ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(22), DOI: 10.17485/ijst/2015/v8i22/78957, September 2015* ISSN (Online) : 0974-5645

Conflict Resolution and Duplicate Elimination in Heterogeneous Datasets using Unified Data Retrieval Techniques

I. Carol* **and S. Britto Ramesh Kumar**

Department of Computer Science, St. Joseph's College, Trichy - 620002, Tamil Nadu, India;
carolcrl23@gmail.com, brittork@gmail.com

Abstract

Background/Objective: Creating queries for a single search term and identifying the viable solutions for the query are the two specific problems in retrieving the data. To resolve this issue, an effective information fusion technology should be provided to obtain effective results. This paper presents a method for resolving conflicts and eliminating duplicates with increased accuracy. **Methods/Statistical Analysis:** Universal wrappers are designed to retrieve the actual information from the heterogeneous data sources. The process of getting input itself is modified such that the retrieved results are relevant to the context. Ranking and duplicate eliminations are done accordingly to refine the obtained results to the user. **Findings:** Experimental results show that the improved accuracies of the data being fetched and with reduced conflicts and duplicates. This work uses major data sources from Google, New York Times and other offline data sources. By applying the proposed data retrieval techniques, the produced data is consistent by the help of wrappers. The proposed approach improves the data consistency which is relatively better than the existing technique. Finally, this proposed research work concludes that it is used to identify and resolve the conflict data and delivers the consistent data to the users in a ranked manner. **Applications/Improvements:** To create a unified repository which can be used for knowledge mining and warehouse based analysis of existing data and retrieve the result.

Keywords: Conflict Identification, Conflict Resolution, Data Retrieval, Ranking, Wrappers

1. Introduction

Increase in the amount of information in all areas and reduction of the cost of storage devices has led to the creation of increased repositories. The repositories used for storage vary considerably in their mode of storage and storage structure. Most of these structures are plain information stores and hence do not support any kinds of processing. Hence it becomes mandatory to design a system that retrieves this data and processes it[1]. Besides structured data, a large amount of information in the Internet is unstructured, retrieved from operational systems and stored. Storing all the required information in a single warehouse is not a feasible solution. The only feasible solution is to access the data from its data source as and when required. This mechanism is complex due to the sheer heterogeneity associated with it. The required data is usually distributed and the distribution is in several formats. Two specific problems existing in this scenario is the complexity associated with the retrieval of data due to the variety associated in creating queries for a single search term and the complexity associated with integrating the data into a single format and identifying the viable solutions for the query.

A geo temporal web gazetteer web service, it provides details about various entities in a temporal manner. This method serves as a unified repository of geographic and temporal information[2]. A query set designed to build a data warehouse is presented using metadata[3]. Due to the huge amount of information handled by organizations, an effective system that processes these data has become a necessity. This process is carried out. Rather than building

*Author for correspondence

a specific result set, this method tends to build a complete warehouse, on which OLAP and OLTP operations can be performed to enhance the result base. A CASE tool that builds a data warehouse from a set of relational databases is used in this work[3]. The major downside of this approach is that it is designed specifically for relational databases, hence it is format constrained. A similar method is presented in XML schema, which describes the web warehouse itself as a fusion of data integrated from web[4]. A web warehouse system suitable for knowledge mining is presented with a four layer architecture that also aids in decision support[5]. This method constructs an Extraction-Fusion-Mapping-Loading (EFML) process model and uses a variety of wrapper services in the construction process. A heterogeneous data source fusion method is described; its major advantage of this method is that it works on both static and dynamic data sources[6]. The data can also be structured or semi-structured. Unstructured data is not well supported here.

