Big Data Integration for Industry 4.0[⋆]

Daniel Obraczka[0000−0002−0366−9872], Alieh Saeedi[0000−0002−1066−1959], Victor Christen[0000−0001−7175−7359], and Erhard Rahm[0000−0002−2665−1114]

Leipzig University, Germany
{obraczka,saeedi,christen,rahm}@informatik.uni-leipzig.de

Abstract. The fourth industrial revolution promises a new quality of automation with smart manufacturing devices sharing enormous amounts of data. A crucial step in fulfilling this promise is developing advanced data integration methods that are able to consolidate and combine heterogeneous data from multiple sources. We outline the use of knowledge graphs for data integration and provide an overview of proposed approaches to create and update such knowledge graphs, in particular for schema and ontology matching, data lifting and especially for entity resolution. Furthermore, we present data integration use cases for Industry 4.0 and discuss open problems.

Keywords: Industry 4.0 · Big Data · Data Integration · Knowledge Graph

1 Introduction and related work

The success of Industry 4.0 is based on the transforming technologies of the last decade: the Internet of Things and Big Data [37]. The Internet of Things enables communication and exchange of data between physical objects (e.g., sensors) to implement certain services and reach autonomous decisions. In the medical domain, for example, the Internet of Things can improve services such as monitoring, diagnostics, and treatment by utilizing interconnected devices that observe the vitality of persons [72]. The idea of Industry 4.0 is similarly based on the close interaction of decentralized systems such as production systems and products, to achieve self-controlled and self-optimizing processes. Big Data comes into play due to the enormous amount of different kinds of data that are continuously generated, exchanged, and to be processed. This data has to be standardized to enable their interpretation and autonomous decisions. Moreover, the different kinds of data can be collected, transformed, and integrated to support a holistic analysis and optimization of the different production processes, production lines, etc [22].

The challenges of Big Data are usually characterized by the "V" properties of *Volume*, *Velocity*, *Variety* and *Veracity*. These challenges are all relevant for

[⋆] This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig

Industry 4.0. In particular, disconnected sources in manufacturing processes generate a massive amount of data (Volume) at a high rate (Velocity) for further processing [22]. Variety refers to the need to process different kinds of heterogeneous data, in particular structured data (such as events or database records), semi-structured data (documents, log files, error reports), and unstructured data (e.g., images, audio files, and videos). Veracity finally asks for providing a high data quality to enable valid analysis results.

Data integration is the task to combine and enrich data from multiple sources for data analysis. *Big Data Integration* is data integration for Big Data that has to address the V challenges, in particular, Variety to deal with heterogeneous data of different kinds and Veracity to achieve high data quality. Additionally, the requirements Volume and Velocity lead to high-performance demand to deal with the massive amount of continuously produced data. The high data quality and performance requirements are best met with so-called physical data integration approaches that bring the data from different sources into a dedicated repository such as a data warehouse or knowledge graph. Such repositories can be maintained and used on a distributed cluster platform with many processors to achieve fast data processing and analysis. Furthermore, such approaches can apply comprehensive data preprocessing to improve data quality, in particular by extracting information from semi- and unstructured sources and for performing transformation and cleaning approaches for data consolidation [75,32]. Physical data integration such as the creation and continuous update of a data warehouse or a knowledge graph also entails several steps, including the task of entity resolution to identify (match) and fuse different representations of the same real-world entity such as for a product part or customer.

While there is a huge amount of previous research and commercial activities in the area of data integration [13,74], there is only little work focusing specifically on data integration for Industry 4.0. Some work has been done on the use of dedicated process knowledge repositories for workflow analysis [61], and for the enrichment and maintenance of unstructured documents such as failure and performance reports [52]. Most repositories focus on certain data types, applications or certain phases in the value chain of products. Process knowledge data consists of structured rules, information about data mining models and results as structured data. On the other hand, documents such as failure reports and unstructured data are essential as well. Groeger et al. [23] propose a repository for maintaining these types of data for each manufacturing step.

In the remainder of this chapter, we focus on (Big) data integration with knowledge graphs that can semantically integrate and interrelate many entities of different types for data analysis. Knowledge graphs are more flexible than data warehouses that are built on relational databases with a rather static, predefined schema that prevents the easy addition of new kinds of heterogeneous entities and their relationships. We begin by motivating the topic by outlining selected industrial use cases for data integration in Section 2. In Section 3, we introduce knowledge graphs and give an overview of the methods for constructing them. The important task of entity resolution is the topic of Section 4 that explains the main steps and how its performance can be improved to deal with Big Data. We close with a summary and outlook to open problems.

2 Data Integration Use Cases

Knowledge Graphs (KG) and other semantic technologies have become a viable option for companies to organize complex information in a meaningful manner. The semantic representation of data can improve understandability of complex data making development of new technologies more efficient [18], and improve quality control in manufacturing processes [97]. Not only software giants like Facebook, Google and Microsoft, but also production companies like Siemens [78] or news conglomerates like Thomas-Reuters [91] turn towards semantic representations of their data. Aibel, a service company in the energy sector, has reportedly saved more than 100 million Euros through better representation of their products using ontologies [90].

In the following we will look at some examples, where companies integrated heterogeneous data sources into semantic repositories.

In a Bosch factory[38] Surface Mount Technology is used to mount electrical components directly on circuit boards. Different machines are needed in this process, e.g., to place the electronic parts or inspect the solder joints. To detect failures in the manufacturing process, the integration of several data sources coming from machines of different vendors is necessary. This data integration relies on a domain ontology. An ontology is a semantic data structure, which contains known concepts and relationships and can be used to ensure the consistency in the data integration process. The machine components in the manufacturing pipeline produce log data in the form of JSON files. These are extracted and stored in a PostgreSQL database which is then manually mapped to the ontology. Through the use of the Ontop[1] framework a Virtual Knowledge Graph is created from the ontology and the mappings to the original data sources. The manufacturing process data can then be analyzed by sending SPARQL (a semantic querying language) queries which are translated to SQL queries to the original data sources. In an evaluation this approach returned results in tens of seconds, which the researchers deemed a reasonable amount of time for their use case. What is still missing is a more comprehensive data analysis that goes beyond the use of queries, e.g., the use of machine learning to identify erroneous processing steps.

Siemens relies on a similar approach to unify multiple data sources in their smart manufacturing process [78]. A common ontology is used and the heterogeneous sources are mapped to this ontology. The resulting KG is used as a basis to integrate dynamically occurring events in their factory into the KG. The researchers present an approach for event-enhanced KG completion using a machine learning approach to jointly learn KG embeddings as well as event sequence data embeddings. In their evaluation they show, that their approach

[1] `https://ontop-vkg.org`

leads to good quality KG completion and can aid in the synchronisation of the physical and digital representations of a smart factory.

Jirkovský et al. [35] investigate the use of semi-automatic ontology matching to integrate an Excel File containing Ford spare part records and the Ford supply chain ontology. They utilize extensive preprocessing to enrich the Excel records with implicit information contained in part numbers and abbreviations. Multiple similarity measures are used for element pairs which are fed into a self-organizing map, which is a type of artificial neural network that can be trained in an unsupervised fashion. The trained model can classify entity pairs and present the user with examples, where it is least confident about its classification.

