

Global Detection of Complex Copying Relationships Between Sources

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

Yifan Hu
AT&T Labs-Research
yifanhu@research.att.com

Laure Berti-Equille^{*}
Université de Rennes 1
berti@irisa.fr

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

ABSTRACT

Web technologies have enabled data sharing between sources but also simplified copying (and often publishing without proper attribution). The copying relationships can be complex: some sources copy from multiple sources on different subsets of data; some co-copy from the same source, and some transitively copy from another. Understanding such copying relationships is desirable both for business purposes and for improving many key components in data integration, such as resolving conflicts across various sources, reconciling distinct references to the same real-world entity, and efficiently answering queries over multiple sources. Recent works have studied how to detect copying between a pair of sources, but the techniques can fall short in the presence of complex copying relationships.

In this paper we describe techniques that discover global copying relationships between a set of structured sources. Towards this goal we make two contributions. First, we propose a global detection algorithm that identifies co-copying and transitive copying, returning only source pairs with direct copying. Second, global detection requires accurate decisions on copying direction; we significantly improve over previous techniques on this by considering various types of evidence for copying and correlation of copying on different data items. Experimental results on real-world data and synthetic data show high effectiveness and efficiency of our techniques.

1. INTRODUCTION

Web technologies have enabled data sources to publish and share their data, but also made it easy for sources to copy from each other (and often publish without proper attribution). The copying relationships can be complex: some sources act as data hubs and aggregate data from multiple sources; some provide only a small set of data independently, copying the rest of the data from their “friend” sources, who may also copy from others; some sources are well known and widely copied by many other sources.

Understanding the copying relationship between sources and the data flow has many benefits [1]. First, data are valuable and many

data providers have put a lot of money and effort in collecting and cleaning their data, so they may want to understand such relationships for business purposes (and possibly protect their own rights). Second, in data integration, considering the copying relationship can help improve truth discovery, entity resolution, schema mapping, and further optimize query answering over multiple sources. Third, identifying provenance of data can be critical for in-depth data analysis and for the study of dissemination of information. Finally, independence of sources can form an important criteria in source (user) recommendation. This paper aims at discovering complex copying relationships between a set of sources, illustrated in the following example.

EXAMPLE 1.1. *We consider a data set extracted from AbeBooks.com; it includes 1263 CS books and 877 online bookstores (sources)¹. Our copying-detection model (explained in the paper) predicted that between 465 pairs of sources the probability of copying is above .5 (visualized in Appendix Fig. 10, generated by GMap [9]).*

First, we can cluster the bookstores by the copying relationship (as GMap does) and obtain some interesting clusters. For example, cluster Departmentstoria² includes many big department bookstores, such as A1Books.com, Quartermelon.com, and Powell’s Books; cluster Textbookistan includes many textbook stores such as www.textbooksrus.com, LGTextbooks, and brandnewtextbooks.

Second, copying relationships can be complex. Some sources (e.g., Deepak Sachdeva) seem to copy from multiple sources. Some sources (e.g., Browns Books) seem to be copied by multiple sources (and those co-copyers often do not copy from each other). Some sources seem to transitively copy from other sources; for example, Gunars Store or Gunter Koppon (one of them is a copier of the other, but the direction is unclear) transitively copies from World-OfBooks via Books Down Under. We found that a source can copy from up to 17 sources and be copied by up to 9 sources; and there are transitive paths (where the last source indeed copies data provided by the first) of length up to 9. □

Discovering copying relationships between structured sources has been studied recently in [6] for static data and in [7] for dynamic data (with updates). In particular, [6] makes pairwise decisions based on common mistakes made by the sources, and [7] considers in addition similarity of update patterns. Such techniques can detect source *dependence* and improve truth discovery, but may generate inaccurate *copying* relationships in the presence of complex copying. In particular, they have the following limitations.

¹We thank the authors of [13] for providing us the data.

²We named the clusters manually.

^{*}Mobility research program supported by the European Commission (Grant FP6-MOIF-CT-2006-041000).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Articles from this volume were presented at The 36th International Conference on Very Large Data Bases, September 13-17, 2010, Singapore.

Proceedings of the VLDB Endowment, Vol. 3, No. 1

Copyright 2010 VLDB Endowment 2150-8097/10/09... \$ 10.00.

First and most importantly, these techniques consider every pair of sources in isolation of other sources and make *local* decisions; as a result, they cannot distinguish co-copying, transitive copying, and direct copying from multiple sources. Second, they neglect possible correlations on copying of data items; for example, a source that copies the name of a book tends to also copy its author list. Third, they view common mistakes as important evidence of copying but neglect other kinds of evidence such as whether the data are formatted in the same way, and whether two sources provide similar sets of real-world objects. Experimental results show that the second and the third limitations often lead to wrong copying directions, which in turn can lead to wrong choices among co-copying, transitive copying, and multi-source copying.

This paper proposes techniques for *global* copying detection on static data and these techniques can be extended for dynamic data following the ideas in [7]. Our detection proceeds in two steps: the first step *locally* decides possibility of copying and copying direction between each pair of sources, and the second step *globally* identifies co-copying and transitive copying.

This paper makes three contributions. First, for making more accurate decisions on the copying direction, critical for global detection, we enhance the previous model by gleaning more evidence such as completeness and formatting of data (Sec.3), and considering correlated copying on data items (Sec.4). Second, as a key to global detection, we introduce the techniques for discovering co-copying and transitive copying, and distinguish them from a source indeed copying from multiple sources (Sec.5). Third, we experimented on both real-world data and synthetic data, showing effectiveness and efficiency of our techniques (Sec.6 and Appendix E).

2. OVERVIEW

This section defines the problem we solve and describes how we profile characteristics of data.

2.1 Problem definition

Consider a set of real-world objects in the same domain, denoted by $\mathcal{O}$. Each object is described by a set of attributes $\mathcal{A}$, among which we assume one uniquely identifies the objects (key)³; we call an attribute $A \in \mathcal{A}$ of an object $O \in \mathcal{O}$ a *data item* and denote it by $O.A$. An attribute value can be atomic (e.g., string, numeric value), or a set or list of atomic values (e.g., a list of authors, a set of phone numbers), which we consider as a whole. We assume for each non-key attribute an object has a *true* value that reflects the reality, and many wrong values⁴, but for the key attribute there cannot be any wrong value (we assume entity resolution is already performed using known techniques [10]). We assume as input, we know the probability of each non-key value v being true, denoted by $P(v)$ (we can compute such probabilities according to [6]).

Consider a set of sources, denoted by $\mathcal{S}$, each describing a subset of objects in $\mathcal{O}$. For each object, each source can provide values for a subset of attributes in $\mathcal{A}$ and we assume a key value must be provided. Different sources may format the same value differently (e.g., “John Smith” and “SMITH, John”); such formatting differences should be easily detectable according to some standardization or normalization rules⁵. For each attribute there is a set of *formatting elements* (e.g., for an author list, the elements can be

³It is easy to extend our techniques for the case with multi-attribute keys or the case where some attributes apply to only a subset of objects.

⁴Some wrong values are partially correct (e.g., misspellings and partial lists) and we can handle this case by considering value similarity as in [8].

⁵A standardization is not necessary; even if it is performed, the formatting information should not be discarded as it helps in copying detection.

Table 1: Sources in the motivating example.

	ISBN	name	authors
S_1	1	IPV6: Theory, Protocol, and Practice	Loshin, <i>Peter</i>
	2	Web Usability: A User-Centered Design Approach	Lazar, Jonathan
S_2	1	IPV6	-
	2	<i>Web Usability</i>	Jonathan Lazar
S_3	1	IPV6: Theory, Protocol, and Practice	Loshin, <i>Peter</i>
	2	<i>Web Usability</i>	Jonathan Lazar
S_4	1	IPV6: Theory, Protocol, and Practice	Loshin
	2	<i>Web Usability</i>	Lazar

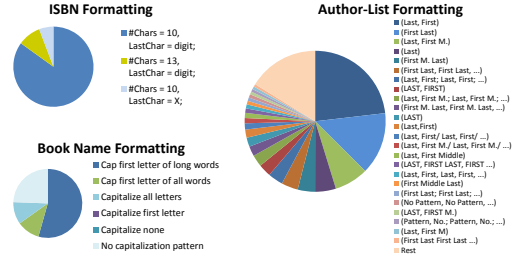


Figure 1: Formatting of attributes in the AbeBooks data set.

list completeness, name completeness, name component ordering, capitalization, etc.), each with a set of *options* (e.g., options for list completeness can be *full author list*, *only first author*, etc.). A *formatting pattern* is a combination of options of the formatting elements; such patterns can be pre-defined by observing the data. Note that some formatting patterns may contain less information than others (e.g., *only first author* vs. *full author list*) and thus they form a partial order.

EXAMPLE 2.1. Consider the four sources in Tbl.1, each providing data on the same two books. A book is described by its ISBN (the key), name, and authors (S_2 does not provide authors for book 1). The sources may provide incorrect values (marked in *italic font*); e.g., S_2 provides wrong (or partially correct) values for the name of both books. They may also provide the same value but format them differently; e.g., S_1 and S_2 provide the same authors for book 2 but in different formats.

Fig.1 shows the variety of observed formatting patterns on each attribute in the AbeBooks data set. We observe much higher variety on authors than on ISBN and name. □

Among the sources, some are *independent* and provide data independently, and some are *copiers* and copy all or a portion of data from other sources. A copier may verify some values and modify them when appropriate; we consider such values as independently provided, as they reflect independent observation of the real world by the copier. A copier may also reformat some copied values; we consider such values still as copied if the new format contains equal or less information (e.g., copying only the first author), and as independent if the new format contains more information (e.g., add more authors). Note that there is another kind of “dependence” between sources—*negative correlation* (e.g., data items provided by S_1 and S_2 are complementary, or S_1 chooses to provide different values or use different formats from S_2); in such cases, we consider neither source as a copier.

EXAMPLE 2.2. Continue with the motivating example. S_1 and S_2 are independent; S_3 copies the first book from S_1 and the second one from S_2 ; S_4 copies from S_3 but has reformatted the values of authors and provides only their last names. □

This paper aims to solve the following problem.

PROBLEM STATEMENT 1. Given a set of objects $\mathcal{O}$ and a set of sources $\mathcal{S}$, for each pair of sources $S, S' \in \mathcal{S}$, decide the probability of S directly copying from S' and vice versa. $\square$

We make a *closed-world* assumption on $\mathcal{O}$ and $\mathcal{S}$. This assumption on $\mathcal{O}$ should not affect the results much. That on $\mathcal{S}$, however, can lead to predicting direct copying between co-copyers of a hidden source or between a source and its transitive copier when the direct copier is hidden. We assume there is no *mutual copying*; that is, S_1 copies from S_2 and S_2 copies from S_1 (on different objects).

