Automating Data Integration in Adaptive and Data-intensive Information Systems

João Galvão[1[0000-0003-4263-8726]], Ana Leon[2[0000-0003-3516-8893]],
Carlos Costa[1[0000-0003-0011-6030]], Maribel Yasmina Santos[1[0000-0002-3249-6229]],
Óscar Pastor López[2[0000-0002-1320-8471]]
ALGORITMI Research Centre, University of Minho, Guimarães, Portugal
{joao.galvao,carlos.costa,maribel}@dsi.uminho.pt
[2] Research Center on Software Production Methods (PROS), Universitat Politècnica de València, Valencia, Spain
{aleon,opastor}@pros.upv.es

Abstract. Data acquisition is no longer a problem for organizations, as many efforts have been performed in automating data collection and storage, providing access to a wide amount of heterogeneous data sources that can be used to support the decision-making process. Nevertheless, those efforts were not extended to the context of data integration, as many data transformation and integration tasks such as entity and attribute matching remain highly manual. This is not suitable for complex and dynamic contexts where Information Systems must be adaptative enough to mitigate the difficulties derived from the frequent addition and removal of sources. This work proposes a method for the automatic inference of the appropriate data mapping of heterogeneous sources, supporting the data integration process by providing a semantic overview of the data sources, with quantitative measures of the confidence level. The proposed method includes both technical and domain knowledge and has been evaluated through the implementation of a prototype and its application in a particularly dynamic and complex domain where data integration remains an open problem, i.e., genomics.

Keywords: Big Data, Data Integration, Schema Matching, Similarity Measures.

1 Introduction

Data is becoming more and more relevant as decision support in organizations can benefit from retrieving value from the vast amounts of data that are nowadays collected from a wide range of data sources. Many efforts in automating data collection and storage provided the context to have access to a wide range of data sources that can be used to support the decision needs of organizations. Nevertheless, those efforts were not extended to the context of data integration, as many data transformation and integration tasks remain highly manual. This problem is even more critical when we move to a Big Data context in which the volume, variety and velocity of data impose several challenges to this data integration needs.

The problem that motivates this work occurs when new data sources become available and there is the need to integrate this new data into existing data systems. This

problem is amplified when those data sources need to be added and removed, in highly dynamical domains due to the variability of the available repositories. In those cases, data engineers need to inspect those data sources to identify the available attributes, their possible values and distribution, assess their quality and then think about their integration in the destination systems. Parts of this work are supported by several tools that can automate data transformation and integration tasks, but data pipelines definition and data modeling (and remodeling) are tasks that are deeply connected to the business knowledge of the data engineer, being often manually performed and highly time consuming.

To overcome these limitations, this work proposes a method for the automatic inference of the appropriate data mapping of new data sources, supporting the data integration process between these data sources and the corresponding destination systems. This method supports data engineers in this process, providing a semantic overview of the data sources, with various quantitative measures of the confidence level of the relationships between the data, taking as input the data sources and the characteristics (e.g., possible values and data distribution) of their several attributes.

This paper is organized as follows. Section 2 presents the related work. Section 3 describes the method for data integration. Section 4 addresses the demonstration case in the genomics field, while Section 5 presents the obtained results, and concludes with some proposals of future work.

2 Related Work

In data-intensive systems, general data processing approaches include data extraction, transformation and management with the purpose of making available information to the user [1]. In a Data Warehouse, a data system built to consolidate and make available relevant information for decision support, data from different sources goes through a complex process of data integration that ensures a unified and coherent view of the organizational or application domain data.

Nowadays, with the advent of Big Data, new challenges emerge in Data Warehousing or other data storage systems, as the volume, variety, or velocity of the data require performant solutions able to deal with these data characteristics [2]. In this context, data storage systems need to be seen as flexible, scalable and highly performant systems that use Big Data techniques and technologies to support mixed and complex analytical workloads (e.g., streaming analysis, ad hoc querying, data visualization, data mining, simulations) in several emerging contexts [3]. The data understanding and integration tasks are generally manual-based, supported by tools that give some hints about the available data, but that miss an overall and integrated approach about the data and their characteristics and how a semantic integration of the data sources can be made. Matching entities and attributes of an application domain is usually a human-based time consuming task, which is not suitable for Big Data contexts [4] that must consider highly dynamic environments with data needs that can add or remove data sources in a very dynamic way due to the variability of the available repositories.

The work of [5] highlights a road map for researching in Big Data Management, including data integration and data matching issues. In [6], the authors argue that data

matching can be defined as the challenge of proposing a match between elements of two different datasets, with the aim of proposing a unified dataset based on datasets developed and made available in an independent way.