3 Knowledge Graphs

In this section we first present the foundations of semantic technologies for knowledge graphs. We then present the necessary steps to semantically integrate heterogeneous data sources for creating and evolving such knowledge graphs.

3.1 Knowledge Graph Foundations

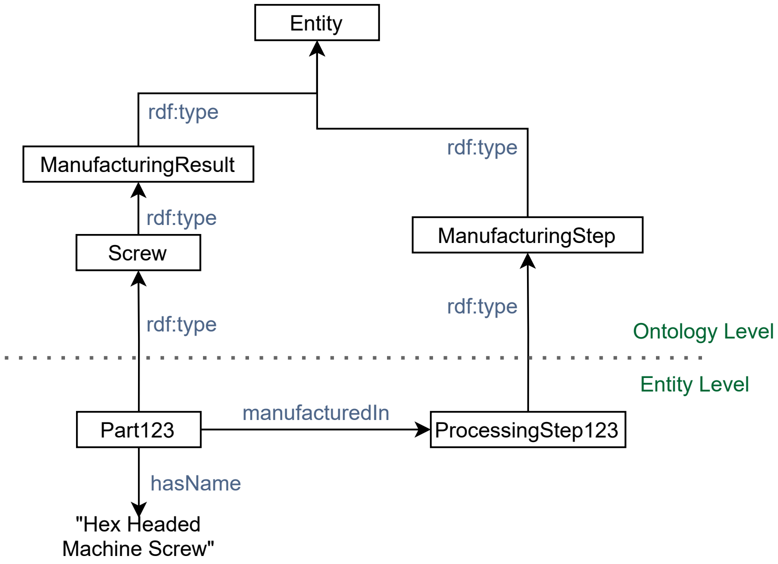

Fig. 1. Example snippet of a KG

In Fig. 1 we can see an example snippet of a KG. We will use this illustration to subsequently introduce RDF, ontologies and finally what a KG is.

RDF The standard that is used to create KGs with their entities and relationships is called RDF (Resource Description Framework), which is a recommendation[2] of the W3C (World Wide Web Consortium). An RDF graph is a set of triples. Using such triples we can make statements about entities and their relations. An example of a triple we can see in Fig. 1 is

```
Part123 manufacturedIn ProcessingStep123 .
```

An RDF Graph can have three different kinds of nodes: IRIs (Internationalized Resource identifiers), literals or blank nodes. IRIs are generalizations of URIs and give each resource a unique identifier. To express values such as strings, dates or numbers literals are used. RDF enables the user to also state the datatype and if the literal is a string a language tag can be provided. Blank nodes are anonymous resources, that enable more complex structures.

Ontologies An ontology is a formal description of knowledge using machine-processable specifications. These specifications have well defined meanings and contain known concepts and relationships [30]. For example, in Fig. 1 we express that the entity `Part123` belongs to the class `Screw` with the triple

```
Part123 rdf:type Screw .
```

Ontologies build on description logic, which enables reasoning engines to check logical consistency and correctness. Such reasoning possibilities are advantageous in the Industry 4.0 setting to make implicit information explicit. For example, [49] use reasoning as data enrichment step to infer compatibility of parts.

Furthermore, ontologies provide a so-called vocabulary, which is a set if IRIs, that can be used in RDF graphs[3]. In Fig. 1 for example we use the RDF vocabulary, by utilizing `rdf:type` to express that an entity is an instance of a class. Incorporating vocabularies is a common technique to rely on already existing ontologies and makes integration of different semantic systems easier. For the Industry 4.0 context there already exist ontologies like e.g. CORA (Core Ontology for Robotics and Automation) [71] that can be a useful starting point for companies. An overview over other ontologies for Industry 4.0 can be found here [86].

Knowledge Graph The terms ontology and KG are sometimes erroneously used as synonyms. KGs often integrate multiple sources into a single ontology and are able to derive new knowledge through reasoning [16]. While ontologies often focus on the conceptual modeling, knowledge graphs include a large number of entities and relations as instances of concepts and relationships, which introduces the need of instance-level data integration such as entity resolution. In the industry 4.0 context often the more specific term *industrial knowledge*

[2] `https://www.w3.org/TR/rdf-primer/`

[3] `https://www.w3.org/TR/rdf11-concepts/#vocabularies`

graph (e.g. at Siemens [31]) is used. A notable example of an open-source KG is the *Industry 4.0 Knowledge Graph* [3]. This KG contains information about standards used in smart manufacturing and relations between standards.

3.2 Knowledge Graph Construction

The construction of a KG entails the integration of (heterogeneous) data sources and enriching the data with semantic information. The integration process generally necessitates the following steps:

1. Creation of the KG ontology
2. Mapping of data sources to the KG ontology which requires *schema or ontology matching*
3. Preprocessing of data sources to extract, and clean entities and transform them into the RDF format which is also known as *data lifting*
4. Categorization of entities to assign them to the ontology concepts, e.g. for entities extracted from documents. This task can be addressed with machine learning by utilizing already assigned entities as training data [79]
5. *Entity Resolution* to identify duplicate entities and fuse them together in the knowledge graph.

Bear in mind, that some of these tasks can happen in different order (e.g., integrate the data sources first and then perform data lifting or vice versa) or even overlap (e.g., classification of entities can happen in the data lifting step). Moreover, the knowledge graph has to be continuously updated to incorporate new data and even new data sources. This asks for incremental methods to evolve the KG ontology and to add entities incrementally.

In the following we will start by presenting schema and ontology matching, followed by the data lifting task. Entity Resolution will be discussed separately in Section 4.

Schema and Ontology Matching Smart factories produce a plethora of different data formats from a vast number of sensors, databases, spreadsheets etc. To tackle this *variety* aspect of Big Data, companies have to unify these data collections under a common schema, a task that is referred to as schema matching. Schema matching aims to determine semantic correspondences between metadata, database schemata or in the special case of ontology matching between ontology elements. The high degree of semantic heterogeneity between sources makes this a difficult task, especially since not only one-to-one matches have to be found, but also more complex relationships like e.g., generalizations or part-of relations.

A central element of schema matching systems are matchers, which determine the similarity between concepts/attributes of the given schemata. Different types of matchers exist, namely instance- and metadata-based matchers. Instance-based matchers rely on already known instance matches between data sources and mostly rely on the instance overlap among concepts to determine how similar concepts are. Matchers that rely on metadata can further be divided into element-level and structure-level matchers, where the former use similarity between concept names sometimes utilizing dictionaries and the latter exploit structural information in ontologies e.g., the children or parents of concepts. Matching frameworks typically rely on a combination of different types of matchers to achieve a high quality result [24]. Matchers can be executed sequentially, in parallel or a mixture of both.

