Online Collective Entity Resolution

Indrajit Bhattacharya
IBM India Research Lab
New Delhi, India
indrajitb@gmail.com

Lise Getoor
Computer Science Dept.
University of Maryland, College Park
getoor@cs.umd.edu

Abstract

Entity resolution is a critical component of data integration where the goal is to reconcile database references corresponding to the same real-world entities. Given the abundance of publicly available databases that have unresolved entities, we motivate the problem of quick and accurate resolution for answering queries over such 'unclean' databases. Since collective entity resolution approaches — where related references are resolved jointly — have been shown to be more accurate than independent attribute-based resolution, we focus on adapting collective resolution for answering queries. We propose a two-stage collective resolution strategy for processing queries. We then show how it can be performed on-the-fly by adaptively extracting and resolving those database references that are the most helpful for resolving the query. We validate our approach on two large real-world publication databases where we show the usefulness of collective resolution and at the same time demonstrate the need for adaptive strategies for query processing. We then show how the same queries can be answered in real time using our adaptive approach while preserving the gains of collective resolution. This work extends work presented in (Bhattacharya, Licamele, & Getoor 2006).

Introduction

Entity resolution is a practical problem that comes up in in a variety of information processing scenarios. First, it can be viewed as a data cleaning problem, the 'deduplication' problem, where the goal is to identify and consolidate pairs of records or references within the same information sources that are duplicates of each other. The other manifestation of entity resolution is the data integration problem, the 'fuzzy match' problem, where tuples from two heterogeneous databases with different keys and possibly different schemas, need to be matched and consolidated. Recently, research on entity resolution has focused on a new aspect of the problem which makes use of additional relational information between database references to improve resolution accuracy; an incomplete list includes (Bhattacharya & Getoor 2004; Singla & Domingos 2004; Pasula *et al.* 2003; Li, Morie, & Roth 2005; Culotta & McCallum 2005). This performance improvement is made possible by resolving related references jointly, rather than independently.

Intuitively, figuring out that two references map to the same underlying entity may in turn give us useful information to link, and possibly resolve, other references; we refer to this as *collective entity resolution*. While it has been shown that collective resolution significantly improves entity resolution accuracy, the added improvement comes at a considerable computation cost.

The computational expense of entity resolution, both collective and otherwise, being non-trivial, it has traditionally been viewed as an off-line process that is not to be undertaken very frequently. So a database is cleaned periodically after every so many updates. Little work has been done on efficiently performing such *incremental* resolution using collective entity resolution approaches. Furthermore, entity resolution decisions are based on currently available evidence, and as updates are made, resolution decisions for existing references may need to be modified as well. This is particularly true if all the resolutions have not been manually verified by a curator, which is often the case in large databases.

Motivated by the difficulties of maintaining an entity resolved database, we address the problem of answering entity resolved queries over an unresolved, or partially resolved, database. Specifically, we consider selection queries that request information about a small portion of the references and propose techniques that perform collective resolution at query-time.

Our approach can be seen as a form of "query-time" or interactive data cleaning. All data cleaning, including entity resolution, is a semi-automatic task at best. While an automated technique can suggest resolutions, for many applications they must still be manually verified by an analyst or data curator. Our approach can provide a powerful tool to help a user understand and resolve data. A user may issue a query for the data of interest. Using a standard query processor, the query answer contains only the dirty, unresolved data. But in addition, we provide a technique to return the resolved answer to the query. Hence, the user may view both the resolved and unresolved query results, and use this information to better understand, and act upon, the available information.

The key challenge of being to able to answer entity resolution queries in real time is the computational aspect. First, collective entity resolution is computationally expensive, and secondly, the set of references that influence a query may be huge. In this research, we present adaptive resource-constrained algorithms for extracting the relevant references for a query that enables us to answer entity resolution queries in real time, while preserving the gains of collective resolution.

Problem Formulation

Formally, in the entity resolution problem, we have a collection of references, $\mathcal{R} = \{r_i\}$, with attributes $\{\mathcal{R}.A_1, \ldots, \mathcal{R}.A_k\}$. Let $\mathcal{E} = \{e_j\}$ be the unobserved domain entities. For any particular reference r_i, we denote the entity to which it maps as $E(r_i)$. In the case of an unresolved database, this mapping $E(\mathcal{R})$ is *not provided*. Further, the domain entities $\mathcal{E}$ and even the number of such entities is not known. However, in many domains, we may have additional information about relationships between the references. To model relationships in a generic way, we use a set of hyper-edges $\mathcal{H} = \{h_i\}$. Each hyper-edge connects multiple references. To capture this, we associate a set of references $\mathcal{H}.R$ with each hyper-edge.

