takes into consideration that different values provided on the same data item are disjoint and their probabilities should sum up to 1; in other words, like 2-Estimates, 3-Estimates and Cosine, if a source s provides $v' \neq v$ on item d, s is considered to vote against v. To make the Bayesian analysis possible, it assumes that there are N false values in the domain of d and they are uniformly distributed.

PopAccu [9]: PopAccu augments AccuPr by removing the assumption of having n uniformly distributed false values. It computes value distribution from the observed data.

AccuSim [6]: AccuSim augments AccuPr by considering also value similarity in the same way as TruthFinder does.

AccuFormat: AccuFormat augments AccuSim by considering also formatting of values as we have described.

AccuCopy [6]: AccuCopy augments AccuFormat by considering the copying relationships between the sources and weighting the vote count from a source s by the probability that s provides the particular value independently. In our implementation we applied the copy detection techniques in [6], which treats sharing false values as strong evidence of copying.

Table 6 summarizes the features of different fusion methods. We can categorize them into five categories.

- *Baseline:* The basic voting strategy.
- *Web-link based:* The methods are inspired by measuring webpage authority based on Web links, including HUB, AvgLog, Invest and PooledInvest.
- *IR based:* The methods are inspired by similarity measures in Information Retrieval, including Cosine, 2-Estimates and 3-Estimates.
- *Bayesian based:* The methods are based on Bayesian analysis, including TruthFinder, AccuPr, PopAccu, AccuSim, and AccuFormat.
- *Copying affected:* The vote count computation discounts votes from copied values, including AccuCopy.

Finally, note that in each method we can distinguish trustworthiness for each attribute. For example, AccuFormatAttr distinguishes the trustworthiness for each attribute whereas AccuFormat uses an overall trustworthiness for all attributes.

4.2 Fusion performance evaluation

We now evaluate the performance of various fusion methods on our data sets. We focus on five measures.

- *Precision:* The precision is computed as the percentage of the output values that are consistent with a gold standard.
- *Recall:* The recall is computed as the percentage of the values in the gold standard being output as correct. Note that when we have fused all sources (so output all data items), the recall is equivalent to the precision.
- *Trustworthiness deviation:* Recall that except Vote, each method computes some trustworthiness measure of a source. We sampled the trustworthiness of each source with respect to a gold standard as it is defined in the method, and compared it with the trustworthiness computed by the method at convergence. In particular, given a source $s \in \mathcal{S}$, we denote by $T_{sample}(s)$ its sampled trustworthiness and by $T_{compute}(s)$ its computed trustworthiness, and compute the deviation as

$$dev(\mathcal{S}) = \sqrt{\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}(T_{sample}(s) - T_{compute}(s))^2}. \quad (4)$$

- *Trustworthiness difference:* The difference is computed as the average computed trustworthiness for all sources minus the average sampled trustworthiness.

Table 7: Precision of data-fusion methods on one snapshot of data. Highest precisions are in bold font and other top-3 precisions are in bold italic font.

Category	Method	*Stock* prec w. trust	*Stock* prec w/o. trust	*Stock* Trust dev	*Stock* Trust diff	*Flight* prec w. trust	*Flight* prec w/o. trust	*Flight* Trust dev	*Flight* Trust diff
Baseline	VOTE	-	.908	-	-	-	.864	-	-
Web-link based	HUB	.913	.907	.11	.08	.939	.857	.2	.14
	AVGLOG	.910	.899	.17	-.13	.919	.839	.24	.001
	INVEST	.924	.764	.39	-.31	.945	.754	.29	-.12
	POOLEDINVEST	.924	.856	1.29	0.29	.945	***.921***	17.26	7.45
IR based	2-ESTIMATES	.910	.903	.15	-.14	.87	.754	.46	-.35
	3-ESTIMATES	.910	.905	.16	-.15	.87	.708	.95	-.94
	COSINE	.910	.900	.21	-.17	.87	.791	.48	-.41
Bayesian based	TRUTHFINDER	.923	.911	.15	.12	***.957***	.793	.25	.16
	ACCUPR	.910	.899	.14	-.11	.91	.868	.16	-.06
	POPACCU	.909	.892	.14	-.11	***.958***	***.925***	.17	-.11
	ACCUSIM	.918	***.913***	.17	-.16	.903	.844	.2	-.09
	ACCUFORMAT	.918	.911	.17	-.16	.903	.844	.2	-.09
	ACCUSIMATTR	***.950***	***.929***	.17	-.16	.952	.833	.19	-.08
	ACCUFORMATATTR	***.948***	**.930**	.17	-.16	.952	.833	.19	-.08
Copying affected	ACCUCOPY	**.958**	.892	.28	-.11	**.960**	**.943**	.16	-.14

