www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

Development of Data Quality Federation for Adaptive Product Search Engine Using Big Data and Fuzzy Logic Algorithms

A.V. Praveen Krishna[1], Katragadda Raghuveer[2]; N. Tirumala Rao[3]; K. Venkata Prasad[4]

[1]Associate Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur (Dist), India.

[1]praveenkrishna@kluniversity.in

[2]Department of Business Management, V R Siddhartha Engineering College, Vijayawada, India.

[2]katragadda.raghu@gmail.com

[3]J.B. Institute of Engineering & Technology, Hyderabad, India.

[3]thirumalrao.cse@jbiet.edu.in

[4]Associate Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur (Dist), India.

[4]prasad_kz@yahoo.co.in

Abstract

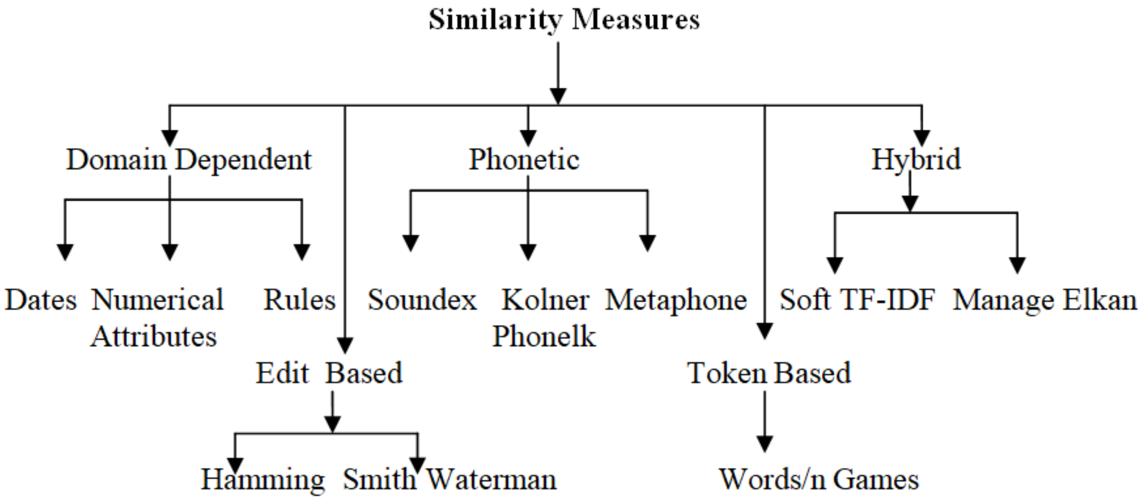
The rapid growth in data volumes and the need to integrate data from various heterogeneous resources bring to the fore the test of making the efficient detection of the duplicate copy of records in databases. Various ways have been proposed in recent days to address the problem; however each technique has one or more flaws that prohibit it from being successfully implemented. For this task, we offer an online machine learning method that incrementally learns a composite similarity function based on a linear combination of basic functions. The proposed work suggests an approach to improve the accuracy of the duplicate record detection process which when used in combination with two other concepts of text similarity and edit distance leads to a well filtered data. This paper advocates to usage of X - Query extension functions for XML data cleaning application scenarios, and detail the implementation on top of an existing X-Query engine. The quantitative measures of common elements in medical records are considered in this article, and fuzzy logic is used to link to language notions. Duplication Detection and Incompleteness Resolution (DDIR) approach has been proposed to improve the quality of the end users' data. Record Linkage and Weighted Component Similarity Summing (WCSS) approach are used to detect and remove the duplicate records. The fuzzy logic framework is a powerful tool for dealing with linkage issues. The described multiple valued logic method could be used to solve similar problems in different databases. The normalized URLs are tokenized and pattern tree is constructed.

Key-words: Big Data, Data Integration, Data Quality, Duplication, Data De-duplication, Machine Learning Clustering, Trigrams, Similarity.

