Journal of Computer and Communications, 2016, 4, 79-86
Published Online May 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.45012

Scientific Research Publishing

Improve Data Quality by Processing Null Values and Semantic Dependencies

Houda Zaidi[1,2], Faouzi Boufarès[3], Yann Pollet[1]

[1]Laboratory CEDRIC, Conservatoire National des Arts et Métiers, Paris, France
[2]Laboratory RIADI, University Manouba, Tunis, Tunisia
[3]Laboratory LIPN, University Paris 13, Sorbonne Paris Cité, Villetaneuse, France
Email: houda.zaidi@cnam.fr, faouzi.boufares@lipn.univ-paris13.fr, yann.pollet@cnam.fr

Received 12 May 2016; accepted 19 May 2016; published 26 May 2016

Abstract

Today, the quantity of data continues to increase, furthermore, the data are heterogeneous, from multiple sources (structured, semi-structured and unstructured) and with different levels of quality. Therefore, it is very likely to manipulate data without knowledge about their structures and their semantics. In fact, the meta-data may be insufficient or totally absent. Data Anomalies may be due to the poverty of their semantic descriptions, or even the absence of their description. In this paper, we propose an approach to better understand the semantics and the structure of the data. Our approach helps to correct automatically the intra-column anomalies and the inter-columns ones. We aim to improve the quality of data by processing the null values and the semantic dependencies between columns.

Keywords

Data Quality, Big Data, Contextual Semantics, Semantic Dependencies, Functional Dependencies, Null Values, Data Cleaning

1. Introduction

Data quality represents a very main challenge because the cost of anomalies can be very high especially for large databases in enterprises that need to exchange information between systems and integrate large amounts of data. Decision making using erroneous data has a bad influence on the activities of organizations. Quantity of data continues to increase as well as the risks of anomalies. The automatic correction of these anomalies is a topic that is becoming more important both in business and in the academic world [1] [2]. The data may be derived from different sources for which metadata can be totally absent and most often not sufficient to reflect the actual content of the data and treat any anomalies. Therefore, it is interesting to create new data integration tools to better understand the semantics and structure of the data. We develop this work in collaboration with the Talend Company which is an editor of one the most known of open source data integration and data quality tools [3]. The first part of the project consisted to treat the inter-lines anomalies [4]-[7]. Indeed, the automatic correction of anomalies in a same column gives better results, insofar as their semantics is known. Moreover, the discovery of semantic links that may exist between the columns may avoid the violation of various types of depen-

How to cite this paper: Zaidi, H., Boufarès, F. and Pollet, Y. (2016) Improve Data Quality by Processing Null Values and Semantic Dependencies. *Journal of Computer and Communications*, **4**, 79-86. http://dx.doi.org/10.4236/jcc.2016.45012

dencies constraints between them. This is what constitutes the second part of the project which is the subject of this article. Constraint checking dependencies from large amounts of data will help to correct automatically the anomalies such as null values and some functional dependencies. In this paper, our proposal is to try to understand the semantics of the data before correcting it. An intra-column study allows the automatic correction of anomalies in the data related to its context. For example, the following two strings "Londres" and "London" are equivalent if they represent the context of *city names* but they are different if they are *names of people*. Algorithms of similarity distance calculation such as Levenshtein [8], Jaro-Winkler [9], Soundex [10] and Metaphone [11] do not take into account the context. Our approach is to illustrate the data structure (*i.e*. the semantic schema) of the data source. First, the syntactic and semantic automatic corrections inside one column can be better focused. Second, eventual semantic links that may exist between columns can be discovered. Then, the process of anomalies caused by the violation of dependency constraints will start. In a context of Big Data, we use the MapReduce technology to verify the discoveries dependencies and correct anomalies caused by the violation of these dependencies. The rest of the article is organized as follows. The second section describes the step of semantic recognition of data. Section three discusses the intra-column and inter-columns data cleaning step. Finally, we compare our approach with other work and we address our future goals.

