Data Cleaning: Current Approaches and Issues

Conference Paper · January 2011

CITATIONS
16

READS
19,477

2 authors:

Ratnadeep R. Deshmukh
Dr. Babasaheb Ambedkar Marathwada University
292 PUBLICATIONS **1,357** CITATIONS
SEE PROFILE

Vaishali Wangikar
MIT Academy of Engineering
8 PUBLICATIONS **35** CITATIONS
SEE PROFILE

CHAPTER 9

Data Cleaning: Current Approaches and Issues

[1]Vaishali Chandrakant Wangikar and [2]Ratnadeep R. Deshmukh
[1]MCA Department, Maharashtra Academy of Engineering, Alandi, Pune (MS), India,
[2]Deptartment of Computer Science & IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS), India
E-mail: vaishali.wangikar@gmail.com, ratnadeep_deshmukh@yahoo.co.in

ABSTRACT

The data cleaning is the process of identifying and removing the errors in the data warehouse. While collecting and combining data from various sources into a data warehouse, ensuring high data quality and consistency becomes a significant, often expensive and always challenging task. Without clean and correct data the usefulness of Data Mining and data warehousing is mitigated. This paper analyzes the problem of data cleansing and the identification of potential errors in data sets. The differing views of data cleansing are surveyed and reviewed and a brief overview of existing data cleansing techniques is given. We also give an outlook to research directions that complement the existing systems.

Keywords: Sorted Neighborhood Methods, Fuzzy Match, Clustering and Association, Token-Based Data, Record Linkage.

1. INTRODUCTION

Common data quality problems(anomalies) include inconsistent data conventions amongst sources such as different abbreviations or synonyms; data entry errors such as spelling mistakes inconsistent data formats, missing, incomplete, outdated or otherwise incorrect attribute values, data duplication, irrelevant objects or data. Data that is incomplete or inaccurate is known as “dirty” data.

The various types of anomalies occurring in data that have to be eliminated. The type of anomalies can be classified under several types of it. Based on this classification we evaluate and compare existing approaches for data cleansing with respect to the types of anomalies handled and eliminated by them.

The paper categorizes the data cleansing into two categories: cleansing string data and record or attribute de-duplication.

Data cleaning offers the fundamental services for data cleaning such as attribute selection, formation of tokens, selection of clustering algorithm, selection of similarity function, selection of elimination function and merge function etc.

The paper is organized as follows. Related Research Work describes various existing data cleaning techniques, Comparison of existing techniques, Conclusion and Future Work.

2. RELATED RESEARCH WORK

2.1 Cleansing String Data

This category of data cleaning removes 'dirt' in strings (words). Algorithm that identifies a group of strings that consists of (multiple) occurrences of a correctly spelled string plus nearby misspelled strings. All strings in a group are replaced by the most frequent string of this group.

2.1.1 Data Cleaning for Misspelled Proper Nouns (Border Detection Algorithm)

The method is proposed by Arturas Mazeika and Michael H.B¨ohlen in 2006. The method targets proper noun databases, including names and addresses, which are not handled by dictionaries. *Center Calculation* and *Border Detection algorithms* [1] are suggested. Data cleansing is done in two steps. First, the string data is clustered by identifying center and border of hyper-spherical clusters, and second, the cluster strings are cleansed with the most frequent string of the cluster. All strings within the overlap threshold from the center of the cluster are assigned to one cluster.

Border Detection algorithm is a simple and effective strategy to compute clusters in string data. One starts with a string in the database and selects the border that separates the cluster from the other clusters. If the initial string was chosen close to the center of the cluster, the border detection will yield good and robust results. If one chooses the initial string close to the border, two separate clusters might be assigned. As the cluster size increases, the relative clustering error decreases. The algorithm successfully identifies borders of clusters provided a sufficient sample size. The robustness of the algorithm is not affected by the cluster size.

Experiments show that the border detection is robust provided a sufficient sample size. The investigation indicates that very few q-grams of the center strings are sufficient to identify strings of the cluster. An algorithm that robustly finds the identifying q-grams of the cluster is an interesting challenge.

