Automatic generation of Data Warehouse Instances from Requirements

Carlos Pizarro, Gabriel Novillo Rangone, German Montejano & Ana Garis

Universidad Nacional de San Luis, Argentina

kpizar@gmail.com gnovillorangone@gmail.com
gmonte@unsl.edu.ar agaris@unsl.edu.ar

Abstract. Data Warehouse projects in organizations as a basis for obtaining information are having a great development today. The maturity of generation methodologies has reached significant acceptance. This allows us to focus on addressing the study of the lack of adaptability to the high dynamics of the changes in requirements of this type of projects. Data Science and Natural Language Processing allow us to integrate areas of knowledge and provide tools to solve this problem. A methodological model is proposed to partially automate the stages of generation of Data Warehouse instances based on requirements.

Keywords: Data Science, Data Warehouse, Natural Language, Business Intelligence.

1 Introduction

The different organizations are in search of obtaining information from the large volume of data that is currently stored. Data Warehouses (DWH) and Data Mining (DM) have supported the Business Intelligence (BI) area to respond to information requirements for decision-making. The indicators obtained have made it possible to optimize planning at the management level of the same.

As for the DWH, they have had a wide dissemination in terms of generation methodologies, which has consolidated them and is used in most of the projects that are started.

Thus it can be seen how these methodologies are being applied in the most diverse domains, Educational Establishments, Health Centers, Economic Studies Centers and all kinds of Companies, which advance in management and planning based on information. In this type of project, they seek to have the largest possible volume of data for analysis. The existing methodologies allow us an appropriate design of the DWH and the definition of the different models to determine knowledge [1, 2, 3, 4, 8]. We propose use fundamentals on DM, BI, Data Quality, Data Integration and Natural Language (NL) in order to specify a Model for the generation of DWH instances from the processing of requirements expressed in NL, with the aim of providing a solution to the lack of flexibility in the face of new DWH requirements.

1.1 Data Quality and Data Integration

Applying data quality concepts has helped solve many problems that arise in the early stages of creating a DWH. Data quality is a multidimensional concept that encompasses different aspects depending on the needs of data consumers or system designers. A set of data quality dimensions serve as a benchmark for measuring data quality. Therefore, Data Warehousing Institute (DWI) proposes 6 fundamental dimensions for data quality management: Accuracy, Completeness, Consistency, Timeliness, Uniqueness and Validity [4, 15, 16, 17].

In addition to data quality, data integration must also be considered since it is responsible for allowing users to access data stored in heterogeneous data sources, presenting a single, unified view of that data [13, 14].

It is the process that allows us to combine heterogeneous data from many different sources in the form and structure of a single application. This makes it easier for different types of information, such as databases, files, and spreadsheets, to be merged with specific processes. Generally, buffers are used where integration takes place and measurable and quality data is generated from which the final DWH is designed and implemented [10, 11, 12].

Often, a data integration project involves accessing data from all sources and locations whether it is on premises, in the cloud, or a combination of both. Integrating data implies that records in one data source map to records in another. It is a type of data preparation essential for analytics and for using the data successfully.

1.2 Natural Language

Natural Language Processing (NLP) is the field of knowledge of Artificial Intelligence that deals with investigating the way machines communicate with people through the use of NL, such as Spanish or English. Limitations of economic or practical interest mean that only the most widely spoken or used languages in the digital world have applications in use [8, 18, 19].

The components of NLP depend on the purpose of the application, namely, some types could be excluded of a NLP task. Therefore, a NLP task, typically includes Morphological or lexical analysis, Syntactic analysis, Semantic analysis, and Pragmatic analysis.

Separate the elementary pieces of language. Often, NLP tasks divide language into shorter elementary pieces, attempt to understand the relationships between the pieces, and explore how the pieces work together to create meaning. These implicit tasks are often used on higher-level NLP resources, such as content categorization, discovery and modeling of themes, contextual extraction, sentiment analysis, speech-to-text-summarization of documents and machine-based translation.

The separation of the elementary pieces of the language that the PLN makes, allows us to focus its use in the requirements processing stage. In BI, decision makers constantly carry out their requirements in NL and with great dynamics. Being able to identify and separate by examples nouns, verbs and types of words of a requirement, would allow us to process it automatically [8, 18, 19].

