Compact Explanation of Data Fusion Decisions

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

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

ABSTRACT

Data fusion aims at resolving conflicts between different sources when integrating their data. Recent fusion techniques find the truth by iterative Bayesian analysis that reasons about trustworthiness of sources and copying relationships between them. Providing explanations for such decisions is important, but can be extremely challenging because of the complexity of Bayesian analysis and that of the iterations during decision making.

This paper proposes two types of explanations for data-fusion results: *snapshot explanations* target casual users, taking the provided data and any other decision inferred from the data as evidence; *comprehensive explanations* target advanced users, taking only the provided data as evidence. We propose techniques that can efficiently generate correct and compact snapshot explanations and comprehensive explanations. Experimental results show that (1) the explanations we generate are correct, (2) our techniques can significantly reduce the sizes of the explanations, and (3) we can generate the explanations efficiently.

1. INTRODUCTION

Despite the abundance of useful information on the Web, different sources often provide conflicting data, some being out-of-date, inaccurate, or erroneous. *Data fusion* (see [3] for a survey) aims at resolving conflicts between different sources in data integration and creating consistent and clean data that best reflect the real world. An easy way to resolve conflicts is to apply voting, choosing the value provided by the most sources. This often leads to incorrect results, so recently proposed fusion techniques consider in addition trustworthiness of the providers and copying relationships between them in finding the truth [2, 7, 9, 10, 12, 19, 20, 22, 23, 24].

In real systems, simply presenting data fusion results is often inadequate; curious users ask not only "what" but also "why". They may raise questions such as "*Why is this value provided by fewer sources but considered true?*", "*Why is this source considered as a copier of that one?*", and "*Why is that source considered as more accurate?*". Only when we are able to provide convincing explanations, the users would believe us. Administrators of data fusion systems may wish to see such explanations as well, so they can diagnose and debug potential problems.

Table 1: Data from five sources on the affiliation of five DB researchers. False values are in italic font.

	S_1	S_2	S_3	S_4	S_5
Stonebraker	MIT	*berkeley*	MIT	MIT	*MS*
Dewitt	MSR	msr	*UWisc*	*UWisc*	*UWisc*
Bernstein	MSR	msr	MSR	MSR	MSR
Carey	UCI	*at&t*	*BEA*	*BEA*	*BEA*
Halevy	Google	google	*UW*	*UW*	*UW*

Explaining such decisions is not only important, but also extremely challenging, especially to non-technical users. There are two reasons for this. First, *Bayesian analysis* is conducted for decision making, including deciding the true value, judging whether a source copies from another, and so on. Unlike conventional (provenance-style) reasoning, Bayesian analysis considers all alternate decisions, computes the inverse probability of the observed fact conditioned on each decision, and then computes the probability of each alternative accordingly. We are not aware of any existing techniques that explain Bayesian reasoning ([11, 17] explained evidence propagation in Bayesian networks, which is different). As we illustrate next, a detailed description of the underlying Bayesian analysis can be hard to understand and frustrating to most users.

EXAMPLE 1.1. *Consider data provided by five sources on the affiliation of five DB researchers (Table 1). Source S_1 provides all correct affiliations; S_2 provides affiliation names in lower case; S_4 and S_5 copy from S_3, while S_5 provides the value for* Stonebraker *independently. We are able to find all correct affiliations if we apply the Bayesian analysis in [9], but a curious user may ask "Why is* UCI *considered as the correct affiliation of* Carey*?" Suppose we know the accuracy of the sources and probability of copying between sources (we explain in Sec.2 how we may obtain them), a detailed (and possibly agonizing) explanation can go like this.*

Three values are provided for Carey's affiliation.

If UCI is true, then we reason as follows.

1. Source S_1 provides the correct value. Since S_1 has accuracy .97, the probability that it provides this correct value is .97.
2. Source S_2 provides a wrong value. Since S_2 has accuracy .61, the probability that it provides a wrong value is $1-.61 = .39$. If we assume there are 100 uniformly distributed wrong values in the domain, the probability that S_2 provides the particular wrong value AT&T is $\frac{.39}{100} = .0039$.
3. Source S_3 provides a wrong value. Since S_3 has accuracy .4, the probability that it provides BEA is $\frac{1-.4}{100} = .006$.[1]

[1] We have omitted some repeating words and some details that would appear in the explanation to save space.

4. Source S_4 either provides a wrong value independently or copies this wrong value from S_3. It has probability .98 to copy from S_3 so probability $1 - .98 = .02$ to provide the value independently. In this case, its accuracy is .4 so the probability that it provides BEA is .006.
5. Source S_5 either provides a wrong value independently or copies this wrong value from S_3 or S_4. It has probability .99 to copy from S_3 and probability .99 to copy from S_4, so probability $(1 - .99)(1 - .99) = .0001$ to provide the value independently. In this case, its accuracy is .21, so the probability that it provides BEA is .0079.

Thus, the probability of our observed data conditioned on UCI being true is $.97 * .0039 * .006 * .006^{.02} * .0079^{.0001} = 2.1 * 10^{-5}$.

If AT&T is true, then ...; thus, the probability of our observed data conditioned on AT&T being true is $9.9 * 10^{-7}$.

If BEA is true, then ...; thus, the probability is $4.6 * 10^{-7}$.

If none of the provided values is true, then ...; thus, the probability is $6.3 * 10^{-9}$.

After we apply the Bayesian analysis, assuming the same a-priori probability for each value in the domain (100+1=101 values) to be true, we compute the probability of UCI being true conditioned on our observed data as $\frac{2.1*10^{-5}}{2.1*10^{-5}+9.9*10^{-7}+4.6*10^{-7}+6.3*10^{-9}*98} = .91$; that of AT&T as .04; that of BEA as .02; and that of one of the $100 - 2 = 98$ unprovided values as .0003. Thus, UCI has the highest probability to be the true value.

Obviously, such an explanation is too difficult for ordinary users to understand; even for technical users, the explanation gives too many details unnecessarily and is extremely verbose.

A much simpler explanation might be "(1) S_1, the provider of value UCI, has the highest accuracy, and (2) copying is very likely between S_3, S_4, and S_5, the providers of value BEA"*. For many users, this explanation is much easier to understand and the level of detail is adequate (further details can be provided on demand). However, automatically extracting such* key *evidence from the detailed description of the Bayesian analysis is not easy.* □

The second reason for the challenging nature of data-fusion explanations is that there can be different components in data fusion, such as quantifying trustworthiness of data sources, detecting copying between sources, and finding true values. These tasks are interdependent; advanced fusion techniques *iteratively* perform them until the results converge. Existing work on explaining iterative reasoning (*e.g.*, [21]) provides exhaustive answers, such as finding all extraction patterns that contribute to an extracted tuple in data extraction, but does not show how to explain the iterative process.

EXAMPLE 1.2. *Continue with Ex.1.1. Given the proposed explanation, an advanced user may further wonder (1) why S_1 is considered as having a higher accuracy than other sources and (2) why copying is considered likely between $S_3 - S_5$.*

While we can certainly continue answering these questions, careful choices need to be made in the explanations. Taking the copying between S_3 and S_4 as an example, the explanation might be "S_3 and S_4 share all five values, and especially, make the same three mistakes UWisc, BEA, UW; this is unusual for independent sources so copying is likely". *This explanation would further trigger explanation for why* UWisc, BEA, UW *are considered as wrong. However, recall that one of the reasons for* BEA *to be considered as wrong (*i.e., UCI *being correct) is the copying between $S_3 - S_5$, so we end up with a loop.*

On the other hand, if we provide provenance-style explanation and trace back the iterations (see Fig.1), the explanation again can be verbose, containing a lot of highly similar fragments. □

In this paper we propose two types of explanations. Targeting casual users who wish to understand a decision, we provide *snapshot explanations* that take the provided data and any *other* decision inferred from the data as evidence. The explanation in Ex.1.1 is a snapshot explanation, as the two reasons it gives are both inferred from the data. Targeting advanced users or system administrators, we provide *comprehensive explanations* that take only the provided data as evidence and explain any decision that requires inference over the data. Essentially, we study how to *find and organize evidence that we would show in each type of explanation.*

We have three goals in producing such explanations. First, the evidence we show should be consistent with the Bayesian analysis and gives the *correct* reasoning. For example, Bayesian analysis considers various alternate choices and reasons about them using all available positive and negative evidence, so showing only the positive evidence to explain a decision is inappropriate. Second, rather than providing a big chunk of evidence that contains every detail of the Bayesian reasoning but can be long and overwhelming, it is desirable that the evidence lists are *succinct*. Third, explanations are often generated at runtime on users' demand; thus, the evidence should be selected *efficiently*.

To the best of our knowledge, this paper is the first that aims at explaining data fusion decisions made by iterative Bayesian analysis. In particular, we make the following contributions.

1. We propose *explaining our decisions* for casual users by snapshot explanations, which list both positive and negative *evidence* considered in Bayesian analysis. We show how we can efficiently shorten such explanations by categorizing and aggregating evidence and selectively removing *unimportant* evidence.
2. We propose *explaining our (snapshot) explanations*[2] for advanced users by comprehensive explanations, which construct a DAG (directed acyclic graph) where children nodes represent evidence for the parent nodes according to the iterations. We show how we can efficiently shorten such explanations by considering only the *critical* points at which we change our decision in the iterations.
3. We show through experiments on real-world data that (1) we generate correct explanations, (2) our techniques can significantly reduce the size of the explanations, and (3) our algorithms are efficient.

Note that our techniques focus on finding the evidence that we would show to the user in the explanations. Making the evidence list *correct* and *succinct* is critical for generating intuitive explanations and has been a goal for explanation in the literature [14, 15]. On the other hand, how to present the evidence (*i.e.*, which words and layout to use, whether to use text, tables, or graphs) to improve the understandability of the explanations is beyond the scope of this paper. We have implemented our techniques for snapshot explanations in the SOLOMON system[3] [8] and demonstrated a text presentation and a graph presentation.

Although our techniques are based on existing work on data fusion [2, 7, 9, 12, 19, 20, 23, 24], none of the core ideas, including how to explain iterative Bayesian analysis and how to efficiently shorten such explanations, has been discussed in any previous work. In addition, our ideas for snapshot explanations can be applied in explaining other Bayesian-analysis results (*e.g.*, classification), and our ideas for comprehensive explanations can be applied in explaining iterative reasoning involving confidence or probabilities (*e.g.*, iterative information extraction such as [1]).

[2] Lord Byron wrote in *Don Juan* "I wish he would explain his explanation."

[3] http://www2.research.att.com/∼yifanhu/SourceCopying/

In the rest of the paper, Sec.2 defines our problem and briefly reviews data fusion techniques. Sec.3-4 describe generation of snapshot and comprehensive explanations. Sec.5 presents experimental results. Sec.6 discusses related work and Sec.7 concludes.

2. PROBLEM DEFINITION

We first formally define our explanation problems and then briefly review advanced data fusion techniques.

