Hide the Duplicate WebPages

[1]Bolla Anil Kumar, [2]Satya P Kumar Somayajula
[1]Avanthi Institute of Engg & Tech, Tamaram, Visakhapatnam, A.P, India.
[2]Dept. of CSE, Avanthi Institute of Engg & Tech, Tamaram, Visakhapatnam, A.P, India.

Abstract
Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task. Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors. In this paper, we present a thorough analysis of the literature on duplicate record detection. We cover similarity metrics that are commonly used to detect similar field entries, and we present an extensive set of duplicate detection algorithms that can detect approximately duplicate records in a database. We also cover multiple techniques for improving the efficiency and scalability of approximate duplicate detection algorithms. We conclude with coverage of existing tools and with a brief discussion of the big open problems in the area.

Keywords
Meta data, Info Quilt architecture, Semantic web, Encapsulation agents, Meta base, correlation agents

I. Introduction

There are now many searchable databases on the Web. These databases are accessed through queries formulated on their query interfaces only which are usually query forms. The query results from these databases are dynamically generated Web pages in response to form-based queries. The number of such dynamically generated Web pages is estimated around 500 times the number of static Web pages on the surface Web. In many domains, users are interested in obtaining information from multiple sources. Thus, they have to access different Web databases individually via their query interfaces. For large-scale data integration over the Deep Web, it is not practical to manually model and integrate these Web databases. We aim to provide a uniform query interface that allows users to have uniform access to multiple sources. Users can submit their queries to the uniform query interface and be responded with a set of combined results from multiple sources automatically. Schema matching across query interfaces is a critical step inWeb data integration, which finds attribute correspondences between the uniform query interface for a local database. In general, schema matching takes two schemas as input and produces a set of attribute correspondences between the two schemas. The problem of schema matching has been extensively studied. Some of these methods make use of information about schemas, including structures, linguistic features, data types, value ranges, etc to match attributes between schemas. Match results from individual matchers are not accurate and certain, because they rely on individual aspects of information about schemas only, which are not sufficient for finding attribute correspondences between schemas. Individual matchers however can generate some degree of belief on the validity of possible attribute correspondences.

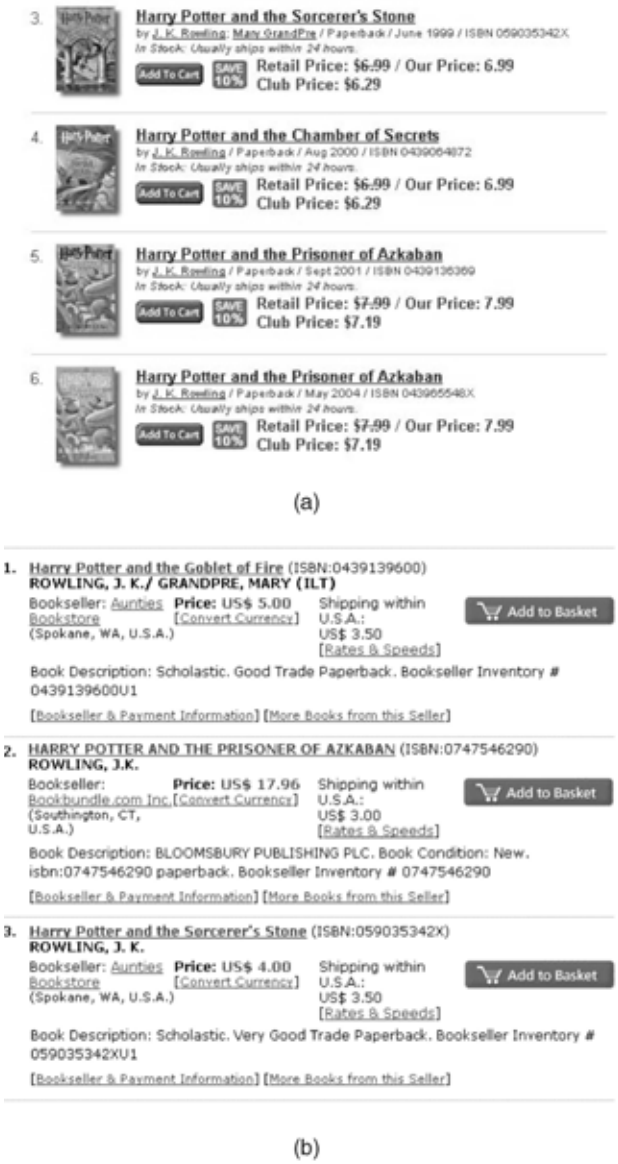

Fig. 1(a) & (b) : Multiple databases after user submit the query

In addition, sometimes given a source attribute, there might be two or more attribute correspondences that are not clearly distinguishable from each other by an individual matcher. For example, a data type matcher may not be able to distinguish some attribute correspondences for the same source attribute if they all have the same data type as the source attribute. Recent research efforts have been focused on combining multiple matchers. Finally, sometimes two or more different source attributes may have the same attribute correspondence. In our approach, we keep the top-k matches of each source attribute. We then use some heuristics to resolve any conflicts between the matches of different source attributes.

