A Method to Identify Duplicate Refresh Records with Continuous Query based Multiple Web Databases

SUDHAKAR
HOD – CSE
MKCE Karur

SHRIRAM K VASUDEVAN
Assistant.Professor,CSE
Amrita University

SIVARAMAN
R/Technical faculty
Amrita University

JAI VIGHNESHWAR
J/MNC
Coimbatore.

Abstract -- **Record matching, which identifies the records that represent the same real world entity is an important step for data integration. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Most existing work requires human-labelled training data (positive, negative, or both), which places a heavy burden on users. Existing supervised record matching methods require users to provide training data and therefore cannot be applied for web databases where query results are generated on-the-fly. A new record matching method named Unsupervised Duplicate Refresh Elimination (UDRE) is proposed for identifying and eliminating duplicates among refresh records in dynamic query results. The idea of this research is to adjust the weights classifier record fields in calculating similarities among refresh records. Three classifiers namely weight component similarity summing time bound classifier, support vector machine classifier and threshold-based support vector machine classifier are iteratively employed with UDRE where the first classifier utilizes the weights concentrated on string similarity measures for comparing records from different data sources. We also design a new record alignment algorithm that aligns the attributes in Identify Duplicate Refresh Records.**

Keywords— Duplicate detection, SVM classifier, user instant query, Continuous queries, document streams

I. INTRODUCTION

The Web contains a vast amount of non-crawlable content. This hidden part of the Web is comprised of a large number of online Web databases consisting of a searchable interface (usually an HTML form) and a backend database, which dynamically provides information in response to user queries. In the hidden Web, it is usually difficult or even impossible to directly obtain the schemas of the Web databases without cooperation from the web sites. Instead, the web sites present two other distinct schemas, interface and result schema, to users. The interface schema presents the query interface, which exposes attributes that can be queried in the Web database. The result schema presents the query results, which exposes attributes that are shown to users. The interface schema is useful for applications, such as mediators, that query multiple Web databases, since they need complete knowledge about the query interface of each database. The result schema is critical for applications, such as data extraction, which extract instances from the query results.

Data De-duplication: Data de-duplication or Single Instancing essentially refers to the elimination of redundant data. In the de-duplication process, duplicate data is deleted, leaving only one copy (single instance) of the data to be stored. However, indexing of all data is still retained should that data ever be required.

A typical email system might contain 100 instances of the same 1 MB file attachment. If the email platform is backed up or archived, all 100 instances are saved, requiring 100 MB storage space. With data de-duplication, only one instance of the attachment is actually stored; each subsequent instance is just referenced back to the one saved copy reducing storage and bandwidth demand to only 1 MB.

To our knowledge, this is the first work that studies and solves the online duplicate detection problem for the Web database scenario where query results are generated on-the-fly. In this scenario, the importance of each individual field needs to be considered, which may vary widely from query to query. This makes existing work based on hand coded rules or offline learning inappropriate. This is also the first work that takes advantage of the dissimilarity among records from the same Web database for record matching. Most existing work requires human-labelled training data (positive, negative, or both), which places a heavy burden on users. Our focus is on Web databases from the same domain, i.e., Web databases that provide the same type of records in response to user queries.

We present the assumptions and observations on which Unsupervised Duplicate Detection based on User Instant Query Results is based matching Query.

First, we make the following two assumptions.

A global schema for the specific type of result records is predefined and each database's individual and clustering based query result schema has been matched to the global schema. Record extractors, i.e., wrappers, are available for each source to extract the result data from XHTML pages and insert them into a relational database according to the global schema methods. We also make use of the following two observations. The records from the same data source usually have the same format. Most recent duplicates from the same data source can be identified and removed using an

exact matching method. Duplicate records exist in the query results of many Web databases, especially when the duplicates are defined based on only some of the fields in a record.

Using a straightforward pre-processing step, exact matching, can merge those records that are exactly the same in all relevant matching fields. We investigate 40 Websites for four popular domains on the Web. The simple exact matching step can reduce duplicates by 87 percent, on average. The main reason that exact matching is so effective at reducing duplicates User Instant Query Results is that the data format for records from the same data source is usually the same for all records.

II. RELATED WORK

Users do not want questions to be answered in the same way for the first couple of results when asking a question to a question answering system; they would like the system to return only those questions — or similar questions — with corresponding answers that differ from each other.

This translates to a duplicate detection and alternative answer detection task. The goal is to mark duplicate questions and answers so that search interface can remove the duplicates, or at least group these together. [1] One way to evaluate duplicate detection is to consider it a search task for items that match a value of a feature, in this case duplicate, over a number of input candidates and in which the number of returned results can vary. Two standard measures for evaluating performance in this case are precision and recall. [1]

The similarity calculation quantifies the similarity between a pair of record fields. As the query results to match are extracted from HTML pages, namely, text files, we only consider string similarity. Given a pair of strings (Sa and Sb) a similarity function calculates the similarity score between Sa and Sb, which must be between 0 and 1. Since the similarity function is orthogonal to the iterative duplicate detection, any kind of similarity calculation method can be employed in UDD (e.g., [2] and [3]). Domain knowledge or user preference can also be incorporated into the similarity function. In particular, the similarity function can be learned if training data is available [4]. Record linkage algorithms fundamentally depend on string similarity functions for record fields as well as on record similarity functions for string fields.

