Pattern Recognition Letters 166 (2023) 134–142

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A semi-automatic data integration process of heterogeneous databases

Marcello Barbella*, Genoveffa Tortora

Department of Computer Science, University of Salerno, Fisciano, Italy

ARTICLE INFO

Article history:
Received 13 July 2022
Revised 15 December 2022
Accepted 10 January 2023
Available online 14 January 2023

Edited by Maria De Marsico

Keywords:
Semi-automatic data integration
Support decision systems

ABSTRACT

One of the most difficult issues today, is the integration of data from various sources. Thus, it arises the need of automatic Data Integration (DI) methods. However, in the literature there are fully automatic or semi-automatic DI techniques, but they require the involvement of IT-experts with specific domain skills. In this paper we present a novel DI methodology for which it is not required the involvement of IT-experts; in this methodology syntactically/semantically similar entities present in the sources are merged, by exploiting an information retrieval technique, a clustering method and a trained neural network. Although the suggested process is completely automated, we planned some interactions with the Company Manager, a figure who is not required to have IT-skills, but whose only contribution will be to define limits and tolerance thresholds during the DI process, based on the interests of the company. The validity of the proposed approach showed an integration accuracy between 99% – 100%.

1. Introduction

Many businesses need to make decisions quickly in order to stay competitive in the global market. This can be a good strategy in some cases, but it can also lead to poor decision-making if the wrong criteria are used. So, it is important to consider all the available information that, in the era of dematerialization, are represented in digital form. Companies have the possibility of accessing to huge amount of data and documents (from databases, social networks, geospatial data, and other sources), from which they can extract and analyze the most important ones for their objectives.

In the digital age, several tasks, including Information Retrieval (IR), document classification, and question-answering, can be supported by document understanding. Additionally, the automatic collection and organization of data from a sizeable collection of documents can be used to improve knowledge management. In this context, Artificial Intelligence (AI) provides effective methods for better interacting with data. Then, it is essential to have the ability of integrating data from different sources [1] so providing the end user with a unified representation of these data. Due to conflicts involving duplicate data, homonyms, synonyms, and various data writing styles, this problem becomes increasingly difficult as more data sources are integrated. Several methods can be used to perform data integration, including Extract, Transform, and Load (ETL), Enterprise Information Integration (EII) [2], Data Federation (DF) and the Data Virtualization (DV) [3].

Data management is essential to the development of a company and its operations. In particular, it is extremely important to explore methods that can guarantee the good quality of data, typically deriving from different sources. The lower the data quality, the greater the inherent difficulties of integration. Besides having different schemes and levels of quality, these sources are distributed, autonomous and heterogeneous. Therefore, the primary goal of a data integration system is that of providing accurate, complete, timely and consistent data output [4].

The analysis of the state of the art showed that the existing research proposals try to make the DI procedures completely or at least partially automatic thanks to a full or partial involvement of an IT-expert, because the automatic results that are generally produced, in most cases, need to be refined.

In this paper, we propose a semi-automatic process that allows one to implement the DI process of different sources in a single reconciled data source schema, without requiring the presence of experts, thanks to the use of Machine Learning (ML) and Information Retrieval techniques. It is only needed the presence of a Company Manager, who is in charge of defining the acceptable constraints and error thresholds for the adopted techniques. The DI process, selects the tables that can be merged by performing both a syntactic analysis and a semantic analysis on the accurately pre-processed input documents. For systems with consistent input data, the final outcomes result in accuracy in the range 99% – 100%.

* Corresponding author.
E-mail addresses: mbarbella@unisa.it (M. Barbella), tortora@unisa.it (G. Tortora).

https://doi.org/10.1016/j.patrec.2023.01.007

The paper is structured as follows: Section 2 discusses related work, Section 3 describes the proposed DI process. Section 4 provides the experimental results and finally, Section 5 concludes the paper and outlines future works.

2. Related works

In Data Integration processes, one of the most difficult problems when integrating heterogeneous databases is in identifying the corresponding attributes of a source and target database schema, during the schema-matching phase. To solve this issue, the majority of approaches is based only on attribute schema information rather than on the content of the data.

From this perspective, [5] propose a novel technique, providing a two-step schema-matching process based on the idea that two attributes from different data sources are comparable if their data patterns or data ranges are similar. In the first step, some neural networks are trained using the data model to discover which pairs of attributes are candidates for matching. Then, the output of the first step becomes the input for the second phase, which provides a rule-based algorithm to eliminate ambiguous mappings and to extract the pairs of attributes that really match. Since no criteria of classification for these rules is specified, this approach requires the presence of an IT-expert who can describe the rules to be applied; so, it results a non-fully automatic process.

The DI system MOMIS (***M****ediator Envir****O****nment for* ***M****ultiple* ***I****nformation* ***S****ources*) [6] is a system for semi-automatic integration of data from many sources. It is targeted at IT-designers because it provides them a variety of functionalities to fine-tune the integration results, including a top-notch graphical interface. Furthermore there are various tools to visualize the outcomes of each stage of the process, and to annotate lexical information useful for the attribute mappings, as well as preview tools of the effects that are reached as a result of the choices made. The ability to gradually improve the results of the integration process, which is typically not available until the completion of the procedure, is possible by this system.

Liu et al. [7] focus on the challenges of semantic integration across various data sources. Specifically, the merits and limitations of a number of automatic and semi-automatic algorithms for performing schema-matching are addressed. Finally, they offer a schema-matching model that incorporates data instances as well as the source schema. Specifically, the first phase compares the schemes to be merged and extracts the features required for the subsequent phases. Then, the *Cluster schema elements* and *Schema-level matching* phases work on the schema sources to be integrated to determine which attributes are semantically connected to each other. Self-Organizing Maps (SOM) and other hierarchical and non-hierarchical clustering techniques are utilized for clustering. After that, the Structure-level matching phase begins, which assesses the similarity of clustered attributes in relation to their context, i.e. data instances. Finally, Mapping-Generation uses a bipartite graph to investigate the final schema-matching problem. This technique, however, has certain disadvantages, because it constructs the schema structure by evaluating only the source schemes without taking into account the data instances.