Based in [6, 7], some matching technics are enumerated: (1) Schema matching – based on the comparison between data schemas; (2) Graph matching – based on the comparison of the relations between different elements of the schemas; (3) Usage-based matching – based on the databases' logs that show the users' frequent joins between different datasets; (4) Linguistic matching – based on the name or the description of the elements using similarity on strings; (5) Auxiliary matching – using dictionaries and incompatibility lists; (6) Instance-based matching – using the elements statistics and metadata similarity analysis; and (7) Constraint-based matching – based on data types, values' distributions, foreign keys, unique values, among other constrains.

These different techniques have different degrees of compliance with the data integration tasks, namely in the accuracy and utility of the results. Namely, techniques that address the content of the data tend to have better results, such as the (1) Cosine, (2) Jaccard, (3) Jaro-Winkler, and (4) Levenshtein [8–12] measures**.**

Due to the characteristics of those measures, mainly analyzing data content in a highly detailed way and comparing all pairs of possible combinations, they may not be the most suited ones for Big Data contexts, if the objective is to compare all pairs of strings and the characters inside them. In some techniques, instead of comparing all sequences of characters, if the comparison is made between strings, the computation of the similarity can be enhanced. This is the case of the Jaccard Index. Based on some of these well-known measures, Zhu et al. [13] present the Set Containment similarity measure, analyzing if a new dataset is contained in an existing one, a measure that is helpful in Big Data contexts. Although the utility of this measure, it cannot be used in an isolated way, as the complexity of the Big Data domain, with a high diversity of new data sources, requires the use of an integrated and automated approach.

In the last years, the scientific community has been addressing the data integration challenge, proposing some works with interesting results. The work of [14] states that the main issues in this area are related with the use of different names and structures for describing the same information. So, the authors propose the use of semantic dictionaries to overcome this problem and verified that, although the obtained time reduction, the dictionary is specific and for a broader use it requires the definition of new ontologies by domain experts. Moreover, a set of data integration frameworks based on ontologies, for unified multidimensional models, are presented in [15, 16].

A semi-automatic approach for Data Warehouses integration with similarity measures applied at a syntactic, semantic and structural level is proposed in [17]. The results of this work show that the proposed similarity methods are not adequate for vast amounts of data, as the ones existing in Big Data contexts.

The KAYAK framework [18] aims to help data scientists in the definition and optimization of the data preparation processes for Data Lakes. This framework uses the available metadata to calculate similarity measures like joinability and affinity. The usage of metadata for the integration of data in Big Data contexts is also explored in [19]. Although their capability to work in Big Data contexts, these works do not address the problem of adapting existing data models and repositories, in a dynamic way, in

order to adjust these models and repositories to new data requirements and decision-support needs.

Having addressed the related work and the main limitations of the existing approaches, this paper presents a method for data integration in Big Data contexts that makes use of several similarity measures to identify the inter datasets similarity in an automated way. This method is supported by a prototype that computes the measures and maps the several attributes, in an approach that supports the integration of the several external data sources and in the integration of those data sources with existing data systems.

3 Data Integration in Adaptive and Data-intensive Information Systems

The method for data integration here proposed assumes that in a Big Data context there is the continuous need for searching new data sources relevant to the organization, application domain or problem at hands. Those data sources are then extracted, analyzed (applying the similarity measures), transformed, validated and loaded to a destination data system that has a unified and coherent view of the data. As the structured practice of this method is here instantiated and evaluated through the implementation of a prototype and its application to a demonstration case, respectively, this section embeds the structured practices proposed for the method in a prototype, presenting its system architecture, with its several components, interfaces and data flows. For the several components, the main technologies used in their implementation are also pointed.

3.1 System Architecture

The Domain Knowledge side, that can be considered an external component of the Data Integration system (**Fig. 1**), aims to include semantic knowledge in the method using knowledge obtained from a Domain Expert and/or Conceptual Schema. The Conceptual Schema is a domain dependent data model, which may already exist, including the relevant entities of the application domain, their attributes and how the entities are related to each other. It is important to highlight that this conceptual schema provides the context and the ontological background required to perform an accurate validation. In case this schema does not exist, or is not updated, the transformed data can be used to infer such schema or to update it.

Metadata Conceptual Schemas are general knowledge about how the data should be organized, depending on the Destination System that will store the data. For example, if the data will be stored in a relational database, the Metadata Conceptual Model defines what elements a relational database should have, such as Tables, Attributes, Primary and Foreign Keys, among others. This information is needed to automatically devise the most suited data structure to store the data. In a Big Data context, where file systems, NoSQL, NewSQL, or other data systems can be used, this component ensures that a broad range of destination systems can be used.

