Entity Identification in XML Documents

Leonardo Ribeiro, Theo Härder

University of Kaiserslautern
{ ribeiro | haerder}@informatik.uni-kl.de

Abstract. Abstract: As a natural result of the dissemination of a large variety of XML databases, the well-known problem of data integration must be faced from the XML viewpoint. One of the basic functions of an integration system is the record linkage, the task of comparing records to determine those that are differently represented, but relate to the same entity. As a consequence of the intrinsically high computation cost, the majority of the approaches to record linkage are based on off-line procedures. Such approaches, however, just meet the requirements of data integration architectures that materialize the data such as data warehouses. Recent approaches based on approximate joins are aimed at enabling duplicate identification in on-line procedures with reasonable results. In this paper, we proceed along this research direction and outline our current ideas how to account for the specific characteristics of XML documents.

1 Introduction

The suitability of XML to represent a rich variety of data sources makes it a reasonable choice to deal with the so-called information explosion coming along with the surge of the Internet. Indeed, recent work of the database research community has signaled a long-term orientation towards XML database support. As natural consequence of the dissemination of XML databases, the well-known problem of data integration must be faced from the XML viewpoint. Similarly to other aspects of data management, the properties of the XML data model require careful consideration when combining various data sources.

One of the basic functions of an integration system is record linkage, the task of comparing records to determine those that are differently represented but relate to the same entity. The problem was originally examined in the seminal paper of Newcombe et. al. [1], in the context of statistical "follow-up" in demographic records. Ten years later, Newcombe's key insights were formalized in a theoretical framework using a probabilistic model developed by Fellegi and Sunter [2]. These works still provide the main foundation that guides many modern approaches to record linkage coming under several variant terms, among others, name matching, object identification, merge-purge, reference reconciliation, and fuzzy matching.

The most common application of record linkage techniques is discovering duplicates within a unique database during data cleaning procedures or identifying overlapping entities across multiple databases during the integration. In single-source settings duplicates can be caused by several type of errors during data creation, such as typographical errors or different representations of the same value. The same factors that cause the occurrence of duplicates plague the task of identifying them, because they make the underlying data unreliable for a straight equality comparison using record fields. In multiple-source settings, the problem is still worse. Besides the problems occurring in each single source, objects may be represented differently, overlap, or contradict across source collections [3]. Just ignoring the presence of duplicates is not a solution in many cases. It can erroneously inflate estimates and produce spurious quantitative data in different categories. Moreover, in multiple-source scenarios an important functionality of integration systems is at the bottom of the capability to identify overlapping objects. For example, distributed healthcare networks must identify and merge different records to be able to build a patient's medical history.

As a consequence of the intrinsically high computation cost, the majority of the approaches to record linkage are based on off-line procedures. Such approaches, however, do not meet the requirements of data integration architectures that do not materialize the data such as data warehouses. Systems based based on virtual data integration, for example, detection and elimination or reconciliation of duplicates

from answers must be done "on the fly", before presenting the result of the user query.

Recent approaches based on approximate joins are aimed at enabling duplicate identification in on-line procedures with reasonable results [4]. The big challenge addressed by these approaches is to accommodate record linkage techniques in an efficient join framework. Yet, XML data sources pose due to the structure of XML documents additional complexities that fundamentally differ from that of the relational model which the existing record linkages methods were developed for. Hence, new techniques must be devised to directly deal with the hierarchical semi-structured nature of XML.

Along the rest of this paper we will concentrate on the problem of performing on-line record linkage among XML data sources. More specifically, we consider the identification of entities between a couple of XML documents using approximate joins. The rest of the paper is organized as follows. In Section 2, we sketch a use case scenario sharpening the focus of our work, whereas we deal in Section 3 with the selection of comparison fields. In Section 4, we make some considerations about similarity metrics, a crucial component of an entitity identification task. In Section 5, we discuss some approaches to perform entity identification in XML documents through approximate joins, before we finally wrap up with a summary.