2.1 Problem definition

This paper studies how to explain iterative Bayesian analysis. We consider two types of explanations. First, *Snapshot explanations* are targeted to casual users who wish to understand a decision and would believe us on any other decision. The explanation can thus take any decision in fusion other than the one for explanation as supporting evidence.

DEFINITION 2.1 (SNAPSHOT EXPLANATION). *Given a decision W in data fusion, a* snapshot explanation *for W takes the provided data and all decisions in fusion except W as evidence and explains how Bayesian analysis leads to decision W.* □

Second, *comprehensive explanations* are targeted to advanced users who wish to understand also how we come to other conclusions required as evidence for explaining a decision. The explanation thus takes only the data as evidence. Such explanations can also help system administrators understand and debug the results.

DEFINITION 2.2 (COMPREHENSIVE EXPLANATION). *Given a decision W in data fusion, a* comprehensive explanation *for W takes only the provided data as evidence and explains how iterative Bayesian analysis leads to decision W.* □

Ex.1.1 illustrates a snapshot explanation, and the comprehensive explanation would further explain why S_1 is considered more accurate and why copying is considered likely between $S_3 - S_5$.

For example, as illustrated in the introduction, the snapshot explanation for "*Why is v the correct value even though it is provided by fewer sources than v'*" can be "*(1) many providers of v' copy it from the same source and (2) v is provided by several high-accuracy independent sources.*". However, the comprehensive explanation needs to go further and explain why some providers of v' are considered as copiers and why some providers of v are considered as more accurate and independent.

Before we describe how we generate such explanations, we first briefly review advanced techniques for data fusion.

2.2 Preliminaries for data fusion

Consider a set $\mathcal{D}$ of *data items*, each representing a particular aspect of a real-world object (*e.g.*, the affiliation of a researcher) and having a single true value. Also consider a set $\mathcal{S}$ of *data sources* that provide data on these data items. For the same item, different sources may provide conflicting values. Data fusion aims at *finding the true value for each item according to the provided values*.

Advanced fusion techniques [2, 7, 9, 12, 19, 20, 23, 24] find the true value on data item $D \in \mathcal{D}$ by conducting Bayesian analysis: it computes the probability that the observed data on D are provided conditioned on each value in D's domain being true, and selects the value that corresponds to the highest probability. In probability computation, it considers the following aspects.

1. *Source accuracy:* The probability that a source S provides a true value depends on its *accuracy*: the higher the accuracy, the higher the probability (similar for providing a particular false value). The accuracy of S is computed as the average probability of S's values being true in [9, 23].
2. *Copying relationship:* When computing the conditional probability of our observation, we wish to consider a source only if it provides an examined value independently of any other source. Copying is considered likely if we observe a lot of common unpopular data, especially common false values, since it is typically much less likely for independent sources to share such data.

There is inter-dependence between truth discovery, copy detection, and source accuracy; techniques in [2, 9, 12, 20, 23] conduct iterative computation until the results converge.

In this report we focus on explaining Bayesian analysis for truth discovery and copy detection.[4] We next give more details on Bayesian analysis for these two types of decisions.

Truth discovery: Let D be a data item and $\mathcal{V}(D)$ be the domain of D. Let Φ_D denote our observation on data provided for D and $\Phi_D(S)$ denote the observation for source S on D. [9, 23] follow the Bayes rule, compute for each value $v \in \mathcal{V}(D)$ the probability of it being true conditioned on Φ_D, and take the value with the highest probability as the true value. In particular, assuming the same a-priori probability for each value in $\mathcal{V}(D)$ leads to

$$P(v\, true|\Phi_D) = \frac{P(\Phi_D|v\, true)}{\sum_{v' \in \mathcal{V}(D)} P(\Phi_D|v'\, true)}; \tag{1}$$

$$P(\Phi_D|v\, true) = \Pi_{S \in \mathcal{S}} P(\Phi_D(S)|v\, true)^{I(\Phi_D(S))}. \tag{2}$$

Computation of Eq.(2) considers two aspects. First, computing $P(\Phi_D(S)|v\, true)$ considers the accuracy of S, denoted by $A(S)$: the higher $A(S)$, the more likely that S provides a true value v. Accuracy $A(S)$ is computed as the average probability of S's values being true in [9, 23]. Second, $I(\Phi_D(S))$ denotes the probability of S providing data on D independently of any other source and downweights copied values (details in [9]). We describe source copying between a pair of sources in more detail next.

Copying detection: Let Φ be our observation of the data. Let $S \rightarrow S'$ denote that S copies from S' and $S \perp S'$ denote that S and S' do not copy from each other; then, $P(S \rightarrow S') + P(S' \rightarrow S) + P(S \perp S') = 1$ (no-loop copying is assumed in previous work; that is, $S \rightarrow S'$ and $S' \rightarrow S$ do not happen together). Techniques in [2, 7, 9] follow the Bayes rule and compute the probability of each case conditioned on Φ. For example,

$$P(S \perp S'|\Phi) = \frac{\beta P(\Phi|S \perp S')}{\alpha P(\Phi|S \rightarrow S') + \alpha P(\Phi|S' \rightarrow S) + \beta P(\Phi|S \perp S')}. \tag{3}$$

Here, $0 < \alpha < .5$ is the a-priori probability of a source copying from another and $\beta = 1 - 2\alpha$. Assuming independence between different data items, the probability of observation Φ can be computed as the product of the probabilities on each data item $D \in \mathcal{D}$. In particular, let $S \not\rightarrow S'$ denote that S does not copy from S' and $\Phi_D(S)$ denote the observation for S on D, then

$$P(\Phi|S \perp S') = \Pi_{D \in \mathcal{D}} P(\Phi_D(S)|S \not\rightarrow S') P(\Phi_D(S')|S' \not\rightarrow S); \tag{4}$$

$$P(\Phi|S \rightarrow S') = \Pi_{D \in \mathcal{D}} P(\Phi_D(S)|S \rightarrow S') P(\Phi_D(S')|S' \not\rightarrow S). \tag{5}$$

When computing $P(\Phi_D(S)|S \not\rightarrow S')$, [7] considers (but is not limited to) three aspects: the probability of S providing data on D, that of S providing a particular value v, and that of S using a particular format f. The product of these probabilities is taken:

[4] Explaining source accuracy is not the focus of this paper as the computation does not involve Bayesian analysis; as an example, the explanation can show the histogram of the probabilities of the values from the source.

$$P(\Phi_D(S)|S \not\rightarrow S') = P(\Phi_D(S) \neq \emptyset|S \not\rightarrow S')$$
$$\cdot P(val(\Phi_D(S)) = v|S \not\rightarrow S') \cdot P(fmt(\Phi_D(S)) = f|S \not\rightarrow S'). \quad (6)$$

Here, $val(\Phi_D(S))$ denotes the observed value and $fmt(\Phi_D(S))$ denotes the observed format. Details of how each probability is computed can be found in [7].

When computing $P(\Phi_D(S)|S \rightarrow S')$, note that a copier may or may not copy on a particular data item, and if it copies the value, it may or may not keep the same format. [7] considers the *selectivity* (probability of copying on a data item), denoted by s, and the probability of keeping the same format in copying, denoted by k ($0 \leq s, k \leq 1$ and [7, 9] discussed how to set them). As an example, computation of the conditional probability of S providing the same value as S' but using a different format considers the possibility that S provides the data item independently (with probability $1 - s$) and the possibility that S copies the data item from S' but changes the format (with probability $s(1 - k)$); thus,

$$P(\Phi_D(S)|S \rightarrow S') = (1 - s)P(\Phi_D(S)|S \rightarrow S', D \text{ not copied})$$
$$+ s(1 - k)P(fmt(\Phi_D(S)) = f|S \rightarrow S', D \text{ not copied}). \quad (7)$$

As another example, computation of the conditional probability of S providing a different value only considers the possibility that S provides the data item independently (with probability $1 - s$):

$$P(\Phi_D(S)|S \rightarrow S') = (1 - s)P(\Phi_D(S)|S \rightarrow S', D \text{ not copied}). \quad (8)$$

Consider the probability that S provides a rare data item, provides an unpopular value (e.g., a particular false value), or uses an unpopular format. When S' has the same behavior, this probability conditioned on $S \rightarrow S'$ can be much higher than that conditioned on $S \not\rightarrow S'$, so such observations serve as strong evidence for copying. We next illustrate the Bayesian reasoning by an example.

EXAMPLE 2.3. *Continue with Ex.1.1 and consider the copying relationship between S_1 and S_2. We observe that they share neither rare data items nor false values (all values they share are correct) and they use different formats, so copying is unlikely. With $\alpha = .25, s = k = .8$, the Bayesian analysis goes as follows.*

We start with computation of $P(\Phi|S_1 \perp S_2)$, which requires computing $P(\Phi_D(S_1)|S_1 \not\rightarrow S_2)$ and $P(\Phi_D(S_2)|S_2 \not\rightarrow S_1)$ for each $D \in \mathcal{D}$ (Eq.(4)). All values S_1 provides are correct. Assuming we have decided that the accuracy of S_1 is .97, then the probability for S_1 to provide a true value is .97. On the other hand, as S_1 provides all data items and uses consistent formatting, the probability of providing a particular item and that of using the format on a data item are both 1. Thus, for each $D \in \mathcal{D}$ we have $P(\Phi_D(S_1)|S_1 \not\rightarrow S_2) = 1 * .97 * 1 = .97$ (Eq.(6)). In a similar way, assuming S_2 has accuracy .61 and there are 100 uniformly distributed false values, we compute $P(\Phi_D(S_2)|S_2 \not\rightarrow S_1) = .61$ if S_2 provides a true value on D, and $P(\Phi_D(S_2)|S_2 \not\rightarrow S_1) = \frac{1-.61}{100} = .0039$ if S_2 provides a false value on D. Thus, $P(\Phi|S_1 \perp S_2) = (.97^5) * (.61^3 * .0039^2) = 3 * 10^{-6}$.

Next consider $P(\Phi|S_1 \rightarrow S_2)$, which requires computing $P(\Phi_D(S_1)|S_1 \rightarrow S_2)$ and $P(\Phi_D(S_2)|S_2 \not\rightarrow S_1)$ for each $D \in \mathcal{D}$ (Eq.(5)). Source S_1 shares three values with S_1 and they are all correct. According to Eq.(7), the probability for such data item D is $P(\Phi_D(S_1)|S_1 \rightarrow S_2) = (1-.8)*.97+.8*(1-.8)*1 = .354$. On the other hand, S_1 provides two different values from S_2 and each of them is true. According to Eq.(8), the probability for such data item D is $P(\Phi_D(S_2)|S_1 \rightarrow S_2) = (1 - .8) * .97 = .194$. Thus, $P(\Phi|S_1 \rightarrow S_2) = (.354^3 * .194^2) * (.61^3 * .0039^2) = 5.8 * 10^{-9}$.