2. Semantic Categorization of Data

The step of data categorization consists to determine the semantics of each column of a data source. Indeed, in order to qualify a syntactically incorrect data, it should be evaluated in its context. Several examples can illustrate this: 1) The string "Pari" can be considered syntactically incorrect only if it is the French name of the city "Paris". The words "Pékin" and "Beijing" mean the same thing in two different languages if we know that these are names of cities. "Beijing" could be considered semantically incorrect if the used language (dominant language) is French; 2) The three strings "16-10-1996", "10-16-1996" and "1996-16-10" may represent the same type of information "date with different formats" defined by a regular expression; 3) The string "1996-10" is not a date.

Then, we introduce the use of stored knowledge in a referential called Data Dictionary (DD) [5] [12] [13]. The DD contains knowledge identified by two modes: 1) Data defined by extension: a priori given list such as city names, country names and names of organizations. Valid Strings (DDVS) and Key Words (DDKW) are stored in the DD; 2) Data defined by intention: this knowledge must verify some properties such as regular expressions or belonging to an interval of values (DDRE). For example *Emails*, *Websites* and *Dates* must be conforming to a model which can be a regular expression. In other words, the DD can then be seen as a set of *categories*. Each category corresponds to only one *data type* (String, Number, Date). Some categories can have *subcategories* such as languages. We propose a DD that contains not only information grouped by categories (Valid Strings, Rules such as Regular Expression) but also semantically valid knowledge such as constraints or functional dependencies pre-stored. **Figure 1** below shows examples of pre-stored knowledge.

2.1. Recognition of the Data Structure

As metadata can be totally absent, we can consider that the Data Source (DS) will have to be corrected is in a CSV format and therefore without schema. The process of semantic categorization uses the DD to assign each column of the source to a context and thus a semantic (a category). It returns a new semantic structure of the DS enriched with constraints and comments. The objective is to propose for each column: 1) a semantic name (the category), 2) eventually a subcategory (the language), 3) a data type (the syntactic domain), 4) intra and inter-columns constraints and 5) comments. Details are given in [12].

Given the following example of DS (**Figure 2**), the *semantic recognition* consists to find similarities between the DS and the valid strings in the DD to infer the semantic name of each column. We interested by the contextual data quality. We use measurements of similarity distance. We propose a mixture of methods of similarity measure, on the one hand, "it is written as", such as Jaro-Winkler or Levenshtein, and on the other hand, "is pronounced as", such as Soundex or Metaphone. The application of our semantic recognition process on "Patient.csv" file (**Figure 2**) gives the results in **Figure 3**. Let us note that the DS has no significant schema:

S (Column 1: String, Column 2: String, Column 3: String, Column 4: String, Column 5: String, Column 6: String, Column 7: String, Column 8: String, Column 9: String, Column 10: String).

The semantic process which we propose gives a new semantic structure of data summarized by:

Categories	Sub-Categories		Links between categories		Semantic dependencies
Category	***English***	***French***	***Primary key***	***Foreign key***	
Continent	AFRICA AMERICA ASIA EUROPE	AFRIQUE AMÉRIQUE ASIE EUROPE	Continent1 Continent2 Continent3 Continent4		City → Country Country → Continent Civility → Sex Museum → City Airport → City University → City Laboratory → City Stadium → City …→ … City → Continent University → Country …→ …
Country	FRANCE TUNISIA CHINA ITALY CREECE	FRANCE TUNISIE CHINE ITALIE GRÈCE	Country0052 Country0121 Country0035 Country0068 Country0058	Continent4 Continent4 Continent3 Continent4 Continent4	
City	TUNIS PARIS BEIJING	TUNIS PARIS PÉKIN	TN00007 FR00024 CN00001	Country0121 Country0052 Country0035	
FirstName(*)	HOUDA CLÉMENT ADAM PARIS	HOUDA CLÉMENT ADAM PARIS	FirstName1 FirstName4 FirstName5 FirstName6		

Figure 1. Extract of the data dictionnary for valid strings and semantic dependencies.