1.3 Related Work

So far, no investigations of new methodological proposals have been found in order to solve flexibility problems in the face of new requirements. Although there are some related works that have focused on automating processes, these are carried out in the definition and creation stages of the DWH, without taking into account the adaptive maintenance stage, as is done in this work.

Moukhi et al. [20] analyze the set of requirements-based methods for designing data warehouses and classify them into two broad categories: user-based methods and goal-based methods. Subsequently, they propose a new method called XCube Assist that allows to take advantage of all the advantages of the user-oriented approach.

Winter et al. [21] perform an information requirements analysis for data storage systems, differing significantly from the requirements analysis for conventional information systems. Its comprehensive methodology supports the entire process of determining the information requirements of DWH users, matching the information requirements with the supply of real information.

Phipps et al. [22] propose algorithms for the development and automatic evaluation of conceptual schemes. They present a creation algorithm using an operational database business schema as a starting point for source-based DWH schema design. Candidate conceptual schemas are created using the ME/R model, extended to indicate where additional user input can be used to further refine a schema.

Song et al. [23] present the SAMSTAR method, which semi-automatically generates star diagrams from an Entity Relationship Diagram (DER), analyzing their semantics and structure. The novel features of SAMSTAR are the use of the Connection Topology Value notion to identify the candidates for facts and dimensions.

Nazri et al. [24] describe a methodology for developing a dimensional DWH by integrating the three development approaches: supply-driven, goal-driven, and demand-driven. By having the combination of all three approaches, the final design will ensure that user requirements, business interest, and existing data source are included in the model. They propose an automatic system using ontology as the domain of knowledge. From an operational DER, the fact selection table, the term check, and the consistency check will use the domain ontology.

Most of the research lines found have as their main strategy to automate processes, with emphasis on the stages of definition and creation of the DWH, either starting from the storage designs, such as DER, or from the requirements already starting from there, relate them to the storages to achieve a data storage design that covers all the information needs. Unlike the investigations described above, the present work proposes a new Model. Said Model focuses on the adaptive maintenance necessary in the software development process, which allows that in the face of new requirements (nonexistent in the initial definition) it is possible to adapt the data storage minimizing time, costs and work, automating some stages of the process, and generating instances from the use and integration of Data Science tools. Specifically, we propose to define a Model that allows generating a DWH from requirements expressed in NL. Therefore, we specify a Model for the automatic derivation of DWH instances from the processing of requirements expressed in NL, with the aim of providing a solution to the lack of flexibility in the face of new DWH requirements.

2 Scope of the problem

The BI projects consolidated the existing methodologies for creating DWH. The increasing emergence of specific tools was the basis for the development of the data science industry. The management level of the organizations determines information objectives that are requirements (input) for the design and implementation of the DWH [5, 6, 7]. This process is characterized by being long and expensive. It highlights a problem regarding the DWH's ability to adapt to changes in requirements.

The dynamics of organizational management translated into changes in the requirements produces an important impact, since, as a maximum consequence it would imply having to redesign and re-implement the DWH, this lack of adaptability in some opportunities calls into question the execution of these projects.

For instance, we can mention the impact that the request for new KPIs (Key Performance Indicator) [2, 8, 9] would have by the managers of an organization, which must be validated: verify if they are possible to achieve at Based on the design of the existing DWH, if the impact is less since it would only require incorporation into the visualization tool and some modification to the data processes. But what if the new KPIs cannot be derived from the existing DWH design? If the DWH does not contain the data to respond to a new requirement, a process is triggered that among other steps requires, analyze if the data sources contain the data to respond to the new requirement, if so, analyze the incorporation to the design of the DWH, implement the extract, transform and load (ETL) to feed the DWH to finally include the new requirement to the data visualization.

The complexity could be increased if the buffers lack the necessary data to respond to the new requirement. Thus, data sources must be analyzed to verify if it is possible to incorporate the new need, perform the quality analysis of the data and start the entire process until the data is available in the DWH and in the visualization tool. It is clear that the high cost produced by this lack of flexibility of DWH projects, which, in many cases, causes them to be abandoned or limited, this leads to the information responses of the Requirements are processed outside the DWH, which finally gradually becomes totally or partially limited in its use. Usually, these answers requires keeping numerous specialists working for long periods of time assigned to eventual tasks, producing a marked increase in the cost of the projects. Note that requirements analysis, data quality, ETLs, DWH designers and implementers as well as those in charge of data visualization, are generally part of a large team of qualified specialists that make up large teams of job.

