Automatic Integration Issues of Tabular Data for On-Line Analysis Processing

Yuzhao Yang*, Jérôme Darmont**, Franck Ravat*, Olivier Teste*

* Institut de Recherche en Informatique de Toulouse -IRIT CNRS (UMR 5505)-
Université de Toulouse, 118, Route de Narbonne, 31069 Toulouse Cedex 9, France
{Yuzhao.Yang, Franck.Ravat, Olivier.Teste}@irit.fr,

**ERIC UR 3083, Université de Lyon, Lyon 2
5 avenue Pierre Mendès France, F69676 Bron Cedex, France
Jerome.Darmont@univ-lyon2.fr

Abstract. Companies and individuals produce numerous tabular data. The objective of this position paper is to draw up the challenges posed by the automatic integration of data in the form of tables so that they can be cross-analyzed. We provide a first automatic solution for the integration of such tabular data to allow On-Line Analysis Processing. To fulfill this task, features of tabular data is analyzed and the challenges of automatic multidimensional schema generation should be addressed. Hence, we propose a typology of tabular data and a automatic process based on different steps to integrate one or more data sources.

1 Introduction

Business Intelligence (BI) plays an important role in numerous companies and administrations to efficiently support decision making processes. With the current digitization trend, even small companies and organizations can exploit a large number of data every day and the rise of open data make various data even more accessible. Nevertheless, the implementation of a BI project needs to be realized by people who have the professional knowledge and deep skills in BI technologies such as data warehousing and data visualization. Such projects are also usually expensive and time-consuming. As a result, it is necessary to find a solution to automate the BI process to allow small enterprises, organizations and even individuals without deep technical expertise to easily analyze data. Up to now, there is no platform that achieves this goal.

In current BI systems, data are extracted and stored in a data warehouse to allow On-Line Analysis Processing (OLAP) and visual data rendering. Thus, automating the data warehousing process is crucial to allow non-specialist to exploit such approaches. There exist various forms of data, but most of the data in small enterprises and organizations, as well as most of the open data are in tabular form from spreadsheet software. Although there are commercial BI tools allowing the exploitation of tabular data such as Excel, Qlikview or Tableau, none of them automates the multidimensional analysis of tabular data.

Unlike relational databases that are well-structured and where we can easily retrieve keys, cardinalities and other table metadata, or XML files which sometimes provide schemas such as Document Type Definitions (DTDs), spreadsheet tabular data do not directly provide such information. Moreover, there are different types and structures of tabular data, which makes it difficult to automatically identify labels, values and aggregates. Without input from the user, it is hard to understand the semantics of some data and to choose the appropriate measures for OLAP. Different tables also need to be integrated by matching and mapping. Facing all these requirements and problems, we propose a possible solution in this paper. Table identification is the first significant step we introduce to reach our goal. To carry out this step, we also define an extended typology of tabular data to classify different table types.

The remainder of this paper is organized as follows. In Section 2, we review the related works about automatic multidimensional schema generation and typologies of tabular data, respectively. In Section 3, we define our typology of tabular data. In Section 4, we present our tabular data integration solution, and discuss related perspectives. Finally, in Section 5, we conclude this paper and hint at future research.

2 Related works

2.1 Automatic multidimensional schema generation

OLAP (On-Line Analytical Processing) systems allow analysts to improve decision-making process by consulting and analyzing aggregated data. A multidimensional schema organizes data according to analysis subjects (facts) associated to analysis axes (dimensions). Each fact is composed of measures. Each dimension contains one or several analysis viewpoints (hierarchies). Each hierarchy contains various data granularities of analysis data.

There are different approaches for the generation of multidimensional schemas (Romero and AbellÃş, 2009). There are top-down (also called demand-driven) approaches that start from user requirements and generate the schema to satisfy these requirements manually or automatically. Conversely there are bottom-up approaches that are mostly automatic or semi-automatic solutions to generate the schema from the data source. Moreover, there are hybrid approaches taking both user requirements and the data source into account. Our work focuses on the bottom-up approaches because the users we target do not necessarily know or anticipate precise requirements.

