A Survey of Schema Matching Research using Database Schemas and Instances

Ali A. Alwan
International Islamic University Malaysia, IIUM,
Kuala Lumpur, Malaysia

Azlin Nordin
International Islamic University Malaysia, IIUM
Kuala Lumpur, Malaysia

Mogahed Alzeber
International Islamic University Malaysia, IIUM
Kuala Lumpur, Malaysia

Abedallah Zaid Abualkishik
College of Computer Information Technology
American University in the Emirates
Dubai, United Arab Emirates

Abstract*—Schema matching is considered as one of the essential phases of data integration in database systems. The main aim of the schema matching process is to identify the correlation between schema which helps later in the data integration process. The main issue concern of schema matching is how to support the merging decision by providing the correspondence between attributes through syntactic and semantic heterogeneous in data sources. There have been a lot of attempts in the literature toward utilizing database instances to detect the correspondence between attributes during schema matching process. Many approaches based on instances have been proposed aiming at improving the accuracy of the matching process. This paper set out a classification of schema matching research in database system exploiting database schema and instances. We survey and analyze the schema matching techniques applied in the literature by highlighting the strengths and the weaknesses of each technique. A deliberate discussion has been reported highlights on challenges and the current research trends of schema matching in database. We conclude this paper with some future work directions that help researchers to explore and investigate current issues and challenges related to schema matching in contemporary databases.

Keywords*—Data integration; instance-based schema matching; schema matching; semantic matching; syntactic matching

I. Introduction

Nowadays, integrating and managing a tremendous amount of data has been extremely simplified due to the advancement in information technology. Several solutions have been proposed to combine data from different heterogeneous sources to form a unified global view. This process, called data integration aim to represent data in one single view and facilitate the process of interacting with the data to be appearing as one single information system [1]. However, it is very challenging to integrate and manage data from several sources that are being independently developed. This is due to the fact that there are different representations of these sources, and data sources might not be designed in a way to adopt the same abstraction principles or have similar semantic concepts to be fully used [2]. Besides, there might be various terminologies used to describe and store information which might negatively influence in the process of integrating the data [3].

Many firms might attempt to integrate some developed heterogeneous data sources where these businesses have various databases, and each database might consist of a vast number of tables that encompass different attributes. The heterogeneity in these data sources leads to increasing the complexity of handling these data, which result in the need for data integration [4]. Identifying the conflicts of (syntax (structure) and semantic heterogeneity) between schemas is a significant issue during data integration. For this reason, schema matching has been proposed to handle the process of discovering the correspondence between schema and resolve conflicts when occurred.

Nevertheless, using schema matching approach is inappropriate when databases are developed separately and without unified standards [5]. Furthermore, it is impractical to employ the schema design information “schema attributes” to determine the correspondences attributes when different abbreviations of attribute names “column’s names” is used to represent the same real world entities or objects [2]-[5].

Consequently, discovering instance correspondences become an alternative approach for schema matching when schema information is not available or insufficient to be used for matching purposes. Instance-based schema matching attempts to extract the semantic relationship between targeted attributes via their values “instance”. Therefore, if the schema matching approach fails to detect the match, then the instances will be looked at to carry out the matching process. In this paper, we surveyed and examined some well-known techniques of instance-based schema matching. We described the strengths and the weaknesses of these techniques and end the paper with some future work directions that can benefit the researchers in the area of data integration.

This paper is organized as follows. Section II presents the schema information levels. Section III presents and explains the classification of schema matching methods and the process of instance-based schema matching. In Section IV, the techniques applied based on instance level matching has been explained. The related works for instance-based schema matching have been reviewed and reported in Section V. The discussion on the topics presented in this paper is reported in Section VI. Conclusion is presented in the final Section VII.

II. Schema Information Levels

Due to the rapid development of information systems, the demand for schema matching solutions is growing dramatically [7], [8]. For example, the role and tasks of the enterprise databases evolved from the traditional use of storing and manipulating data to be an effective tool for data analysis and interpretation. Different heterogeneous databases might need to be integrated for various purposes. The heterogeneity between databases encompasses the structure and the semantic, which have resulted in the necessity of the schema matching [2]. The driving force behind the significant development in database role is due to the complexity in obtaining data from various heterogeneous sources. Besides, the need for intelligent decision supports tools that extract heterogeneous data to ensure the best decision for users. Identifying the correspondences (matches) between database schemas has been commonly referred to as a schema matching problem [6], [7], [9].

There are three types of information that commonly used to solve the problem of schema matching by identifying the semantic of schema attributes and detect the correspondences between database schemas, i.e., 1) schema information; 2) instances; and 3) auxiliary information [9], [10]. Several solutions have been proposed aiming at handling schema matching based on the available schema information [12], [13]. These information help in preventing the incorrect match between schema attributes and lead to detect the similarities between schema attributes, particularly for semantically complex matching process. There are many beneficial levels of information that can be utilized to identify the schema matching. This includes metadata level, instance level, and auxiliary level [2], [10]. Apparently, several approaches have been proposed employed levels of schema information. Some of these approaches rely on utilizing each level independently as identified individual matcher based on their problematic situations and information available [10], [15]. While, other approaches involve a combination of the individual matcher to enhance the matching results [7], [16]. Basically, schema information has been classified into three levels, namely, schema level, instance level, hybrid level and auxiliary level. These schema information levels are further elaborated below:

A. Schema Levels

Schema level information consists of three levels of information, which are 1) linguistic level; 2) constraints level; and 3) structure level [16]. Linguistic level uses meta-data information such as the attribute’s names or abbreviations and available textual descriptions to find the correspondences between schemas [5], [8], [13]. While, constraint level relies on the data types of the database attributes such as (string, numeric, and char), the ranges of instances, and different types of keys (primary, super, uniqueness) [13], [16]. Lastly, the structure level utilizes the internal and external structure of the schema and the cardinalities between schema attributes [13], [16].