3 Work Hypothesis

Added to the lack of adaptability of DWH to the new requirements is the great dynamics of the management of modern organizations. This is why a model of automatic generation of DWH instances is proposed [2].

For the purposes of obtaining information and trying to minimize the impact on time and costs in the adaptation of the DWH, this hypothesis is based on the maturity reached

by the DWH design methodologies and the different IT areas that can be used in the different stages in the generation of data warehouses.

Data Quality and Integration applied in the early stages of construction provide an important basis for achieving consolidated buffers from heterogeneous sources that are the basis for enabling better data warehouse designs. Additionally, the use of NL processing in the requirements analysis stage allow us treating them automatically, producing the separation into shorter elementary pieces, in order to identify entities and relationships from which to implement managerial indicators.

The lack of flexibility of the DWH produces frustration in decision makers since the DWH has little adaptability with respect to the dynamism of the organization. In this framework, combining all these tools, applying each one at the correct stage of the project, would allow us to create an automatic DWH analyzer and generator, which has a greater capacity to absorb changes in management requirements generated in NL, reducing time and implementation costs. This would allow teams to dynamically incorporate new data into the DWH and its creation in a more agile and efficient way.

4 Proposed Model

The proposed model allow to receive new requirements and to analyze whether there are data sources in the buffers and, if so, automatically to generate the DWH design. Thus, it produces different instances of a DWH in a more agile way.

As shown in Fig 1. the proposed model includes two main stages.

In the first stage, the model starts the process by receiving the new requirements through an interface, expressed in NL by those responsible for making decisions. Requirements should be expressed, trying to make them express as much as possible the data sources in terms of objects and the desired units of measurements. This will allow for greater efficiency in the identification process. Supported by the PLN algorithms, the words that could potentially be used to evaluate the existence of source data is identified and selected. Words that are nouns, verbs and those that express measurement units (amounts, amount, etc.) are selected and presented in a list to be analyzed by the DWH designer. Of the nouns, those that, due to their empirical knowledge, already exist in the structure are discarded. The designer can then start the process of verifying the existence of source data from the intermediate storage that can be selected as a functionality of the interface.

The second stage begins with the selection in the interface by the designer of the intermediate storage and then the automatic process searches for entities, relationships, keys and fields that generate the instance of the DWH. The search is carried out on the intermediate storage where, in a previous process, quality data is obtained. The result of this process is shown to the designer through the interface, where he determines if the number of entities, relationships and fields found reach the objectives of the requirements and is proceed to the construction of the DWH. The architecture that will have the new design will be a star.

Once the creation of the DWH instance is completed, it is possible to work on the presentation of the data as a final stage to be able to respond to the requirements. According to the visualization tool, the most appropriate type of table or graph is selected

to represent the requested indicators. The designer manually fills the DWH, defining the ETLs and the most convenient update strategies.