Ibrahim et al. [8] propose a DI approach that involves three steps and uses both structural and semantic data. In the first phase, users evaluate the different schemes, and choose which ones to integrate. To get the best possible matching between the various schemes, the second phase employs five matchers based on Relation Schemes, Attribute Name, Data Type, Constraints, and Instance Data. The final stage of merging and restructuring completes the process, after intermediate integrated schemes are created, to determine if the semantics of the schemes are preserved or require restructuring.

Mehdi et al. [9] offer an instance-based pattern matching method that fully exploits only the instances of the schemes, using approaches that combine the power of Google as a semantic web and regular expressions as pattern recognition. This method attempts to address a flaw in the schema-matching method. Various instance-based schema-matching systems have employed different strategies for detecting correspondences between schema attributes. The majority of these systems handled instances, including those having numeric values, as strings, making it impossible to find common patterns or do statistical computations between them. As a result, unidentified matches occur, particularly for numeric instances. The findings reveal that this method is capable of detecting 1-1 schema-matching with a high degree of accuracy.

An extensible framework for the automatic effort estimation for mapping and cleaning tasks, is proposed in [10]. It includes a collection of metrics and techniques to calculate the complexity and total cost of integration, taking into account heterogeneities in instances and schemes. They estimate the actual effort necessary to complete the mapping and cleaning tasks, and that about 10% of the integration work cannot be completed entirely by machines and requires the assistance of an IT-designer.

Sahay et al. [11] take into consideration the process of database merging because it has become one of the most important tasks for every organization to preserve product efficiency. They offer a new search approach in which attribute searches are executed in database fields that are linguistically close to one another or reflect the same data. Various approaches for implementing automatic one-to-one schema-matching (centroid, self-organizing maps, etc.) are investigated, and a hybrid methodology is proposed by introducing a global dictionary method for one-to-many correspondence.

Zhang et al. [12] propose SMAT (***S****chema* ***M****atching* ***A****u****T****omated model*), a deep learning model in which the semantic correlation between the attributes of the source and destination schemes is captured based only on the semantic meaning of their descriptions; it uses bidirectional Long short term memory networks (BiLSTM) and the most recent techniques in Natural Language Processing. Even though SMAT has been proved to be not yet sufficient for practical use, this model has the potential to automatically produce the matching between the source and destination schemes, without the need of encoding domain information.

In [13] a new framework for the schema matching challenge in the integration of heterogeneous data is provided. The *knowledge representation learning* and the *schema-matching network* (SMN) are the two phases of which it is composed. External configurable knowledge bases (taxonomy, ontologies, logical constraints and rules, etc.) are introduced in the SMN to be trained, in order to improve the result of complex mappings in the integration. This step enhances the schema matching outcomes obtained, reducing the need for IT experts to refine them.

3. A semi-automatic data integration process

The proposed methodology and the DI process developed in this work are described in detail in Fig. 1. The input data sources are transformed to render the input usable for both the syntactic and the semantic analyses of data. Syntactic and semantic analyses will produce the details on which tables should be integrated, depending on what has been processed for that specific analysis. So, the Company Manager will be in an ideal position to prefer which is the best and plausible hypothesis for the subsequent automatic integration of the identified tables.

To better understand the defined data integration process, we provide a step by step description of all its phases, through a running example, starting from data sources and ending to the reconciled schema.

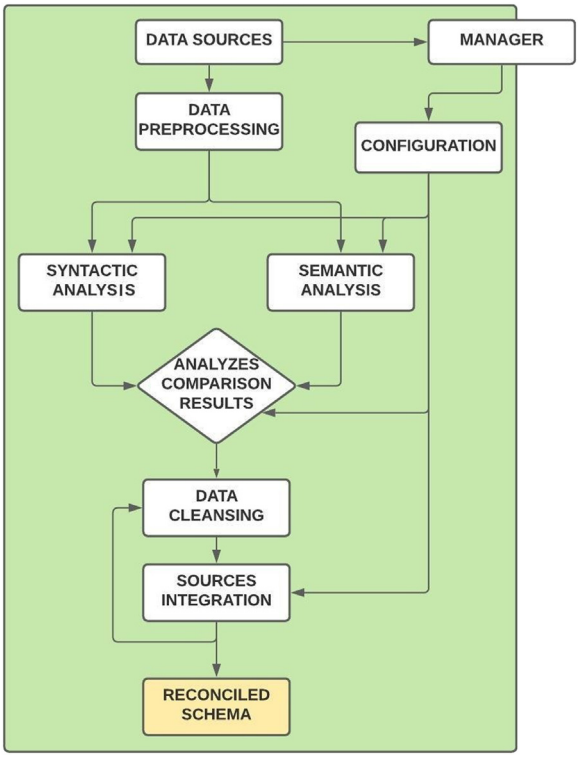

Fig. 1. The proposed Data Integration methodology.

3.1. Data sources phase

For the integration process to be successful, the data sources must model the same real-world domain [14]. Four systems were exploited in this research project, one of which (related to the Cooking System) is currently open accessible to the public. The other three cannot be publicly available, since they contain data subject to privacy protection.

For our analysis, we consider the system *CarShopping* that concerns an international company, which buys collectible models of cars, trains, trucks, buses and ships from manufacturers, and sells them to distributors around the world. Figure 2 shows the two main relational databases of this company: *Human Resources Management* consisting of 3 tables: clients, offices and employees and *Marketing and Sales* made up of 6 tables with customers, orders and products.