II. Meta Data

Metadata represent information about the data in individual databases and data repositories. They may represent relationships between individual media objects. These metadata descriptions may be extracted using various mappings/extractors (e.g., see,

[SSK95, KSS95]) associated with the various types of digital data. In this paper, we consider the following types of metadata (see [KSS95] and [B98] for two classifications, [BKS98] for a review of research and standards on metadata of digital media):

A. Content-independent metadata

This type of metadata is independent of the content of the artifact or document1 it describes, e.g. location, date-of-creation etc.

B. Content-dependent metadata

This type of metadata captures the information content of the document. We define three types of content-dependent metadata.

C. Content-dependent metadata

This type of metadata depends directly on the document content, e.g. keywords appearing in a document, colors appearing in an image document. One method of representing content-based metadata is using a collection of attribute-value pairs. A discussion of attribute-based access for textual data is discussed in [SKL95]. The attributes chosen may be media specific (e.g. color) or media independent (e.g. location, relief).

D. Content-descriptive metadata

This is a special case of Content-dependent metadata where the content of a document is described in a manner which may not be directly based on the contents of the document. Examples of content-descriptive metadata for images may be found in [OS95, KKH94] where textual annotations are associated with images and are used to correlate information across image and textual documents.

E. Domain-specific metadata

This is a special case of content-descriptive metadata typically represented in an attribute-based manner where the attributes used to characterize documents are domain-specific in nature, e.g. relief for the Geographical Information Systems domain. Metadata may be precompiled (and possibly stored in a database) or it may be computed when needed (at a query processing time), in which case it may be represented by a computation (e.g., an image processing routine giving values for land-cover metadata of a satellite image, executed when needed).

III. MREF

MREFs are views defined using metadata of various types and of various media. As with HREFs, an end user may only see a link on a (possibly dynamically created) Web-page. However, MREFs can represent information requests or views involving keyword-based, attribute-based and content-based specifications involving various types of metadata.

They are treated as virtual objects in the Info Quilt system. In relational databases a view is an abstract model that does not exist as a static object in the system. A SQL query is one way of representing a relational view. Other representations can be constructed for the same abstract view object. Usually a view itself is materialized when the query, or some other representation, is processed by the system. Alternatively, it is possible to have materialized views that hold data prior to the submission of a query. As described in this paper, the MREF abstract metadata based view is analogous to a relational view. The MREF objects are treated as virtual objects that can be referenced from Web objects. As with traditional views, they can be materialized in the system at run time or can be precomputed. Furthermore, parts of MREFs can be precomputed and others materialized at run-time.

IV. Info Quilt Architecture

The Info Quilt system has its roots in the Info Harness [SSK95] system, which was commercialized as the Adapt/X Harness system at Bellcore. The InfoHarness system provided the proof-of-concepts for the various building blocks that form the core of the InfoQuilt system. InfoHarness addressed the system level issues of metadata extraction and management. However, InfoHarness was not intrinsically distributed. InfoQuilt elevates the ideas and goals of the InfoHarness system to a higher level of abstraction. It focuses on the logical representation and correlation of encapsulated information artifacts and is fully distributed from the ground up. MREFs play a central role in the InfoQuilt system. A high level view of the InfoQuilt system (see Figure 1 for its architecture) is useful for sketching a complete picture of how MREFs (discussed in detail in the next section) provide the glue for metadata enabled information management and resource discovery. The functionality of the various subsystems is discussed next.

A. Encapsulation Agents:

These mobile agents are responsible for determining the type of the underlying information artifacts to be encapsulated, and processing the artifacts themselves to extract content-dependent and content-independent metadata. This extracted metadata is modeled as a RDF object and is handed over to the metadata store (met abase).

B. Meta base

This is a persistent RDF object store. The meta base provides functionality to process keyword, attributed, and content-based queries. The meta base also provides support for structural modeling of the metadata repository to aid in user browsing, visualization, etc.

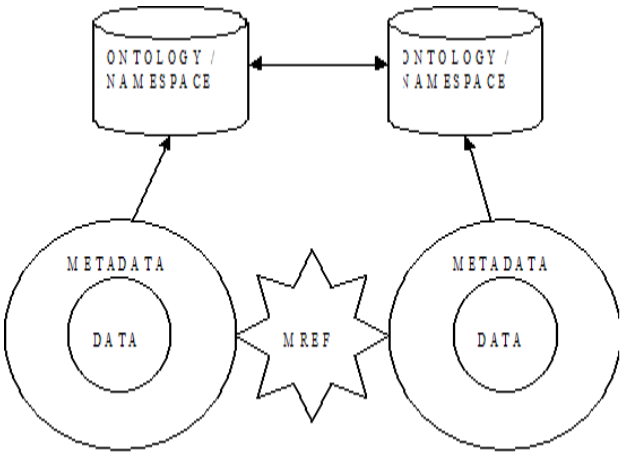

Fig. 2: Abstraction Layers

C. Broker Agents

These agents are responsible for decomposing the MREFs into partial queries. The broker agents interact with the meta bases for this purpose. The broker agents also decide which meta base(s) to contact for a given MREF component based on the user profiles, traffic, etc. These brokers are also responsible for merging the results that come back from the various meta bases. Further correlation can be done by interacting with the correlation agents.