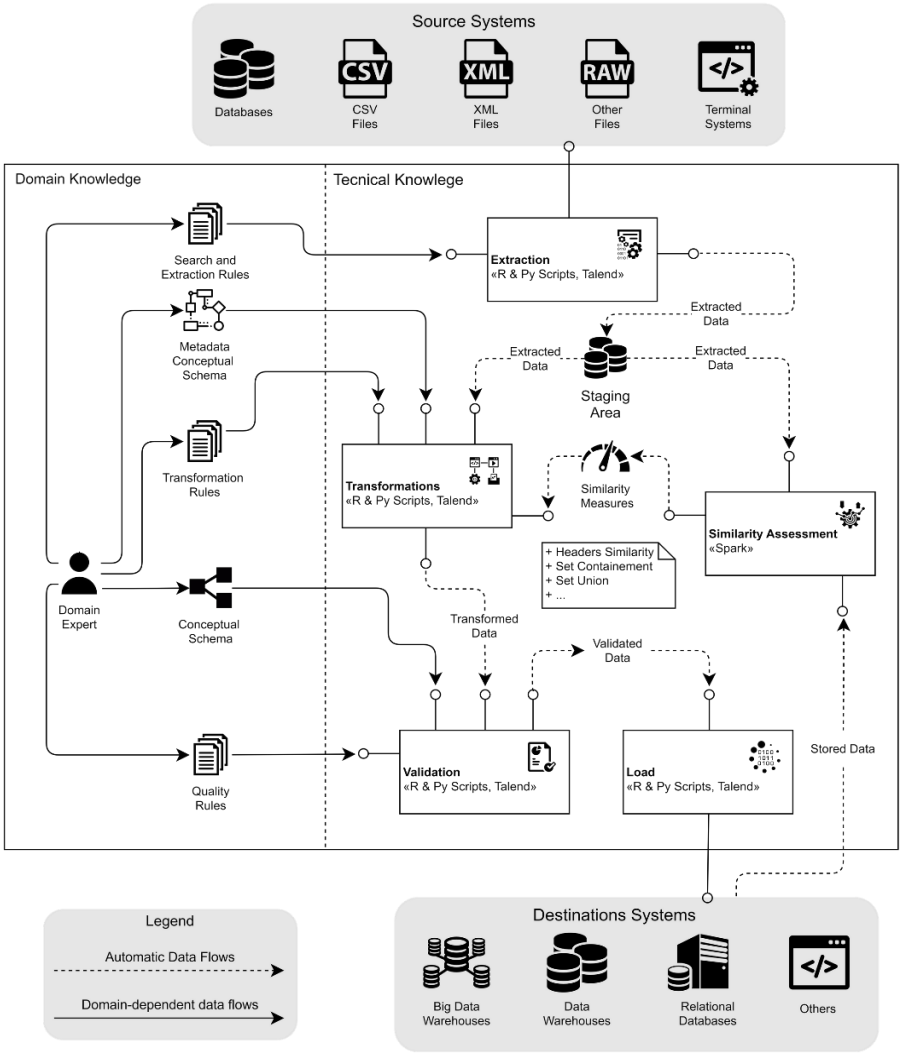

Fig. 1. System Architecture

For extracting the relevant data, a set of general Search and Extraction Rules needs to be defined by a domain expert, including some guidelines about the relevant data to be found and the extraction rules to be followed, in order to ensure that the data has some value, as the number of potential (existing) data sources is quite high. As an example, and using now the application domain in which the demonstration case will be based, the genomic domain and the supporting information system, the search rules can include guidelines that express the need to find out datasets about the DNA variants associated to the Alzheimer disease. An extraction rule can include constraints such as the format of the information to be retrieved (VCF files, XML, etc.) or the source

systems to be considered, ensuring that only data from trusted data sources are used, such as well-known repositories in the genomic domain.

Beside the transformations made to map the attributes, other transformations are usually needed in any data context, in order to clean and to put the data in the appropriate data format. For that, the Transformation Rules includes the set of data-dependent transformation rules needed, such as adding or removing prefixes, normalizing Booleans, among others. Also, dependent on the Domain Knowledge are the Quality Rules, making explicit quality measures that the data must comply in order to be considered in the data integration method. These quality rules can express general data quality measures like handling missing values, noise, outliers, among others.

Looking for relevant data, the proposed method looks for external data sources, collecting data from those systems (Source Systems), using different technological approaches that can include several database drivers, Web Services that can send the output as CSV, XML or other raw format, and Terminal Systems like production machines able to send data in real-time.

Regarding the Technical knowledge, the Extraction component uses the information made available in the Search and Extraction Rules, extracting the identified data into the Staging Area, which in the case of the proposed prototype uses the Hadoop Distributed File System. This Extraction component can be implemented using different scripting languages or specific applications like Talend Open Studio for Big Data, the case in this prototype.

The Metadata Conceptual Schema, the Extracted Data, the Transformation Rules, and the Similarity Measures are inputs of the Transformation component, that will do the transformations needed to map the Extracted Data to the Destination Systems, integrating those external data sources between them and, additionally, with the data already available in the Destination Systems. The Transformation component can be implemented using a scripting language or any available Extraction, Transformation and Loading (ETL) tool, such as Talend.