B. Instance Level

Instance level information, which is also known as contents level has been extensively applied as an effective tool to determine the correspondence between schemas. In many cases, it is not easy to obtain information from the schema structure as either it is not available or the information is meaningless and could not be used for the matching purpose [5], [10], [17]. Thus, in such cases, instances are considered as the most efficient and reliable source of information to identify the correspondences between attributes and determine the similarities and corresponding attributes of schema based on exploiting the characters of available values/instances.

C. Hybrid Level

Hybrid level retrieves information from the combination of both schema metadata (attribute names, data type, structure and description) and instance level (values/instances) [8], [13], [15]. Several criteria and sources of information might be taken into consideration to achieve the matching between schemas. Among these criteria in sources includes name matching and thesauri together with compatible data types that lead to improving the performance through providing best-combined match candidates compared to the individual performance of different matchers [15].

D. Auxiliary Level

Auxiliary level information is the process of combing existing schema information along with additional information obtained throughout external sources. Examples of external sources include WordNet/Thesauri, and dictionaries can be used for identifying the semantic relationships between schema attribute names or abbreviations such as synonymy and hyponyms in order to determine the similarities if it exists [5], [13].

III. Classification of Schema Matching Methods

In the literature, there have been many schema matching methods developed with the aim of identifying the match between database tables. There are a good number of surveys that discussed, classified and examined these methods [16], [18]. For instance, E. Rahm and P. Bernstein [11] have suggested a taxonomy that classified the schema matching methods into two categories, namely: individual matcher and combining matchers as depicted in Fig. 1.

For individual matchers, only one single parameter takes into consideration to compute the mapping between instances. Individual matchers concentrate on the available schema metadata (metadata) in terms of integrity constraints, attributes names, descriptions, and schema structures with disregard to the lowest level of information (instance) [16]. It is very common to use schema information to perform the matching between less complex databases, and it is very beneficial in the case of absence of instance level data [8]. In contrast, combining matchers either involves many criteria (i.e. name and type equality) to form hybrid matcher or combines multiple match results from various match methods to form a composite match.

Individual matcher has been predominantly adopted by a considerable number of researches and studies which reflected the trends toward concerning single matcher. Studies conducted by [4], [5], [19] emphasized the essential role of instance level matching (instance matcher) in extracting semantic similarity of schemas. These studies attempt to improve the matching process in different application domains. The application domains include Domain-Specific-Quire, data integration, and mediating databases. Additionally, D. George [20] suggested different classifications of schema matching via data integration approaches. He categorized them into two layers, namely, semantic (meaning), syntax (format), and schema (structure). He argued that there are different kinds of conflicts occurred between database tables, such as naming conflicts and structure conflicts, which is different terminologies used to represent entities or attributes names such as synonyms and homonyms. Structure conflicts that involve several types such as type conflicts, dependency conflicts, key conflicts, and behavioral conflicts. In the following, we examine and discuss the schema matching approaches illustrated in Fig. 1.

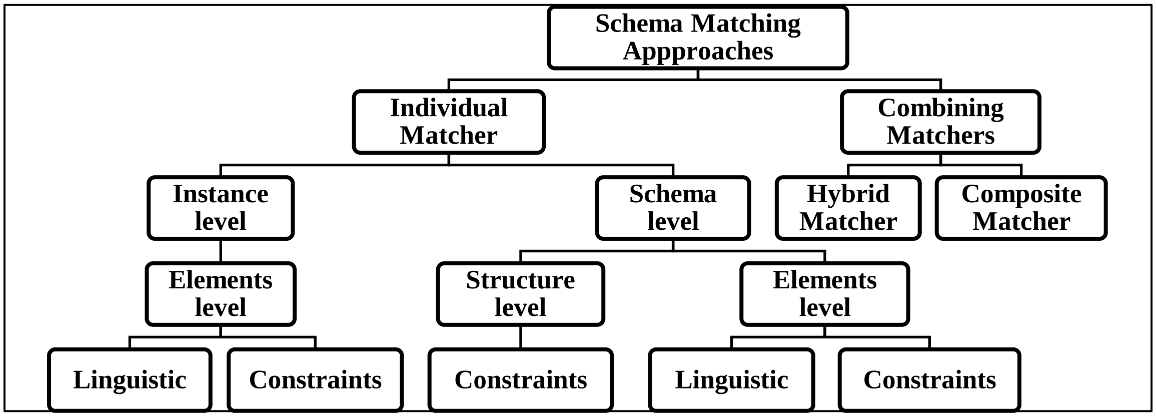

Fig. 1. Classification of schema matching methods.