1. Introduction

Databases play important role in contemporary digitalized world where emphasis is on encouraging paper less alternatives. A number of organizations require quality data for critical decision making like various entitlements, concessions or may it be a distribution system. [1] Data mining assists both technologists and technology users in lowering costs and increasing profits. Furthermore, data mining is one of the most effective analytical methods for analyzing, categorizing, and summarising data [2]. Multiple legacy and information systems support the health-care system's health-care providers. Because of the poor design of the information systems, they enable the integration and simplification of healthcare delivery processes in order to improve quality. The adoption of comprehensive information systems in the health care system has proven to be a route wrought with risks and perils. database management systems and poor performance [3]. Historically, one of the most studied examples of record linking has been establishing if two database records for a person correspond to the same person, which is a crucial data cleaning step [4]. Prior to data mining, it is required to improve the quality of data in a data warehouse. process. For various goals, a variety of data cleansing techniques are used [5]. Record duplication is the process of detecting these various versions. In general, duplicate records have a higher resemblance than random pairs of records [6]. Furthermore, many record linkage tasks, such as product normalisation, require many fields to be connected. These records could be pre-structured, or the fields might have been derived from a written description [7].

Figure 1 - Structure of Similarity Measures



ISSN: 2237-0722
Vol. 11 No. 3 (2021)
Received: 30.04.2021 – Accepted: 16.05.2021

2. Related Works

Block-level compression on individual database pages is the most prevalent way operating DBMSs minimize data storage space. When pages are evicted from memory and written to disc, MySQL's InnoDB, for example, can compress them [3] [8].

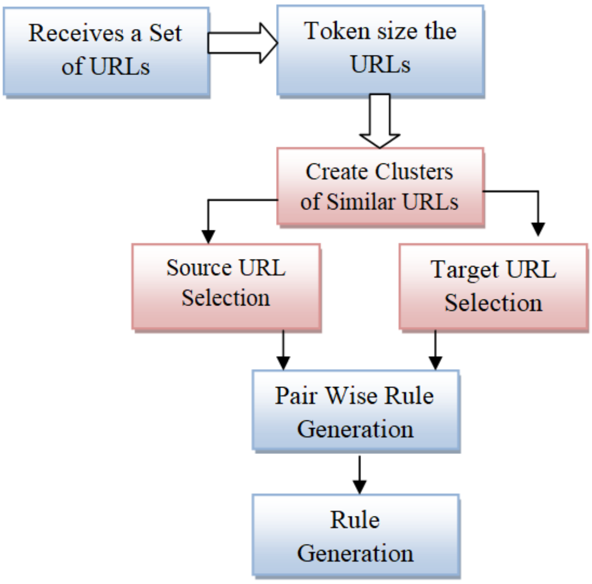
The framework for this essay is the fuzzy logic approach process. A set of general domain-independent transformation functions is given in the framework to reconcile the various text formats of attributes or fields in records [9]. Data mining is used by Blockbuster Entertainment to recommend products to customers based on their video rental history in its database, which uses point-of-sale transaction data and saves it in its data warehouse [10]. To deal with the huge query, it employs hash. The notion that new data is constantly becoming accessible as product offers is a significant aspect of the product normalization domain offers arrive from merchants [11]. Simultaneously, a tiny percentage of inbound product offers have the Universal Product Code (UPC) feature, which uniquely identifies items functions across different offer fields [12]. The weights of basic functions in the linear combination are learned from labelled training data using a variant of the voted perception approach [13], which is very efficient for enormous volumes of streaming data in an online learning scenario. And, with the text, the duty of locating duplicates and near duplicates becomes even more critical, as every scholarship scheme has a limited number of sheets that must be filled with these types of data, and no one may earn two scholarships at the same time [14]. The efficiency of the sorted neighborhood method comes from sliding a window across the dataset and comparing just records within that window. The authors propose using XML parent and child relationships to compare objects, and applying the windowing methodology in a bottom-up manner to detect duplication at each level of the XML hierarchy [15].

3. System Architecture

A framework was proposed to perform data transformation, duplicate elimination and multi-table matching with a set of purposely designed macro-operators [16]. Other characteristics to consider are the number of in-links minimum hop distance and the number of out-links minimum hop distance. Source Selection: The statistically based rating aids in the selection of a source URL. The proposed work's fundamental flaw was that it increased the number of unstructured files and crowded the name space, which addressed the de-duplication issue utilizing the Uniform Resource Locator [17].

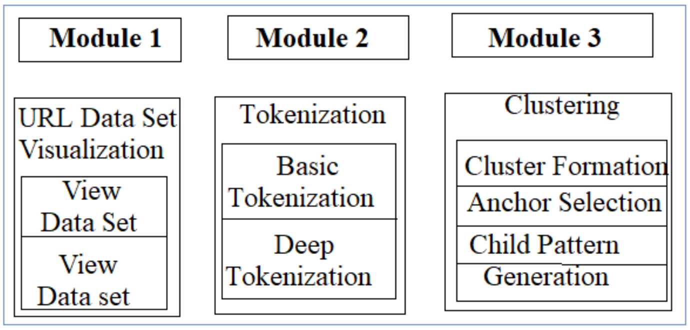
We offer a novel application domain of product normalization for comparison shopping as an example of the record linking problem in a data stream scenario.