t01	100;Jean;M;M;A+;12/12/1984;0033645456932;Parix;France;Europe
t02	101;Eve;Mme;Femme;B+;10-Sep-85;0033741256811;Paris;;Europe
t03	102;Stephane;M;F;AB+;15-mars-65;0033644356922;Nice;Franc;
t04	103;Adem;;M;AB-;03-avr-80;0033645336941;Tunis;Tunisia;
t05	104;Anne;Mlle;0;1+;16-10-1996;0033645412944;Beijing;Chine;Asie
t06	105;Karine;Mademoisel;0;O-;05/07/1983;0033638456945;Pikin;China;
t07	106;Simon;M;Homme;AB-;04/02/1974;0033645144031;Par;Franca;Europe
t08	107;Robert;M;;AB-;04/06/1975;0033675404932;Paris;Houda;EUROPE
t09	108;Joseph;M;1;AB-;1996-16-10;0033775334900;Paris;Frence;Europe
t10	109;Karine;Mme;Femme;B+;1996-16-10;0033752232901;Paris;France;
t11	110;Martin;M;Femme;B+;1996-10;0033732132421;Paris;;
t12	111;Anne;Mlle;Femme;A+;02/03/1928;0033732011427;;;
t13	112;Karine;Mme;Femme;B+;1996-16-10;0033752232901;Paris;France;
t14	113;;M;Femme;B+;1996-10;0033732132421;Paris;Fr;EUROPE
t15	114;Yang;Mlle;Femme;A+;02/03/1983;0033732011427;Paris;Cnam;EUROPE
t16	115;Tang;Mlle;Femme;A+;02/10/1990;0033632047419;Pikin;Chine;Asie
t17	116;Yang;M;Femme;A+;02/10/1995;0033632047419;Pékin;Chine;Asy

Figure 2. Patient.csv file extraction (from the DS).

New Semantic schema					
IdColumn	**Category**	**SubCategory**	**Data Type**	**Constraints**	**Comments**
Column1	Number	*	Number		Interval of values
Column2	FirstName	*	String		
Column3	Civility	French	String	ϵ {M, Mme, Mlle} Civility → Sex	a finite set of values
Column4	Sex	French	String	ϵ {Homme, Femme}	a finite set of values
Column5	BloodGroup	*	String	ϵ {A+, B+, AB+, O+, …}	a finite set of values
Column6	Date	French	Date	DD-MM-AAAA	
Column7	Phone	French	Number		
Column8	City	French	String	City → Country City→Continent	
Column9	Country	French	String	Country → Continent	
Column10	Continent	French	String		

Figure 3. Semantic categorization of data (Patient.csv).

S (Number: Number, First Name: String, Civility: String, Sex: String, Blood Group: String, Date: Date, Phone: Number, City: String, Country: String, Continent: String); with constraints such as Civility ϵ {M;Mme;Mlle} and functional dependencies such as *Civility* → *Sex* and *Country* → *Continent*.

2.2. Recognition of Semantic Dependencies between Columns

Let us note that it is possible to exploit the syntactic domain of data to infer the implausible dependencies. Thus, the knowledge pre-stored can guide the user and reduce the search space in the process of discovering dependencies constraints. Suggestions of semantically valid dependencies can be given to the user. Contrary to some works such as [14] [15] which search to verify the functional dependencies between all columns. For example, trying to verify the following dependencies has no meaning: *BloodGroup* → *City* or *Date* → *Country* or *Date* → *FirstName or FirstName* → *Date or FirstName* → *BloodGroup.*

3. Data Cleaning

3.1. Correction of Intra-Column Anomalies

In this section, we present our approach which allows exploiting the semantic knowledge deduced from the step of semantic categorization to correct intra-column anomalies. We call this phase homogenization/standardization of data. The syntactical correction of data is done by approximating the values of the data source to those which are similar in the Data Dictionary. We use methods of similarity distance calculation. Our cleaning process can correct misspelled values. It allows standardizing formats. Some examples of syntax corrections are given in **Figure 4**. Note that certain transformations are not feasible at this level such as the string "Fr" or null values (NULL). The step of the semantic categorization of the data allows recognizing the dominant category and eventually the dominant subcategory (language) in each column. We propose to unify the data in the same subcategory. Then values that do not belong to the dominant subcategory are translated to their synonym in the dominant language. Various formats can be used in the same column. We propose unified coding values in a dominant Format (**Figure 5**).