To illustrate this let us look at an example from the smart product lifecycle, where products from different vendors generate data, that we need to integrate [88]. Table 1 lists six sample products from five different provider sources such as *www.ebay.com* and *www.buzzilions.com.* The descriptions represent six cameras from two manufacturers *Canon* and *Nikon.* As shown, *entity 1* and *entity 2* as well as *entity 4*, *entity 5*, and *entity 6* represent the same real-world camera. We can see that schemata between data sources vary immensely. This is not only apparent by the different number of properties for the same entities, but also in the very different representation of the same attributes. For example, *entity 1* has an attribute *effective megapixel count* with a value `10.1`, as well as an attribute *pixel count* with the value `10 Megapixel`, while the matching *entity 2* has an attribute *megapixels* with the value `10.1 MP`. All three attributes would have to be determined to be the same. Data preprocessing can alleviate some heterogeneity e.g., replacing common abbreviations like `MP` for `Megapixel`. A schema matching approach will first have to classify entities from the given sources. In the example, the entities are all of the type camera, but the data sources might contain e.g., camera cases, which have to be separated from camera entities. Secondly, classification of properties helps to reduce the search space e.g., the property *compatible with macintosh* in *entity 1* should be treated as a Boolean variable rather than a string, and therefore not compared with other string attributes.

Reduction of search space is a general problem in schema matching. Given two schemata, the comparison of every element of one schema with every element of the other schema has quadratic complexity. This large search space has not only detrimental effects with regards to scalability but can also negatively impact match quality given the higher number of error possibilities. The main strategies to narrow the search space are early pruning of dissimilar elements and partitioning of the ontologies [73]. Early pruning means discarding element pairs with low similarity early in the matching process. Especially, in sequential matching workflows this enables early matchers to alleviate the burden of unnecessary comparisons for subsequent matchers. For example, after determining the attribute name similarity of *usb port* in *entity 1* and *brand* in *entity 2* is low, the comparison of these attributes can be omitted in further steps. Peukert et al. [70] employ filters to discard element pairs beneath a certain similarity threshold. The threshold can be predefined or dynamically set depending on already calculated comparisons and mapping results. Partitioning-based approaches divide the ontologies in smaller parts so that only partitions have to be compared.

Table 1. Example Raw Data

property	value
entity 1	
”source”	”www.buzzillions.com”
”page title”	”Canon EOS 40D Digital SLR Camera”
”compatible with macintosh”	”Yes”
”depth inches”	”2.9”
”digital slr”	[”Body Only”, ”Body With Lens”]
”effective megapixel count”	”10.1”
”height inches”	”4.2”
”lcd display size inches”	”3”
”lcd viewer”	”3 Inch”
”manufacturers warranty hardware”	”1 Year”
”megapixels”	”10.0”
”optical zoom”	”4x”
”pixel count”	”10 Megapixel”
”shutter speed”	”1/8000-30 second”
”skuprice”	”1299.9900”
”still image resolution max”	”3888 x 2592”
”usb port”	”(1) Mini-B”
”weight pounds”	”1.63”
”width inches”	”5.7”
entity 2	
”source”	”www.ebay.com”
”brand”	”Canon”
”megapixels”	”10.1 MP”
”model”	”40D”
”mpn”	”EOS 40D”
”screen size”	”3”̈
”type”	”Digital SLR”
entity 3	
”source”	”www.priceme.co.nz”
”page title”	”Canon EOS 400D New Zealand Prices - PriceMe”
”focus adjustment”	”Automatic focus, Manual focus”
”image stabilizer”	”Without Image Stabilizer”
”light sensitivity”	”ISO 100, ISO 1600, ISO 200, ISO 400, ISO 800, Auto”
”optical sensor”	”CMOS”
entity 4	
”source”	”www.gosale.com”
”page title”	”Nikon D3100 14.2MP Digital SLR on sale for $461.20”
”camera type”	”SLR”
”ean13”	”0018208097982”
”manufacturer”	”Nikon”
”megapixels”	”14.2 MP”
”product number mpn”	”D3100 18-55 5”
”retail price”	”$949.00”
”upc”	”018208097982”
entity 5	
”source”	”www.ebay.com”
”page title”	”Nikon D3100
”mpn”	”33858”
”screen size”	”3”̈
”upc”	”018208254866”
entity 6	
”source”	”www.walmart.com”
”page title”	”Nikon 14.2MP DSLR Camera with VR Lens, 3L̈CD”
”model no”	”Nikon D3100 Kit”
”shipping weight in pounds”	”3.6”
”walmart no”	”000609532”

This not only reduces the number of necessary comparisons, but makes these match tasks easily parallelizable.

Several different aspects of the data will have to be considered in order to create a high quality match result. A schema matching workflow will have to incorporate the similarity of attribute names and attribute values. The use of pre-trained word embeddings or synonym dictionaries can be beneficial to match attributes, that are dissimilar on character level, while being close semantically like `brand` and `manufacturer`. LeapME [2] relies on word embeddings and meta-information of property names and property values as input for a dense neural network. The classifier is trained on labeled property pairs and the corresponding feature vectors. The trained model can then be used to obtain matching decisions between unlabeled property pairs and their similarity scores. To integrate data about smart energy grids, Santodomingo and colleagues [87] use background knowledge from a database of electrical terminology. This background knowledge is used to find words with similar meanings to extend the strings of entities in the given ontologies. The authors utilize several matcher components, such as a linguistic module, which reduces words to their root form and filters out stop words, that are uninformative in the matching process (e.g., "the"), as well as threshold-based similarity components to derive matching decisions.

While binary matching approaches, unifying two sources, are most common, schema matching in the industry 4.0 context usually requires more holistic approaches that are able to consolidate multiple sources as shown in the example. Although it is possible to perform this task by sequentially matching two sources until all sources are integrated, specific approaches have been developed that cluster elements of multiple sources directly. Gruetze et al. [26] align large ontologies by clustering concepts by topic. Topical grouping is done by using Wikipedia pages related to concepts which result in category forests, that are a set of Wikipedia category trees. Utilizing the tree overlap alignments are generated. Megdiche et al.[54] model the holistic ontology matching task as maximum-weighted graph matching problem, which they solve within a linear program. Their approach is extensible with different linear constraints, that are used to reduce incoherence in resulting alignments. Roussille et al. [81] extend existing pairwise alignments of multiple sources by creating a graph with entities from ontologies as nodes, and correspondences as edges. They determine graph-cliques to detect the holistic alignment.

For a more general overview over ontology matching we refer the interested reader to this survey [63] and for a more detailed discussion of large-scale ontology and schema matching to [73].

Data Lifting The data in organizations usually has to be semantified, since it resides in formats which contain no machine-readable semantics such as relational databases or spreadsheets or even unstructured formats such as plain text. The necessary conversion process is called *data lifting*, since the data is not only transformed, but also "lifted" to a higher data level which contains semantic information [92].

While schema matching and data lifting both are concerned with mappings between different aspects of data sources, they have a different focus. Schema matching aims to consolidate heterogeneity between data sources and any enrichment of the data consists of implicit information that was scattered among different data sources. Data lifting seeks to transform data into RDF. While the mapping of e.g., a relational database to an existing ontology can be seen as a form of schema/ontology matching, data lifting is mainly concerned with transformation of the data into a different format.