3.2. Data pre-processing phase

The first step in performing analysis and classification is to transform textual documents into an algebraic model [15]. An essential step in identifying the shared concepts among the different tables of the data sources is the schema-matching process [14]. Consequently, pre-processing is required to standardize and make the schemes comparable. For this purpose, we adopted an IR technique [16,17], the *Latent Semantic Indexing* (LSI) [18–20]. IR is used for two major problems, which can sometimes cause inconsistencies in searches: *Synonymy* (terms with similar meanings) and *Polysemy* (terms with different context meanings). To avoid the issue of retrieving documents unrelated to the topic of interest, LSI aims to represent documents in terms of concepts rather than terms.

Starting from this concept, we applied a pre-processing phase based on the *normalization* of the input documents of the text mining phase [21], which are represented by relational database tables. Tables with only numeric attributes are excluded from the process, because they do not give any useful semantic information. The normalization minimizes redundant information and reduces the computational times. In particular, we followed the steps below:

- Removal of the columns of tables with many null values (in our case they had to be at least the 70%), because null values make the Term-Document Matrix (TDM) sparse and do not provide enough information to detect similarities between documents, increasing the computation time;
- Elimination of punctuation and white-spaces, since they are useless and do not add information to text;
- Conversion of each word into a lowercase word, for processing convention;
- Removal of stop-words, (e.g. articles, prepositions, conjunctions and so on) that are words with a very high frequency, and which do not provide any useful information to discriminate the topics covered in a document;
- Execution of stemming algorithms [22], that map inflected form of words into their corresponding root form;
- Creation of the TDM, that provides the frequency distribution of each term in each document [23].

The frequencies of the TDM are usually weighted before the matrix is further processed. The overall weight assigned to each entry in the matrix can depend from the frequency of the single term in a document and the frequency of the single term in the whole document collection.

In our case, we have chosen the weight function tf_idf (*term frequency-inverse document frequency*), often used in IR to measure the importance of a term for a single document with respect to all the documents. This function increases linearly with the number of times the term appears in the document, but inversely with the frequency of the term in the collection. The purpose of this behavior is to provide more weight to terms that occur in the document but are not used very frequently. It is defined by:

$$tf\_idf_{i,j} = tf_{i,j} \times idf_i \tag{1}$$

The first factor $tf_{i,j}$ of the function measures the importance of the ith term for the jth document and is expressed by:

$$tf_{i,j} = \frac{n_{i,j}}{|d_j|} \tag{2}$$

where $n_{i,j}$ is the number of occurrences of the term t_i in the document d_j, whilst the denominator is the cardinality (number of terms) of the document d_j, so as to avoid favoring longer documents. Instead, the second factor measures the importance of the same ith term relatively to the entire collection of documents and is expressed as:

$$idf_i = log\frac{|D|}{|\{d : t_i \in d\}|} \tag{3}$$

that is, the logarithm of the ratio between the number of documents in the collection and the number of documents that contain the term t_i. The application of this weight function allows the computing of the TDM.

3.3. Syntactic analysis phase

We conducted a syntactic analysis of the data, based on their syntax and content information such as names, data types, as well as domains and data descriptive information. Essentially, the information of data is the basic information that intuitively reflects its semantics. This type of analysis is performed on the computed TDM, splitting the work in two sub-phases: *a)* calculation of the Levenshtein Distances (LD) matrix [24]; *b)* cluster analysis.

a) *Calculation of the LD matrix.* Starting from the TDM, the LD-matrix is computed. To this aim, each document is treated as