The intra-column transformations do not correct the errors of violation of semantic dependencies. Null values are not treated at this level of the process.

	Category	Old values	New values
Syntax corrections	City	Parix	PARIS
	City	Pikin	PÉKIN
	Country	Franca	FRANCE
Semantic corrections	City	Beijing	PÉKIN
	Country	Tunisia	TUNISIE
Unification of formats	Sex	M	Homme
	Sex	F	Femme
	Date	10-Sep-85	10-09-1985
	Date	03-avr-80	03-04-1980
	Civility	Mademoisel	Mlle
Infeasible transformations	Country	Fr	Fr
	Country	NULL	NULL
	Date	1996-10	1996-10
	Country	Cnam	Cnam

Figure 4. Examples of intra-column automatic transformations.

	Number	FirstName	Civility	Sex	BloodGroup	Date	Phone	City	Country	Continent
t01	100	Jean	M	Homme	A+	12-12-1984	33645456932	PARIS	FRANCE	EUROPE
t02	101	Eve	Mme	Femme	B+	10-09-1985	33741256811	**PARIS**	**NULL**	EUROPE
t03	102	Stephane	**M**	**Femme**	AB+	15-03-1965	33644356922	NICE	**FRANCE**	**NULL**
t04	103	Adem	NULL	Homme	AB-	03-04-1980	33645336941	TUNIS	**TUNISIE**	**NULL**
t05	104	Anne	Mlle	0	1+	16-10-1996	33645412944	PÉKIN	CHINE	ASIE
t06	105	Karine	Mlle	0	O-	05-07-1983	33638456945	PÉKIN	**CHINE**	**NULL**
t07	106	Simon	M	Homme	AB-	04-02-1974	33645144031	PARIS	FRANCE	EUROPE
t08	107	Robert	**M**	**NULL**	AB-	04-06-1975	33675404932	**PARIS**	**Houda**	EUROPE
t09	108	Joseph	M	1	AB-	16-10-1996	33775334900	PARIS	FRANCE	EUROPE
t10	109	Karine	Mme	Femme	B+	16-10-1996	33752232901	PARIS	**FRANCE**	**NULL**
t11	110	Martin	**M**	**Femme**	B+	1996-10	33732132421	**PARIS**	**NULL**	NULL
t12	111	Anne	Mlle	Femme	A+	02-03-1928	33732011427	NULL	NULL	NULL
t13	112	Karine	Mme	Femme	B+	16-10-1996	33752232901	PARIS	**FRANCE**	**NULL**
t14	113	NULL	**M**	**Femme**	B+	1996-10	33732132421	**PARIS**	**Fr**	**EUROPE**
t15	114	Yang	Mlle	Femme	A+	02-03-1983	33732011427	**PARIS**	**Cnam**	**EUROPE**
t16	115	Tang	Mlle	Femme	A+	02-10-1990	33632047419	PÉKIN	CHINE	ASIE
t17	116	Yang	**M**	**Femme**	A+	02-10-1995	33632047419	PÉKIN	CHINE	ASIE

Figure 5. Intra-column automatic corrections done on the DS (Patient.CSV).

3.2. Correction of Inter-Columns Anomalies

The correction of intra-column anomalies facilitates the verification of semantic links between the columns which is our ultimate goal. Let S(C) the schema of the DS such as C is the set of columns, X and Y two subsets of columns such as X and Y are subsets of C. We call X functionally determines Y (noted $X \rightarrow Y$) if and only if for all $x_i = x_j$ then $y_i = y_j$. In other words for every value x_i of X, there is only one corresponding value y_j of Y. For instance, the functional dependency Country → Continent can not be verified using the instances given below: {(France, NULL); (Frence, Europe); (China, NULL); (Chine, Asy)}.