The transformation process can be done manually by using specific mapping languages. The simplest is the *direct mapping*[4], which performs a quick conversion of a relational database to RDF. The relational database should have well-defined primary and foreign keys and meaningful table and column names. While being simple, the direct mapping approach has the drawback of not being able to reuse existing popular vocabularies. For a more sophisticated conversion the mapping language *R2RML*[5] can be used. It enables the user to have more control over the mapping process. The use of manually created mappings is frequently mentioned in the industry 4.0 context. The German industrial control and automation company Festo describes their struggles with their previous monolithic Java application for data transformation in this paper [49]. They have since moved to use custom R2RML mappings to transform relational data into entities of their KG. Similarly, Kotis and Katasonov [46] propose rule-based mappings in their semantic smart gateway for the Web of Things.

While mapping languages enable powerful transformations, they require domain experts to go through a laborious process of writing many mapping rules, even with tool support. To address this problem learning-based transformation approaches have been devised in a research field called *ontology learning.* In the following we will present some examples from the field. For a more thorough overview over the field of ontology learning we refer the reader to this recent survey [50].

Maedche and Staab [51] first conceptualize ontology learning to address the need of simplifying the ontology engineering process by enabling the semi-automatic integration of a wide range of sources including web documents, XML files as well as databases and existing ontologies. They rely on dictionaries to extract concepts and use hierarchical clustering to build a taxonomical structure in their ontology. Using association rule mining with a class hierarchy as background they derive possible relationships that are presented to the user. Modoni et. al [56] present a rule-based approach to automatically transform relational databases to ontologies. Their ontology integration approach uses the mediator pattern, which does not physically integrate the ontologies but rather provides a common interface to distributed data sources. The mediation is done through custom mapping rules. The authors illustrate their approach with a case study of a mould production company, which is faced with integrating their various data sources.

[4] `http://www.w3.org/TR/rdb-direct-mapping/`

[5] `https://www.w3.org/TR/r2rml/`

4 Entity Resolution

In the smart product lifecycle and Industry 4.0. in general, a deluge of data from numerous sources is generated [88] requiring Big Data techniques for the collection, integration and analysis of heterogeneous data. Entity Resolution (ER) or data matching is a main step for data integration and the creation/evolution of knowledge graphs. It is the task of identifying entities within or across sources that refer to the same real-world entity. ER for Industry 4.0 requires fast and scalable solutions (Volume) as well as advanced methods to incrementally add new data or even new data sources either in a real-time or evolutionary way (Velocity) [21].

ER is typically implemented by a multistep workflow, as shown in Fig. 2. The input is data from multiple sources that may differ enormously in size and quality, and the output is a set of clusters, each of which contains all matching entities referring to the same real-world entity. The shown preprocessing step has already been discussed and entails data cleaning actions such as handling missing values, smoothing noisy values, and identifying and correcting inconsistent values [9]. Furthermore, schema matching can be applied to identify matching properties that can be used for determining the similarity of entities for ER. To match the cameras shown in Table 1, preprocessing may include transforming values into the same unit, lower casing strings, applying canonical abbreviations to harmonize property values, and assigning the same name to matching properties to facilitate similarity computations.

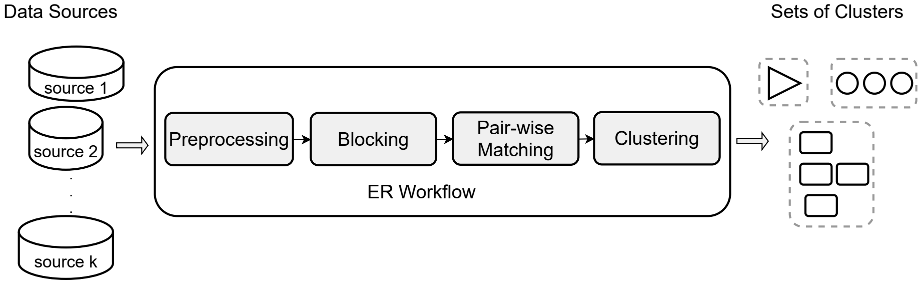

Fig. 2. Entity Resolution Workflow

The blocking step prevents comparing irrelevant entities with each other. For instance, in our running camera example (Table 1), cameras with different manufacturers will be placed in different blocks in order to avoid comparing *Nikon* cameras with *Canon* cameras. Then in the pair-wise matching step, the similarity of candidate pairs are computed by applying a set of similarity methods on the property values of the entities. Finally, the clustering step uses computed similarities to group the same entities in the same cluster. Clustering facilitates fusion of the same entities into one unique representative entity.

The main ER steps of blocking, matching and clustering will be discussed in the following subsections, emphasizing on techniques related to Big Data. We will also outline incremental ER solutions to deal with the incremental addition of new entities and even new sources to a knowledge graph. Finally, we briefly discuss some ER prototypes for Big Data.

4.1 Blocking

Blocking aims at improving performance and scalability by avoiding that every entity has to be compared with every other entity for determining matching entity pairs, leading to a quadratic complexity. Therefore, blocking methods intend to restrict the comparisons only to those pairs that are likely to match. Standard Blocking (SB) [19] and Sorted Neighborhood (SN) [28] are two popular blocking methods that both utilize a so-called *blocking key* to group entities. The key is mostly specified by an expert and is the result of a function on one or several property values, e.g. the initial five letters of the manufacturer name or page title property for the camera example (Table 1). Since real data is noisy, generating one blocking key per entity may not allow finding all matches. Hence, it can be necessary to generate multiple blocking keys per entity, leading to multi-pass blocking [29,44] that can find more matches and thus improve recall over the use of single blocking key. Since determining suitable blocking keys can be a tedious and difficult task, approaches based on both supervised [6,20] and unsupervised [39] Machine Learning (ML) have been proposed to learn blocking keys. [67] gives a comprehensive overview of blocking techniques.

To further improve runtime and scalability, the blocking methods can be parallelized to utilize multiple machines in a cluster. This is relatively easy to achieve on partitioned input data by utilizing the MapReduce [12] framework or newer frameworks such as Apache Spark [95] that build on MapReduce. Moreover, since the sizes of the output blocks can be skewed, achieving good load balancing is the major challenge for parallel blocking and ER. Kolb et al. propose the load-balanced SB [43] and SN [42] based on the MapReduce framework.

For semi-structured, textual data or in absence of an aligned schema across sources, schema-agnostic token-based blocking approaches have been proposed. The basic Token Blocking (TB) [65] generates a candidate match based on the common tokens of property values of a pair. Like with traditional blocking methods, scalability can be improved by a MapReduce-based implementation [67] and ensuring load balancing [11]. Since the basic TB may create too many candidate pairs, newer schema-agnostic approaches reduce them by pairing tokens from synthetically similar properties, considering only selected properties, or comparing only the entities of the same type [67]. Furthermore, block post-processing approaches such as meta-blocking [89,66] can largely reduce the number of candidate matches. A very different approach is [62] that totally ignores property values but determines candidate matches based on relations between entities.