Similarity computation functions depend on the data type. Therefore the user must choose the function according to the attribute's data type, for example numerical, string and so on. This step uses Jaccard similarity function to compare token values of adjacent field values for selected attribute. Tokenization is typically formed by treating each individual word of certain minimum length as a separate token or by taking first character from each word.

Token has been created for the selected attributes. Each function measures the similarity of selected attributes with other record fields and assigns a similarity value for each field. In the next step, the clustering techniques have been selected to group the fields based on the similarity values. Accurate similarity functions are important for clustering and duplicate detection problem. Better string distance might also be useful to pair the record as match or non-match. This matching and non-matching pairs is used for clustering and to eliminate the duplicates [5]. The rule based duplicate detection and elimination approach is used for detecting and eliminating the records.

During the elimination process, only one copy of duplicated records are retained and eliminated other duplicate records [5] [6]. The elimination process is very important to produce a cleaned data. The above steps are used to identify the duplicate records. This step is used to detect and remove the duplicate records from one cluster or many clusters.

Before the elimination process, the similarity threshold values for all the records in the dataset are calculated. The similarity threshold values are important for the elimination process. The threshold criteria and certainty factors are used to detect and eliminate the duplicate records. Finally one record is retained as prime representative and maintained this value in the log file. This primary copy will be used for the incremental cleaning process also for further work. This approach can substantially reduce the probability of false mismatches, with a relatively small increase in the running time.

The simplest full-text approach is to adapt methods originally developed for search engines, for example, vector-space model, which treats a document as bag-of-words, with term weights determined by tf.idf values, and similarity determined by cosine similarity. Traditional cosine-similarity measure focuses on finding a semantic relevant document while near-duplicate detection focuses more on syntactic similarity.

Several previous works thus have been done in finding suitable similarity measures to address syntactic similarity among documents. [7] The identity measure proposed by [8] emphasizes that the gap between rare words' term frequency in two documents should be smaller than that between common words' and their best ranking is giving by a term weighting function biased towards rare terms. Metzler et al. [9] used statistical translation models to estimate the probability that one sentence in a document is a translation of another sentence in another document. The probability of aligning to an absent term is estimated by the background language model.

The translation probability serves as the basis of the sentence-level and the document-level similarity scores [9]. A Web database is usually comprised of a query interface and a backend database. When a user query is

submitted through the query interface, the site accesses its backend database for relevant data and returns the results to the user. Specifically, the query interface of the Web database usually contains multiple input elements, each of which may be associated with a schema attribute of the backend database. Data objects that the Web database returns to users are usually semi-structured, as their attribute values are encoded into HTML tags

III. PRELIMINARIES AND BACKGROUND

A. Problem Definition

To our knowledge, solve the online duplicate detection problem for the Web database scenario where query results are generated on-the-fly.

The importance of each individual field needs to be considered, which may vary widely from query to query.

This makes existing work based on hand coded rules or offline learning inappropriate. This is also the first work that takes advantage of the dissimilarity among records from the same Web database for record matching.

Most existing work requires human-labelled training data (positive, negative, or both), which places a heavy burden on users. Our focus is on Web databases from the same domain, i.e., Web databases that provide the same type of records in response to user queries. Suppose there are s records in data source A and there are t records in data source B, with each record having a set of fields/attributes. Each of the t records in data source B can potentially be a duplicate of each of the s records in data source A.

B. Ontology Web Ontology Language

The set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. The use of a web-based knowledge representation format enables developers to discover sharable domain models and knowledge bases from internal and external repositories.

The string transformations considered include Initial (one token is equal to the first character of the other), Substring (one token is a substring of the other), and Abbreviation (characters in one token are a subset of the characters in the other token). After the k transformation steps are identified, the similarity score between Sa and Sb is calculated as the cosine similarity between the token vectors Sa and Sb with each token associated with a TF-IDF weight, where TF means the frequency of a token in a string and IDF the inverse of the number of strings that contain the token in the field. The TF value of a token is usually 1 and IDF will reduce the weight for tokens, such as the tokens in the user query that appear in multiple strings in a field.

C. Unsupervised Duplicate Refresh Elimination

Existing supervised record matching methods require users to provide training data and therefore cannot be applied for web databases where query results are generated on-the-fly. A pre learned query method using training examples from previous query may fail on the results of a new query. Existing supervised method does not be effective to generate a new query. A new record matching method named Unsupervised Duplicate Refresh Elimination (UDRE) is proposed for identifying and eliminating duplicates among refresh records in dynamic query results. The idea of this research is to adjust the weights classifier record fields in calculating similarities among refresh records.