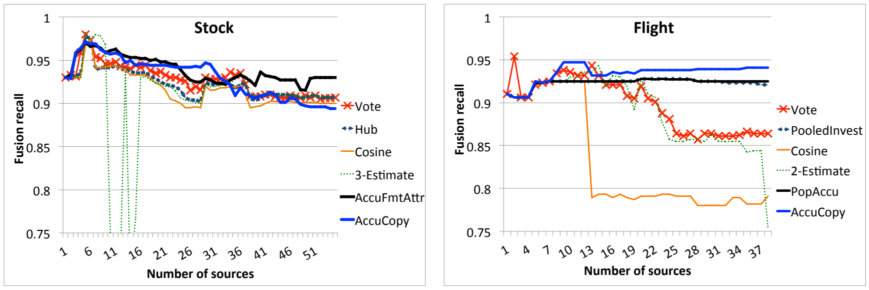


Figure 9: Fusion recall as sources are added.

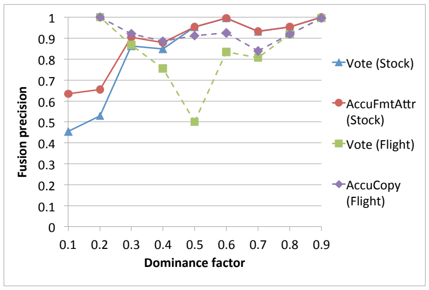


Figure 10: Precision vs. dominance factor.

- *Efficiency:* Efficiency is measured by the execution time on a Windows machine with Intel Core i5 processor (3.2GHz, 4MB cache, 4.8 GT/s QPI).

Precision on one snapshot: We first consider data collected on a particular day and use the same snapshots as in Section 3. For each data set, we computed the coverage and accuracy of each source with respect to the gold standard (as reported in Section 3), and then ordered the sources by the product of coverage and accuracy (*i.e.*, recall). We started with one source and gradually added sources according to the ordering, and measured the recall. We report the following results. First, Table 7 shows the final precision (*i.e.*, recall) with and without giving the sampled source trustworthiness as input, and the trustworthiness deviation and difference for each method in each domain. Second, Figure 9 shows the recall as we added sources on each domain; to avoid cluttering, for each category of fusion methods, we only plotted for the method with the highest final recall. Third, Table 8 compares pairs of methods where the second was intended to improve over the first. The table shows for each pair how many errors by the first method are corrected by the second and how many new errors are introduced. Fourth, to understand how the advanced fusion methods improve over the baseline VOTE, Figure 10 compares the precision of VOTE and the best fusion method in each domain with respect to dominance factor. Fifth, Figure 11 categorizes the reasons of mistakes for a randomly sampled 20 errors by the best fusion method for each domain.

Stock data: As shown in Table 7, for the *Stock* data ACCUFORMATATTR obtains the best results without input trustworthiness and it improves over VOTE by 2.4% (corresponding to about 350 data items). As shown in Figure 10, the main improvement occurs on the data items with dominance factor lower than .5. Note that on this data set the highest recall from a single source is .93, exactly the same as that of the best fusion results. From Figure 9 we observe that as sources are added, for most methods the recall peaks at the 5th source and then gradually decreases; also, we observe some big change for 3-ESTIMATE at the 11th-16th sources.

We next compare the various fusion methods. For this data set, only Bayesian based methods can perform better than VOTE; among other methods, Web-link based methods perform worst, then ACCUCOPY, then IR based methods (Table 7). ACCUCOPY does not perform well because it considers copying as likely between many pairs of sources in this data set; the major reason is that the copy-detection technique in [5] does not take into account value similarity, so it treats values highly similar to the truth still as wrong and considers sharing such values as strong evidence for copying. From Table 8, we observe that considering formatting and distinguishing trustworthiness for different attributes improve the precision on this data set, while considering trustworthiness at the data-item level (3-ESTIMATE) does not help much.