The Similarity Assessment uses a set of similarity measures to compare two datasets and returns a graph of possible data matches and the corresponding Similarity Measures. As one of the main goals of this work is the use of the similarity measures to automatically identify the data matches, the Similarity Assessment component is further detailed in the next sub-section.

The Validation component uses the Transformed Data, the Quality Rules and the domain Conceptual Schema (when available) to run validation tasks that ensure that the data to be stored in the Destination Systems comply with the domain needs. The Quality Rules can have a lower or higher degree of complexity, requiring different approaches when handling them. In this method, it is proposed that the rules are expressed in a scripting language, in order to be automatically used by the Validation component.

After all the Validation procedures are completed, the Load component is used to send the data to the Destination Systems. This component can be implemented using a scripting language or an ETL tool.

Finally, the Destination Systems component includes the data repositories used as storage components. Several storage technologies can be used, depending on the data characteristics, volume of the data or even organizational technological constraints.

3.2 Similarity Assessment

Given the nature of this work, propose a method for data integration in Big Data contexts, there is the need to identify and implement the similarity measures that allow an automatic data integration process. This component depicted in **Fig. 1**, central in the proposed method, is now detailed in **Fig. 2**. The Similarity Assessment component will produce a set of measures that indicate if two attributes are related and, therefore, they can be joined, mapped or merged.

The Similarity Assessment uses new data (Extracted Data) and already existing data (Stored Data), when available in the Destination Systems, comparing the headers (name

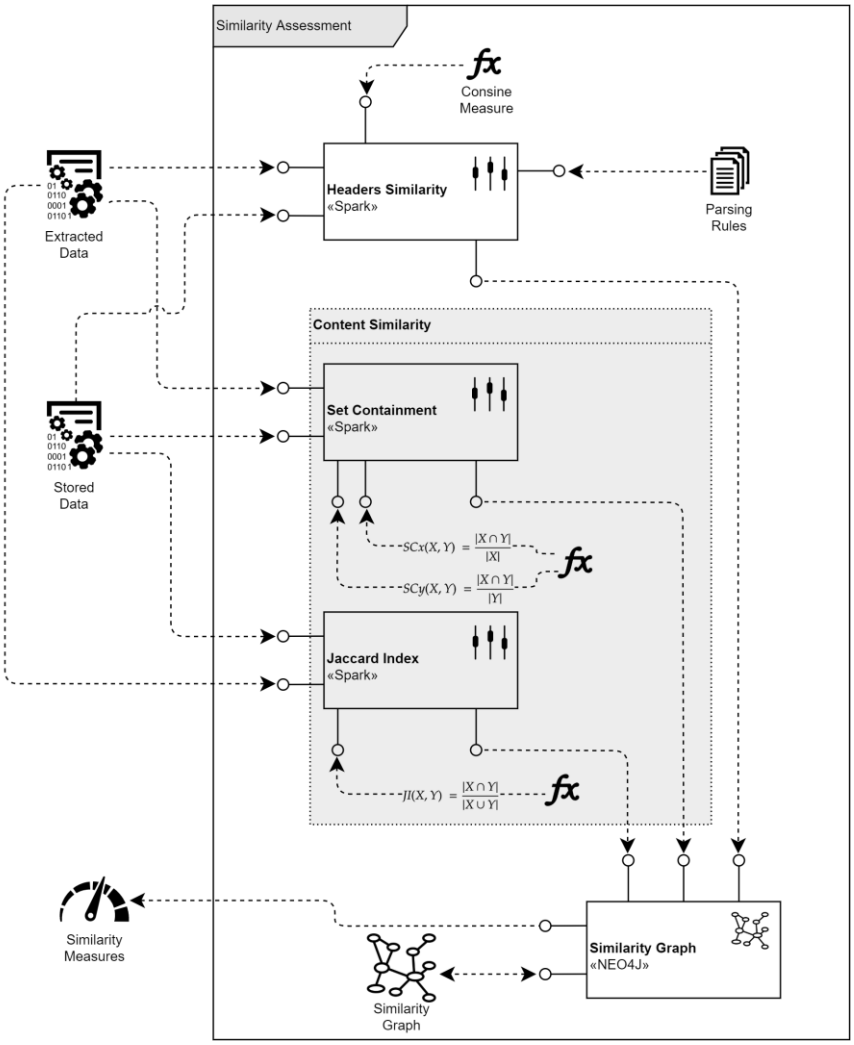

Fig. 2. Similarity Assessment component

of the attributes) and the content (different values for those attributes) for the different attributes and computing a set of similarity measures, such as Headers Similarity, Set Containment and Jaccard Index. These are then combined in order to evaluate the similarity of the data sets and how their integration could be achieved. This semantic knowledge is stored in a Similarity Graph. The graph includes edges to express relationships between the attributes and the corresponding computed measures.

In order to support the adequate computation of Headers Similarity, the headers need to be parsed. The rules here applied are defined in the Parsing Rules, and can include tasks such as removing all spaces, removing all special characters, renaming to lowercase, among others. The Headers Similarity computation is done using the Cosine Similarity Measure, which is suitable for contexts where the comparison of a few strings is needed [8].