Multi database languages have also gained prominence in recent years due to their usage in web warehouses. MSQL[7] and MDSL[8] are two of the most prominently used multi database languages. Though both these languages are used to access structured data, the mapping methods used in them are effective; hence they are used in most warehouses. There exist several systems that perform web information fusion, some of them are listed here; TSIMMIS[9], GARLIC[10], MIX[11], DISCO[12], MOMIS[13], INFORMATION MANIFOLD[14], AGORA[15,16], C-WEB[17] and XYLEME[18]. Other systems that are developed in peer-to-peer context are AXML[19], SENPEER[20] and PIAZZA[21].

The remainder of this paper is structured as follows; Section 2 provides the system architecture and provides a brief description of the proposed system. Section 3 explains the proposed contribution in detail, Section 4 presents the results and Section 5 concludes the study.

2. System Architecture

Retrieving data from various data sources and finally combining them to provide the results are carried out in the process of conflict identification and resolution. But the problem in such an approach is that some data sources might not contain any data pertaining to the current query, or multiple data sources might contain similar data, which will lead to conflicts. The current system presents methods that reduce the need for unnecessary querying being carried out in the data sources and performs effective duplicate eliminations.

The process of conflict identification is divided into four broad phases; the input segregation phase, query building phase, data source short listing and query execution phase and the result analysis phase. The input segregation phase obtains the input from the user in the form of advanced input parameters, rather than the conventional form of input. These parameters play a vital role in the building up of appropriate queries to perform the data retrieval.

The process of query building is carried out by using universal wrappers rather than using separate wrappers for each data source. An XML query template is prepared, containing query codes that can be used to build up queries. In order to add further data sources, it is sufficient to add particular template sections.

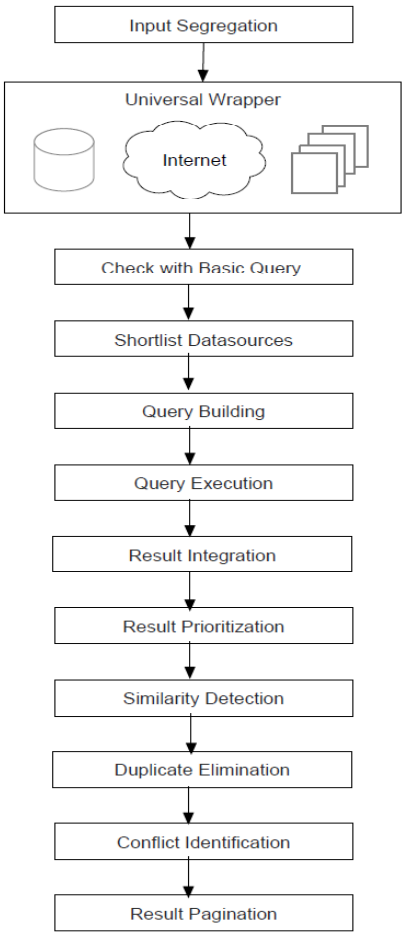

Figure 1. Conflict resolution and duplicate reduction using unified data retrieval techniques.

Every data source is initially tested to find out if it contains the results corresponding queries. This is a simple count query that does not require much computation. The shortlisted data sources are then executed with the actual query and results are retrieved.

The retrieved results are then passed to the similarity detection and the duplicate elimination phases. The shortlisted results are passed to the conflict identification phase and are finally ranked appropriately for the final results.

Conflict resolution is one of the major concerns pertaining to information retrieval. The longer it takes for the system to process and return the results, the lesser the impact it creates with the customers. Effective data retrieval requires appropriate use of time and hence ignoring areas with least importance becomes mandatory. Further, it also becomes mandatory for the system to provide appropriate rankings to the results. Lesser the results, better and faster would be their rankings. The proposed method presents a four phase mechanism that provides effective utilization of time to fetch only appropriate results in the best possible time.

2.1 Input Segregation

The input segregation phase acquires the input phrase from the users. The difference here comes from the fact that the input phrase is not composed of just a simple string. Instead, the input is composed of several factors, which not only determines keywords, but also key phrases and phrases to be eliminated, such that the results that are returned are refined to the maximum possible extent. A sample screenshot of the input phase is presented in Figure 2.