Similarly, we have $P(\Phi|S_2 \rightarrow S_1) = 2.3 * 10^{-7}$. According to Eq.(3), $P(S_1 \perp S_2|\Phi) = \frac{.5*3*10^{-6}}{.5*3*10^{-6}+.25*5.8*10^{-9}+.25*2.3*10^{-7}} = .96$, so no-copying is very likely. □

Table 2: List explanation for no-copying between S_1 and S_2.

	Score	Evidence
Pos	3.2	S_1 provides different values from S_2 on 2 data items
	3.06	Among the data items for which S_1 and S_2 provide the same value, S_1 uses different formats for 3 data items
	.7	The a-priori belief is that S_1 is more likely to be independent of S_2
Neg	.06	S_1 provides the same *true* value for 3 data items as S_2

Note again that the reasoning in the example is how a detailed description of the Bayesian analysis would look like (many details already skipped) for a no-copying decision. Obviously it is very hard to understand for non-technical users and can even be overwhelming for people who understand Bayesian analysis. We next show how we can explain such decisions more elegantly.

3. EXPLAINING THE DECISION

We start with snapshot explanations: given a decision W, we take the data and all decisions made at the convergence round except W as input and explain W. We first describe how we generate the explanation that strictly follows the Bayesian analysis (Sec.3.1-3.2). We then show how we can shorten the explanation by orders of magnitude (Sec.3.3-3.4).

3.1 List explanation

Bayesian analysis considers all possible choices, collects *evidence* and computes the probability for each of them. To explain a decision W, rather than showing only the positive evidence for W, we shall show for each alternative W' that the accumulated evidence for W is stronger than that for W'. We thus propose the following form for a snapshot explanation.

DEFINITION 3.1 (LIST EXPLANATION). *The* list explanation *for a decision W versus an alternative W' is in the form $(\mathbf{L}^+, \mathbf{L}^-)$, where $\mathbf{L}^+$ is the list of positive evidence for W and $\mathbf{L}^-$ is the list of negative evidence for W (but positive for W'). Each evidence $l \in \mathbf{L}^+ \cup \mathbf{L}^-$ is associated with a positive score, denoted by $s(l)$.*

A snapshot explanation for W contains a set of list explanations, one for each alternative choice W'. □

Ideally, a list explanation should be *correct* and *complete*. A list explanation is correct if the sum of the scores of positive evidence is higher than that for negative evidence. A list explanation is complete if all evidence considered in the Bayesian analysis is included. Obviously, a complete list explanation must be correct as it strictly reflects the Bayesian analysis; however, as we show soon, such a list explanation is often huge in size and can be overwhelming to users. In Sec.3.4 we show how we can relax the completeness requirement and shorten a list explanation, such that the result is *correct and comparable to the complete list explanation*.

EXAMPLE 3.2. *Table 2 shows the list explanation for "S_1 does not copy from S_2" versus "S_1 copies from S_2" in Ex.1.1. There are three pieces of positive evidence showing no-copying and one piece of negative evidence showing copying. The list explanation is correct: the total positive score $3.2 + 3.06 + .7 = 6.96$ is higher than the total negative score .06. The list explanation is also complete, showing all evidence considered in the Bayesian analysis.* □

3.2 Generating list explanations

We next describe how we generate a list explanation strictly following the Bayesian analysis. We illustrate the main idea on no-copying between S and S'.

Table 3: List explanation for no-copying between S_1 and S_2 strictly following the Bayesian analysis.

	Score	Evidence
Pos	1.6	S_1 provides a different value from S_2 on Stonebraker
	1.6	S_1 provides a different value from S_2 on Carey
	1.0	S_1 uses a different format from S_2 although shares the same (true) value on Dewitt
	1.0	S_1 uses a different format from S_2 although shares the same (true) value on Bernstein
	1.0	S_1 uses a different format from S_2 although shares the same (true) value on Halevy
	.7	The a-priori belief is that S_1 is more likely to be independent of S_2

No-copying: Recall that between two sources there are three possible relationships: $S \perp S'$, $S \to S'$ and $S' \to S$. To explain $S \perp S'$, we shall show $P(S \perp S'|\Phi) > P(S \to S'|\Phi)$ and $P(S \perp S'|\Phi) > P(S' \to S|\Phi)$; thus, the snapshot explanation includes two list explanations. According to the Bayesian analysis (Eq.(3)), we shall show $\beta P(\Phi|S \perp S') > \alpha P(\Phi|S \to S')$ for the former and similar for the latter. As we assume independence of data items, we need to show the following (derived from Eq.(4-5)).

$$\Pi_{D \in \mathcal{D}} P(\Phi_D(S)|S \not\to S') > \Pi_{D \in \mathcal{D}} P(\Phi_D(S)|S \to S') \cdot \frac{\alpha}{1-2\alpha}. \tag{9}$$

Recall that we compare the *sum* of the scores for positive and negative evidence; we thus rewrite (9) as follows.

$$\sum_{D \in \mathcal{D}} \ln P(\Phi_D(S)|S \not\to S') > \sum_{D \in \mathcal{D}} \ln P(\Phi_D(S)|S \to S') + \ln \frac{\alpha}{1-2\alpha}. \tag{10}$$

Each data item D is involved in the computation of both sides of the inequality. We decide if it supports $S \not\to S'$ or $S \to S'$ by comparing $P(\Phi_D(S)|S \not\to S')$ and $P(\Phi_D(S)|S \to S')$. If the former is higher, D is positive evidence for no-copying with score $\ln \frac{P(\Phi_D(S)|S \not\to S')}{P(\Phi_D(S)|S \to S')}$; if the latter is higher, D is negative evidence with score $\ln \frac{P(\Phi_D(S)|S \to S')}{P(\Phi_D(S)|S \not\to S')}$; otherwise, D is not evidence for either decision. In other words, we rewrite Eq.(10) as

$$\begin{aligned} &\sum_{D \in \mathcal{D}, P(\Phi_D(S)|S \not\to S') > P(\Phi_D(S)|S \to S')} \ln \frac{P(\Phi_D(S)|S \not\to S')}{P(\Phi_D(S)|S \to S')} \\ > &\sum_{D \in \mathcal{D}, P(\Phi_D(S)|S \not\to S') < P(\Phi_D(S)|S \to S')} \ln \frac{P(\Phi_D(S)|S \to S')}{P(\Phi_D(S)|S \not\to S')} \\ &+ \ln \frac{\alpha}{1-2\alpha}. \end{aligned} \tag{11}$$

Finally, in Eq.(11) $\ln \frac{\alpha}{1-2\alpha}$ represents the evidence coming from the a-priori belief (α, β are not involved in any other part of Eq.(11)). This evidence is negative if $\alpha > 1 - 2\alpha$ ($\alpha > \frac{1}{3}$).

Obviously, the explanation is complete and correct: $P(S \perp S'|\Phi) > P(S \to S'|\Phi)$ if and only if the scores of positive evidence sum up to be higher than those of negative evidence.

EXAMPLE 3.3. *Consider explaining $S_1 \perp S_2$ in Ex.1.1. The strict list explanation for $S_1 \not\to S_2$ is shown in Table 3.*

For item Stonebraker, *denoted by D_1, S_1 provides a different value from S_2. Recall from Ex.2.3 that $P(\Phi_{D_1}(S_1)|S_1 \not\to S_2) = .97$ and $P(\Phi_{D_1}(S_1)|S_1 \to S_2) = .194$. Thus, D_1 serves as positive evidence for no-copying with score $\ln \frac{.97}{.194} = 1.6$. We compute the same score for item* Carey.

For item Dewitt, *denoted by D_2, S_1 provides the same value as S_2 but uses a different format. Recall that $P(\Phi_{D_2}(S_1)|S_1 \not\to S_2) = .97$ and $P(\Phi_{D_2}(S_1)|S_1 \to S_2) = .354$. Thus, D_2 also serves as positive evidence and the score is $\ln \frac{.97}{.354} = 1.0$. Note that there are actually two pieces of sub-evidence hidden in this evidence, one about providing the same value and serving as negative evidence, and one about using a different format and serving as positive evidence; we discuss how to separate them in Sec. 3.3. We compute the same score for items* Bernstein *and* Halevy.

Finally, the a-priori belief when $\alpha = .25$ serves as positive evidence with score $|\ln \frac{.25}{1-2*.25}| = .7$.

In total, there are 6 pieces of positive evidence and no negative evidence. Note that by equation transformation and evidence extraction, the explanation is already much simpler than the description of Bayesian analysis in Ex.2.3. □

We now summarize how we explain a Bayesian decision W.

1. List each alternative choice other than W.
2. Generate a list explanation for each choice in four steps.
   (a) Write and expand the inequation that we need to show according to the Bayesian analysis.
   (b) Take the logarithm of each side of the inequation.
   (c) For each involved element (*e.g.*, data item in case of copying detection), compare the probability computed on each side and decide if it serves as positive or negative evidence.
   (d) Handle evidence from the constant term.

We next briefly discuss how we generate explanations for other types of decisions according to this algorithm.

Copying detection: When we explain $S \to S'$, we shall show $P(S \to S'|\Phi) > P(S \perp S'|\Phi)$ and $P(S \to S'|\Phi) > P(S' \to S|\Phi)$ using two list explanations in a similar fashion.

Truth discovery: Consider a data item D and we wish to explain why a particular value v is decided to be correct. Here, each alternative choice is another value $v' \in \mathcal{V}(D), v' \neq v$ (Step 1). For each v', we generate a list explanation showing $P(v \text{ true}|\Phi_D) > P(v' \text{ true}|\Phi_D)$ (Step 2); thus, the explanation consists of $|\mathcal{V}(D)| - 1$ list explanations (in fact, we can generate a single list explanation for all values that are not provided by any source). After expanding the inequation according to Eq.(1-2) and taking the logarithm (Step 2a-2b), we have

$$\sum_{S \in \mathcal{S}} I(\Phi_D(S)) \ln P(\Phi_D(S)|v \text{ true}) > \sum_{S \in \mathcal{S}} I(\Phi_D(S)) \ln P(\Phi_D(S)|v' \text{ true}). \tag{12}$$

For each element $S \in \mathcal{S}$, as $I(\Phi_D(S))$ appears on both sides, we only need to compare $\ln P(\Phi_D(S)|v \text{ true})$ and $\ln P(\Phi_D(S)|v' \text{ true})$ (Step 2c). If S is a provider of v, the former is larger and S serves as positive evidence; if S is a provider of v', the latter is larger and S serves as negative evidence; if S provides another value, they are the same and S is neither positive nor negative evidence. Finally, if S is a copier, the score would be much lower than otherwise.

3.3 Evidence categorization and aggregation

The current explanation generation algorithm lists each data item as a piece of evidence. Since there can be a lot of data items in general, the explanation can be long and overwhelming. We observe from Table 3 that a lot of evidence can actually look the same except that they are on different data items; a natural thought is to categorize such evidence and present the aggregated evidence. The question is then how to categorize the evidence wisely without ending up with too many categories. We do so in two steps.

