SOLOMON: Seeking the Truth Via Copying Detection

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

Laure Berti-Equille
Université de Rennes 1
berti@irisa.fr

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

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

ABSTRACT

We live in the Information Era, with access to a huge amount of information from a variety of data sources. However, data sources are of different qualities, often providing conflicting, out-of-date and incomplete data. Data sources can also easily copy, reformat and modify data from other sources, propagating erroneous data. These issues make the identification of high quality information and sources non-trivial.

We demonstrate the SOLOMON system, whose core is a module that detects copying between sources. We demonstrate that we can effectively detect copying relationship between data sources, leverage the results in truth discovery, and provide a user-friendly interface to facilitate users in identifying sources that best suit their information needs.

1. INTRODUCTION

We live in the Information Era: the Web has enabled the availability of a huge amount of useful information and eased sharing of data among sources. Despite the richness of information surrounding us, an information user is often overwhelmed by the huge volume of raw, heterogeneous, and even conflicting data. Data sources can be of different qualities, providing information of different levels of accuracy, freshness, and completeness, and data can flow between data sources, being copied, reformatted, verified, and modified. There is an increasing need to help users find the information and the sources that are of highest quality and authority, to help data producers understand how their data are being used (and possibly protect their rights), and to help analysts and auditors understand how information has been disseminated and how rumors have been propagated.

We are building the SOLOMON system that offers users a useful tool for finding the truths among conflicting values, identifying authoritative sources, and understanding the information flow between sources. One of the major difficulties for truth discovery on a real-world data set is that sources can copy from each other, so errors can easily propagate and lead to wrong conclusions. The core of SOLOMON is a copying-detection module that applies statistical analysis under the assumptions that (1) copying is indicated by an overlap on uncommon values and (2) copying direction is indicated by changes in quality of different parts of data from the same source. With copying detection, SOLOMON is able to achieve two goals. First, it is able to ignore copied data and improve truth-discovery results. Second, based both on the detected copying and the discovered true values, it is able to evaluate the quality of sources more accurately and assist users to identify sources that best suit their needs.

This demonstration illustrates the novel features SOLOMON provides. In particular, our demonstration focuses on two aspects. First, we show how SOLOMON can detect copying between sources, measure quality of sources, apply the results in data fusion (resolving data conflicts) and decide the true values. Second, we show how we provide a user-friendly interface that helps non-technical users identify data sources that suit their needs, and explain to users the various decisions we have made in the data-fusion process.

The rest of the proposal is structured as follows. Section 2 describes the functionality of SOLOMON that we will demonstrate. Section 3 describes the architecture of SOLOMON and some of the technical challenges we face in its construction. Section 4 describes implementation and demonstration. Section 5 describes existing literature that is relevant to SOLOMON and Section 6 concludes.

2. THE SOLOMON SYSTEM

This section describes how a user interacts with SOLOMON and the different features of the system; see [1] for a discussion of the research challenges in the SOLOMON project.

