Enhancing Data Deduplication methodology of Query results from multiple sources

Mrs. T. Sunitha #1, Mr. Maddineni Lakshmi Sainath #2, Mr. Nandyala Venugopal #3, Mr. Nurubhashu. Kaleshavali #4, Mr. Madala Anil Kumar #5,

#1 Associate Professor, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#2 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#3 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#4 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#5 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)

Abstract: The results from multiple databases compose the deep or hidden Web, which is estimated to contain a much larger amount of high quality, usually structured information and to have a faster growth rate than the static Web. The system that helps users integrate and more importantly, compare the query results returned from multiple Web databases, an important task is to match the different sources' records that refer to the same real-world entity. After removal of the same source duplicates, the assumed non duplicate records from the same source can be used as training examples. Unsupervised Duplicate Detection (UDD) uses two cooperating classifiers, a Weighted Component Similarity Summing (WCSS) classifier and a Support Vector Machine (SVM) classifier that iteratively identifies duplicates in the query results from multiple Web databases. For String Similarity calculation UDD uses any kind of similarity calculation method. Various experiments are conducted on a dataset to verify the effectiveness of the unsupervised algorithm in general and the additional blocking classifier in particular. Blocking is a technique to group data. Normally in order to classify records in a table, a unique hash function is generated for each record and compared with all other records in the table. Records having the same or similar hash value are categorized into groups. These groups are categorized as duplicates and non duplicates.

Index Terms—Record normalization, data quality, data fusion, web data integration, deep web

I. INTRODUCTION

A lot of data has been dumped every day into the World Wide Web (WWW) which makes fetching a desired detail through this web has become a tiresome activity. To recover from this, data mining provides a solution, through mining process which analyzes data and summarizes to useful non-redundant data [1], [2]. This process of data mining helps technologists and also the technology users to reduce costs and increase their profit. Moreover, data mining proves to be one of the best analytical tools to analyze, categorize and summarize data. In real-time applications, major industries are using the data mining technology to their favor. The concept of data mining is used to correlate the internal and external factors. This gives clues to find out the sales impact, customer mindset, economic indicators etc. For instance, Blockbuster entertainment uses data mining to suggest products to their customers based on their video rental history in its database [3]-[5]. Also Walmart uses the point-of-sale transactions data and stores them in its data warehouse. This enables the Walmart to identify the suitable merchandising opportunity and has transformed its supplier relationship. The data mining can be performed with different level of analysis, namely, Artificial Neural Network(ANN), Genetic Algorithms(GA), Decision Trees, Nearest neighbor method, Rule induction, Data Visualization. The technological infrastructure required for the data mining applications are ranging from a small space PC platform to the mainframes. The main parameters that should be considered are the size of the database and the query complexity. Search engines are huge such databases with web page files generated from the existing external sources and are automatically assembled. The search engines can be classified as Individual search engine and Meta searchers here the individual search engine possesses its own database [6] where information was compiled to be accessed by the users [7]. While Meta searchers don't have their own databases but uses different individual search engines and displays the best results. These Meta searchers use either one of the two ways to show their results. (i) After removing the duplicated results from the results collected from various individual search engines [8]-[10], a merged single list of data is presented. (ii) Without removing the duplicate data multiple lists are produced. So this duplication is the undesired property which can be eliminated by the process of de-duplication. This helps to decrease the storage requirement as only distinct data is stored. The redundant data is avoided

by replacing a pointer to the distinct data copy. The disadvantage of duplication [11] is that it affects crawling and relevance of the results. Hence rule induced mining is performed over the URL's in this proposal. Since all the search engines strive to be the popular one based on the closest results that tends to provide. Also the other main parameter to be considered is data duplication.
The demerits if the data duplication is

- High storage space occupied
- Less efficient use of disk space
- Low Recovery Time Objectives (RTO)
- Need of tape backups
- Requires more transmissions