As an example, consider a database of academic publications similar to DBLP, CiteSeer or PubMed. Each publication in the database has a set of author names. For every author name, we have a reference r_i in $\mathcal{R}$. For any reference r_i, $r_i.Name$ records the observed name of the author in the publication. All the author references in any publication are connected to each other by a co-author relationship. This can be represented using a hyper-edge $h_i \in \mathcal{H}$ for each publication and by having $r_j \in h_i.R$ for each reference r_j in the publication. Given this representation, the **entity resolution task** is defined as the partitioning or clustering of the references according to the underlying entity-reference mapping $E(\mathcal{R})$. Two references r_i and r_j should be assigned to the same cluster if and only if they are coreferent, i.e., $E(r_i) = E(r_j)$.

We focus on the use of such co-occurrence relationships between references for collective entity resolution, in which the entities for related references are determined jointly. We explore different techniques for solving the collective entity resolution problem. We have designed a relational clustering algorithm, where references are iteratively clustered into entities taking into account the clusters of co-occurring references (Bhattacharya & Getoor 2004; 2007). We show that this approach locally minimizes a cut-based clustering cost that considers the co-occurrence relations in addition to the similarity between references (Bhattacharya 2006). In addition, we have proposed a probabilistic generative model for co-occurring references that uses Latent Dirichlet Allocation to find hidden group structures among the domain entities as evidence for resolving entities (Bhattacharya & Getoor 2006). We have developed an efficient unsupervised inference algorithm for this model using Gibbs Sampling techniques.

Entity Resolution Queries

In spite of the widespread research interest and the practical nature of the problem, many publicly accessible databases remain unresolved, or partially resolved, at best. The popular publication databases, CiteSeer and PubMed, are representative examples. CiteSeer contains several records for the same paper or author, while author names in PubMed are not resolved at all. Yet, millions of users access and query such databases everyday, mostly seeking information that, implicitly or explicitly, requires knowledge of the resolved entities. For example, we may query the CiteSeer database of computer science publications looking for books by 'S Russell' (Pasula *et al.* 2003). This query would be easy to answer if all author names in CiteSeer were correctly mapped to their entities. But, unfortunately, going by CiteSeer records, Stuart Russell and Peter Norvig have written more than 100 different books together. Additionally, it is not sufficient to simply return records that match the query name 'S. Russell' exactly. In order to retrieve all the references correctly, we may need to retrieve records with similar names as well. But importantly, for the results to be useful, we need to partition the records that are returned according to their entities to which they correspond.

Formally, any query to a database of references is called an **entity resolution query** if answering it requires knowledge of the underlying entity mapping $E(\mathcal{R})$. We consider two different types of entity resolution queries. Most commonly, queries are specified using a particular value a for an attribute $\mathcal{R}.A$ of the references that serves as a 'quasi-identifier' for the underlying entities. Then the answer to the query $Q(A, a)$ should partition all references that have $r.A = a$ according to their underlying entities. For references to people, the name often serves as a weak or noisy identifier. For our example bibliographic domain, we consider queries specified using $\mathcal{R}.Name$. To retrieve all papers written by some person named 'S. Russell', we issue a query using $\mathcal{R}.Name$ and 'S. Russell'. Since names are ambiguous, treating them as identifiers may lead to undesirable results. Additionally, the answer should include any reference to 'Stuart Russell' that maps to the same person.

One of the main reasons behind databases having unresolved entities is that entity resolution is generally perceived as an expensive process for large databases. Also, maintaining a 'clean' database requires significant effort to keep pace with incoming records. We have proposed an alternative solution where we obviate the need for maintaining resolved entities in a database. Instead, we investigate entity resolution at query-time, where the goal is to enable users to query an unresolved or partially resolved database and resolve the *relevant* entities on the fly. A user may access several databases everyday and he does not want to resolve all entities in every database that he queries. He only needs to resolve those entities that are relevant for a particular query. For instance, when looking for all books by 'Stuart Russell' in CiteSeer, it is not useful to resolve all of the authors in CiteSeer. Additionally, the resolution needs to be quick, even if it is not entirely accurate.

While it has been shown that collective resolution significantly improves entity resolution accuracy, its application for query-time entity resolution is not straight-forward. The first difficulty is that collective resolution works for a database as a whole and not for a specific query. Secondly, the accuracy improvement comes at a considerable computation cost arising from the dependencies between related resolutions. This added computational expense makes its application in query-time resolution challenging.

