Extracting a Unified Database from Heterogeneous Bills of Materials via Data Integration Approaches

Luigi De Lucia[1,2], Guglielmo Faggioli[1], Nicola Ferro[1] and Fabio Orobelli[2]

[1]*University of Padova, Padova, Italy*
[2]*Bedeschi S.p.A., Limena, Italy*

Abstract
Since the beginning of the information era, manufacturing companies have collected a huge amount of data and acknowledged the importance of proper collection and curation policies. Nevertheless, an open problem is the exploitation of historical and legacy data. Such data is heterogeneous: it has often been accumulated throughout decades by different curators with different degrees of attention toward the data. In this work, we focus on the historical data collected by Bedeschi S.p.A., an international company dealing with the design and production of turn-key solutions for Bulk Material Handling, Container Logistics and Bricks. Our task is to extract knowledge, available under the form of partially structured Bills of Materials (BoMs), to distil a unified database over a mediated schema that can be used by designers and sales department to rapidly have a global overview of the products developed over the years. We present a feasibility study investigating the performance of several traditional approaches of data integration applied to our specific task. In particular, we treat BoMs as sources that need to be aggregated in the unified mediated schema. We achieved satisfactory results indicating the direction that the company should follow to properly extract knowledge from their historical data.

1. Introduction

Data is arguably becoming day by day one of the most valuable assets for an increasing number of companies; large companies have digitalized several production processes, making it easy to collect, curate and exploit their data. This is not true for what concerns historical and legacy data, which is often underlooked and poorly used. Three main challenges impair the handling of historical data. First, legacy data is fragmented and scattered over different sources, formats and media, making it challenging to organize it as unitary knowledge. Secondly, historical data is unstructured and weakly linked to the other information, it also contains dirty values and information expressed via natural language [1]. Finally, historical data is characterized by high degree of heterogeneity: several annotators collected it on digital archives over decades, with underlying production processes and paradigms changing. These challenges make legacy and historical datasets unexploitable automatically. Our work stems from a specific industrial need: exploiting historical data to help the design and sales departments by providing them with a unifying overview of all the products built over the years. We exploit historical data collected by Bedeschi S.p.A, an international company specialized in the production of turn-key solutions for Bulk Material Handling, Container Logistics and Bricks. We aim at extracting information from a set of heterogeneous semi-structured Bills of Materials (BoMs) to organize

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy

it in a uniform database over a global schema providing an overview of the different models of Bedeschi machines developed over decades. This work is a feasibility study to determine whether it is possible to process these BoMs automatically. In particular, to determine the most promising approaches, we focus on a single Bedeschi machine type, the BEL-C, in production at Bedeschi S.p.A. for decades. Furthermore, machines of this type have rich BoMs that allow us to recognize emerging challenges and critical issues. Moreover, different designers and curators have produced, collected, and organized the data about this type of machine over the years using different media, starting from paper sheets – now digitalized – to modern information access systems.. We interpret our task under the data integration framework. In particular, we consider BoMs associated with different machines as our data sources. These data sources are then integrated to obtain a global schema.

Our contributions are the following:

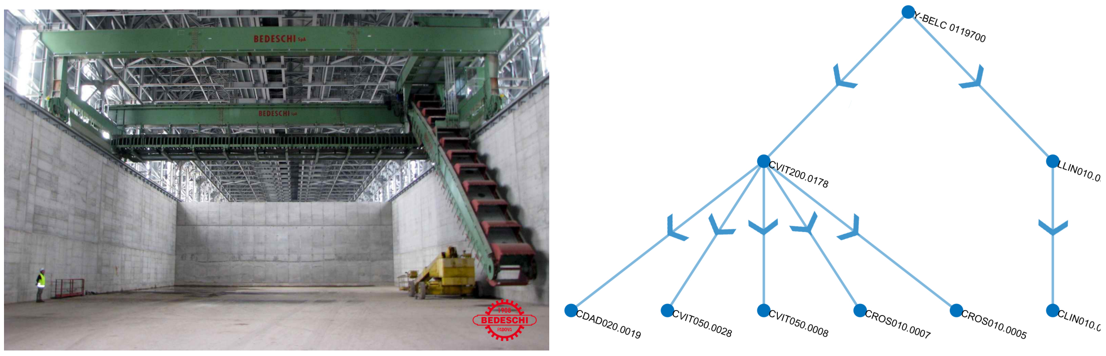
- We formalize the unified database extraction task from an industrial perspective as a data integration task;
- We identify a set of techniques drawn from the schema matching and instance matching domains to address the task;
- We empirically evaluate the performance of such techniques.