2.1.2 Robust and Efficient Fuzzy Match for Online Data Cleaning(Fuzzy Match similarity Algorithm)

To ensure high data quality, data warehouses must validate and cleanse incoming data tuples from external sources. In many situations, clean tuples must match acceptable tuples in reference tables. A significant challenge in such a scenario is to implement an efficient and accurate fuzzy match operation that can effectively clean an incoming tuple if it fails to match exactly with any tuple in the reference relation. A few similarity function which overcomes limitations of commonly used similarity functions is proposed, and an efficient fuzzy match algorithm is developed in 2003 by Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti Rajeev Motwani. Edit distance similarity [2] is generalized by incorporating the notion of tokens and their importance to develop an accurate fuzzy match similarity function for matching erroneous input tuples with clean tuples from a reference relation. The error tolerant index [2] and an efficient algorithm is developed for identifying with high probability the closest fuzzy matching reference tuples. Using real datasets, demonstration of the high quality of proposed similarity function and the efficiency of algorithms given.

2.1.3 Data Cleaning by Clustering and Association Methods (Data Mining Algorithms)

The two applications of data mining techniques in the area of attribute correction: context-independent attribute correction implemented using clustering techniques and context-dependent attribute correction using associations are proposed by Lukasz Ciszak, 2008 IEEE[3].

Attribute correction solutions require reference data in order to provide satisfying results.

Context-independent attribute correction means that all the record attributes are examined and cleaned in isolation without regard to values of other attributes of a given record.

Context-dependent means that attribute values are corrected with regard not only to the reference data value it is most similar to, but also takes into consideration values of other attributes within a given record

Experimental results of both algorithms created by the author show that attribute correction is possible without an external reference data and can give good results. As it was discovered in the experiments, the effectiveness of a method

depends strongly on its parameters. The optimal parameters discovered here may give optimal results only for the data examined and it is very likely that different data sets would need different values of the parameters to achieve a high ratio of correctly cleaned data.

2.2 Record or Attribute De-duplication

A process for determining whether two or more records defined differently in a database, actually represent the same real world object. During data cleaning, multiple records representing the same real life object are identified, assigned only one unique database identification, and only one copy of exact duplicate records is retained.

2.2.1 A Token-Based Data Cleaning Technique

Most existing work on data cleaning, identify record duplicates by computing match scores compared against a given match score threshold. Some use the entire records for long string comparisons that involve a number of passes. Determining optimal match score threshold in a domain is hard and straight long string comparisons with many passes is inefficient.

The proposed token based technique proposed by Timothy E., Ohanekwu and C.I. Ezeife eliminates the need to rely on match threshold by defining smart tokens that are used for identifying duplicates. This approach also eliminates the need to use the entire long string records with multiple passes, for duplicate identification.

Existing algorithms use token keys extracted from records for only sorting and/or clustering. The results from the experiments show that the proposed token-based algorithm [5] outperforms the other two algorithms.

2.2.2 Record Linkage: Similarity Measures and Algorithms

In the presence of data quality errors, a central problem is the ability to identify whether two entities (e.g., relational tuples) are approximately the same The techniques used here are record linkage and approximate join in the sequel.

A variety of approximate match predicates that have been proposed to quantify the degree of similarity or closeness of two data entities. The authors Nick Koudas, Sunita Sarawagi, Divesh Srivastava have compared and contrasted them based on their applicability to various data types, algorithmic properties, computational overhead and their adaptability. Most approximate match predicates return a score between 0 and 1 (with 1 being assigned to identical entities) that effectively quantifies the degree of similarity between data entities. Such approximate match predicates will consist of three parts.

Atomic Similarity Measures: This part measures to assess atomic (attribute value) similarity between a pair of data entities. Several approximate match predicates including edit distance, phonetic distance (soundex), the Jaro and Winkler measures, tf.idf and many variants thereof. Several approaches to fine tune parameters of such measures are considered.

Functions to combine similarity measures : Given a set of pairs of attributes belonging to two entities (tuples), in which each pair is tagged with it's own approximate match score (possibly applying distinct approximate match predicates for each attribute pair), how does one combine such scores to decide whether the entire entities (tuples) are approximately the same. For this basic decision problem several proposed methodologies like statistical and probabilistic, predictive, cost based, rule based, user assisted as well as learning based are given. Moreover, several specific functions including Naive Bayes, the *Fellegi-Sunter model*, *linear support vector machines* (SVM) and approaches based on voting theory are covered.

Similarity between linked entities: Often the entities over which we need to resolve duplicates are linked together via foreign keys in a multi-relational database. Author has presented various graph-based similarity measures that capture transitive contextual similarity in combination with the intrinsic similarity between two entities.

Record Linkage Algorithms: Once the basic techniques for quantifying the degree of approximate match for a pair (or subsets) of attributes have been identified, the next challenging operation is to embed them into an approximate join framework between two data sets. A common feature of all such algorithms is the ability to keep the total number of pairs (and subsequent decisions) low utilizing various pruning mechanisms. These algorithms can be classified into two main categories.