Evidence separation: Since our observation on a data item D consists of three aspects: existence of the item, provided value(s), and used format(s) (see Eq.(6)), we first divide a piece of evidence into three, one for each aspect. This enables categorization on each aspect instead of on combinations of different aspects.

Table 4: List explanation for no-copying between S_1 and S_2 after evidence separation.

	Score	Evidence
Pos	1.6	S_1 provides a different value from S_2 on Stonebraker
	1.6	S_1 provides a different value from S_2 on Carey
	1.02	S_1 uses a different format from S_2 on Dewitt
	1.02	S_1 uses a different format from S_2 on Bernstein
	1.02	S_1 uses a different format from S_2 on Halevy
	.7	The a-priori belief is that S_1 is more likely to be independent of S_2
Neg	.02	S_1 and S_2 share the same *true* value on Dewitt
	.02	S_1 and S_2 share the same *true* value on Bernstein
	.02	S_1 and S_2 share the same *true* value on Halevy

Evidence separation requires the score on a data item D to be split for each different aspect, denoted by $score_{ext}(D)$, $score_{val}(D)$, and $score_{fmt}(D)$ for our case. A careless approach would lead to computing these scores as

$$score_{ext}(D) = \ln \frac{P(\Phi_D(S) \neq \emptyset | S \not\rightarrow S')}{P(\Phi_D(S) \neq \emptyset | S \rightarrow S')}; \tag{13}$$

$$score_{val}(D) = \ln \frac{P(val(\Phi_D(S)) = v | S \not\rightarrow S')}{P(val(\Phi_D(S)) = v | S \rightarrow S')}; \tag{14}$$

$$score_{fmt}(D) = \ln \frac{P(fmt(\Phi_D(S)) = f | S \not\rightarrow S')}{P(fmt(\Phi_D(S)) = f | S \rightarrow S')}. \tag{15}$$

However, in this way the three scores do not sum up to the overall score for D, because $P(\Phi_D(S)|S \rightarrow S') \neq P(\Phi_D(S) \neq \emptyset | S \rightarrow S') \cdot P(val(\Phi_D(S)) = v | S \rightarrow S') \cdot P(fmt(\Phi_D(S)) = f | S \rightarrow S')$ (the equation holds though for $P(\Phi_D(S)|S \not\rightarrow S')$).

Instead, we compute (1) $sc_1 = score_{ext}(D)$, (2) $sc_2 = score_{ext}(D)$ + $score_{val}(D)$, and (3) $sc_3 = score_{ext}(D) + score_{val}(D) + score_{fmt}(D)$, from which we can then infer $score_{ext}(D)$, $score_{val}(D)$, and $score_{fmt}(D)$.

Consider the case of both sources providing the same value v as an example. First, sc_1 can be computed by Eq.(13), where

$$\begin{aligned} & P(\Phi_D(S) \neq \emptyset | S \rightarrow S') \\ = & (1-s)P(\Phi_D(S) \neq \emptyset | S \rightarrow S', D \text{ not copied}) + s. \end{aligned} \tag{16}$$

Second, sc_2 can be computed as follows.

$$\begin{aligned} & sc_2 = score_{ext}(D) + score_{val}(D) \\ = & \ln \frac{P(\Phi_D(S) \neq \emptyset, val(\Phi_D(S)) = v | S \not\rightarrow S')}{P(\Phi_D(S) \neq \emptyset, val(\Phi_D(S)) = v | S \rightarrow S')}; \end{aligned} \tag{17}$$

$$\begin{aligned} & P(\Phi_D(S) \neq \emptyset, val(\Phi_D(S)) = v | S \rightarrow S') = s + (1-s) \\ \cdot & P(\Phi_D(S) \neq \emptyset, val(\Phi_D(S)) = v | S \rightarrow S', D \text{ not copied}). \end{aligned} \tag{18}$$

Third, sc_3, which sums up all scores on D, is obviously the final score on D; that is, $\ln \frac{P(\Phi_D(S)|S \not\rightarrow S')}{P(\Phi_D(S)|S \rightarrow S')}$.

A positive score shows that the specific aspect serves as positive evidence for no-copying and vice versa. Note that even if D as a whole serves as positive evidence, it is not necessary that $score_{ext}(D)$, $score_{val}(D)$, and $score_{fmt}(D)$ are all positive. As shown in Ex.3.3, item Dewitt (D_2) serves as positive evidence. However, we compute $score_{ext}(D_2) = \ln \frac{1}{.2*1+.8} = 0$, $score_{val}(D_2) = \ln \frac{.97}{.2*.97+.8} - 0 = -.02$, $score_{fmt}(D_2) = 1 - (-.02) - 0 = 1.02$. Thus, providing D_2 is neither positive nor negative evidence, sharing the same value is *negative* evidence for no-copying, and using different formats is *positive* evidence. Being able to expose such hidden evidence is an extra benefit of evidence separation. The evidence list after separation is shown in Table 4.

Categorization: Now for each aspect we can categorize according to *why* the aspect of a data item serves as positive or negative evidence given the feature of the data. Take the value aspect as an example. There are four categories: (1) *different values*; (2) *sharing false value*; (3) *sharing true value*; (4) *providing a value that is different but more likely to be provided if the source is a copier but provides this item independently* (*e.g.*, if S copies high-accuracy data from S' but all of its independently provided data are wrong, then for a value different from S''s, S has a probability of 1 to provide a wrong value conditioned on it being a copier of S' but a lower probability otherwise). Among these categories, the first forms positive evidence for no-copying and the others form negative evidence. Note that *sharing false value*, which is typically strong evidence, and *sharing true value*, which is typically weak evidence, are essentially the same, but by separating them we can emphasize significant evidence. We have similar categories for the existence aspect and the format aspect.

Algorithm 1: GenerateList($S, S', \mathcal{D}$)

```
Input  : $S, S'$ sources, $\mathcal{D}$ data items
Output : $(\mathbf{L}_1^+, \mathbf{L}_1^-)$ and $(\mathbf{L'}_2^+, \mathbf{L'}_2^-)$ as explanation for $S \perp S'$
$Score[1:2][1:asp][1:cat] = 0$;
$Count[1:2][1:asp][1:cat] = 0$; // $asp$ is the number of aspects
  and $cat$ is the number of categories for each aspect
// Collect, categorize, and aggregate evidence
foreach $D \in \mathcal{D}$ do
    COLLECTEVIDENCE($S, S', D, Score, Count$);
// Generate explanation
$\mathbf{L}_1^+ = \mathbf{L}_1^- = \mathbf{L'}_2^+ = \mathbf{L'}_2^- = \emptyset$;
foreach $d = 1:2$ do
    foreach $i = 1:asp$ do
        foreach $j = 1:cat$ do
            if $Score[d][i][j] > 0$ then
                Add GENERATEEXPLANATION($S, S', d, i, j,$
                $Score[d][i][j], Count[d][i][j]$) to $\mathbf{L}_d^+$;
            else if $Score[d][i][j] < 0$ then
                Add GENERATEEXPLANATION($S, S', d, i, j,$
                $-Score[d][i][j], Count[d][i][j]$) to $\mathbf{L}_d^-$;
SORT($\mathbf{L}_1^+$); SORT($\mathbf{L}_1^-$); SORT($\mathbf{L'}_2^+$); SORT($\mathbf{L'}_2^-$);
return $(\mathbf{L}_1^+, \mathbf{L}_1^-)$, $(\mathbf{L'}_2^+, \mathbf{L'}_2^-)$;
```

Algorithm 2: CollectEvidence($S, S', \mathcal{D}, Score, Count$)

```
foreach $i = 1:asp$ do
    // Evidence collection
    $s_1$ = COMPUTESCORE($D, S \rightarrow S', i$);
    $s_2$ = COMPUTESCORE($D, S' \rightarrow S, i$);
    // Evidence categorization
    $cat_1 \leftarrow$ CATEGORIZEREASON($D, S \rightarrow S', i$);
    $cat_2 \leftarrow$ CATEGORIZEREASON($D, S' \rightarrow S, i$);
    // Evidence aggregation
    if $s_1 \neq 0$ then
        $Score[1][i][cat1] \leftarrow Score[1][i][cat1] + s_1$;
        $Count[1][i][cat1]++$;
    if $s_2 \neq 0$ then
        $Score[2][i][cat2] \leftarrow Score[2][i][cat2] + s_2$;
        $Count[2][i][cat2]++$;
```

Aggregation: We show the framework of our algorithm with categorization and aggregation in Algorithm 1. The algorithm uses two arrays, $Score$ and $Count$, to store the sum of the scores and the count of the involved instances. The first dimension of the array corresponds to each *direction* ($S \not\rightarrow S'$ or $S' \not\rightarrow S$); the second dimension corresponds to each *aspect* in probability computation;

Table 5: Score and count for each category in Ex.3.4.

Aspect	Category1	Category2	Category3	Category4
Exist	0	0	0	0
Value	3.2, 2	0	-.06, 3	0
Format	3.06, 3	0	0	0

and the third dimension corresponds to each *category* of reasons. The algorithm proceeds in two phases.

1. The *evidence collection* phase (Lines 3-4) finds evidence from each data item. It invokes function COLLECTEVIDENCE (Algorithm 2), which for each aspect first computes the score (Lines 2-3), then decides the category of the reasons according to the data (being different, or the same true value, or the same false value, etc.) (Lines 4-5), and finally adds the scores and counts the instances for the corresponding category (Lines 6-9).
2. The *explanation generation* phase (Lines 5-14) decides for each category if the evidence is positive or negative, generates the verbal explanation, and adds it to the corresponding explanation list. It finally sorts the lists in decreasing order of the scores and returns the result list explanation.

EXAMPLE 3.4. *Continue with Ex.3.3. The result of Algorithm 2 is shown in Table 5 (corresponding to Table 4). Take D_2 as an example. Line 2 computes $score_{ext}(D_2) = 0, score_{val}(D_2) = -.02$ and $score_{fmt}(D_2) = 1.02$, as we have shown. For the value aspect, S_1 and S_2 provide the same true value so Line 4 puts it in Category 3; for the format aspect, S_1 and S_2 use different formats so falls in Category 1. Lines 6-7 aggregate the evidence, finds 2 pieces of positive evidence (not including the a-priori belief evidence) and 1 piece of negative evidence. Then, Algorithm 1 generates explanation for each cell (Line 10) and sorts them, resulting in the list explanation of Table 2.*

Recall that without categorization and aggregation, we would show 5 pieces of positive evidence (Table 3), each for a data item, and hide the fact that sharing 3 values actually serves as negative evidence for no-copying.

With evidence categorization and aggregation, the amount of evidence is not determined by the number of data items, but by the number of aspects and categories. In our experiments, the evidence lists can be shortened by orders of magnitude. We next show complexity of this algorithm.