The remainder is organized as follows: Section 2 describes the main efforts in the Data Integration domain. Section 3 contains the description of the task at hand and the data available. Section 4 details on our methodology, while Section 5 contains its evaluation. Finally, in Section 6 we draw same conclusions and outline future steps.

2. Related Work

Data integration aims to provide unified access to data that usually resides on multiple, autonomous data sources, like different databases or data retrieved by internet websites [2]. One of the major challenges to data integration is data ambiguity. This ambiguity arises whenever there is some uncertainty about the meant data information. Among data integration systems developed, some handle the whole data integration process, such as Do and Rahm [3], Fagin et al. [4], Seligman et al. [5], while others, e.g., Bernstein et al. [6], Falconer and Noy [7], focus only on the schema mapping task. Among the most prominent endeavours, we list the following. COMA [3, 8, 9] automatically generates matchings between source and target schemas, by exploiting multiple matching approaches to compute the matching. Clio, developed by Fagin et al. [4], supports a visual matching representation, and is capable of working with schemas without knowing the mediated schema. Cupid [10] is a schema matcher algorithm that exploits similarities of different aspects of the schema elements: keys, referential constraints, and views.

Several different endeavours have been devoted to the schema matching task. Rahm [11] presents an overview of the possible approaches based on different types of matching to detect the correspondences between the schema elements. When the instances are available, it is possible to state that two schema elements are similar if they have close instances Bernstein et al. [6], Madhavan et al. [12].



(a) BEL-C (b) Bill of Materials (BoM)

Figure 1: A BEL-C machine and – a part of – its BoM

Additionally, different works exploit the information provided by instance values. COMA++, an extension of COMA proposed by Engmann and Massmann [8], considers the average length of instance characters, average values and standard deviations. A second approach has been proposed by Bilke and Naumann [13] and exploits duplicate instances to find matches. Despite the popularity of the task, there is no universal benchmark to evaluate schema matching approaches [14].

Among the ancillary techniques needed to achieve data integration, we can list data profiling, reverse database engineering, and matching approaches combination. Data profiling is the set of processes and actions performed to retrieve specific information, also called *metadata*, about a given dataset [15]. Generally, profiling tasks support manual data exploration and improve the data gazing [16] process. Data profiling further helps data integration by revealing inconsistencies and errors in the data errors [17]. Reverse database engineering allows identifying different characteristics of the database entities like relations, attributes, domain semantics, foreign keys and cardinalities that can be introduced into the data integration process via explicit rules [18]. Combining different matching approaches allows detecting a higher number of correct matches with respect using a single matcher [19]. We can see some approaches in [12, 10, 20]. Hybrid and composite matchings are two different ways of combining matching approaches. The former uses multiple matching criteria together to compute the score. The latter, which we employ in our work, computes the matching scores independently and then combine them.

3. The Task and Dataset

3.1. Task

In the design process of a new product, it is common practice to look at the design of previous ones to retrieve reference specifications and understand the design choices. To this end, it is necessary to have an overview of products realized previously. In this work, we focus on a specific product: a machine type, called BEL-C, which has been in production for the Bedeschi company for decades and has multiple versions. Figure 1a depicts an example of BEL-C machine: this machine includes several "parts", called assemblies, e.g., the mechanical arm on the right side is an assembly composed of multiple articles such as wheels and buckets.

The description of each realization of a BEL-C machine is available under the form of Bill of Materials (BoM). A BoM is the list of the components, also called *articles*, needed to manufacture a certain product. Multiple articles can be further aggregated into *assemblies*. Thus, the elements in a BoM are organized hierarchically as a tree. The root node represents the final product, while internal and leaves nodes represent respectively assemblies and articles. Figure 1b illustrates the apical area of the BoM for the machine on Figure 1a.