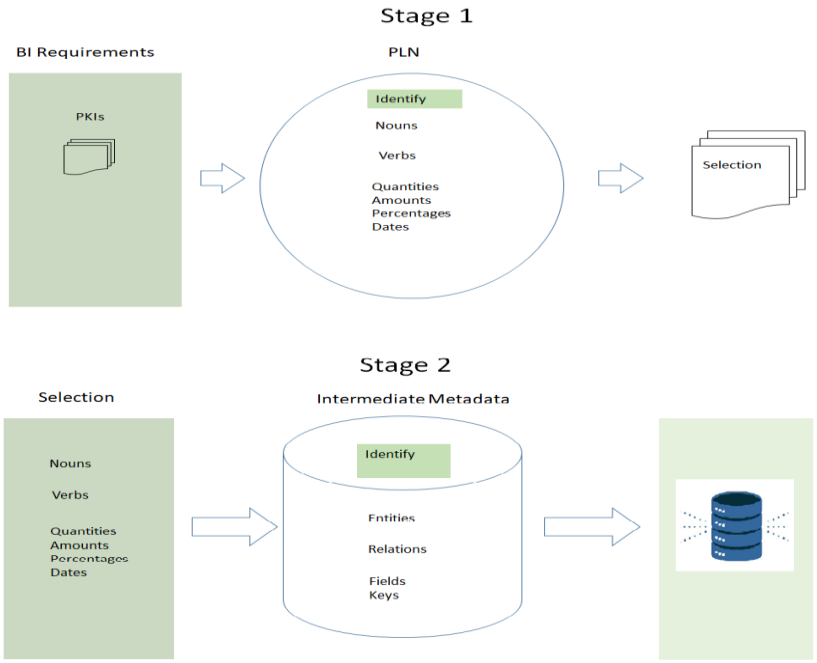

Fig. 1. Stage 1 and Stage 2 of the Proposed Model

We complete the proposal with a fundamental stage that enables the automatic generation of DWH instances. This stage is a prerequisite and is carried out in the first creation of the DWH in order to achieve intermediate storages that are not only limited to supporting the initial requirements, but also to covering the greatest amount of information possibilities. Note that it is not intended to have an intermediate repository with a very high volume of data or with information that has a very low probability of use, keeping this repository up-to-date may require a high upgrade effort. A balance must be found in terms of the amount of metadata to store since this can impact the performance of ETLs. The update frequency must be taked into account when designing the strategy for applying the tasks. However, despite the cost of updating, the volume of the intermediate repositories does not impact the performance of the DWH since not only the data that meets the requirements of the instance.

In Stage 0 (Fig. 2) Data Quality and Data Integration lines play a fundamental role in having Quality buffers. We already mentioned the dimensions, the integration of heterogeneous data and the hard work that this stage implies, which is directly impact the possibility of success of the model.

The more information on the problem domain exists in the buffers, the more successful results the model will find to automatically respond to the creation of DWH instances to respond to a broad set of requirements.

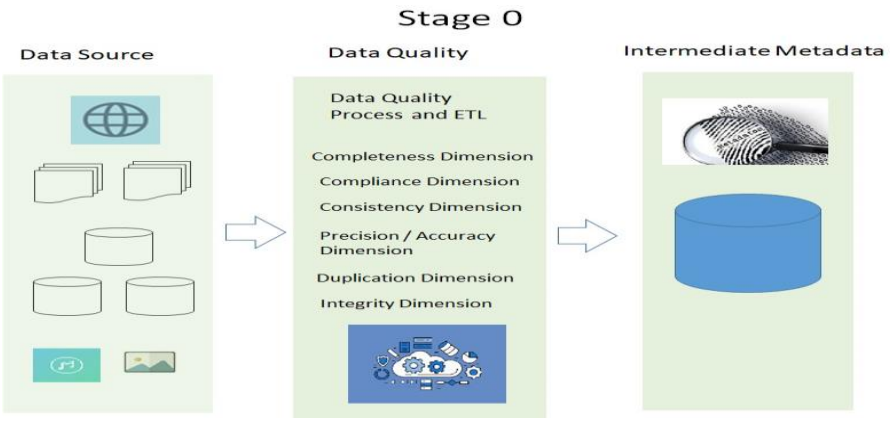

Fig. 2. Stage 0 of the Proposed Model

To achieve greater success in the search for information for each instance and because it is an automatic search and creation, the definition of the nomenclature of the entities and relationships in the buffers takes on a relevant importance. When designing the repositories it is very valuable keep in mind the keywords of the requirements or possible indicators of the domain. Finally, Fig. 3 shows the complete scheme of the model to generate automatic instances where all the stages can be viewed based on the requirements.

5 Validation

To validate the proposal, the Model is evaluated in the domain of education. In particular, it is used to generate a DWH in the scope of an Argentine national university.