PROPOSITION 3.5. *Let $|\mathcal{D}|$ be the number of provided data items, asp be the number of aspects, and cat be the maximum number of categories in each aspect.* GENERATELIST *generates a correct and complete explanation in time $O(|\mathcal{D}| + asp \cdot cat \cdot \log(asp \cdot cat))$.* □

PROOF. Evidence categorization requires one scanning of the data items and takes time $O(|\mathcal{D}|)$. Explanation generation takes time $O(asp \cdot cat)$. Sorting up to $asp \cdot cat$ pieces of evidence takes time $O(asp \cdot cat \cdot \log(asp \cdot cat))$. □

To summarize, we decide categories manually by enumerating various aspects that are considered in probability computation, such as Eq.(6), and various reasons why a particular aspect can serve as positive or negative evidence given the feature of the data. We then categorize and aggregate evidence accordingly. The same principle applies to other decisions as well. For example, for truth discovery decisions, there are two aspects: accuracy of the source and copying of the source from others; for the former aspect, there are two reasons for positive evidence: *provided by an accurate source* and *provided by a less accurate source*.

3.4 List shortening

Although evidence aggregation can significantly reduce the amount of evidence, the result lists can still contain twenty or thirty pieces of evidence, depending on the number of categories. We can further remove "unimportant" evidence, illustrated as follows.

EXAMPLE 3.6. *Consider the following list explanation (we show only scores).*

$$\mathbf{L}^+ = \{1000, 500, 60, 2, 1\};$$
$$\mathbf{L}^- = \{950, 50, 5\}.$$

Obviously, removing the evidence whose scores are below 100 still shows that the positive evidence is much stronger than the negative evidence. However, if we further remove the negative evidence with score 950, it gives the wrong impression that there is no negative evidence. On the other hand, if we further remove the positive evidence with score 500, it gives the wrong impression that the positive evidence is only slightly stronger. □

Note that once we remove some evidence, the list explanation is not complete any more. Instead, we wish the explanation to be *correct and comparable to the complete explanation*. Before we formally define what we mean by "comparable", we first describe a few simple ways of list shortening. Such strategies include showing only top-k evidence, and showing only evidence whose score is above a given threshold θ. However, using the same k and θ everywhere may cause over-shortening for some instances, where the explanation is incorrect, and under-shortening for some other instances, where further shortening will generate a shorter but still correct and comparable explanation. We next propose two better solutions. Both of them remove evidence from the end of the list, as typically the lower is the score, the less important is the evidence; on the other hand, each follows a different principle of what is considered as *comparable* to the complete list explanation.

3.4.1 Tail cutting

First, given a shortened list explanation, we can guess the bound of the accumulated scores: the minimal accumulated score for all positive evidence happens when each removed positive evidence has score 0; the maximal score for negative evidence happens when each removed negative evidence has a score as high as the lowest remaining score for negative evidence. If even in such a worst case, the remaining positive evidence is still stronger, we consider the explanation as *comparable*. In Ex.3.6, if we remove the last negative evidence and inform the user that "*there is 1 more piece of negative evidence with lower score*", then the removed score is at most 50, so the negative evidence (total score $950 + 50 + 50 = 1050$) is still weaker. We can further remove the last 3 pieces of positive evidence and the positive evidence is still stronger (total score $1000 + 500 = 1500 > 1050$). According to this intuition, we shall solve the following optimization problem.

DEFINITION 3.7 (TAIL-CUTTING PROBLEM). *Consider the following list explanation (we show only scores):*

$$\mathbf{L}^+ = \{x_1, x_2, \dots, x_n\};$$
$$\mathbf{L}^- = \{y_1, y_2, \dots, y_m\}.$$

The tail-cutting problem *minimizes $s + t, 1 \leq s \leq n, 1 \leq t \leq m$, under the constraint*

$$\sum_{i=1}^{s} x_i > \sum_{j=1}^{t} y_j + y_t(m-t). \quad \square \qquad (19)$$

In this definition, constraint (19) compares the minimum positive score, obtained when all removed scores are 0, with the maximum negative score, obtained when the $m-t$ pieces of removed evidence

Algorithm 3: CUTTAIL($\mathbf{L}^+, \mathbf{L}^-$)

```
   Input  : List explanation $(\mathbf{L}^+, \mathbf{L}^-)$
   Output : $s, t$
1  $X \leftarrow \sum_{i=1}^{n} x_i; Y \leftarrow \sum_{j=1}^{m} y_j; s_0 \leftarrow n; t_0 \leftarrow m;$
   // Find the minimum $s$
2  while $\sum_{i=1}^{s_0} x_i > Y$ do
3      $s_0 - -;$
4  $min \leftarrow s_0 + m; s \leftarrow s_0; t \leftarrow m;$
   // Try each $t$ and find the minimum $s + t$
5  while $X > Y$ do
6      $t_0 - -;$
7      $Y \leftarrow Y + (y_{t_0} - y_{t_0+1})(m - t_0);$
8      while $\sum_{i=1}^{s_0} x_i < Y$ and $s_0 < n$ do
9          $s_0 + +;$
10     if $s_0 + t_0 < min$ then
11         $min \leftarrow s_0 + t_0; s \leftarrow s_0; t \leftarrow t_0;$
12 return $s, t$;
```

Algorithm 4: KEEPDIFF($\mathbf{L}^+, \mathbf{L}^-$)

```
   Input  : List explanation $(\mathbf{L}^+, \mathbf{L}^-)$
   Output : $s, t$
1  $X_0 \leftarrow \sum_{i=1}^{n} x_i; Y_0 \leftarrow \sum_{j=1}^{m} Y_j; X \leftarrow 0; Y \leftarrow 0;$
2  $min \leftarrow \frac{\beta_1}{\beta_2}; s \leftarrow n; t \leftarrow m; s_0 \leftarrow n; t_0 \leftarrow m;$
3  $l \leftarrow x_n \leq y_m ? 0 : 1;$// $l$ records the list to consider next
4  while $s_0 > 1, t_0 > 1$ do
5      if $l = 0$ then
6          $X \leftarrow X - x_{s_0}; s_0 - -;$
7          if $X \leq Y + y_{t_0}$ and $X + x_{s_0} > Y + y_{t_0}$ then
8              $l = 1;$
9      else
10         $Y \leftarrow Y - y_{t_0}; t_0 - -;$
11         if $Y \leq X + x_{s_0}$ and $Y + y_{t_0} > X + x_{s_0}$ then
12             $l = 0;$
13     if $X_0 - X > Y_0 - Y$ then
14         $score = \frac{|X-Y|+\beta_1}{\max(X,Y)^\gamma+\beta_2};$
15         if $score < min$ then
16             $min = score; s = s_0; t = t_0;$
17 return $s, t$;
```

all have the maximum possible score y_t, and so guarantees that the shortened list is comparable to the complete explanation.

Algorithm CUTTAIL (Algorithm 3) proceeds in three steps. First, Lines 2-4 iteratively try $s = n, n-1, \ldots$, as far as $\sum_{i=1}^{s} x_s > \sum_{j=1}^{m} y_j$; the resulting s is the minimum when we do not remove any negative evidence ($t = m$) and serves as the starting point for the next step. Second, Lines 5-7 iteratively try $t = m, m-1, \ldots$, as far as $\sum_{i=1}^{n} x_i > \sum_{j=1}^{t} y_j + y_t(m-t)$ (at this point removing any more negative evidence cannot satisfy constraint (19), even if we add back all positive evidence). For each t, Lines 8-9 increase s when needed to guarantee constraint (19), and Lines 10-11 record $s + t$. Finally, Line 12 returns the s and t with minimal $s + t$. Obviously, CUTTAIL finds the optimal solution.

PROPOSITION 3.8. *Algorithm* CUTTAIL *solves the* TAIL-CUTTING *problem in time $O(m+n)$.* □

PROOF. Finding the minimum s takes time $O(n)$. Finding minimum $s + t$ decreases t from m to 1 in the extreme case, and increases s from 1 to n in the extreme case, so takes time $O(m + n)$. □

EXAMPLE 3.9. *Consider applying* CUTTAIL *to the full list explanation in Ex.3.6. The algorithm first removes the last 3 pieces of evidence from $\mathbf{L}^+$; at this point, $1000 + 500 > 950 + 50 + 5$ and $s + t = 2 + 3 = 5$. It then tries $t = 2$, where increasing s is not needed ($1000 + 500 > 950 + 50 + 50 * 1$); so $s + t = 2 + 2 = 4$. When it tries $t = 1$, even if all positive evidence is added back, we still have $1000 + 500 + 60 + 2 + 1 < 950 + 950 * 2$, so it stops. Finally, it returns $s = 2, t = 2$ as the result.* □

3.4.2 Difference keeping

In our second method, we consider the shortened list as comparable to the complete one if it keeps the difference between the accumulated scores for the two evidence lists; this corresponds to dividing both the numerator and the denominator of Eq.(3) by a constant. In other words, we wish that the sum of the scores for removed positive evidence is nearly the same as that for removed negative evidence. Meanwhile, we wish to make the lists as short as possible, equivalent to making the sum of the removed scores as large as possible. Thus, we solve the following problem.

DEFINITION 3.10 (DIFFERENCE-KEEPING PROBLEM). *Consider the same list explanation as in Defn.3.7. Let $X = \sum_{i=s+1}^{n} x_i$ and $Y = \sum_{j=t+1}^{m} y_j$ ($1 \leq s \leq n, 1 \leq t \leq m$). The* difference-keeping problem *minimizes*

$$\frac{|X - Y| + \beta_1}{\max(X, Y) + \beta_2}, \tag{20}$$

where $0 < \beta_1, \beta_2 \leq \min\{\min_{i \in [1,n]}\{x_i\}, \min_{j \in [1,m]}\{y_j\}\}$ are small positive numbers, under the constraint

$$\sum_{i=1}^{s} x_i > \sum_{j=1}^{t} y_j. \quad \square \tag{21}$$

In the objective function (20), we use β_1 to guarantee a non-zero numerator such that we do not necessarily choose a solution where $X = Y$ but s and t are large. We use β_2 to guarantee a non-zero denominator such that we do not necessarily shorten the list when there does not exist any s and t to make X and Y close enough. Instead of $\max(X, Y)$, we can use $\max(X, Y)^\gamma$ to control how much we wish to emphasize *small* list length; in our experiments, we did not observe a noticeable difference when we range γ from 1 to 2. Constraint (21) guarantees correctness of the explanation. Note that one may add other constraints such as giving upper bounds of X and Y to guarantee that at most a given fraction of evidence is removed, or use $\max(X, Y)^\gamma, \gamma > 0$, to control how much we wish to emphasize *small* list length.