2.2 Data and source profiling

A source is more likely to be a copier if the probability that it provides the observed data independently is very low. Judging this would require computing the a-priori probability that a particular source provides some particular data. There are several variations in data for a particular data item, including, but not limited to, whether a value is provided, which value is provided, and in which format the value is provided. We thus profile the data by *completeness*, *accuracy*, and *formatting style*, respectively, and one can define other measures similarly. These profiling measures typically fall in one of the three classes: *existence measure* measures whether a piece of data exists (e.g., *completeness*); *correctness measure* measures correctness of data (e.g., *accuracy*); and *distribution measure* measures distribution of values, formats, etc. (e.g., *formatting style*).

Note that the probability that a source provides a piece of data can depend both on source-wise statistics and data-item-wise statistics; for example, S is likely to provide an object O if S has a high completeness or O is popular. Thus, we need to define each measure both for each source and for each data item.

Completeness: The *object-level completeness* of a source S , denoted by $C_O(S)$, measures the percentage of objects in $\mathcal{O}$ that S provides. The *completeness of an object* O , denoted by $C(O)$, measures the percentage of sources in $\mathcal{S}$ that provide O . Similarly, we can define attribute-level completeness.

Note that in the presence of copiers, we want to avoid being biased by them when computing completeness and other measures; for example, an object may seem popular, but most of its providers just copy data for it from a common source. We may wish to consider only independent providers; e.g., we can compute $C(O)$ by

$$C(O) = \frac{\sum_{S \in \bar{S}(O)} P(S, O)}{\sum_{S \in \bar{S}(O)} P(S, O) + |\mathcal{S} - \bar{S}(O)|}, \quad (1)$$

where $\bar{S}(O)$ is the set of sources that provide O , $P(S, O)$ denotes the probability that S independently provides O (its computation depends on the result of copying detection), and so $\sum_{S \in \bar{S}(O)} P(S, O)$ computes the “number” of independent providers for O and $|\mathcal{S} - \bar{S}(O)|$ counts the number of sources that do not provide O .

Formatting style: The *formatting style* of a source S measures the distribution of formatting patterns used by S . For each formatting pattern f for $A \in \mathcal{A}$, we compute its popularity, $F_{A,f}(S)$, as the percentage of objects for which S uses f for the value of A . The *formatting style of an item* $O.A$ measures the distribution of formatting patterns on $O.A$ used by different sources. We compute $F_{A,f}(O.A)$ as the percentage of sources that use f for the value of $O.A$ among all providers of $O.A$.

Accuracy: The *accuracy* of a source S measures correctness of its data. We adopt techniques presented in [6] and compute accuracy by $A(S) = \text{Avg}_{v \in \bar{V}(S)} P(v)$, where $\bar{V}(S)$ is the set of values provided by S . We can easily refine this measure for each attribute. The corresponding measure for $O.A$ is the correctness of each of $O.A$ ’s values v and is captured by $P(v)$.

We next illustrate usage of these measures in copying detection.

EXAMPLE 2.3. Consider S_1, S_2 and S_3 in Table 1. “Peter Loshin” has a misspelling (the correct spelling is “Pete Loshin”) and has a low value probability, so indicates dependence between S_1 and S_3 . Similarly, “Web Usability” is a wrong book name and has a low value probability, so indicates dependence between S_2 and S_3 . It is more likely that S_3 copies from S_1 and S_2 than the opposite direction, as S_3 keeps the format of the copied data and thus formats authors differently for the two books, so the popularity of each formatting pattern is low (50%). $\square$

Among these measures, we note that (1) as we show later, source copying probabilities depend on all measures we have defined; (2) the copying probabilities affect item-wise measures if we compute them considering only independent sources (e.g., by Eq.(1)), but do not affect source-wise measures; (3) the item-wise measure and the source-wise measure are independent of each other for completeness and formatting styles; however, source accuracy and value probability are inter-dependent, unless value probabilities are given upfront as input. Therefore, there is inter-dependence between data-item-wise measures, source-wise measures, and copying probabilities; we compute them iteratively until convergence, as detailed in [6].

3. A BASIC LOCAL-DETECTION MODEL

We now present the basic model for local copying detection assuming *item-wise independence*; that is, whether source S copies an item $O.A$ from S' is independent of whether it copies $O'.A'$ from S' , $O \neq O'$ or $A \neq A'$. We relax this assumption in Sec.4. Consider two sources $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$. The key in deciding whether S_1 copies from S_2 ($S_1 \rightarrow S_2$) is to decide if the probability of S_1 providing the observed data conditioned on it being independent of S_2 is much lower than that conditioned on it being a copier of S_2 . Intuitively, the former probability will be much lower than the latter in two cases: first, when the two sources share low-completeness items, low-probability values, or low-popularity formats; second, when there is a big difference between the profile of the overlapping data and that of S_1 ’s self-provided data.

Specifically, according to the *item-wise independence* assumption, we consider each data item and denote by $\Phi_{O.A}(S)$ the data provided by S on $O.A$. We say $\Phi_{O.A}(S) = \emptyset$ if S does not provide a value for $O.A$, and $\Phi_O(S) = \emptyset$ if S does not provide a value for O .key (and so not for any other attribute either). We denote the two conditional probabilities by $P(\Phi_{O.A}(S_1)|S_1 \not\rightarrow S_2)$ (simplified hereafter as $P(\Phi_{O.A}(S_1))$) and $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$ respectively. We next focus on how we compute them according to our data profiling; our methods can be easily extended when other measures are present. The complete basic model applies Bayesian analysis by accumulating evidence from all data items and we give details in Appendix A; we give proofs of the theorems in [5].

Not copying: We start with $P(\Phi_{O.A}(S_1))$. Here, S_1 does not rely on S_2 and there are three cases:

1. S_1 does not provide $O.A$ and A is the key. Then S_1 does not provide O (i.e., $\Phi_O(S_1) = \emptyset$) and
$$P(\Phi_{O.A}(S_1)) = 1 - P(\Phi_O(S_1) \neq \emptyset). \quad (2)$$
2. S_1 does not provide $O.A$ and A is not the key. Then
$$P(\Phi_{O.A}(S_1)|\Phi_O(S_1) \neq \emptyset) = 1 - P(\Phi_{O.A}(S_1) \neq \emptyset); \quad (3)$$

$$P(\Phi_{O.A}(S_1)|\Phi_O(S_1) = \emptyset) = 1. \quad (4)$$
3. Otherwise, suppose S_1 provides a value v and formats it in pattern f . Then,
$$P(\Phi_{O.A}(S_1)) = P(\Phi_{O.A}(S_1) \neq \emptyset) \cdot P(\text{value}(\Phi_{O.A}(S_1)) = v) \cdot P(\text{format}(\Phi_{O.A}(S_1)) = f). \quad (5)$$

Computation of $P(\Phi_O(S_1) \neq \emptyset)$ depends on the completeness of S_1 and O (similar for $P(\Phi_{O.A}(S_1) \neq \emptyset)$), that of $P(\text{value}(\Phi_{O.A}(S_1)) = v)$ depends on S_1 's accuracy and v 's correctness, and that of $P(\text{format}(\Phi_{O.A}(S_1)) = f)$ depends on the formatting styles of S and $O.A$. We now present how we compute $P(\Phi_O(S_1) \neq \emptyset)$ and give details for the rest of the probabilities in Appendix A.

Intuitively, a source S provides $|\mathcal{O}|C_O(S)$ objects, so the probabilities for providing each object should sum up to $|\mathcal{O}|C_O(S)$; similarly, the probabilities of each source providing O should sum up to $|S|C(O)$. We thus shall solve the following equations:

$$\forall S \in \mathcal{S}, \sum_{O \in \mathcal{O}} P(\Phi_O(S) \neq \emptyset) = |\mathcal{O}|C_O(S); \quad (6)$$

$$\forall O \in \mathcal{O}, \sum_{S \in \mathcal{S}} P(\Phi_O(S) \neq \emptyset) = |S|C(O). \quad (7)$$

There are $|S| \cdot |\mathcal{O}|$ variables but only $|S| + |\mathcal{O}|$ equations, so an infinite number of solutions. We choose the one with the maximum entropy [4], so has the least bias. We can prove that in most cases⁶ such a solution is obtained when we assume the probability that S provides each object O is proportional to $C(O)$; thus,

$$P(\Phi_O(S_1) \neq \emptyset) = \frac{|\mathcal{O}|C_O(S_1)C(O)}{\sum_{O_0 \in \mathcal{O}} C(O_0)} = \frac{|S|C(O)C_O(S_1)}{\sum_{S_0 \in \mathcal{S}} C_O(S_0)}. \quad (8)$$

Copying: We next compute $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$. Note that even if S_1 copies from S_2 , S_1 may or may not copy a particular data item. We call the probability of copying a particular item the *selectivity*, and denote it by s . Even when S_1 copies, it can then choose to keep the original format, or to reformat it; we assume the probability of the former is k . We discuss how to set s and k in Appendix B. The computation requires comparing values and formats provided by S_1 and S_2 ; there are four possible cases.

1. *One of S_1 and S_2 does not provide $O.A$.* We do not penalize providing a value that the other source does not provide or vice versa (common for a copier), so

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = P^c(\Phi_{O.A}(S_1)). \quad (9)$$

For the probability that a copier independently provides a piece of data, we mark by ^c and explain in more detail shortly.

2. *S_1 and S_2 provide different values on $O.A$ or S_1 uses a format with richer information.* Then, S_1 does not copy:

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = (1-s)P^c(\Phi_{O.A}(S_1)). \quad (10)$$

3. *S_1 provides the same value as S_2 but in a different format f' (f' contains no richer information than that of S_2).* Then, S_1 might copy (w. probability s) but reformat:

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = (1-s)P^c(\Phi_{O.A}(S_1)) + s(1-k)P^c(\text{format}(\Phi_{O.A}(S_1)) = f'). \quad (11)$$

4. *S_1 provides the same value in the same format f .* Then, S_1 might copy (w. probability s) and might follow the original format (w. probability k):

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = (1-s)P^c(\Phi_{O.A}(S_1)) + s(k + (1-k)P^c(\text{format}(\Phi_{O.A}(S_1)) = f)). \quad (12)$$

We note that we use $P^c(\Phi_{O.A}(S_1))$ instead of $P(\Phi_{O.A}(S_1))$ under condition of copying. We compute $P^c(\Phi_{O.A}(S_1))$ in the same way as $P(\Phi_{O.A}(S_1))$, except that we use “independent” measures computed only on S_1 's data that are not copied. As we often do not know in advance which data are copied, and such computation needs to be performed for every pair of sources so needs to

⁶The only exception happens when $\frac{|\mathcal{O}|C_O(S_1)C(O)}{\sum_{O_0 \in \mathcal{O}} C(O_0)} > 1$ for some S_1 and O ; in this case, we can estimate by setting $P(\Phi_O(S_1) \neq \emptyset) = 1$ for such S_1 and O , and compute by Eq.(8) for other sources and objects.

