UDD for Multiple Web Database

1Bhadakwan Shrikant, 2Jagadale Prashant, 3Bhanarkar Prasad, 4Sandeep Bhatia
1, 2, 3, 4 Shardchandra Pawar College of Engineering
Pune University, Pune

Abstract
UDD is algorithm that detects duplicates from unsupervised data in Web Database. UDD is more effective for Record matching. Some web database uses supervised method that is very complex to apply big Web database as well as time consuming process. UDD has solution of this problem and also is suitable for dynamically generated data on web. When user puts query on web, data is gathering from multiple web databases and filter by UDD to detect duplicates results to show unique result set. For that we use WCSS classifier that performs string comparisons to identify duplicates. This Paper Shows that UDD works good for Multiple Web database where it is not possible by supervised method.

Keywords - Record Matching, WCSS classifier, Multiple Web Database, Duplicate Detection, DD Algorithm

1 Introduction

Now days ,Web database is a main part of web service that store huge amount of data .Web Database involves various category of data like text, images, documents etc. This database is used to generate dynamically web page. End user have don’t control of such pages. If query is send by user to web database, then it returns some data related to query. But user may get repeated data from multiple web databases. (i.e. same Book from multiple web database). The Web Database contains more amount of redundancy in that data. The main work in that Web database is detecting the duplicate entries by observing and exact matching fields from various data sources. In this paper we focuses on removing duplicates data that returned by Database. It can be done by comparing results on return from database. It is more important step to compare data from different sources. Fig. a shows the result that return from two different online bookstores, amazon.com and books.rediff.com in response same query “asp.net” over title page. It can be seen that result 3 in fig.1 and result 1 in fig b refer to the same book, since they have same title although their price is different.
To overcome this problem, Unsupervised Duplicate Detection (UDD) algorithm is used.
UDD is record matching technique to detect duplicates among records from different web database. The primary aim of this paper is detection of duplicate, like two or more books having same name, ISBN no or author. To identify duplicates from multiple web databases is based on WCSS.

In the WCSS classifier, weight of data is calculate between records, if two or more records having same weight are consider as duplicate and vector table are used to sort duplicate data set and non-duplicate data set. These can be used for only multiple web database of single domain and provides limited result.

UDD attentive on methods for regulate the weights of the entry fields in finding the resemblance between two records. Two entries are supposed as duplicates while they are exact similar on their entry fields. In the Figure 1 & 2 two figures shows the same book name as Asp.net MVC 4 book and its author name is also same as below. The method that deal with duplicate identification can be mostly classified as those necessary training data and those that can purpose without a predefined training data .In this paper we have idea to explore unsupervised methods and discover a method that can role with negligible supervision. The evidence of using search engine example is to focus the essential for an algorithm that can handle huge amounts of data and be able to find a unique set that is more relevant to the user query.

![Figure 1: Query Result from Amazon.com](image1)

![Figure 2: Query Result From books.rediff.com](image2)
By comparing the above Figures, we can find out the duplicate results. For these type of duplications we avoid by using the Unsupervised Duplicate Detection Algorithm.

2 Related Work
2.1 Record Matching
Record matching, which identify duplicates record in response to user query. It represents identified records as same real-world entity.

1. Problem Definition
In this we are calculating the duplicates and non-duplicates results by using WCSS classifier and in unsupervised duplicate detection algorithm we finding out the unique result.

2. Element Detection
By using the Record matching we compare user query with all the records in the site1 and site2 Web Databases i.e here we considering Amazon.com and book.rediffmail.com respectively as shown in the following table1.

<table>
<thead>
<tr>
<th>Book Name</th>
<th>Author</th>
<th>Price</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professional Asp.net MVC 4</td>
<td>Jon Galloway</td>
<td>399</td>
<td>Amazon.com</td>
</tr>
<tr>
<td>Beginning ASP.NET 4.5 in C# and VB</td>
<td>Imar Spaanjaars</td>
<td>483.90</td>
<td>Amazon.com</td>
</tr>
<tr>
<td>ASP.NET 4 Unleashed</td>
<td>Stephen</td>
<td>680</td>
<td>books.rediff.com</td>
</tr>
<tr>
<td>Professional Asp.net MVC 4</td>
<td>Jon Galloway</td>
<td>520</td>
<td>books.rediff.com</td>
</tr>
</tbody>
</table>

Here we perform the exact matching as shown above table. Unsupervised Duplicate Elimination (UDE) removes the problems of duplications.