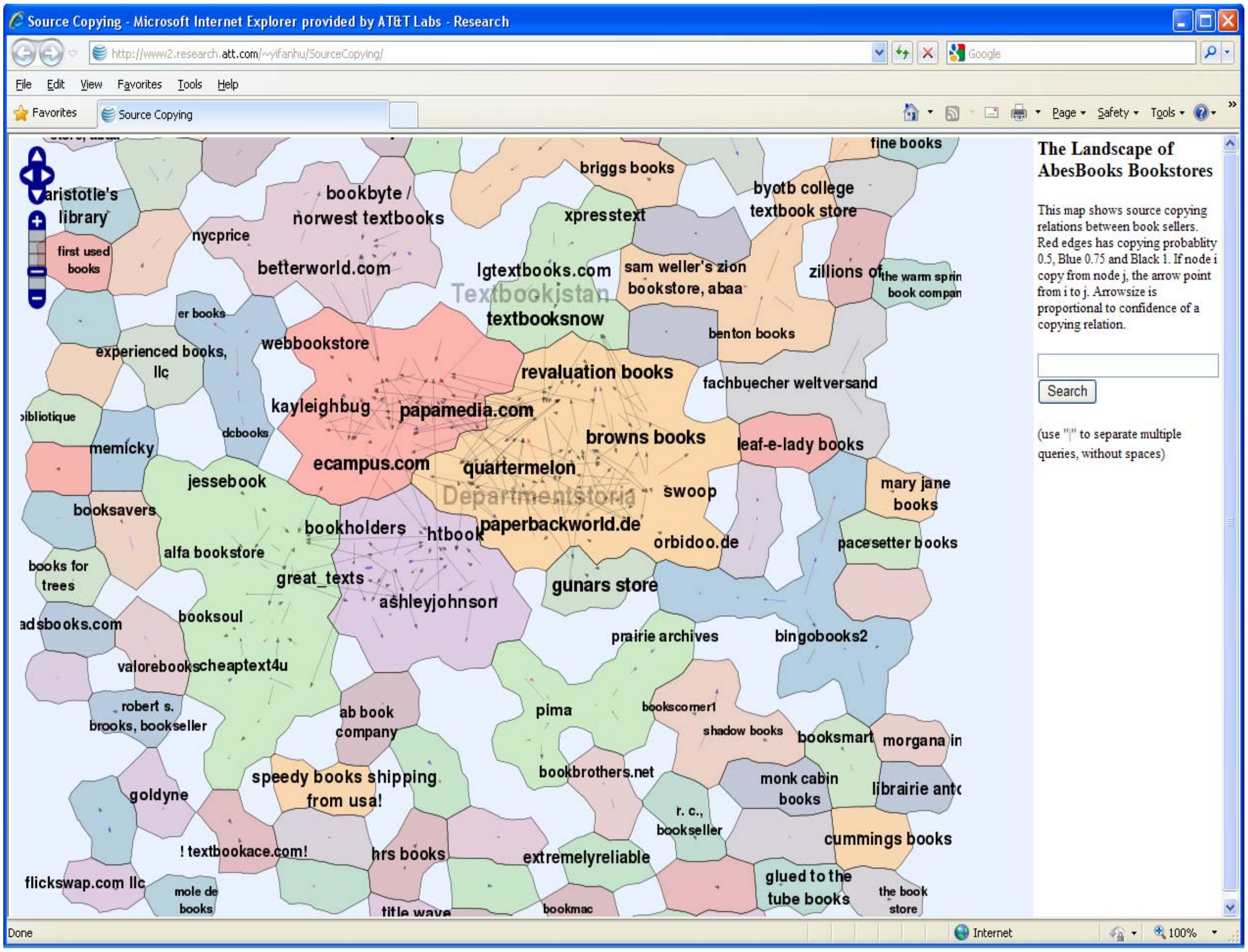
Data storage: SOLOMON considers structured data and assumes a *domain* of *objects*, where each object corresponds to a real-world entity in the domain, and is described by a set of *attributes*. Each *data source* provides a subset of objects in the domain, and provides all or some of the attribute values for each object. A source S is considered as a *copier* of S', if S (*directly*) copies some or all of the values from S'; meanwhile, a copier can provide some values by itself in addition. We assume that mappings between attribute names used by different sources are already obtained using existing schema-matching techniques [11], and linkage between object representations are obtained using existing record-linkage techniques [10].

We next illustrate the usage of SOLOMON using a data set extracted in 2007 from *AbeBooks.com* by searching computer-science books[1]. In the data set there are 877 bookstores (data sources) and 1263 books; each book is described by attributes ISBN, name, and authors.

Global view of data sources: One important functionality pro-

Proceedings of the VLDB Endowment, Vol. 3, No. 2
Copyright 2010 VLDB Endowment 2150-8097/10/09... $ 10.00.

[1] We thank authors of [14] for providing us the data set.



The Landscape of AbesBooks Bookstores

This map shows source copying relations between book sellers. Red edges has copying probablity 0.5, Blue 0.75 and Black 1. If node i copy from node j, the arrow point from i to j. Arrowsize is proportional to confidence of a copying relation.

(use "|" to separate multiple queries, without spaces)

Figure 1: Interface of SOLOMON on the AbeBooks data set.

vided by SOLOMON is the ability to view the copying relationships between data sources as well as the quality of data sources.

SCENARIO 1. *Consider a data analyst who wishes to understand the quality of the data sources that contribute information to AbeBooks. On the starting page of* SOLOMON *there is a map of the data sources, as shown in Figure 1. Each "node" in the map represents a data source and the size of the font corresponds to the number of provided books (overlapping sources are skipped in a coarse-granularity map to avoid cluttering). An edge $S_1 \rightarrow S_2$ indicates that S_1 copies from S_2; the size of the arrow indicates the 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 [4] based on their copying relationships. For example, Figure 1 highlights two countries:* Departmentstoria[2] *includes many big department bookstores, such as* Revaluation Books, Quartermelon.com, *and* paperbackworld.de*;* Textbookistan *includes many textbook stores such as* textbooksNow, LGTextbooks.com, *and* brandnewtextbooks. *The analyst can zoom in to see more details.*

Suppose the analyst wishes to know more about a particular source Powell's Books. *She can click on the node representing* Powell's *or search it by name.* SOLOMON *will give details about the source in two aspects. First, it will show quality of the source, such as its completeness* (e.g., *how many books are provided, for how many books authors are provided,* etc.) *and accuracy (how many provided values are correct). Second, it will show a part of the map centered at* Powell's Books, *containing the sources that* Powell's *copies from, those that copy from* Powell's, *and the relationships between these sources; the analyst can thus understand from it the independence of* Powell's *and the data flow regarding it.*

Suppose the analyst wishes to know more details about the copying relationship between Powell's Books *and* QuarterMelon, *she can click on the corresponding edge in the map or search the pair of sources by name.* SOLOMON *will give details about the copying relationship, including the confidence of the copying relationship and the copying direction, data items that are likely to be copied, and so on.* □

[2] We named the clusters manually.