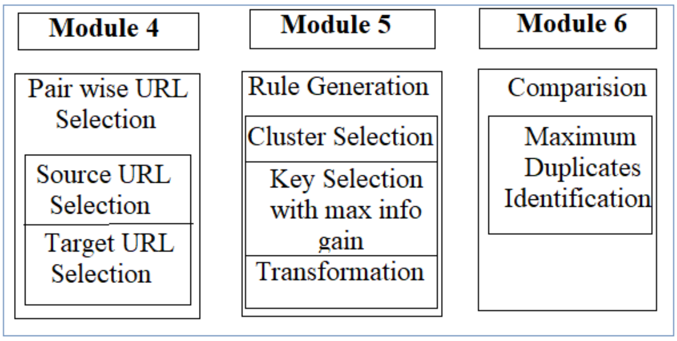
Figure 2 - Overview of System Architecture



We have captured the data via online and offline mode in form of excel sheets there are possibilities of data inconsistency in case of data being collected like typographical errors while typing in excels while typing for district name it may be mistyped due to human error. [18]. The URLs are parsed for host-specific tokens and delimiters. The source and target URLs are chosen when pair-wise rule creation is completed. Machine learning is used to fine-tune the rules using the generalisation methodology [19]. The results are precise and duplicative-free. The suggested system's basic modules are as follows:

Figure 3 - Modules of Proposed System





URL Dataset Visualization: To achieve data duplication, the work involved two sets of data. It is well recognised that data sets, both small and large, can be used for experimentation. These datasets either contain the URLs of a large number of websites or the URLs of a large number of web pages.

1. **Tokenization:** From the supplied standard delimiters, the protocols, hostnames, query arguments, and path components are also extracted. The URLs in the datasets are used to create clusters first. Then, from the URL clusters created in the preceding phase, anchors are chosen.
2. **Clustering:** Clustering is the process of forming clusters using URLs. Module 3's first step is to build the cluster, which is then sent to the rule generalization module.
3. **Pair wise rule generation:** The pair wise rule generation module is used to generate pair wise rules from duplicate cluster URL pairs. This module lays forth the transformational rules.
4. **Rule generalization:** The cluster groupings were chosen. A key is chosen from the cluster that was previously picked. Knowing that all keys contain information gains, the key selection is made by examining the key's maximal information gains.

Comparison: This is the final step in presenting users with non-redundant duplicated data. The number of duplicated data is evaluated using two data sets.

4. Algorithms XML

Data Cleaning

Relatively to the process of XML Data Cleaning is possible to create a methodology which ensures the cleaning of XML data sources [20]. In relational databases it is possible to ensure the data cleaning of relations ensuring the cleaning of theirtuples of their attributes are resolved. Since the

Received: 30.04.2021 – Accepted: 16.05.2021

XML structure varies depending on the data source the methodology for XML data cleaning must be different and has to take into account the hierarchical structure (tree based) of XML data[21].

X-Query Extension Functions for Data Cleaning

These functions return a normalized value for a given field depending on its type separate class of functions into two sub- groups: those that handle basic types and those that handle complex entities, such as addresses phone numbers, city and country names, and zip codes [22]. These functions return the most similar entry in a dictionary for a given input word. The following functionalities were supplied. . They are applied to a set of values of the same type and return a single value. They are useful when we need to merge or consolidate data records.

Training the Composite Similarity Function

To identify co-referent records, each candidate pair of records must be classified as either co-referent pairings M or non-equivalent pairings U. Given a domain ∆R from which each record is sampled and a set of K similarity functions fk:∆R

× ∆R → R that operate on pairs of records, we can produce a pair-space vector xi ∈R K+1 for every pair of records (Ri1, Ri2): xi=[1,f1(Ri1,Ri2),...,fK(Ri1,Ri2)]T. By classifying the associated feature vector xi and interpreting classification confidence as similarity, any binary classifier that returns confidence scores can be used to assess the overall similarity of a record pair (Ri1,Ri2).