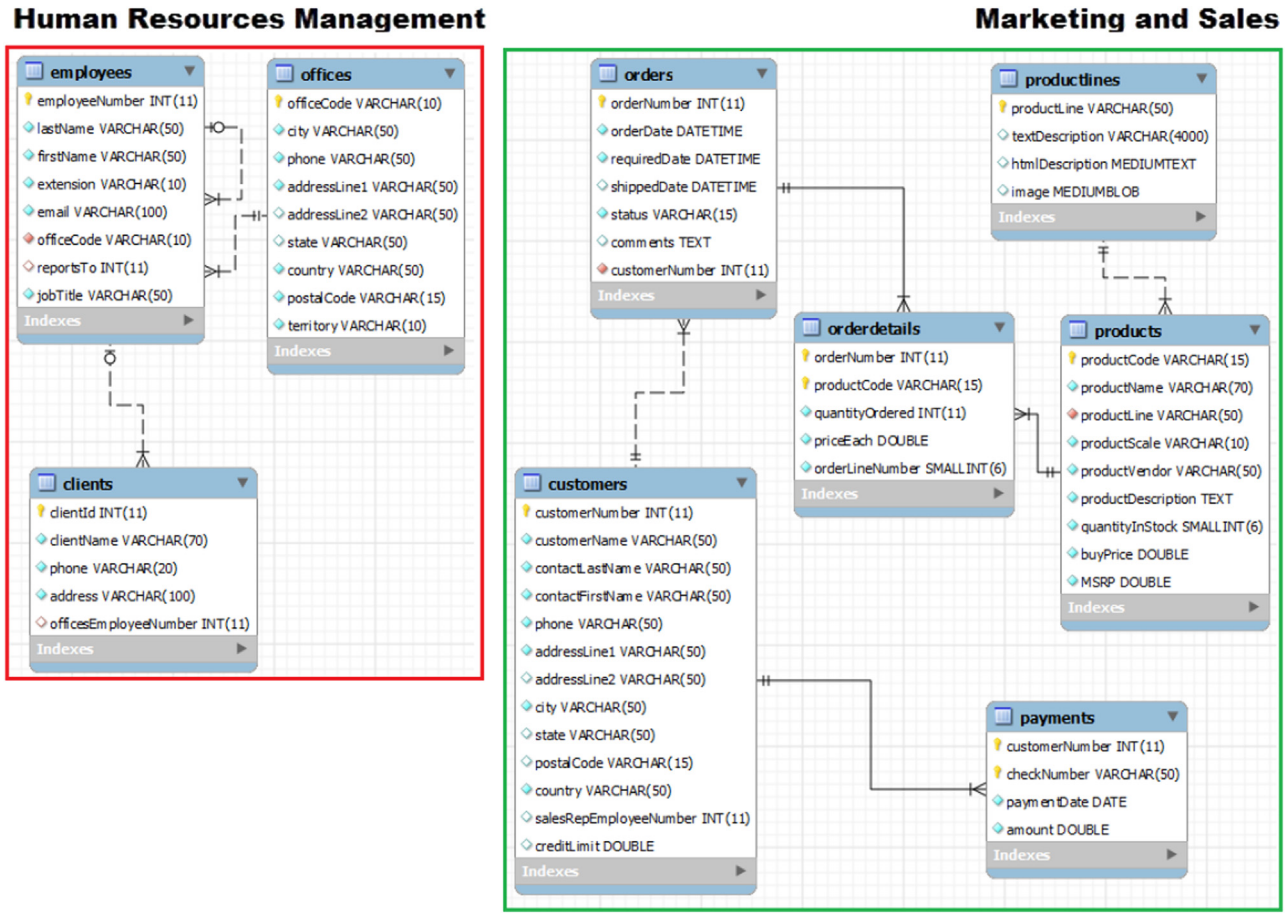

Fig. 2. The Database CarShopping.

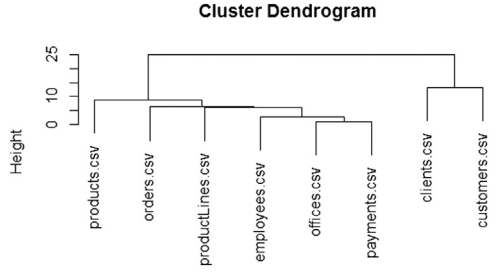

Fig. 3. Syntactic analysis dendrogram for the CarShopping database.

a single column vector. By applying the LD metric to all possible pairs of vectors, the LD-matrix is obtained. The LD matrix will be used to compare the syntactic differences among the various documents, so evaluating how far apart they are from one another.

b) *Cluster analysis.* The LD matrix is the input for the cluster analysis. This allows us to know how each document is distributed in the clusters thanks to the found syntactic similarity. Both hierarchical and non-hierarchical clustering [25], are used in the implemented process. Hierarchical clustering is performed with the only purpose of viewing the dendrogram (see Fig. 3), which gives a first global view of the possible distribution of documents, by similarity. We used an agglomerative hierarchical clustering with the LD metric and the *complete linkage* criterion [26].

For non-hierarchical clustering it was used the k-means algorithm [27] by setting the *number of clusters* equal to half of the number of the input documents and the *number of iterations* equal to 10.000, using the LD matrix as metric.

Before running the algorithm, some constraints have been established to guarantee its stability:

1. Run the k-means algorithm until there are no improvements in the ***B**etween **S**um of **S**quares/**T**otal **S**um of **S**quares* (Bss/Tss) index output; we have experimented that at least five consecutive iterations of the k-means algorithm are necessary. The Bss/Tss index measures the goodness of the algorithm [28]. Ideally, we want a clustering that has the properties of internal cohesion and external separation (i.e. the Bss/Tss ratio as close to 1 as possible).
2. At each execution, it is necessary to remove the cluster containing a single document from the result; the new set of documents, decreased by one, will represent the input for the next execution of the algorithm, with the threshold decreased by one.
3. The minimum number of possible clusters is 2.

The choice of these constraints significantly conditions the outcome of the Data Integration; they were chosen after various tests and following some heuristics available in the literature [29]. The output of this algorithm will influence the integration of the data sources.

The results are reported in Table 1, where each row represents an iteration result of the k-means algorithm. In this table: (k) is the number of clusters in the current iteration; (N) is the number of documents involved; (*Documents*) shows the remaining documents (one of the documents is deleted at each iteration and can be viewed in the 'Doc Out' column); (*Cluster*) shows the clusters

Table 1
K-Means for Syntactic analysis of CarShopping System

Iter	Param		Documents	Clusters	Bss/Tss	Doc Out
	k	N				
1	4	8	1,2,3,4,5,6,7,8	C_1={1,2}, C_2={3,4,6}, C_3={5,7}, C_4={8}	85,0%	8
2	3	7	1,2,3,4,5,6,7	C_1={1,2}, C_2={3,4,6}, C_3={5,7}	82,5%	

1=clients 2=customers 3=employees 4=offices 5=orders 6=payments 7=productlines 8=products.

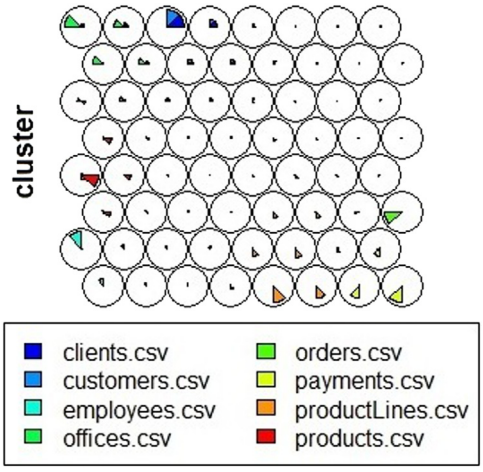

Fig. 4. SOM grid (64 Unit SOM) for CarShopping System.

that the documents under consideration form and finally (*Bss*/*Tss*) is the k-means cohesion coefficient for that iteration.

The table shows that the clustering of the first seven documents into three clusters requires two algorithm executions, with a fitting rate of roughly 83%. The existence of matching between the two source schemes is suggested by the clusters C1 and C2 obtained during the second execution.

3.4. Semantic analysis phase

Starting from the pre-processed files, the semantic analysis [30] is performed to find their similarity. By using the tables from the two normalized schemes and the TDM as input, we ran the following three steps to find the semantic correlations between the tables:

1. Unsupervised training of the SOM network [31,32];
2. Calculation of the covariance matrix of the weights assigned by the SOM to the documents;
3. Cluster analysis.

