Print ISSN: 2249 – 3492, Online ISSN: 2249 – 3506

International Journal of Research in Management, Science and Technology

Issue - 13, Vol-07, pp. 57-64, Jan-Jun 2017

CLEAR International Journal of Research in Management, Science and Technology

RESEARCH ARTICLE

DUPLICATE ELIMINATION IN BIBLIOGRAPHIC DATABASES BY ACTIVE LEARNING APPROACH

[1]S.Meenakshi Sundaram,.[2]U.Revathy, [3]S.Pradeep

[1]Associate Professor - Department of Computer Science and Engineering, Aringer Anna College of Engineering and Technology, Dindigul,India, E-mail bosemeena@gmail.com
[2]Assistant professor- Department of Computer Science, Government Arts College for women, Sivagangai, India, E- mail urevathymeena@gmail.com
[3]Assistant professor- Department of Computer Science, Pannai College of Engineering and Technology, Sivagangai, India, E-mail csspradeep@gmail.com

---

Article History:

Received 12th, May 2017

Received in revised form 25th May 2017

Accepted 15.06.2017

Published on 30.06.2017

Keywords: Active learning, Automatic wrapper generation, Data extraction, Data record alignment, Duplicate elimination, Bibliographic Databases

Corresponding Author:

Mr. S.Pradeep

Email: csspradeep@gmail.com

ABSTRACT

Various data quality problems arise when data is integrated from different heterogeneous sources into a data warehouse. Records duplication is one of the prominent problems in data warehouse. This research focuses on the identification of fully as well as partially duplicated records. In real time applications, identification of records that represent the same real-world entity is a major challenge to be solved. Detection and removal of duplicate records that relate to the same entity within one dataset is an important task in data preprocessing. We present our design of a learning-based deduplication system that uses a novel method of interactively discovering challenging training pairs using active learning. Our experiments on real-life datasets show that active learning significantly reduces the number of instances needed to achieve high accuracy. We investigate various design issues that arise in building a system to provide interactive response, fast convergence, and interpretable output. This method automatically extracts data from query result pages by first identifying and segmenting the query result records in the query result pages and then aligning into a table, in which the data values from the same attribute are put into the same column.

Print ISSN: 2249 – 3492, Online ISSN: 2249 – 3506

International Journal of Research in Management, Science and Technology

1. INTRODUCTION

Data warehouses store large amount of data that is used in analysis and decision making process. Data is integrated from various heterogeneous sources. In heterogeneous sources, data has different formats. Data is noisy in nature and needs to be cleaned in a data warehouse. Data cleaning is a process of detecting and correcting incorrect, redundant and missing values. This process also checks the format, completeness and violation of business rules in data. Data cleaning process is used to improve the quality of data. Some data quality problems occur because of data entry operator errors such as spelling mistakes, missing integrity constraints, mismatch field, noise or contradicting entry, null values, misuse of abbreviations and duplicated records. Data quality measures the accuracy, integrity, completeness, validity, consistency and redundancy aspects of data.

In a data warehouse, data cleaning has a vital role. If the quality of data is not good, the strategic decisions taken on the basis of that data may not be good. Records duplication is one of the major issues in data quality. It is the representation of the same real world object more than once in the same table. It is necessary to eliminate the duplicated records in order to bring consistency and to improve the quality of the data. To identify duplicated records and remove them efficiently, the researchers proposed different techniques in the area of data mining and data warehousing. Records duplication is also known as entity resolution, record linkage or merge purge. Identification and removal of the duplicated records is an important issue in data cleaning which is the subject of this research. We designed a learning based deduplication system that allows automatic construction of the deduplication function by using a novel method of interactively discovering challenging training pairs. Our key insight is to simultaneously build several redundant functions and exploit the disagreement amongst them to discover new kinds of inconsistencies amongst duplicates in the dataset.