The analysis of the Content Similarity has two elements that compute three measures. The Set Containment is useful to identify the different data flows that can be followed when looking into the integration of different data sets. For this, the Set Containment measure proposed by Zhu et al. [13] is applied bidirectionally, measuring the set containment from X to Y (Equation 1) and from Y to X (Equation 2), as when several external data sources are available, different paths of integration can be followed.

$$SCx(X,Y) = \frac{X \cap Y}{X} \tag{1}$$

$$SCy(X,Y) = \frac{X \cap Y}{Y} \tag{2}$$

The use of the Jaccard Index (Equation 3) is to verify the shared data values between the different attributes of the analyzed data sets.

$$JI(X,Y) = \frac{X \cap Y}{X \cup Y} \tag{3}$$

The implementation of this Similarity Assessment component was done in Spark, using a computational distributed Hadoop platform, and makes available a CSV file with the pairs of all compared attributes and the corresponding computed measures, as well as a Cypher Script that can be used for implementing a graph database with this semantic knowledge in Neo4J.

4 Demonstration

The demonstration case used to validate the Similarity Assessment component is based on the genomic domain, which is characterized by its complexity, what means that the information required to succeed in a complex research is not usually available in only one data source. Frequently, several heterogeneous public databases with different sizes, formats and structures must be queried in order to join all the puzzle pieces together. New databases appear at a high pace thanks to the development of powerful sequencing technologies but, at the same time, a significant number of sources can become obsolete quickly because over time they lose the technological maintenance required or because the information stored is no longer updated affecting its potential

usefulness for researchers. This situation makes the genomic domain very dynamic in terms of analysis and integration of new sources. In addition, the lack of a clear ontological basis to define the key concepts of the field means that the same concept can be represented in different and sometimes ambiguous ways.

This analysis and integration processes are mainly manual, which require a deep study of the structure used to represent the data on each source, as well as the mapping of the common concepts among different sources (name, format, etc.). Consequently, the process is tedious, repetitive and time consuming as well as prone to human errors due to the lack of explicit and systematic methods to support it.

With the aim of evaluating the usefulness of the proposed method to help mitigating the above-mentioned problems, a prototype for the Similarity Assessment was developed in Java to run on Apache Spark. This technology was chosen to be capable to deal with data that has Big Data characteristics. The demonstration will be done by running the prototype for the datasets presented in the next subsection.

4.1 Datasets Description

For this work four datasets coming from different sources have been used. The information of each dataset represents DNA variants associated to the risk of suffering Alzheimer's Disease. Each variant is mainly characterized by its structural information (its location in the genome), the change that occurs and the statistical evidence regarding the studies performed over specific populations. Each database provides a different number of attributes that must be integrated in order to provide a global information system that can support the genetic diagnosis of a patient. Next, a brief description of each dataset is provided.

The Ensembl Dataset was extracted from the Ensembl database, developed as a joint project between EMBL-EBI and the Wellcome Trust Sanger Institute. It is composed of 36 attributes that represent data about the location of each variant in the genome, its pathogenicity and statistical information about the different studies found in the literature. Furthermore, the dataset that has called AlzForum has been extracted from the AlzForum database, a repository specialized in the different types of Alzheimer's Disease and its genetic causes. The dataset includes 14 attributes with data about the location of each variant, its pathogenicity and identifiers to external bibliography resources. This dataset does not include statistical information about the mentioned studies. GWAS dataset was extracted from the GWAS Catalog database, a repository that contains data about different genotype-phenotype association studies. It is composed of 38 attributes focused on the characteristics of each study regarding statistical significance, population and type of study performed. The dataset does not contain information about the pathogenicity of the variants. A least one, the DisGeNet dataset, has been extracted from the DisGeNet database, a repository that contains information about DNA variants associated to different human diseases. It is composed of 16 attributes about the location of the variants in the genome, specific statistics to measure the research interest (relevance) of each variant and identifiers to external bibliography resources.

4.2 Similarity Graph

For the demonstration, four similarity measures were calculated: Header Similarity (HS), Set Containment in both directions (SCx, SCy) and Jaccard Index (JI). The HS measure uses the Cosine Measure comparing the characters inside each header.

The computation of all similarity measures was done for all pairs of attributes between all datasets pairs, resulting in a graph with 1647 relationships between attributes that score at least one measure with a value higher than zero. Due to the complexity of showing the measures in a graph with this amount of relationships, a sample of the Similarity Graph is presented in Fig. 3, including 3 nodes and 3 relationships. The label of each node is the name of the attribute, while the dataset is identified by the pattern of the node's line. The relationships are named as *has_similarity*, meaning that at least one of the similarity measures has a computed value higher than zero, and the measures are represented as a relationship's property.