4.2 Pair-wise Matching

The decision on whether a pair of entities is a likely match is based on the similarity of the two entities, which is determined by one or multiple similarity functions. These functions mostly determine the similarity of property values depending on the data type (string, numerical, date, geographical coordinates etc.). Typically, several such similarity values need to be combined to derive a match or non-match decision. Traditional approaches such as threshold-based or rule-based methods classify the matching status for each pair independently. In threshold-based classification, a specified threshold considers all pairs with similarity above a certain value as matches. On the other hand, in rule-based classification, a rule specifies a match predicate consisting of property-specific similarity conditions that are combined with logical operations [9]. For the camera example (Table 1), the match decision may be based on the similarity of the properties ”page title” and ”megapixels” although the latter property is not present for all entities shown.

Another line of research called collective ER [5] uses both property value similarity and relational information for determining the similarity of two entities. Here, the ER process is mostly iterative because changes in similarity or matching status of one pair affects the similarity value of the neighbouring pairs. Such approaches are more difficult to scale than with the standard approaches, where candidate pairs are compared independently. To better scale collective ER, Rastogi et al. [77] propose a generic approach that executes multiple instances of the matching task and constructs the global solution by message passing.

Manually determining the properties to match, similarity functions and similarity thresholds is a complex task, especially for heterogeneous and noisy data. Hence, a better alternative is often to apply supervised ML approaches to find optimal match configurations to determine matching entity pairs. These approaches can utilize traditional ML techniques such as SVM, logistic regression or random forests [40] but also newer approaches based on deep learning. Barlaug et al. [4] provides an overview about ER proposals utilizing deep neural networks including the approaches DeepER [14], DeepMatcher [57] and Hi-EM [96]. These approaches typically utilize embeddings for textual property values by transforming either words or their characters to numerical representations that preserve the semantic similarity between property values. Word embeddings are able to convert a long sequence to a short one, but they can not necessarily cover all possible words for specialized domains. The generation of embeddings can make use of pretrained models such as word2vec [55], GloVe [69] or fastText [7] that are derived from large corpora such as Wikipedia [4].

4.3 Clustering

The matches determined by the pair-wise similarity calculations are often contradicting and therefore only match candidates. The final matches are determined by applying a clustering approach on the set of candidate match pairs that form a similarity graph where matching entities are linked with each other. The baseline approach for entity clustering is to determine the transitive closure or connected components over the match links. Note, that general clustering algorithms like K-means that need a predefined number of clusters are not suitable for ER.

Connected components algorithm does not consider the strength or similarity of candidate matches, and can thus cluster even weakly similar entities. There is a large spectrum of alternatives some of which, e.g. Stable Marriage [53] and Hungarian algorithm [47] are suited when the input consists of two duplicate-free sources. For deduplicating a single source, Hassanzadeh et al. [27] comparatively analyzed several clustering algorithms. For some of them, such as Correlation Clustering, parallel implementations based on iterative processing and message passing have been proposed [64,8]. Saeedi et al.[83] comparatively evaluate the effectiveness and scalability of parallel implementations of several clustering schemes from [27] for the case of multiple data sources. Recently, Yan et al. [94] proposed a novel hierarchical clustering approach that avoids so-called hard conflicts inside clusters where the weakest similarity in a cluster is below a critical threshold. This is achieved by not merging candidate cluster pairs if this would lead to such a hard conflict. The approach is used within an industrial ER framework that is applied on billions of customer records on a daily basis.

Another line of research focuses on designing methods and algorithms for clustering entities from multiple duplicate-free sources [84,59] or clustering entities from combined duplicate-free and dirty (duplicate-containing) data sources [48]. The proposed approaches outperform more general approaches such as correlation clustering.

4.4 Incremental ER

Incremental ER approaches are needed to address the "Velocity" characteristic of Big Data to deal with dynamic or evolving data such as new incoming entities or even new data sources. Incremental approaches generally fall into two categories: 1) real-time approaches that are mostly applied in query processing and deal with individual new entities and 2) evolutionary approaches that deal with the addition of several entities of even a complete new data source in order to update an already existing knowledge graph without repeating the ER process for all data.

1) Real-time approaches leverage dynamic blocking and indexing techniques [76] as well as dynamic pair-wise matching methods [93,33,1] that support the fast matching of entities at query time.

2) Evolutionary approaches focus on updating the knowledge graph. Gruenheid et al. propose a generic greedy approach for such an incremental ER and clustering [25]. This method is extended in [58] to avoid computations on already integrated portions of the data that are unlikely to be affected by the new data. Scalable approaches for incremental entity clustering that also support the addition of new data sources are investigated in [60,85]. In particular, [60] proposes an incremental entity clustering based on a `Max-Both` strategy that adds a new entity to the maximally similar cluster only if there is no other new entity of the same input source with a higher similarity. [85] proposes a method called n-depth reclustering for incremental linking and clustering that is even able to repair existing clusters for improved quality and a reduced dependency on the insert order of new entities.

4.5 ER Prototypes

There are several ER prototypes suitable for Big Data that are surveyed in [10] including Dedoop [41], Magellan [45], FAMER [82], Silk [34], MinoanER [15], and JedAI [68]. Each of them implements the whole ER pipeline in a parallel way and includes novel Big-Data-specific approaches for at least one step of the pipeline. Dedoop is one of the early systems and based on MapReduce [12]; it implements the load balancing techniques discussed in the subsection on blocking. Silk, MinoanER, JedAI and a non-public version of Magellan are implemented on top of Apache Spark[6] while FAMER uses Apache Flink[7]. FAMER additionally supports the incremental addition of new entities and new data sources [85] and can deal with entities from multiple sources (>2), while MinoanER supports schema-agnostic ER methods to deal with heterogeneous and noisy web entities.

5 Conclusion & Open Problems

We presented an overview over the Big Data challenges for data integration posed by the fourth industrial revolution. We advocated the use of knowledge graphs for the integrated and semantically consolidated representation of heterogeneous data as a basis for data analysis and production optimization. Creating and continuously updating knowledge graphs is challenging and we presented approaches for the tasks of schema/ontology matching, data lifting/semantification and especially for entity resolution. We also discussed some published data integration use cases for Industry 4.0.

The current state for Big Data integration using knowledge graphs in Industry 4.0 is still in an early stage and requires too much manual effort. The common use of manual mapping rules for data lifting and/or schema matching can be justifiable for horizontal integration cases with already well structured high quality data. However, more efforts are needed to bridge the gap between the (semi-)automatic data integration tools developed in academia and manual matching efforts that are prevalent in the industry to establish robust methods for integrating the complex data of industrial applications. Especially the increasing interconnection of different domains (e.g. IoT, Smart Factories and Smart Grids) calls for more automated integration concepts, that could enable "plug & play" capabilities of smart machinery [17]. Solving these challenges is not reasonably possible without incremental ER solutions that keep knowledge graphs in sync with the physical realities present in smart factories, within a reasonable time frame. The possibility of integrating increasingly larger data sources

[6] https://spark.apache.org/

[7] https://flink.apache.org/

asks for scalable solutions. The triple stores used in Semantic Web applications can become a bottleneck, which necessitates alternative solutions [36]. The use of frameworks that rely on property graph models (e.g. Neo4j[8] or Gradoop [80]) can be a viable alternative to triple stores in some use cases.