Hence, it is necessary to begin with the intra-column corrections. The search space is reduced. **Figure 5** represents the results of data categorization and intra-column correction. Let us note that the source contains inter-columns anomalies. Using the DD some semantic dependencies are not plausible such as Date → City or BloodGroup → City. On the contrary, Civility → Sex, City → Country, City → Continent and Country → Continent are proposed for the verification.

The step before the inter-columns correction is the verification of each dependency constraint. The dependencies constraints verification algorithm (Algorithm 1, **Figure A1**) consists to count the number $ß_i$ of different values of y_i for each x_i. If there is a $ß_i \geq 1$ then the dependence is not verified. The algorithm contains two phases Map and two phases Reduce [16]. The two phases Map1 and Reduce1 take as input the DS and they return the number of occurrences (x_i; y_i, α_i). The two phases Map2 and Reduce2 take the result of the previous two phases as input in order to count the number $ß_i$ of occurrences of each x_i. If $ß_i \geq 1$ then the dependency constraint ($X \rightarrow Y$) is violated. We propose an algorithm (Algorithm 2, **Figure A1**) to correct the dependency anomalies. We use the valid values stored in the DD. Some null values are then processed and corrected according deductions extracted from the data. Let us note that the order of treatment of functional dependencies is very important to complete null values. It is necessary to start with the columns that are less empty. **Figure 6** contains the results of the automatic correction of inter-columns anomalies.

	Number	FirstName	Civility	Sex	BloodGroup	Date	Phone	City	Country	Continent
t01	100	Jean	M	Homme	A+	12-12-1984	33645456932	PARIS	FRANCE	EUROPE
t02	101	Eve	Mme	Femme	B+	10-09-1985	33741256811	PARIS	FRANCE	EUROPE
t03	102	Stephane	M	Homme	AB+	15-03-1965	33644356922	NICE	FRANCE	EUROPE
t04	103	Adem	NULL	Homme	AB-	03-04-1980	33645336941	TUNIS	TUNISIE	AFRIQUE
t05	104	Anne	Mlle	Femme	1+	16-10-1996	33645412944	PÉKIN	CHINE	ASIE
t06	105	Karine	Mlle	Femme	O-	05-07-1983	33638456945	PÉKIN	CHINE	ASIE
t07	106	Simon	M	Homme	AB-	04-02-1974	33645144031	PARIS	FRANCE	EUROPE
t08	107	Robert	M	Homme	AB-	04-06-1975	33675404932	PARIS	FRANCE	EUROPE
t09	108	Joseph	M	Homme	AB-	16-10-1996	33775334900	PARIS	FRANCE	EUROPE
t10	109	Karine	Mme	Femme	B+	16-10-1996	33752232901	PARIS	FRANCE	EUROPE
t11	110	Martin	M	Homme	B+	1996-10	33732132421	PARIS	FRANCE	EUROPE
t12	111	Anne	Mlle	Femme	A+	02-03-1928	33732011427	NULL	NULL	NULL
t13	112	Karine	Mme	Femme	B+	16-10-1996	33752232901	PARIS	FRANCE	EUROPE
t14	113	NULL	M	Femme	B+	1996-10	33732132421	PARIS	FRANCE	EUROPE
t15	114	Yang	Mlle	Femme	A+	02-03-1983	33732011427	PARIS	FRANCE	EUROPE
t16	115	Tang	Mlle	Femme	A+	02-10-1990	33632047419	PÉKIN	CHINE	ASIE
t17	116	Yang	M	Homme	A+	02-10-1995	33632047419	PÉKIN	CHINE	ASIE

Figure 6. Inter-columns automatic corrections done on the DS (Patient.CSV): (Civitiy → Sex, City → Country, Country → Continent).