BoMs referring to different products of the same type, as in our case, might be highly heterogeneous. In fact, they might have been realized decades apart, can be the digitalization of old analogical production data and might include heterogeneous descriptions. This results in data ambiguities that need to be resolved. We aim to realize a tool that resumes all the relevant information of the different BEL-C machines produced during the years. More in detail, we have a manually compiled example of a database constructed over a mediated schema and we want to learn how to extract a similar database automatically from the BoMs. Such example unified database is based on recent realizations of the BEL-C. Therefore, mapping historical BoMs into the database over the mediated schema presents two main challenges: i) example data is expressed in natural language and it is not possible to find the very same attribute values among our data; ii) modern BoMs used to construct the example of unified database are not among the historical ones and thus are not in our dataset. Therefore, we cannot use traditional entity resolution and linkage approaches. Even though they share similar structure, the heterogeneity between different BoMs makes them closer to different data sources rather than to different instances of the same database. In this sense, the schema underlying the unified database that combines the information on different BoMs can be seen as a mediated schema. Therefore, We model our task in a data integration framework as follows: given a set of BoMs, the description of the articles they contain and an example of unified database over the mediated schema, we have to: *1)* understand how BoMs are mapped into the mediated schema; *2)* construct automatically the unified database for previously unseen BoMs.

3.2. Dataset

To address our task, we have a set of BoMs, a database of the articles, and a manually built example of the required final mediated schema.

BoMs Different BEL-C machines might have diverse specifications, i.e., the same component can have different dimensions or distinct motors can be installed. Each BoM, identified by a unique serial number, is an instance representing a specific realization of a BEL-C machine with its characteristics. The BoM contains the articles' ID used in building the machine. Inside each BoM articles might be organized into assemblies. Each assembly has a unique identifier and a description. The assembly description allows understanding what part of the machine the tree portion is referring to. In our historical data, we have BoMs associated with 16 distinct machines. BoMs contain from 6 to 375 articles.

Articles The article database contains the description of the characteristics of articles listed in BoMs. Articles are organized into *classes* – e.g., chains, engines, etc – uniquely identified

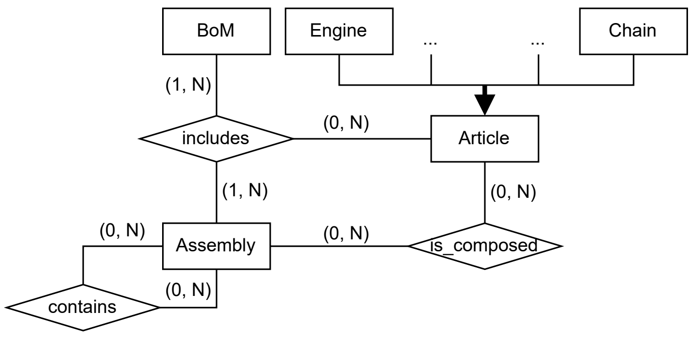


Figure 2: The ideal Entity-Relationship Schema of our data. Each BoM includes multiple assemblies and/or single articles. An assembly can be composed either by articles or by other assemblies. Articles are divided into several different classes.

by a mnemonic code. Articles belonging to different classes have distinct sets of attributes. The domains of such attributes are not formally described: in most cases, attributes values are expressed via natural language strings even when only typed (e.g., numeric or Boolean) values are expected. The only attribute common to all classes is "description": a free text field containing the description of the article and its usage. Several articles have multiple null values – such values are not really empty: they have not been correctly reported or lost while moving from paper to digital archives. In this sense, the "description" attribute is often the only one (ab-)used, containing also the values for all the other attributes. Figure 2 reports the ideal Entity–Relationship (ER) schema for our dataset, BoMs are composed of assemblies or articles and assemblies include either other assemblies or articles.

Example Database over the Global Schema To understand which are the relevant attributes and their domain, domain experts provided us an example of the database constructed over the global schema that we would like to distil from the BoMs. This database was constructed manually and often the values for its fields are not in the domain of those in the articles dataset – e.g., different units of measure, strings instead of numbers. Furthermore, it is based on recent BEL-C machines that are not among historical BoMs. The main challenge is understanding how to map articles' attributes to the global schema fields comparing their domains and relations. The example database organizes the data as follows: each column contains a different machine realization – 20 of them – while each row represents a machine specification. Our dataset includes 47 unique *fields* for the machine specification. Rows are divided into *groups* – e.g. technical features, chain – that provide semantic meaning to the fields they include.