1. *Training of the SOM Network.* The training set for the SOM consists of the reduced TDM, obtained by removing the sparse terms. The sparsity ratio was set to 0.95 since for values lower than this threshold it would result the removal of an excessive number of terms (more than half of the total) while for values greater than it would result the removal of a too little number of terms (about 1%). The value of 0.95 has led to the elimination of the 15% of the total words. The output will be a two-dimensional grid, where in each SOM unit (or cell) there will be similar documents. In particular, the neighboring cells will contain groups of documents that share similar features. An example of the obtained grid for the *CarShopping* System is shown in Fig. 4.
2. *Calculation of the covariance matrix of the weights assigned from the SOM to the documents.* After the initial training of the unsupervised SOM network, the second step is to find a classification of the output weight matrix assigned to each document. Considering n input documents, the SOM output vectors weights will be contained in a matrix $n^2 \times n$, that represents the content of each SOM unit. It will be normalized to the interval $[0, 1]$. It is important to note that in this phase all the non-significant elements will be excluded; the excluded elements are those that have much lower values than the maximum weight associated with the SOM units, because they are not competitive. For this purpose we set these values to 0.
Each column vector will therefore have n^2 components (the ith component will represent the weight assigned by SOM for that vector to the ith SOM unit), and the covariance is thus calculated for each pair of column vectors. The result will be the so-called *covariance matrix*, which relates every possible pair of documents, showing the variation of each variable with respect to the others. This matrix will be the input for the cluster analysis phase.
3. *Cluster analysis.* The last phase of the semantic analysis is the clustering phase, whose goal is to understand how documents are located in the various clusters, through the use of the covariance matrix to find semantic similarity. The process will follow the same path made for the syntactic analysis; the results are shown in Table 2.

3.5. Comparison of the results obtained by the Syntactic and Semantic analyses

The final step of the integration process concerns the comparison of the results achieved by the syntactic and semantic data analyses, in order to understand which of them better "captures" the matching between data sources. The starting inputs of the two analyses are different; first of all it is very important to determine an index that can highlight which is the best reached clustering for the goal, since in both the analyses the k-means is used as clustering algorithm. The chosen index is the one already defined for the output of the k-means algorithm, that is Bss/Tss. This index is very representative of the k-means algorithm, because it indicates the goodness of the input data fitting. Since this index is not comparable for the two analyses, it is necessary to define a new index I_{SynSem} that takes into account the results of both the syntactic and semantic analyses; the new index can be defined as the linear combination of the fitting indexes of the k-means of the two analyses:

$$I_{SynSem} = \alpha * Max_{syn}(Bss/Tss) + \beta * Max_{sem}(Bss/Tss) \quad (4)$$

where :

- $\alpha, \beta \geq 0$;
- $\alpha + \beta = 1$;
- Max_{syn} is the Best Index for the syntactic analysis;
- Max_{sem} is the Best Index for the semantic analysis.

As a result, the setting of α and β parameters will allow the Company Manager to choose which of the two analyses produced the best clustering findings, by assessing the dendrograms, the SOM map, the outcomes of the two k-means algorithms, as well as his business experience. Generally, the semantic analysis is more

Table 2
K-Means for Semantic analysis of CarShopping System.

Iter	Param		Documents	Clusters	Bss/Tss	Doc Out
	k	N				
1	4	8	1,2,3,4,5,6,7,8	C_1={1,2}, C_2={3,5,6}, C_3={7,8}, C_4={4}	86,4%	4
2	3	7	1,2,3,4,5,6,7	C_1={1,2}, C_2={3,5,6}, C_3={7,8}	82,1%	

1=clients 2=customers 3=employees 4=offices 5=orders 6=payments 7=productlines 8=products.

Table 3
Match Matrix for Tables clients and customers of the CarShopping System.

		Customers					
		customerNumber	customerName	contactLastName	contactFirstName	phone	...
Clients	**clientId**	0	0	0	0	0	...
	clientName	0	**122**	0	0	0	...
	clientPhone	0	0	0	0	**121**	...
	clientAddress	0	0	0	0	0	...
	...	...	...	...	...	...	...

exhaustive, so it is preferred. So the Company Manager usually tends to choose $\beta > \alpha$.

The intersection between syntactic (82.5% fitting) and semantic (82.1% fitting) analyses, highlights the cluster formed by the {*clients*, *customers*} files, so the integration focuses on these two tables, which describe exactly the same domain of interest. Since the semantic analysis is more exhaustive, a good choice to make it prevail over the syntactic one, could be $\alpha = 0.40$ and $\beta = 0.60$.

3.6. Data cleansing phase

The *Data cleansing* phase refers to the identification of incomplete, irrelevant or incorrect parts of the data records and also to the replacement or modification of the same data records, by eliminating the dirty or coarse ones [33]. After this "cleansing", a dataset should be consistent with other similar datasets. In this process, it is adopted the *Schema-Level* approach [34], which attempts to create correspondence between various files or database structures by taking advantage of appropriately defined similarities. By using the k-means algorithm results, we find similar tables by clusters and identify the fields that may correspond. This method aims to determine the matching between tables to be merged (in a cluster there are at least two tables to merge).

In this phase, the entire tables are considered, except columns with numeric attributes that will be kept only to support the merging (with the remaining tables that should not be integrated). The data cleansing phase therefore aims to determine a correspondence matrix between the columns of the tables to be merged; here the case for two tables is shown, but the method works for any number of tables by means of successive iterations. Let us assume to have two tables in a cluster: A (of m columns) and B (of n columns) that can be merged together. Then we construct a $m \times n$ matrix where each cell (i, j) contains the number of elements, that occur both in the ith attribute of table A and e in the jth attribute of table B. In the comparison only alphanumeric elements are considered. In this matrix it is necessary to search the maximum value for each row (or column) and this value will identify the attributes to be unified.