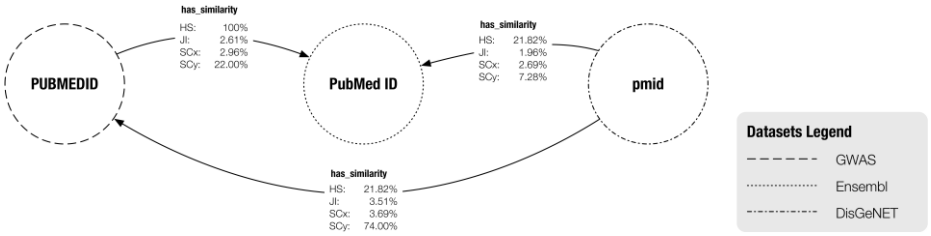

Fig. 3. Sample of the Similarity Graph

5 Discussion and Conclusions

In order to evaluate the results obtained after the analysis of the four datasets, a comparison between the Similarity Graph and a manual mapping performed by a domain expert was done. With the aim of increasing the legibility when presenting the results, a filter was applied removing an extensive set of low similarity values: `HS>20 OR (JI>0 AND SCx>0 AND SCy>0)`.

5.1 Analysis and Discussion of Results

The intersection between the manually mapped graph and the Similarity Graph is presented in **Fig. 4**. The result is a set of subgraphs containing 31 nodes and 36 relationships, extending the one presented in Fig. 3, by adding relationships with full lines that represent the relationships that were manually and automatically detected, and the dashed lines that represent the relationships that were only manually detected. Using were identified, even in a complex context were the names of the attributes are very different. Take, for example, the name of the variant: *Variant name*, *snpid*, *SNPS* and *ID*. By comparing the manually detected relationships (36) with those automatically detected (28), a match rate of 77.7% was achieved, representing a noteworthy result for a first approach. Other particularly difficult attributes, such as those associated with the