Three classifiers namely weight component similarity summing time bound classifier, support vector machine classifier and threshold-based support vector machine classifier are iteratively employed with UDRE where the first classifier utilizes the weights concentrated on string similarity measures for comparing records from different data sources.

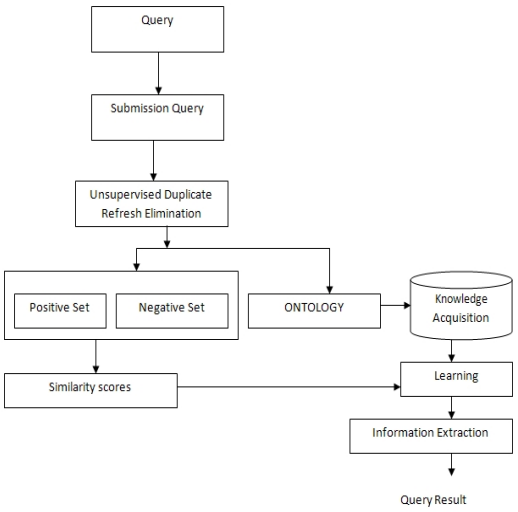

Figure1: Unsupervised Duplicate Refresh Elimination result

The Refresh Elimination identification is the classified weighted component similarity database. It is used to separate the user requested resource and the unrelated resources. It contains the number of elements i.e., Keywords are stored. Based on the elements the user requested resource are identified and displayed to the users. That means the result data table will be displayed.

We also present Active relevant log Detection, ARLD, which, for given query, can effectively identify recent active log relevant from the query result records in multiple Web databases. Starting from the Active Log duplicate set, we use two cooperating classifiers, a weighted component similarity summing classifier and an SVM classifier, to iteratively identify recent active log duplicates in the query results from multiple Web

databases. Our approach is comparable to previous work UDD that identifies duplicates from the query results of multiple Web databases. The describable relationships among them are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. The use of a web-based knowledge representation format enables developers to discover sharable domain models and knowledge bases from internal and external repositories

In a normalized database there is less possibility of duplicates existing. But in the case of relevant Web databases this may not be the case. Also when we have multiple databases there is always a chance that the same or similar records exist that refers to the same real-world entity. This step involves in identifying all the similar refresh records that match the query. There might be a challenge in streamlining the data i.e. each database might have different structure and layout. A standardized structure is used to store the records from these multiple databases.

The recent record base matching query results weight component similarity in relevant weight component similarity applications. These applications are to serve as a central repository of refresh data in an organization and the main task is to filter duplicate records that refer to the same real-world entity. Refresh data is used to manage recent transactional data in an organization for example customer or material master information. When there is a need to create for example a new customer. This process streamlines information and will avoid multiple accounts being created which refer to the same entity.

IV. CONCLUSION

In this paper concentrated on the development of an Unsupervised Duplicate Refresh Elimination (UDRE) is proposed for identifying and eliminating duplicates among refresh records in dynamic query results for developing applications that use Web databases. Web based retrieval system in which records to match are greatly query-dependent, a pre-trained approach is not appropriate as the set of records in response to a query is a biased subset of the full data set. With exponential growth of data, duplicate detection is an important problem that needs more attention, using an UDD algorithm that learns to identify duplicate records has some advantages over offline/supervised learning methods. The idea of this research is to adjust the weights classifier record fields in calculating similarities among refresh records. Three classifiers namely weight component similarity summing time bound classifier, support vector machine classifier and threshold-based support vector machine classifier are iteratively employed with UDRE where the first classifier utilizes the weights concentrated on string similarity measures for comparing records from different data sources.

REFERENCES

[1] Weifeng Su, Jiying Wang, and Federick H.Lochovsky, " Record Matching over Query Results from Multiple Web Databases" IEEE transactions on Knowledge and Data Engineering, vol. 22, N0.4,2010.

[2] Topic and duplicate detection in QA data TREC QA track can be found at http: //trec.nist.gov/data/qamain.html

[3] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and Efficient Fuzzy Match for Online Data Cleaning," Proc. ACM SIGMOD, pp. 313-324, 2003.

[4] L. Gravano, P.G. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava, "Approximate String Joins in a Database (Almost) for Free," Proc. 27th Int'l Conf. Very Large Data Bases, pp. 491-500, 2001.

[5] S. Tejada, C.A. Knoblock, and S. Minton, "Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification," Proc. ACM SIGKDD, pp. 350-359, 2002

[6] R. Ananthakrishna, S. Chaudhuri, and V. Ganti, Eliminating Fuzzy Duplicates in Data Warehouses. VLDB, pages 586-597, 2006

[7] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate Record Detection: A Survey, IEEE TKDE, 19(1):1-16, 2007

[8] Near-Duplicate Detection by Instance-level Constrained Clustering SIGIR'06, August 6–11, 2006, Seattle, Washington, USA

[9] T. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents. In Journal of the American Society or Information Science and Technology, Vol 54, I 3, 2003