Table 3 shows part of the results for the tables *clients* and *customers* of the *CarShopping* System. We can note correspondences between the attributes *(clientName, customerName)*, and between (*clientPhone*, *phone*). These matching attributes will be considered key attributes for the subsequent merging of the two tables.

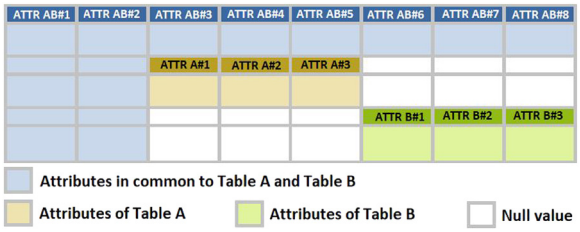

Fig. 5. Merging of tables A and B.

3.7. Sources integration phase

After the Data Cleansing phase, the two tables must be merged. To do this, it is necessary to use the join operation of the relational databases, considering the columns of the key attributes. In the algorithm, both inner join and full outer join were used; the keys involved are those produced by the data cleansing phase. With reference to Fig. 5, if A and B are the tables to be merged, the integrated table will contain first the columns in common to A and B (the part in blue in Fig. 5), followed by the remaining columns of the two tables that do not contribute to the merging (for Table A the part in yellow, and for Table B the part in green in Fig. 5). The Sources Integration Phase and the Data Cleansing phase can be iterated when the number of tables to be integrated is greater than two (see Fig. 1).

For the *CarShopping* System, the integration proceeds performing the inner join of the two files *clients* and *customers*. The new integrated table will be composed of 16 columns where 2 are shared by the tables (one had 5 columns, the other 13). After this, it is simple to obtain the construction of its reconciled sources schema (see Fig. 6). It is important to underline that data cleansing and merging of the tables will also consider the columns of numeric type, because these columns can be foreign keys and must therefore be preserved to guarantee the connections (i.e., relationships) of the new integrated table (A+B) with those present in the sources that are related to the single tables of the two initial databases.

4. Case study

In order to test the quality of the proposed approach, extensive tests were carried out on three more systems: *Panda, Plants* and

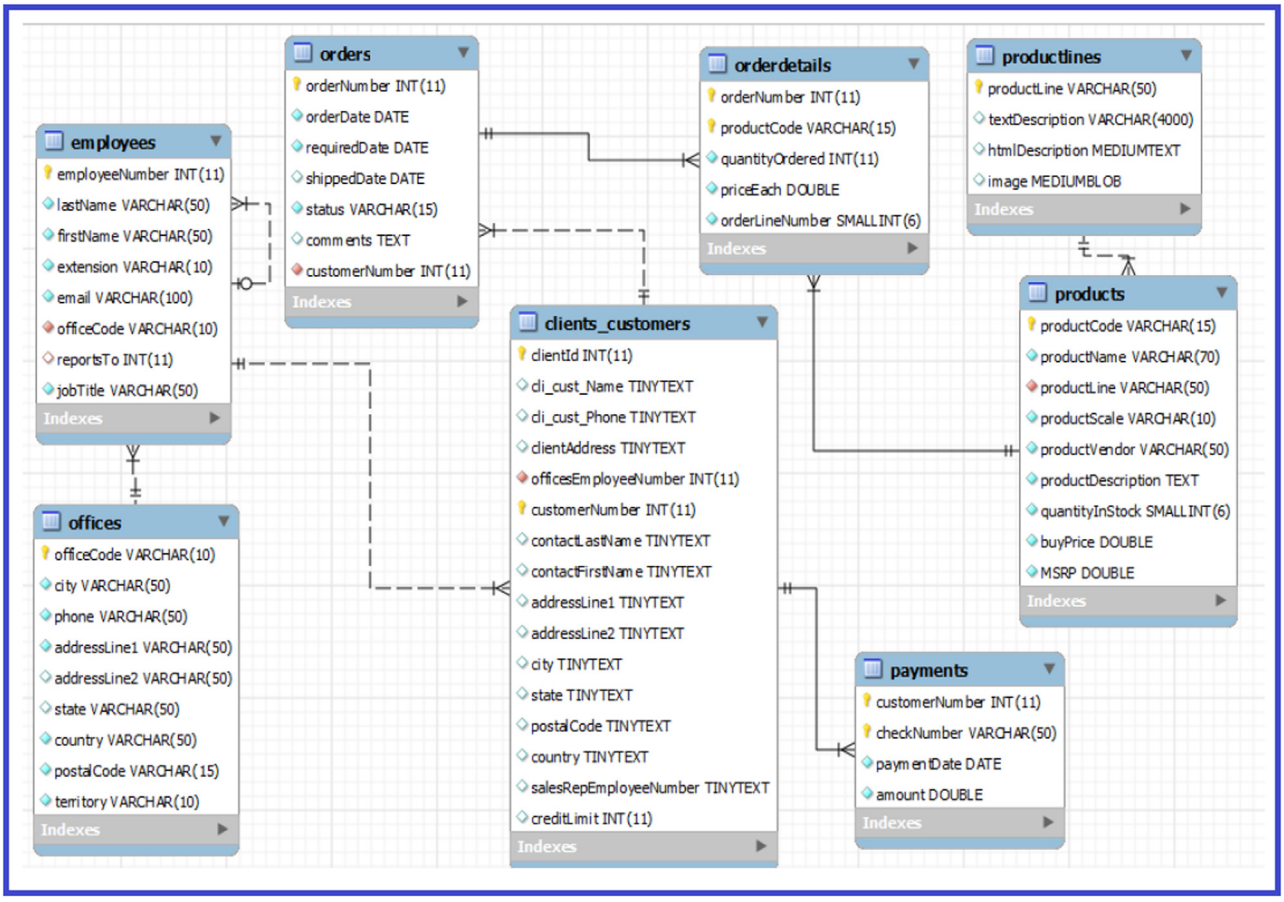

Fig. 6. Reconciled Schema for the CarShopping System.