Manual mapping As aforementioned, historical BoMs and those available in the example of unified database do not overlap. Furthermore, values for fields in the example unified database are manually set and not available in the domain of the attributes in the article's data set. Therefore, we cannot determine whether the automatic mapping produced using data integration techniques is correct. Thus, with the help of domain experts, we built a manual mapping between global schema fields and articles' attributes. Such mapping allows us to define a query that provides the mediated schema immediately. The construction of such manual mapping took several hours and has two main drawbacks: it does not work on different machines besides the BEL-C, nor it works if new attributes are taken into account. Therefore, we use the manual mapping to evaluate different automatic integration approaches, so that they can be used on previously unseen scenarios for which we do not have time or resources to construct the manual mapping.

4. Methodology

Our objective is to understand the mapping between fields of the target schema and article attributes. Given $\mathcal{F}$ the set of fields of the mediated schema and $\mathcal{A}$ the set of articles' attribute, for each $f \in \mathcal{F}$ our objective is to select the articles attribute $a \in \mathcal{A}$ that maps into f. To do so, we compute a similarity measure $s(f, a)$ which describe how similar are f and a. Then the value of the field f is extracted from the attribute a such that:

$$\underset{a \in \mathcal{A}}{\operatorname{argmax}}\ s(f, a)$$

Our task becomes identifying the best similarity function s to describe the relation between a field f and an article attribute a. To do so, we model three aspects: the similarity function s, the representation for the field f and the representation for the article attribute a. Depending on the representation used for f and a, there are two main categories of matching approaches: schema-based and instance-based matching approaches. The former relies on schema information about both field and article, such as their names, group or class. Instance-based matching approaches represent articles and fields using their instances. A third class of approaches is the one based on composite matchings, where the outputs of multiple matchers are combined.

Finally, using the manual mapping developed with the contribution of two domain experts, we can evaluate the performance of each similarity function or mapping strategy.

4.1. Data Normalization

Following common practices in the schema matching domain [10, 21], we start by normalizing the entities in our dataset. The normalization pipeline includes converting characters to lower-case, removing trailing spaces, removing non-informative tokens (e.g., single characters and stopwords).

4.2. Schema-Based Matching

Using the schema information, such as labels, data types, and structural properties is a widely used approach [3, 10, 12, 22, 23]. In this scenario, we can exploit elements from the schema of the unified database, the mediated schema, and the schema of the articles database. For the mediated schema fields f we consider the following representations:

- Field name (F): names of the attributes of the target schema.
- Group name (G): name of group the field is in.

- Concatenation between the field and the group it belongs to (GF).

Concerning the articles attribute a, on the other hand, we can represent them as follows:

- Article attribute name (A): name of the article attribute in the articles database.
- Article class (C): Name of the article class. Being a mnemonic code, it contains relevant semantic information.
- Concatenation between the article class and attribute name (CA).

Once we have defined the representation for the fields and articles, we need to define the similarity s. In particular, we employ the following matching approaches:

- Dice's coefficient (CW): size of the intersection between the terms of two strings.
- Edit Distance (ED): edit operations required to transform one string into the other.
- Pairwise Edit Distance (EDP): if two strings contain identical words in diverse order, they have a high edit distance. EDP computes the edit distance between each possible pair of tokens from the two strings, returning the average distance of all the combinations.
- N-Grams (NG_n): number of common n-grams between two strings [24].

Besides the pairwise edit distance, normalized by default, all the above-mentioned matching approaches are employed in both their original and normalized forms.

4.3. Instance-Based Matching

In the case of instance-based matching, we represent the field f and the article attribute a using the values that they assume for the different instances. In particular, given a field f, we can concatenate the values of that field for each BoM in the example global schema and obtain a document d_f. Similarly, taking all the values for the attribute a in the articles database, we concatenate them and define a document d_a, that represents the article. Given the size of d_f and d_a, the similarities used for the schema-based scenario are computationally too demanding. Therefore, we propose to use documents' similarity measures, such as those typically used in Information Retrieval (IR). Given a document field d_f we rank all the attribute documents d_a, based on their similarity, and select the attribute that maximizes the similarity with the document field. Among the similarity measures s, we investigate the following: Jaccard's coefficient [25, 26], cosine similarity [25] and BM25 [27].

4.4. Composite Matchings

To further improve matching performance, we also investigate matching strategies fusion. In particular, similarly to what typically done in IR, for each field, we rank attributes based on their similarity with the field. Then, we fuse different rankings using CombSUM and CombMNZ [28].