This phase acts as a major contributor to the reduction of results and inclusion of appropriate results during the retrieval of results.

2.2 Query Building using Universal Wrappers

The query building phase is performed by universal wrappers. A query template is maintained, that helps in the query building phase. The template maintains all the data pertaining to the query constructs. This data helps in building up the appropriate query based on the input parameters specified by the user. Due to the generality associated with the query template, providing slight modifications to the template is sufficient to include additional data sources operating on a different query construct.

```
All these words: Windows 10 release Date
Exact words: Windows 10
Any of these words: microsoft windows
None of these words: ubuntu
```

Figure 2. Snapshot of the Input phase.

When a query is presented by the user, the query parameters are separated into constructs and these constructs are passed to the query builders to construct a query. The query building phase is called twice in the actual process. As soon as the query is presented, the query builder returns a counter query with minimal conditions that identifies the availability of results in the data sources. After the shortlisting process, the wrappers are again used to build the actual query for data retrieval.

2.3 Data Sources Shortlisting and Query Execution

The first query to be executed in the data sources is the counter query. Not all data sources might contain data pertaining to the current query. While some data sources contain huge amount of information, others may contain no information at all. Hence passing the query to all the data sources and treating them as equals will lead to wastage of time and processing resources. Hence it is effective to identify the appropriate data sources and query only those selected data sources for faster and more appropriate results.

The data retrieval query is then built and is passed to the shortlisted data sources. Due to the inclusion of appropriate query parameters in the input phase, the results that are fetched are generally of high importance. This also reduces the retrieval of duplicates, which is common while using a general query with several key phrases. Due to the specific nature of the query, general results are eliminated and the results that are returned are lesser and much appropriate and hence processing them becomes faster.

The results that are returned by the query are in different formats. Web based results are in JSON formats and results from data bases are in the table formats and document based records return results in plain text format or XML formats. Due to the inconsistency in the data returned, they cannot be directly used for processing. Hence an integrated result repository is created to collect the records in a common format. JSON and XML data formats tend to contain various information pertaining to the data. These metadata are collected and stored in the

property storage, and are useful during the resolution of conflicts. The actual data is identified and are integrated to the data repository. In case of tables, column names are used to identify the appropriate data. Certain tables tend to distribute the data in several columns. Such schema also needs to be identified and appropriate integration strategies are to be framed. This completes the creation of the data repository for the current query.

2.4 Result Analysis

The next phase is the result prioritization phase. The results that have been returned are ranked according to the key phrases provided in the query. The number of key phrases contained in the document and the precision of the phrases will determine the rank of an entry. Precision refers to the multi word key phrases and the location of their presence in the document. In case of a tie, the property storage containing Metadata of the queries are used to rank the results.

The system then moves to the process of similarity detection. Due to the retrieval of data from several sources, the system is prone to contain duplicates[22]. The similarity scores help to identify the level of similarity existing in the given text. Several methods exist in literature for calculating the similarity of elements. The current method calculates the directional similarities of text and finally the total document similarity[23,24].

In Equation 1, a directional similarity score $sim_d(T_i, T_j)$ is computed from a text T_i to a second text T_j Therefore, for each word W_i in T_i, its best-matching counterpart in T_j is required (maxSim(W_i, T_j)). The similarity scores of all these matches are summed up and weighted according to their inverse document frequency, and then they are normalized. The final document-level similarity is the average of applying this strategy in both directions, from T_i to T_j and vice versa.

$$sim_d(T_i, T_j) = \frac{\sum_{w_i} maxSim(w_i, T_j) \cdot idf(w_i)}{\sum_{w_i} idf(w_i)} \quad (1)$$

$$sim(T_i, T_j) = \frac{1}{2}\left(sim_d(T_i, T_j) + sim_d(T_J, T_i)\right) \quad (2)$$

In Equation 2, the similarity scores are compared and in document pairs (i, j) exhibiting 90% similarities, one of the document is eliminated to reduce the total result set. The eliminated document is the one containing the least weight of the two documents in consideration. If multiple documents contain duplicated data then the last (n-1) entries are eliminated. Since the major concentration of this method is to present only the precise results, it becomes mandatory to eliminate data that are redundant and would be of very less usage to the user. It could be observed from Section 5 that due to the improved query construction techniques, duplicates are eliminated to the maximum extent.