The existing system that is chosen to compare our result is Detecting Near-Duplicates for Web crawling. It uses simhash to address the large query. It also develops the hamming distance problem for both single and multiqueries online. However the existing system has the disadvantages like De-duplication performed after a web page downloads, hence High bandwidth usage during crawling. Costs high, Limit of accuracy our focus in this paper is on efficient and large-scale de-duplication of documents on the WWW. Web pages which have the same content but are referenced by different URLs, are known to cause a host of problems. Crawler resources are wasted in fetching duplicate pages, indexing requires larger storage and relevance of results are diluted for a query [12]-[15]. The problem statement of the proposed work is for a given set of duplicate clusters and their corresponding URLs, Learning Rules from URL strings which can identify duplicates; Utilizing learned Rules for normalizing unseen duplicate URLs into a unique normalized URL Applications such as crawlers can apply these generalized Rules on a given URL to generate a normalized URL. The proposed work involves mining of the crawling logs and the transformational rules are extracted from the clusters of similar pages. The cluster URLs is then normalized [13]. Instead of using every mined rule, the machine learning technique is introduced to de-duplicate web resource with ease.

II. RELATED WORK

Data consolidation is a challenging issue in data integration. The usefulness of data increases when it is linked and fused with other data from numerous sources. The promise of Big Data [14] hinges upon addressing several data integration challenges, such as record linkage at scale, data fusion, and integrating Deep Web. Although much work is conducted on these problems, there is limited work on creating a uniform, standard record from a group of records corresponding to the same real-world entity. We refer to this task as record normalization. Such a record representation, coined normalized record, is important for both front-end and back-end applications [16]. In this paper, we formalize the record normalization problem, present in-depth analysis of normalization granularity levels and of normalization forms. We propose a framework for computing the normalized record. It includes a suit of normalization methods, from naive ones, which use only the information [17] gathered from records themselves, to complex strategies, which globally mine a collection of duplicate records before selecting a value for an attribute of the normalized record.

Record Matching Technique Primarily record matching techniques can be broadly classified into the following

- Character or string based
- Token based
- Phonetic based
- Numeric similarity

Character or String Based Similarity Metrics These set of techniques deal with various ways in comparing strings and finding a similarity metric that can be used to group as well as identify potential duplicates. There are many papers published and algorithms that were developed in analyzing the correct technique for comparing strings and arriving at a differentiating factor in order to measure their similarity [18]. The character-based similarity metrics are designed to handle common typographical errors. A typographical error (often shortened to typo) is a mistake made in the typing process (such as spelling) of printed material. Historically, this referred to mistakes in manual type-setting (typography). The term includes errors due to mechanical failure or slips of the hand or finger, but excludes errors of ignorance, such as spelling errors. String based similarity metrics measure how similar (or dissimilar) two strings are, two strings are deemed identical if they have same characters in the same order.

Token Based Similarity Metrics Character based comparison work effectively in catching typographical errors, but they sometime fall short when comparing a rearranged string that has the same meaning. For example when comparing "Jane Doe" to "Doe, Jane", characters based metrics fail and wrongly classify the two strings being different even though they refer to the same person name. In order to avoid such error token based similarity measures are used, where comparison of two strings is done first by dividing them into a set of tokens (a token is a single word). A common practice is to split the string at white space and form the tokens. Thus in our example the string "Jane Doe" becomes a token array ["Jane", "doe"].

Phonetic Similarity Metrics Strings may be phonetically similar even if they are not similar at character or token level [19]. For example the word "Color" is phonetically similar to "Colour" despite the fact that the string representations are very different. The phonetic similarity metrics try to address such issues.

Numeric Similarity Metrics There are numerous approaches that have been developed for comparing strings. When comparing numerical values the methods are primitive. In most cases when it makes sense to compare numbers, it's a basic comparison and queries can be developed to extract numerical data with ease. There has been continuing research in using cosine similarity [20] and other algorithms in analyzing numerical data. For example data in numbers can be compared with primitive operators like equal, greater than and can used to calculate the difference between two numeric strings.

III. PROPOSED METHODOLOGY

The presented work provides few techniques to prevent data de-duplication in World Wide Web (WWW). The approach mine rules from the URLs. These rules are utilized for data de- duplication. The