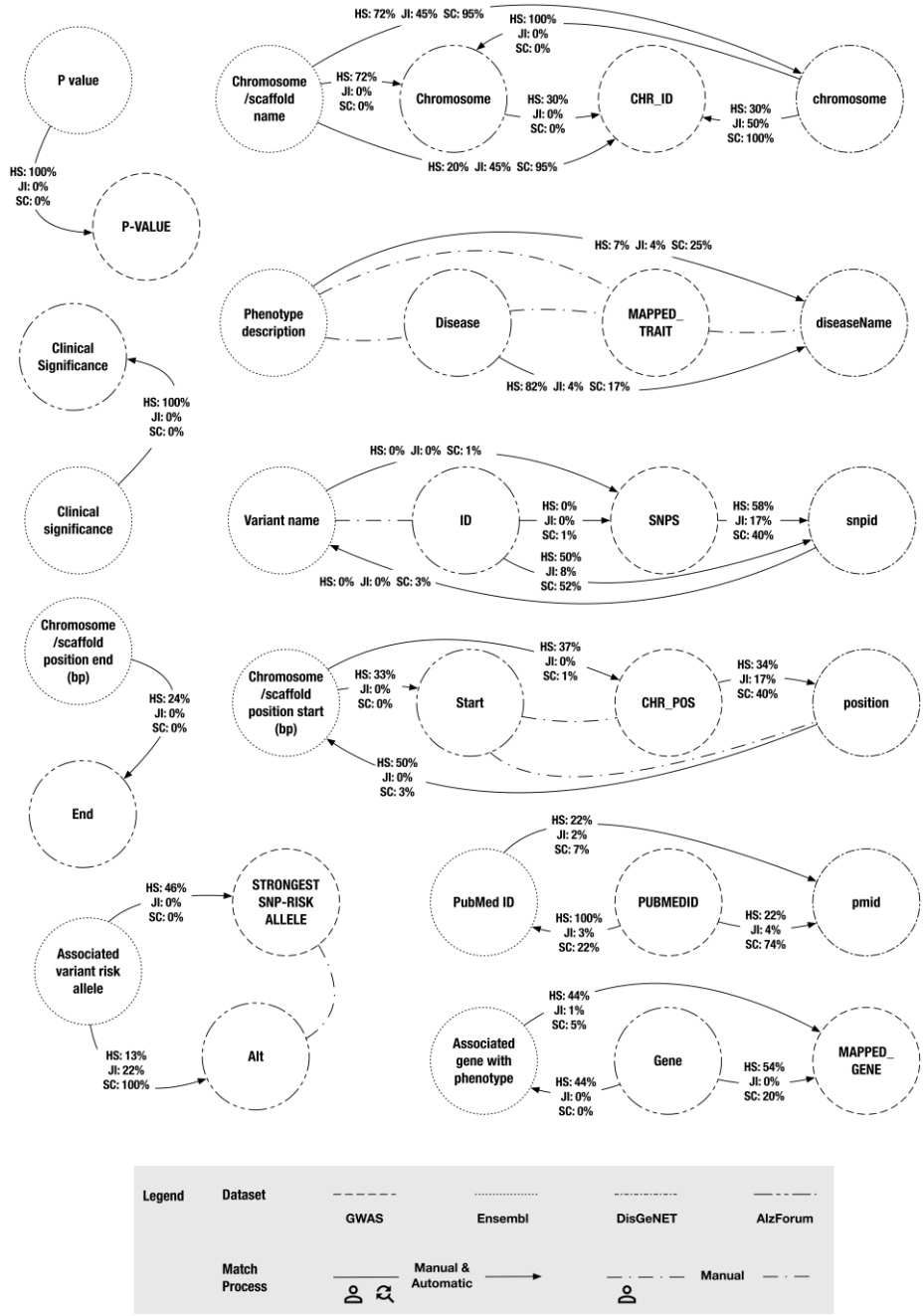

Fig. 4. Graph with manual and automatic mapping relationships

phenotype, were partially identified. The first analysis of the graph points that some of the missed relationships could be inferred by transitivity. For example, if *Variant name* is similar to *SNPS* and *SNPS* is similar to *ID*, would not be *Variant name* similar to *ID*? Also, after sharing the results with the domain expert, the pairs (*Reference*, *Minor allele (ALL)*) and (*Alt*, *Minor allele (ALL)*) that were not manually identified, were considered required to do the integration of the different datasets

To better understand the identified false positives, further analyses were made in other to identify the False Positive Challenges (FPC). In this context, CM (Content Measures) stands for all content similarity measures considered in this work (JI, SCx and SCy). All pairs with HS=100 and CM = 0 automatically classified were also manually classified, but there are cases with HS values around 70%-80% that alone cannot be used to say that two attributes are similar. For example, the pair of attributes *Disease* and *diseaseType* with HS= 78% and CM=0 do not match, as one is the name of the disease and the other its type (FPC1). Also, comparisons with attributes of low cardinality tend to increase the number of false positives (FPC2). Higher thresholds for the metrics reduce the number of the of False Positives, but also increase the number of unmatched pairs (FPC3).

Furthermore, the analysis of results showed the Unmatched Pair Challenges (UPC) identified in this domain: i) some attributes contain more information than needed. For instance, the attribute *STRONGEST SNP-RISK ALLELE* has values like `rs144573434-T`, including both an alternative allele and a variant identifier. This situation was noticed by the human domain expert when performing the manual mapping who identified that the attribute is referring to the allele (UPC1); ii) the range of values for the attributes are different. For example, the variant identifiers of two datasets associated to different types of diseases or chromosomes (UPC2); iii) the free writing of attributes, with values that do not match (UPC3); iv) the lack of patterns or standards to represent the data that needs prefixes or any other coding (UPC4).

A set of suggestions for future work will be made next in other to overcome the identified challenges: (FPC1) The addition of a new dimension of analysis for the HS, like the use of a semantic analysis using word dictionaries, as those could improve the reliability of the HS; (FPC2) Include in the Content Similarity some basic statistics such as the number of district values, frequency distribution, among others; (FPC3) Identify relevant thresholds for filtering the obtained results; (UPC1) Apply rules for data cleaning; (UPC2) Analyze the syntax of the possible values for the attributes with Frequent Pattern Mining techniques (for instance, if two attributes have different values like `rs123456` and `rs654321`, they do not automatically match, but they share the same prefix/syntax, `rs<number>`, meaning that they are likely similar); (UPC3) Identify additional string content measures suitable for Big Data contexts, able to analyze if there is a match between the intersection of the characters of two strings, detecting similarities between two different strings with the same meaning (like `late-onset Alzheimer disease` and `Alzheimer disease`); (UPC4) Identify a set of rules for data cleaning and transformation using frequent pattern detection.

Those suggestions for possible improvements could be applied to the proposed method without the need to adapt it or extend it, just by adding those new metrics or rules.

5.2 Conclusions

This paper highlighted the challenge of the manual effort needed for data integration tasks, namely in highly dynamical domains due to the variability of the available repositories. This work proposes a method capable of automating data integration tasks in adaptive and data-intensive information systems. The instantiation of the proposed method was implemented in Apache Spark to support Big Data contexts.

The evaluation was made by comparing the manual and automatic mapping of the attributes present in four datasets from a particularly complex and dynamic domain: genomics. Besides being satisfactory and showing a high matching rate, the results highlighted some challenges that need to be further addressed, like the occurrence of false positives and the threshold that can be considered to automatically have a certain degree of confidence on the obtained results.

Some improvements were identified for future work, aiming to increase the match rate, highlighting the concept of transitivity that infer the missed relationships, the use of word dictionaries to have a semantic dimension of analysis and the syntax analysis. With these future improvements this method will be applied in other contexts, such as manufacturing, to better understand how the method handles data from other domains.