Most of the bottom-up processes are designed for relational databases. (Phipps and Davis., 2002) introduce an automatic data warehouse design approach whose input is an Entity-Relationship (ER) schema. They consider numeric fields as measures, the more numeric fields a table contains, the more likely it is to be a fact, so they create for each table a multidimensional schema by taking the numeric fields in the table as measures. Then, they identify the datetime type data to create temporal dimensions. The fields that are not key, numeric or date of the table become dimensions. Finally, they complete dimensions by verifying if the tables connecting with the fact are in the many side of the relationship and complete hierarchies by verifying the many-to-one relationship of the remaining tables. However, they use queries to evaluate what candidate schemas best meet the userâĂŹs needs, which is done manually. Thus, the process is not fully automated.

(I.-Y.Song et al., 2007) also propose a semi-automatic method to generate star schemas from ER diagrams. The main difference with the approach by (Phipps and Davis., 2002) is that the fact table is chosen by calculating the Connection Topology Value (CTV) of each entity, which is a composite function of the topology value of direct and indirect many-to-one relationships. The selected entities are those that have a CTV higher than a threshold calculated by the mean and standard deviation of the CTVs. In the end, the user must check the redundancy of the time dimension, merge the related dimensions and rename the elements.

There are also methods for other data types. (Golfarelli et al., 2001) propose a semi-automatic method to construct a conceptual data warehouse schema from XML sources. In their approach, schema design is mainly based on DTDs, as they define the elements and attributes and describe the relationships between elements. They simplify the DTDs and create a DTD graph which represent the structure of the XML. Then, based on the facts chosen by the designer and the DTD graph, they build an attribute tree representing the conceptual schema.

Eventually, data mining methods are also used to automate multidimensional schema generation. (Usman et al., 2011) propose to use hierarchical clustering. Their architecture includes a data mining layer to preprocess the dataset and cluster data, and an automatic schema generation layer to generate the multidimensional schema. The solution is implemented and tested with the dataset $ForestCoverType$, but the authors did not specify the similarity metric they use nor the schema generation algorithm.

In summary, existing multidimensional schema generation approaches are designed for specific data sources, such as relational databases and XML documents, and they are not fully automated. There is no such approach specially designed for tabular data. Moreover, the automatic generation of multidimensional schema for tabular data is specific and there are still many challenges to face compared to other data types (Section 1).

2.2 Typologies of tabular data

There is a lot of research concerning tabular data and especially web tables. There are different table classification methods. (Wang and Hu, 2002) and (Crestan and Pantel, 2011) divide web tables into two types: genuine or relational knowledge tables, and non-genuine or layout tables, based on whether tables contain relational knowledge or used for only grouping contents for easy viewing.

We are only interested in genuine tables/relational knowledge tables because non-genuine/layout tables are only used for the navigational or formatting purpose on the web. (Crestan and Pantel, 2011) further divide relational knowledge tables into listing tables, attribute/value tables, matrix tables, enumeration tables and form tables. This classification method is refined and completed by (Lautert et al., 2013), who add new types of relational knowledge tables: concise tables, nested tables, multivalued tables and split tables.

(Yoshida et al., 2001) define nine table types according to whether attributes are arranged vertically or horizontally, whether there is a header and how the header is positioned. These types are presented by a graph. (Milosevic et al., 2016) give a classification of structural tables through dimensionality: one-dimensional tables, two dimensional tables and multidimensional tables that can also be typed into super-row tables and multi-tables. (Chen and Cafarella, 2013) also introduce the concept of hierarchical spreadsheet which contains the header with multi-levels.

All these classifications describe tabular data by different aspects such as the direction of the arrangement of data, the characteristics of cells, the dimensionality of tables, the functionality of tables, etc. However, when considering all table types, none of these classifications are complete. For example, the case of a table whose header is distributed in two rows due to the limit of the space is considered in the classification of (Yoshida et al., 2001), but is not considered by the other people, while the matrix table is not considered in their classification.