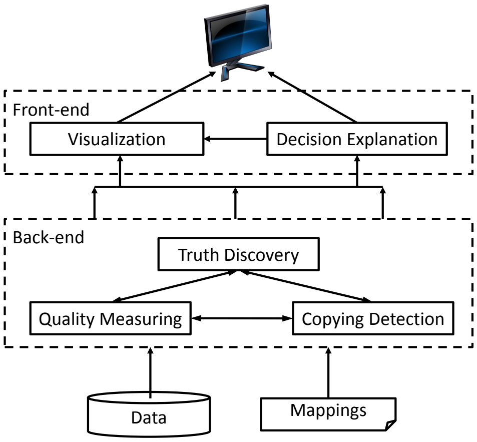


Figure 2: Architecture of the SOLOMON system.

Copying-detection explanation: A user may wonder why SOLOMON draws a particular conclusion on the copying relationship between a pair of sources. SOLOMON provides evidence and explanations of its decisions.

SCENARIO 2. *Consider a data producer who wishes to find data sources that have copied her data. She can search the name of her own source in* SOLOMON *and find the copiers. Suppose among the copiers, she wishes to understand why* SOLOMON *concludes that a particular source has copied her data, she can click on the corresponding edge and require "Explanation".* SOLOMON *will list the positive evidence for copying (*e.g.*, highly overlapping in provided objects, making a lot of common mistakes) and the negative evidence for copying (*e.g.*, providing different values, using different formats,* etc.*), and show how the positive evidence dominates the negative evidence. In addition,* SOLOMON *provides visualization of the evidence to facilitate understanding. When she clicks on a particular item of evidence such as "common mistakes",* SOLOMON *will show more details such as the list of the common errors. The data producer may also wish to find out why* SOLOMON *does not consider another source as a copier of her data and can find the reasoning in a similar way.* □

Truth discovery: SOLOMON in addition decides the true values for each object and explains the decisions to the user.

SCENARIO 3. *Consider a user who wishes to find the authors of the book* XML in data management*. She can do a keyword search and* SOLOMON *will return the book, together with other books that may match the keywords, and show information for each book. She might wonder why* SOLOMON *considers* Allen *in addition to* Aiken *as an author of the book and require "Explanation".* SOLOMON *will then show a graph with several author lists provided by various sources for this book, the quality (in particular, accuracy) of the providers, and the copying relationship between the sources, and explain why "*Aiken and Allen*" is considered as more likely to be the correct author list than others.* □

3. ARCHITECTURE AND TECHNICAL CHALLENGES

Figure 2 depicts the architecture of the SOLOMON system. The back-end of SOLOMON takes the data from various sources and the schema mappings as input, performs data fusion and infers quality measures of sources, copying relationships between sources, and true values for each attribute of each object. It contains three components: *Copying detection, Truth discovery*, and *Quality measuring*. The front-end provides a search and browsing interface to the user, generating visualizations and explanations on users' demand. It contains two components: *Decision explanation* and *Visualization*. We next describe each of the components in more detail.

Copying detection: Copying detection is the core of SOLOMON and it proceeds in two steps [6, 7]. The first step, *local detection*, discovers copying for each pair of sources in isolation of other sources. The key intuition is that if the probability of a source S_1 providing the observed data conditioned on it being independent is much lower than that conditioned on it being a copier of S_2, S_1 is more likely to be a copier of S_2. According to this intuition, we apply Bayesian analysis, where we consider providing the same objects, sharing common values, and using the same formatting as evidence for copying, especially if the objects, the values, and the formats are rarely provided by other sources (*e.g.*, a particular wrong value is typically rarely provided by others). We assume no *mutual* copying (*i.e.*, S_1 copies from S_2 and S_2 copies from S_1) and copying direction is indicated by changes in quality of different parts of data from the same source. The second step, *global detection*, identifies co-copying (multiple sources copying from the same source) and transitive copying (a source copying from a second source, which in turn is copying from a third one), and distinguishes them from direct copying.