Table 2: Ex. 3.2 and 4.1. Each table describes data provided by two sources on 5 objects, each with 5 attributes (K is the key). “S” in the table indicates that the two sources provide the same value in the same format, and “D” indicates that they provide different values. Copying seems more likely for (b) than for (a).

	K	A_1	A_2	A_3	A_4
O_1	S	S	S	D	D
O_2	S	D	S	S	D
O_3	S	S	D	S	D
O_4	S	S	S	D	S
O_5	S	D	S	S	S

(a)

	K	A_1	A_2	A_3	A_4
O_1	S	S	S	S	S
O_2	S	S	S	S	S
O_3	S	S	S	S	S
O_4	S	D	D	D	D
O_5	S	D	D	D	D

(b)

be very fast, we estimate these measures. As an example, we estimate S_1 's “independent” object-level completeness w.r.t. S_2 as its completeness on objects not provided by S_2 :

$$C_O(S_1|\neg S_2) = \frac{|\bar{O}(S_1)| - |\bar{O}(S_1) \cap \bar{O}(S_2)|}{|\mathcal{O}| - |\bar{O}(S_2)|}, \quad (13)$$

where $\bar{O}(S)$ denotes the set of objects provided by S .

Discussion: The following result lists positive evidence for copying, conforming to our intuitions.

PROPOSITION 3.1. *Given sources S_1 and S_2 and data item $O.A$, in the following cases $O.A$ forms positive evidence for $S_1 \rightarrow S_2$.*

- S_1 provides the same value in the same format as S_2 on $O.A$, and $P(\Phi_{O.A}(S_1)) < sk$;
- S_1 provides the same value but uses a different format f' , and $P(\Phi_{O.A}(S_1)) < s(1-k)P^c(\text{format}(\Phi_{O.A}(S_1)) = f')$;
- $P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$ and S_2 does not provide $O.A$;
- $(1-s)P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$. $\square$

EXAMPLE 3.2. *Consider sources S_1 and S_2 , each providing 5 objects and 5 attributes for each object (shown in Tbl. 2(a)). Assume $P(\Phi_{O.K}(S_1)) = P^c(\Phi_{O.K}(S_1)) = .9$, $P(\Phi_{O.A_i}(S_1)) = P^c(\Phi_{O.A_i}(S_1)) = .5$, $i \in [1, 4]$, and $P^c(\text{format}(\Phi_{O.A}(S_1))) = .8$ for each attribute A . We set $s = .6$, $k = .5$.*

We first compute $P(\Phi(S_1)|S_1 \rightarrow S_2)$ ($\Phi(S_1)$ denotes observation of S_1 's data). There are 5 key items on which the two sources provide the same value in the same format; the probability is $(1-.6).9+.6*(.5+(1-.5)*.8) = .4*.9+.6*.9 = .9$ (Eq.(12)). There are 12 non-key items on which the two sources provide the same value in the same format; the probability is $.4*.5+.6*.9 = .74$ (Eq.(12)). Finally, there are 8 items on which the two sources provide different values and the probability is $.4*.5 = .2$ (Eq.(10)). So $P(\Phi(S_1)|S_1 \rightarrow S_2) = .9^5 * .74^{12} * .2^8$.*

*On the other hand, it is obvious that $P(\Phi(S_1)|S_1 \not\rightarrow S_2) = .9^5 * .5^{20}$. So $\frac{P(\Phi(S_1)|S_1 \rightarrow S_2)}{P(\Phi(S_1)|S_1 \not\rightarrow S_2)} = .07$ and S_1 is unlikely to be a copier of S_2 . This is reasonable because S_1 provides a lot of values differently from S_2 , and for the values they share, S_1 has a relatively high probability to provide them by itself. $\square$*

Comparison with [6]: There are three differences between our basic model and the model presented in [6].

1. The basic model allows the flexibility of plugging in evidence of various types, including completeness and formatting of data in addition to correctness of data.
2. In addition to source-wise measures, we consider also item-wise measures when computing $P(\Phi_{O.A}(S))$.
3. Instead of using $P(\Phi_{O.A}(S_1))$, we use $P^c(\Phi_{O.A}(S_1))$ under condition $S_1 \rightarrow S_2$.

Note that difference 3 is a correction of the previous model; however, our experiments show that it does not necessarily improve the results when we consider only data correctness. Finally, none of the techniques in Sec.4-5 is included in [6].

4. CORRELATED COPYING

The basic model assumes *item-wise independence*, which seldom holds in reality. For example, the copier may compose a SQL query and copy all returned objects; when it copies an object, it often copies all provided attributes or the attributes in its own schema. This section discusses object copying (the latter); similar techniques can be applied for query-driven copying (the former).

One can imagine that a copier often copies in one of two modes: 1) it copies a subset of objects on a subset of attributes, called *per-object copying*; 2) it copies on a subset of attributes for a set of independently provided objects (or objects copied from other sources), called *per-attribute copying*. The difference is whether the copier also copies the key values or not. Thus, when we compute $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$, we need to consider whether S_1 copies on O or only on A and treat them differently.

Specifically, we denote by $S_1 \xrightarrow{O} S_2$ that S_1 copies O from S_2 , by $S_1 \xrightarrow{O.A} S_2$ that S_1 copies $O.A$ from S_2 , and simplify $P(S_1 \xrightarrow{O} S_2|S_1 \rightarrow S_2)$ as $s(O)$ ($s(O)$ can be viewed as the selectivity on O for $S_1 \rightarrow S_2$, but we omit $S_1 \rightarrow S_2$ for simplicity). Then, we have

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = s(O)P(\Phi_{O.A}(S_1)|S_1 \xrightarrow{O} S_2) + (1 - s(O))P(\Phi_{O.A}(S_1)|S_1 \xrightarrow{O} S_2, S_1 \rightarrow S_2). \quad (14)$$

The computation of $P(\Phi_{O.A}(S_1)|S_1 \xrightarrow{O} S_2)$ and $P(\Phi_{O.A}(S_1)|S_1 \xrightarrow{O} S_2, S_1 \rightarrow S_2)$ is the same as $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$, except that we shall set the selectivity differently. We denote the selectivity for an attribute $A \in \mathcal{A}$ of a copied object by $s_A^\rightarrow$ and that of an uncopied object (per-attribute copying) by $s_A^\leftarrow$. According to our definition, $s_{key}^\rightarrow = 1$ and $s_{key}^\leftarrow = 0$.

The next question is how to compute $s(O)$ depending on our observation of data provided on O . Let s_{obj} be the a-priori probability that a copier copies an object. Then, by Bayesian analysis,

$$s(O) = P(S_1 \xrightarrow{O} S_2 | \Phi_O(S_1), S_1 \rightarrow S_2) = \frac{s_{obj} P(\Phi_O(S_1) | S_1 \xrightarrow{O} S_2)}{s_{obj} P(\Phi_O(S_1) | S_1 \xrightarrow{O} S_2) + (1 - s_{obj}) P(\Phi_O(S_1) | S_1 \xrightarrow{O} S_2, S_1 \rightarrow S_2)} \quad (15)$$

We can compute $P(\Phi_O(S_1) | S_1 \xrightarrow{O} S_2)$ from $P(\Phi_{O.A}(S_1) | S_1 \xrightarrow{O} S_2)$, $A \in \mathcal{A}$, assuming all attributes are independent (we can relax this assumption by further grouping the attributes).

One big challenge for applying the enhanced model is parameter setting. The enhanced model introduces new parameters, s_{obj} , $s_A^\rightarrow$, and $s_A^\leftarrow$; they are essentially conditional probabilities and can vary from source pair to source pair. Setting them appropriately is important in achieving accurate results. We set them for each direction of each pair of sources in two steps: first, we initialize them empirically according to the data; second, we adjust them later according to copying-detection results and re-apply our model accordingly. Appendix B gives the details.

EXAMPLE 4.1. Continue with Ex.3.2 and now consider S_1 and S_2 in Tbl. 2(b). With the same calculations as in Ex.3.2, S_1 appears unlikely to be a copier of S_2 . Now consider per-object copying and we set $s_{obj} = .6$ and $s_A^\rightarrow = .9$, $s_A^\leftarrow = .1$ for each attribute.

For each non-key item in $\{O_1, O_2, O_3\}$, if S_1 copies the object, the probability that it provides the data is $.1 * .5 + .9 * .9 = .86$ (Eq.(12)); otherwise, if S_1 copies from S_2 but not on the object, the probability becomes $.9 * .5 + .1 * .9 = .54$. For each non-key item in $\{O_4, O_5\}$, if S_1 copies the object, the probability is $.1 * .5 = .05$ (Eq.(10)); otherwise, the probability is $.9 * .5 = .45$. Finally, for each key attribute, the probability that S_1 provides it is always .9.

Table 3: Three sets of data sources. In each one, S_1 and S_2 share 50 values, S_1 and S_3 share 50 values, and S_2 and S_3 share 30 values.

Src	$\mathcal{D}_1$ (Multi-copy)	$\mathcal{D}_2$ (Co-copy)	$\mathcal{D}_3$ (Transitive)
S_1	$v_1 \sim v_{100}$, where $v_{81} \sim v_{100}$ are popular values		
S_2	$v_1 \sim v_{50}, v_{101} \sim v_{130}$	$v_1 \sim v_{50}$	$v_1 \sim v_{50}$
S_3	$v_{51} \sim v_{130}$	$v_{21} \sim v_{70}$	$v_{21} \sim v_{50}, v_{81} \sim v_{100}$

Accordingly, the probability that S_1 copies O_1, O_2 or O_3 is $\frac{.6 * (.9 * .86^4)}{.6 * (.9 * .86^4) + .4 * (.9 * .54^4)} = .9$ (Eq.(15)). The probability that it provides each non-key value is thus $.9 * .86 + .1 * .54 = .83$ (Eq.(14)). Similarly, the probability that S_1 copies O_4 or O_5 is .002 and the probability that it provides each non-key value is .45. Therefore, $\frac{P(\Phi(S_1)|S_1 \rightarrow S_2)}{P(\Phi(S_1)|S_1 \not\rightarrow S_2)} = \frac{.9^5 * .83^{12} * .45^8}{.9^5 * .5^{20}} = 188$ and S_1 is likely to be a copier of S_2 .

To compare, for Tbl.2(a), considering per-object copying obtains a ratio of 1.37 and still does not strongly imply copying. $\square$

5. GLOBAL COPYING DETECTION

Local detection aims at discovering (positive) dependence between sources; however, such dependence is not always caused by direct copying, but can also be due to co-copying or transitive copying. In the motivating example (Tbl.1), local detection may conclude with $S_4 \rightarrow S_1$ and $S_4 \rightarrow S_2$, although S_4 only transitively copies from S_1 and S_2 . Global detection tries to fix this problem. However, identifying co-copying and transitive copying is non-trivial, as the following example illustrates.