A. Schema Level Matching

Schema level matching methods utilize the available schema information of the database such as name, description, type of data, constraint, and schema structure in order to identify the match between two attributes of the database schemas. Most often, more than one candidate match might be produced for each candidate, with estimated degree of similarity in the range between 0 and 1. The closer the degree of similarity from one is the more similar. Two levels under schema level matching can be exploited to define the correspondence between attributes, which are element level and structure level. Moreover, there has been a trend to consider the logs query as an additional level of information for schema matching by a number of researchers and studies [6]. This new approach attempts to extract attributes usage of each targeted schema from the logs of queries concerning the schema relationships, and their features such as joins and with aggregate functions [6]. Besides, the click logs have been mainly utilized for keyword queries of an entity search engine in order to determine the terms of the search. This will let to categorize the schema attributes that share similar search queries as candidates' match [16].

1) Element Level Matching

Element level matching aims at employing the elements belongs to the source schema to determine the matching elements of the input target schema. In many cases, it is possible to exploit the schema elements at the finest level, which called atomic level, such as attributes in an XML schema or attributes in a relational schema. An example of atomic-level for the schema fragments is illustrated in Table 1. It can be observed that Address.ZIP ≅ CustomerAddress.PostalCode represents an atomic-level schema matching between S1 and S2 elements [10], [11]. Element level matching also focuses on exploiting two levels that are linguistic matcher and constraint matchers.

a) Linguistic Matcher

Linguistic matcher involves the available linguistic information of the database schemas such as attributes names and descriptions of the attributes in order to determine the match between the schemas [21]. This approach is very common for databases with the centralized environment, where the similarities between attributes names can be described by their meanings. It is also used for semi-structure databases, where schema descriptions are well-defined. The idea of the linguistic match is to transform the attribute's names into a canonical model (form) through the tokenization method in order to compare these names equality [22].

TABLE I. FULL VERSUS PARTIAL STRUCTURAL MATCH

S1 elements	S2 elements	
Address Street City State ZIP	Customer Address Street City USState PostalCode	Full structural match of Address and Customer Address
Account Owner Name Address Birthdate TaxExempt	Customer C nae CAddress CPhone	Partial structural match of Account Owner and Customer

b) Constraint Matcher

Constraints are a very useful facility that most often used on database schemas to define the data types, the range of values, the uniqueness, the relationships types and cardinalities. In many cases, if the source and target input schemas contain a sufficient amount of constraint information, it can help the matcher technique to identify the similarity between schemas and provide a precise match between schema attributes. For instance, the similarity score can be introduced based on many factors such as the similarity of data types or domains. Besides, some key characteristics can also be involved to compute the similarity score, including primary key and foreign key. Furthermore, the relationship cardinality that identifies the level of relationship between the attributes such as 1:1 relationship and of is-a relationships [11], [15], [16]. However, it is not always applicable to use constraint information alone to obtain an accurate matching result. In some cases, constraint information might lead to imprecise match due to the comparable constraints among attributes in the schemas. Nevertheless, exploiting constraints information helps to reduce the number of match candidates and might be combined with other matchers (e.g., linguistic matcher) [2], [11], [15].

2) Structure Level Matching

Structure level matching used the structural information about database schemas to determine the match between schemas. Structure level matching concentrates on the structures and the constraints information about the targeted schemas to extract the similarity between the attributes [24]. There are many possibilities to match combinations of various attributes in a structure, depending on the completeness of the structural information and the required precision. In the ideal case, there should be a full matching of all the attributes of the source and target schemas. However, in some cases, a partial match between some attributes might be introduced, which is needed when there is a comparison of the sub-schemas. Notice the example given in Table 1, where partial match occurred between Account Owner and Customer schemas. It is also possible to use constraint-based matcher as an alternative matcher in this level, exploiting the constraints information such as data types, value ranges, nullability, and referential integrity (foreign keys) [2], [9], [10], [15], [23].

B. Instance Level Matching

Instance level approaches employ the available instances as a source to identify the correspondences between schema attributes. It is not always possible to utilize the schema information to perform an accurate match between schemas. In many cases such as semi-structured databases, information about schema might not be available or limited to be used for precise schema matching result [2], [10], [17], [24]. Hence, for such cases, instances could be used as a source for determining the corresponding of attributes. Instance-level data could be used as a significant alternative source contributing toward accurate matching due to its valuable contents and the meaning of schema attributes.

C. Combination of Multiple Matcher

There are several approaches with many variations of matchers have been proposed in the literature. Each approach has its strengths and weaknesses, and no single approach fits all cases and can give a reliable match. Many attempts have been conducted employing multiple approaches to form hybrid matcher that involves multiple criteria to identify the match between schema attributes. Besides, other approaches endeavor to develop a composite matcher benefiting from the independent matching results produced by different approaches [8], [10], [11]. Hybrid and composite matchers are further explained as follow.

1) Hybrid Matcher

Hybrid match aims at combining several matching approaches in a single approach to performing a precise match between possible candidates, taking into consideration multiple criteria and different sources of information. This includes name matching and thesauri combined together with data types to provide more accurate matching results while maintaining high performance compared with separated individual matchers.