We now examine how well we estimate source trustworthiness and the effect on fusion. If we give the sampled source trustworthiness as input (so no need for iteration) and also ignore copiers in Table 5 when applying ACCUCOPY (note that there may be other copying that we do not know), ACCUCOPY performs the best (Table 7). Note that for all methods, giving the sampled trustworthiness improves the results. However, for most methods except INVEST, POOLEDINVEST and ACCUCOPY, the improvement is very small; indeed, for these three methods we observe a big trustworthiness deviation. Finally, for most methods except HUB, POOLEDINVEST and TRUTHFINDER, the computed trustworthi-

Table 8: Comparison of fusion methods.

Basic method	Advanced method	*Stock*			*Flight*		
		#Fixed errs	#New errs	ΔPrec	#Fixed errs	#New errs	ΔPrec
HUB	AvgLog	3	25	-.008	2	12	-.018
Invest	PooledInvest	376	121	+.09	101	10	+.167
2-Estimates	3-Estimates	6	2	+.002	70	95	-.046
TruthFinder	AccuSim	37	32	+.002	29	1	+.051
AccuPr	AccuSim	70	31	+.014	1	14	-.024
AccuPr	PopAccu	7	26	-.007	46	15	+.057
AccuSim	AccuSimAttr	47	3	+.016	5	11	-.011
AccuSimAttr	AccuFormatAttr	7	5	+.001	0	0	0
AccuFormatAttr	AccuCopy	33	136	-.038	70	10	+.11

ness is lower than the sampled one on average. This makes sense because when we make mistakes, we compute lower trustworthiness for most sources. TruthFinder tends to compute very high accuracy, on average .97, 14% higher than the sampled ones.

Finally, we randomly selected 20 data items on which AccuFormatAttr makes mistakes for examination (Figure 11). We found that among them, for 4 items AccuFormatAttr actually selects a value with finer granularity so the results cannot be considered as wrong. Among the rest, we would be able to fix 7 of them if we know sampled source trustworthiness, and fix 2 more if we are given in addition the copying relationships. For the remaining 7 items, for 1 item a lot of similar false values are provided, for 1 item the selected value is provided by high-accuracy sources, for 3 items the selected value is provided by more than half of the sources, and for 2 items there is no value that is dominant while the ground truth is neither provided by more sources nor by more accurate sources than any other value.

Flight data: As shown in Table 7, on the *Flight* data AccuCopy obtains the best results without input trustworthiness and it improves over Vote by 9% (corresponding to about 550 data items, half of the mistakes made by Vote). AccuCopy does not have that many false positives for copy detection as on *Stock* data because none of the attributes here is numerical, so similar values is not a potential problem (recall that [6] reports good results also on a domain with non-numerical values–*Book*). As shown in Figure 10, AccuCopy significantly improves the precision on data items with dominance factor in $[.4, .7)$, because it ignores copied values in fusion. Note that on this data set the highest recall from a single source is .91, 3.4% lower than the best fusion results. From Figure 9 we observe that as sources are added, for most methods the recall peaks at the 9th source and then drops a lot after low-quality copiers are added, but for AccuCopy, PopAccu and PooledInvest the recall almost flattens out after the 9th source; also, we observe a big drop for Cosine at the 14th source.

Among other methods, only PooledInvest, PopAccu and AccuPr perform better than Vote (Table 7). Actually, we observe that all methods perform better than Vote if sampled trustworthiness are given as input, showing that the problem lies in trustworthiness computation; this is because in this data set some groups of sources with copying dominate the values and are considered as accurate, while other sources that provide the true values are then considered as less accurate. This shows that if we are biased by low-quality copiers, considering source trustworthiness can bring even worse results, unless as PopAccu does, we keep into account the non-uniform distribution of false values and treat some copied values as popular false values. Interestingly, PooledInvest obtains the third best results on *Flight* data (but the second worst results on *Stock* data). Also, we observe that considering similarity and formatting, or distinguishing trustworthiness for each attribute does not improve the results for this data set (Table 8).

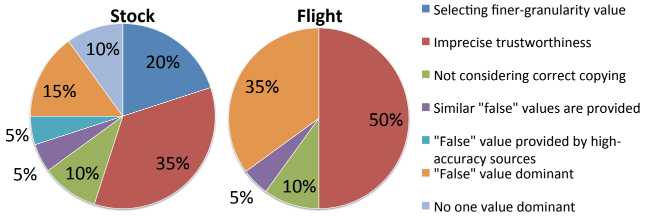


Figure 11: Error analysis of the best fusion method.