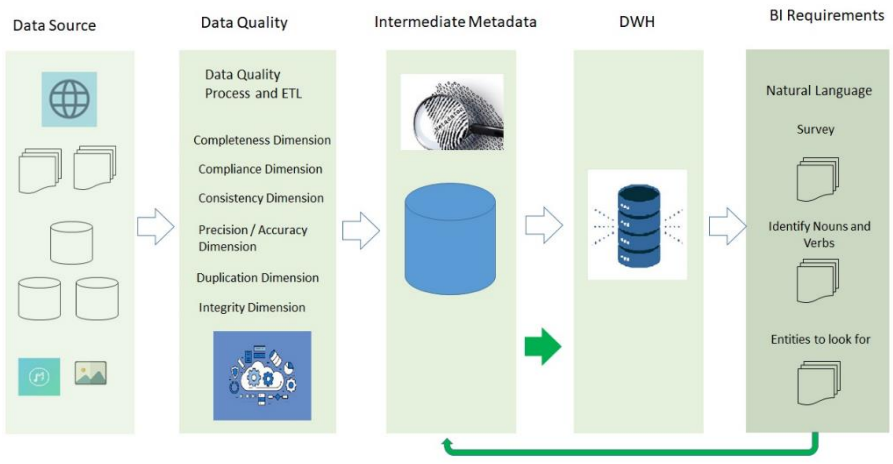

Fig. 3. Representation of the Proposed Model

Most of the Argentine national universities agreed to use the same Information Management System provided by the Ministry of Education of the Nation. For this purpose, we developed a prototype that executes the stages proposed in the model. The prototype architecture is divided into two large layers, the front-end and the back-end, as illustrated in Fig. 4.

The front-end is the view and the prototype controller. The view provides the graphical interface for the administration of requirements as well as the interface of interaction between the engineer and the process of automation of instances. It was developed in Java Script, with Bootstrap framework which provides powerful JavaScript plugins and style sheets (CSS), and the ANGULAR framework, which allows to separate the front-end and the back-end in the application thanks to its MVC pattern (Model-View-Controller).

On the back-end is the database manager and the processes for handling the requirements in NL. The Database Manager used by the prototype is SQL Server 2019. SQL Server allows working directly with Python in an embedded way. We execute Python scripts from the database engine. The prototype processes in the back-end the requirements entered in the prototype by means of Python script invoking libraries (NTLK) for NL processing.

The execution of the prototype begins with the incorporation of new requirements through the interface. This requirements are processed in the back-end with a Python script that allows us to identify the possible entities. These are returned to the front-end and presented in the interface. Therefore, the execution of their processing is started. It consists of an exhaustive search in the buffers to verify that the candidate entities exist and can be included into the new instance. In the interface, a list of connectors (ODBC) is defined to choose the intermediate source and perform the processing. At this stage, it is done in the data dictionary of the selected database, all with SQL language.

5.1 Execution Results

We describe the main conclusions of the prototype executions.

• The execution of the prototype generated a new instance of the DWH.

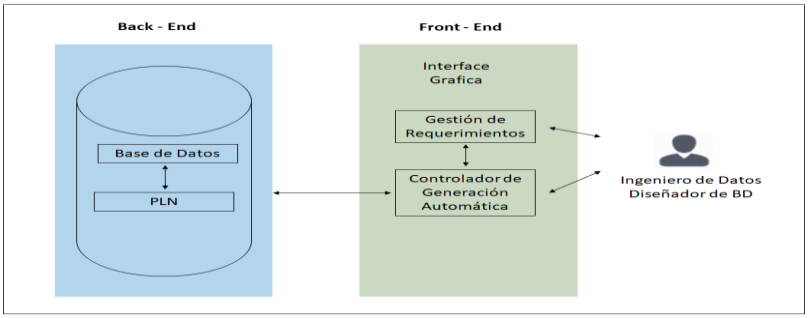

Fig. 4 Prototype architecture

- The performance of processing requests through Python was very good. The integration achieved between the SQLServer 2019 engine and Python for the execution of the scripts evaluating computational performance was excellent, no measurable differences were found using Python in a non-integrated way.
- The model will better show its capabilities when working with large data warehouses, where finding entities is much more expensive. The exhaustive search performance performed on SQL buffers in the SQLServer 2019 engine was very good. However, it can be not definitive since the executions were carried out with a small database.

5.2 Automation vs Flexibility

The development of the prototype and its execution allowed to draw some remarks. The degree of automation can be decided in the construction of the prototype. For a high degree of automation we must incorporate behavior rules into the prototype, some of the necessary rules would be:

• Define the treatment of requirements, such as what to do with repeated requirements or weigh the importance of the requirements.

• Define if the instance is going to be created from scratch or if it is going to be incorporated into an existing instance.

• In case of incorporating entities to an existing instance, define what to do with the entities that are already in the instance.