Collective Resolution at Query-time

We have investigated the application of collective resolution for queries (Bhattacharya & Getoor 2007). We propose a recursive 'expand and resolve' strategy for processing queries. The relevant records necessary for answering the query are extracted by a recursive expansion process and then collective resolution is performed only on the extracted records. We use two expansion operators for extracting the relevant records. We alternate between **attribute expansion**, or A-expansion, that takes an attribute value and includes all database references having the same attribute value, and **hyper-edge expansion**, or H-expansion, that takes a reference and includes all references sharing a hyper-edge with it. The expansion process can be terminated at reasonably small depths for accurately answering any query; the returns fall off exponentially as neighbors that are further away are considered.

However, the problem is that this unconstrained expansion process can return too many records even at small depths; and thus the query may still be impossible to resolve in real time. We address this issue using an adaptive strategy that only considers the most informative of the related records for answering any query. Our strategy is based on estimating the **ambiguity** of individual attribute values. The ambiguity of an attribute value is defined as the probability that it is shared by references corresponding to *different* entities. References having ambiguous attributes are not informative without further evidence. For example, 'Peter Norvig' is more informative as a collaborator name than 'S Johnson'. We estimate the ambiguity of one attribute value by using a second attribute. For example, the ambiguity of a last name may be estimated by counting the number of different first names that go with it for all references in the dataset. Given the ambiguity estimates, we modify the H-expansion operator so that it includes only the least ambiguous of the references connected by hyper-edges. A-expansion is similarly modified so that it expands only the most ambiguous of the references currently in the relevant set. This significantly reduces the number of records that need to be investigated at query time, but, most importantly, does not compromise on the resolution accuracy for the query.

Experimental Results

For experimental evaluation of our query-time resolution strategies, we used two real-world citation datasets. **arXiv** contains papers from high energy physics and was used in KDD Cup 2003[1]. Our second dataset is the **Elsevier BioBase** database[2] of publications from biology. For entity resolution queries in arXiv, we selected the 75 ambiguous names that correspond to more than one author entity. For BioBase, we selected as queries the top 100 author names with the highest number of references.

We evaluate several algorithms for entity resolution queries. We compare entity resolution accuracy of the pair-wise co-reference decisions using the F1 measure. For the algorithms, we compare *attribute-based entity resolution* (**A**), *naïve relational entity resolution* (**NR**) that uses *attributes* of related references, and our *relational clustering algorithm for collective entity resolution* (**RC-ER**) using unconstrained expansion up to depth 3. We also consider transitive closures over the pair-wise decisions for the first two approaches (**A*** and **NR***).

Table 1: Entity resolution accuracy (F1) for different algorithms over 75 arXiv queries and 100 BioBase queries

	arXiv	**BioBase**
A	0.721	0.701
A*	0.778	0.687
NR	0.956	0.710
NR*	0.952	0.753
RC-ER Depth-1	0.964	0.813
RC-ER Depth-3	0.970	0.820

Table 1 shows that **RC-ER** improves accuracy significantly over the baselines. This demonstrates the potential benefits of collective resolution for answering queries. Most of the accuracy improvement comes from the depth-1 relevant references, but there is significant improvement beyond depth-1 for queries with high ambiguity.[3] On one hand, this shows that considering related records and resolving them collectively leads to significant improvement in accuracy. On the other hand, it also demonstrates that while there are potential benefits to considering higher order neighbors, they fall off quickly beyond depth 1.

Next, we focus on the time that is required to process these queries in the two datasets using unconstrained expansion up to depth 3. For arXiv, the average processing time of 1.6 secs, showing the effectiveness of our two-phase strategy with unconstrained expansion. However, the time taken for BioBase is more than 10 minutes, which is unacceptable for answering queries.

Finally, we focus on BioBase for evaluating our adaptive strategies. For each of the 100 queries, we construct the relevant set up to depth 3 using adaptive H-expansion and adaptive exact A-expansion. Since most of the improvement from collective resolution comes from depth-1 references, we consider two different experiments. In the first experiment (**AX-2**), we use adaptive expansion only at depths 2 and beyond, and unconstrained H-expansion at depth 1. In the second experiment (**AX-1**), we use adaptive H-expansion even at depth 1.