Algorithm: Averaged Perceptions Training

Input: Training set of record pairs {(Ri1, Ri2,yi)}, y ∈ {−1, +1} number of epochs T similarity functions F = {fi(·, ·)} K i=1

Output: Weight vector αavg = {αi} K i=0

```
Algorithm:
Initialize αavg = α = 0
Initialize xi = [1,f1(Ri1 ,Ri2 ),...,fk(Ri1 ,Ri2)]
fori = 1...M For t = 1...T For i = 1...M
Compute ˆyi = sign(α · xi)
```

ISSN: 2237-0722
Vol. 11 No. 3 (2021)
Received: 30.04.2021 – Accepted: 16.05.2021

If ˆyi 6= yi α = α + yixi αavg = αavg + α αavg = αavg T·M

Other loss functions, such as log-loss llog(xi,yi) = ln(1 + exp(yi • xi), are used in the algorithm. Because changing the loss function did not result in a qualitative change in final performance on the linking task in our studies, we only publish findings using hinge loss.

5. Results

We have examined the proposed solutions output for the smaller data set /training dataset to set our threshold parameter values for similarity score and normalized edit distance. Then we have executed proposed algorithms to the nearly crore datasets and we are able to identify the lacs records that are near duplicates. The suggested pair wise rule generalisation strategy outperforms the previous techniques. The efficiency of the system is defined as the degree of compatibility of the system with the targeted issue as measured against multiple datasets.

Figure 4 - Performance Comparison

Choosing when to cease further cluster merging so that the residual clusters may be reported as groupings of co-referent data is an important part of the hierarchical cluster merging stage. The measure of achievement of a certain task when compared to past achievements is called performance. The new method's performance is compared to that of the existing approach.

6. Conclusion

This work has presented an algorithm for overcoming the issue of inconsistent data and detecting the near duplicates withan advance approach. Due to the high quantity of duplication present in the web, crawling relevance and indexing, two crucial components of Internet search via a search engine, have been affected. This is a serious problem for consumers of internet search engines. Because there is so much duplicated data created even by autonomous users, duplication has become a need. In the process of obtaining unique URLs, the methodology is unique. The data from two different data sets is compared and the duplicated data is examined. The uniqueness weight of an attribute is calculated using decision tree learning for matching rules generation by dividing the total number of unique attribute values contained in the attribute set by the total number of values for that attribute set. We discovered that linkage methods that examine the similarity of numerous pairs of records at the same time perform much better than a pair-wise technique in which each linkage choice is made separately. We discovered that linkage approaches that assess the similarity of numerous pairs of records at the same time perform much better than a pair-wise technique where each connection is considered separately decision is made in isolation. Another work done in the future to improve the results generated by the data cleaning programs implemented withthe data cleaning library, is the support for user feedback and the interactive cleaning.

References

K.C.C. Chang and J. Cho, *“Accessing the web: From search to integration,” in SIGMOD,* 2006, 804–805.

Alvero E. Monge *“An adaptive and efficient algorithm for detecting approximately duplicate database record”* California State University Lond Beach CECS Department CA 90840 8302

Li Yujian And Liu Bo “A Normalized Levenshtein Distance Metric” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 29(6), June 2007

Oleg Bartunov, Teodor Sigaev “Effective Similarity Search In PostgreSQL” *Lomonosov Moscow State University,* PGCon-2012, Ottawa

Deepa, K. and Rangarajan, R. A Comprehensive Review of Significant Researches on Duplicate Record Detection in Databases, *Advances in Computational Sciences and Technology,* 2(2), 117-134, 2009.

Elmagarmid AK, Horowitz B, Karabatis G, Umar A. Issue in multisystem integration for achieving data reconciliation and aspects of solution. Technical report, Bellcore Research;1996.

Trillium Software. How data profiling & analysis saves companies $millions. *White paper in data integration and data quality management.* 2003 [cited11.08.11].

Wei M, Sung AH, Cather ME. Improving database quality through eliminating duplicate records. *Data Sci J* 2006; 5(19): 127–142.

Dasu T, Johnson T, editors. *Exploratory data mining and data cleaning.* New Jersey: John Wiley & Sons, Inc.; 2003.

R. Saha Roy, R. Sinha, N. Chhaya, and S. Saini, "Probabilistic deduplication of anonymous web traffic,*" in Proceedings of the 24th International Conference on World Wide Web Companion,* 2015, 103-104.

S. Zawoad, R. Hasan, G. Warner, and A. Skjellum, "UDaaS: A Cloud-based URL- Deduplication-as-a-Service for Big Datasets," *in IEEE International Conference on Big Data and Cloud Computing (BdCloud)* 2014, 271-272.

X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, pp. 97-107, 2014.

M.Y. Lin, P.Y. Lee, and S.C. Hsueh, "Apriori-based frequent itemset mining algorithms on MapReduce," *in Proceedings of the 6th international conference on ubiquitous information management and communication,* 2012, 76.

L. Kolb, A. Thor, and E. Rahm, "Dedoop: efficient deduplication with Hadoop," *Proceedings of the VLDB Endowment,* 5, 1878-1881, 2012.

A. Das Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon, "An automatic blocking mechanism for large- scale de-duplication tasks," *in Proceedings of the 21st ACM international conference on Information and knowledge management,* 2012, 1055-1064.

K. Shim, "MapReduce algorithms for big data analysis," *Proceedings of the VLDB Endowment,* 5, 2016-2017, 2012.

M. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. of the American Statistical Association,* 84: 414– 420, 1989.

G.R.G. Lanckriet, N. Cristianini, P.L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *J. of Machine Learning Research,* 5: 27–72, 2004.

S. Lawrence, K. Bollacker, and C.L. Giles. Autonomous citation matching. *In Proc. of Agents*-1999, 392–393,1999.

X. Li, P. Morie, and D. Roth. Robust reading: Identification and tracing of ambiguous names. *In Proc. of NAACL-* 2004, 17–24, 2004.

C.D. Manning and H. Schutze. ¨*Foundations of Statistical Natural Language Processing.* MIT Press,1999.

A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. *In Proc. of ACM SIGKDD-2000,* 169– 178, 2000.

S.S. Harsha, H. Simhadri, K. Raghu, K.V. Prasad, “Distinctly trained multi-source cnn for multi-camera based vehicle tracking system” *Published in International Journal of Recent Technology and Engineering,* 2019, 8(2), 624–634.

Atmakuri Krishna Chaitanya, K.V. Prasad, “A Comparative study on Prediction of Indian Air Quality Index Using Machine Learning Algorithms” *Published in Journal of Critical Reviews,* 2020, 7(13), 41–46.

P. Vidya Sagar, S.S. Harsha., K.V. Prasad., Moparthi, N.R. “Transferable deep learning assisted radar signal processing model for sea-target detection and classification” *Published in Journal of Green Engineering,* 2020, 10(10), 7661–7671.

T. Chava, A.T. Srinivas, A.L. Sai and V. Rachapudi, "IoT based Smart Shoe for the Blind," *2021 6th International Conference on Inventive Computation Technologies (ICICT),* 2021, 220-223, doi:10.1109/ICICT50816.2021.9358759

Rama Krishna, V., Subhamastan Rao, T., Narayana, G.V.S., Rachapudi, V., “A model for stock price predictions using deep learning techniques”, *International Journal of Advanced Trends in Computer Science and Engineering,* 2020, 9(5), 8266–8271.

Venubabu Rachapudi, Chandra Harsha Talapaneni, Dhanush Kolluri, Abdul Nadeem Akthar, S Anjali Devi. (2020). Improved Convolutional Neural Network for Classification of White Blood Cells. *International Journal of Control and Automation, 13*(02), 883 - 888. http://sersc.org/journals/index.php/IJCA/article/view/11968

Venubabu Rachapudi; G. Lavanya Devi,” Optimal bag-of-features using random salp swarm algorithm for histopathological image analysis”, International Journal of Intelligent Information and Database Systems (IJIIDS), 13(2/3/4), 2020, DOI:10.1504/IJIIDS.2020.109450.

Anjali Devi, S., Vishnu Priya, M., Akhila, P., Vasundhara, N, “Analysis and prediction of student placement for improving the education standards”, *International Journal of Engineering and Technology (UAE),* 7, 2.8, 303-306, 2019.

S.S. Vavilapalli, P. Reddy Korepu, S. Saggam, M. Pentyala and S.A. Devi, "Summarizing & Sentiment Analysis on Movie Critics Data," *2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021,* 1-5, doi:10.1109/ICICT50816.2021.9358563.

Anjali Devi, S., Sapkota, P., Rohit Kumar, K., Pooja, S., Sandeep, M.S. “Comparison of classification algorithms on twitter data using sentiment analysis”, *International Journal of Advanced Trends in Computer Science and Engineering, 2020, 9*(5), 8170–8173.