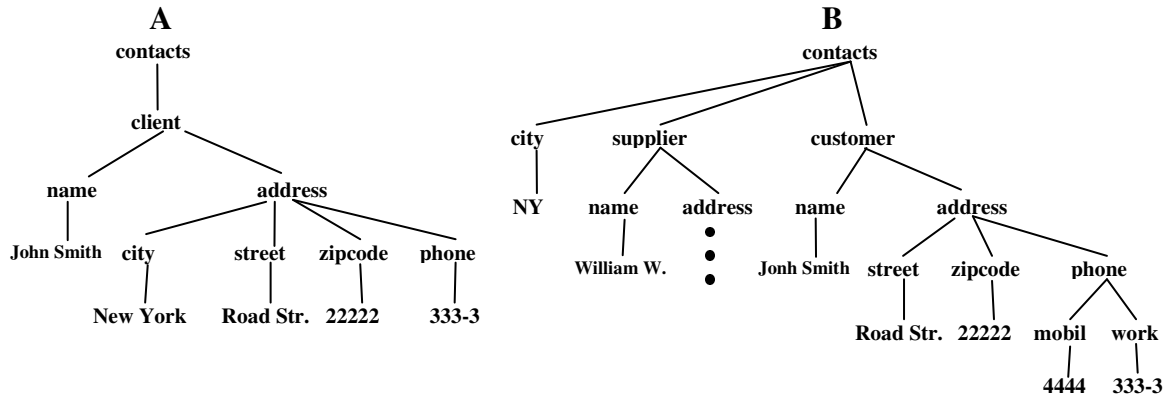

Figure 1. Two XML fragments related to business

2 Motivating Example

Consider the example in Figure 1 showing two XML document fragments from data source A and B. Both sources describe business contacts but are organized in different formats. Source A arranges contacts according to their type (*client* or *supplier*) . In contrast, source B sorts contacts by the city element. The ground work of an integration application is bringing together document fragments that identify objects of interest. Note that a common follow-up task should be the whole document integration that embodies the fusion of objects by removing redundancies and resolving inconsistencies between them. Papakonstantinou et. al. [5] provide a study about object fusion in mediator systems. In this paper, we concentrate on the first task, the object identification.

Consider the scenario where the user holds the contact list A and wants to identify related contacts in the source B. The document fragment describing the contact *John Smith* in the source A has apparently a counterpart in the source B. A data integration application, however, will have several difficulties to accomplish its task in this example. First, the well-known problems of record linkage tasks are present here, such as misspelling (e.g., *John Smith* versus *Jonh Smith*), and abbreviations (e.g., New York versus NY). Besides these problems, the underlying XML document structure poses additional complications. Different names are used for elements (e.g., *client* versus *customer*) and structural differences are present (further classification *mobile* and *work* in the *phone* element as well as various multiplicities, and *city* appearing at the second level in B). In the following sections, we will discuss these problems in detail.

3 Object Description and Comparison Fields

The first step to identify entities is the selection of the elements to be used in the comparison procedure. In [6], Weis observed the difficulty to ascertain which XML elements make up the object description of a real-world object in XML documents and proposed heuristics to determine automatically these elements. In relational record linkage procedures typographical mistakes can make the information provided by one relation insufficient to identify the real-world object it relates to. For example, the address information is often used to identify individuals in record linkage procedures and could be stored in a separate relation. Normally, a pre-processing step brings all the relevant fields together to an auxiliary table, which are not adequate for on-line procedures.

In any case, the selection of the comparison fields must be done with care. As a rough example, the *zipcode* element of the data sources A and B from Figure 1 seems to be a reasonable identifying parameter under the common assumption that having more information for matching would improve the matching[1]. However, the values of *zipcode* and *city* do not hold under conditional independence (naïve Bayes). This property is reinforced in small cities that could have just one Postal ZIP code. Several record linkage methods work under this assumption to make the computation of the total agreement weight more tractable using the individual agreement weights of each comparison parameter [2]. Erroneously computed weights can therefore affect the classification rule. Expectation-Maximization (EM) procedures could be used to deal with such a lack of independence [7].

Other desirable properties of comparison fields are discriminating power, underlying quality, and low probability of misreporting. These properties can however be conflicting. For example, values in the field *sex*, in spite of its low discriminating power, usally is properly recorded in a data source.

The considerations above emphasis the difficulty of a domain-independent and automatic selection of the comparison fields for any object of interest. One option is the utilization of learning methods or access to databases statistics. As more simple measure, matching tools could alleviate this task for the user providing hint fields for common objects (individual, book, etc).

4 Considerations Concerning Similarity Metrics

A similarity metric is a critical component of any record linkage procedure. A typical record linkage application must be able to assess the "closeness" between string-valued fields and numeric fields as well. We focus on string similarity because the problem is more challenging and more appealing for XML documents where data of string type are prevalent.

String similarity metrics can be classified in three categories: *edit distance, token based*, and *hybrid*. Roughly speaking, edit-distance models measures the distance between strings as the cost of the best sequence of *edit operations* (insert, delete, replace) that converts one string into the other. Popular examples from this class of metrics are *Jaro*/*Jaro-Winkler* and *Levenstein*. Token based similarity measures convert the string to be compared to token multisets and assess similarity manipulating these multisets where the order of the tokens is unimportant. An example of this class is the *term frequency/inverse document frequency* (TF-IDF), widely used in the information retrieval community. Hybrid measures combine token-based and string-based methods. For example, the SoftTFIDF measure, a variant of TFIDF, that uses a secondary edit distance to account not only by tokens that appear in both multisets, but also those that are similar.

The choice of a suitable similarity metric is not a trivial task. The performance of a metric can vary significantly depending on the domain of the dataset. The *Jaro-Winkler* metric is intended for short strings, what makes it suitable for comparing element labels but not their content. W. Cohen et. al. have observed that token-based methods do not perform well on Census datasets [8]. Surprisingly for such data sets, the *Jaccard* similarity, that simply computes $(|S \cup T|/(|S \cap T|)$ of two word sets, seems to perform better than a more sophisticated measure like SoftTFIDF. Bilenko et. al. provide adaptive methods representing edit distance with affine gaps using a hidden Markov model [9].

---

1 [7] observed that having more than 6-10 fields for matching does not improve the result quality.

5 Approximate Joins of XML Documents

As soon as the comparison fields are determined, the next step is to bring together the related entities in the XML documents under consideration using record linkage techniques. The closeness of agreement of each field is used to calculate the total agreement weight ultimately used by the classification rule. In on-line procedures such as approximate joins, the classification rule is normally a threshold value.

The problem is how to perform a join between sources as described in Figure 1 where the relationship among elements from each source is asymmetric. Moreover, even the correctness of element tag names is not guaranteed due the presence variations such as synonyms.

The latter problem can be mitigated using a thesaurus. A domain-specific one could be obtained from ontologies, expert support, or through information retrieval techniques such as bootstrapping. Otherwise, a universal thesaurus such as WordNet could be used as well. Reference [10] presents a method to perform term expansion in a query using an additional normalized structural index with an entry for each set of synonyms within the thesaurus.

5.1 Using the tree edit distance

Reference [4] uses the tree edit distance [11] as a join predicate applied to pairs of subtrees of the XML documents. Pairs within a tree edit distance above or equal a determined threshold are reported in the output. To alleviate the cost of computing the tree edit distance ($O(|T1||T2|h(T1)h(T2))$, where $|Ti|$ is the number of nodes in tree Ti and h(Ti) is the height of Ti), upper and lower bounds were developed. In addition, a vector of distances to a reference set is calculated for each document from the sources to be integrated. The combination of both methods achieves a considerable pruning of the comparison space by tree edit distance. An adaptation of the R-tree to index the difference vectors were also presented. The authors argued that your framework can be easily adapted to any distance between trees, as long as it is a metric.

A question is to what extent a general distance function such as the tree edit distance can capture the semantic similarity between entities represented by an XML subtree. An obviously improvement (as suggested in [4]) is to apply a similarity metric function to string node labels instead of assuming edit operations of unit cost for node relabeling. This measure, however, makes the total cost even more expensive and stress the necessity of suitable filters.

Even with the support of a reasonable similarity measure, using the tree edit distance on the entire subtree still seems a coarse method. For example, it does not takes advantage of the XML hierarchical structure to improve the matching of elements (Section 5.2). A better approach could be to apply tree edit distance just to align the structure of the pair of document fragments when the schema of the data sources are heterogeneous. A string similarity metric suitable for short strings such as *Jaro-Winker* and the support of a thesaurus seem to be reasonable. This approach needs further investigation, specially into how to weight the cost of tree edit distance with the similarity measures applied on the content in the subsequent task.

5.2 Using the XML structure to improve the matching

Once the structure is aligned. one can perform the record linkage based on the comparison fields mentioned in Section 2. The hierarchical structure of XML allows the use of structural relationships to improve the match of entities. Such improvements can be obtained exploring relationships between the axes types *descendant* and *ancestor* to perform adjustments in the similarity (or likehood ratios) of matching elements and pruning the comparison space as well.