2) Composite Matcher

Composite matcher intends to carry out the independent match on database schemas using different approaches and then combine the outcomes. Doing so allows selection of the most appropriate matchers to be implemented. Composite matcher has a greater flexibility compared to hybrid matcher as it exploits the application domain and input schemas information, while different approaches can be used for structured versus semi-structured schemas [10], [11], [25].

IV. Techniques Applied for Instance Level Matching

Most of the previous approaches for instance based schema matching is designed with the aim of determining the correlations and identify the correspondences between attributes depend on data instances that are more semantically and syntactically [5], [10], [13]. This concern on data instances reflects the fact that we need to utilize a certain technique to explore the semantic and syntactic similarities throughout the matching process [20]. In many real-world database applications, the sources of attributes are developed separately by different developers, which results in differences in terms of syntax and semantics of the schema attributes. This research work examines the most predominant techniques that rely on syntactic and semantic. Syntactic techniques encompass N-gram, and regular expression [2], [14]. While, semantic techniques include Latent Semantic Analysis (LSA), WordNet, Thesaurus and Google similarity [2], [10]. These techniques are explained in further details in the following subsections.

A. Syntactic Techniques

Many schema matching techniques have been developed for the syntactic heterogeneity of the database schemas. Identifying the similarities between different schemas via matching process would not be a trivial task, due to these heterogeneities [13]. In addition, data with numerical values and acronyms are typically described according to certain patterns, which are better suited for syntactic heterogeneity analysis [14]. In this respect, some strategies have been suggested to draw syntactical patterns, and identify related values ranges, for instance-based schema matching [11]. The following subsections demonstrate the details of syntactic techniques that have been utilized widely by previous approaches.

1) N-gram

The N-gram is a model that has been extensively used for different tasks such as spelling correction, word breaking and text summarization and recently for analyzing matching purposes [4]. The analytical process involves the fragmentation of words or texts sequentially into consecutive tokens. As a result, N will be a computer, which represents the possible tokens of the desired word, which are so-called “unigram”, and a string of M letters would subsequently have (M-2) grams. For instance, considering the desired word is “address” and its grams are three sets as the desired word in the matching task. The possible tokens of the word “address” would be denoted as St (“address”) = {add, ddr, dre, res, ess}, where S is a string and t is an integer that represents the word and its length’s set of grams respectively. Similarly, N-gram can be obtained via fragmentations of the characters of strings [5]. Although the N-gram technique is well understood and easy to implement, its reliability is questionable in the case of absence or the lack of a common and shared values between schema attributes [2], [4], [5], [10].

2) Regular Expression

Some studies have suggested the utilization of regular expression in term of instance based schema matching [2], [10]. It is known as RegEx, which defined as a technique that describes both statistical data and texts using pattern recognition concepts to define a specific data path [2]. In fact, for each schema attributes, instances are exposed to define its pattern class, and then schemas are matched based on these patterns classes. Therefore, schemas attributes considered as a match, if they explicitly correspond to the same regular expression of the same class patterns [14]. As a result, this has led to the idea of combing constraint-based with the instance based schema matching for further enhancement of the efficiency and accuracy of the matching results.

B. Semantic Techniques

For semantic techniques, the evaluation criteria are based on both the instances point out to the same definitions of the concepts of the real world entities or represent the same meanings [5], [10]. Different types of semantic heterogeneity of a schema have been defined in the literature such as classes, data sets, and structure [20]. Hence, considerable numbers of techniques that can extract the semantic relationships among schemas have been proposed in the literature. In this research work, we have focused on three techniques, namely, Latent Semantic Analysis (LSA), WordNet/Thesaurus and Google similarity. These techniques have been used most frequently in the literature representing semantic technique due to their accurate results in identifying the match between attributes [26], [27].

1) Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA), which is also known as Latent Semantic Indexing (LSI) applies a word-to-word matching called a corpus-based semantic similarity [28]. It is typically performed by considering the occurrences of the words in the corpus over the certain collection of documents [10]. The main advantage of the LSA is the appropriate representative of the synonymy, polysemy, and term dependence over the documents. However, LSA is a lack of efficiency and time constraint. These are because, during the search, the targeted query is compared to every document in the collection, including some terms that do not share in common with the query. Besides, LSA works within a limited number of closed collections of documents [10], [28].

2) WordNet/Thesaurus

WordNet/Thesaurus defined as a huge lexical English language database that has been developed and maintained by Princeton University as the product of a research project drawn up in the home (insourcing). It consists of three integrated sub- databases. These sub-databases contain a variety of English terms including nouns, verbs, adjectives and adverbs grouped into arrays of cognitive synsets (synonyms), and antonyms. One of the advantages of WordNet is the ability to interlink words by their specific senses, and to label the words neatly by writing the word semantic relations [29]. However, it does not produce obvious patterns other than the meaning similarities [29]. On the other hand, the use of WordNet is considered lacks the ability to interpret compound nouns (non-dictionary words), abbreviations or even acronyms [10].