3 Matching Records
Highly Similar Records are considered as same entity. Identifying same entities from web databases i.e. Duplicate Records. Compared to these above works, UDD algorithm developed for Web database scenario where entries match of single type with multiple fields. Our work focus based on assigning weight to every entry rather than the Similarity function. Moreover, Our work is also casually connected to classification issues about single class of training data, to find the identical data to the given dataset. Here we considered the totally same entry say as Equality ,and assign weight is 1.If entry is half same say as stemming and weight value is 0.5 and entry is totally unmatched say as drop and weight value is 0.

3 Proposed System
New filtering methodologies are designed by exploiting the ordering information that are integrated into the existing methods and drastically reduce the user sizes and hence improve the quality then new technique Unsupervised Duplicate Detection (UDD) was developed for the specific record matching.
problem of identifying duplicates among records in query results from multiple Web databases. A new exact similarity join algorithms is introduced with application to near duplicate detection. The procedures are used for adjusting the weights of the record fields in manipulating the similarity between two entries. Two entries are considered as identical if they are “similar enough” on their fields. Various entries may need to be assigned various importance weights in an accommodative and changing manner. Finally an efficient algorithm is developed, several optimizations that significantly reduce the overall computation time using real data set and synthetic data set. The important goal is on the Web database from the single domain. Various users enter the dissimilar queries in the search box of search engine. An initial solution to this difficulty is that we can learn a Weighted Component Similarity Summing Classifier (WCSS).

3.1 Weighted Component Similarity Summing Classifier

WCSS Classifier is utilized to detect some duplicate vectors when there are no positive examples available. This dissertation builds up on the idea of UDD and is an attempt to enhance the algorithm by introducing an additional classifier known as “Blocking Classifier”. The Weighted Component Similarity Summing (WCSS) Classifier where the importance of the fields is determined and duplicates are identified without any training. The idea for this classifier is to calculate the similarity between pair of records by doing a field to field comparison. Duplication methods in UDD are used which attention on Web databases from the similar domain. Every function measures the equality of selected attributes with other record fields and assigns a equality value for each field. The clustering techniques have been selected to group the fields based on the similarity values. This matching and non-matching pairs is used for clustering and to eliminate the duplicates. The rule based duplicate detection and elimination approach is used for detecting and eliminating the records.

3.1.1 Weighted Component Assignment

In this WCSS Classifier assign weight to each component to shows the adjuvant field. The Weight Assign contains as follow:

1. **Equality**: By checking the each field with every field in the view table. If that field match exactly to another field exactly then we assign weight as 1 known as Equality.
2. **Stemming**: Similarly, we check the field with another field, if that field match halfy we assign weight as 0.5 say as Stemming.
3. **Drop**: The field in the view table not match totally then we assign weight as 0 and this known as Drop.

Suppose In the view table number of entries of Asp.net and C# book and check s all entries to every entry and we assign weight in the following figure.

![Figure 3: Weight Assign in WCSS Classifier](image)

4 Unsupervised Duplicate Detection (UDD)
Duplicate records do not allocate a common key and they contain errors that create duplicate matching a various task. Faults are presented as the result of record errors, imperfect information, absence of standard formats or any grouping of these factors. we present a detailed analysis of the history on duplicate record detection. We cover similarity metrics that are normally used to detect similar field entries, and we present an general set of duplicate detection algorithms that can detect duplicate records in a database. We here used the technique for Unsupervised Duplicate Detection.

4.1 Unsupervised Learning
As we stated earlier, the comparison space involves of comparison vectors which contain information about the differences between fields in a couple of records. Except some data exists about which comparison vectors correspond to which variety (match, non-match, or possible-match), the tagging of the comparison vectors in the training data set should be done physically. One way to escape manual tagging of the comparison vectors is to use classification algorithms, and group composed similar comparison vectors. The hint behind most unsupervised learning methods for duplicate detection is that similar comparison vectors look like to the same class.