EXAMPLE 5.1. Consider S_1, S_2 and S_3 , where local detection decides that $S_2 \rightarrow S_1, S_3 \rightarrow S_1$, and $S_3 \rightarrow S_2$. We need to decide if S_3 copies only from S_1 (co-copying with S_2), only from S_2 (transitively copying from S_1), or from both (multi-source copying).

One may consider finding a clue from the copying probabilities, but this often does not work when each pair of sources share a lot of values and thus have a copying probability of 1.

One may then consider comparing the numbers of overlapping values, but this is insufficient. Consider $\mathcal{D}_1$ and $\mathcal{D}_2$ in Tbl.3. Each pair of sources share the same number of values for the two cases; however, because of the different distribution of the shared values, $\mathcal{D}_1$ seems more likely to have multi-source copying, whereas $\mathcal{D}_2$ seems more likely to have co-copying.

One may next consider comparing the sets of overlapping values, but this is still insufficient. Consider $\mathcal{D}_2$ and $\mathcal{D}_3$ in Tbl.3. Values that all of S_1, S_2 and S_3 share are the same ($v_{21} \sim v_{50}$) in the two cases; however, $\mathcal{D}_2$ seems more likely to have a co-copying, whereas $\mathcal{D}_3$ seems more likely to have a transitive copying, because the rest of the 20 values shared between S_1 and S_3 in $\mathcal{D}_3$ are popular values and S_3 may provide them independently. Thus, we need to reason for each data item in a more principled way. $\square$

Our key intuition is that since co-copying and transitive copying can often be inferred from direct copying, we first find a set of copying relationships $\mathbf{R}$ that significantly influence the rest of the relationships, and then adjust the rest accordingly and decide if each is a direct or indirect copying (the results are denoted by $P(S_1 \rightarrow S_2 | \mathbf{R})$, $(S_1, S_2) \notin \mathbf{R}$). In this process we need to solve two problems: (1) how to select the set $\mathbf{R}$; and (2) how to compute $P(S_1 \rightarrow S_2 | \mathbf{R})$. The first problem relies on the second, so we start from our solution for the second problem.

Computing $P(S_1 \rightarrow S_2 | \mathbf{R})$: As we have illustrated, we cannot derive this probability directly from the copying probabilities in $\mathbf{R}$, but have to reason for each data item if S_1 is likely to copy it from S_2 even in the presence of the copying between S_1 and other sources. Thus, when we compute $P(S_1 \rightarrow S_2 | \mathbf{R})$, we replace

$P(\Phi_{O.A}(S_1))$ everywhere with $P(\Phi_{O.A}(S_1)|\mathbf{R})$, the probability that S_1 independently provides the data or copies the data from other sources according to $\mathbf{R}$. We next illustrate how we compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ using the case of $\Phi_{O.A}(S_1) \neq \emptyset$.

Consider the set of sources that are associated with S_1 by some copying relationship in $\mathbf{R}$. Consider two subsets: $\tilde{S}_f(O.A)$, those providing the same value in the same format on $O.A$ as S_1 ; $\tilde{S}_v(O.A)$, those providing the same value in a different format. The probability that S_1 does not copy $O.A$ from any source in $\tilde{S}_f(O.A)$ is

$$P_f = \sum_{S \in \tilde{S}_f(O.A)} (1 - P(S_1 \rightarrow S)P(S_1 \xrightarrow{O.A} S)) \cdot (k + (1 - k)P(\text{format}(\Phi_{O.A}(S_1)) = f)) \quad (16)$$

where $P(S_1 \rightarrow S)$ and $P(S_1 \xrightarrow{O.A} S)$ are inferred from $\mathbf{R}$. Similarly, we can compute the probability that S_1 does not copy $O.A$ from any source in $\tilde{S}_v(O.A)$ and provide the observed format, denoted by P_v . Then S_1 either provides the data by itself (with probability $P_v P_f$), or copies $O.A$ from $\tilde{S}_v(O.A)$ or $\tilde{S}_f(O.A)$, so

$$P(\Phi_{O.A}(S_1)|\mathbf{R}) = (1 - P_v P_f) + P_v P_f P(\Phi_{O.A}(S_1)). \quad (17)$$

EXAMPLE 5.2. Continue with Ex.5.1. Consider $\mathcal{D}_1$ and suppose $\mathbf{R} = \{(S_1, S_3)\}$. For each $v \in \{v_{101}, \dots, v_{130}\}$, $P(\Phi_v(S_3)) = P(\Phi_v(S_3)|\mathbf{R})$, so S_3 still looks like a copier of S_2 .

Consider $\mathcal{D}_2$ and suppose $\mathbf{R} = \{(S_1, S_3)\}$. For each $v \in \{v_{21}, \dots, v_{50}\}$, $P(\Phi_v(S_3)|\mathbf{R})$ is much larger than $P(\Phi_v(S_3))$, so S_3 looks much less likely a copier of S_2 .

Finally, consider $\mathcal{D}_3$ and suppose $\mathbf{R} = \{(S_2, S_3)\}$. For each $v \in \{v_{21}, \dots, v_{50}\}$, again $P(\Phi_v(S_3)|\mathbf{R})$ is much larger than $P(\Phi_v(S_3))$; for each $v \in \{v_{81}, \dots, v_{100}\}$, $P(\Phi_v(S_3)|\mathbf{R}) = P(\Phi_v(S_3))$ but is high. Thus, S_3 looks less likely a copier of S_1 . $\square$

Finding $\mathbf{R}$: Finding a proper set of relationships for $\mathbf{R}$ is crucial. As an example, for $\mathcal{D}_2$ in Tbl.3, if we include $S_3 \rightarrow S_2$ in $\mathbf{R}$, we will not be able to detect the real copying $S_3 \rightarrow S_1$. We wish to include in $\mathbf{R}$ the most influential copying relationships; that is, our goal is to find the set $\mathbf{R}$ that maximizes

$$\psi(\mathbf{R}) = \sum_{(S_1, S_2) \notin \mathbf{R}} (P(S_1 \rightarrow S_2) - P(S_1 \rightarrow S_2|\mathbf{R})). \quad (18)$$

(We shall consider only positive influence and if $P(S_1 \rightarrow S_2) - P(S_1 \rightarrow S_2|\mathbf{R}) < 0$, we set it to 0.) We can prove the NP-hardness of this problem by a reduction from the HITTING SET problem.

THEOREM 5.3. The problem of finding the most influential copying relationships $\mathbf{R}$ is NP-complete. $\square$

We now present several observations, based on which we propose a fast greedy algorithm. The first observation is that according to Eq.(17), $P(S_1 \rightarrow S_2|\mathbf{R})$ relies on only relationships involving S_1 . Thus, we can construct $\mathbf{R}$ by finding for each source the most “influential” sources among those it may copy from.

PROPOSITION 5.4. If we denote by $\bar{D}(S_1)$, $S_1 \in \mathcal{S}$, the sources with which S_1 has a copying relationship in $\mathbf{R}$, by $\mathbf{R}(S_1)$ the relationships in $\mathbf{R}$ involving S_1 , and by $\Delta(S_1 \rightarrow S_2|\bar{D}(S_1)) = P(S_1 \rightarrow S_2) - P(S_1 \rightarrow S_2|\mathbf{R}(S_1))$, we have $\psi(\mathbf{R}) = \sum_{S_1 \in \mathcal{S}} \sum_{S_2 \notin \bar{D}(S_1)} \Delta(S_1 \rightarrow S_2|\bar{D}(S_1))$. $\square$

The second observation reveals the relationship between the joint influence of sources in $\bar{D}(S_1)$ and the individual influence of each of them. Accordingly, we can simplify our algorithm by considering influence of an individual copying relationship on another.

PROPOSITION 5.5. If we denote by $\Delta(S_1 \rightarrow S_2|S) = P(S_1 \rightarrow S_2) - P(S_1 \rightarrow S_2|\{S_1 \rightarrow S\})$, then, (1) $\Delta(S_1 \rightarrow S_2|\bar{D}(S_1)) \geq \Delta(S_1 \rightarrow S_2|S)$ for each $S \in \bar{D}(S_1)$; and (2) $\Delta(S_1 \rightarrow S_2|\bar{D}(S_1)) \leq \sum_{S \in \bar{D}(S_1)} \Delta(S_1 \rightarrow S_2|S)$. $\square$

Based on Proposition 5.5, we wish to greedily select relationships that have the highest accumulated influence on others. The next two observations state which relationships should be pruned (not added to $\mathbf{R}$). The third observation shows that we want to prune relationships that cause less accumulated changes on others than being affected by others; its proof is based on Proposition 5.5.

PROPOSITION 5.6. For any $S \in \mathcal{S}$, if there exist S_1 and S_2 , $S_1 \neq S_2$, such that (1) $\Delta(S \rightarrow S_1|S_2) > \sum_{S_0 \neq S, S_1, S_2} \Delta(S \rightarrow S_0|S_1)$, (2) $(S, S_2) \in \mathbf{R}$, and (3) $\mathbf{R}' = \mathbf{R} \cup \{(S, S_1)\}$, then $\psi(\mathbf{R}) > \psi(\mathbf{R}')$. $\square$

The final observation shows that we should prune a relationship if it can be significantly influenced by those already selected into $\mathbf{R}$, because it is more likely to be a co-copying or transitive copying and its influence on others will be dominated by the relationships in $\mathbf{R}$.

OBSERVATION 5.7. For any $S \in \mathcal{S}$, if there exist S_1 and S_2 , $S_1 \neq S_2$, such that (1) $\Delta(S \rightarrow S_1|\{S \rightarrow S_2\}) > .5$, (2) $(S, S_2) \in \mathbf{R}$, and (3) $\mathbf{R}' = \mathbf{R} \cup \{(S, S_1)\}$, then typically $\psi(\mathbf{R}) > \psi(\mathbf{R}')$. $\square$

Based on these observations, for each source $S \in \mathcal{S}$ our algorithm proceeds in four steps.