The next phase is the conflict identification and elimination phase. Documents having moderate similarity scores are considered as documents in conflict. In such cases, the current method follows two principles.

If the entries in conflict have a huge difference in their ranking, then the conflicts are ignored. This is due to the fact that entries in conflict do not appear together and one of them is mostly in the last percentile of the rank. Hence it obviously has very less importance and has very low probability of being processed by the user. The difference threshold is provided by the user.

If the entries in conflict appear in a location difference less than the defined threshold then conflict elimination is to be carried out by eliminating one of the entries. Both the entries are analyzed with their corresponding property sets and the document containing the least score from the property set is eliminated and its counterpart is retained. The rank of each document also plays a vital role in determining the status of the document.

3. Results and Discussion

The process of inconsistency resolution was carried out using data sources from Google, New York Times and other offline data sources. Data source variety depends on the user and due to the usage of the universal wrapper the system handles all varieties of data. The query template is modified slightly according to the requirements of the data source being included.

The retrieved data formats vary in their structure. Google and NYT APIs return results in the form of JSON data, while the local data sources returns their results in the form of tables or XML data.

Figure 3 shows the time taken for data retrieval on queries with varying complexities. It could be observed that the time taken for such process is not defined and varies to a very large extent. On comparison with the result graph in Figure 3, it could be observed that though the graphs depict variations in the values, their structures

show similarities. They are found to be proportional to each other. Hence it can be concluded that the time taken for retrieval of data varies relative to the number of data retrieved from the data sources.

Queries that do not belong to the current ontology were also provided to the system for testing. Several result entries are found to point to 0 in the y axis. This can be better observed in the result graph. If the required result for the query is not present in any of the data sources being used then the time taken for the query is almost negligible, so it takes 0 ms. This is one of the most useful properties associated with our contribution, which shows the absence of the fixed constant time required to return results.

Figure 5 shows the analysis of duplicate records that are retrieved from the data sources. It could be observed from the graph that the number of records removed from the final results is very minimal. The maximum record removal is of the order of 6 records from a result of 20 records. The improved input segregation and query building procedures reduce the amount of data retrieved and improved the accuracy of the retrieved results in such a manner that the possibility of redundancies occurring in the result set has been reduced.

The above Figure 6 shows the rate of elimination in the queries retrieved. It could be observed that the complexity of the query does not play a role in the elimination of results. A maximum of 40% and a minimum of 0% elimination rate have been observed in the current contribution. This exhibits the efficiency of the query retrieval mechanism.

Figure 7 shows the number of conflicted entries existing in the current method vs. the total entries retrieved from the current query. It could be observed that the conflicts are also very less. This depicts the efficiency of the current contribution.

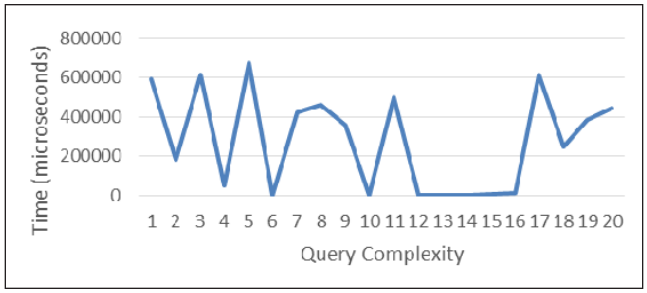

Figure 3. Time analysis.