3) Google Similarity

Google Similarity was initially called Google Similarities Distance (GSD). In its application, this technique relies on the largest online databases that contain a tremendous amount of online pages. Its main strength is utilizing the Google engine search methods for establishing the semantic relationships between the phrases and words, while it is applicable to other search engines and database application [30]. The automatic extraction of similarities between words and phrases used online, based on Google page counts results. As a result, the searching task for certain index terms is typically performed by counting the number of hits (where index terms exist via Google pages) [5]. The main advantage of Google similarity distance is the high level of reliability achieved through establishing the semantic relationships between words and phrase, which is based on the actual application of the English language within the society [27], [30]. In addition to the reliable interpretation of semantic, Google distance is more efficient in processing a huge collection of documents, in contrast to WordNet, and LSA [2], [10]. In short, GSD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. These concepts are represented by their labels by GSD, which are fed to the Google search engine as search terms.

V. Related Work of Instance-Base Schema Matching

Instances-based schema matching has been investigated by numerous studies that concentrate on enhancing the accuracy of the schema matching result. Different approaches have been proposed, adopted various strategies for precise determination of correspondence between attributes of schemas. From the literature, it can be summarized that there are four main strategies that exploited the contents of the database (instances) to detect the correspondence between attributes that leads to schema matching [31], [32]. These strategies are 1) neural network; 2) machine learning; 3) information theoretic discrepancy; and 4) rule based. Hence, this research work further discusses these four strategies that have been used for instance based schema matching.

A. Neural Networks

Neural network strategy relies on utilizing the available instances to generate the similarities among data, and empirically infer solutions from data without using the knowledge about the regularities [10], [33]. The idea of the neural network in identifying schema matching between schemas is as follow. It attempts to create a cluster for those attributes with instances that are uniformly characterized using feature vectors of constraint-based criteria. However, neural network strategy is very specific and domain-dependent and can only be used with that specific domain since it is trained based. In the following, we discuss the previous works related to schema matching based on neural network strategy.

L. S. Wen, and C. Clifton [33] have addressed the issue of schema matching in heterogeneous databases utilizing neural network strategy to determine the correspondences between attributes. The proposed approach attempted to employ both information (schema and instance) to derive the matching rules of the attribute automatically. However, the performance of the approach negatively influenced when using naming-based approach. You, Dong, and Wei (2005) [34] introduce a neural network Schema Matching technique based on Data Distribution (SMDD). SMDD technique attempts to benefit from the analysis of the characteristics of data distribution to capture the correspondences between schema attributes. Furthermore, a Content-Based Schema Matching Algorithm (CBSMA) adopts neural network strategy is proposed in [35]. CBSMA relies on the full discovery of data content to identify the match by first analyzing the data pattern, which is conducted by training a set of neural networks. Then, attempts to extract the identified features and cluster them to get training data and classifying data with Back Propagation Neural Network. K. S. Zaiss [15] introduced two instance based matching methods utilizing neural network strategy. The first method relies on the syntactic facts of the database schema to generate regular expressions or sample values that result into characterizing the concepts of ontology by their instance sets. The second method uses the instance sets to describe the contents of every instance using a set of regular expressions. Table 2 summarizes the neural network approaches for instance based schema matching presented throughout this section.

TABLE II. SUMMARY OF THE NEURAL NETWORK APPROACHES

Author & Year of Publication	**Accuracy**	**Handling Instances**	**Matching Process**	**Technique**	**Matching-based Approach**
L. S. Wen and C. Clifton, (2000)	P = 80%, R = 90%	String and Numeric	Semi	Semantic Integrator (SEMINT)	Semantic
L. You et al., (2005)	F= 0.65%	String and Numeric	Auto	Instance Similarity	Semantic
Y. Yuan et al., (2008)	P = 96%, R = 90%	String and Numeric	Auto	Feature Vectors	Syntactic and Semantic
K. S. Zaiss (2010)	P= 90%, R= 64% (Regular Expression) P=85%, R=66% (Feature Analysis)	String, Numeric and Date	Semi	Regular Expression & Features Matcher	Syntactic and Semantic

B. Machine Learning

In contrast, machine learning strategy develops a solution based on machine learning methods such as Naïve Bayesian classification to produce accurate matching results based on schema information. Typically, machine learning methods use both information (schema and instance) during the matching process. However, machine learning methods need to involve a training data set of correct matches that might require a large training data set to derive the most appropriate matches between schemas. There have been a variety of approaches proposed exploit machine learning methods to learn the instance characteristics of the matching or non-matching attributes and then use them to determine if a new attribute has instances with similar characteristics or not [5], [10], [32], [37]. Doan et al., (2001) [32] proposed a machine learning based system called, Learning Source Descriptions (LSD) that locates attributes matching in a semi-automatic manner. LSD achieved a limited accuracy, in the range of 71%-92% due to the mismatch of some tags, and also some tags need different types of learning because they are ambiguous. The work contributed by J. Berlin and A. Motor [36] introduced a machine learning strategy based approach named Autoplex to identify the match between schema attributes exploiting data instances. However, the experiment result showed that Autoplex performed only 0.81 for both soundness and completeness.