Active learning methods also rely on a similar insight for selecting instances for labeling from a large pool of unlabeled instances. Unlike an ordinary learner that trains using a static training set, an active learner actively pick subsets of instances which when labeled will provide the highest information gain to the learner. With this approach the more difficult task of bringing together the potentially confusing record pairs is automated by the learner. The user has to only perform the easy task of labeling the selected pairs of records as duplicate or not. We designed an active learning algorithm that can meet our design goals of interactive response, fast convergence, and high accuracy.

Finally, our system outputs a deduplication function that is easy to interpret and efficient to evaluate when deployed on large record lists. This required evaluating various non-obvious design tradeoffs that arise when using current active learning methods in a practical setting.

2. RELATED WORK

I. Ahmed and A. Aziz [14] discussed different techniques to improve accuracy rate of data quality. They introduced data cleansing framework which consists of attribute selection, token formation, clustering and eliminator functions. The drawback of this technique is that it is based on a token based technique, so that large number of false positive values was introduced.

F. Panse, M. V. Keulen, A. D. Keijzer and N. Ritter [15] proposed the method for duplicate detection in probabilistic data. But probabilistic data does not provide accurate results.

G. Beskales, M. A. Solimon, I. F. Ilyas, S. Ben-David and Y. Kim [16] introduced new probabilistic ETL tool for identification and elimination of duplicated records. The function of this ETL tool is data transformation. Specific threshold is used to identify either duplicated or non-duplicated records.

J. Wang and F. H. Lochovsky [10] presented a novel data extraction method, ODE (Ontology-assisted Data Extraction), which automatically extracts the query result records from the HTML pages. To label attributes it is necessary that the labels appear in the query interfaces or query result pages within a domain. If the query result records are arranged into two or more different formats in the query result pages, then only one format will be identified as the query result section. Finally, the performance of ODE on certain types of query result pages is far from satisfactory.

P. Christen and K. Goiser [8] presented an overview of the issues involved in measuring data linkage and deduplication quality and complexity. It is shown that measures in the space of record pair comparisons can produce deceptive accuracy results. It is

recommended that the quality be measured using the precision-recall or F-measure graphs rather than single numerical values, and that quality measures that include the number of true negative matches should not be used due to their large number in the space of record pair comparisons.

A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios [11] made a thorough analysis of the literature on duplicate record detection. The similarity metrics that are commonly used to detect similar field entries, and an extensive set of duplicate detection algorithms that can detect approximately duplicate records in a database are covered. The lack of standardized, large scale benchmarking data sets can be a big obstacle as it is almost impossible to convincingly compare new techniques with existing ones.

Y. Zhai and B. Liu [13] studied the problem of structured data extraction from arbitrary Web pages. In this paper, a novel and effective technique (called DEPTA) to perform the task of Web data extraction automatically is proposed. This method has the following drawbacks: When an object is very dissimilar to its neighboring objects, DEPTA misses it. This also causes a few identified data records to contain extra information or to miss part of their original data items.

H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu [3] presented a technique for automatically producing wrappers that can be used to extract search result records from dynamically generated result pages returned by search engines. The main problem with this method is its reliance on the tag structure in the query result pages, due to which it suffers from poor results.

K. Simon and G. Lausen [5] addressed the problem of unsupervised Web data extraction using a fully-automatic information extraction tool called ViPER. The tool is able to extract and separate data exhibiting recurring structures out of a single Web page with high accuracy by identifying tandem repeats and using visual context information. However, this technique lacks performance in few datasets.

S. Chaudhuri, V. Ganti, and R. Motwani [9] proposed two novel criteria, compact set and sparse neighborhood, that enable characterization of fuzzy duplicates more accurately than is possible with existing techniques. This method does not deal with blocking strategies.

M. Bilenko and R. J. Mooney [6] presented a framework for improving duplicate detection using trainable measures of textual similarity. Learnable text distance functions for each database field are employed to show that such measures are capable of adapting to the specific notion of similarity that is appropriate for the field's domain. However, this method suffers from overfitting issues in few cases.