1. *Algorithms inspired by relational duplicate elimination and join techniques including sort-merge, band join and indexed nested loops*: In this context, techniques like Merge/Purge [9] (based on the concept of sorted neighborhoods), Big Match (based on indexed nested loops joins) and Dimension Hierarchies (based on the concept of hierarchically clustered neighborhoods) are reviewed .
2. *Algorithms inspired by information retrieval that treat each tuple as a set of tokens, and return those set pairs whose (weighted) overlap exceeds a specified threshold:* In this context, a variety of set join algorithms are reviewed [6].

2.2.3 Adaptive Sorted Neighborhood Methods for Efficient Record Linkage

A variety of record linkage algorithms have been developed and deployed successfully. Often, however, existing solutions have a set of parameters whose values are set by human experts off-line and are fixed during the execution. Since finding the ideal values of such parameters is not straightforward, or no such single ideal value even exists, the applicability of existing solutions to new scenarios or domains is greatly hampered. To remedy this problem, an argument is made by Su Yan, Dongwon Lee, Min-Yen Kany, C. Lee Giles that one can achieve significant improvement by adaptively and dynamically changing such parameters of record linkage algorithms. To validate the hypothesis, a classical record linkage algorithm, the Sorted Neighborhood Method (SNM)[7] are used and demonstrated how one can achieve improved accuracy and performance by adaptively changing its fixed sliding window size.

Two adaptive versions of the SNM algorithm, named as the incrementally-adaptive SNM (IA-SNM) and the accumulatively-adaptive SNM (AA-SNM) are proposed, both of which dynamically adjust the sliding window size, a key parameter used in SNM, during the blocking phase to adaptively fit the duplicate distribution. Comprehensive experiments with both real and synthetic data sets of three domains validate the effectiveness and the efficiency of the proposed adaptive schemes.

3. COMPARISON

3.1 The following table shows comparison of three different algorithms especially used for cleaning *string type of data*.

	Border Detection Algorithm	*Data Mining Algorithm-Attribute Correction Algorithm*	*Fuzzy Match Similarity FunctionAlgorithm*
Features	(1) Simple, effective to compute clusters in the string data. (2) It helps in selection of border that separates one cluster from the other. If the initial string was chosen close to the center of the cluster, the border detection will yield good and robust results . If one chooses the initial string close to the border, two separate clusters might be assigned	(1) The given attributes are validated against reference data to provide cleansing solution (2) fuzzy match similarity (*fms*) function that explicitly considers IDF token weights and input errors while comparing tuples.	(1) If the tuple or attribute fails to match the reference data then fuzzy match operation is applied on it. (2) The two applications of data mining techniques in the area of attribute correction are: *context-independent* attribute correction implemented using clustering techniques and *context-dependent* attribute correction using associations.
Significance/ Performance	It produces good cleansing results for string data with large distances between centers of clusters and small distances within the clusters.	Quality of *fms* is better than *ed* (edit distance) using two Datasets. The algorithms are 2 to 3 orders of magnitude faster than the naïve algorithm	The algorithms displays better performance for long strings as short strings would require higher value of the parameter to discover a correct reference value. This method produces as 92% of correctly altered elements which is an acceptable value.

Limitations	Our data cleansing algorithm is less is applicable for natural language databases.	There is always cost associated with transformations of IDF tokens.	The major drawback of this method that may classify as 'incorrect a value that is correct in context of other attributes of this record, but does not have enough occurrences within the cleaned data set

3.2 The following table shows comparison of three different algorithms used *for cleaning duplicate entries of attributes as well as records.*