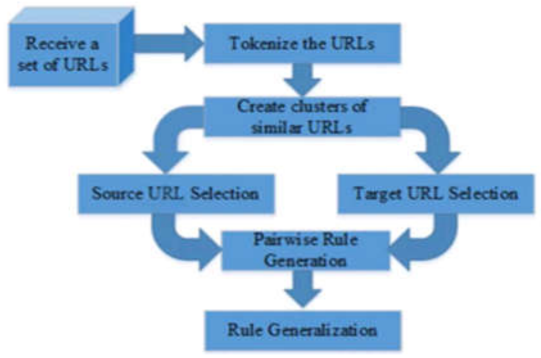

Fig 1. System Architecture of the proposed System

host specific tokens and delimiters are extracted from the URLs. Pair wise rule generation is performed and hence the source URL and target URL is selected. The rules are fine-tuned using generalization technique through machine learning. Target Selection: The target URL is selected from the duplicate clusters of the URLs. This is done by selecting the shortest matching URL in the cluster. Another criterion for target selection is the length of the URL, which must be shorter. The other parameters that could be considered are the number of in links, minimum hop distance etc. Source Selection: The statistical based ranking helps to decide the URL to be chosen as a source. The online page ranker is utilized for this task.

URL Dataset Visualization: The work included two sets of data to achieve data deduplication. The feasibility of data sets, both small and large data sets are used for experimentation are known. These datasets contains either the URLs of many websites or the URLs of many web pages. The pre-requisite to select these small and big data set is that it should contain at least 2 sized duplicate clusters in both data sets. Also it is characterized by the number of hosts, URLs and the duplicate clusters present in them. The collection Mixer is constructed with the data sets containing web pages and clusters. Based on the human judgment, the core content is collected. The stop words are also chosen in order to get proper understanding and good performance for de-duplication.

Tokenization: Basic tokenization is the process of parsing URL to extract tokens. The protocols, hostnames, query arguments and the path components are also extracted from the specified standard delimiters. Firstly, clusters are formed with the URLs in the datasets. Then, anchors are selected from the URL clusters formed in the previous step. The selected anchors are validated and if the anchors are found to be valid, then the child pattern is generated. If the anchors are not valid, they do not generate child pattern. Then, the process of generating tokenized key value pairs and associates them to the original URL in order to generate deep tokenized URLs is known as Deep tokenization. The URL encodings are learnt by a specialized technique that doesn't require any supervision. This process is iterative defined and conducted as per the decision tree generated.

Clustering The process of cluster formation with the URLs is known as Clustering. It is the basic step of module 3 in which the cluster is formed and is produced to the rule generalization module. The URLs which consists of more similarity in the web page content is termed as a duplicate cluster. The rules are generated for all the URL pair present in the duplicate clusters.

Rule Generalization: In the process of rule generalization, one of the clusters is selected from the cluster groups. A key is selected from the previously selected cluster. We know that all keys have information gains, so the key selection is made by studying the maximum information gains possessed by the key. Finally the transformation process of transmitting the source to the target is performed. Generalization is performed by generating a decision tree. This tree is constructed with the selected keys and their branch is formed with the key's matching pair else it is branched out with a wildcard. Number of linear rules generated by the generalization technique. Only after rule generalization, the new values can be accommodated. The decision tree based generalization enabled the work to be error proof and robust [21]. The so generated decision tree follows bottom-Up approach [22]. The rules are used in online mode and hence the memory requirement to store these rules are minimal. The contexts as well as transformation format are generated by the rule generalization process [23]. Thus it provides a compatible contexts as well as target URLs. The feasibility is improved as the iteration counts high. During the initial phase, the frequency is generated for each key [24]. Then context generalization is performed. The generalization is performed over the contexts.

Comparison: This is the ultimate step to present the non-redundant, de-duplicated data for the users. With two data sets in hand, the numbers of de-duplicated data are estimated. The comparison module is presented, where the maximum numbers of duplicates are detected. Thus the matches are identified and are produced for display without any duplicated data. There is a steady improvement as the proposed pair wise rule generalization is an iterative algorithm. The proposed work cross verifies both the data sets so as to provide the optimal results. Through comparison of both the data sets the data set with maximum number of de-duplicated data are known.