3 Augmented tabular data classification

Data are presented differently in various types of tables. How to extract them in different types of tables is thus difficult. Therefore, identifying the type of table is important and requires a complete classification of tabular data. Thus, we propose a new typology of tabular data (FIG. 1) that fuse and complements the classifications from Section 2.2 with respect to three aspects: the structure of tables, the content of cells and headers which are respectively in gray, blue and green in the figure.

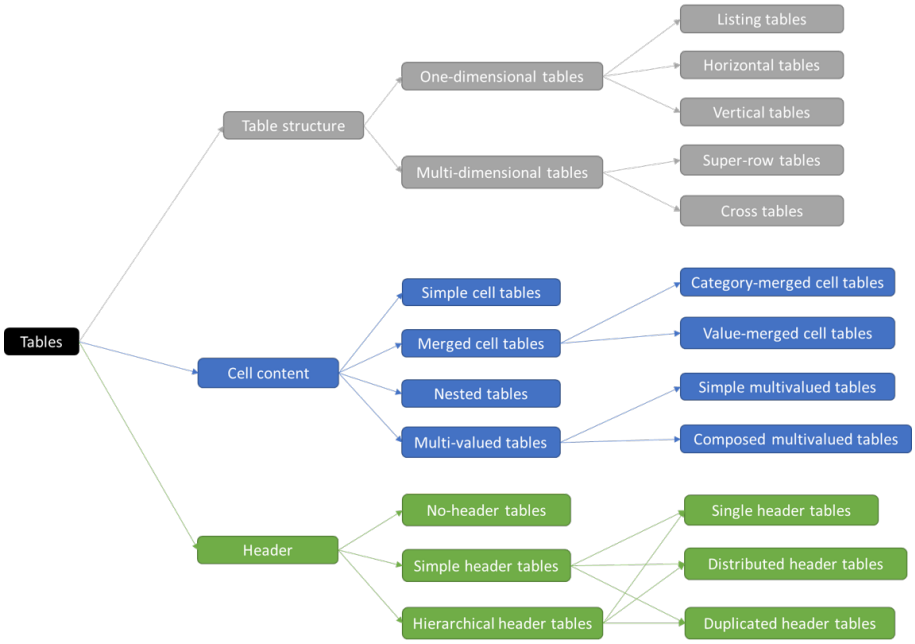


FIG. 1 – *Typology of tabular data*

3.1 Table structure

The structure of tables aspect concerns the dimensionality of data and the arrangement of rows and columns. With respect to the dimensionality of tables, these latter are classified into

one-dimensional tables and multidimensional tables, which also bear sub-types.

	Computer Science		Management	
Score	Database	Programming	Project management	Strategy
Student1	12	13	15	11
Student2	11	16	7	14
Student3	10	8	10	9

FIG. 2 – *Cross table, Hierarchical header table*

1. $One\text{-}dimensional\ tables$ have only one dimension.
   — $Listing\ tables$ list values belonging to a same attribute.
   — In $horizontal\ tables$, data are arranged horizontally, $i.e.$, each column represents one attribute, each row represents a tuple containing values of different attributes.
   — In $vertical\ tables$, data are arranged vertically, $i.e.$ each row represents one attribute, each column represents a tuple containing values of different attributes.
2. $Multidimensional\ tables$ have multiples dimensions.
   — $Super\text{-}row\ tables$ contain multiple dimensions arranged into different level of one row/column, as classified and exampled by Milosevic et al. (2016)
   — $Cross\ tables$ are usually two-dimensional tables where there is one dimension arranged in column and one other arranged in row (Table in FIG. 2). The value of each cell is determined by the dimensions of the column and the row in which it is located. In the cases where there are more than two dimensions, we can have cross tables for different dimensions.

3.2 Cell content

This concerns the characteristics of cell content, as classified and exampled by (Lautert et al., 2013).