3.ACTIVE LEARNING

An active learner starts with a limited labeled and a large unlabeled pool of instances. The labeled set forms the training data for an initial preliminary classifier. The goal is to seek out from the unlabeled pool those instances which when labeled will help strengthen the classifier at the fastest possible rate. What criteria should we use for picking such instances? The initial classifier will be sure about its predictions on some unlabeled instances but unsure on most others. The unsure instances are those that fall in the classifier's confusion region. This confusion region is large when the training data is small. The classifier can perhaps reduce its confusion by seeking predictions on these uncertain instances. This intuition forms the basis for one major criteria of active learning, namely, selecting instances about which the classifier(s) built on the current training set is most uncertain. There are three primary inputs to the system shown in Fig. 2:

1. Database of records (D) The original set D of records in which duplicates need to be detected. The data has d attributes a1…. ad, each of which could be textual or numeric. The goal of the system is to find the subset of pairs in the cross-product D×D that can be labeled as duplicates.
2. Initial training pairs (L) An optional small(less than ten) seed L of training records arranged in pairs of duplicates or non-duplicates.
3. Similarity functions (F) A set F of n_f functions each of which computes a similarity match between two records r1, r2 based on any subset of d attributes. Examples of such functions are edit-distance, soundex, abbreviation match on text fields, and absolute difference for integer fields.

Print ISSN: 2249 – 3492, Online ISSN: 2249 – 3506

International Journal of Research in Management, Science and Technology

1. Input: L, D, F.

2. Create pairs L_p from the labeled data L and F.

3. Create pairs D_p from the unlabeled data D and F.
4. Initial training set T = L_p
5. Loop until user satisfaction

- Train classifier C using T.
- Use C to select a set S of n instances from D_p for labeling.
- If S is empty, exit loop.
- Collect user feedback on the labels of S.
- Add S to T and remove S from D_p.

6. Output classifier C

Fig. 1 Active Learning Algorithm

Many of the common functions could be inbuilt and added by default based on the data type. However, it is impossible to totally obviate an expert's domain knowledge in designing specific matching functions. These functions can be coded in the native language of the system (C++ in our case) and loaded dynamically. The functions in the set can be highly redundant and unrelated to each other because finally our automated learner will perform the nontrivial task of finding the right way of combining them to get the final deduplication function.

A rough outline of the main steps is given in Fig. 1. The first step is to map the initial training records in L into a pair format via a mapper module. The mapper module takes as input a pair of records r1, r2, computes the n_f similarity functions F and returns the result as a new record with nf attributes. For each duplicate pair we assign a class-label of "1" and for all the other pairs in L× L we assign a class label of "0". At the end of this step we get a mapped training dataset Lp. These Lp instances are used to initialize the learning component of the system.