Acknowledgements. This work has been supported by FCT – *Fundação para a Ciência e Tecnologia* within the Project Scope: UID/CEC/00319/2019, the Doctoral scholarship PD/BDE/135100/2017 and European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 039479; Funding Reference: POCI-01-0247-FEDER-039479]. We also thank both the Spanish State Research Agency and the Generalitat Valenciana under the projects DataME TIN2016-80811-P, ACIF/2018/171, and PROMETEO/2018/176. Icons made by Freepik, from www.flaticon.com

References

1. Krishnan, K.: Data warehousing in the age of big data. Newnes (2013).
2. Vaisman, A., Zimányi, E.: Data Warehouses: Next Challenges. In: Aufaure, M.-A. and Zimányi, E. (eds.) Business Intelligence: First European Summer School, eBISS 2011, Paris, France, July 3-8, 2011, Tutorial Lectures. pp. 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27358-2_1.
3. Costa, C., Santos, M.Y.: Evaluating Several Design Patterns and Trends in Big Data Warehousing Systems. In: Krogstie, J. and Reijers, H.A. (eds.) Advanced Information Systems Engineering. pp. 459–473. Springer International Publishing (2018).
4. Bellahsene, Z., Bonifati, A., Duchateau, F., Velegrakis, Y.: On Evaluating Schema Matching and Mapping. In: Bellahsene, Z., Bonifati, A., and Rahm, E. (eds.) Schema Matching and Mapping. pp. 253–291. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-16518-4_9.
5. Santos, M.Y., Costa, C., Galvão, J., Andrade, C., Pastor, O., Marcén, A.C.: Enhancing Big Data Warehousing for Efficient, Integrated and Advanced Analytics - Visionary Paper. In: Cappiello, C. and Ruiz, M. (eds.) Information Systems Engineering in Responsible

Information Systems - CAiSE Forum 2019, Rome, Italy, June 3-7, 2019, Proceedings. pp. 215–226. Springer (2019). https://doi.org/10.1007/978-3-030-21297-1_19.
6. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic Schema Matching, Ten Years Later. PVLDB. 4, 695–701 (2011).
7. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic Schema Matching with Cupid. In: Proceedings of the 27th International Conference on Very Large Data Bases. pp. 49–58. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001).
8. Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y.: A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. PLOS ONE. 10, e0144059 (2015). https://doi.org/10.1371/journal.pone.0144059.
9. Xiao, C., Wang, W., Lin, X., Shang, H.: Top-k Set Similarity Joins. In: Proceedings of the 2009 IEEE International Conference on Data Engineering. pp. 916–927. IEEE Computer Society, Washington, DC, USA (2009). https://doi.org/10.1109/ICDE.2009.111.
10. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady. 10, 707 (1966).
11. Jaccard, P.: Etude comparative de la distribution florale dans une portion des Alpes et du Jura. Impr. Corbaz, Lausanne (1901).
12. Winkler, W.E.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage [microform] / William E. Winkler. Distributed by ERIC Clearinghouse, [Washington, D.C.] (1990).
13. Zhu, E., Nargesian, F., Pu, K.Q., Miller, R.J.: LSH Ensemble: Internet-scale Domain Search. Proc. VLDB Endow. 9, 1185–1196 (2016). https://doi.org/10.14778/2994509.2994534.
14. Banek, M., Vrdoljak, B., Tjoa, A.M.: Using Ontologies for Measuring Semantic Similarity in Data Warehouse Schema Matching Process. In: 2007 9th International Conference on Telecommunications. pp. 227–234 (2007). https://doi.org/10.1109/CONTEL.2007.381876.
15. Deb Nath, R.P., Hose, K., Pedersen, T.B.: Towards a Programmable Semantic Extract-Transform-Load Framework for Semantic Data Warehouses. In: Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP. pp. 15–24. ACM, New York, NY, USA (2015). https://doi.org/10.1145/2811222.2811229.
16. Abdellaoui, S., Nader, F.: Semantic data warehouse at the heart of competitive intelligence systems: Design approach. In: 2015 6th International Conference on Information Systems and Economic Intelligence (SIIE). pp. 141–145 (2015). https://doi.org/10.1109/ISEI.2015.7358736.
17. El Hajjamy, O., Alaoui, L., Bahaj, M.: Semantic Integration of Heterogeneous Classical Data Sources in Ontological Data Warehouse. In: Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications. p. 36:1–36:8. ACM, New York, NY, USA (2018). https://doi.org/10.1145/3230905.3230929.
18. Maccioni, A., Torlone, R.: KAYAK: A Framework for Just-in-Time Data Preparation in a Data Lake. In: Krogstie, J. and Reijers, H.A. (eds.) Advanced Information Systems Engineering. pp. 474–489. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_29.
19. Hai, R., Geisler, S., Quix, C.: Constance: An Intelligent Data Lake System. In: Proceedings of the 2016 International Conference on Management of Data. pp. 2097–2100. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2882903.2899389.