For example, matching or non-matching decisions between children elements can be propagated to augment or decrease the similarity score of their parents. For example, the similarity from two elements labeled *book* can be incresed when the children elements *author* are matched. On the other hand, miss-matching between parents can decide definitively for classifying the whole subtree above them as non-matchings and avoid further comparisons. [12] constructs a graph structure to capture the dependencies between entities and later to propagate similarities after reconciliations decisions.

Another important decision is how to transverse the document subtree to perform the elements comparisons. A top-down strategy seems to be preferable because has it the advantage of allowing earlier pruning when an ancestor is classified as non-matched. A hybrid approach could compare at least one child before classify two parents pair as non-matching.

6. Conclusion

In this paper, we introduced an approach to entity identification in XML documents based on approximate joins. We pointed out important aspects for an on-line record linkage architecture. The first step is a good selection method for the comparison fields of an entity. Desirable properties for such a method are conditional independence, discriminating power, underlying quality, and low probability of misreporting. Furthermore, string similarity metrics play a central role in a gerenal framework for record linkage techniques. Because the performance of different metrics could vary significantly depending on the domain of the dataset, we need empirical results to evaluate their quality and practical usefulness.

Our future work includes the implementation of such a framework in our prototype XML database system called XTC (XML Transaction Coordinator [13]. Furthermore, we will design and explore appropriate record linkage methods. XTC can be considered as an ideal testbed for comparative studies, because we can easily implement alternative approaches and run them under identical conditions. Therefore, by running benchmarks and comparative experiments against competing approaches (see Section 5), we can stepwise improve our approach based on empirical evidence. Most important for this proceeding is the definition of suitable baseline experiments at the beginning to observe progress in our endeavor to improve the quality of record linkage techniques.

References

1. Newcombe, H., Kennedy, J., Axford S., James, A.: Automatic linkage of vital records. In Science 130, no. 3381, 1959, pp. 954-959.
2. Fellegi, P., Sunter, A. B.: A theory for record linkage. In Journal of the American Statistical Association, 64, 1969, pp. 1183-1210.
3. Rahm, E., Do, H.-H.: Data Cleaning: Problems and Current Approaches. IEEE Data Eng. Bull. 23(4), 2000, pp. 3-13.
4. Guha, S., Jagadish, H. V., Koudas, N., Srivastava D.: Integrating XML Data Sources Using Approximate Joins. In Transactions on Database Systems, Upcoming Version, 2006.
5. Papakonstantinou, Y., Abiteboul S., Garcia-Molina, H.: Object Fusion in Mediator Systems. In Proc. of the 22nd VLDB Conference. Bombay, India, 1996, pp. 413-424.
6. Weis, M., Naumann, F.: DogmatiX Tracks down Duplicates in XML. In Proc. SIGMOD Conf.. Baltimore, USA, 2005, pp. 431-442
7. Winkler, W. E.: Overview of Record Linkage and Current Research Directions. Survey Research Report Series, Statistical Research Division, U.S Census Bureau. Washington, USA, 2006.
8. Cohen, W. W., Ravikumar, P., Fienberg, S. E.: A comparison of string metrics for matching names and records. KDD Workshop on Data Cleaning and Object Consolidation. Washington, USA, 2003.
9. Bilenko, M., Mooney, R. J., Cohen, W. W., Ravikumar, P., Fienberg, S. E.: Adaptive Name Matching in Information Integration. IEEE Intelligent Systems 18(5): 2003, pp. 16-23.
10. Li, Y. Y., Yu, C., H. V. Jagadish: Schema-Free XQuery. In Proc. of the 30th VLDB. Toronto, Canada, 2004, pp. 72-83.
11. Zhang, K., Shasha, D.: Tree Pattern Matching. Pattern Matching Algorithms, Apostolico and Galil Editors, Oxford University Press, 1997.
12. Dong, X., Halevy, A. Y., Madhavan J.: Reference Reconciliation in Complex Information Spaces. In Proc. SIGMOD Conf. Baltimore, USA, 2005, pp. 85-96.
13. Haustein, M. P.:Feingranulare Transaktionsisolation in nativen XML-Datenbanksystemen, Verlag Dr. Hut, München, Januar 2006.