A naive way of solving this problem tries each combination of m and n and can take time $O(mn(m + n))$. We now sketch an algorithm that solves the problem in only linear time. Recall that we remove evidence from the end of the lists, so we call a removed subset of evidence a *suffix sublist*. The key idea is that for each suffix sublist L, there is a *key evidence* $k(L)$ in the other list, such that the longest suffix without $k(L)$ has lower or the same accumulated score as L and the shortest suffix with $k(L)$ has higher accumulated score than L. Then, for each suffix, we only need to examine these two suffix sublists from the other list. Consider Ex.3.6 and the suffix list $\{5\}$ from $\mathbf{L}^-$. Its key evidence in $\mathbf{L}^+$ is the evidence with score 60. The longest suffix without this evidence, $\{2, 1\}$, has a lower score than 5, and the shortest suffix with the evidence, $\{60, 2, 1\}$, has a higher score. Obviously, any other suffix sublist in $\mathbf{L}^+$ has a higher difference from $\{5\}$ than these two.

According to this intuition, Algorithm KEEPDIFF (Algorithm 4) scans $\mathbf{L}^+$ and $\mathbf{L}^-$ bottom-up (Line 2 sets the cursors s_0 and t_0 to

Table 6: Applying KEEPDIFF in Ex.3.6.

Rnd	Remove from $\mathbf{L}^+$	Remove from $\mathbf{L}^-$	Difference	Objective
0	$\emptyset$	$\emptyset$	0	$\frac{1}{1} = 1$
1	$\{1\}$	$\emptyset$	1	$\frac{2}{2} = 1$
2	$\{1, 2\}$	$\emptyset$	3	$\frac{4}{4} = 1$
3	$\{1, 2\}$	$\{5\}$	2	$\frac{3}{6} = .5$
4	$\{1, 2\}$	$\{5, 50\}$	52	$\frac{53}{56} = .95$
5	$\{1, 2, 60\}$	$\{5, 50\}$	8	$\frac{9}{64} = .14$

n and m respectively). At the beginning, the algorithm starts with removing evidence with the lowest score (Line 3). Then, in each round, it decides the evidence to remove at the next round as follows: it picks evidence from the same list until reaching the key evidence of the next suffix (the current suffix plus the next evidence) of the other list (Lines 5-12). In addition, each round checks constraint (21), computes the objective function, and records the solution if its value is lower than the recorded lowest value (Lines 13-16). This process continues until we reach the first evidence of a list. Algorithm KEEPDIFF is optimal, shown as follows.

PROPOSITION 3.11. *Algorithm* KEEPDIFF *solves the* DIFFERENCE-KEEPING *problem in time $O(m + n)$.* □

PROOF. For each s_0 (similar for t_0), the algorithm finds the t_0 with the minimum difference of accumulated removed scores. If we further increase t_0, $|X - Y|$ increases and $\max(X, Y)$ decreases, so the new value of the objective function will increase. If we further decrease t_0 and assume Y increases by Δ, then $|X - Y|$ increases by Δ and $\max(X, Y)$ increases by Δ as well, so the new value of the objective function will again increase. This proves the optimality of the algorithm.

The algorithm at best reduces s_0 from n to 1 and t_0 from m to 1 interleavingly, so takes time $O(m + n)$. □

EXAMPLE 3.12. *Continue with Ex.3.6. Table 6 shows removed evidence and the value of the objective function in each round; here, we set $\beta_1 = \beta_2 = 1$. We start with $\mathbf{L}^+$, as it contains the lowest score. Initially, the next suffix sublist in $\mathbf{L}^-$ is $\{5\}$, and its key evidence in $\mathbf{L}^+$ has score 60; thus, we pick scores 1 and 2 first and then switch to list $\mathbf{L}^-$. We continue till reaching the first element of $\mathbf{L}^-$. The result of Round 5 is optimal, even though its difference is not the smallest.* □

Extensions: We next discuss several extensions of the DIFFERENCE-KEEPING problem. First, in some cases we may wish to emphasize small list length and be tolerant with a slightly larger difference. We can change the objective function to

$$\frac{|X - Y| + \beta_1}{\max(X, Y)^\gamma + \beta_2}, \tag{22}$$

where we use $\gamma > 0$ to balance score difference and list length: a large γ emphasizes small list length and a small γ emphasizes small difference. Note that Algorithm KEEPDIFF may not obtain the optimal results when $\gamma \neq 1$ and we will need to try different combinations of *s* and *t* for finding the optimal solution.

Second, if we consider the shortened list is comparable to the complete list when it keeps the ratio of the accumulated scores of the two lists, we can either adjust the objective function, or set a ratio range as a constraint. We skip the details.

In practice, we apply both CUTTAIL and KEEPDIFF and choose the solution with shorter lists (*i.e.*, minimal $s + t$). Our experiments show that these strategies can further shorten the lists by 51%.

EXAMPLE 3.13. *Continue with explaining no-copying between S_1 and S_2 for the running example. For evidence in Table 2,* CUTTAIL *would remove the last two pieces of positive evidence, while* KEEPDIFF *would not remove any evidence. We thus choose the results of* CUTTAIL. *The final explanation can go like this.* "There are 3 pieces of positive evidence for no-copying, where the strongest is that S_1 provides 2 different values from S_2 (with score 3.2). There is 1 piece of negative evidence for no-copying: S_1 provides the same *true* value on 3 data items as S_2 (with score .06). The positive evidence is stronger so no-copying is likely." □

4. EXPLAINING THE EXPLANATION

We next consider generating comprehensive explanations, where we take only provided data as evidence. Again, we start with showing how we generate the full explanation according to the iterative Bayesian analysis (Sec.4.1), and then describe how we shorten the explanation efficiently (Sec.4.2)

4.1 DAG explanation

A comprehensive explanation needs to in addition explain every "evidence" inferred over the data. A natural presentation for such an explanation is the DAG structure, where each node explains a decision, and the children are the evidence.

DEFINITION 4.1 (DAG EXPLANATION). *The* DAG explanation *for a decision W is a DAG in the form of $(\mathbf{N}, \mathbf{E}, R)$, where (1) each node in $\mathbf{N}$ represents a decision and its list explanations, (2) each edge in $\mathbf{E}$ indicates that the decision of the child node is evidence for that of the parent node, and (3) there is a single node R that has no parent and represents the decision W.* □

Similar to snapshot explanations, an ideal DAG explanation should also be *correct* and *complete*. It is correct if (1) the explanation represented by each node is correct, and (2) each child supports its parents as positive evidence. It is complete if for every node, each of its positive evidence that is inferred from the data corresponds to a child node. Note that although an explanation on a node contains both positive evidence and negative evidence, we do not expand the DAG for negative evidence, since the opposite of negative evidence will only further strengthen our decision.

Consider explaining "UCI *is more likely than* BEA *to be the correct affiliation of* Carey" in the motivating example. As we have shown in Ex.1.2, careless generation of the DAG can cause loops. Similar to explaining "WHY" according to provenance [4], we explain by tracing the decisions from the last round of iterations back to the first round. In particular, we start with generating the root node for the decision at the convergence round and its children for the supporting evidence at the same or the previous round. We repeat until all leaf nodes can be inferred directly from the data. We call the result a *full explanation DAG*. Obviously, a full explanation DAG is both correct and complete.

EXAMPLE 4.2. *Fig.1 shows the full explanation DAG for our example. The root node has two children, showing that we make this decision at the convergence round, the 11th round, because we detect copying between $S_3 - S_5$ at the 11th round, and compute a higher accuracy for S_1 than S_3 at the 10th round. We make both of these two decisions based on our decisions at the 10th round that* UWisc, UW *and* BEA *are wrong. Among them, the decision on* BEA *at Round 10 is made for the same two reasons as at Round 11; the decisions on* UWisc *and* UW, *on the other hand, are made because of copying between $S_3 - S_5$ and no-copying between $S_1 - S_2$ (the reasoning is that these two values are provided by three sources with copying and the correct values are provided by two independent sources), both decided at the 10th round. While copying between $S_3 - S_5$ is detected for the same reasons as at*

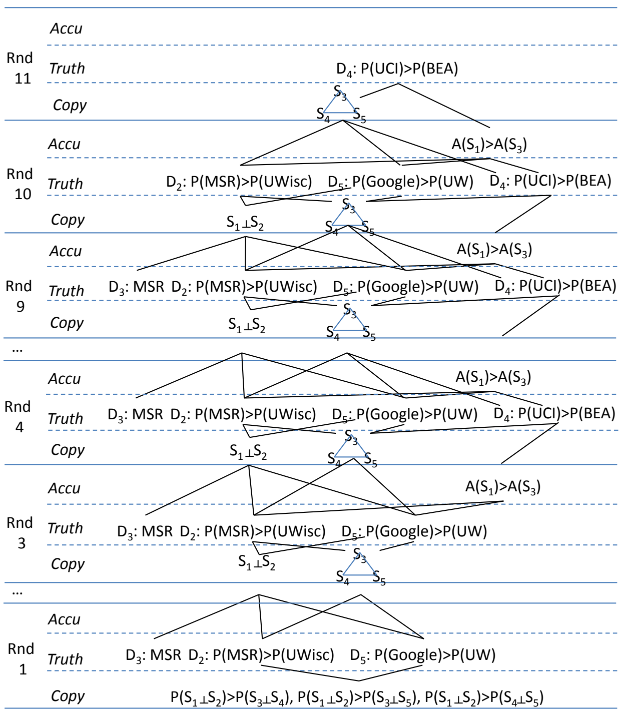


Figure 1: Full explanation DAG for the decision “UCI *is more likely than* BEA *to be the affiliation of* Carey” (represented by $D_4 : P(UCI) > P(BEA)$). The triangle between S_3, S_4 and S_5 represents copying between them; $A(S_1) > A(S_3)$ represents that S_1 has a higher accuracy than S_3; $D_3 : MSR$ represents that MSR is the correct affiliation for Bernstein.

Round 11, no-copying between $S_1 - S_2$ is decided based on the decisions at the 9th round that the shared values MSR, MSR, *and* Google *(for D_2, D_3, D_5 respectively) are all correct. We decide that* MSR *is correct for D_3 purely from the data, because no other value is provided on D_3, so the node is a leaf node. We further expand the DAG for other decisions.*

When we trace back to the 4th round, we show that we decide copying between $S_3 - S_5$ only because we decided at Round 3 that UWisc *and* UW *are wrong, which again are decided because of copying between $S_3 - S_5$ and no-copying between $S_1 - S_2$. When we trace back to the 1st round, we show that we made decisions on* UWisc *and* UW *because the no-copying probability between $S_1 - S_2$ is higher than that between $S_3 - S_5$, which in turn is inferred from the raw data because S_1 and S_2 share fewer values (initially we assume the same probability for each value to be true). We can thus terminate.* □

4.2 Shortening DAG explanations

A full explanation DAG is often huge because some parts can be repeated many times; for example, in Fig.1 the subgraphs for Round 4 to 9 are exactly the same. We wish to reduce the size of the DAG by removing the repeated subgraphs. We observe that if the same decision is made at two consecutive rounds, their supporting positive evidence are typically the same. The only difference is the exact scores, which may change slightly between rounds, but such small changes are not significant in understanding the decision. We thus shorten the explanation by explaining a decision only at its *critical round*, the last round when we change our decision; in other words, we explain how we initially make this decision. Such a DAG is called a *critical-round DAG*.