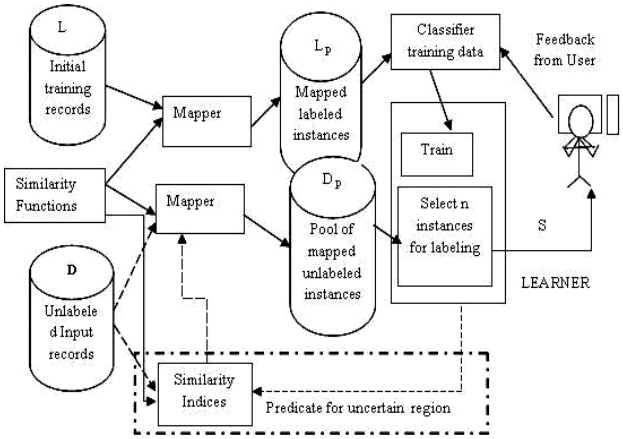


Fig. 2 Overall Architecture of Active Learning Approach

The next step is to map the unlabeled record list D. The mapper is invoked on each pair of records in D ×D to generate an unlabeled list of mapped records Dp. If the size of D is large the quadratic size of the cross-product could be intolerable. We next describe the interactive active learning session on Dp with the user as the tutor. The learner chooses from the set Dp a subset S of n (a user-configurable parameter, typically less than five) instances that it would most benefit from labeling. The user is shown the set of instances S along with the current prediction of the learner. The user corrects any mistakes of the learner. The newly labeled instances in S are added to the training dataset Lp and the active learner is retrained. The user can inspect the trained classifier and/or evaluate its performance on a known test dataset. If (s)he is not happy with the learner trained so far, the active learner can select another set of n instances. This process continues in a loop until the user is happy with the learnt model. In each iteration, the user aids the learner by providing new labeled data. A useful side effect of the user inspecting the model's prediction at each iteration is that, he can discover newer sources of discrepancies and errors in the data and decide to modify his similarity functions or add new ones. The output of our system is a deduplication function I that when given a new list of records A can identify which subset of pairs in the cross-product A ×A are duplicates.

Print ISSN: 2249 – 3492, Online ISSN: 2249 – 3506

International Journal of Research in Management, Science and Technology

4. EXPERIMENTS

4.1 Data Sets

We now present overall evaluation figures for our chosen active learning approach. Our experiments were on the following two datasets:

The Bibliography dataset consists of citation entries obtained from CiteSeer by searching on the last names of the 100 most frequently referred authors. The data consisted of 254 citations, 54 of which were found duplicates (after careful manual searching). In the pair format, this led to (254×253)/2= 32131 instances of which only 169 were duplicates - that is, only 0.5% of the instances were of the positive class. The raw data had no underlying structure. We segmented the text record into five fields namely, author, title, year, page number and rest using CiteSeer's scripts.

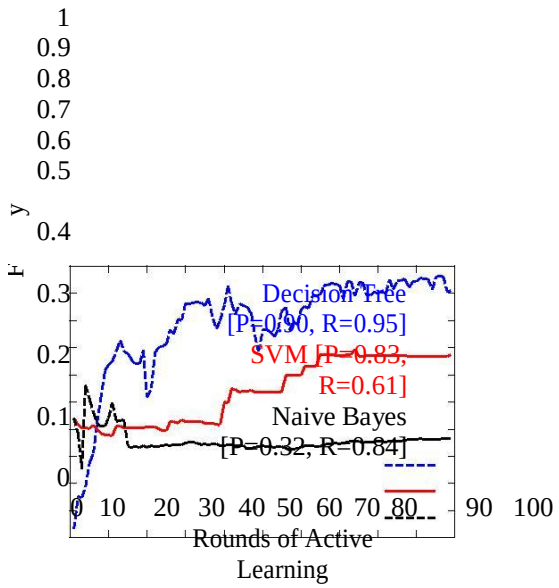
1. Input: L_p: current training data, N number of committees, D_p unlabeled instances
2. Train N classifiers C_1, C2… CN on L_p by randomizing the choice of the parameters for all but the first classifier.
3. For each unlabeled instance x in D_p,
   (a) Find prediction y_1… y_N from the N members.
   (b) Compute uncertainty U(x) as the entropy of the above N predictions.
4. Return n instances by (weighted) sampling on the instances with the weight as U(x).

Fig 3: Algorithm used by active learning for selecting n instances for labeling

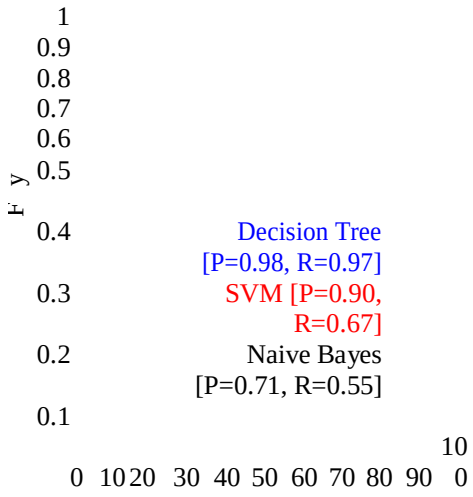
The Address dataset consists of names and addresses of customers with the local telephone company. The data had ten attributes. The six address fields did not follow any meaningful breakup of the address. We had 300 records, 98 of which were found duplicates (again by manual search). In the pair format, this led to (300×299)/2 = 44850 instances of which 105 were duplicates – a skewness of 0.25%.