The interdiscplinary nature of Industry 4.0 necessitates a close cooperation between domain experts of the respective manufacturing domain, ontology engineers and data scientists [38]. We believe, that this is not only true for individual projects in this domain, but for the research in this direction as a whole.

References

1. Altwaijry, H., Kalashnikov, D.V., Mehrotra, S.: Query-driven approach to entity resolution. Proceedings of the VLDB Endowment **6**(14), 1846–1857 (2013)
2. Ayala, D., Hernández, I., Ruiz, D., Rahm, E.: Leapme: Learning-based property matching with embeddings (2020)
3. Bader, S.R., Grangel-González, I., Nanjappa, P., Vidal, M.E., Maleshkova, M.: A knowledge graph for industry 4.0. The Semantic Web **12123**, 465 – 480 (2020)
4. Barlaug, N., Gulla, J.A.: Neural networks for entity matching: A survey. arXiv preprint arXiv:2010.11075 (2020)
5. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data (TKDD) **1**(1), 5–es (2007)
6. Bilenko, M., Kamath, B., Mooney, R.J.: Adaptive blocking: Learning to scale up record linkage. In: Sixth International Conference on Data Mining (ICDM'06). pp. 87–96. IEEE (2006)
7. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
8. Chierichetti, F., Dalvi, N., Kumar, R.: Correlation clustering in mapreduce. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 641–650 (2014)
9. Christen, P.: The data matching process. In: Data Matching, pp. 23–35. Springer (2012)
10. Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., Stefanidis, K.: An overview of end-to-end entity resolution for big data. ACM Computing Surveys (2020)
11. Chu, X., Ilyas, I.F., Koutris, P.: Distributed data deduplication. Proceedings of the VLDB Endowment **9**(11), 864–875 (2016)
12. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
13. Dong, X.L., Srivastava, D.: Big data integration. Synthesis Lectures on Data Management **7**(1), 1–198 (2015)
14. Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., Tang, N.: Distributed representations of tuples for entity resolution pp. 1454–1467 (2018)
15. Efthymiou, V., Papadakis, G., Stefanidis, K., Christophides, V.: Minoaner: Schema-agnostic, non-iterative, massively parallel resolution of web entities. arXiv preprint arXiv:1905.06170 (2019)

[8] `https://neo4j.com/`

16. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. In: SEMANTiCS (Posters, Demos, SuCCESS) (2016)
17. Ekaputra, F.J., Sabou, M., Biffl, S., Einfalt, A., Krammer, L., Kastner, W., Ekaputra, F.J.: Semantics for Cyber-Physical Systems: A cross-domain perspective. Semantic Web **11**(1), 115–124 (2020). https://doi.org/10.3233/SW-190381, `https://doi.org/10.3233/SW-190381`
18. Elmer, S., Jrad, F., Liebig, T., Ul Mehdi, A., Opitz, M., Stauß, T., Weidig, D.: Ontologies and reasoning to capture product complexity in automation industry. CEUR Workshop Proceedings **1963**, 1–2 (2017)
19. Fellegi, I.P., Sunter, A.B.: A theory for record linkage. Journal of the American Statistical Association **64**(328), 1183–1210 (1969)
20. Giang, P.H.: A machine learning approach to create blocking criteria for record linkage. Health care management science **18**(1), 93–105 (2015)
21. Gölzer, P., Cato, P., Amberg, M.: Data processing requirements of industry 4.0 - use cases for big data applications. In: Becker, J., vom Brocke, J., de Marco, M. (eds.) 23rd European Conference on Information Systems, ECIS 2015, Münster, Germany, May 26-29, 2015 (2015), `http://aisel.aisnet.org/ecis2015_rip/61`
22. Gröger, C.: Building an industry 4.0 analytics platform - practical challenges, approaches and future research directions. Datenbank-Spektrum **18**(1), 5–14 (2018). https://doi.org/10.1007/s13222-018-0273-1, `https://doi.org/10.1007/s13222-018-0273-1`
23. Gröger, C., Schwarz, H., Mitschang, B.: The manufacturing knowledge repository - consolidating knowledge to enable holistic process knowledge management in manufacturing. In: Hammoudi, S., Maciaszek, L.A., Cordeiro, J. (eds.) ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, Volume 1, Lisbon, Portugal, 27-30 April, 2014. pp. 39–51. SciTePress (2014). https://doi.org/10.5220/0004891200390051, `https://doi.org/10.5220/0004891200390051`
24. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: On matching large life science ontologies in parallel. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6254 LNBI**, 35–49 (2010). https://doi.org/10.1007/978-3-642-15120-0_4
25. Gruenheid, A., Dong, X.L., Srivastava, D.: Incremental record linkage. Proceedings of the VLDB Endowment **7**(9), 697–708 (2014)
26. Gruetze, T., Böhm, C., Naumann, F.: Holistic and scalable ontology alignment for linked open data. CEUR Workshop Proceedings **937** (2012)
27. Hassanzadeh, O., Chiang, F., Lee, H.C., Miller, R.J.: Framework for evaluating clustering algorithms in duplicate detection. Proceedings of the VLDB Endowment **2**(1), 1282–1293 (2009)
28. Hernández, M.A., Stolfo, S.J.: The merge/purge problem for large databases. ACM Sigmod Record **24**(2), 127–138 (1995)
29. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. Data mining and knowledge discovery **2**(1), 9–37 (1998)
30. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
31. Hubauer, T., Lamparter, S., Haase, P., Herzig, D.: Use cases of the industrial knowledge graph at siemens. CEUR Workshop Proceedings **2180** (2018)
32. Ilyas, I.F., Chu, X.: Data cleaning. Morgan & Claypool (2019)
33. Ioannou, E., Nejdl, W., Niederée, C., Velegrakis, Y.: On-the-fly entity-aware query processing in the presence of linkage. Proceedings of the VLDB Endowment **3**(1-2), 429–438 (2010)

34. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. arXiv preprint arXiv:1208.0291 (2012)
35. Jirkovský, V., Kadera, P., Rychtyckyj, N.: Semi-automatic ontology matching approach for integration of various data models in automotive. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10444 LNAI**(August), 53–65 (2017). https://doi.org/10.1007/978-3-319-64635-0_5
36. Jirkovsky, V., Obitko, M., Marik, V.: Understanding data heterogeneity in the context of cyber-physical systems integration. IEEE Transactions on Industrial Informatics **13**(2) (2017). https://doi.org/10.1109/TII.2016.2596101
37. Kagermann, H., Wahlster, W., Helbig, J.: Recommendations for implementing the strategic initiative industrie 4.0 – securing the future of german manufacturing industry. Final report of the industrie 4.0 working group, acatech – National Academy of Science and Engineering, München (2013), https://en.acatech.de/wp-content/uploads/sites/6/2018/03/Final_report__Industrie_4.0_accessible.pdf
38. Kalaycı, E.G., Grangel González, I., Lösch, F., Xiao, G., Ul-Mehdi, A., Kharlamov, E., Calvanese, D.: Semantic Integration of Bosch Manufacturing Data Using Virtual Knowledge Graphs, vol. 12507 LNCS. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-62466-8_29, http://dx.doi.org/10.1007/978-3-030-62466-8{_}29
39. Kejriwal, M., Miranker, D.P.: An unsupervised algorithm for learning blocking schemes. In: 2013 IEEE 13th International Conference on Data Mining. pp. 340–349. IEEE (2013)
40. Koepcke, H., Thor, A., Rahm, E.: Learning-based approaches for matching web data entities. IEEE Internet Computing **14**(4), 23–31 (2010)
41. Kolb, L., Rahm, E.: Parallel entity resolution with dedoop. Datenbank-Spektrum **13**(1), 23–32 (2013)
42. Kolb, L., Thor, A., Rahm, E.: Parallel sorted neighborhood blocking with mapreduce. arXiv preprint arXiv:1010.3053 (2010)
43. Kolb, L., Thor, A., Rahm, E.: Load balancing for mapreduce-based entity resolution. In: 2012 IEEE 28th international conference on data engineering. pp. 618–629. IEEE (2012)
44. Kolb, L., Thor, A., Rahm, E.: Multi-pass sorted neighborhood blocking with mapreduce. Computer Science-Research and Development **27**(1), 45–63 (2012)
45. Konda, P., Das, S., Suganthan GC, P., Doan, A., Ardalan, A., Ballard, J.R., Li, H., Panahi, F., Zhang, H., Naughton, J., et al.: Magellan: Toward building entity matching management systems. Proceedings of the VLDB Endowment **9**(12), 1197–1208 (2016)
46. Kotis, K., Katasonov, A.: Semantic interoperability on the web of things: The semantic smart gateway framework. In: Barolli, L., Xhafa, F., Vitabile, S., Uehara, M. (eds.) Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2012, Palermo, Italy, July 4-6, 2012. pp. 630–635. IEEE Computer Society (2012). https://doi.org/10.1109/CISIS.2012.200, https://doi.org/10.1109/CISIS.2012.200
47. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
48. Lerm, S., Saeedi, A., Rahm, E.: Extended affinity propagation clustering for multi-source entity resolution. Datenbank-Spektrum (2021)

49. Liebig, T., Maisenbacher, A., Opitz, M., Seyler, J.R., Sudra, G., Wissmann, J.: Building a knowledge graph for products and solutions in the automation industry. CEUR Workshop Proceedings **2489**, 13–23 (2019)
50. Ma, C., Molnár, B.: Use of Ontology Learning in Information System Integration: A Literature Survey. Communications in Computer and Information Science **1178 CCIS**, 342–353 (2020). https://doi.org/10.1007/978-981-15-3380-8_30
51. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**(2), 72–79 (2001). https://doi.org/10.1109/5254.920602, `https://doi.org/10.1109/5254.920602`
52. Mazumdar, S., Varga, A., Lanfranchi, V., Petrelli, D., Ciravegna, F.: A knowledge dashboard for manufacturing industries. In: Garcia-Castro, R., Fensel, D., Antoniou, G. (eds.) The Semantic Web: ESWC 2011 Workshops - ESWC 2011 Workshops, Heraklion, Greece, May 29-30, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7117, pp. 112–124. Springer (2011). https://doi.org/10.1007/978-3-642-25953-1_10, `https://doi.org/10.1007/978-3-642-25953-1_10`
53. McVitie, D.G., Wilson, L.B.: Stable marriage assignment for unequal sets. BIT Numerical Mathematics **10**(3), 295–309 (1970)
54. Megdiche, I., Teste, O., dos Santos, C.T.: An extensible linear approach for holistic ontology matching. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9981, pp. 393–410 (2016). https://doi.org/10.1007/978-3-319-46523-4_24, `https://doi.org/10.1007/978-3-319-46523-4_24`
55. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26**, 3111–3119 (2013)
56. Modoni, G.E., Doukas, M., Terkaj, W., Sacco, M., Mourtzis, D.: Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models. International Journal of Computer Integrated Manufacturing **30**(10), 1043–1059 (2017). https://doi.org/10.1080/0951192X.2016.1268720, `http://dx.doi.org/10.1080/0951192X.2016.1268720`
57. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V.: Deep learning for entity matching: A design space exploration. In: Proceedings of the 2018 International Conference on Management of Data. pp. 19–34 (2018)
58. do Nascimento, D.C., Pires, C.E.S., Mestre, D.G.: Heuristic-based approaches for speeding up incremental record linkage. Journal of Systems and Software **137**, 335–354 (2018)
59. Nentwig, M., Groß, A., Rahm, E.: Holistic entity clustering for linked data. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 194–201. IEEE (2016)
60. Nentwig, M., Rahm, E.: Incremental clustering on linked data. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 531–538. IEEE (2018)
61. Niedermann, F., Schwarz, H., Mitschang, B.: Managing insights: A repository for process analytics, optimization and decision support. In: Filipe, J., Liu, K. (eds.)

KMIS 2011 - Proceedings of the International Conference on Knowledge Management and Information Sharing, Paris, France, 26-29 October, 2011. pp. 424–429. SciTePress (2011)

62. Nin, J., Muntés-Mulero, V., Martinez-Bazan, N., Larriba-Pey, J.L.: On the use of semantic blocking techniques for data cleansing and integration. In: 11th International Database Engineering and Applications Symposium (IDEAS 2007). pp. 190–198. IEEE (2007)
63. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Systems with Applications **42**(2) (2015). https://doi.org/10.1016/j.eswa.2014.08.032
64. Pan, X., Papailiopoulos, D., Oymak, S., Recht, B., Ramchandran, K., Jordan, M.I.: Parallel correlation clustering on big graphs. In: Advances in Neural Information Processing Systems. pp. 82–90 (2015)
65. Papadakis, G., Ioannou, E., Palpanas, T., Niederee, C., Nejdl, W.: A blocking framework for entity resolution in highly heterogeneous information spaces. IEEE Transactions on Knowledge and Data Engineering **25**(12), 2665–2682 (2012)
66. Papadakis, G., Papastefanatos, G., Palpanas, T., Koubarakis, M.: Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking. In: EDBT. pp. 221–232 (2016)
67. Papadakis, G., Skoutas, D., Thanos, E., Palpanas, T.: A survey of blocking and filtering techniques for entity resolution. CoRR, abs/1905.06167 (2019)
68. Papadakis, G., Tsekouras, L., Thanos, E., Pittaras, N., Simonini, G., Skoutas, D., Isaris, P., Giannakopoulos, G., Palpanas, T., Koubarakis, M.: Jedai3: beyond batch, blocking-based entity resolution. In: EDBT. pp. 603–606 (2020)
69. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
70. Peukert, E., Berthold, H., Rahm, E.: Rewrite techniques for performance optimization of schema matching processes. Advances in Database Technology - EDBT 2010 - 13th International Conference on Extending Database Technology, Proceedings pp. 453–464 (2010). https://doi.org/10.1145/1739041.1739096
71. Prestes, E., Carbonera, J.L., Fiorini, S.R., Jorge, V.A.M., Abel, M., Madhavan, R., Locoro, A., Gonçalves, P.J.S., Barreto, M.E., Habib, M.K., Chibani, A., Gérard, S., Amirat, Y., Schlenoff, C.: Towards a core ontology for robotics and automation. Robotics Auton. Syst. **61**(11), 1193–1204 (2013). https://doi.org/10.1016/j.robot.2013.04.005, `https://doi.org/10.1016/j.robot.2013.04.005`
72. Qadri, Y.A., Nauman, A., Zikria, Y.B., Vasilakos, A.V., Kim, S.W.: The future of healthcare internet of things: A survey of emerging technologies. IEEE Commun. Surv. Tutorials **22**(2), 1121–1167 (2020). https://doi.org/10.1109/COMST.2020.2973314, `https://doi.org/10.1109/COMST.2020.2973314`
73. Rahm, E.: Towards Large-Scale Schema and Ontology Matching. Schema Matching and Mapping pp. 3–27 (2011). https://doi.org/10.1007/978-3-642-16518-4_1
74. Rahm, E.: The case for holistic data integration. In: Proc. ADBIS. pp. 11–27. Springer (2016)
75. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Eng. Bull. **23**(4), 3–13 (2000)
76. Ramadan, B., Christen, P., Liang, H., Gayler, R.W.: Dynamic sorted neighborhood indexing for real-time entity resolution. Journal of Data and Information Quality (JDIQ) **6**(4), 1–29 (2015)