	Token-Based Algorithm and Algorithms	*Record Linkage Similarity Measures Linkage*	*Adaptive Sorted Neighborhood Methods For Efficient Record*
Features	For finding duplicates of attributes as well as records smart tokens are used instead match score comparison against match threshold. This approach also eliminates the need to use the entire long string records with multiple passes, for duplicate identification.	Approximate match and approximate join techniques are proposed to quantify the degree of similarity Atomic Similarity Measures, Functions to combine similarity measures, Similarity between linked entities are considered for matching Two approximate join techniques proposed, the first is concerned with procedural algorithms operating on data, applying approximate match predicates, without a particular storage or query model in mind. The second is concerned with declarative specifications of data cleaning operations.	Among many parameters of record linkage algorithms, the main focus is on the size of the sliding window in *SNM* and the adaptive version of *SNM* is proposed. The size of the window in SNM amounts to the size of the block, which in turn is related to the aggressiveness of a blocking method. Two adaptive versions of the *SNM* algorithm, named as the incrementally-adaptive *SNM (IA-SNM*) and the accumulatively-adaptive *SNM* (AA-SNM) are proposed.
Significance /Performance	By using short lengthened tokens for record comparisons, a high recall/precision is achieved. It has drastically lowers the dependency of the data cleaning on match "threshold" choice. It has a recall close to 100%, as well as negligible false positive errors. It succeeded in reducing the number of token tables to a constant of 2, irrespective of the number of fields selected by the user. The smart tokens are more likely applicable to domain-independent data cleaning, and could be used as warehouse identifiers to enhance the process of incremental cleaning and refreshing of integrated data.	A non-declarative specification offers greater algorithmic flexibility and possibly improved performance (e.g., implemented on top of a file system without incurring RDBMS overheads). A declarative specification offers unbeatable ease of deployment (as a set of SQL queries), direct processing of data in their native store (RDBMS) and flexible integration with existing applications utilizing an RDBMS	Adaptive sorted neighborhood methods significantly outperform the the original SNM method. AA-SSNM has better performance than *IA-SNM*. The F-score of *AA-SNM* is 49% larger than that of *SNM.* The F-score difference between *IA-SNM* and *AA-SNM* is about 4%, but the PC difference is around 13%. This shows that AA-SNM is a better blocking method than IA-SNM since it finds more potential duplicate pairs with similar F-score

Limitation	Existing algorithms use token keys extracted from records for only sorting and/or clustering Token-based cleaning technique on unstructured, and semi-structured data yet to be considered.	The output of the approximate join needs to be post processed to cluster together all tuples that referto the same entity. The approximate join operation above might produce seemingly inconsistent results like tuple A joins with tuple B, tuple A joins with tuple C, but tuple B does not join with tuple C. A straightforward way to resolve such inconsist-encies is to cluster together all tuples via a transitive closure of the join pairs. In practice, this can lead to extremely poor results since unrelated tuples might get connected through noisy links.	The adaptive schemes are robust to the variance in the size of each individual block, which can range from moderate to severe. Besides, the adaptive schemes show better resistance to the errors in the blocking fields. The performance of the algorithm depends upon the appropriate size of window. Several methods for adjusting window sizes, which are used in the adaptive methods, are proposed and compared. Among them, the full adjustment method is shown to be near optimal.

4. CONCLUSION

Various data cleaning algorithms and techniques are presented in the paper but each method can be used to identify a particular type of error in the data. The technique suitable for one type of data cleaning may not be suitable for the other. As data cleaning has a wide variety of situations that need to cater efficiently by some comprehensive data cleaning framework. Future research directions include the review and investigation of various methods to address wide area of data cleaning. A better integration of data cleaning approach in the frameworks and data decision processes should be achieved.

Acknowledgements

For reviewing different algorithms of data cleaning, experiments and contributions done by several authors are referred. The papers are enlisted in the references.

References

1. Arturas Mazeika Michael H.B¨ohlen: Cleansing Databases of Misspelled Proper Nouns, Clean DB, Seoul, Korea, 2006
2. Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti Rajeev Motwani: Robust and Efficient Fuzzy Match for Online Data Cleaning. ACM, SIGMOD 2003, June 9-12, 2003, San Diego__CA.
3. Rohit Ananthakrishna1 Surajit Chaudhuri Venkatesh Ganti,: Research Eliminating Fuzzy Duplicates in Data Warehouses. Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.
4. Lukasz Ciszak: Application of Clustering and Association Methods in Data Cleaning .Proceedings of the International Multiconference on ISBN 978-83-60810-14-9 Computer Science and Information Technology, pp. 97 – 103.
5. Timothy E. Ohanekwu, C.I. Ezeife: A Token-Based Data Cleaning Technique for Data Warehouse Systems.
6. Nick Koudas, Sunita Sarawagi, Divesh Srivastava: Record Linkage: Similarity Measures and Algorithms. SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.
7. Su Yan, Dongwon Lee, Min-Yen Kany, C. Lee Giles: Adaptive Sorted Neighborhood Methods for Efficient Record Linkage. JCDL'07, June 17.22, 2007, Vancouver, British Columbia, Canada.
8. M. Hernandez and S. Stolfo :The Merge/Purge Problem for Large Databases. Proc. ACM SIGMOD Int'l Conf. Management of Data. pp. 127-138, May 1995.
9. M.A. Hernandez and S.J. Stolfo,:Real-World Data Is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge Discovery, Vol. 2, pp. 9-37, 1998.