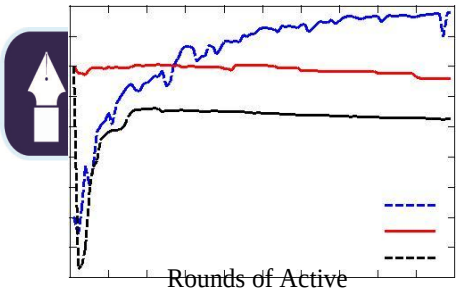
4.2 Results and Discussions

The final algorithm used by our system for picking the n instances for labeling is given in Fig. 3. We used the following three classification methods: C4.5 decision tree classifier, MLC++'s naive Bayes classifier, and SVMTorch Support Vector Machine classifier (SVM). Our experiments were performed on a three processor Pentium III server running Linux redhat 7.0 with 512 MB of RAM. All our experiments were obtained by averaging over ten runs with different seeds of a random number generator that gets deployed in different stages of our algorithm. In Fig. 4 we plot the performance of active learning under three different classification methods.



(a) Bibliography data





(b) Address data

Fig. 4. Comparing performance of different classification methods with active learning approach.

These graphs show that decision trees provide the best F accuracy overall. In the legend of Fig. 4 we show the precision and recall values at the last round of active learning. D-trees dominate SVMs which in turn dominate NB in both the precision and recall values. However, D-trees show a much larger fluctuation in accuracy in the initial stages. This is to be expected because decision trees are known to be unstable classifiers. For the address dataset, SVMs are better in the initial stages of active learning when the training data is small but they loose out later. This does not imply that for a fixed training set SVMs would be worse than D-trees. In these graphs we are evaluating a classifier both on its capability to return meaningful uncertainty values and its overall accuracy. SVMs are known to excel on accuracy but the uncertainty value measured as distance from SVM separator is perhaps not too meaningful. D-trees turn out to be better in the combined metric. This is good news because D-trees also offer other advantages of interpretability and indexability.

4.3 Comparing active learning with random selection

We evaluate the overall performance of active learning by comparing its speed of convergence to the peak accuracy with that of a random selection of the same number of instances in Fig. 5. The graphs show three lines, one each for active learning, random, and optimal selection. We will discuss the comparison of active learning with random first. For both datasets, active learning shows clear superiority over random selection. Within just 100 of the more than 30,000 instances available, active learning is able to achieve a peak accuracy of 97% for the bibliography and 98% for the address dataset. The accuracy does not improve beyond these first 100 instances. The same number of instances selected randomly, achieve accuracy of just 30% and 50% respectively for the bibliography and address datasets. In fact, to achieve even 90% of the peak accuracy random selection needs 5600 instances for the address data and 2700 instances for the bibliography data.

Another interesting observation from these experiments is that in the first 100 selected instances duplicates form 44% of the total for both data sets — a jump from the less than 0.5% fraction duplicates in the original unlabeled pool. Does this mean that the primary gain of active learning is due to correcting the extreme skewness in the original data? Or, are the particular sets of instances important? We performed a second experiment by randomly selecting 100 instances but this time keeping the number of duplicates the same as after active learning. This yielded an average accuracy of only 40% on the bibliography data and 31% on the address dataset.

These numbers are important. They confirm our original intuition that manually collecting large number of duplicates will not achieve high accuracy unless proper care is taken in selecting a confusing enough set of non-duplicates to go with it. This is hard not only because the number of non-duplicates is large but also because it is not easy to know what non-duplicate would be misclassified as a duplicate with an existing training set.