Moreover, learners need retraining when Autoplex applied to a new domain. F. Ji et al. (2009) [7] proposed new instance based schema matching approach based on machine learning strategy. The approach assumes that corresponding attributes are relatively equally important. The work presented by F. Ji et al. (2009) [7] is unlike the traditional approaches, which assumed that all attributes have the same degree of importance. In contrast, the proposed approach employs machine learning methods to prioritizing all schema attributes according to some predefined ranks and classes. However, the approach is suitable only for numeric instances, as the result of precision (P) dropped when string instances are considered [2], [10]. Lastly, the work contributed by M. A. Osama et al., (2017) [2] tackled the issue of schema matching based on data instances in the relational database. He has proposed an efficient schema matching approach to identify the correspondences between attributes by fully exploiting the instances for numeric, alphabetic and mix data types. The proposed approach employs the concept of pattern recognition to create regular expression based on instances in order to identify attributes matches for numeric and mix data types. Besides, for the alphabetic data type, the approach involves

Google similarity to compute the semantic similarity score to capture the semantic relationships between instances. Table 3 summarizes the neural network approaches for instance based schema matching presented throughout this section.

TABLE III. SUMMARY OF THE MACHINE LEARNING APPROACHES

Author & Year of Publication	**Accuracy**	**Handling Instances**	**Matching Process**	**Technique**	**Matching-based Approach**
A. Doan et al., (2001)	Accuracy 71% - 92%.	String and Numeric	Auto	LSD	Semantic
J. Berlin and A. Motro, (2002)	Soundness = 0.81 Completeness = 0.81	String and Numeric	Auto	Bayesian learner and classifier	Semantic
F. Ji et al., (2009)	P=85% (Numeric) P=66% (String)	String and Numeric	Auto	Random Forest (RF) based Decision Tree	Syntactic
M. A. Osama et al., (2017)	P= 96%, R= 93%, F= 95%	String Numeric and Mixed	Auto	Similarity Metrics	Syntactic & Semantic

C. Information Theoretic

The third strategy that has been used to determine the matching between database schemas is information theoretic discrepancy. Most of the approaches applied this strategy employs the mutual information and distribution values to identify the correspondence between attributes [5], [10]. Mutual information indicates either the degree of dependency between attributes, or the information shared between any pair of attributes in the source schema to determine the relationship between the attributes of the target schema [5], [37]. It helps to reduce the uncertainty between known attributes and unknown attributes. Applying information theoretic discrepancy strategy is skillful and does not need prior knowledge about the constraints. Nevertheless, methods of information theoretic discrepancy need to analyze the probabilities of overlapping in the values being compared [2], [10].

Two approaches for schema matching based on information theoretic discrepancy are proposed by K. Jaewoo, and F. J. Naughton [38] and K. Jaewoo, and F. J. Naughton [39]. The idea of these two approaches is similar to the approach proposed by L. Yan [37]. However, these approaches are further extended to handle the problem of opaque data values beside the issue of opaque column names and schema information. The work in [39] handles the remaining unsolved challenge of the previous work. This includes improving the computational complexity process of the graph-matching problem. Giunchiglia et al. (2004) [40] address the issue of the semantic match between database schemas. They have proposed an information theoretic discrepancy based approach utilizes WordNet as a knowledge source to determine the semantic relations of two concepts instead of similarity coefficient with values between 0 and 1. L. Yan [37] introduced information theoretic discrepancy based approach that tackles the issue of schema matching between schema when the interpretations of schema information are incorrect or ambiguous. This is achieved by evaluating the instances in schemas, playing as equivalent role as schema information.

In addition, T. B. Dai et al. (2008) [19] suggested an instance schema matching approach based on information theoretic discrepancy to identify the correspondences between schemas. However, the work comprises a technique that finds semantic similarity instances between compared attributes in different tables. Lastly, the work introduced by J. Partyka, et al. [41] has also highlighted the issue of syntactic and semantic schema matching in the database. They have proposed information theoretic discrepancy based approach that aims at identifying the semantic as well as syntactic correspondences attribute via their instances sets. Table 4 summarizes the neural network approaches for instance based schema matching presented throughout this section.

TABLE IV. SUMMARY OF THE INFORMATION THEORETIC APPROACHES

Author & Year of Publication	**Accuracy**	**Handling Instances**	**Matching Process**	**Technique**	**Matching-based Approach**
K. Jaewoo, F. J. Naughton, (2003)	P = 75%, R=79%	String	Semi	Un-interpreted matching technique & Two-steps technique	Syntactic
F. Giunchiglia et al. (2004)	P=100%, R=90%, F=95%	String	Auto	Ontology-based	Semantic
L. Yan, (2008)	P = 70%	String	Auto	Domain-independent schema matching technique	Syntactic
T. B. Dai et al., (2008)	Integrability = 92%	String	Auto	N-gram	Syntactic
J. Partyka et al., (2009)	-	String and Numeric	Auto	N-gram & Google Similarity	Syntactic and Semantic

D. Rule Based

Last but not least, applying rule-based methods for schema matching between database schemas leads to various benefits. This encompasses the low cost of the matching process; it is not necessary to use training data and produce a quick and concise result in capturing valuable user knowledge about the domain.