If we take input trustworthiness, AccuCopy performs the best (Table 7). All methods perform better with input trustworthiness and the improvement is big. As we have said, these are mainly because of bias from copied values. Again, except HUB, Invest, PooledInvest and TruthFinder, all other methods compute much lower trustworthiness than the sampled ones.

Finally, we randomly selected 20 data items on which AccuCopy makes mistakes for examination (Figure 11). We found that we would be able to fix 10 of them if we know precise source trustworthiness, and fix 2 more if we know correct copying relationships. For the remaining 8 items, for 1 item a lot of similar false values are provided, for 7 items the selected value is provided by more than half of the sources (the value provided by the airline website is in a minority and provided by at most 3 other sources).

Precision vs. efficiency: Next, we examined the efficiency of the fusion methods. Figure 12 plots the efficiency and precision of each method for each domain.

On the *Stock* data, Vote finished in less than 1 second; 8 methods finished in 1-10 seconds; 4 methods, including Invest, PooledInvest, 3-Estimate, Cosine, finished in 10-100 seconds but did not obtain higher precision; AccuSimAttr and AccuFormatAttr finished in 115 and 235 seconds respectively while obtained the highest precision; finally, AccuCopy finished in 855 seconds as it in addition computes copying probability between each pair of sources in each round, but its precision is low.

On the *Flight* data, which contains fewer sources and fewer data items than *Stock*, 4 methods including Vote finished in less than 1 second; 9 methods finished in 1-10 seconds; Invest and AccuFormatAttr finished in 11.7 and 17.3 seconds respectively but did not obtain better results; AccuCopy finished in 17 seconds and obtained the highest precision. On this data set AccuCopy did not spend much longer time than AccuFormatAttr although it in addition computes copying probabilities, because (1) there are fewer sources and (2) it converges in fewer rounds.

Precision over time: Finally, we ran the different fusion methods on data sets collected on different days. Table 9 shows for each method a summary, including average precision, minimum precision, and standard deviation on fusion precision over time.

Our observation in general is consistent with the results on one snapshot of the data. AccuFormatAttr is the best for the *Stock* domain and obtains a precision of .941 on average, whereas Ac-

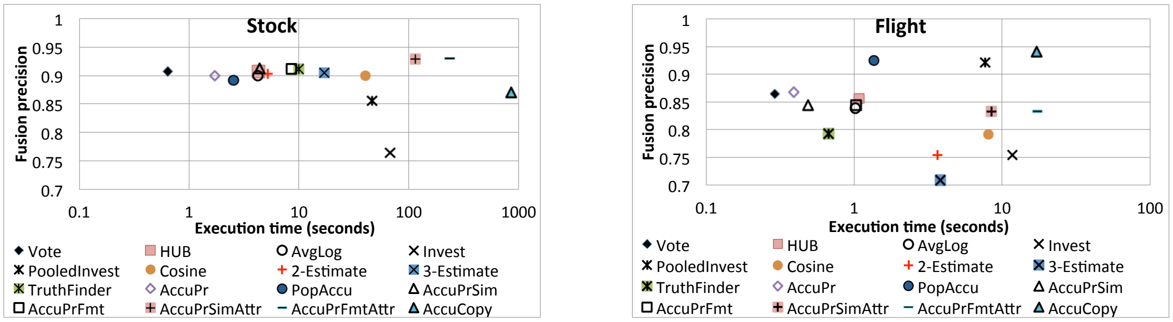


Figure 12: Fusion precision vs. efficiency.

cuCopy is the best for the *Flight* domain and the precision is as high as .987. The major difference from observations on the snapshots is that AccuFormatAttr and AccuSimAttr outperform Vote on average on Flight domain. Finally, we observe higher deviation for *Flight* than for *Stock*, caused by the variety of quality of copied data; we also observe a quite high deviation for Cosine model on *Flight* data.