1. $Simple\ cell\ tables$ contain only âĂIJnormalâĂİ cells without any of the exceptions listed below.
2. $Merged\ cell\ tables$ contain cells that are gathered or merged into one cell.
   — $Category\text{-}merged\ cell\ tables$ contain cells merged together because they belong to the same category.
   — $Value\text{-}merged\ cell\ tables$ contain cells with the same value merged into one cell.
3. $Nested\ tables\ contain\ tables$ nested into cells.
4. $Multivalued\ tables$ contain cells where there are multiple values.
   — $Simple\ multivalued\ tables$ contain multivalued cells whose values belong to the same domain.
   — $Composed\ multivalued\ tables$ contain multivalued cells whose values belong to the different domains.

3.3 Header

We classify table headers with respect to the following characteristics.

1. $No - header\ tables$ refer to tables without any header.
2. $Simple\ header\ tables$ contain only one-level âĂIJusualâĂİ headers.
3. $Hierarchical\ header\ tables$ contain headers with multiple levels (Table 1). $Simple\ header\ tables$ and $Hierarchical\ header\ tables$ are subdivided with respect to header arrangement.
   — Each header in $Single\ header\ tables$ is arranged in only one single row/column, without repetition.

NameCustomer	**Naline**	GenderCustomer	**Female**
CityCustomer	**Toulouse**	CommandDate	**28/04/2020**
CountryCustomer	**France**	CommandQuantity	**10**

FIG. 3 – *Distributed header table*

   — The header of $Distributed\ header\ tables$ is distributed in different rows/columns (Table in FIG. 3).

Rank	Name	Score	Rank	Name	Score
1	**Alice**	**19**	**5**	**Lisa**	**12**
2	**Jack**	**16**	**6**	**Jeanna**	**10**
3	**Pierre**	**15**	**7**	**John**	**9**
4	**Martin**	**14**	**8**	**Tom**	**7**

FIG. 4 – *Duplicated header table*

   — $Duplicated\ header\ tables$ have headers repeated many times (Table in FIG. 4).

(Zhang and Balog, 2020) define elements of a table, including table page title, table caption, table headings, table cell, table row, table column and table entities. Table page title and table caption are explanatory elements of a table. Table entities concern the content and topic of a table. The other elements are all considered in our proposed classification. Moreover, our classification is based on the state of the art, all the possible cases mentioned in existing tabular data typologies are taken into account. Thus, we believe that this typology is complete.

4 Tabular data integration approach

FIG. 5 illustrates our approach which aims to integrate automatically tabular data for the OLAP analysis. We first extract and identify tables (Section 4.1). The data source may contain one or several tables. If there is only one table, we transform the table type (Section 4.2), select the measures (Section 4.3), detect functional dependencies (Section 4.4) and finally generate

the multidimensional schema (Section 4.5). If there are several tables, we either match the tables into one table by using by similarity measures and proceed as above, or generate a multidimensional schema for each table as above and match the schemas.

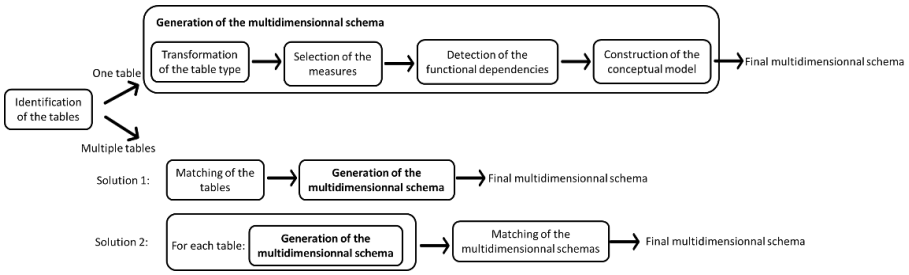


FIG. 5 – *Overview of our tabular data integration approach*

4.1 Table identification

Since the user we target are individuals, small enterprises or organizations without knowledge of BI, data sources may be of various types, e.g. CSV files, spreadsheet files, HTML tables or even images. There may also be several tables in some sources. Therefore, the first thing to do is to identify the sources and extract the tables within. Many table extraction methods are surveyed in (Zhang and Balog, 2020), especially for web tables. There also exist deep learning methods to detect tables from images (Schreiber et al., 2017); (Paliwal et al., 2019). Let us first discuss our integration solution for the one-table case. We generalize it in Section 4.6.

4.2 Table type transformation

First, we should transform multidimensional source tables into a one-dimensional table. (Milosevic et al., 2016) propose an helpful method allowing the identification of multidimensional table structures. However, multidimensional tables may already contain obvious multidimensional structures that we can employ directly for the construction of a multidimensional schema. If this was the case, transforming the multidimensional table would be useless and time-wasting. In this article, we propose to perform the transformation to get a uniform representation of tables so that we can easily get the attributes and values for the following steps of our approach. We consider multidimensional tables as a special case that we will process specifically.

For tables containing non-simple cells, it is necessary to convert these cells into simple cells by decomposing merged cells and multivalued cells by adding new cells or columns. Distributed and duplicated headers need to be detected by the algorithm of (Yoshida et al., 2001) in order to reform the table as a single-header table. In hierarchical header tables, hierarchical relations should be identified as candidates to hierarchies in the multidimensional model. Hierarchies can be extracted by the method proposed by (Chen and Cafarella, 2013). Eventually, there may sometimes be even no attribute header in a table. In this case, we can use column

identification methods (Zhang and Balog, 2020) to retrieve the semantic label of columns by matching them with knowledge bases.

4.3 Measure selection

To integrate a one-dimensional table for OLAP, we must firstly define measures. Measures are usually obtained by aggregation and calculation (max, min, countâĂę), so most of them are numeric data. Normally, during the data warehouse design process, facts and measures are chosen by the user. As we aim to automate this process, we choose all numeric attributes in tables as candidate measures. However, not all numeric data can be considered as measures. For example, some numeric data may be IDs or Booleans. Yet, we can use the semantic labeling of numeric data to determine whether an attribute can be treated as a measure. (Alobaid et al., 2019) propose algorithms to identify the type of numeric data, including nominal data, ordinal data, intervals and ratios. Intervals and ratios are more likely to be measures.

This automatic process can provide candidate measures to the users, but we may wrongly choose or omit some measures. For instance, a measure calculated by the count of an attribute does not need to be numeric. Calculating measures is also a big problem, because without knowing the exact user requirements, we cannot know the exact aggregation function. We can only propose basic functions such as count, sum, average, minimum and maximum, while measures may also be calculated by complex formulas on one or several attributes. Thence, if the user has specific requirements, we should also let her/him specify measures and correct the candidate measures we propose automatically.

4.4 Functional dependency detection

The objective of this step is to determine the multidimensional modelâĂŹs dimensions and associated hierarchies. To meet these needs, we apply the principles of functional dependencies between attributes. There exists a lot of algorithms to discover functional dependencies (Liu et al., 2012); (Papenbrock et al., 2015). For a relation schema R, X and Y represent a subset of the attributes of R. In a statement of functional dependency $X \rightarrow Y$, X is usually called the left-hand side and Y is called the right-hand side. We are only interested in the functional dependencies that have one attribute only in the right-hand side and get the minimal cover which is the minimal set of functional dependencies being able to infer all the functional dependencies. Hence, we obtain the elementary functional dependency such that no strict subset of the left-hand side determines the right-hand side. Moreover, we delete all the transitivity and pseudo-transitivity dependencies. We are also only interested in the functional dependencies whose left-hand side has one attribute only, since there is one attribute in each level of a hierarchy.

To make sure that the functional dependencies that we discover conform to the dependency relationship of attributes in the real world, we hypothesize that there is enough data in terms of quantity and variety so as to represent real dependency relationships. Moreover, there should be no error in data nor empty data, but if this was the case, we could detect approximate functional dependencies. There are many methods for approximate functional dependency (Liu et al., 2012), which propose different error measures. We can set a satisfaction threshold to decide the approximate functional dependencies. We can also search and complete empty data by querying open data (Eberius et al., 2012).

4.5 Multidimensional schema detection

Given all the functional dependencies that satisfy our requirements, we can connect them to infer all hierarchies and even draw a functional dependency diagram to visualize them. For example, let us consider a product order table with attributes $idCustomer$, $nameCustomer$, $cityCustomer$, $countryCustomer$, $classCustomer$, $idProduct$, $nameProduct$, $categoryProduct$ and $quantity$. With quantity chosen as the measure, we get a fact $(F_1, quantity)$. We may get the functional dependencies by applying the algorithm of functional dependency detection if the data of the table satisfy the hypothesis mentioned in the section 4.4: $idCustomer \rightarrow nameCustomer$, $nameCustomer \rightarrow cityCustomer$, $cityCustomer \rightarrow countryCustomer$, $nameCustomer \rightarrow classCustomer$, $idProduct \rightarrow nameProduct$, $nameProduct \rightarrow categoryProduct$. We can then obtain three hierarchies, but we cannot know whether an attribute is a parameter or a weak attribute, so we consider all the attributes as parameters. Thus, the three hierarchies are: $(H_1, < idCustomer, nameCustomer, cityCustomer, countryCustomer >), (H_2, < idCustomer, nameCustomer, classCustomer >), (H_3, < idProduct, nameProduct, categoryProduct >)$. The dependency graph is provided in FIG. 6:

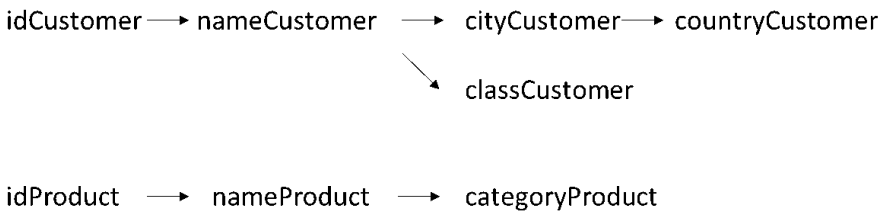


FIG. 6 – *Dependency graph of the example*

All the hierarchies that share the same root have the same dimension, as we can see in FIG. 6, we can get 2 dimensions: $(D_1, \{idCustomer, nameCustomer, cityCustomer, countryCustomer, classCustomer\}, H_1, H_2)$,$(D_2, \{idProduct, nameProduct, categoryProduct\}, H_3)$

The final step of our approach is to simply link measures to dimensions. We do not need to verify whether measures are dependent on the totality of the root parameters, because measures are obtained by the aggregation of attributes. If we aggregate all root parameters, measures are certainly dependent on the totality of the root parameters, since there is no redundancy after aggregation. Therefore, we obtain a conceptual multidimensional model $(nameSchema, F_1, \{D_1, D_2\})$.

If there are equivalent attributes, $i.e. X, Y$ are subsets of a relation schema R, X and Y are equivalent attributes if $X \rightarrow Y$ and $Y \rightarrow X$, there is probably a parameter and its weak attributes inside an equivalent attribute set, so we should know what attribute may be an ID so that it can be considered as a parameter. We could also introduce semantic solutions for solving these problems.

4.6 Integration of multiple source tables

Our approach currently applies to the one-table case. In case of multiple tables, we propose two potential matching solutions, by using different schema matching and data matching technologies with the help of the similarity measures. We can either match all tables into one single table and generate automatically the multidimensional model of the table, or we can generate automatically the multidimensional model of each table and match these schemas to one single schema. The fusion can also be described by a manipulation algebra (Ravat et al., 2005).

5 Implementation

Since we are still studying the integration of multisource tables, we have only implemented the automatic multidimensional schema generation algorithm for one table source in Python. To illustrate our approach, we test with a csv file from the open data site of the French government [1] . This is a file recording speaking time of men and women corresponding to more than a million hours of programs broadcast from 1995 to February 28, 2019.

	A	B	C	D	E	F	G
1	media_type	channel_name	is_public_channel	year	women_expression_rate	speech_rate	nb_hours_analyzed
2	radio	Chérie FM	False	2002	47.10994424	15.73869436	718
3	radio	Chérie FM	False	2003	46.03444471	16.25025819	1617
4	radio	Chérie FM	False	2004	48.38360747	15.03544979	1644
5	radio	Chérie FM	False	2005	45.45162675	16.06377772	1624
6	radio	Chérie FM	False	2006	47.81930726	15.6028421	1604
7	radio	Chérie FM	False	2007	43.94302035	15.67245605	1571
8	radio	Chérie FM	False	2008	51.39922002	17.53800278	1663
9	radio	Chérie FM	False	2009	49.95436887	15.93191099	1677
10	radio	Chérie FM	False	2010	48.06286688	15.94817147	1585
11	radio	Chérie FM	False	2011	47.83215331	17.98112396	1601
12	radio	Chérie FM	False	2012	51.89108386	18.37799131	1636
13	radio	Chérie FM	False	2013	54.11972167	20.89188492	1658
14	radio	Chérie FM	False	2014	50.30261056	18.14301169	1699
15	radio	Chérie FM	False	2015	51.22796482	22.28877115	1616
16	radio	Chérie FM	False	2016	51.17050513	21.62808475	1599
17	radio	Chérie FM	False	2017	53.08764883	19.57975861	1829
18	radio	Chérie FM	False	2018	52.14109768	18.50494497	2753
19	radio	Chérie FM	False	2019	48.67531557	18.21716099	1139

FIG. 7 – *Test file excerpt*

As illustrated in FIG. 7, the file contains 7 columns: media type, channel name, is public channel, year, women expression rate, speech rate and number of the analyzed hours. There are 21 radio stations and 34 TV channels recorded in the file. The result of our algorithm is shown in FIG. 8.

We detect 3 measures: women expression rate, speech rate and number of the analyzed hours. We get 3 hierarchies, since there are 2 hierarchies sharing the same root, they are in the same dimension. So, we get 2 dimensions.

1. https://www.data.gouv.fr/fr/datasets/temps-de-parole-des-hommes-et-des-femmes-a-la-television-et-a-la-radio/

```
Hierarchies :
['channel_name', 'media_type']
['channel_name', 'is_public_channel']
['year']

Mesure :
['women_expression_rate', 'speech_rate', 'nb_hours_analyzed']
```

FIG. 8 – *Test file excerpt*

The user may also need the men expression rate, she/he can just add this measure by a simple calculation (1 âĂŞ women expression rate). The user can also give a name to the dimensions, then we can get the multidimensional schema (FIG. 9).

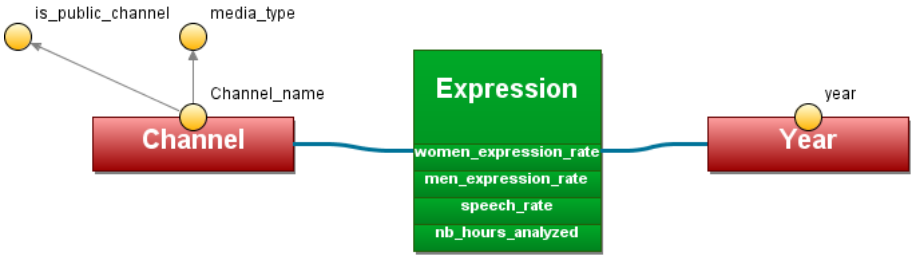


FIG. 9 – *Final multidimensional schema*

6 Conclusion and future research

In this paper, we study how to automate the integration of tabular data to generate multidimensional data warehouses. First, we propose a taxonomy of tabular data to help in the table identification step. Then, our table identification solution consists in identifying the table, transforming it into a one-dimensional table and generating the multidimensional schema by exploiting functional dependencies. Matching the tables then generating a multidimensional schema or generating a multidimensional schema for each table and then matching the schemas are two possible solutions for the integration of multiple tables.

In ongoing work, we will implement and refine the idea that we propose. We will test the two possible methods for the generalization of multiple tables and evaluate their feasibility and performance to decide the one to use. Once we succeed in generating automatically the conceptual multidimensional schema, we will define an automatic process to populate the schema from the data and the content of the source table.

Acknowledgements

The research depicted in this paper is funded by the French National Research Agency (ANR), project ANR-19-CE23-0005 BI4people (Business Intelligence for the people).

References

Alobaid, A., E. Kacprzak, and O. Corcho (2019). Typology-based semantic labeling of numeric tabular data. *Semantic Web 1*(0), 1–5.

Chen, Z. and M. Cafarella (2013). Automatic web spreadsheet data extraction. In *SSW '13: Proceedings of the 3rd International Workshop on Semantic Search Over the Web*, pp. 1–8.

Crestan, E. and P. Pantel (2011). Web-scale table census and classification. In *WSDM '11: Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 545âĂŞ554.

Eberius, J., M. Thiele, K. Braunschweig, and W. Lehner (2012). Drillbeyond: Enabling business analysts to explore the web of open data. In *Proceedings of the VLDB Endowment*, Volume 5.

Golfarelli, M., S. Rizzi, , and B. Vrdoljak (2001). Data warehouse design from xml sources. In *DOLAP '01: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP*, pp. 40âĂŞ47.

I.-Y.Song, R. Khare, and B. Dai (2007). Samstar: A semi-automated lexical method for generating star schemas from an entity-relationship diagram. In *Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP*, pp. 9–16.

Lautert, L. R., M. M. Scheidt, and C. F. Dorneles (2013). Web table taxonomy and formalization. *ACM SIGMOD Record 42*(3), 28âĂŞ33.

Liu, J., J. Li, C. Liu, and Y. Chen (2012). Discover dependencies from dataâĂŤa review. *IEEE Transactions on Knowledge and Data Engineering 24*(2), 251âĂŞ264.

Milosevic, N., C. Gregson, R. Hernandez, and G. Nenadic (2016). Disentangling the structure of tables in scientific literature. In *Natural Language Processing and Information Systems*, Volume 9612, pp. 162âĂŞ174.

Paliwal, S., V. D, R. Rahul, M. Sharma, and L. Vig (2019). Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 128âĂŞ133.

Papenbrock, T., J. Ehrlich, J. Marten, T. Neubert, J. Rudolph, M. SchÃűnberg, J. Zwiener, and F. Naumann (2015). Functional dependency discovery: An experimental evaluation of seven algorithms. In *Proceedings of the VLDB Endowment*, Volume 8, pp. 1082âĂŞ1093.

Phipps, C. and K. C. Davis. (2002). Automating data warehouse conceptual schema design and evaluation. In *Proceedings of 4th International Workshop on Design and Management of Data Warehouses*, pp. 23–32.

Ravat, F., O. Teste, and G. Zurfluh (2005). Manipulation et fusion de donnÃl'es multidimensionnelles. *Revue des Nouvelles Technologies de lâĂŹInformation (RNTI-E-3) - Extraction et Gestion des Connaissances I*, 349âĂŞ354.

Romero, O. and A. AbellÃş (2009). A survey of multidimensional modeling methodologies. *International Journal of Data Warehousing and Mining 5*(2), 1âĂŞ23.

Schreiber, S., S. Agne, I. Wolf, A. Dengel, and S. Ahmed (2017). Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1162âĂŞ1167.

Usman, M., S. Asghar, and S. Fong (2011). Hierarchical clustering model for automatic olap schema generation. *Journal of E-Technology 2*(1), 9âĂŞ20.

Wang, Y. and J. Hu (2002). A machine learning based approach for table detection on the web. In *Proceedings of the 11th International Conference on World Wide Web*, pp. 242âĂŞ250.

Yoshida, M., K. Torisawa, and J. Tsujii (2001). A method to integrate tables of the world wide web. In *Proceedings of the 1st international workshop on Web document analysis*, pp. 31–34.

Zhang, S. and K. Balog (2020). Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology 11*(2), 1–35.

Résumé

Les entreprises et les particuliers produisent de nombreuses données tabulaires. L'objectif de cet article est de mettre en avant les défis posés par l'intégration automatique des données tabulaires pour effectuer des analyses en ligne (OLAP). Pour remplir cette tâche, les caractéristiques des données tabulaires doivent être analysées et les défis de la génération automatique de schémas multidimensionnels doivent être relevés. Par conséquent, nous proposons une typologie des données tabulaires et une démarche automatique basée sur différentes étapes pour intégrer aussi bien une source de données que plusieurs.