C. H. E. Cecil, et al. [42] introduced rule-based approach exploits attribute identification to explore data instances that identify the correspondence between attributes. The correspondence between attributes can be detected and integrate together; in the worst case schema information might be insufficient or misleading. To achieve accurate matching between schemas, a set of rules has been described to classify schema attributes. However, the approach needs to identify the entity identification prior the match; therefore, the approach might fail to identify precise match if entity identification is not available. A. Bilke and F. Naumann [43] introduced a rule-based approach that benefits from the existence of duplicates in a data set to automatically identify matching attributes. The approach uses the rule "two attributes match if they have the same data values".

The work presented by B. Zapilko et al. [14] addressed the issue of instance based schema matching in the database. They have proposed a rule-based approach which utilizes a predefined regular expression to identify the matching patterns of instances. The idea of the proposed approach relies on employing the available statistical data to develop precise patterns and regular expressions that can be fully exposed for schema matching. Table 5 summarizes the neural network approaches for instance based schema matching presented throughout this section.

TABLE V. SUMMARY OF THE RULE BASED APPROACHES

Author & Year of Publication	Accuracy	Handling Instances	Matching Process	Technique	Matching-based Approach
C. H. E. Cecil, et al. (2003)	Matched attributes =72%	String, Numeric and Mixed	Auto	Attribute Identification Method	Syntactic
A. Bilke and F. Naumann (2005)	P=75%, R= 87%	String	Auto	Instance Similarity	Semantic
B. Zapilko et al., (2012)	-	Statistical Data	Auto	Regular Expression	Syntactic

VI. DISCUSSION AND RESEARCH WORK DIRECTIONS

From the work presented throughout this paper, it can be concluded that matching heterogeneous databases is considered as one of the biggest challenges of data integration in database applications. Many approaches relying on metadata schema information to solve the heterogeneities among different information sources such as classes and structure information [9], [11]. However, relying only on schema information is insufficient, and in many cases might be meaningless. Furthermore, it is not always necessary that metadata schema information is present and appropriate to be used in schema matching process [2]. Due to these issues, there have been various approaches of instance-based schema matching proposed to find the correspondences between schema attributes. Most of these previous approaches attempt to exploit the available instances by treating them as strings including instances with numeric values [5], [17], [18], [37], [43].

It can also be observed that shifting to instance matching may not be an easy task as it seems due to some difficulties relevant to its application and time constraints as well as other reasons. Numerous researchers highlighted some challenges regarding instance based schema matching usage or application. For example, even though, the instance matcher is more reliable and accurate, however, it is much slower and time consuming compared to the schema (metadata) matcher because it relies on the entire contents (instances) of the schema to be verified [2]. Moreover, the content of the database is updated more frequently compared with schemas in real-world databases.

In the following we set out the most interesting areas that should be discovered by researchers raising the issue of schema matching in database. In these subsection many research opportunities can be exploited by interested researchers in the database community.

A. Incomplete and Crowd-Sourcing Databases

An interesting area that should be explore is identify schema matching based on instances in a partially incomplete database. The incompleteness of the data contained in the database adds another crucial challenge for instance-based schema matching process. In some real-world databases such as web and crowdsourcing, there might be many attributes with missing values, outdated data, or duplicated data. Therefore, these incomplete and inaccurate data have a negative impact on the reliability of the matching results. Hence, many proposals argued that the results extracted from sampling include inaccurate, or incomplete data should not be trusted [44], [46]. This reflects the challenges of sampling selections for the instance level matching which requires more attention. Besides, in cowdsourcing database the work is done by human, thus, humans are much more expensive than the machine [45]. Hence, we suggest that further research needs to be conducted to investigate the impact of the incompleteness of the data on sample selection which ultimately influences the accuracy of the matching result. Several important metrics related to cowdsourcing should be taken into consideration when design schema marching approach. This include quality control, latency control and cost control [45].

B. Uncertain Databases

Another interesting area that should be explored is an instance-based schema matching in uncertain databases. In uncertain databases, the values are not discrete and vary in a range of values [45]. Data uncertainty might also have a negative impact on the matching process and the accuracy as well. Thus, it might not be possible to directly apply the conventional instance-based schema matching technique on uncertain databases as it might incur higher processing cost and compromising the match quality. We also urge to explore new matching techniques that best fit with uncertain databases ensuring high matching accuracy and shortest processing cost.

C. Big Data

Last but not the least, big data become a formidable research area and attract many researchers due to the rapid increase in the data volumes. A hot research area that should investigated in big data is schema matching in which there are tens or hundreds of millions of records and analyzing the sample might lead to an exhaustive process that consumes a significant amount of time. Hence, applying the traditional instance-based schema matching might be inadequate and impractical due to the large size of the database which results in longer processing time and more expensive cost [46]. Thus, it is important to continue investigating and attempt to develop techniques that work for data with high volumes.

VII. CONCLUSION

Schema matching is a challenging issue in many contemporary database applications, including data integration, data warehousing, E-commerce, and semantic query processing. Schema matching aims at discovering the correspondences between attributes of database schemas. This paper investigates the current problems related to schema matching process in database systems. Besides, we provide a comprehensive classification of schema matching approaches designed for instance-based schema matching. In particular, we distinguished between schema level and instance-level, element level, and structure level, and linguistics and constraint matchers, and discussed the combination of multiple matchers (hybrid and composite matcher).

REFERENCES

[1] M. Lenzerini, “Data integration: A theoretical perspective,” Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, pp. 233-246, 2002.