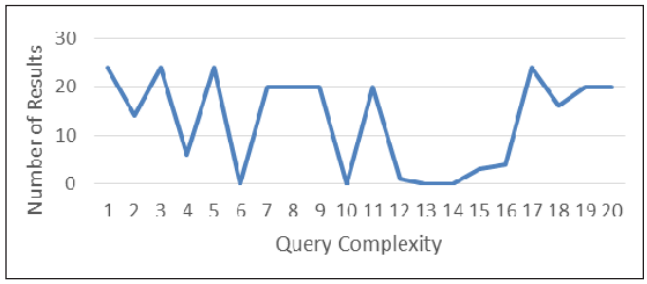

Figure 4: Result count.

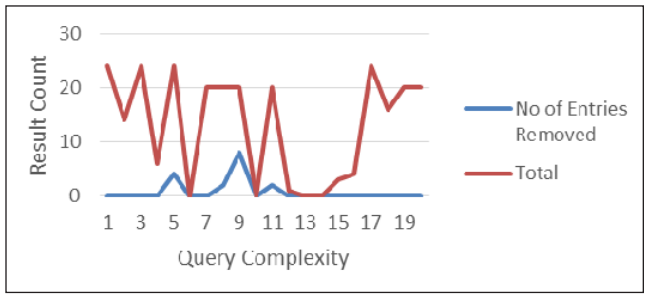

Figure 5. Duplicate analysis.

4. Conclusion

This paper presents an effective method that retrieves the most relevant documents from several heterogeneous data sources and eliminates conflicts and duplicates effectively. The process of getting input from the user is modified such that the system is able to retrieve only the appropriate results and eliminate the other results effectively. Future directions for expansion include optimization of

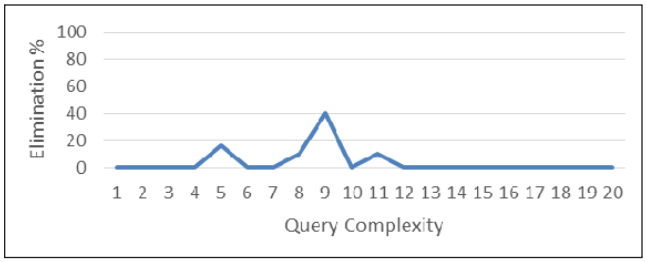

Figure 6. Elimination rate.

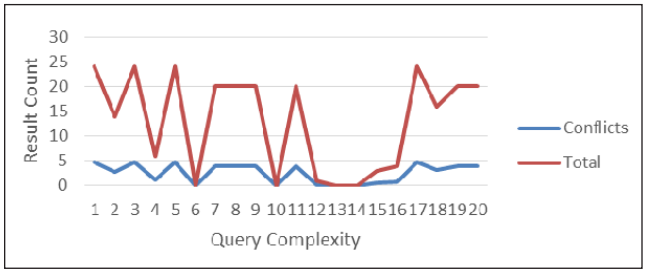

Figure 7. Conflict rate.

the query construction phase. Since several data sources are involved, constructing an optimized query will lead to faster query execution and hence faster results. The current method is presented as a retrieval mechanism alone. A metadata based approach that creates a warehouse structure to enable knowledge mining would provide a platform for enhanced analysis of the existing data rather than retrieving it for a use and discard method.