Summary and comparison: We found that in most data sets, the naive voting results have an even lower recall than the highest recall from a single source, while the best fusion method improves over the highest source recall on average. We obtain very high precision for *Flight* (.987) and a reasonable precision for *Stock* (.941). Note however that for *Stock* the improvement of recall over a single source with the highest recall is only marginal. Also, on all data snapshots we observe that fusing a few high-recall sources (5 for *Stock*, 9 for *Flight*) obtains the highest recall, while adding more sources afterwards can only hurt (reducing by 4% for *Stock* and by .4% for *Flight* on the snapshot). Among the mistakes, we found that about 50% can be fixed by correct knowledge of source trustworthiness and copying; for 10% the selected values have a higher granularity than the ground truth (so not erroneous); and for the remaining 40% we do not observe strong evidence from the data supporting the ground truth.

The *Stock* data and the *Flight* data represent two types of data sets. The one represented by *Flight* has copying mainly between low-accuracy sources. On such data sets, considering source accuracy without copying can obtain results with even lower precision, while incorporating knowledge about copying can significantly improve the results. The data sets represented by *Stock* have copying mainly among high-accuracy sources. In this case, ignoring copying does not seem to hurt the fusion results much, whereas considering copying should further improve the results; this is shown by the fact that Vote improves from .908 to .923 when excluding copiers, and that AccuCopy obtains the highest precision (.958) among all methods when we take sampled source accuracy and discovered copying as input. Note however that the low performance of AccuCopy on *Stock* is because the copy-detection method does not handle similar values well, so it generates lots of false positives in copy detection. We note that other differences between the two domains do not seem to affect the results significantly (*e.g.*, despite a higher heterogeneity and more numerical values on *Stock*, most methods obtain a better results on *Stock* data), so we expect that our observations can generalize to other data sets.

Among the different fusion methods, we did not observe that one definitely dominates others on all data sets. Similarly, for fusion-method pairs listed in Table 8, it is not clear that the advanced method would definitely improve over the basic method on all data sets except for Invest vs. PooledInvest, and TruthFinder vs. AccuSim. For example, distinguishing trustworthiness for different attributes helps on *Stock* data but not on *Flight* data. However, AccuSimAttr and AccuFormatAttr in general obtain higher precision than most other methods in both domains. Typically more complex fusion methods achieve a higher fusion precision at the expense of a (much) longer execution time. This is affordable for off-line fusion. Certainly, longer execution time does not guarantee better results.

The fusion results without input trustworthiness depend both on how well the model performs if source trustworthiness is given and on how well the model can estimate source trustworthiness. In general the lower trustworthiness deviation, the higher fusion precision, but there are also some exceptions.

5. FUTURE RESEARCH DIRECTIONS

Based on our observations described in Sections 3-4, we next point out several research directions to improve data fusion and data integration in general.

Improving fusion: First, considering source trustworthiness appears to be promising and can often improve over naive voting when there is no bias from copiers. However, we often do not know source trustworthiness *a priori*. Currently most proposed methods start from a default accuracy for each source and then iteratively refine the accuracy. However, trustworthiness computed in this way may not be precise and it appears that knowing precise trustworthiness can fix nearly half of the mistakes in the best fusion results. Can we start with some seed trustworthiness better than the currently employed default values to improve fusion results? For example, the seed can come from sampling or based on results on the data items where data are fairly consistent.

Second, we observed that different fractions of data from the same source can have different quality. The fusion results have shown the promise of distinguishing quality of different attributes. On the other hand, one can imagine that data from one source may have different quality for data items of different categories; for example, a source may provide precise data for UA flights but low-quality data for AA-flights. Can we automatically detect such differences and distinguish source quality for different categories of data for improving fusion results?

Third, we neither observed one fusion method that always dominates the others, nor observed between a basic method and a proposed improvement that the latter always beats the former. Can we combine the results of different fusion models to get better results?

Fourth, for both data sets we assumed that there is a single true value for each data item, but in the presence of semantics ambiguity, one can argue that for each semantics there is a true value so there are multiple truths. Current work that considers precision and recall of sources for fusion [20] does not apply here because each source typically applies a single semantics for each data item. Can we effectively find all correct values that fit at least one of the semantics and distinguish them from false values?

Table 9: Precision of data-fusion methods on data over one month. Font usage is similar to Table 7.