4.4 Comparison with optimal selection

Another important question is how close our active selection method is to some absolute best method. We designed an optimal method that knows the labels of all instances in our unlabeled set D_p through an oracle. At each round of active learning, it then picks one instance (n = 1) as follows:

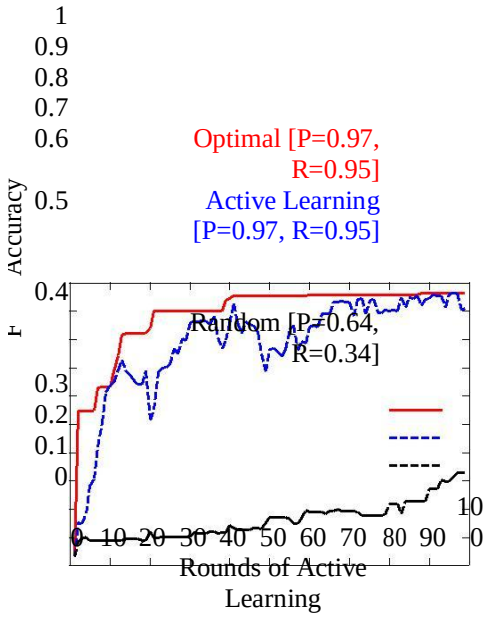
1. For each instance x in D_p
   (a) Add x with its correct label to the current training data T and train a classifier C_x.
   (b) Compute accuracy a_x of C_x in predicting class labels of instances in D_p-x
2. Pick the instance x for which accuracy a_x was the highest.

This is the best algorithm one can design in the one-instance-at-a-time category of algorithms. This does not guarantee to give us the best subset of k instances for a fixed training size k, it just ensures optimality at

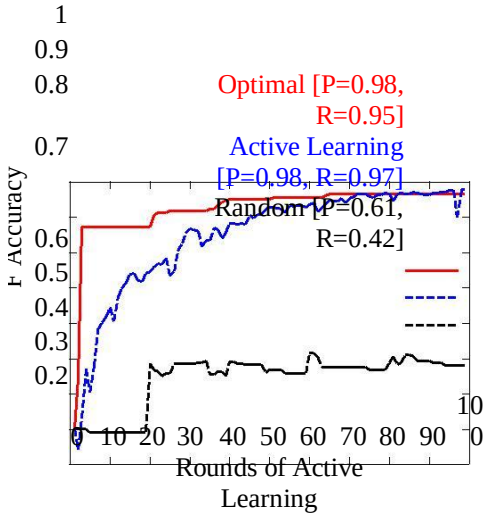
Print ISSN: 2249 – 3492, Online ISSN: 2249 – 3506

International Journal of Research in Management, Science and Technology

each step where we pick one instance at a time. In Fig. 5, we plot the accuracy of this optimal approach. For both datasets we notice that our chosen criteria of instance selection is indeed very close to the accuracy provided by the optimal approach that unrealistically assumes that all labels are known. One major difference is that optimal is smooth and monotonic whereas with active .



(a) Bibliography data



(b) Address data

Fig. 5. Speed of convergence with active learning, random selection and optimal selection

learning the accuracy fluctuates. Along both these metrics separately active learning is close to the optimal approach. The problem of deduplication has long been relevant for library cataloging concentrate on hand-coding deduplication functions for the bibliography domain. The deduplication problem is also of interest to the statistics community in organizations like Census Bureau. Much effort has been spent in designing domain-specific similarity functions for Census datasets. The learning approach is restricted to one-shot conventional classification using logistics regression and naive Bayes. Some of our similarity functions have been inspired by this literature. However, none of these systems address the difficulty of collecting a good covering set of training instances to start with.

Recently, there has been renewed interest in the database community on the data cleaning problem comprising several aspects, including, data segmentation, deduplication, outlier detection, standardization and schema mapping. For the specific problem of deduplication, most recent work has concentrated on the performance aspects assuming that the deduplication function is input by the user.

Our approach of learning the deduplication function interactively bears resemblance to interactive relevance feedback used to refine queries over text and multimedia content. In relevance feedback the goal is to learn a relevance function which in most cases boils down to learning appropriate weights of a weighted distance function. The key difference between relevance feedback and active learning is the type of examples shown to the user for collecting feedback. In most relevance feedback systems the user is shown the top few most relevant answers in each round whereas in active learning fast convergence rests on showing the user the most uncertain answers. Active learning has been applied in several domains in the past, including, text classification and information extraction. We believe ours is one of the first attempt at using active learning to solving a large-scale, practically motivated problem.