5. References

1. Kimball R. The data warehouse toolkit: Practical techniques for building dimensional data warehouses. New York: John Wiley and Sons Inc.; 1996.
2. Manguinhas H, Martins B, Borbinha J. A geo-temporal web gazetteer integrating data from multiple sources. IEEE International Conference on Digital Information Management (ICDIM 2008); IEEE; p. 146–53.
3. Wua L, Millera L, Nilakanta S. Design of data warehouses using metadata. Information and Software Technology. 2001; 43(2):109–19.
4. Vrdoljak B, Banek M, Rizzi S. Designing web warehouses from XML schemas. Proceedings of 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003); 2003. p. 89–98.
5. Yua L, Huangc W, Wanga S, Laib K. Web warehouse – A new web information fusion tool for web mining. Information Fusion. 2008; 9(4):501–11.
6. Nachouki G, Chastang M. Multi-data source fusion approach. The International Journal of Database Management Systems (IJDMS). 2010; 2(1):60–79.
7. Litwin W, Abdellatif A, Zeroual A, Nicolas B, igier Ph. MSQL: A multidatabase language. Journal on Information Sciences. 1989; 49(1–3):59–101.
8. Litwin W, Abdellatif A. An overview of the multidatabase manipulation language MDSL (Invited paper). Proceedings of the IEEE; 1987. 75(5):621–32.
9. Molina H, Hammer J, Ireland K, Papakonstantinou Y, Ullman J, Widom J. Integrating and accessing heterogeneous information sources in TSIMMIS. Proceedings of the AAAI Symposium on Information Gathering; 1995. p. 61–4.
10. Carey MJ, Haas LM, Schwarz PM, Arya M, Cody WF, Fagin R, Flickner M, Luniewski AW, Niblack W, Petkovic D, Thomas J, Williams JH, Wimmers EL. Towards heterogeneous multimedia information systems: The garlic approach. Proceedings of the 5th International Workshop on Research Issues in Data Engineering-Distributed Object Management (RIDE-DOM'95); 1995. p. 161–73.
11. Baru C, Gupta A, Ludaesscher B, Marciano R, Papakonstantinou Y, Velikhov P, Chu V. XML-based information mediation with MIX. Proceedings of the International Conference on Management of Datal; 1999. p. 597–9.
12. Tomasic A, Raschid L, Valduriez P. Scaling heterogeneous databases and the design of Disco. Proceedings of the Distributed Computing Systems Conference (DCSC); 1996. p. 449–59.
13. Beneventano D, Bergamaschi S. The MOMIS methodology for integrating heterogeneous data sources. IFIP Congress Topical Sessions; 2004. p. 19–24.
14. Kirk T, Levy AY, Sagiv Y, Srivastava D. The information manifold. The American Association for Artificial Intelligence (AAAI) Press. 1995; p. 85–91.
15. Manolescu I, Florescu D, Kossmann D, Olteanu D, Xhumari F. Agora: Living with XML and relational. Proceedings of the International Conference on Very Large Databases (VLDB); 2000. p. 623–6.
16. Manolescu I, Florescu D, Kossmann D. Answering XML queries over heterogeneous data sources. Proceedings of the International Conference on Very Large Databases (VLDB); 2001. p. 241–50.
17. Amann B, Beeri C, Fundulaki I, Scholl M. Ontology-based integration of XML Web resources. Proceedings of the International Semantic Web Conference (ISWC); 2002. p. 117–31.
18. Delobel C, Reynaud C, Rousset MC, Sirot JP, Vodislav D. Semantic integration in xyleme: A uniform tree-based approach. Journal of Data and Knowledge Engineering. 2003; 44(3):267–98.
19. Benjelloun O. Active XML: A data-centric perspective on Web services [PhD thesis]. Orsay, France: Paris XI University; 2004.
20. Halvey AY, Ives ZG, Mork P, Tartarinov I. Piazza: Data management infrastructure for semantic Web applications. Proceedings of the 12th International Conference on World Wide Web; 2003. p. 556–67.
21. Faye DC, Nachouki G, Valduriez P. SenPeer: Un system e pair-a-pair de mediation de donnees. International Journal ARIMA. 2006; 4:24–48.
22. Zesch Torsten D, Gurevych I. Text reuse detection using a composition of text similarity measures. Proceedings of COLING; 2012.
23. Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity. Proceedings of the 21st National Conference on Artificial Intelligence; Boston, MA, USA. 2006. p. 775–80.
24. Hatzivassiloglou V, Judith LK, Eskin E. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora; 1999.