Category	Method	*Stock*			*Flight*		
		Avg	Min	Deviation	Avg	Min	Deviation
Baseline	VOTE	.922	.898	.014	.887	.861	.028
Web-link based	HUB	.925	.895	.015	.885	.850	.027
	AVGLOG	.921	.895	.015	.868	.838	.029
	INVEST	.797	.764	.027	.786	.748	.032
	POOLEDINVEST	.871	.831	.015	***.979***	.921	.013
IR based	2-ESTIMATES	.910	.811	.026	.639	.588	.052
	3-ESTIMATES	.923	.897	.014	.718	.638	.034
	COSINE	.923	.894	.015	.880	.786	.086
Bayesian based	TRUTHFINDER	.930	.909	.013	.818	.777	.031
	ACCUPR	.922	.893	.015	.893	.861	.030
	POPACCU	.912	.884	.016	***.972***	.779	.048
	ACCUSIM	***.932***	.913	.012	.866	.833	.032
	ACCUFORMAT	***.932***	.911	.012	.866	.833	.032
	ACCUSIMATTR	**.941**	.921	.011	.956	.833	.050
	ACCUFORMATATTR	**.941**	.924	.010	.956	.833	.050
Copying affected	ACCUCOPY	.884	.801	.036	**.987**	.943	.010

Improving integration: First, source copying not only appears promising for improving data fusion (ACCUCOPY obtains the highest precision on *Flight*), but has many other potentials to improve various aspects of data integration [1]. However, the copy-detection method proposed in [6] falls short in the presence of numerical values as it ignores value similarity and granularity. Can we develop more robust copy-detection methods in such context? In addition, copy detection appears to be quite time-consuming. Can we improve the scalability of copy detection for Web-scale data?

Second, even though we have tried our best to resolve heterogeneity at the schema level and instance level manually, we still observed that 50% of value conflicts are caused by ambiguity. In fact, observing a lot of conflicts on an attribute from one source is a red flag for the correctness of schema mapping, and observing a lot of conflicts on an object from one source is a red flag for the correctness of instance mapping. Can we combine schema mapping, record linkage, and data fusion to improve results of all of them?

Third, on both data sets we observed that fusion on a few high recall sources obtains the highest recall, but on all sources obtains a lower recall. Such quality deterioration can also happen because of mistakes in instance de-duplication and schema mapping. This calls for source selection–can we automatically select a subset of sources that lead to the best integration results?

Improving evaluation: No matter for data fusion, instance deduplication, or schema mapping, we often need to evaluate the results of applying particular techniques. One major challenge in evaluation is to construct the gold standard. In our experiments our gold standards trust data from certain sources, but as we observed, this sometimes puts wrong values or coarse-grained values in the gold standard. Can we improve gold standard construction and can we capture our uncertainty for some data items in the gold standard? Other questions related to improving evaluation include automatically finding and explaining reasons for mistakes and reasons for inconsistency of data or schema.

6. CONCLUSIONS

This paper is the first one that tries to understand correctness of data in the Deep Web. We collected data in two domains where we believed that the data should be fairly clean; to our surprise, we observed data of quite high inconsistency and found a lot of sources of low quality. We also applied state-of-the-art data fusion methods to understand whether current techniques can successfully resolve value conflicts and find the truth. While these methods show good potential, there is obvious space for improvement and we suggested several promising directions for future work.

7. REFERENCES

[1] L. Berti-Equille, A. D. Sarma, X. L. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*, 2009.

[2] L. Blanco, V. Crescenzi, P. Merialdo, and P. Papotti. Probabilistic models to reconcile complex data from inaccurate data sources. In *CAiSE*, 83–97, 2010.

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

[4] N. Dalvi, A. Machanavajjhala, and B. Pang. An analysis of structured data on the web. *PVLDB*, 5(7):680–691, 2012.

[5] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 3(1):1358-1369, 2010.

[6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1):550–561, 2009.

[7] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1):562–573, 2009.

[8] X. L. Dong and F. Naumann. Data fusion–resolving data conflicts for integration. *PVLDB*, 2(2):1654–1655, 2009.

[9] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6(2), 2013.

[10] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, 131–140, 2010.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, 668–677, 1998.

[12] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava. Truth Finding on the Deep Web: Is the Problem Solved? http://lunadong.com/publication/webfusion_report.pdf.

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

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

[15] D. Srivastava and S. Venkatasubramanian. Information theory for data management. *PVLDB*, 2(2):1662–1663, 2009.

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

[17] M. Wu and A. Marian. A framework for corroborating answers from multiple web sources. *Inf. Syst.*, 36(2):431–449, 2011.

[18] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowl. Data Eng.*, 20:796–808, 2008.

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

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