Cooking. All details are published on github.[1] Here we summarize the results obtained.

4.1. The Panda system

System description. The PANDA system (Presences And Notes Data Analysis using Data Warehousing) is a DW that allows the execution of analyses on typical university activities, focusing on students' productivity.

Sources. The system uses two relational databases to support its business:

1. *SDI (Student Digital Identification)* is a system for detecting the attendance in the university context, used from 2008 to 2010 by the University of Molise, made up of 4 tables. The system database stores around 18,000 tuples for attendance and over 54,000 tuples for users;
2. *UnderDesk*, is a web application that allows users to share their notes. The system stores more than 54,000 users and 3,000 notes in 5 tables.

The final fittings of the syntactic and semantic analyses are 91.9% and 95.2% respectively. In this case, the two analyses suggest the same tables to be integrated, so the semantic analysis prevails since it is more thorough. So a good choice could be $\alpha = 0.45$ and $\beta = 0.55$.

[1] https://github.com/mbUnisa/A-Semi-automatic-Data-Integration-process.

4.2. The plants system

System description. The third examined system regards the creation of a DW for the management of flower sales services, by integrating two data sources from two different companies working in that sector.

Sources. The system uses two relational databases:

1. *DB Vivaio E-commerce*, a database of a flower sales company that does sales plants and several related services, composed of 6 tables;
2. *DB Vivaio Interventi*, a database of a company that deals exclusively with the provision of flower sales services, similar activities of the first company, composed of 3 tables.

The final fittings of the syntactic and semantic analyses are 88.7% and 90.2% respectively. In this case, the details of the analyses show that the syntactic analysis does not suggest the possibility of integrating the two tables *cliente_interventi* and *cliente_vivaio* which intuitively, from the name, would suggest the existence of possible correlations. In fact, the subsequent semantic analysis is able to determine it. So a good choice to make semantic analysis prevail over syntactic analysis could be $\alpha = 0.45$ and $\beta = 0.55$.

4.3. The cooking system

System description. As a last case study it is proposed a system to integrate three sources of data modeling the world of the kitchen. In particular, there are three CSV files that can be freely downloaded from the open data catalog, made available by the

Table 4
System test results table.

System	Source num	Source tables	Integrated tables	Fitting syn Fitting sem	α β	Integrated tables sizes	Accuracy
CarShopping	2	3-5	2	82.5% 82.1%	0.40 0.60	A=122 B=122	100%
Panda	2	4-5	2	91.9% 95.2%	0.45 0.55	A=54458 B=54110	99%
Plants	2	6-3	2	88.7% 90.2%	0.45 0.55	A=32 B=14	17%
Cooking	3	1-1-1	2	99.6% 78.7%	0.35 0.65	A=109 B=49	100%

Legend System = Name of the system under investigation.
Source Num = Number of system sources to be integrated.
Source Tables = Number of tables with not only numeric data involved, for each of the starting sources.
Integrated Tables = Number of tables integrated for the system under investigation.
Fitting Syn/Fitting Sem = Best fitting indexes (Bss/TSs) achieved in the analyses.
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ = Parameters decided by the Company Manager on the basis of all available tools.
Integrated Tables size = Number of records of the integrated tables.
Accuracy = Measure of the quality of integration.

italian Public Administration Ministry.[2] *Sources.* The three files describe:

1. *ospitalita*, containing essential data for catering;
2. *prodottiTradizionali*, a list of traditional Trentino products, listed by name, category, and production area;
3. *ricette*, a collection of typical Trentino recipes of traditional Trentino products (appetizers, first dishes, second dishes, desserts).

The final fittings of the syntactic and semantic analyses are 99.6% and 78.7% respectively. Similarly to the *Panda system* even in this case, the syntactic and semantic analyses suggest the same tables to be integrated. So, if required to select between them, the semantic one usually prevails since it is more exhaustive. Therefore a good choice to make semantic analysis prevail over syntactic analysis could be $\alpha = 0.35$ and $\beta = 0.65$.

4.4. Final results comparison

The results obtained for the four analyzed systems are reported in Table 4; here, for every tested system all the features are reported (number of sources and total tables with not only numeric data of each source database, number of integrated tables, fitting of syntactic and semantic analyses and α and β parameters, original sizes of the integrated tables and obtained accuracy). The accuracy measures the quality of integration, evaluated by comparing the results obtained from the process described, with that produced as an oracle by experts in the sector. By our tests, it can be said that the described procedure is valid in the range between 99% – 100% in case of consistent starting data. For completeness, we have considered a case of high level of inconsistency starting data, the Plants system. In that case, the reached accuracy is of 17%. This result is not caused by a process error, but from a syntactic anomaly in one of the columns mapped by the process. The integration process gave rise to a reconciled table containing only one tuple, caused by the too small size of the source tables.

5. Conclusions and future work

The process presented in this paper focuses on the Source DI phase, taking advantage of Data Mining and ML techniques. Here a semi-automatic procedure has been proposed, with the marginal involvement of the Company Manager. The obtained results are very satisfactory in terms of accuracy, in the range 99% – 100%, for the considered systems. In the future, we plan to make the process completely automatic, by implementing algorithms able to self-determine acceptable error thresholds, making the Company Manager intervention useless. Furthermore, the schemes resulting from the integration, could be optimized by eliminating redundant columns. A further step towards the optimization of this procedure will be the definition of a strategy for a one-shot integration of n sources.

[2] http://www.dati.gov.it.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgment

The authors sincerely wish to remember Michele Risi, who actively contributed to this work, but passed away at a young age a few months ago.