4. Related Work

We studied and compared the functionalities of some data quality tools such as Talend Data Quality [3] and Pentaho Data Integration [17]. These tools are used to verify only the functional dependencies given by the user. Then, the user must have knowledge about the data schema and the dependencies to be verified. These Tools do not correct anomalies caused by the violation of semantic dependencies between columns. They do not process null values. The different algorithms of discovery of functional dependencies [14] [15] [18] consist to find dependencies across all possible combinations for the different columns of a data source. This increases the size of the search space. These algorithms do not take into account the semantics of data, while, in our approach we focus on contextual data quality. The application of the principle of MapReduce in Big Data will allow to validate algorithms on large volumes of data with good perfermance. Few studies include the principle of MapReduce (distribution of data and treatments).

5. Conclusion

The goal of our work is to contribute to the development of new data integration tools in order to assist the user in the contextual data quality process. Our contribution is to understand the semantic of data before correcting them. In fact, our approach allows the categorization of data by giving them a category and eventually a subcategory. Semantic categorization allows to infer semantic links that can exist between the different columns. The recognition of the structure and semantics of data facilitates the detection and correction of various intra-column, inter-columns and inter-lines anomalies in the same data source. We propose a MapReduce algorithm of the verification of dependency constraints to detect anomalies caused by the violation of these constraints. We present an algorithm for the automatic correction of inter-columns anomalies especially the treatments of null values. The enrichment of the data dictionary will be the subject of our future work.

References

[1] Chu, X., Morcos, J., Ilyas, I.F, Ouzzani, M., Papotti, P., Tang, N. and Ye, Y. (2015) KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing, *Proceedings of the* 2015 *ACM SIGMOD International Conference on Management of Data*, Melbourne, 31 May-4 June 2015, 1247-1261.

[2] Dallachiesay, M., Ebaidz, A., Eldawy, A., Elmagarmid, A., Ilyas, I.F., Ouzzani, M. and Tang, N. (2013) NADEEF: A Commodity Data Cleaning System. *Proceedings of ACM SIGMOD International Conference on Management of Data*, IEEE Press, New York, 22-27 June 2013, 541-552.

[3] Talend. https://www.talend.com/

[4] Ben Salem, A., Boufarès, F. and Correia, S. (2014) Semantic Recognition of a Data Structure in Big-Data. *Proceedings of the* 6*th International Conference on Computational Intelligence and software Engineering*, Vol. 2, Beijing, 11-13 July 2014, 93-103.

[5] Ben Salem, A., Qualité contextuelle des données (2015) Détection et nettoyage guidés par la sémantique des données. Ph.D. Thesis, Université Paris 13, Sorbonne Paris cité, Paris.

[6] Boufarès, F., Ben Salem, A. and Correia, S. (2012) Qualité de données dans les entrepôts de données: Elimination des similaires. *Proceedings*: 8*èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne*, Bordeaux France, 12-13 Juin 2012, 32-41.

[7] Boufarès, F., Ben Salem, A., Rehab, M. and Correia, S. (2013) Similar Elimination Data: MFB Algorithm. *Proceedings of IEEE*-2013 *International Conference on Control, Decision and Information Technologies*, Hammamet Tunisie, 6-8 May 2013, 289-293.

[8] Levenshtein. http://fr.wikipedia.org/wiki/Distance_de_Levenshtein

[9] Jaro-Winkler. http://fr.wikipedia.org/wiki/Distance_de_Jaro-Winkler

[10] Soundex. https://fr.wikipedia.org/wiki/Soundex

[11] Metaphone. https://fr.wikipedia.org/wiki/Metaphone

[12] Zaidi, H., Boufarès, F., Pollet, Y. and Kraiem, N. (2015) Semantic of Data Dependencies to Improve the Data Quality. *Proceedings of the* 5*th International Conference on Model & Data Engineering*, LNCS, Vol. 9344, Springer, Rhodes Greece, 26-28 September 2015, 53-61.