1. Find all sources from which S is likely to copy or copying is likely but the direction is unclear, denoted by $\bar{R}(S)$.
2. For each $S_1, S_2 \in \bar{R}(S)$, $S_1 \neq S_2$, compute $\Delta(S \rightarrow S_1|S_2)$ and $\Delta(S \rightarrow S_2|S_1)$ (influence). For each $S' \in \bar{R}(S)$, compute $\sigma(S') = \sum_{S_0 \neq S, S'} \Delta(S \rightarrow S_0|S')$ (influence on others) and $\Lambda(S') = \max_{S_0 \neq S, S'} \Delta(S \rightarrow S'|S_0)$ (max influence by others).
3. Find the source S' with the highest $\sigma(S')$ (most influential) and remove it from $\bar{R}(S)$. If $\sigma(S') > \Lambda(S')$ (affecting others more than being affected), (1) add (S, S') to $\mathbf{R}$, (2) for each $S_0 \in \bar{R}(S)$, if $P(S \rightarrow S_0|\{S \rightarrow S'\}) < .5$, remove S_0 from $\bar{R}(S)$ (pruning); (3) update $\sigma(S_0)$, $S_0 \in \bar{R}(S)$, by ignoring the removed sources.
4. Go to step 3 until $\bar{R}(S) = \emptyset$.

EXAMPLE 5.8. Consider S_4 in the motivating example (Tbl.1). Assume $\Delta(S_4 \rightarrow S_1|S_3) = .8$, $\Delta(S_4 \rightarrow S_2|S_3) = .8$, $\Delta(S_4 \rightarrow S_3|S_1) = .5$, $\Delta(S_4 \rightarrow S_3|S_2) = .5$. $S_4 \rightarrow S_3$ has the highest accumulated influence (1.6) and is less influenced by others (.5), so we add it to $\mathbf{R}$. Since both $S_4 \rightarrow S_1$ and $S_4 \rightarrow S_2$ are significantly influenced by $S_4 \rightarrow S_3$, we can prune them and terminate. $\square$

Appendix C gives the detailed algorithm GLOBALDETECTION and describes opportunities for further approximation.

PROPOSITION 5.9. Let $m = \max_{S \in \mathcal{S}} |\bar{R}(S)|$ and r be the maximum number of sources related to a source in $\mathbf{R}$. The complexity of GLOBALDETECTION is $O(m^2|\mathcal{S}||\mathcal{O}| + r|\mathcal{S}|^2|\mathcal{O}|)$. $\square$

6. EXPERIMENTAL RESULTS

We experimented on two real-world data sets: the weather data (described in Sec. 6.1) and the AbeBooks data (Sec. 6.2). They are very different in the size of the domain, the number of the sources, and the characteristics of the data. We have a partial golden standard for the first but no golden standard for the second; thus, we focus on the correctness of our results on the first and the efficiency

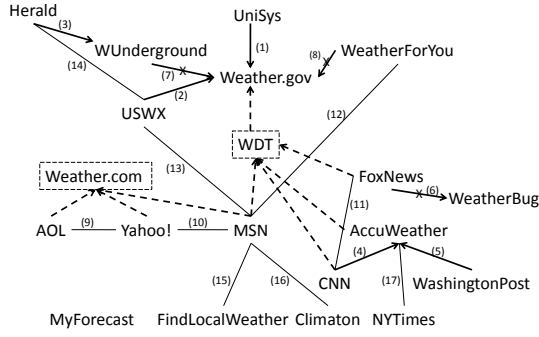


Figure 5: Copying between weather sources. There are 18 sources and 2 hidden sources (in dashed box). A solid arrow represents a copying indicated by the source website (non-crossed ones are “golden” dependencies); a dashed arrow represents a copying associated w. a hidden source; and a thin line represents a “silver” dependence that we derive.

of our algorithm on the second. Together with synthetic data generated from the AbeBooks data (Appendix E), we are able to test various aspects of our models, showing their effectiveness, stability, and efficiency.

We implemented the model in [6] (ACCU), the basic model (BASIC), the correlated-copying model (LOCAL), and global detection (GLOBAL). We set the parameters as described in Appendix B, with initial values $\alpha = .25$ and $k_0 = .8$. We did not assume knowledge of correctness or popularity of values and conducted truth finding and copying detection iteratively (Sec.2). We used Java and experimented on a WindowsXP machine with 2.66GHz Intel CPU and 3.48GB of RAM.

6.1 Results on the weather data

Data: We collected weather data for 30 major USA cities from 18 websites about every 45 minutes. We consider (city, time) as the key. There are in total 33 collections in a day and thus 990 objects. We manually map the attributes and there are 28 distinct attributes. Among them, 10 are provided by at least 10 sources and 11 are provided by only 1 source; on the other hand, a source on average provides 11 attributes, while the max is 15 and the min is 3.

This data set introduces four new challenges. First, there is hardly a true or false notion for weather-related data; thus, we need to consider popularity of provided values and we do so in a similar way as we consider formatting popularity. Second, the weather data are often updated frequently and a copier may not have copied the most up-to-date data at some time of crawling; thus, we need to be able to detect copying even when the copying percentage is not high and so setting proper parameters is critical. Third, most sources have the same object-level completeness and similar completeness for each attribute, and each source has a consistent formatting style for the same attribute (by applying some style sheet); thus, we lack evidence from completeness and formatting for direction decision. Fourth, there are two hidden sources, *WDT*⁷ and *Weather.com*, that are co-copied by sources in our data set, but we cannot crawl them because of commercial or technical reasons.

Golden standard: To find copying between sources, we investigated the websites for explicitly mentioned partnerships, clients, and resources; in addition, we checked source code of the webpages for URLs and citations of other sources. We found 8 copyings between the crawled sources and 8 from the crawled sources to the hidden ones (Fig. 5). Accordingly, we created a semi-golden standard as follows. First, we manually examined every pair of sources

⁷*WDT* collects raw data from *Weather.gov* and applies some aggregation model, then resells the data to online media agencies.

Table 4: Results of various methods on the weather data.

	ACCU	BASIC	LOCAL	GLOBAL
Precision	.5	1	.33	.79
Recall	.43	.14	.86	.79
F-measure	.46	.25	.48	.79

with investigated copying; we removed 3 of them where we observed very small commonality of data with no particular pattern. The remaining 5 are called *golden dependencies*. Second, for each copier of a hidden source, we tried to find its co-copyers with which it shares a large portion of common data; if such co-copyers exist, we added a dependence (with no particular direction) with the one with the largest overlap. We added 5 dependencies in this way and call them *silver dependencies*. Third, for each source where we cannot find any claim from the website, we manually checked if there exist sources that share a large portion of common data; if so we added a dependence (without direction) with the one with the largest overlap. There are 5 such sources and we added 4 dependencies, also called *silver dependencies*. In total, there are 14 copyings (5 golden and 9 silver) in the semi-golden standard and we list the reasons for including them in Appendix D.

Measures: Here we focus on whether our model correctly finds direct copying and defer testing correctness of copying direction to the synthetic data (Appendix E). This is because as mentioned, the weather data lack evidence from formatting for direction detection, which is shown to be critical from results on the synthetic data. Specifically, we measured *precision*, the proportion of identified copying that are real (maybe in the wrong direction), and *recall*, the proportion of real copying that are identified. *F-measure* is computed as $\frac{2PR}{P+R}$, where P is the precision and R is the recall.

Results: GLOBAL obtained both precision and recall as .79. (1) Among the 5 golden dependencies, GLOBAL finds 4 of them, 2 in the correct direction and 1 with uncertain direction. It misses *Herald* $\rightarrow$ *WUnderground*: among the 8 common non-key attributes, they highly overlap only on conditions and visibility, and the shared values are fairly common also among other sources (especially for visibility), so the accumulated positive evidence is slightly weaker than the negative evidence, even with a reasonable parameter setting. (2) Among the 9 silver dependencies, GLOBAL finds 7. It misses the dependence between *Yahoo!* and *MSN*: even though local detection detects it, global detection removes it because the common data are covered by those between *Yahoo!* and *USWX*. This decision is reasonable because *Weather.com*, from which *Yahoo!* copies, and *USWX* might derive data from the same source. GLOBAL also misses the dependence between *Climaton* and *MSN*: it finds copyings from other sources to *Climaton* in local detection but eliminates them in global detection. If *Climaton* is indeed a copier, then the error is caused by wrong direction decisions in local detection. (3) Finally, GLOBAL has three false positives (including *Yahoo!* $\rightarrow$ *USWX*). For each of such pairs there is more or less co-copying and the global detection does not eliminate it because of some additional shared data (maybe shared at some crawlings). To summarize, except the false negative on *Herald* $\rightarrow$ *WUnderground*, errors are mainly caused by wrong direction decisions (because of the *lack-of-evidence* issue) and uneliminated co-copying (because of the *updating* issue).

Tbl.4 compares the results of various methods. First, LOCAL obtains a higher recall (it detects in addition the dependence between *Yahoo!* and *MSN*) but a much lower precision. It returns 38 copyings, among which 38-12=26 are false positives, and GLOBAL removes 26-3=23 (88.5%) of them. Second, BASIC, in contrast, finds only 2 correct copyings. It misses a lot of copyings as it ignores evidence from per-object copying (as illustrated in Ex.3.2 and

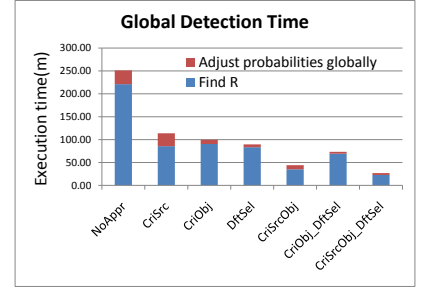
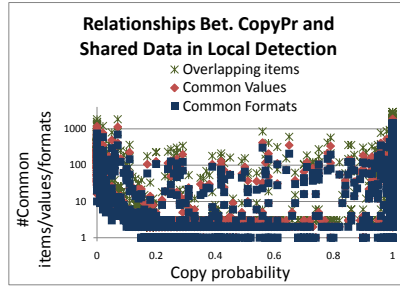
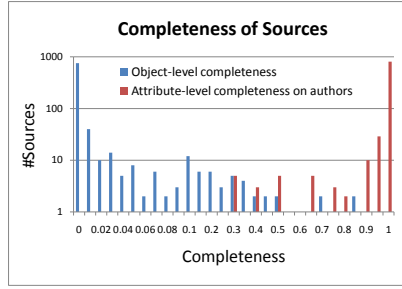


Figure 2: Coverage of AbeBooks sources. 92% of the sources provide less than 5% of the books.

Figure 3: Copy probability vs. shared items, values, formats.

Figure 4: Execution time of various approximations for global detection.

Ex.4.1); in addition, it tends to set s (selectivity) to a high value. Third, ACCU finds 2 golden dependencies (in the correct direction) and 4 silver dependencies. It finds more copyings than BASIC because it considers each shared uncommon value as a wrong value and so accumulates more positive evidence. It has 6 false positives, all co-copyings and transitive copyings. Note, however, that experiments on the synthetic data show improvement of BASIC over ACCU when the true/false notion does apply and there is extra evidence from completeness and formatting of data.