[2] M. A. Osama, I. Hamidah, and A. S. Lilly, “An approach for instance based schema matching with Google similarity and regular expression,” The Int. Arab J. of Info. Tech. Pp. 755- 763, 2017.

[3] R. Gligorov, W. Ten Kate, Z. Aleksovski and F. Van Harmelen, “Using Google distance to weight approximate ontology matches,” Proceedings of the 16th international Conference on World Wide Web. ACM, pp. 767-776, 2007.

[4] P. P. A. L. Leme, M. A. Casanova, K. K. Breitman, and L. A. Furtado, “Instance-Based OWL schema matching,” Proceedings of the 11th International Conference, ICEIS 2009, pp. 14- 26, 2009.

[5] S. Munir, F. Khan, and M. A. Riaz, “An instance-based schema matching between opaque database schemas,” In Proceedings of the 4th International Conference on Engineering Technology and Technopreneuship (ICE2T). Kuala Lumpur, IEEE, pp. 177-182, 2014.

[6] H. Elmeleegy, M. Ouzzani, and A. Elmagarmid, “Usage-based schema matching,” Proceedings of the 2008 IEEE 24th International Conference on Data Engineering. IEEE, pp. 20-29, 2008.

[7] F. Ji, H. Xiaoguang, and Q. Yuanbo, “An instance-based schema matching method with attributes ranking and classification,” Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery FSKD'09. IEEE, pp. 522-526, 2009.

[8] H. D. Hong, and E. Rahm, “COMA: a system for flexible combination of schema matching approaches,” Proceedings of the 28th International Conference on Very Large Data Bases, VLDB Endowment, pp. 610-621, 2002.

[9] H. Do, “Schema matching and mapping-based data integration: architecture, approaches, and evaluation,” Saarbrucken, German: VDM Verlag, 2007.

[10] O. A. Mahdi, I. Hamidah., and S. A. Lilly, “Instance based matching using regular expression,” Procedia Computer Science, 10, pp. 688-695, 2012.

[11] E. Rahm and P. Bernstein, “A survey of approaches to automatic schema matching,” Very Large Data Bases Journal, 10(4), pp. 334-350, 2001.

[12] M. E. Ferragut and J. Laska, “Nonparametric bayesian modeling for automated database schema matching,” Proceedings of the 14th International Conference on Machine Learning and Applications. Pp. 82- 88, 2015.

[13] H. Zhao, and S. Ram, “Combining schema and instance information for integrating heterogeneous data sources,” Data & Knowledge Engineering, 61(2), pp. 281-303, 2007.

[14] B. Zapilko, M. Zloch, and J. Schaible, “Utilizing regular expressions for instance-based schema matching,” Proceedings of the 7th International Conference on Ontology Matching, pp. 240-241, 2012.

[15] K. S. Zaiss, “Instance-based ontology matching and the evaluation of matching systems,” Unpublished doctoral Dissertation. University of Dusseldorf, Germany, 2010.

[16] A. P. Bernstein, J. Madhavan and E. Rahm, “Generic schema matching, ten years later,” Proceedings of the 37th International Conference on Very Large Data Bases, 4(11), pp. 695-701, 2011.

[17] G. M. De Carvalho, H. A. Laender, A. M. GonçAlves, and S. A. Da Silva, “An evolutionary approach to complex schema matching,” Information Systems, 38(3), pp. 302-316, 2013.

[18] T. B. Dai, N. Koudas, D. Srivastava, K. A. Tung, and S. Venkatasubramanian, “Validating multi-column schema matchings by type,” Proceedings of the 24th International Conference on Data Engineering, IEEE, pp. 120-129, 2008.

[19] W. Jiying, W. R. Ji, F. Lochovsky, M. Y. Wei, “Instance-based schema matching for web databases by domain-specific query probing,” Proceedings of the 30th International Conference on Very Large Data Bases, pp. 408-419, 2004.

[20] D. George, “Understanding structural and semantic heterogeneity in the context of database schema integration,” Technical Report, 2005.

[21] P. A. Ambrosio, E. Métais, and N. J. Meunier, “The linguistic level: contribution for conceptual design, view integration, reuse and documentation,” Data & Knowledge Engineering, 21(2), pp. 111-129, 1997.

[22] C. Namyoun, S. S. I. Song and H. Hyoil, “A survey on ontology mapping,” ACM Sigmod Record, 35(3), pp. 34-41, 2006.

[23] S. Anam, S. Y. Kim, H. B. Kang, Q. Liu, “Review of ontology matching approaches and challenges,” International Journal of Computer Science and Network Solutions, 3(3), pp. 1-27, 2015.

[24] S. Jain and S. Tanwani, “Schema matching technique for a heterogeneous web database,” Proceedings of the 4th International Conference on the Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions). IEEE, pp. 1-6, 2015.

[25] Y. Gozudeli, H. Karacan, O. Yildiz, M. Baker, A. Minnet, M. Kalenderand M. Akcayol, “A new method based on Tree simplification and schema matching for automatic web result extraction and matching,” Proceedings of the International MultiConference of Engineers and Computer Scientists. Hong Kong, China, IMECS, pp. 1-5, 2015.

[26] J. Partyka, P. