Nwagwu *et al. J Big Data (2017) 4:36*
DOI 10.1186/s40537-017-0100-9

Journal of Big Data

METHODOLOGY Open Access

Mining and visualising contradictory data

Honour Chika Nwagwu[1*], George Okereke[2] and Chukwuemeka Nwobodo[3]

*Correspondence:
honour.nwagwu@unn.edu.ng
[1] Computer Science Department, University of Nigeria, Room 434, Abuja Building, Nsukka, Enugu State, Nigeria
Full list of author information is available at the end of the article

Abstract

Big datasets are often stored in flat files and can contain contradictory data. Contradictory data undermines the soundness of the information from a noisy dataset. Traditional tools such as pie chart and bar chart are overwhelmed when used to visually identify contradictory data in multidimensional attribute-values of a big dataset. This work explains the importance of identifying contradictions in a noisy dataset. It also examines how contradictory data in a large and noisy dataset can be mined and visually analysed. The authors developed 'ConTra', an open source application which applies mutual exclusion rule in identifying contradictory data, existing in comma separated values (CSV) dataset. ConTra's capability to enable the identification of contradictory data in different sizes of datasets is examined. The results show that ConTra can process large dataset when hosted in servers with fast processors. It is also shown in this work that ConTra is 100% accurate in identifying contradictory data of objects whose attribute values do not conform to the mutual exclusion rule of a dataset in CSV format. Different approaches through which ConTra can mine and identify contradictory data are also presented.

Keywords: ConTra, Comma separated values, Dataset, Contradictions, Contradictory data, Mutual exclusion values

Introduction

A noisy dataset can contain contradictory data. Contradictory data is synonymous to incorrect data and it is important that such data be investigated and evaluated when analysing a noisy dataset. Different approaches to dealing with contradictory data have been proposed by different researchers. For example [1, 2] proposed methods for identifying and removing contradictory data in noisy datasets. However, the removal of contradictory data from a noisy dataset will increase the incompleteness in the dataset thereby reducing the soundness of any information from such set of data. It is therefore important to identify and evaluate contradictory instances when analysing a large and noisy dataset. This will improve the soundness of the analysis from such a dataset. Evidently, the analysis of big data is identified as the next frontier for innovation and advancement of technology [3, 4]. There is therefore the need to identify appropriate approaches to dealing with contradictions in a large and noisy dataset.

There are different forms of contradictions. For example, there are contradictions from the use of modal words, structural, subtle lexical contrasts, as well as world knowledge

(WK) as evident in Natural Language Processing (NLP). Some contradictions in NLP can occur where there are antonyms, negations, and date/number mismatch [1, 2]. Contradictory data can exist in a single source dataset in instances where there are systematic errors, arbitrary errors or different value representations [5, 6]. A more common source for contradictions is federated data from multiple sources such as data exchange [7], data fusion [8, 9] and data warehousing [10, 11]. In addition, data that are retrieved from internet sources such as data from Blogs and social networking sites are likely to be contradictory.

A set of data consists of information about real world objects. Some examples of real world objects are your dog, house, or car. Real world objects '*G*' can be associated with different attributes '*M*' which may have many values '*V*'. For example, dogs can have different colours (attributes) which can be black, white or brown (values). Contradictory data can exist in any dataset when the data contain conflicting information. Consequently, an object ($g \in G$) that is associated with an attribute ($m \in M$), can contain contradictory values such that m is associated with A and ¬A. For example, the grade associated to a student's score in a module can be said to be contradictory if it is associated with a 'pass' and a 'fail'. Even so, the metadata of a dataset specifies what kind of dependencies exists between the different values of the attributes in the dataset. It provides descriptive information about the characteristics of a given item in a dataset [12]. As a result, contradictory data can be said to be evident in a noisy dataset, where the data does not conform to the metadata of the dataset. For example, a dataset about students can contain the information that a student passed mathematics in the result from a particular examination body and failed it in the result from another examination body in the same year. Such data is not contradictory where the metadata describes that a student can be assessed on multiple results from different examination bodies of the same year. But the same data will be regarded as contradictory where the metadata describes that a student cannot be assessed on multiple results from different examination bodies of the same year. Accordingly, an object ($g \in G$) that is associated with an attribute ($m \in M$), where m is associated with A and ¬A, can be described as contradictory if the dependencies existing between the different values of the attribute does not conform to the metadata of its dataset.

The importance of identifying and evaluating contradictory data in a noisy dataset cannot be overstated. It is stated in [4] that "noisy Big Data could be more valuable than tiny samples because general statistics obtained from frequent patterns and correlation analysis usually overpower individual fluctuations and often disclose more reliable hidden patterns and knowledge". Ennals et al. [13] identify that the analysis of contradictions will enable the data user to recognise when the information he reads online is disputed and by what source. Marneffe, Rafferty and Manning explain in [14] how contradiction detection systems can be applied in intelligence reports, bioinformatics, and political candidate debates. Tsytsarau et al. [15] state that the usefulness of aggregation and analysis of sentiments based contradictions on the web includes the provision of the ability to track the evolution of contradictory opinions or discussions in the blogosphere. Kim and Zhai [16] outline the importance of generating a comparative summary of contradictory opinions. Leser and Freytag explain in [5] that the identification of the patterns in contradictory data will help in providing answers to questions like "Which are

the conflict-causing attributes, values, or value pairs?" and "What kind of dependencies exists between the occurrence of contradictions in different attributes?".

On the other hand, contradictory data existing in a large dataset can be difficult to visualise especially when traditional data analysis and visualisation tools are employed. As explained by Keim et al. and Keim [17, 18], traditional data processing tools such as (x, y) plots, linear and bar-charts, histogram, and pie charts are rendered ineffective when a dataset contains tens, hundreds or thousands of dimensions and when the dataset does not have natural mapping to the display space. This work explains how to visually identify contradictory values which are associated with mutually exclusive attributes in a large and noisy comma separated values (CSV) dataset. It addresses the challenge of using traditional data processing tools in visually identifying contradictions. It answers the research question "how can contradictions in mutually exclusive data of a large and noisy dataset, be visually identified?"

This paper presents the importance of identifying contradictions in a noisy dataset and how to apply mutual exclusion rule in identifying contradictory data. It presents novel approaches for visually identifying contradictory data in a large and noisy dataset. The authors herein explain how contradictory data can be mined and visually analysed using ConTra. ConTra is an application developed by the authors of this work. It applies the mutual exclusion rule in mining contradictory data of a noisy CSV dataset. Also, the authors evaluated Contra's capability to identify contradictory data in different sizes of datasets.

Section two of this paper explains how mutual exclusion rule is applied in identifying contradictions. ConTra's mutual exclusion approach to mining and visualising contradictory data is presented in section three. A description of a real life noisy dataset and the results of its analysis using ConTra are presented in section four. The performance evaluation of ConTra is presented in section five. A conclusion and discussions on the way forward is presented in section six.

Application of mutual exclusion rule in identifying contradictory data

The mutual exclusion rule has over time been applied in dealing with contradictory data such as in resolving the problem of system resource sharing in a distributed environment as explained by Le [19]. Barbara et al. [20] describe how voting can be used to enforce mutual exclusion rule and ensure system integrity during catastrophic failures such as partition failures in distributed systems. Lately, many researchers [21–23] have proposed different approaches on how mutual exclusion algorithm can be applied in improving distributed computing systems. Also, database management systems such as Oracle and MySQL provide platforms for retrieving contradictory data from a noisy dataset. For example, queries that enforce the mutual exclusion rule on selected attribute values can be written for a particular dataset. Even so, the structured query language (SQL) non-specialist will not be able to write such queries while the SQL specialist may likely write wrong queries/scripts which will lead to wrong results.

There are very few publications that apply the mutual exclusion rule as a technique to visually identify contradictory data in a noisy dataset. Indeed, visual analysis of contradictory data in noisy CSV dataset is rarely discussed. There are calls for visual analysis applications with interactive capabilities [24, 25]. Nwagwu and Orphanides [26, 27]

demonstrate how FcaBedrock [28] and Concept Explorer [29] can be used to visually analyse and identify gene expressions contradicting the mutually exclusion rule in gene expression dataset where a gene in a particular tissue of a particular Theiler stage is expected to be associated to only one type of expression. They applied formal concept analysis (FCA) tools and techniques in visualising the contradictory data in a large dataset. ConTra is another visual analysis application which applies mutual exclusion rule in dealing with contradictory data. "Mining and visual analysis of contradictory data using ConTra" describes how ConTra deals with contradictory data in a large dataset.

Mining and visual analysis of contradictory data using ConTra

The authors of this work developed ConTra which enables the mining and visual analysis of contradictions in a dataset. ConTra provides a platform which allows its users to mine and analyse the contradictions in 'many' or 'single' valued attribute(s) whose data does not abide by the mutual exclusion rule of the dataset. It is a web application which allows the visualisation of contradictions in a large CSV dataset. The mutual exclusion rule in ConTra was implemented using PHP programming language. The user interface of ConTra was developed with HTML, CSS and JavaScript. ConTra's source code is deposited on GitHub (a code hosting platform for version control and collaboration) at https://github.com/ncjoes/contra.

A typical CSV file has its content depicted in rows and columns. ConTra enables its users to select only one of the columns as an object column. The user also selects attribute columns from the other columns in the dataset. ConTra's interface enables the application user to upload a CSV document for investigation. The application user selects either *single attribute value* or *multiple attribute values* approach, when investigating a dataset for contradictory values. He further selects his object of interest, the attribute(s), and associated values which he wants to investigate for contradictions. A click on 'analyse' will reveal any contradictory data associated with his selections and display such in a pie chart as evident in Fig. 2. ConTra adopts the following approaches in identifying contradictory data:

Mining contradictory data in objects associated with multiple attribute values

This approach is applied in an object associated with an attribute which has mutually exclusive values such that a value can contradict another value of the same attribute. For example, in records associated to a student, many values (such as 'Pass', 'Fail' and 'Not available') may be associated to his grade in a particular module. ConTra identifies contradictions by allowing its users to select two or more values from an investigated attribute before mining the dataset for the existence of such contradicting attribute values in a particular object. Algorithm 1 depicts the approach by which ConTra mines contradictory and consistent data.

Algorithm 1: ConTra's Algorithm for mining contradictory and consistent

1. Given a set of records in CSV format
2. Let **G** = Set of Objects from a selected column
3. Let **M** = Set of Attributes (titles of every column excluding the Object column)
4. Let **O**(**a**,**b**) = empty list where **a** = contradictory object index and **b** = contradictory attribute values
5. Let **C**(**c**,**d**) = empty list where **c** = consistent object index and **d** = consistent attribute values
6. For each Object '**g**' in the set of objects '**G**' which are associated to a set of mutually exclusive attributes '**M**'

   i. If '**M**' contains more than one mutually exclusive value then
      Store '**g**' and also store each of the contradictory values in the list **O**(**a**,**b**)
   ii. Else
      Store '**g**' in set of consistent objects and also store each of the consistent values in the list **C**(**c**,**d**)

   End

7. Print contradictory objects **O**(**a**, **b**) and consistent objects **C**(**c**, **d**)

Mining contradictory data in objects associated with single attribute value

This approach is applied in an object associated with many attributes when its single attribute contains a value which may contradict any other value in another attribute. For example, records associated to a student can contain many attributes such as '*grade in Maths*', '*grade in English*', and '*status of the student*'. The values '*Pass*' in '*grade in Maths*', a '*Fail*' in '*grade in English*', and a '*Promoted*' in '*status of the student*' can be described as contradictions in a dataset where they are mutually exclusive for a particular student in the identified attributes. ConTra allows its user to select only single values from each attribute before mining the dataset for the existence of such contradictions. It enables its application user to query attributes of particular objects whose values are mutually exclusive. Objects associated with the contradicting attribute values are tagged as contradictory. Again, the associated attribute values of the contradictory object are tagged as contradictory attribute values.

Graphical representation of contradictory data

ConTra graphically represents contradictory data from a noisy dataset by mining the contradictory data in the investigated dataset and enabling the visualisation of such data in a pie chart. An open source chart library "Chart.js" is integrated into ConTra as to enable the visualisation of contradictions. The pie chart in Chart.js is programmatically structured in ConTra, to accept identified percentage of contradictions in an investigated dataset and present such against the consistent data. It should be noted that the term "consistent data" in this context, refers to the data that conforms to the mutual exclusion rule as described in the metadata of the dataset.

Dataset analysis and results

The authors conducted an experiment in which ConTra was used in visually identifying contradictory data in objects whose attributes are associated with mutually exclusive values. They used ConTra to visually analyse the 'Normal Tissue' dataset [30]. The Normal Tissue contains expression profiles for proteins in human tissues based on immunohistochemisty using tissue micro arrays. The dataset is in comma-separated file format. The following columns are available in the dataset: '*Gene*', '*Gene name*', '*Tissue*', '*Cell type*', '*Level*', and *Reliability*. The dataset has a size of 79.5 MB. It contains six columns and over a million rows of data.

The *Gene* column consists of gene identifiers while the *Gene name* identifies the name of each of the gene identifier. The *Tissue* column consists of tissue names. The *Cell type* column contains data which shows the annotated cell types. The *Level* column contains gene expression values, and the *Reliability* column contains data which shows the degree of reliability associated to the expression values.

There are many records in the investigated dataset whose Tissue is 'Pancreas' and which are associated to the gene '*TSPAN6*'. *TSPAN6* was selected as a value of the attribute '*Gene name*' (see Fig. 1). The interest of the authors is to find out if there are any *TSPAN6* which have contradictory values such that it expresses a 'low' and a 'medium' expression levels in a Tissue. Contradictions were found in only two of the records (9.09%) as depicted in Fig. 2. Thus, any analysis about the *TSPAN6* expression levels notably its low and medium expression levels in the Tissue '*Pancreas*' of the Normal Tissue dataset, must include the amount of contradictions identified in the expressions of *TSPAN6*.

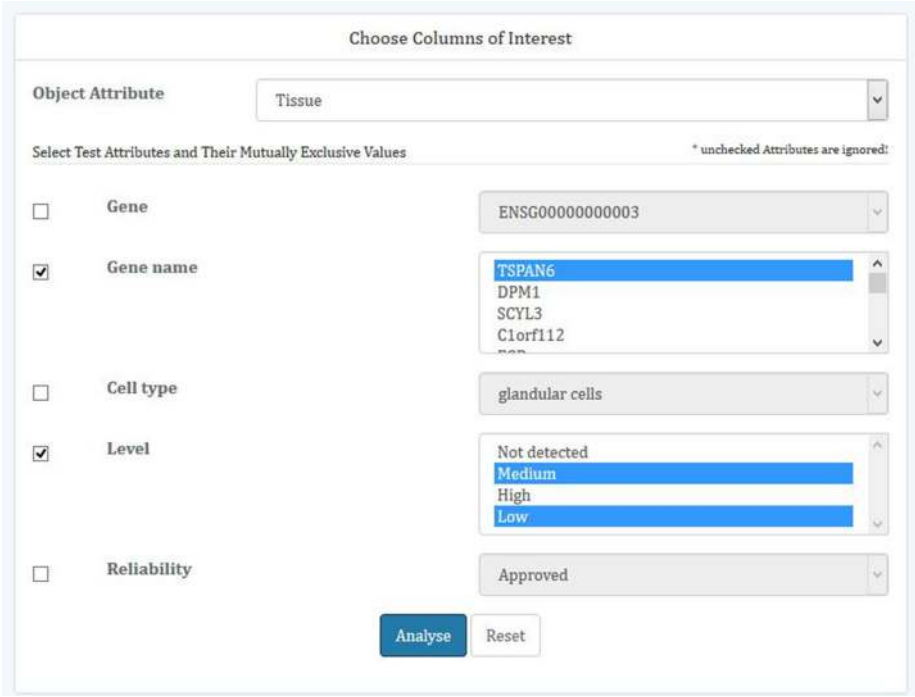

Fig. 1 Analysis of the normal tissue dataset by ConTra's multiple attribute values approach

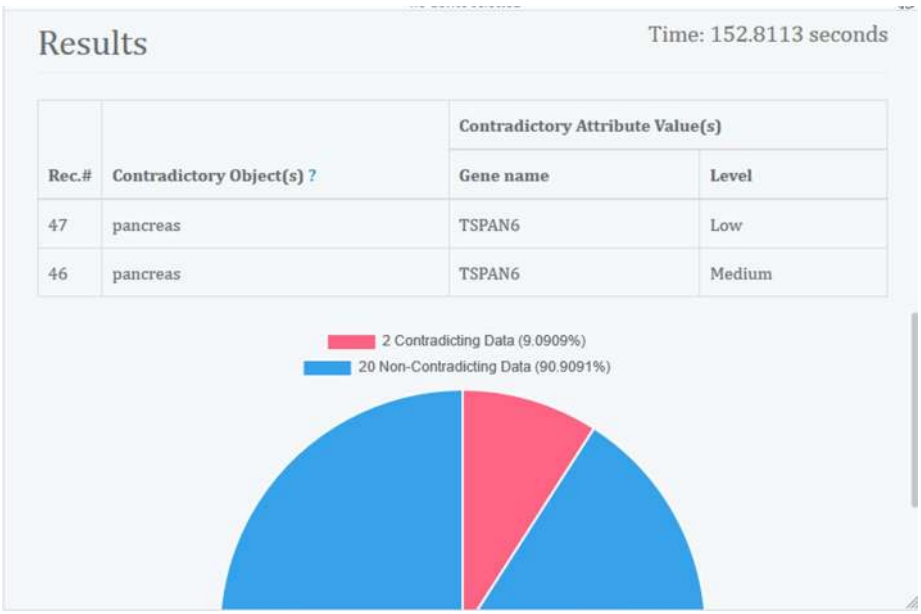

Fig. 2 Result of the analysis of the normal tissue dataset by ConTra's multiple attribute values approach

Performance evaluation of ConTra

In this section, the authors describe how they evaluated the accuracy, uniqueness, and processing speed of ConTra. In addition, they present experimental results involving the use of ConTra in analysing datasets of different sizes.

The authors assessed the accuracy of ConTra by storing the natural tissue dataset in MySQL database. They queried the dataset for instances where a tissue is associated with the gene *TSPAN6* which expresses a medium and a low expression levels. The query is designed to identify any contradictory data in the same objects and attribute values as the ones explored by ConTra (see Figs. 1, 2). The authors identified two contradicting data associated with *Pancreas* tissue. This number of contradictions is the same with the amount of contradictions identified when the dataset was explored using ConTra. The dataset was also manually explored for issues of contradictory data in tissues associated with the gene "TSPAN6" where the gene expresses a medium and a low expression levels. The same contradictions as identified by ConTra and the use of query were observed. This confirms that ConTra is 100% accurate in retrieving contradictory data from objects associated with mutually exclusive attribute values in an investigated CSV dataset.

More so, ConTra's approach enables the evaluation of the identified contradictions. Unlike existing approaches such as in [26, 27], ConTra improves on the use of traditional visualisation tool (pie chart) in visual analysis of contradictory data, by mining and evaluating only the contradicting data. This enables the use of traditional visualisation tools in visually analysing the contradictory data in a large CSV dataset. ConTra also displays the precise locations of its identified contradictions (the objects and contradicting attribute values) in the investigated dataset (see the Table embedded in Fig. 2).

ConTra's ability to process high volume of data was assessed through an experimental approach. It is noted in [31] that the ability to process high volume of data

is an issue that has to be considered when designing an effective time-critical infrastructure for big-data applications. The ability to process high volume of data was assessed in ConTra by evaluating its execution time when it processes data from large dataset. The authors installed ConTra in two servers (P1 and P2) whose RAM sizes are the same (8.00 GB) but different processor speeds. The processor speed of P1 is 1.6 GHz while the processor speed of the second server (P2) is 2.1 GHz. The authors downloaded a large consumer complaint dataset from the United State Government's open data website [32]. The dataset was resized into five different sets. This was achieved by deleting the rows of each saved version of the dataset and saving the reduced set of data with a new name. Each of the saved set of data was analysed for the existence of contradictory data in a selected object, selected attributes and associated values using ConTra. The authors selected from each set of data, the 'Consumer Complaint Narrative' as the object, 'Company response to consumer' as an attribute and 'untimely response' as its value. Also they selected 'Timely response' as the second attribute and 'Yes' as its value. These data were mined by ConTra in each dataset and the execution time recorded. Table 1 and Fig. 3 present the sizes of the datasets and the execution times of the different servers in which ConTra was installed.

It can be observed from Table 1 and Fig. 3 that in both servers (P1 and P2), the execution time increases with an increase in the size of the dataset. Obviously, there are challenges with system performance when processing big data as observed in [33–35]. This has led to different proposals and implementations of techniques which can improve the performance of systems that process big data. For example, [36] describe how parallelism technique is applied in Dremel. Dremel scales to thousands of CPUs and petabytes of data using shared clusters of commodity machines. A big data application can be developed to use parallel processors to increase its processing speed as evident in [37–39]. Consequently, ConTra's processing time can be improved to process very large dataset by implementing its algorithm on a parallel processor environment.

It is also observed that P2 server has shorter execution time for each processed dataset when compared to P1 server. This is because P2 server has higher processor speed than P1 server. Consequently, ConTra can deal with a large dataset with short execution time when it is processed by faster processors. Even so, there is need to develop better version of ConTra which can process tens of GigaByte (GB) of data as to enable the identification of contradictory data in industrial CSV datasets. The authors hope to address this concern in future versions of ConTra.

Table 1 ConTra's processing speed on different servers for different dataset sizes

S/no	Size of dataset (MB)	P1 execution time (s)	P2 execution time (s)
1	53.3	568.5666	348.0682
2	74	997.4232	402.3342
3	113	1499.844	613.1654
4	161	2141.9178	946.8948
5	182	2345.6694	985.3766

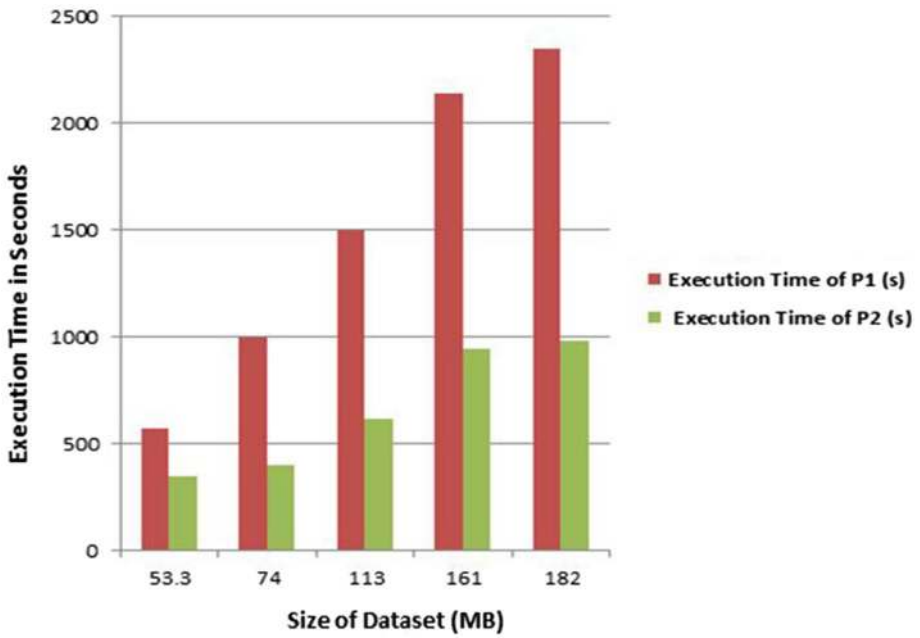

Fig. 3 Execution times vs dataset sizes of the different servers hosting ConTra

Conclusion and the way forward

Contradictory data can lead to an unsound analysis and eliminating its instances does not enable sound analysis when dealing with a noisy set of data. This work has identified novel approaches for mining and visualising contradictory data which exists in a noisy CSV dataset. It is hoped that future work will examine how objects, attributes and values can be mined from other dataset formats such as text, resource description framework in attributes RDFa and XML. This will enable the use of ConTra in visualising contradictory data in such data formats. Also, there is need to combine the mutual exclusion technique (as presented in this work) with other contradictory detection techniques. This is because the use of mutual exclusion technique is limited to contradictions which results from allocating conflicting values to mutually exclusive attributes. Arbitrary errors such as human errors in tabulating data, or numeric mismatch are some of the examples of contradictory data which ConTra is not designed for. The authors hope to introduce a newer version of ConTra with improved performance such that it can process tens of GigaByte (GB) of data in a short interval of time. They also hope to take advantage of parallel processor programming in enhancing ConTra's processing speed in its future versions.

Authors' contributions

HCN initiated the article, wrote part of the literature review and supervised the programming of the application—ConTra. GO wrote part of the literature and supervised the programming of the application—ConTra. CN programmed the application—ConTra. All authors read and approved the final manuscript.

Author details

[1] Computer Science Department, University of Nigeria, Room 434, Abuja Building, Nsukka, Enugu State, Nigeria. [2] Computer Science Department, University of Nigeria, Room 429, Abuja Building, Nsukka, Enugu State, Nigeria. [3] Care of Dr. Nwagwu Honour Chika, Computer Science Department, University of Nigeria, Room 434, Abuja Building, Nsukka, Enugu State, Nigeria.

Acknowledgements
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
Open source data (http://www.proteinatlas.org/about/download). Deposit of the developed application (ConTra) (https://github.com/ncjoes/contra). Open source data (https://catalog.data.gov/dataset/consumer-complaint-database).

Consent for publication
Not applicable.

Ethics approval and consent to participate
Not applicable.

Funding
None.

Publisher's Note

Received: 4 July 2017 Accepted: 21 October 2017
Published online: 30 October 2017

References

1. Fong S, Biuk-Aghai RP, Si YW, Yap BW. A lightweight data preprocessing strategy with fast contradiction analysis for incremental classifier learning. Math Probl Eng. 2015:1–11. doi:10.1155/2015/125781.
2. Hodge V, Austin J. A survey of outlier detection methodologies. Artif Intell Rev. 2004;22(2):85–126.
3. Samuel SJ, RVP K, Sashidhar K, Bharathi CR. A survey on big data and its research challenges. ARPN J Eng Appl Sci. 2015;10(8):3343–7.
4. Jagadish HV, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C. Big data and its technical challenges. Commun ACM. 2014;57(7):86–94.
5. Leser U, Freytag JC. Mining for patterns in contradictory data. In: Proceedings of the 2004 international workshop on information quality in information systems. New York City: ACM; 2004. p. 51–8.
6. Fürber C, Hepp M. Using SPARQL and SPIN for data quality management on the semantic web. In: Business information systems. Berlin: Springer; 2010. p. 35–46.
7. Hernich A, Libkin L, Schweikardt N. Closed world data exchange. ACM Trans Database Syst. 2011;36(2):14.
8. Bleiholder J, Naumann F. Data fusion. ACM Comput Surv. 2009;41(1):1.
9. Kumar M, Garg DP, Zachery RA. A method for judicious fusion of inconsistent multiple sensor data. IEEE Sens J. 2007;7(5):723–33.
10. Kimball R, Caserta J. The data warehouse? ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data. New York: Wiley; 2011.
11. Calvanese D, De Giacomo G, Lenzerini M, Nardi D, Rosati R. Data integration in data warehousing. Int J Coop Inf Syst. 2001;10(03):237–71.
12. Guptill SC. Metadata and data catalogues. Geogr Inf Syst. 1999;2:677–92.
13. Ennals R, Trushkowsky B, Agosta JM. Highlighting disputed claims on the web. In: Proceedings of the 19th international conference on World Wide Web. New York City: ACM; 2010. p. 341–50.
14. Marneffe MC, Rafferty AN, Manning CD. Finding contradictions in text. In: ACL, vol. 8. 2008. p. 1039–47.
15. Tsytsarau M, Palpanas T, Denecke K. Scalable detection of sentiment-based contradictions. DiversiWeb, WWW. 2011.
16. Kim HD, Zhai C. Generating comparative summaries of contradictory opinions in text. In: Proceedings of the 18th ACM conference on information and knowledge management. New York City: ACM; 2009. p. 385–94.
17. Keim D, Andrienko G, Fekete JD, Gorg C, Kohlhammer J, Melançon G. Visual analytics: definition, process, and challenges. Lect Notes Comput Sci. 2008;4950:154–76.
18. Keim DA. Visual exploration of large data sets. Commun ACM. 2001;44(8):38–44.
19. Le Lann G. Distributed systems-towards a formal approach. In: IFIP congress, vol. 7. 1977. p. 155–60.
20. Barbara D, Garcia-Molina H, Spauster A. Protocols for dynamic vote reassignment. In: Proceedings of the fifth annual ACM symposium on principles of distributed computing. New York City: ACM; 1986. p. 195–205.
21. Bertier M, Arantes L, Sens P. Distributed mutual exclusion algorithms for grid applications: a hierarchical approach. J Parallel Distrib Comput. 2006;66(1):128–44.
22. Malek S, Mikic-Rakic M, Medvidovic N. A decentralized redeployment algorithm for improving the availability of distributed systems. In: Dearle A, Eisenbach S, editors. Component deployment. Berlin, Heidelberg: Springer; 2005. p. 99–114.
23. Cao J, Zhou J, Chen D, Wu J. An efficient distributed mutual exclusion algorithm based on relative consensus voting. In: Parallel and distributed processing symposium, 2004. Proceedings. 18th international. New York: IEEE; 2004. p. 51.
24. Harshbarger J, Kratz A, Carninci P. DEIVA: a web application for interactive visual analysis of differential gene expression profiles. BMC Genom. 2017;18(1):47.

25. Kandel S, Heer J, Plaisant C, Kennedy J, Van Ham F, Riche NH, Weaver C, Lee B, Brodbeck D, Buono P. Research directions in data wrangling: visualizations and transformations for usable and credible data. Inf Vis. 2011;10(4):271–88.
26. Nwagwu HC. Visualising inconsistency and incompleteness in RDF gene expression data using FCA. Int J Concept Struct Smart Appl. 2014;2(1):68–82.
27. Nwagwu HC, Orphanides C. Visual analysis of a large and noisy dataset. Int J Concept Struct Smart Appl. 2015;3(2):12–24.
28. FcaBedrock Formal Context Creator. https://sourceforge.net/projects/fcabedrock/. Accessed 20 Aug 2017.
29. The Concept Explorer. http://conexp.sourceforge.net/. Accessed 20 Aug 2017.
30. The Human Protein Atlas. http://www.proteinatlas.org/about/download. Accessed 4 May 2017.
31. Basanta-Val P, Audsley NC, Wellings AJ, Gray I, Fernández-García N. Architecting time-critical big-data systems. IEEE Trans Big Data. 2016;2(4):310–24.
32. The United State Government's open data website. https://catalog.data.gov/dataset/consumer-complaint-database. Accessed 16 Aug 2017.
33. Steve L, Eric L, Rebecca S, Hopkins MS, Kruschwitz N. Big data, analytics and the path from insights to value. MIT Sloan Manag Rev. 2011;52:21–32.
34. Pradhananga Y, Karande S, Karande C. High performance analytics of bigdata with dynamic and optimized Hadoop cluster. In: 2016 international conference on advanced communication control and computing technologies (ICACCCT). New York: IEEE; 2016. p. 715–20.
35. Wu X, Zhu X, Wu GQ, Ding W. Data mining with big data. IEEE Trans Knowl Data Eng. 2014;26(1):97–107.
36. Melnik S, Gubarev A, Long JJ, Romer G, Shivakumar S, Tolton M, Vassilakis T. Dremel: interactive analysis of web-scale datasets. Proc VLDB Endow. 2010;3(1–2):330–9.
37. Dakkak A, Pearson C, Li C, Hwu WM. RAI: a scalable project submission system for parallel programming courses. In: Parallel and distributed processing symposium workshops (IPDPSW), 2017 IEEE international. New York: IEEE; 2017. p. 315–22.
38. Radford D. A comparative analysis of the performance of scalable parallel patterns applied to genetic algorithms and configured for NVIDIA GPUs (Doctoral dissertation). 2016.
39. Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, Meng X, Rosen J, Venkataraman S, Franklin MJ, Ghodsi A. Apache spark: a unified engine for big data processing. Commun ACM. 2016;59(11):56–65.