77. Rastogi, V., Dalvi, N., Garofalakis, M.: Large-scale collective entity matching. arXiv preprint arXiv:1103.2410 (2011)
78. Ringsquandl, M., Kharlamov, E., Stepanova, D., Lamparter, S., Lepratti, R., Horrocks, I., Kroger, P.: On event-driven knowledge graph completion in digital factories. Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017 **2018-Janua**, 1676–1681 (2017). https://doi.org/10.1109/BigData.2017.8258105
79. Ristoski, P., Petrovski, P., Mika, P., Paulheim, H.: A machine learning approach for product matching and categorization. Semantic Web **9**(5), 707–728 (2018)
80. Rost, C., Thor, A., Fritzsche, P., Gómez, K., Rahm, E.: Evolution analysis of large graphs with gradoop. In: Cellier, P., Driessens, K. (eds.) Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I. Communications in Computer and Information Science, vol. 1167, pp. 402–408. Springer (2019). https://doi.org/10.1007/978-3-030-43823-4_33, https://doi.org/10.1007/978-3-030-43823-4_33
81. Roussille, P., Megdiche, I., Teste, O., Trojahn, C.: Boosting holistic ontology matching: Generating graph clique-based relaxed reference alignments for holistic evaluation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11313**(November), 355–369 (2018). https://doi.org/10.1007/978-3-030-03667-6_23
82. Saeedi, A., Nentwig, M., Peukert, E., Rahm, E.: Scalable matching and clustering of entities with famer. Complex Systems Informatics and Modeling Quarterly **16**, 61–83 (2018)
83. Saeedi, A., Peukert, E., Rahm, E.: Comparative evaluation of distributed clustering schemes for multi-source entity resolution. In: European Conference on Advances in Databases and Information Systems. pp. 278–293. Springer (2017)
84. Saeedi, A., Peukert, E., Rahm, E.: Using link features for entity clustering in knowledge graphs. In: European Semantic Web Conference. pp. 576–592. Springer (2018)
85. Saeedi, A., Peukert, E., Rahm, E.: Incremental multi-source entity resolution for knowledge graph completion. In: European Semantic Web Conference. pp. 393–408. Springer (2020)
86. Sampath Kumar, V.R., Khamis, A., Fiorini, S., Carbonera, J.L., Alarcos, A.O., Habib, M., Goncalves, P., Howard, L.I., Olszewska, J.I.: Ontologies for industry 4.0. Knowledge Engineering Review **34** (2019). https://doi.org/10.1017/S0269888919000109
87. Santodomingo, R., Rohjans, S., Uslar, M., Rodríguez-Mondéjar, J.A., Sanz-Bobi, M.A.: Ontology matching system for future energy smart grids. Engineering Applications of Artificial Intelligence **32** (2014). https://doi.org/10.1016/j.engappai.2014.02.005
88. Schmidt, M., Galende, M., Saludes, S., Sarris, N., Rodriguez, J., Unal, P., Stojanovic, N., Vidal, I.G.M., Corchero, A., Berre, A., Cattaneo, G., Geogoulias, K., Stojanovic, L., Decubber, C.: Big data challenges in smart manufacturing: A discussion paper on big data challenges for bdva and effra research & innovation roadmaps alignment. Tech. rep., Big Data Value Association (2018), https://bdva.eu/sites/default/files/BDVA_SMI_Discussion_Paper_Web_Version.pdf
89. Simonini, G., Bergamaschi, S., Jagadish, H.: Blast: a loosely schema-aware metablocking approach for entity resolution. pvldb 9, 12 (2016), 1173–1184 (2016)
90. Skjæveland, M.G., Gjerver, A., Hansen, C.M., Klüwer, J.W., Strand, M.R., Waaler, A., Øverli, P.Ø.: Semantic material master data management at AibEL. CEUR Workshop Proceedings **2180**, 4–5 (2018)

91. Song, D., Schilder, F., Hertz, S., Saltini, G., Smiley, C., Nivarthi, P., Hazai, O., Landau, D., Zaharkin, M., Zielund, T., Molina-Salgado, H., Brew, C., Bennett, D.: Building and Querying an Enterprise Knowledge Graph. IEEE Transactions on Services Computing **12**(3), 356–369 (2019). https://doi.org/10.1109/TSC.2017.2711600
92. Villazon-Terrazas, B., Garcia-Santa, N., Ren, Y., Faraotti, A., Wu, H., Zhao, Y., Vetere, G., Pan, J.Z.: Knowledge Graph Foundations, pp. 17–55. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-45654-6_2, `https://doi.org/10.1007/978-3-319-45654-6_2`
93. Wang, J., Krishnan, S., Franklin, M.J., Goldberg, K., Kraska, T., Milo, T.: A sample-and-clean framework for fast and accurate query processing on dirty data. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 469–480 (2014)
94. Yan, Y., Meyles, S., Haghighi, A., Suciu, D.: Entity matching in the wild: A consistent and versatile framework to unify data in industrial applications. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. pp. 2287–2301 (2020)
95. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., Franklin, M.J., Shenker, S., Stoica, I.: Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12). pp. 15–28 (2012)
96. Zhao, C., He, Y.: Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In: The World Wide Web Conference. pp. 2413–2424 (2019)
97. Zhou, B., Svetashova, Y., Byeon, S., Pychynski, T., Mikut, R., Kharlamov, E.: Predicting Quality of Automated Welding with Machine Learning and Semantics: A Bosch Case Study. International Conference on Information and Knowledge Management, Proceedings pp. 2933–2940 (2020). https://doi.org/10.1145/3340531.3412737