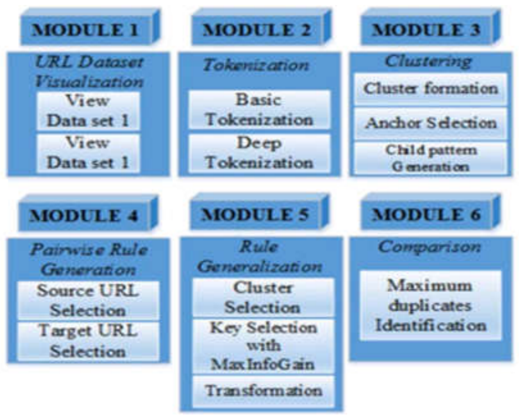

Fig 2: Modules of the proposed System

Data Retrieval The data retrieval module consists of an interface to read the user query along with the actual data retrieval from the database. To make the application interactive and simulate a real world application a search box is given where the user can enter a query which can be for a particular word or run wide open to query all the records in the database for analysis. The data exists in two tables and has the same structure (field names). Element identification is the process of mapping the fields of two tables and reading the information. Dealing with a simple dataset with fixed field names. This may not be the case in a real world application where data can be stored in databases with different names/elements and the process of mapping fields between the data sources is a very crucial step in ensuring the integrity of the application.

Pre-Processing and Blocking the application consists of pre-processing and blocking. In this step generally data is cleansed and parsed into one structure. This step also involves removing special characters from the raw data and converting them to a lower case for accurate comparison. As part of pre-processing, data is sorted and exact matching records are deleted. This is done comparing all the fields (taking each row as one string) and comparing it to others. This ensures the same data doesn't exist and is a basic check that can be done. The next major and crucial step in this module is blocking. Blocking "typically refers to the procedure of subdividing data into a set of mutually exclusive subsets (blocks) under the assumption that no matches occur across different blocks". Normally in order to classify records in a table, a unique hash function is generated for each record and compared with all other records in the table. Records having the same or similar hash value are categorized into groups. The details and techniques that are used for blocking and also look at the efficacy of each of them.

UDD Algorithm The main component of the system is the module that has the UDD algorithm. In this step look at developing an algorithm that can train itself and aid in identifying duplicates. This algorithm consists of a component that calculates the similarity vectors of the selected dataset, assigning weights to the selected vectors and finally using the Support Vector Machine (SVM) to classify the data. The technical details and further information on each of these components is discussed at length later in this section.

Data Presentation The final module consists of presenting the data to the user. The unique data along with statistics is presented to the user. Similarity

Vector Calculation The next step is the similarity vector calculation which holds comparison of two records. Inputs to this process are the potential duplicate dataset and no duplicate dataset (outputs of the blocking classifier). The output of a similarity vector function is a set of attribute similarity scores for each pair of records in the dataset. In this step, the UDD algorithm calculates the similarity of record pairs in both datasets grouped by the blocking classifier. The output of this process serves as input to Weighted Component Similarity Summing (WCSS) Classifier and SVM classifier which are examined in detail in the following sections. For example let's consider two records, one from ZAGAT and another from FODORS.

IV. SYSTEM IMPLEMETNATIONAttribute Weight Assignment

In the WCSS classifier, assign weight to an attribute to indicate its importance. The weights of a field/attribute are given in such a way that the sum of all fields/attributes weights is equal to 1. In non-duplicate vector most of the fields will have small similarity score for all record pairs where as in duplicate vector most of the fields will have large similarity score for all record pairs. In general, WCSS classifier employs duplicate and non-duplicate intuitions for assigning weights. Inputs to this function are the similarity vector of non-duplicate records and duplicate records.

Duplicate Intuition

For duplicate records the similarity between them should be close to 1. For a duplicate vector V12 that is formed by a pair of duplicate records r1 and r2, assign large weights to the fields with large similarity values and small weights to the fields with small similarity values. This will ensure that a record pair with most similarity gets classified as duplicates.

$$w_{di} = \frac{p_i}{\sum_{j=1}^{m} p_j} \quad \text{// m is the number of fields in dataset}$$

$$p_i = \sum_{k=1}^{n} v_{ki} \quad \text{// n is the number of records in duplicate vector set}$$

Non-Duplicate Intuition

In non-duplicate records the similarity between them should be close to 0. Hence, for a non-duplicate vector V12 that is formed by a pair of non-duplicate records r1 and r2, we need to assign small weights to the fields with large similarity values and large weights to the fields with small similarity values. This will ensure that a record pair with less similarity gets classified as non-duplicates.

$$w_{ni} = \frac{q_i}{\sum_{j=1}^{m} q_j} \quad \text{// m is the number of fields in dataset}$$

$$q_i = \sum_{k=1}^{n}(1 - v_{ki}) \quad \text{// n is the number of records in non-duplicate vector set}$$

Where,
wni = Normalized weight for ith attribute
qi = Accumulated ith attribute dissimilarity value for all non-duplicate vectors

In non-duplicate vectors the dissimilarity value of ith field is 1-vi (where vi is the similarity of ith field). For each field, if it usually has a large similarity value in the non-duplicate vectors, it will have a small accumulated dissimilarity (qi) and will, in turn, be assigned a small weight. On the other hand, it will be assigned a large weight if it usually has a small similarity value in the non- duplicate vectors.

Duplicate Identification

Once to get the weights of each field and the similarity vectors of non-duplicate and potential duplicate datasets, the duplicate detection can be done by calculating the similarity between the records. Hence, define the similarity between records as:

$$Similarity(r_1, r_2) = \sum_{i=1}^{n} w_i * v_i$$

Where r1, r2 are the two records for which the similarity is being calculated.
wi is the weight of field(i).
vi is the similarity vector of two records r1, r2 of field(i) .

V. PERFORMANCE ANALYSIS

The experimentation is performed over the presented algorithm with Windows XP Operating System, in Java language and used 6.9.1 version of Netbeans Integrated Development Environment. The algorithm is implemented using Pentium IV computer system. In order to verify the proposed algorithms, the data sets are segregated based on its content. The segregated data are clustered by using number of keyword, selected based its feasibility to satisfy the generalized rule. The adopted rule is fine tuned in every iteration, which makes the proposed technique applicable for various spheres. The performance analysis proved that the proposed work is advantageous than the earlier works because It is efficient in terms of indexing, it reduces the unnecessary usage of crawler resources, It retrieves effective URLs which are more related to the requirement. The research work presented here is tested for its feasibility based on the following metrics with the existing methodologies Performance, Effectiveness, Time Analysis, Mixer Purity The metrics are discussed with the graphs to support the advantage of the proposed pairwise rule generalization technique with respect to the existing methods.

Performance Performance is the measure of achievement of the particular task verified against the previous attainments. The performance of the proposed method is compared with the performance of the existing methods. The analysis result is shown in Fig 3. The performance of the proposed pairwise rule generalization technique is found superior to that of the earlier methods.

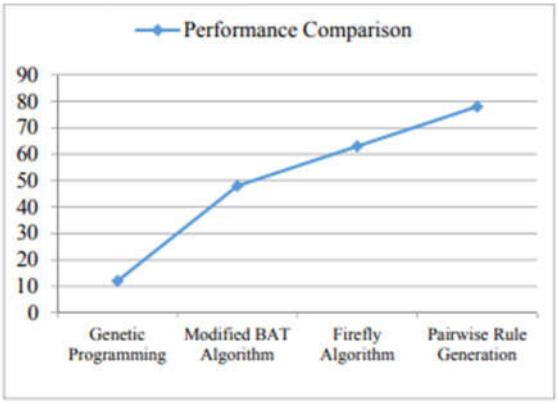

Fig 3. Comparison of performance

Effectiveness The degree of compatibility of a system to the targeted issue, checked against various dataset is called as the effectiveness of the system. The effective detection of duplication leads to easier de-duplication process thereby reduces the noise content in the result. The proposed work is considered to be on par with the other methodologies in its effectiveness measure. The experimental proof is shown in the following fig. 4 in which the effectiveness of all the methods are comparatively studied.

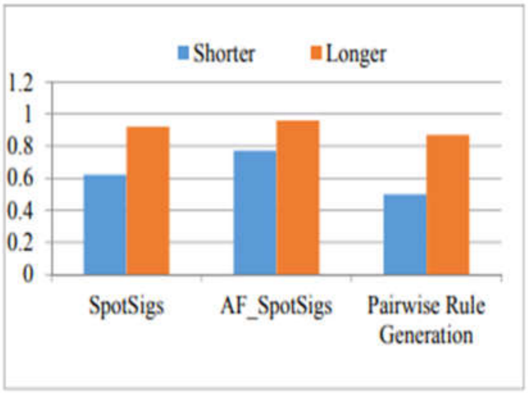

Fig 4. Analysis of effectiveness

VI. CONCLUSION

This Paper concentrated on the development of an Unsupervised Duplicate Detection algorithm that can serve as foundation for developing applications that use Web databases. Seen from the results, using an additional classifier (like blocking) can result in higher accuracy. 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. Although the focus of the UDD application in the thesis was limited to restaurant dataset, the same principles can be used broadly to other domains. When compared to traditional databases, Web-based retrieval systems 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. UDD algorithm which is an unsupervised, online approach for detecting duplicates is a suitable solution, when query results are fetched from multiple Web databases. The core of UDD algorithm relies on using WCSS and SVM **classifiers** to assign weights and classify data. This thesis is a step forward in enhancing the UDD algorithm by adding an additional classifier.

VII. REFERENCES

[1] Fayyad, Usama; Gregory Piatetsky-Shapiro, and Padhraic Smyth (1996) "From Data Mining to Knowledge Discovery in Databases".

[2] SB Kotsiantis, "Supervised learning: A review of classification techniques" Informatica, vol. 31, pp. 249–268, 2007.

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

[4] W. E. Winkler. The state of record linkage and current research problems. Technical Report RR99/04, US Census Bureau, 1999.

[5] I.P Fellegi and A. B. Sunter. A theory for record linkage. Journal of the American Statistical Association, 40, 1969.

[6] M. A. Hernandez ´ and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 2(1):9–37, 1998.

[7] V.S. Verykios, G.V. Moustakides, and M.G. Elfeky, "A Bayesian Decision Model for Cost Optimal Record Matching," The VLDB J., vol. 12, no. 1, pp. 28- 40, 2003.

[8] McCallum, K. Nigam, and L. H. Ungar. "Efficient clustering of high-dimensional data sets with application to reference matching". In KDD, pages 169–178, 2000.

[9] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. "Eliminating fuzzy duplicates in data warehouses". In VLDB, pages 586– 597, 2002.

[10] S. Chaudhuri, V. Ganti, and R. Motwani, "Robust Identification of Fuzzy Duplicates" , Proc.21st IEEE Int'l Conf. Data Eng., pp. 865876,2005.

[11] K. C.-C. Chang and J. Cho, "Accessing the web: From search to integration," in SIGMOD, 2006, pp. 804–805.

[12] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: Exploring the power of tables on the web," PVLDB, vol. 1, no. 1, pp. 538–549, 2008.

[13] W. Meng and C. Yu, Advanced Metasearch Engine Technology. Morgan & Claypool Publishers, 2010.

[14] A. Gruenheid, X. L. Dong, and D. Srivastava, "Incremental record linkage," PVLDB, vol. 7, no. 9, pp. 697–708, May 2014.

[15] E. K. Rezig, E. C. Dragut, M. Ouzzani, and A. K. Elmagarmid, "Query-time record linkage and fusion over web databases," in ICDE, 2015, pp. 42–53.

[16] W. Su, J. Wang, and F. Lochovsky, "Record matching over query results from multiple web databases," TKDE, vol. 22, no. 4, 2010.

[17] H. K¨opcke and E. Rahm, "Frameworks for entity matching: Acomparison," DKE, vol. 69, no. 2, pp. 197–210, 2010.

[18] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," ICDE, 2008.

[19] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," TKDE, vol. 19, no. 1, pp. 1–16, 2007.

[20] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," TKDE, vol. 24, no. 9, 2012.

[21] S. Tejada, C. A. Knoblock, and S. Minton, "Learning object identification rules for information integration," Inf. Sys., vol. 26, no. 8, pp. 607–633, 2001.

[22] L. Shu, A. Chen, M. Xiong, and W. Meng, "Efficient spectral neighborhood blocking for entity resolution," in ICDE, 2011.

[23] Y. Jiang, C. Lin, W. Meng, C. Yu, A. M. Cohen, and N. R.Smalheiser, "Rule-based deduplication of article records from bibliographic databases," Database, vol. 2014, 2014.

[24] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: Is the problem solved?" in PVLDB, vol. 6, no. 2, 2012, pp. 97–108.

Authors Profile

Mrs. T. Sunitha working as Associate Professor of CSE Department in QIS College of Engineering and Technology (Autonomous), Ongole, Andhra Pradesh, India.

Mr. **Maddineni. Lakshmi Sainath** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **Nandyala. Venugopal** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC 'A' Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **Nurubhashu. Kaleshavali** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

Mr. **Madala. Anil Kumar** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.