Finally, on average GLOBAL spent 8 seconds for initialization (finding overlapping items, shared values and formats), 2.7 minutes for local detection, and 10 minutes for global detection (5.9 minutes for finding R and 4.1 minutes for globally adjusting probabilities). Thus, the efficiency of our algorithm is acceptable when the number of sources is small.

6.2 Results on AbeBooks data

Data: The AbeBooks data set was extracted in 2007 from *AbeBooks.com* by searching computer-science books. In the data set there are 877 bookstores, 1263 books, and 24364 listings, each containing attributes ISBN (key), name, and often authors. Unlike the weather data set, the true/false notion does apply and there is much higher variety in formatting and completeness (Fig.1-2).

Results LOCAL finds 1553 pairs of sources with copying. Fig.3 plots the copying probabilities for pairs whose Jaccard distance on data items (intersection over union) is at least .1. The plot is fairly semetric from left to right, showing that the numbers of shared items, values, and formats themselves do not decide the copying relationship, and our model considers in addition the popularity. GLOBAL finds only 465 pairs with direct copying, as it eliminates co-copying and transitive copying. Compared with LOCAL, it finds that most sources copy from or are being copied by only a few sources (the max is 17 and 9 for GLOBAL respectively, but 44 and 37 for LOCAL).

On average, GLOBAL took 1.6 minutes for initialization, 3.8 minutes for local detection, and 251.1 minutes for global detection. As this data set contains a lot of sources, global detection becomes the bottleneck; especially, between finding R and globally adjusting probabilities, the former is much more expensive (221.2 min). Fig.4 compares various approximations, including whether to apply Eq.(16) on only “critical” sources (CRISRC), on only “critical” objects (CRIOBJ), and to use s_{obj} instead of $P(S_1 \xrightarrow{O.A} S)$ (DFTSEL) (Appendix C). We observe that CRISRC considers only critical sources (59%) and reduced time in finding R , CRIOBJ considers only critical objects (6%) and reduced time in both steps, and DFTSEL simplifies computation of global probability and also reduced time in both steps. Finally, CRISRCOBJ.DFTSEL took only 26.8 minutes, fairly acceptable given that dependence detection is offline.

7. RELATED WORK AND CONCLUSIONS

This paper studied copying detection between a set of sources. We first improved previous techniques for pairwise detection by proposing a framework where we can plug in different types of copying evidence, and consider correlations between copying. We then described techniques for global detection where we eliminate co-copying and transitive copying. Experimental results show high effectiveness and efficiency of our algorithms. Interesting directions for future work include visualization of copying, and categorization and summarization of the copied instances.

Existing work on copying detection includes detecting copying between texts or programs [12, 3], between videos [11], and between structured data sources [2, 6, 7]. The work most related to ours is [6], with which we have compared in detail in Sec.3 and in experiments. Blanco et al. [2] extended [6] by considering multiple attributes, but assumed a copier must copy *all* attribute values of an object, whereas we consider both per-object (allowing copying only a subset of attribute values) and per-attribute copying.

8. 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*, 2010.
- [3] P. Buneman. The recovery of trees from measures of dissimilarity. *Mathematics the Archeological and Historical Sciences*, pages 387–395, 1971.
- [4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.
- [5] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 2010.
- [6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1), 2009.
- [7] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1), 2009.
- [8] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.
- [9] E. Gansner, Y. Hu, and S. Kobourov. GMap: Drawing graphs and clusters as map. In *IEEE Pacific Visualization Symposium*, 2010.
- [10] N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. In *SIGMOD*, 2006.
- [11] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *CIVR*, pages 371–378, 2007.
- [12] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proc. of SIGMOD*, 2003.
- [13] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.

APPENDIX

A. DETAILS FOR THE BASIC MODEL

We now give details for the basic model. Consider two sources $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$. As we assume *no mutual copying*, there are three possible relationships between them: S_1 copying from S_2 ($S_1 \rightarrow S_2$), S_2 copying from S_1 ($S_2 \rightarrow S_1$), and neither source copying from the other ($S_1 \perp S_2$). We can compute the probability for each case (they sum up to 1) by Bayesian analysis based on our observations of the data, denoted by Φ :

$$P(S_1 \rightarrow S_2 | \Phi) = \frac{\alpha P(\Phi | S_1 \rightarrow S_2)}{\alpha P(\Phi | S_1 \rightarrow S_2) + \alpha P(\Phi | S_2 \rightarrow S_1) + (1 - 2\alpha) P(\Phi | S_1 \perp S_2)} \quad (19)$$

Here, $0 < \alpha < .5$ is the a-priori probability that a source copies from another. Thus, we need to compute the probability of Φ conditioned on different copying relationships.

Observation Φ consists of observations on each data item; i.e., $\Phi = \{\Phi_{O,A} | O \in \mathcal{O}, A \in \mathcal{A}\}$. According to the *item-wise independence* assumption, we have

$$P(\Phi | \text{cond}) = \prod_{O \in \mathcal{O}, A \in \mathcal{A}} P(\Phi_{O,A} | \text{cond}). \quad (20)$$

In local detection, we consider only data provided by S_1 and S_2 ; i.e., $\Phi_{O,A} = \{\Phi_{O,A}(S_1), \Phi_{O,A}(S_2)\}$. Then, we have (similar for the condition $S_2 \rightarrow S_1$)

$$\begin{aligned} P(\Phi_{O,A} | S_1 \perp S_2) \\ &= P(\Phi_{O,A}(S_1) | S_1 \not\rightarrow S_2) P(\Phi_{O,A}(S_2) | S_2 \not\rightarrow S_1); \\ P(\Phi_{O,A} | S_1 \rightarrow S_2) \\ &= P(\Phi_{O,A}(S_1) | S_1 \rightarrow S_2, \Phi_{O,A}(S_2)) P(\Phi_{O,A}(S_2) | S_2 \not\rightarrow S_1). \end{aligned} \quad (21)$$

Thus, the key to detecting copying is to compute $P(\Phi_{O,A}(S_1) | S_1 \not\rightarrow S_2)$ and $P(\Phi_{O,A}(S_1) | S_1 \rightarrow S_2, \Phi_{O,A}(S_2))$, corresponding to $P(\Phi_{O,A}(S_1))$ and $P(\Phi_{O,A}(S_1) | S_1 \rightarrow S_2)$ respectively (similar for S_2). Sec.3 has described their computation according to the data profiling. We now present details in two aspects.

Details for computing $P(\Phi_{O,A}(S_1))$: Here, we need to compute $P(\Phi_{O,A}(S_1) \neq \emptyset)$, $P(\Phi_{O,A}(S_1) \neq \emptyset)$, $P(\text{value}(\Phi_{O,A}(S_1)) = v)$, and $P(\text{format}(\Phi_{O,A}(S_1)) = f)$. Sec. 3 has described how we compute $P(\Phi_{O,A}(S_1) \neq \emptyset)$. Similarly we can compute $P(\Phi_{O,A}(S_1) \neq \emptyset)$ according to attribute-level completeness and $P(\text{format}(\Phi_{O,A}(S_1)) = f)$ according to format patterns.

Now consider the probability of providing a particular value v (i.e., $P(\text{value}(\Phi_{O,A}(S_1)) = v)$). If A is the key, the probability is 1. Otherwise, assume there are m wrong values in the underlying domain. Then, S_1 provides a true value with probability $A(S_1)$ and a particular wrong value with probability $\frac{1-A(S_1)}{m}$ (we assume equal probability of providing a wrong value and relaxation of this assumption is discussed in [6]). Recall that $P(v)$ denotes the probability of value v being true, so

$$P(\text{value}(\Phi_{O,A}(S_1)) = v) = P(v)A(S_1) + (1 - P(v))\frac{1 - A(S_1)}{m}. \quad (23)$$

Distinguishing negative correlation: Recall that our goal is to find copiers rather than negative correlation, we shall omit evidence for the latter. There are two types of such evidence. First, S_1 and S_2 providing the same value in the same format but $P(\Phi_{O,A}(S_1) | S_1 \rightarrow S_2) < P(\Phi_{O,A}(S_1))$ shows that a “dependent” source is less likely to provide the same data and indeed implies negative correlation; we set $P(\Phi_{O,A}(S_1) | S_1 \rightarrow S_2) = P(\Phi_{O,A}(S_1))$ in this case. Second, $C_O(S_1 | \neg S_2) > C_O(S_1)$ (similar for $O.A$) shows that a “dependent” source is more likely to provide an object not provided by the original source and indeed implies negative correlation; we set $C_O(S_1 | \neg S_2) = C_O(S_1)$ in this case.

B. PARAMETER SETTING

The correlated-copying model involves parameters $s_{obj}, s_A^{\rightarrow}, s_A^{\leftarrow}$ and k , which can be different for different pairs of sources. We now describe how we set them for each possible copying relationship $S_1 \rightarrow S_2$ (thus, for each pair of sources, we set the parameters differently for the two different directions).

Initialization: We start with s_{obj} . We first examine overlapping objects; for each attribute, we compute the ratio of common values. Accordingly we generate the histogram for each range of ratio, find the peak range, and use its middle value as the selectivity for overlapping objects, $s_{overlap-obj}$. Then, the overall selectivity is

$$s_{obj} = \frac{s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}{|\bar{O}(S_2)|}. \quad (24)$$

For each $A \in \mathcal{A}$, we denote by $\bar{O.A}_v$ the items on which the two sources provide the same value. Then, we compute $s_A^{\rightarrow}$ and $s_A^{\leftarrow}$ as

$$s_A^{\rightarrow} = \frac{|\bar{O.A}_v|}{s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}; \quad (25)$$

$$s_A^{\leftarrow} = \frac{|\bar{O.A}_v| - s_A^{\rightarrow} \cdot s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}{(1 - s_{overlap-obj}) |\bar{O}(S_1) \cap \bar{O}(S_2)|}. \quad (26)$$

Now consider k and we set it for each attribute. Let $\bar{O.A}_f$ be the items of A on which the two sources provide the same value in the same format. For the copied objects, the format keeping rate is $\frac{|\bar{O.A}_f|}{|\bar{O.A}_v|}$; for the rest of the objects, we use a default rate k_0 . So

$$s_A = s_{overlap-obj} \cdot s_A^{\rightarrow} + (1 - s_{overlap-obj}) s_A^{\leftarrow}; \quad (27)$$

$$k_A = s_A \cdot \frac{|\bar{O.A}_f|}{|\bar{O.A}_v|} + (1 - s_A) \cdot k_0. \quad (28)$$

Finally, note that we want to avoid extreme values for the parameters and so set them only in a certain range. In our experiments we use range $[.1, .9]$ and truncate values outside this range.

Adjustment: According to our copying detection results, we can adjust the parameters and re-do the detection. In particular, if S_1 copies from S_2 (with probability $P(S_1 \rightarrow S_2)$), we shall use the percentage of copied objects or data items (or preserved formatting) observed from the data; otherwise, we shall use the initial settings. Specifically, we adjust the parameters according to the following equations (similar for $s'_{overlap-obj}$ and $s_A^{\leftarrow'}$):

$$s'_{obj} = \frac{\sum_{O \in \bar{O}(S_2)} s(O)}{|\bar{O}(S_2)|} P(S_1 \rightarrow S_2) + s_{obj} (1 - P(S_1 \rightarrow S_2)); \quad (29)$$

$$s_A^{\leftarrow'} = \frac{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} s(O) P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \xrightarrow{O} S_2)}{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} s(O)} \cdot P(S_1 \rightarrow S_2) + s_A^{\rightarrow} (1 - P(S_1 \rightarrow S_2)); \quad (30)$$

$$k'_A = \frac{\sum_{O.A \in \bar{O.A}_f} P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \rightarrow S_2)}{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \rightarrow S_2)} \cdot P(S_1 \rightarrow S_2) + k (1 - P(S_1 \rightarrow S_2)). \quad (31)$$

Here, $P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \rightarrow S_2)$ denotes the probability that a copier copies a data item $O.A$. We compute it in a similar way as we compute $s(O)$:

$$\begin{aligned} P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \rightarrow S_2) &= s(O) P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \xrightarrow{O} S_2) \\ &+ (1 - s(O)) P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \not\rightarrow S_2, S_1 \rightarrow S_2); \end{aligned} \quad (32)$$

$$\begin{aligned} P(S_1 \xrightarrow{O.A} S_2 | \Phi_{O,A}, S_1 \xrightarrow{O} S_2) &= \frac{s_A^{\rightarrow} P(\Phi_{O,A}(S_1) | S_1 \xrightarrow{O.A} S_2)}{P(\Phi_{O,A}(S_1) | S_1 \xrightarrow{O} S_2)} \\ &= \frac{P(\Phi_{O,A}(S_1) | S_1 \xrightarrow{O} S_2) - (1 - s_A^{\rightarrow}) P(\Phi_{O,A}(S_1))}{P(\Phi_{O,A}(S_1) | S_1 \xrightarrow{O} S_2)}. \end{aligned} \quad (33)$$

Algorithm 1: GlobalDetection($\mathcal{S}, \mathcal{O}$)

Input : $\mathcal{S}$ sources, $\mathcal{O}$ objects
Output : $\mathcal{D}$ Copying probabilities between each pair of sources in $\mathcal{S}$
// Pairwise copying detection

```

1  $\mathcal{D}^0 \leftarrow \text{PairwiseDetection}(\mathcal{S}, \mathcal{O});$ 
   // Find  $\mathbf{R}$ 
2  $\mathbf{R} \leftarrow \emptyset;$ 
3 foreach  $S \in \mathcal{S}$  do
4    $\bar{R} \leftarrow \text{FindOriginals}(S, \mathcal{D}^0);$  // Find sources  $S$  copies from
5   foreach  $S_1 \in \bar{R}$  do
6      $\sigma(S_1) \leftarrow 0; \Lambda(S_1) \leftarrow 0;$ 
7   foreach  $\langle S_1, S_2 \rangle \in \bar{R}, S_1 \neq S_2$  do
8      $\Delta(S_1|S_2) \leftarrow P(S \rightarrow S_1) - P(S \rightarrow S_1|\{S \rightarrow S_2\});$ 
9     if  $\Delta(S_1|S_2) < 0$  then
10        $\Delta(S_1|S_2) \leftarrow 0;$ 
11      $\sigma(S_2) \leftarrow \sigma(S_2) + \Delta(S_1|S_2);$ 
12      $\Lambda(S_1) \leftarrow \max(\Lambda(S_1), \Delta(S_1|S_2));$ 
13   while  $\bar{R} \neq \emptyset$  do
14     find  $S'$  with the max  $\sigma(S')$ ;
15      $\bar{R} \leftarrow \bar{R} - \{S'\};$ 
16     if  $\sigma(S') > \Lambda(S')$  then
17        $\mathbf{R} \leftarrow \mathbf{R} \cup \{(S, S')\};$ 
18       foreach  $S_0 \in \bar{R}$  do
19         if  $P(S \rightarrow S_0) - \Delta(S_0|S') < .5$  then
20            $\bar{R} \leftarrow \bar{R} - \{S_0\};$ 
21           foreach  $S_1 \in \bar{R}$  do
22              $\sigma(S_1) \leftarrow \sigma(S_1) - \Delta(S_0|S_1);$ 
23           else
24              $\sigma(S_0) \leftarrow \sigma(S_0) - \Delta(S'|S_0);$ 
25   // Recompute global copying probabilities
26 foreach  $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$  do
27   if  $(S_1, S_2) \in \mathbf{R}$  then
28      $\mathcal{D}(S_1, S_2) = \mathcal{D}^0(S_1, S_2);$ 
29   else
30      $\mathcal{D}(S_1, S_2) = \text{GlobalPr}(S_1, S_2, \mathbf{R});$ 

```

Discussions: Experimental results show that (1) using $s_{\text{overlap-obj}}$ for overlapping objects and s_{obj} for the rest of the objects obtains better results than using s_{obj} everywhere; (2) setting s_A^-, s_A^+ and k differently for different attributes obtains better results; and (3) setting the parameters empirically can significantly improve over setting arbitrary values, but adjusting the parameters does not show much further benefit.

C. DETAILS FOR GLOBAL DETECTION

Algorithm GLOBALDETECTION shows the detailed algorithm for global copying detection. We can further simplify in three ways.

1. When we generate $\mathbf{R}$, we can include in $\bar{R}(S)$ only sources that are likely to cause significant changes or be significantly changed; in particular, those that share a lot of common values with S (not only a high copying probability). In this way, we can reduce m (see Proposition 5.9). (In experiments we consider sources with which S shares 20% of its values.)
2. When we compute $P(S_1 \rightarrow S_2|\mathbf{R})$, we can compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ only on objects that are strong indicators of copying. Specifically, we start with computing $P(\Phi_{O.A}(S_1))$; then, only if $\frac{P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)}{P(\Phi_{O.A}(S_1)|S_1 \neq S_2)} > \tau$ or $\frac{P(\Phi_O(S_1)|S_1 \rightarrow S_2)}{P(\Phi_O(S_1)|S_1 \neq S_2)} > \tau$, we compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ for each attribute of O . Here, τ is a threshold indicating strong evidence for copying and controls the level of approximation (in experiments we set $\tau = 2$). In this way, we usually need to compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$

for significantly less data items, but where we do compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$, we also need to compute $P(\Phi_{O.A}(S_1))$.

3. When we apply Eq.(16), instead of using $P(S_1 \xrightarrow{O.A} S)$, we can use the default selectivity $s_{\text{obj}} s_A^- + (1 - s_{\text{obj}}) s_A^+$. In this way, computing $P(\Phi_{O.A}(S_1)|\mathbf{R})$ is much cheaper.

D. GOLDEN STANDARD ON WEATHER DATA**Table 5:** Generation of the semi-golden standard for the weather data.

	Rel	Reason
G	(1)	UniSys lists www.nws.noaa.gov (a mirror of <i>Weather.gov</i>) as a resource.
O	(2)	USWX links to <i>Weather.gov</i> in source code.
L	(3)	<i>Herald's</i> source code has icons, links from <i>WUnderground</i> .
D	(4)	<i>AccuWeather</i> lists <i>CNN</i> as a client.
E	(5)	<i>AccuWeather</i> lists <i>WashingtonPost</i> as a client.
N	(6)	<i>WeatherBug</i> lists <i>FoxNews</i> as a partner, but shares only 11.4% of the non-key data.
R	(7)	<i>WUnderground</i> lists www.nws.noaa.gov (<i>Weather.gov</i>) but shares only 2 non-key attributes and 16.5% non-key data.
E	(8)	<i>WeatherForYou</i> lists www.nws.noaa.gov (<i>Weather.gov</i>) but shares only 32% non-key data among attributes and objects.
S	(9)	<i>Weather.com</i> lists <i>AOL</i> and <i>Yahoo!</i> as affiliates.
I	(10)	<i>Weather.com</i> lists <i>MSN</i> and <i>Yahoo!</i> as affiliates.
V	(11)	<i>WDT</i> lists <i>CNN</i> and <i>FoxNews</i> as customers.
L	(12)	Potential co-copyers and share a lot of data.
V	(13)	Potential co-copyers and share a lot of data.
E	(14)	No explicit claim from <i>Herald</i> but share a lot of data.
R	(15)	No explicit claim from <i>FindLocalWeather</i> but share a lot of data.
	(16)	No explicit claim from <i>Climatton</i> but share a lot of data.
	(17)	No explicit claim from <i>NYTimes</i> but share a lot of data.

E. EXPERIMENTS ON SYNTHETIC DATA

To test effectiveness of our models, we also experimented on synthetic data. To generate a copier C , we chose an original source S and a copier template T from the AbeBooks data set. Copier C copies from S and independently provides some values or uses some formats, for which we use those provided by T ; essentially C is a copier of both S and T , but we discarded T . We assume C provides data in three steps: (1) among books in $\bar{O}(S) \cap \bar{O}(T)$, C copies p_o percent on all attribute values from S , then for each attribute A , modifies m_A percent of the copied values (per-object copying); (2) for the rest of the books in $\bar{O}(S) \cap \bar{O}(T)$, for p_a percent of non-key attributes C copies all values from S , then modifies m'_A percent of the values for each A (per-attribute copying); (3) for other data items in $\bar{O}(T)$, C provides values on its own while copies c_A percent for each A . For p_f percent of the copied items, C keeps the copied format and for the rest it changes the format. We believe real-world copiers copy in a more or less similar fashion, though may skip some steps or change their order. We randomly decided which data to copy and modify, and generated m_A, m'_A and c_A between $[0, 2m]$ by Gaussian distribution with mean m . By default we set $p_o = .8, p_a = 0, p_f = 1$, and $m = .1$.

We considered three cases: (1) *transitive copying*: five copiers C_1 - C_5 , where C_1 copies from S and C_{i+1} copies from $C_i, i \in [1, 4]$; (2) *co-copying*: five copiers all copying from S ; (3) *multi-copying*: one copier copying sequentially from S_1 - S_5 (i.e., copying from S_i a random subset of $\bar{O}(S_i) \cap \bar{O}(T) - \cup_{j=1}^{i-1} \bar{O}(S_j), i \in [1, 5]$). We pre-selected 10 sources $\bar{S}$ whose object-level completeness ranges from 0.05 to .9 (not necessarily independent). For (1) and (2), for each $S \in \bar{S}$ we ran the experiments 10 times; at each time we randomly chose 5 templates that are judged as independent of S by local detection (but not necessarily independent be-

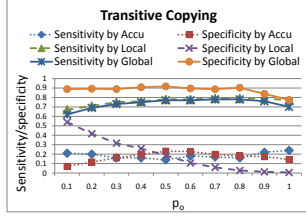


Figure 6: Transitive copying.