4.2 Overview Of UDD Algorithm
The UDD is an Unsupervised way to detect the duplicates fields in the dataset. The key component of the system is the component that has the UDD algorithm. we appearance at emerging an algorithm that can train itself and assistance in identifying duplicates. This algorithm consists of a component that analyzes the similarity vectors of a selection of dataset, allocating weights to the selected vectors. An native solution to this difficulty is that we can study a classifier from N and practice the study classifier to classify P. While there are some works based on studying from only positive (or negative) samples, to our awareness all works in the history assume that the positive (or negative) samples are all correct. The N may contain a tiny set of incorrect negative examples.

![Figure 4: Duplicate vector detection process.](image-url)

We recommend a method that detecting duplicate vectors in P repetitively, in a way like .The various from these two mechanisms, in which only one classifier is utilized during the repetition, we service two classifiers in each repetition that collaborate to detect duplicate vectors since P. Two classifiers are utilized. detected positive instances are not operative enough to reinstruct the classifier to get a
more correct premise, although two classifiers with various characteristics may develop various sets of positive occurrences. Thus, the two classifiers can help since each other by taking importance of duplicate vectors detected by the further classifier.

![Figure 5: UDD Algorithm](http://www.ijifr.com/searchjournal.aspx)

### Figure 5: UDD Algorithm

<table>
<thead>
<tr>
<th>Input:</th>
<th>Potential duplicate vector set $P$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-duplicate vector set $N$</td>
</tr>
<tr>
<td>Output:</td>
<td>Duplicate vector set $D$</td>
</tr>
<tr>
<td></td>
<td>$C_1$: a classification algorithm with adjustable parameters $W$ that identifies duplicate vector pairs from $P$</td>
</tr>
<tr>
<td></td>
<td>$C_2$: a supervised classifier, e.g., SVM</td>
</tr>
</tbody>
</table>

**Algorithm:**
1. $D = \emptyset$
2. Set the parameters $W$ of $C_1$ according to $N$
3. Use $C_1$ to get a set of duplicate vector pairs $d_1$ from $P$
4. Use $C_1$ to get a set of duplicate vector pairs $\ell$ from $N$
5. $P = P - d_1$
6. While $|d_1| \neq 0$
   7. $N' = N - \ell$
   8. $D = D + d_1 + \ell$
   9. Train $C_2$ using $D$ and $N'$
10. Classify $P$ using $C_2$ and get a set of newly identified duplicate vector pairs $d_2$
11. $P = P - d_2$
12. $D = D + d_2$
13. Adjust the parameters $W$ of $C_1$ according to $N'$ and $D$
14. Use $C_1$ to get a new set of duplicate vector pairs $d_1$ from $P$
15. Use $C_1$ to get a new set of duplicate vector pairs $\ell$ from $N$
16. $N = N'$
17. Return $D$

#### 4.3 Evaluation Metric

The actually correctness of the duplicate dataset can be check by using precision, recall and $f$-measure to evaluate the performance of the algorithm, which describes as follow:

1. **Recall:** Recall means the percent of positive entries of duplicates.

   $\text{Recall} = \frac{\text{Correctly Identified Duplicate Pair}}{\text{True Duplicate Pair}}$

2. **Precision:** Precision means the what percent of positive entries are correct

   $\text{Precision} = \frac{\text{Correctly Identified Duplicate Pair}}{\text{Total No of Duplicate Pair}}$

3. **$f$-measure:** $f$-measure of the system is defined as the weighted harmonic mean of its precision and recall

   $\text{f-Measure} = \frac{2*\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
5 Conclusion

Duplicate detection is an significant stage in data integration and supreme state-of-the-art techniques are based on offline studying techniques, which need training data. Now the Web database condition, where records to match are importantly query-dependent, a pertained method is not relevant as the set of records in every query’s outcomes is a partial subset of the whole data set. To overcome this difficulty, we offered an unsupervised, online method, UDD, for identifying duplicates above the query results of multiple Web databases. Two classifiers, WCSS is utilizing supportively in the merging step of record matching to detect the duplicate couples from all potential duplicate pairs repetitively. Tentative results present that our method is distinguishable to preceding work that requires training instances for detecting duplicates from the query results of multiple Web databases.

6. Acknowledgment

We wish to thank our HOD, Mr.A.D.Jadhav and to my guide Mr. S.K. Waman to inspire a lot for writing & providing valuable guidance in writing this paper.

7. References