References

[1] A. Doan, A. Halevy, Z. Ives, Principles of Data Integration, Elsevier, 2012.
[2] C. White, Data integration: using ETL, EAI, and EII tools to create an integrated enterprise, Bus. Intell. J. 10 (I) (2005).
[3] P. Szabó, Data Virtualization and Federation, Stone Bond Technologies, 2014.
[4] F. Boufares, A.B. Salem, Heterogeneous data-integration and data quality: overview of conflicts, in: 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), IEEE, 2012, pp. 867–874.
[5] Y. Yang, M. Chen, B. Gao, An effective content-based schema matching algorithm, in: International Seminar on Future Information Technology and Management Engineering, IEEE, 2008, pp. 7–11.
[6] S. Bergamaschi, D. Beneventano, A. Corni, E. Kazazi, M. Orsini, L. Po, S. Sorrentino, The open source release of the MOMIS data integration system, in: SEBD, Citeseer, 2011, pp. 175–186.
[7] G. Liu, S. Huang, Y. Cheng, Research on semantic integration across heterogeneous data sources in grid, in: Frontiers in Computer Education, Springer, 2012, pp. 397–404.
[8] H. Ibrahim, Y. Karasneh, M. Mirabi, R. Yaakob, M. Othman, An automatic domain independent schema matching in integrating schemas of heterogeneous relational databases, J. Inf. Sci. Eng. 30 (5) (2014) 1505–1536.
[9] O.A. Mehdi, H. Ibrahim, L.S. Affendey, An approach for instance based schema matching with google similarity and regular expression, Int. Arabian J. Inf.Technol. 14 (5) (2017) 755–763.
[10] P. Papotti, F. Naumann, S. Kruse, et al., Systems and methods for data integration, US Patent 10,528,532, 2020.

[11] T. Sahay, A. Mehta, S. Jadon, Schema matching using machine learning, in: 7th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2020, pp. 359–366.
[12] J. Zhang, B. Sin, J.D. Choi, J.C. Ho, SMAT: an attention-based deep learning solution to the automation of schema matching, in: European Conference on Advances in Databases and Information Systems, Springer, 2021, pp. 260–274.
[13] C. Ma, B. Molnár, A. Tarcsi, A. Benczùr, Knowledge enriched schema matching framework for heterogeneous data integration, in: 2022 IEEE 2nd Conference on Information Technology and Data Science (CITDS), 2022, pp. 183–188.
[14] L. Guo-Hui, D. Xiao-Kun, H. Fang-Xiao, D. Jian-Qiang, A schema matching method based on partial functional dependencies, in: Japan-China Joint Workshop on Frontier of Computer Science and Technology, IEEE, 2008, pp. 131–138.
[15] C.C. Aggarwal, C. Zhai, Mining Text Data, Springer Science & Business Media, 2012.
[16] G.G. Chowdhury, Introduction to Modern Information Retrieval, Facet publishing, 2010.
[17] A. Khan, B. Baharudin, L.H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, J. Adv. Inf. Technol. 1 (1) (2010) 4–20.
[18] T. Hofmann, Probabilistic latent semantic indexing, in: 22nd Annual International ACM SIGIR conference on Research and Development in Information Retrieval, 1999, pp. 50–57.
[19] B. Rosario, Latent semantic indexing: an overview, INFOSYS 240 (2000) 1–16.
[20] K.E. Heinrich, R. Homayouni, B.R. Silver, System and method of prediction through the use of latent semantic indexing, US Patent 10,224,119, 2019.
[21] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E.D. Trippe, J.B. Gutierrez, K. Kochut, A brief survey of text mining: Classification, clustering and extraction techniques, arXiv preprint arXiv:1707.02919 (2017).
[22] A.G. Jivani, A comparative study of stemming algorithms, Int. J. Comput. Technol.Appl. 2 (6) (2011) 1930–1938.
[23] I. Antonellis, E. Gallopoulos, Exploring term-document matrices from matrix models in text mining, cs/0602076 (2006).
[24] S. Zhang, Y. Hu, G. Bian, Research on string similarity algorithm based on Levenshtein distance, in: 2nd International Conference on Advanced Information Technology, Electronic and Automation Control (IAEAC), IEEE, 2017, pp. 2247–2251.
[25] F. Bourennani, M. Guennoun, Y. Zhu, Clustering relational database entities using k-means, in: 2nd International Conference on Advances in Databases, Knowledge, and Data Applications, IEEE, 2010, pp. 143–148.
[26] C.K. Reddy, B. Vinzamuri, A survey of partitional and hierarchical clustering algorithms, in: Data clustering, Chapman and Hall/CRC, 2018, pp. 87–110.
[27] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.
[28] M. Chavent, F.d.A. de Carvalho, Y. Lechevallier, R. Verde, New clustering methods for interval data, Comput. Stat. 21 (2) (2006) 211–229.
[29] D.T. Pham, S.S. Dimov, C.D. Nguyen, Selection of k in k-means clustering, Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 219 (1) (2005) 103–119.
[30] Y. Xue, W. Liu, B. Feng, W. Cao, Merging of topic maps based on corpus, in: International Conference on Electrical and Control Engineering, IEEE, 2010, pp. 2840–2843.
[31] H. Ritter, T. Martinetz, K. Schulten, D. Barsky, M. Tesch, R. Kates, Neural Computation and Self-Organizing Maps: An Introduction, Addison-Wesley, Reading, MA, 1992.
[32] S. Kaski, Data exploration using self-organizing maps, in: Acta Polytechnica Scandinavica: Mathematics, Computing and Management in Engineering Series no. 82, Citeseer, 1997, pp. 1–57.
[33] E. Rahm, H.H. Do, Data cleaning: problems and current approaches, IEEE Data Eng. Bull. 23 (4) (2000) 3–13.
[34] P. Shvaiko, J. Euzenat, A survey of schema-based matching approaches, in: Journal on Data Semantics IV, Springer, 2005, pp. 146–171.