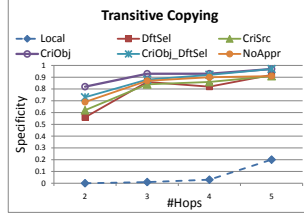


Figure 7: Specificity vs. #hops.

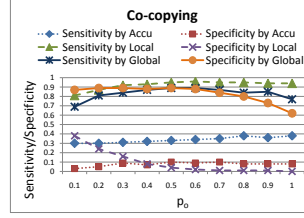


Figure 8: Co-copying.

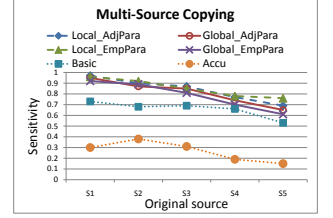


Figure 9: Multi-source copying.

tween themselves) and differ in completeness by at least .05. For (3), we ran the experiment 100 times, each time randomly choosing a sequence of 5 sources from $\bar{S}$ and trying to randomly choose 10 templates that are independent of them and differ in completeness by at least .05; we detected copying for each copier separately. For all cases, we reported the average results.

We set parameters by (1) setting default values (FIXPARA) where $s_{obj} = s = k = .8$, $s_A = .9$, $s_A^* = .4$; (2) setting empirical values (EMPPARA); and (3) first setting empirical values and then adjusting them (ADJPARA) (Appendix B). By default, we applied EMPPARA. We used true values decided from the real-world data, setting $p(v) = 1$ for true values and $p(v) = 0$ for false ones.

To better quantify how we detect transitive and co-copying, we report sensitivity, the proportion of real copying that are identified with the correct direction (a strict version of *recall*), and *specificity*, the proportion of non-copyings that are identified as such.

Transitive copying: We varied p_o from .1 to 1 and examined copying between sources in $\{S, C_1, \dots, C_5\}$ (Fig.6). We observe that while GLOBAL slightly reduces sensitivity (by 3%) compared with LOCAL, it significantly improves specificity (avg .88). Also, GLOBAL obtains fairly stable results when p_o varies: when p_o is very small, the sensitivity is slightly lower because a copier can copy very few data and is not detectable; when p_o is very high, the sensitivity and specificity are slightly lower because a copier may transitively copy a lot of data from its transitive ancestor and even share some additional data (local detection found an average copying probability of .39 between templates), which is indistinguishable from direct copying. Finally, ACCU (hereafter we set $s = p_o(1 - m)$) performs worst; it assumes *item-wise independence* and considers only accuracy, so can often make mistakes about copying direction (it finds all direct copying but only 18% in the correct direction).

Table 6 compares various approximation methods and Fig.7 gives more details on specificity w.r.t. #hops between sources (#hops between C_i and C_j , $i > j$, is $i - j$ and #hops between C_i and S is i ; 1-hop indicates direct copying) when $s_o = .8$. With no surprise, the more hops, the higher specificity; when #hops > 2 , CRIOBJ obtains a specificity of above .9. Among the approximations, CRIOBJ spends half of the execution time as NOAPPR but obtains the best results (as it is not biased by effects on non-critical objects). CRIOBJ_DEFSSEL further cuts the execution time by 70% but still obtains better results than NOAPPR; however, it reduces the specificity by 6% compared with CRIOBJ.

Co-copying: Again, we varied p_o and examined copying between $\{S, C_1, \dots, C_5\}$ (Fig.8). GLOBAL again reduces sensitivity slightly (by 10%) and improves specificity significantly. However, we observe two differences from transitive copying. First, specificity

Table 6: Approximation for detecting transitive copying.

	NOAPPR	CRI SRC	CRI OBJ	DFTSEL	CRI OBJ_DFTSEL
Sensitivity	.75	.75	.77	.7	.78
Specificity	.81	.76	.89	.74	.84
Time(s)	99.3	81.8	50.2	32.4	14.9

is less stable: when $p_o = 1$, the specificity is only .62. When p_o is high, co-copiers can share a large number of copied values and maybe some additional values, so indistinguishable from direct copying. Second, LOCAL obtains higher sensitivity than in the transitive-copying case (avg .94 vs. .76). Actually, we have detected *all* copying in both cases, but did worse in finding the correct direction in case of transitive copying, as there the randomly chosen “original” sources are likely to have high completeness and lead to less precise direction detection.

Multi-copying: We examined copying between the copier and each original source (sensitivity) (Fig.9). We have five observations. (1) When a copier really copies from multiple sources, GLOBAL reduces the sensitivity only very slightly (GLOBAL_ADJPARA by 3%). (2) The sensitivity reduces from S_1 to S_5 , as the copier tends to copy less (if any) data from sources towards the end of the sequence. (3) ADJPARA does not necessarily improve over EMPPARA for LOCAL, but by using it GLOBAL reduces the sensitivity much less (avg 3% vs. 8%) (however, GLOBAL can obtain lower specificity using ADJPARA in case of co-copying and transitive copying; details skipped). (4) BASIC does not consider object copying so obtains much lower sensitivity; (5) ACCU considers only accuracy and uses the same selectivity for all source pairs, so performs the worst.

Finally, we observed similar results for per-attribute copying ($p_o = 0$, $p_a = .5$), and combined copying ($p_o = .5$, $p_a = .5$), and fairly stable results when we varied p_o , p_f , m_a and p_a .

Summary: We summarize our experimental results as follows.

- GLOBAL improves over LOCAL significantly on identifying complex copying relationships.
- Among various approximations for global detection, CRIOBJ can cut the execution time by half or more but still obtain the best results. CRIOBJ_DFTSEL further reduces the execution time without sacrificing the results too much.
- In local detection, BASIC improves over ACCU significantly on copying-direction judgment by considering completeness and formatting in addition to accuracy, using both source-wise and data-item-wise measures, and using $P^c(\Phi_{O.A}(S_1))$ in computation of $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$. LOCAL improves over BASIC further by considering object copying.
- Setting parameters using EMPPARA beats FIXPARA significantly and can obtain quite stable results. ADJPARA does not show further benefits in either local or global detection.

⁸Sometimes this is impossible when there are insufficient number of sources independent of S_1-S_5 .

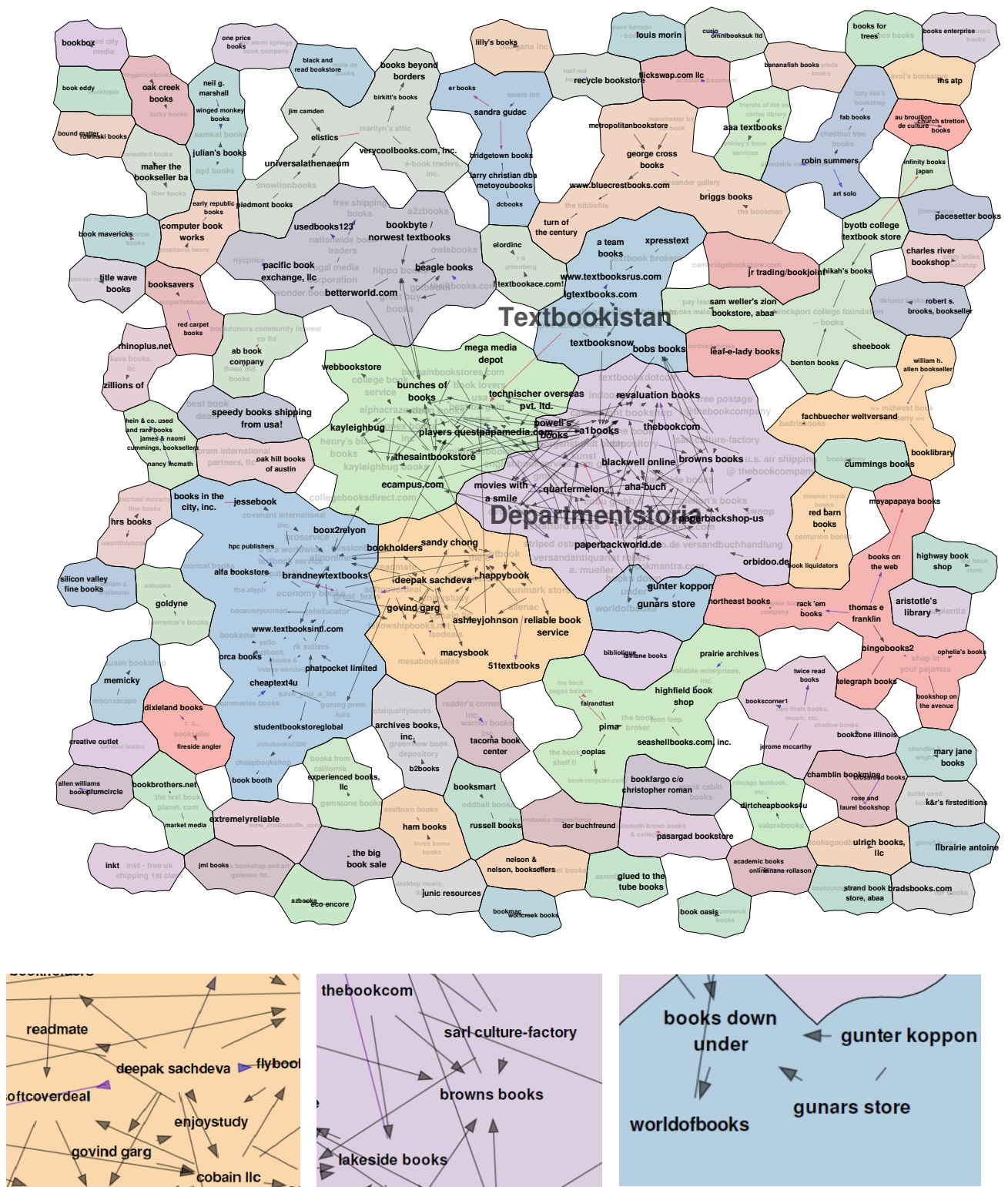


Figure 10: Map of AbeBooks data sources: the copying relationships identified by our copying-detection academic. Each “node” represents a data source and the size of the font corresponds to the number of provided books; to avoid cluttering, we show overlapping ones half transparently. An edge $S_1 \rightarrow S_2$ indicates that S_1 copies from S_2 ; the size of the arrow indicates our confidence of the copying direction; the color indicates the probability of copying (black for 1, blue for .75, and red for .5, and other probabilities are represented by a blend of these colors; e.g., purple for .5-.75.). Each “country” represents a cluster of sources, clustered by modularity clustering (see “Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Physical Review E* 70, 066111 (2004)”) based on their copying relationships.