We also propose an approach, dubbed *weighted average* (WA), that weights the similarity computed by different matching approaches as follows:

$$\mathrm{WA}(f,a) = \frac{\alpha s_1(f,a) + \beta s_2(f,a) + \ldots + \gamma s_M(f,a)}{\alpha + \beta + \ldots + \gamma} \quad (1)$$

where $s_1, s_2, \ldots, s_M$ is the set of similarity matrices.

5. Empirical Evaluation

The subsequent section contains the empirical analysis of the proposed methodology. We adopt Mean Reciprocal Rank (MRR) [29] to evaluate our matching strategies. Furthermore, we also include the proportion of exact matches (precision at 1), using two distinct tie-breaking policies: "first" which considers the first match in alphabetical order and "any", which measures whether, among the ties, one of them is the correct match. To understand the statistical relevance of the different matching strategies performances, we conduct an ANOVA test.

5.1. Schema-Based Matching Evaluation

The performances of schema-based matching strategies are available in Table 1. Concerning the MRR, the best performing strategy is 3-gram algorithm with "Field/Class and Attribute" input. This configuration achieves the best MRR score 33.6% among the set of tested matching strategies, but it's statistically different only from 9 other configurations. Table 1 illustrates how the normalized version of the algorithms outperform the not-normalized ones. Furthermore, 3-gram and 2-gram usually achieve better MRR scores, but Dice's coefficient has high number of matches with the "Any" policy – it exhibits high recall.

5.2. Instance-Based Matching Evaluation

The performances of the three instance-based matching strategies are reported in Table 2. Notice that, the new similarity functions are based on real numbers much more widely spread. This dramatically reduces the chance of ties: thanks to this, we do not need anymore a tie-breaking policy. BM25 and cosine are statistically significantly equivalent, while Jaccard similarity is significantly worse than the others. It is interesting to observe that different matching strategies identify different correct matches. This suggests that a composite matching strategy might improve the overall quality of the approach.

5.3. Composite Matching Strategies

When considering the composite matching strategies, almost any combination of matchers and representations is a suitable candidate - therefore, in the following analysis we report the results for the best performing or most interesting composite matching techniques, selected also with the help of experts of the domain.

Among the composition strategies that we considered, the first approach consists in using simple matching strategies and combine them using traditional rank fusion techniques. Based

Table 1
Schema based matchers performances. † indicates statistical equivalence with the best performing approach.

		P@1 - first		P@1 - any		MRR	
input	**matcher**	**orig.**	**norm.**	**orig.**	**norm.**	**orig.**	**norm.**
F, A	CW	.177	.200	**.400**	.377	.221†	.259†
	ED	.111	.088	.133	.200	.166†	.153
	EDP	.177	—	.377	—	.245†	—
	NG$_2$	.111	.111	.022	.288	.181†	.215†
	NG$_3$	.133	.155	.333	.355	.199†	.248†
F, CA	CW	.177	.200	**.400**	.377	.221†	.259†
	ED	.022	.022	.088	.088	.059	.048
	EDP	.022	—	.022	—	.280†	—
	NG$_2$	.200	.244	.266	.311	.261†	.291†
	NG$_3$	.244	**.288**	.333	.377	.302†	**.336**
GF, A	CW	.133	.200	**.400**	.377	.199†	.259†
	ED	.022	.022	.088	.088	.047	.047
	EDP	.133	—	.266	—	.182†	—
	NG$_2$	.022	.044	.111	.155	.075	.120
	NG$_3$	.088	.111	.288	.311	.160†	.209†
GF, CA	CW	.133	.200	**.400**	.377	.199†	.259†
	ED	.088	.088	.133	.133	.138	.138
	EDP	.155	—	.155	—	.223†	—
	NG$_2$	.133	.222	.177	.222	.209†	.274†
	NG$_3$	.244	.244	.288	.288	.316†	.333†

Table 2
Instance based matchers performances. The symbol † indicates the top tier of systems.

matcher	**P@1**	**MRR**
cos	**.178**	**.275**†
BM25	.155	.257†
Jaccard	.022	.058

on Tables 1 and 2, we select the following matchers as fusion candidate: 2-gram and 3-gram normalized, CW normalized, EDP, BM25 and Cosine similarity. To represent fields and attributes in the schema-based approaches, for each matcher, we use the best performing representation according to Table 1. CombMNZ and CombSUM are then used to aggregate the similarity scores computed by the single matching strategies.

Furthermore, we experiment with the WA fusion strategy. In particular we use the following similarity function and test:

- NG$_2$(F,A): normalized 2-gram similarity between field and attribute names.
- NG$_2$(GF,C): normalized 2-gram similarity between field name and group name concate-

Table 3
Matcher combination performances.

Fusion	matchers	P@1	MRR
CombMNZ	BM25+cos+NG_2+EDP	.288	.379†
	BM25+cos	.200	.288
	BM25+NG_2	.222	.342†
	cos+NG_2	.266	.366
	EDP+NG_2	.244	.291
	cos+EDP	.244	.357†
	cos+CW	.177	.314
	NG_2+NG_3	.244	.304†
	BM25+cos+NG_2+NG_3+CW+EDP	.311	.414†
CombSUM	BM25+cos+NG_2+EDP	.288	.375†
	BM25+cos	.200	.288
	BM25+NG_2	.222	.338†
	cos+NG_2	.266	.365
	EDP+NG_2	.244	.291
	cos+EDP	.244	.357†
	cos+CW	.177	.314
	NG_2+NG_3	.244	.304†
	BM25+cos+NG_2+NG_3+CW+EDP	.311	.409†
WA	cos+BM25+NG_2(F,A)+NG_2(GF,C)+NG_3(F,CA)	**.467**	**.519**†

nation and attribute name.

- NG_3(F, CA): normalized 3-gram similarity between field name and attribute and class names concatenation.
- $\cos(f, a)$: cosine similarity between document representation for field f and article's attribute a.
- $\mathrm{BM25}(f, a)$: BM25 similarity between document representation for field f and article's attribute a.

Among all the different combinations we experiment with, the above-mentioned matchers have been selected due to their diversity. To find the best weight combination, we perform a grid search through a subset of parameters. Each weight can assume the values between 0 and 1 with a pace of 0.2, resulting in 16807 dispositions with repetition.

The resulting formula for the weighted average approach is:

$$\mathrm{WA}(f, a) = \frac{\alpha\,\mathrm{NG_2}(F,A)+\beta\,\mathrm{NG_2}(GF,C)+\gamma\,\mathrm{NG_3}(F,CA)+\delta\cos(f,a)+\zeta\,\mathrm{BM25}(f,a)}{\alpha+\beta+\gamma+\delta+\zeta} \quad (2)$$

The empirical analysis, carried out following a 5-fold cross-validation strategy, shows that the top performing set of parameters are: $\alpha = 0.2$, $\beta = 1$,$\gamma = 1$, $\delta = 0.6$, $\zeta = 0.8$[1]. Table 3 illustrates the results of the composite approach using the CombSUM, CombMNZ, and Weighted Average strategies. It is possible to observe that WA is the best performing strategy, achieving 46.7% correct matches and an MRR score of 51.9%. We observe a general improvement of performances with respect to single matching strategies – Tables 1 and 2). CombMNZ aggregation provides better results than CombSUM. Finally, we observe that, as a general trend, if we include more matching strategies, the quality of the aggregation improves.

6. Conclusion

This work stems from the need for exploiting historical production data to obtain a mediated schema of machines built by a company throughout several years. We interpreted the problem as a data integration task that required us to join multiple semi-structured BoMs in a unified database over the global schema. We, therefore, focused on lexical schema- and instance-based matching strategies. Among schema-based matching strategies, we considered matching strategies that exploit the lexical similarity of the element names. For instance-based matching, we used document similarity measures. Finally, we investigated the combination of the developed matches showed a performance increase regarding the single matching strategies. We showed that the number of correct matches increases if we combine multiple matching approaches. Results are satisfactory and considered promising by The Bedeschi Company. Our study highlighted challenges and issues emerging when approaching this task and indicated promising research paths. Future works will analyze the performances of our proposed solutions on different datasets as well as the use of additional similarity measures.