5. CONCLUSION AND FUTURE WORK

Deduplication, a key operation in integrating data from multiple sources, is a time-consuming, labor-

intensive and domain-specific operation. Active learning is a novel approach to easing this task by limiting the manual effort to input simple, domain-specific attribute similarity functions and interactively labeling a small number of record pairs. We presented a careful evaluation of a number of non-obvious design tradeoffs to ensure that the active learning process is practical, effective and can provide interactive response to the user. The final deduplication function is designed to be easy-to-interpret and efficient to apply on large datasets.
We find that active learning requires one to two orders of magnitude fewer pairs to be labeled than random selection.

Our experiments show that starting from a highly skewed unlabeled pool with less than 0.5% duplicates, we are surprisingly able to selectively sample 100-fold more duplicates than non-duplicates, making the skew 50%. Also, the specific sets of non-duplicates that we pick are important. If the same numbers of non-duplicates are picked without active selection our accuracy drops to half. Finally, we find that our chosen approach is close to an optimal approach.

Future work include more extensive running time evaluation, design of better similarity indices, and aiding users in designing good attribute similarity functions.

7. REFERENCES

[1] Weifeng Su, Jiying Wang, and Fredrick H.Lochovsky, "Record Matching Over Query Results from Multiple Web Databases", IEEE Trans. Knowledge and Data Eng., vol. 22, no. 4, April 2010.

[2] B. Liu and Y. Zhai, "NET - A System for Extracting Web Data from Flat and Nested Data Records," Proc. Sixth Int'l Conf. Web Information Systems Eng., pp. 487-495, 2005.

[3] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc. 14th World Wide Web Conf., pp. 66-75, 2005.

[4] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf., pp. 187-196, 2003.

[5] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," Proc. 14th ACM Int'l Conf. Information and Knowledge Management, pp. 381-388, 2005.

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

[7] M.K. Bergman, "The Deep Web: Surfacing Hidden Value," White Paper, BrightPlanet Corporation, http://www.brightplanet.com/resources/details/deepweb.html, 2001.

[8] P. Christen and K. Goiser, "Quality and Complexity Measures for Data Linkage and Deduplication," Quality Measures in Data Mining, F. Guillet and H. Hamilton, eds., vol. 43, pp. 127-151, Springer, 2007.

[9] S. Chaudhuri, V. Ganti, and R. Motwani, "Robust Identification of Fuzzy Duplicates," Proc. 21st IEEE Int'l Conf. Data Eng., pp. 865-876, 2005.

[10] W. Su, J. Wang, and F.H. Lochovsky, "ODE: Ontology-Assisted Data Extraction," ACM Trans. Database Systems, vol. 34, no. 2, article 12, p. 35, 2009.

[11] Weifeng Su, Jiying Wang, and Fredrick H.Lochovsky, "Combining Tag and Value Similarity for Data extraction and Alignment", IEEE Trans. Knowledge and Data Eng., vol. 24, no. 7, July 2012.

[12] Y. Zhai and B. Liu, "Structured Data Extraction from the Web Based on Partial Tree Alignment," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.

[13] Ahmed. I, A. Aziz, "Dynamic Approach for Data Scrubbing Process", International Journal on Computer Science and Engineering 2(2): 416-423, 2010.

[14] F. Panse, M.V. Keulen, A.D. Keijzer and N. Ritter," Duplicate detection in probabilistic data", ICDE IEEE workshops, 2010.

[15] Beskales. G, M. A. Solimon, I. F. Ilyas, S. Ben-David and Y. Kim, "ProbClean: A Probabilistic duplicate detection system ", ICDE IEEE conference in 2010.

[16] S. Toney. Cleanup and deduplication of an international bibliographic database. Information Technology and libraries, 11(1):19 - 28, 1992.