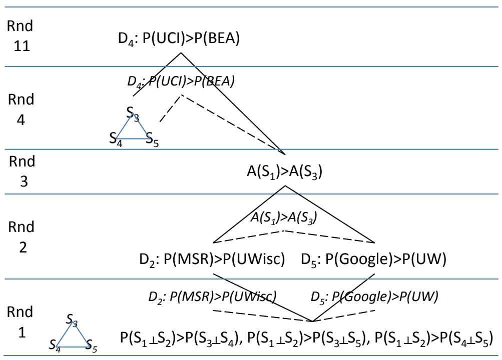


Figure 2: Critical-round DAG. Not being part of the DAG, italic-font nodes indicate the critical round for a decision and dashed lines show the reasons at the critical round.

EXAMPLE 4.3. *Continue with Ex.4.2. Fig.2 shows the critical-round DAG for our example. It shows that the decision is first made at Round 4 (before that we wrongly decide that* BEA *is correct) based on (1) copying between $S_3 - S_5$, decided at Round 4, and (2) that the accuracy of S_1 is higher than S_3, decided at Round 4. The decision of copying between $S_3 - S_5$ is originally made at Round 1 according to the high overlap between these sources; it does not have any child because it is purely inferred from the data. The decision that S_1 has a higher accuracy is originally made at Round 2 (although in that round the difference is not significant for believing that* UCI *is correct), based on the decisions at Round 2 that* UWisc *and* UW*, are false. These two decisions are originally made at Round 1, again based on the decisions at Round 1 that the no-copying probability between $S_1 - S_2$ is higher than that between $S_3 - S_5$, inferred from the raw data. Critical-round DAGs can be significantly smaller; the example DAG includes only 6 nodes.* □

Critical-round DAGs are both correct and complete if we consider only the critical rounds, but one may argue that a decision may have different evidence lists at the critical round and the round that we explore. We can thus enrich the DAG by listing *appearing* and *disappearing* reasons at the exploration round compared with the critical round, and further expand the DAG for these reasons. We call such a DAG an *enriched critical-round DAG* and it is both correct and complete. Our experiments show that there are typically very few appearing reasons and even fewer disappearing reasons, but expanding them can increase the size of the DAG a lot.

We next formally define the critical-round DAG, which we propose to use as the comprehensive explanation.

DEFINITION 4.4 (CRITICAL-ROUND EXPLANATION DAG). *Let W be a decision at Round n. The* critical round *of W, denoted by $r(W, n)$, satisfies the following conditions: (1) W is made in Round $r(W, n) \leq n$, (2) $r(W, n) = 1$, or $\neg W$ is made in Round $r(W, n) - 1$.*

An explanation DAG is called a critical-round DAG *if for each node N and its represented decision W at Round n, N's children represent positive evidence for W at the critical round $r(W, n)$.* □

4.3 DAG construction

We next describe how we construct a critical-round DAG. Obviously, this would require recording the decisions we make in each round. Online construction of explanations requires high efficiency, but this can be challenging for two reasons. First, there can be many rounds before convergence and constructing a DAG would very often require importing the decisions and restoring the status for each round. Second, for each node in the DAG we need to decide the critical round; inefficient algorithms can require importing decisions at different rounds back and forth. We solve the problem by pre-generating the explanation and the evidence list for each decision offline, and storing them in a database.

DB creation: For each round r we create two tables in the database: $\mathsf{Explanation}_r$(decision, explanation) for the explanation, and $\mathsf{Evidence}_r$(decision, evidence) for the evidence list. For each table, we index on decision. Consider a decision W made in the r-th round. If the decision changes from the previous round, there is an entry in $\mathsf{Explanation}_r$ for its explanation and an entry in $\mathsf{Evidence}_r$ for each of its evidence.

For the purpose of supporting diagnosis queries, we also generate identity tuples (W, W) in $\mathsf{Evidence}_r$ if Round r is not any critical round of decision W. Such identity tuples guarantee that all decisions in Round r appear in at least one of the tables for Round r, so making it easy to find decisions at a particular round; they also allow easy querying on all evidence for a decision or all decisions affected by a piece of evidence by joining the tables.

Suppose truth discovery converges in l rounds. Recall that in each round, we perform copying detection, truth discovery, and source accuracy computation. Accordingly we divide decisions in each round into 3 layers (illustrated by Fig.1). We create the database in $3l$ iterations. Starting from the bottom, each time we consider a window of 3 consecutive layers and restore the status. We then generate the explanation for each decision W at the top layer, Layer k, as follows.

1. Decide if the decision has been changed since the previous window (Layer $k - 3$) or if the decision is initial ($k \leq 3$).
2. If W is initial or changed, generate the explanation and the evidence list. Insert the results into the two tables for the corresponding round.

DAG construction: With the database, online explanation generation can be very efficient.

1. We construct the DAG from the root, representing W at the convergence round r.
2. For each node, we (1) query the Explanation from the r-th table to previous tables until finding its explanation (which must be from the critical round for this r-th round decision) and (2) query the corresponding Evidence table for the supporting evidence.
3. For each supporting evidence, we repeat Steps 2-3 to generate a child node and construct the subgraph.

PROPOSITION 4.5. *Let l be the number of rounds and d be the total number of decisions. The database can be constructed in time $O(dl)$. Let n be the size of the result DAG. The DAG explanation can be constructed in time $O(nl)$.* □

5. EXPERIMENTAL RESULTS

We now describe experimental results on real-world data showing that (1) the list of evidence we generate for the explanations are correct (2) our techniques can significantly reduce the amount of evidence; and (3) we can generate the explanations efficiently.

5.1 Experimental setting

We experimented on the AbeBooks data set, which was extracted in 2007 from *AbeBooks.com* by searching computer-science books[5]. In the data set there are 894 bookstores (data sources), 1265 books, and 24364 listings, each containing attributes ISBN, name, and often authors. According to a previous study, a naive voting for deciding the correct list of authors on this data set obtains an accuracy of only .71, while the advanced fusion technique in [9] obtains an accuracy of .89.

We generate explanations for four types of decisions: I. *truth discovery:* true value for the name and author list of each book; II. *copying detection*: copying or no-copying between sources whose Jaccard similarity on data items (intersection over union) is at least .1 (there are 3210 such pairs); III. *copy direction:* direction of copying between sources with detected copying (there are 1552 such pairs); IV. *copy pattern:* copying by object or copying by attribute [7] between sources with detected copying (there are 1340 detected patterns). We consider all for snapshot explanations, and I and II (only copying) for comprehensive explanations.

For snapshot explanation, we compare the following list shortening strategies: (1) TOPK: showing only evidence with the top-k scores; (2) LARGE: showing only evidence whose score is larger than 5; (3) TOPKLARGE: showing only the top-k evidence whose score is larger than 5; (4) CUTTAIL: applying Algorithm CUTTAIL; (5) KEEPDIFF: applying Algorithm KEEPDIFF with $\beta_1 = \beta_2 = .01$; and (6) SHORTEN: trying both CUTTAIL and KEEPDIFF, and selecting the results with shorter lists. By default, we apply SHORTEN.

For comprehensive explanation, we compare full explanation DAG, critical-round DAG, and enriched critical-round DAG. We also generated *trigger explanation DAGs*, where we show only *triggering* reasons at the critical round (*i.e.*, those that do not hold or are not strong enough in the previous round). By default, we used critical-round DAGs.

We used Java and experimented on a WindowsXP machine with 2.66GHz Intel CPU and 3.48GB of RAM. We hosted the database using MySQL.

5.2 Snapshot explanations

Shortening strategies: Fig.3 shows results of generated snapshot explanations for the four types of decisions. We have five observations. (1) Evidence categorization and aggregation shortens the evidence list by an order of magnitude on average. (2) List shortening further shortens the evidence list by 51% on average. (3) Evidence categorization and aggregation can reduce the size of explanations more for copying detection than for truth discovery, because the amount of raw evidence for the former, decided by the number of values provided by the sources, is much larger than that for the latter, decided by the number of sources providing the data item. (4) The final amount of evidence is the largest for decisions of Type I as each explanation involves multiple list explanations, then for those of Type II as each explanation involves two list explanations, and last for Type III and IV as each involves a single list explanation. (5) All evidence lists are correct.

As a case study, we observed that the largest explanation without shortening is for a Type II decision. The original explanation contains two lists, in total containing 4927 pieces of evidence. One can imagine how verbose the explanation could be if we give a detailed description of the Bayesian analysis. After categorization and aggregation, there are still 29 pieces of evidence in total. After list shortening the number further drops to 15.

We next compare different list shortening strategies. We first consider decisions of copying detection (Type II). Table 7 shows the average length of the result lists and Fig. 4 shows the *shortening ratio* (percentage of the size of the shortened lists over that of the full lists) for each method. We have four observations. (1) LARGE and TOP15LARGE obtain the shortest evidence lists; however, this

[5] We thank authors of [23] for providing us the data.

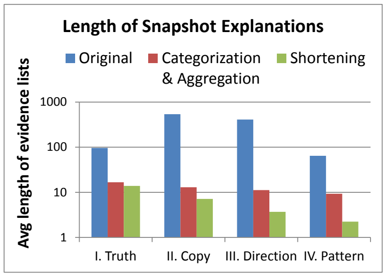


Figure 3: Length of explanations.

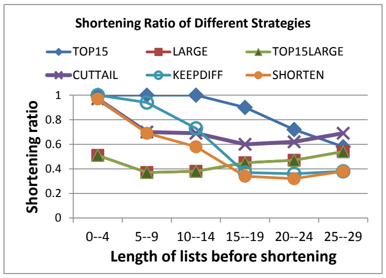


Figure 4: Shortening ratio.

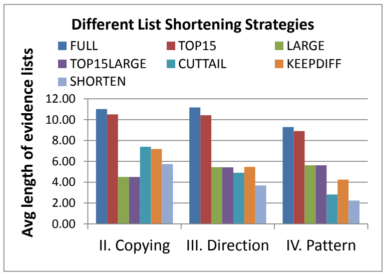


Figure 5: Shortening strategies.

Table 7: Average number of evidence in the explanations generated by each shortening strategy.

Full	Top15	Large	Top15Large	CutTail	KeepDiff	Shorten
11.0	10.5	4.5	4.5	7.4	7.2	5.7

Table 8: Errors in snapshot explanations.