In Table 2, we compare the two adaptive schemes against unconstrained expansion with $d^* = 3$ over all

[1] http://www.cs.cornell.edu/projects/kddcup/index.html

[2] http://help.sciencedirect.com/robo/projects/sdhelp/about_biobase.htm

[3] These numbers are averages over the entire dataset; improvement is as large as 5 - 27% from depth-1 to depth-3 for queries with high ambiguity.

Table 2: Comparison between unconstrained and adaptive expansion for BioBase

	Unconstrained	AX-2	AX-1
relevant-set size	44,129.5	5,510.52	3,743.52
time (cpu secs)	606.98	43.44	31.28
accuracy (F1)	0.821	0.818	0.820

queries. Clearly, accuracy remains almost unaffected for both schemes. First, we note that **AX-2** matches the accuracy of unconstrained expansion, and shows almost the same improvement over depth 1. This accuracy is achieved even though it uses adaptive expansion that expands a small fraction of $Rel^1(Q)$, and thereby reduces the average size of the relevant set from 44,000 to 5,500. More significantly, **AX-1** also matches this improvement even without including many of the depth-1 references. This reduction in the size of the relevant set has an immense impact on the query processing time. The average processing time drops from more than 600 secs for unconstrained expansion to 43 secs for **AX-2**, and further to just 31 secs for **AX-1**, thus making it possible to use the collective approach for query-time entity resolution.

Entity Resolution in the Larger AI Context

The entity resolution problem has assumed critical importance in recent years. As more and more data becomes available from various sources and in various forms, it is imperative for all of this data to be seamlessly integrated before it can be effectively retrieved and used by different applications. As we have seen, entity resolution forms a core component in data integration. It also forms a critical aspect of information extraction. Automated knowledge extraction from web and other digital sources has been the focus of a lot of recent research. As entities are extracted, it is essential to be able to map them to already existing entities. Another area that involves entity resolution and has seen a flurry of recent research is personal information management, where the diverse sources of information available on one's desktop computer needs to be integrated and organized for easier tracking and retrieval. The entity resolution problem has been studied in different guises in AI. It comes up as name coreference resolution and named entity recognition in natural language processing, and as the correspondence problem and activity recognition in computer vision. It is also relevant for ontology integration on the semantic web, where various local ontologies need to be matched and reconciled for global usability.

In all of these application areas, it is possible to leverage available relationships for making collective decisions. For example, in ontology integration, two nodes in two different ontologies are more likely to correspond if some of their parent or children nodes also match. In named entity recognition, "Buffalo" can be more readily recognized as a place rather than an animal if it occurs in the context of an organization such as a university. For most of these situations, where information is likely to be retrieved based on user queries, our online strategies become relevant as well.

Conclusions

In this paper, we have motivated the problem of query-time entity resolution for accessing unresolved third-party databases. The biggest issue in query-time resolution of entities is reducing the computational expense of collective resolution while maintaining its benefits in terms of resolution accuracy. We propose an adaptive strategy for extracting the set of most relevant references for collectively resolving a query. We demonstrate that this adaptive strategy preserves the accuracy of unconstrained expansion while dramatically reducing the number of relevant references, thereby enabling query-time collective resolution. While we have presented results for bibliographic data, the techniques are applicable in other relational domains.

Acknowledgments: This work was supported by the National Science Foundation, NSF #0423845, with additional support from the ITIC KDD program.

References

Bhattacharya, I., and Getoor, L. 2004. Iterative record linkage for cleaning and integration. In *Workshop on Data Mining and Knowledge Discovery (DMKD)*.

Bhattacharya, I., and Getoor, L. 2006. A latent dirichlet model for unsupervised entity resolution. In *The SIAM International Conference on Data Mining (SIAM-SDM)*.

Bhattacharya, I., and Getoor, L. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data* 1(1).

Bhattacharya, I.; Licamele, L.; and Getoor, L. 2006. Query-time entity resolution. In *The ACM International Conference on Knowledge Discovery and Data Mining*.

Bhattacharya, I. 2006. *Collective Entity Resolution In Relational Data*. Ph.D. Dissertation, University of Maryland, College Park.

Culotta, A., and McCallum, A. 2005. Joint deduplication of multiple record types in relational data. In *Conference on Information and Knowledge Management*.

Li, X.; Morie, P.; and Roth, D. 2005. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*.

Pasula, H.; Marthi, B.; Milch, B.; Russell, S.; and Shpitser, I. 2003. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems*.

Singla, P., and Domingos, P. 2004. Multi-relational record linkage. In *The SIGKDD Workshop on Multi-Relational Data Mining (MRDM)*.