• Define the connection objects to the different work storage prior to execution as parameters.

We should incorporate more functionality into the prototype in order to increase flexibility. With the participation of the designer and the percentage of automation is lost and greater flexibility is achieved when interacting through the interface with the process. We consider that the model implemented through a tool with the participation of the designer will be able to achieve better performance in the treatment of large data warehouses.

6. Conclusions and future work

In this work, we have presented a model to generate DWH instances from requirements expressed in natural language. This proposal is based on the lack of adaptability of the DWH to new requirements and on the high cost of their redesign processes, which causes underutilization or abandonment of this type of projects. Therefore, in order to achieve the objective of adaptability and provide data analysts or people who make decisions in organizations for a better use of their data warehouses, this model is presented. Automation will allow you to perform tasks with significant time and cost savings and take advantage of the methodological and solution maturity provided by other lines of data science. An interesting research topic is the automatic generation of the filling processes of the instances where the study of update strategies and their variables will be important.

References

1. Piattini M.,Velthuis,Caballero Muñoz-Reja I., Gomez Carretero A., Cejudo F., Garcia J., Rivas Garcia B.: Calidad de Datos. (2018).
2. Cardador Cabello A.: Data warehouse business intelligence. (2019).
3. Ghavami P.: Big Data Analytics Methods: Modern Analytics Techniques for the 21st Century: The Data Scientist's Manual to Data Mining, Deep Learning & Natural Language Processing. (2016).
4. Giráldez R., Riquelme J., Aguilar-Ruiz J.: Tendencias de la Minería de Datos en España. Journal of Educational Data Mining. Vol 7, No 1. (2015).
5. Corr L.: Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to Star Schema. (2011)
6. Kimball R., Ross M., The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Wisley Publishing, Inc., 2010
7. Inmon W., Strauss D. y G. Neushloss: The Architecture for the Next Generation of Data Warehousing. (2008).
8. Ghavami P. :Big Data Analytics Methods: Modern Analytics Techniques for the 21st Century: The Data Scientist's Manual to Data Mining, Deep Learning & Natural Language Processing. (2016)
9. Golfarelli M. y Rizzi S.: Data Warehouse Design: Modern Principles and Methodologies.(2009)
10. Do N.: Developments in Data Extraction, Management, and Analysis. (2012)
11. SAS Institute, Inc. SAS Rapid Warehousing Methodology, White Paper.(2002).
12. Barroso V.: *Explotación e Integración de Bases de Datos Heterogéneas* para la.
13. Universidad de Valladolid. (2015).
14. Botello C. A.: *Explotación de bases de datos heterogéneas mediante su integración parcial*. (2004).
15. Kotu V., Deshpande B.: *Ciencia de los datos*. (2019).
16. Batini C., Scannapieco M.: Data Quality: Concepts, Methodologies and Techniques. (2006).
17. Naumann F: Quality-Driven Query Answering for Integrated Information Systems. (2002).
18. Piattini M., Félix O, Gracia, Caballero I.: *Calidad de Sistemas Informáticos*. (2006).
19. Indurkhya N., Damerau F.*: Handbook of Natural Language Processing*. (2010).
20. Allen J.: Natural Language Understanding. 2ed. (2012).
21. El Moukhi, N., El Azami, I., Mouloudi, A., El Mounadi, A., *Requirements-driven modeling for decision-making systems.* In International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS). (2018).
22. Winter, R., Strauch, B., A Method for Demand driven Information Requirements Analysis in Data Warehousing Projects. HICSS-36:231-239. (2002).
23. Phipps, C., Davis, KC: *Automation of the design and evaluation of the conceptual scheme of the data warehouse.* In: Proceedings of the International Workshop on Design and Management of Data Warehouses. Vol. 58, págs. 23–32. (2002).
24. Song, Y., Khare, R., Dai, B., *A method for demand-driven information requirements analysis in data warehousing projects*. In: Proceedings of the 10th ACM International Workshop on Data Storage and OLAP, págs. 9–16. (2007).
25. Nazri, MNM, Noah, SAM, Hamid, Z., *Conceptual design of automatic data storage. International Symposium on Information Technology of* 2008, Kuala Lumpur, Malasia, págs. 1 a 7. (2008).