	All	Top20	Top15	Top10
TopK	0	0	0	1
TopKLarge	47	47	47	48

is at the price of introducing errors (the sum of scores for positive evidence is no larger than that for negative evidence) in the explanations as they remove evidence without checking. As shown in Table 8, TopKLarge introduces errors for 47 (2.2%) pairs of sources; TopK in itself introduces only a few errors, but on the other hand, in reduces the list length only slightly. (2) Shorten obtains slightly longer evidence lists than Large and Top15Large, but does not introduce any error. (3) CutTail and KeepDiff obtain similar results in terms of the average length of the result lists; however, the former is better at shortening short lists (5-14 evidence) and the latter is better at shortening long lists (15-29 evidence). Shorten combines them and obtains shorter lists. (4) Finally, most methods have a lower shortening ratio for longer lists, whereas Large and TopKLarge have consistent ratio for lists of various length, and are able to significantly shorten very short lists (0-4 evidence), but this again is at the price of making errors.

We next consider decisions of Type III and IV; Fig.5 shows the length of evidence lists generated by different shortening strategies. The results are in general consistent with our observations for decisions of Type II and we have the following additional observations. First, before shortening, the list explanations for decisions of Type IV are short; since CutTail is better at shortening short lists (see Fig.4), the results of Shorten are affected more by CutTail. Second, for decisions of Type III and IV, each evidence typically has a high score, so Large and Top15Large under-shorten and generate longer lists than Shorten.

Efficiency: Table 9 shows efficiency of generating explanations for each type of decisions. We observe that (1) explanations can be generated very quickly online, and (2) the list shortening strategies introduce a very small overhead. Note that collecting evidence for decisions of Type II-IV all requires scanning provided data and took 62.3 ms on average. Note also that collecting evidence for truth discovery decisions requires computing copying probability for each shared value and thus took longer time.

5.3 Comprehensive explanations

Shortening strategies: The iterative Bayesian analysis on the experimental data set took 9 rounds. Fig.6 plots the size of the critical-round DAGs versus the critical round. We observe that for truth discovery decisions, those that do not change since the first round typically have a small DAG (with less than 15 nodes), whereas those changed at later rounds can have much larger DAGs (the largest DAG has 1035 nodes). In contrast, for copying detection decisions, the DAGs for decisions not changed since the first round have only 1 node (copying detection in the first round is based purely on provided data). Despite the fact that copying decisions typically require more inferred evidence than truth discovery decisions, the former typically have smaller DAGs than the latter; this is because a DAG for a copying decision often has only one node (the root) representing a copying decision, but a DAG for a truth discovery decision can often have several such nodes.

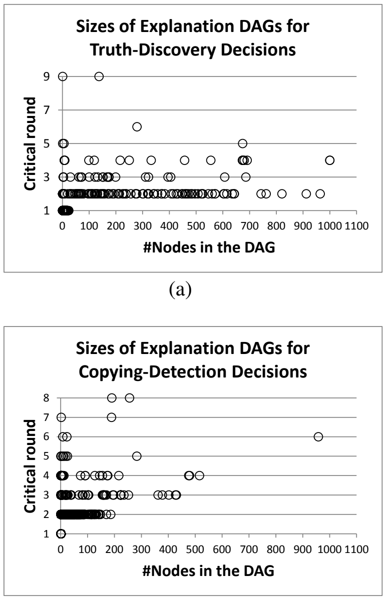
Table 9: Runtime of explanation generation.

(In ms)	I. Truth	II. Copy	III. Direction	IV. Pattern
Evid Collection	350.3	62.3		
Categorization	.08	31.2	8.8	.03
Shortening	.12	.01	.02	.01

Fig.7(a) compares the sizes of different types of DAGs for truth discovery decisions. We have four observations. (1) Most full DAGs either have only 1 node (69%), or have over 1000 nodes (30%), meaning that once a decision is not purely supported by provided data, the full explanation DAG is typically huge. (2) Most critical-round DAGs are small as they show only evidence at the critical rounds: 72.4% of the DAGs have 1 node, 92.6% have less than 10 nodes, and only 1 has more than 1000 nodes. (3) Trigger DAGs are even smaller (94.4% are of size 1) because they show only trigger evidence; however, they can miss important evidence. As shown in Fig.7(b), typically the later rounds at which the decisions are made, the higher percentage of evidence is missing. (4) Finally, enriched critical-round DAGs can be much larger than critical-round DAGs. We observe that on average there are .75 appearing evidence for decisions not changed since the first round, and nearly 0 appearing evidence for other decisions, and nearly 0 disappearing evidence for all decisions. However, explaining such additional evidence at a late round can significantly increase the size of the DAGs: 25.9% of the DAGs are of size larger than 10 and 15% have more than 1000 nodes.

Efficiency: Fig.8(a) compares the efficiency of generating comprehensive explanations from the database and directly from our log files. Constructing explanation DAGs from a database was very efficient: on average it took only 0.3 second and in the worst case it took 22 seconds. DAG construction from files on average took 283.5 times as long as that from a database. For DAGs with up to 10 nodes, using the database reduced runtime by 3 orders of magnitude; even for DAGs of size over 100, using the database reduced runtime by more than 1 order of magnitude.

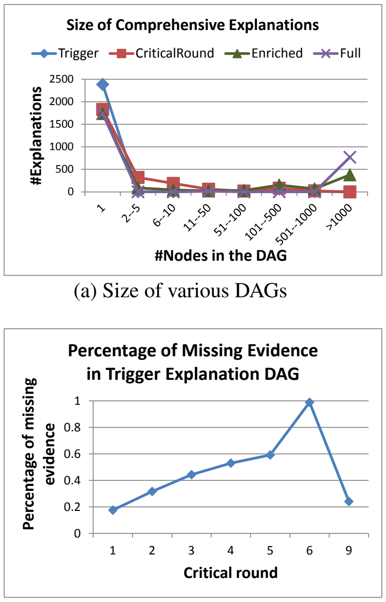
Fig.8(b) reports database creation time. We finished creating the database in 8.4 hours and the size of the database is 766MB. It is acceptable given that it is an offline process. We observe that populating tables for the first round took the longest time (4.6 hours), because most decisions are made at that round; starting from the third round, each round took less than half an hour. We also observe that generating explanation and evidence for accuracy comparison decisions took much longer time than other types of decisions, because there are many more such decisions in each round. Finally, it took 55 hours to create a database for constructing full explanation



(a)

(b)

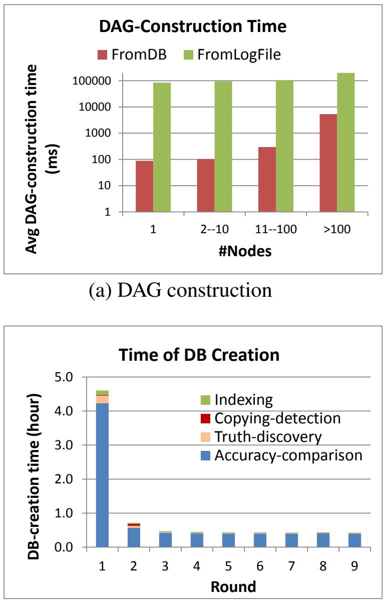
Figure 6: Distribution of sizes of critical-round DAGs.



(a) Size of various DAGs

(b) Trigger explanation DAGs

Figure 7: Size of comprehensive explanations.



(a) DAG construction

(b) Database creation

Figure 8: Efficiency of generating comprehensive explanations.

DAGs (13.7GB), as we need to generate explanation and evidence for each decision at each round; this further shows the huge overhead for generating full explanation DAGs.

6. RELATED WORK

Generating *provenance* (or *lineage*) information to facilitate understanding of data management and data integration results has received recent interest in the database community. Techniques have been proposed for explaining results for queries [4, 5, 14, 18], workflows [6], schema mappings [13], and information extraction [15, 21]. We are unaware of any existing work on explaining data fusion results. The following characteristics of our techniques distinguish our work from previous works.

First, we need to explain results from Bayesian reasoning, where one of our key contributions is evidence-list shortening. [11, 17] proposed explaining evidence propagation in Bayesian networks, but that is different from explaining Bayesian analysis. [14, 15] discussed reducing the number of returned reasons by applying constraints and declaring trust on certain data in information extraction. These techniques do not apply in our context; we instead consider evidence categorization, aggregation and list shortening.

Second, we need to explain results from iterative reasoning. Among existing work, only [21] considers iterative reasoning: it proposed querying all extraction patterns that contribute to an extracted tuple and all tuples that are affected by an extraction pattern over all iterations. By creating an explanation database, we support such queries in the context of data fusion as well, and we in addition generate the whole evidence DAG for comprehensive explanation. Finally, answers to provenance queries are also in the DAG structure and indexing techniques have been proposed recently for accelerating query evaluation [16]. Our techniques differ in that we leverage the repetition in the iterations to reduce the size of the explanation DAG and use a database to accelerate DAG construction.

7. CONCLUSIONS

In this paper we study explaining data fusion results and focus on recent techniques that conduct iterative Bayesian analysis. Targeting different types of users, we proposed snapshot explanations and comprehensive explanations. We showed how we efficiently generate such explanations and significantly reduce the size of the explanations. Our solutions are applicable in other applications that involve Bayesian analysis and iterative reasoning.

Future work includes combining evidence-style explanation with causality reasoning [18] to improve comprehensive explanation, applying our ideas in pinpointing important decisions, and improving data fusion results by seeking user feedback.

8. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *DL*, 2000.

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

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

[4] P. Buneman, J. Cheney, W.-C. Tan, and S. Vansummeren. Curated databases. In *Proc. of PODS*, 2008.

[5] A. Chapman and H. Jagadish. Why not? In *Sigmod*, 2009.

[6] S. Davidson and J. Freire. Provenance and scientific workflows: Challenges and opportunites. In *Proc. of SIGMOD*, 2008.

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

[8] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Solomon: Seeking the truth via copying detection. *PVLDB*, 2010.

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

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

[11] M. J. Druzdzel. Qualitative verbal explanations in bayesian belief networks. *Artificial Intelligence and Simulation of Behaviour Quarterly*, 94:43–54, 1996.

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

[13] B. Glavic, G. Alonso, R. J. Miller, and L. M. Haas. TRAMP: Understanding the behavior of schema mappings through provenance. *PVLDB*, 3(1), 2010.

[14] M. Herschel and M. A. Hernandez. Explaining missing answers to SPJUA queries. *PVLDB*, 3(1), 2010.

[15] J. Huang, T. Chen, A. Doan, and J. F. Naughton. On the provenance of non-answers to queries over extracted data. *PVLDB*, 1(1), 2008.

[16] A. Kementsietsidis and M. Wang. Provenance query evaluation: what's so special about it? In *CIKM*, 2009.

[17] C. Lacave, R. Atienza, and F. J. Diez. Graphical explanation in bayesian networks. *Lecture Notes in Computer Science*, 1933:122–129, 2000.

[18] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. The complexity of causality and responsibility for query answers and nonanswers. *PVLDB*, 4(1), 2010.

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

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

[21] A. D. Sarma, A. Jain, and D. Srivastava. I4e: interactive investigation of iterative information extraction. In *Sigmod*, 2010.

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

[23] 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.

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