[13] Zaidi, H., Boufarès, F. and Pollet, Y. (2016) Nettoyage de données guidé par la sémantique inter-colonne. *Proceedings*: 16*th Conférence Internationale sur l'Extraction et la Gestion des Connaissances*, Vol. RNTI-E-30, Reims France, 18-22 Janvier 2016, 549-550.

[14] Diallo, T. and Novelli, N. (2010) Découverte des dépendances fonctionnelles conditionnelles. *Proceedings*: 10*th Conférence Internationale sur l'Extraction et La gestion des Connaissances*, Hammamet Tunisie, 26-29 Janvier 2010, 315-326.

[15] Simonenko, E. and Novelli, N. (2012) Extraction de dépendances fonctionnelles approximatives: Une approche incrémentale. *Proceedings*: 12*th Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances*, Bordeaux France, 31 Janvier-1 Février 2012, 95-100.

[16] Dean, J. and Ghemawat, S. (2004) MapReduce: Simplified Data Processing on Large Clusters. *Proceedings of the* 6*th Conference on Symposium on Operating System Design and Implementation*, California, 137-150.

[17] Pentaho Data Integration. http://www.pentaho.fr/explore/pentaho-data-integration

[18] Garnaud, E., Hanusse, N., Maabout, S. and Novelli, N. (2013) Calcul parallèle de dépendances. *Proceedings*: 29e Journées Bases de Données Avancées, Nantes France, Octobre 2013, 1-20.

Appendix

Verification of dependency constraints **algorithm 1** (MapReduce)	Inter-columns corrections **algorithm 2**

Map1(key,value)
// key : data source name
// value : Source contents //set of tuples
Begin
For all tuple in value do
EmitIntermediate("X_i ;Y_{ij} ", "1")// count occurences of X_i ;Y_{ij}
End For
End Map1
 	
Reduce1(key,values)
// key : X_i ;Y_{ij}
//value : a list of counts
int α_i = 0
file f
For all X_i ;Y_{ij} in values do
α_i += ParseInt(X_i ;Y_{ij})
End for
Emit(α_i)
Write(f, α_i)
End Reduce1
 	
Map2(key,value)
// key : f
// value : f contents
For all word X_i in value do
EmitIntermediate(X_i, "1") // count occurences of X_{ij}
End For
End Map2
 	
Reduce2(key, values)
//key : X_i
//values : a list of counts
int $ß_i$=0;
Boolean validDF=0
For all v in values do
$ß_i$ += ParseInt(X_i) // count occurences of X_i
End for
Emit($ß_i$)
If $ß_i$ =1 **then**
validDF=1
End if
End Reduce2
	Begin inter-columns corrections
	Input: DD Data Dictionary S data source, X, Y subsets of columns from S,
	SubCat Dominant subcategory of S
	Output: S' the data source with automatic corrections
	E2 ← CreateV alidDFs(DDV S;X; Y; SubCat)
	S0 ← CorrectionsDFs(S;E2)
	End inter-columns corrections
	
	Function CreateValidDFs
	Input: DDVS,X,Y,SubCat
	Output: E2
	// Get from DDVS the concerned categories X and Y
	E1= {SELECT *ALL FROM DDVS
	WHERE DDVS.CATEGORY = X UNION
	SELECT *ALL FROM DDVS
	WHERE DDVS.CATEGORY = Y}
	// Get from DDVS all correct values (x_i,y_i)
	E2= {SELECT A.Subcat, B.Subcat FROM E1 A, E1 B
	WHERE A.PRIMARYKEY=B.FOREIGNKEY}
	End Function CreateValidDFs
	
	Function CorrectionsDFs
	Input: E2, S
	Output: S'
	For l_j from S (j=1;n) d**o** // l_j tuple from S, n (number of tuples)
	While (S_x[j]6=$DDVS_x$[l]) AND (l ≤ nl) do
	// nl number of tuples of DDVS
	If S_X[j]= DDV S_X[l] **then**
	S_y[j]=DDV S_y[l]
	Else
	l++
	End If
	End While
	End For
	End Function CorrectionsDFs

Figure A1. Inter-columns automatic corrections algorithms.