Correlation Agents: These agents are ontology managers and correlate MREFs based on the respective ontologies. Metadata Directories: These sites manage information used by various components of InfoQuilt for dereferencing MREFs based on their metadata components and maps this information to specific metabases.

The metabases themselves register the metadata that they serve with these directories. MREF representations are stored here. The user agents dereference MREFs embedded in Web objects to MREF representations stored in these directories. These are merely representations (as described in the next section); their instantiation is done at run time. Based on various temporal factors and the state of the underlying information artifacts themselves, the MREFs could have different instantiations at different points in time. The Info Quilt MREF namespace management is also done here. The broker agents use these directories as described next.

User Agents: MREF construction, interpretation, and translation are done here. As described above, MREFs can be embedded in Web objects (e.g., HTML files) or can exist as standalone artifacts themselves. The user agents are responsible for constructing user queries and for initial MREF processing.

IV. Building logical, semantic webs

The Web as it exists today is a graph of information artifacts and resources. The graph nodes are represented by embedded HREFs. These enable the implicit linking of related (or unrelated) web artifacts. This web is very suitable for browsing but provides little or no direct help for searching. Web crawlers and search engines try to impose some sort of an order by building indices on top of the web artifacts, which are primarily textual. These efforts face an ever-increasing problem of scalability resulting in lower precision and incomplete coverage. However, in this scenario we can trivially say that a keyword query imposes a correlation (logical relationship) at a very basic (limited) level between the artifacts that make up the result set for that query.

Metadata is the key to this correlation. For a keyword query we can conceptually view the keyword index as content-dependent metadata and the keywords in the query as specific resource descriptors for the index, the evaluation of which would result in a set of correlated resources. To be more general, we need a framework for expressing metadata based, media independent correlation across federated digital media.

How much of the correlation is done automatically by the query processing system? The level of automation usually depends (inversely) on the information content captured in the metadata. How meaningful is the correlation? This, on the other hand, depends (directly) on the information content captured. For query processing systems to adequately address these design considerations, it is desirable to move towards location-independent, media-independent, and content-dependent methods of correlation specific to the domain of information [SK96].

VI. Conclusion

Duplicate detection is an important step in data integration and most state-of-the-art methods are based on offline learning techniques, which require training data. In the Web database scenario, where records to match are greatly query-dependent, a pretrained approach is not applicable as the set of records in each query's results is a biased subset of the full data set.

To overcome this problem, we presented an unsupervised, online approach, UDD, for detecting duplicates over the query results of multiple Web databases. Two classifiers, WCSS and SVM, are used cooperatively in the convergence step of record matching to identify the duplicate pairs from all potential duplicate pairs iteratively. Experimental results show that our approach is comparable to previous work that requires training examples for identifying duplicates from the query results of multiple Web databases.

VII. Future Scope

As we know that our website support based on different domains we are getting the same domains, books etc. To reduce the duplicates our present system is very useful. So, mainly based on url we have to reduce the duplicates. This is very useful because we don't gather the same information from different domains. Actually this is time wasting process. To solve this problem, we are using this present system.

References

[1] R. Ananthakrishna, S. Chaudhuri, V. Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses," Proc. 28th Int'l Conf. Very Large Data Bases, pp. 586-597, 2002.

[2] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval. ACM Press, 1999.

[3] R. Baxter, P. Christen, T. Churches, "A Comparison of Fast Blocking Methods for Record Linkage," Proc. KDD Workshop Data Cleaning, Record Linkage, and Object Consolidation, pp. 25-27, 2003.

[4] O. Bennjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S.E.Whang, and J. Widom, "Swoosh: A Generic Approach to Entity Resolution," The VLDB J., vol. 18, no. 1, pp. 255-276, 2009.

[5] M. Bilenko, R.J. Mooney, "Adaptive Duplicate Detection Using Learnable String Similarity Measures," Proc. ACM SIGKDD, pp. 39-48, 2003.

Anil Kumar Bolla, is persuing M.Tech (CSE) degree at Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, A.P., India. He has received his Bachelors Degree from Loyola Institute of Technology and Management affiliated to Jawaharlal Nehru Technological University Kakinada.

Mr. Satya P Kumar Somayajula is working as an Asst.Professor, in CSE Department, Avanthi Institue of Engg & Tech, Tamaram, Visakhapatnam,A.P., India. He has received his M.Sc(Physics) from Andhra University, Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM University), Visakhapatnam, A.P., INDIA. He published 7 papers in reputed International journals & 5 National journals. His research interests include Image Processing, Networks security, Web security, Information security, Data Mining and Software Engineering.