References

1. F. Azzalini, S. Jin, M. Renzi, L. Tanca, Blocking techniques for entity linkage: A semantics-based approach, Data Science and Engineering 6 (2021) 20–38.
2. A. Doan, A. Halevy, Z. Ives, Principles of data integration, Elsevier, 2012.
3. H.-H. Do, E. Rahm, Coma—a system for flexible combination of schema matching approaches, in: Proceedings of the VLDB, Elsevier, 2002, pp. 610–621.
4. R. Fagin, L. M. Haas, M. Hernández, R. J. Miller, L. Popa, Y. Velegrakis, Clio: Schema mapping creation and data exchange, in: Conceptual modeling: foundations and applications, Springer, 2009, pp. 198–236.
5. L. Seligman, P. Mork, A. Halevy, K. Smith, M. J. Carey, K. Chen, C. Wolf, J. Madhavan, A. Kannan, D. Burdick, Openii: an open source information integration toolkit, in: Proceedings of the 2010 ACM SIGMOD, 2010, pp. 1057–1060.
6. P. A. Bernstein, J. Madhavan, E. Rahm, Generic schema matching, ten years later, Proceedings of the VLDB Endowment 4 (2011) 695–701.
7. S. M. Falconer, N. F. Noy, Interactive techniques to support ontology matching, in: Schema Matching and Mapping, Springer, 2011, pp. 29–51.

[1]Due to space reasons, we omit the full analysis on the parametrization

8. D. Engmann, S. Massmann, Instance matching with coma++., in: BTW workshops, volume 7, 2007, pp. 28–37.
9. S. Massmann, S. Raunich, D. Aumüller, P. Arnold, E. Rahm, Evolution of the coma match system, Ontology Matching 49 (2011) 49–60.
10. J. Madhavan, P. A. Bernstein, E. Rahm, Generic schema matching with cupid, in: vldb, volume 1, Citeseer, 2001, pp. 49–58.
11. E. Rahm, Towards large-scale schema and ontology matching, in: Schema matching and mapping, Springer, 2011, pp. 3–27.
12. J. Madhavan, P. A. Bernstein, A. Doan, A. Halevy, Corpus-based schema matching, in: 21st International Conference on Data Engineering (ICDE'05), IEEE, 2005, pp. 57–68.
13. A. Bilke, F. Naumann, Schema matching using duplicates, in: 21st International Conference on Data Engineering (ICDE'05), IEEE, 2005, pp. 69–80.
14. Z. Bellahsene, A. Bonifati, F. Duchateau, Y. Velegrakis, On evaluating schema matching and mapping, in: Schema matching and mapping, Springer, 2011, pp. 253–291.
15. Z. Abedjan, L. Golab, F. Naumann, T. Papenbrock, Data profiling, Synthesis Lectures on Data Management 10 (2018) 1–154.
16. A. Maydanchik, Data quality assessment, Technics publications, 2007.
17. S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, J. Heer, Profiler: Integrated statistical analysis and visualization for data quality assessment, in: Proceedings of AVI, 2012.
18. Z. Abedjan, L. Golab, F. Naumann, Profiling relational data: a survey, The VLDB Journal 24 (2015) 557–581.
19. E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, the VLDB Journal 10 (2001) 334–350.
20. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, P. Domingos, imap: Discovering complex semantic matches between database schemas, in: Proceedings of the ACM SIGMOD, 2004, pp. 383–394.
21. P. Mork, L. Seligman, A. Rosenthal, J. Korb, C. Wolf, The harmony integration workbench, in: Journal on Data Semantics XI, Springer, 2008, pp. 65–93.
22. F. Giunchiglia, M. Yatskevich, Element level semantic matching (2004).
23. W. H. Gomaa, A. A. Fahmy, et al., A survey of text similarity approaches, International Journal of Computer Applications 68 (2013) 13–18.
24. G. Kondrak, N-gram similarity and distance, in: International symposium on string processing and information retrieval, Springer, 2005, pp. 115–126.
25. A. Huang, et al., Similarity measures for text document clustering, in: Proceedings of NZCSRSC, volume 4, 2008, pp. 9–56.
26. R. Ferdous, et al., An efficient k-means algorithm integrated with jaccard distance measure for document clustering, in: Proceedings of the AH-ICI, IEEE, 2009, pp. 1–6.
27. S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Now Publishers Inc, 2009.
28. J. H. Lee, Analyses of multiple evidence combination, in: Proc. SIGIR, 1997, pp. 267–276.
29. E. M. Voorhees, et al., The trec-8 question answering track report, in: Trec, 1999, pp. 77–82.