Truth discovery: Another key task that SOLOMON performs is to decide the true values. A naive way of doing so is to take the value that the majority of sources vote for. SOLOMON advances this naive method in two ways. First, it considers the copying relationship and ignores a vote if the provider copies the value from another source. Second, it considers the quality of the sources and gives higher weight to votes from sources of higher accuracy. Based on these two intuitions, SOLOMON applies Bayesian analysis and decides the probability of each observed value being true, considering the one with the highest probability as the true value [7].

Quality measuring: SOLOMON measures quality of sources using two orthogonal measures: *completeness* and *accuracy*. The former measures the percentage of objects in the domain that are provided and the percentage of values of a particular attribute that are provided. The latter measures the correctness of the provided values; intuitively, a source that provides more true values is more accurate, we thus compute the accuracy of a source as the average probability of its provided values being true [7].

Note that there is an inter-dependence between results of copying detection, quality measuring, and truth discovery. Thus, each component at the back-end takes the results of the other two as input and they perform their tasks iteratively till the results converge. The whole process can take up to hours [6], but it is acceptable as this is an offline process.

Decision explanation: To answer questions such as "why S is considered as a copier of S'" and "why V is considered as the correct value", SOLOMON provides explanation of various decisions. Such explanations are generated at run-time according to the user's requirement, and interpret the underlying Bayesian analysis in a way that non-technical users can understand.

Visualization: One important goal of SOLOMON is to provide an effective visualization to assist understanding of source quality and copying relationship. To provide a high-level picture to the users, SOLOMON applies the GMap [9] techniques, clustering data sources on their copying relationships and showing the sources in a map where closely related (by copying) sources are put close to each other. The visualization component also takes explanations generated by the *Decision explanation* component and generates visualization for evidence of copying and of truths.

Finally, we highlight the technical challenges we addressed in the SOLOMON system.

1. The first and utmost challenge is to detect copying between sources; it is especially difficult for structured sources as we lack clues from usage of words and sentences. This demo demonstrates how we apply statistical analysis for copying detection, identify the direction of copying, and distinguish direct coping, co-copying, and transitive copying.
2. The second challenge is to discover the true values in presence of sources of various qualities and copying of data without proper attributions. This demo demonstrates how we leverage the knowledge of source quality and dependence in solving this problem.
3. A user often wonders not only "what" but also "why". Our truth discovery and copying detection both conduct Bayesian analysis and consider evidence of various kinds; explaining the decisions to non-technical users can thus be very hard. This demo demonstrates how we show evidence in an aggregated fashion and provide visualizations to assist understanding.
4. Finally, when there are a large number of data sources in a domain, helping users understand the quality of and the relationships between the sources without overwhelming them can be a big challenge. "A picture is worth a thousand words". This demo demonstrates how we can provide effective visualizations for relationships and for multi-dimensional quality measures.

4. IMPLEMENTATION AND DEMONSTRATION

We implemented the back-end of SOLOMON in Java and the UI in JavaScript, and generated the map using the GMap tool [9]. We demonstrate typical use cases that SOLOMON supports; in particular, we present the three scenarios described in Section 2. We show how end users can benefit from SOLOMON through three main processes: discovering quality of and relationships between data sources, understanding the copying-detection decisions, and finding true values. Our demonstration uses the AbeBooks data set and the audience can test the usability of the interface, the credibility of the generated explanations, and the correctness of the discovered truths.

5. RELATED WORK

We are not aware of any existing system on detecting copying between sources and applying such knowledge in data fusion. *FuSem* [2] has demonstrated various data-fusion functions and [8] has surveyed existing strategies; SOLOMON is different in that it studies how to leverage the detected copying relationships to improve truth discovery and perform data fusion in a principled fashion. Winnowing [12] detects plagiarism of programs by comparing their fingerprints (k-grams), but it focuses on unstructured data rather than databases. Recently, there has been a lot of works studying how to manage *provenance* and *lineage* of data [3, 5, 13]. They all assume that the provenance or lineage information has already been provided by users or applications, and focus on how to effectively represent and retrieve such information.

6. CONCLUSIONS

The Web has accelerated the rate at which useful information is produced and disseminated, but has also eased the ability to spread false information. Whereas previous work on managing data from multiple sources focused on resolving heterogeneity of the data (including heterogeneity of the schema and of the representation of values), they often assumed consistency of the values and independence of sources. SOLOMON makes a first step towards detecting copying between sources and managing conflicting and false information. We demonstrate that we can effectively detect copying between data sources, leverage the results in truth discovery, and provide a user-friendly interface to facilitate users in identifying sources that best suit their needs.

7. REFERENCES

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

[2] J. Bleiholder, K. Draba, and F. Naumann. FuSem-exploring different semantics of data fusion. In *VLDB*, pages 1350–1353, 2007.

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

[4] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(066111), 2004.

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

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

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

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

[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] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[12] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proc. of SIGMOD*, 2003.

[13] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proc. of CIDR*, 2005.

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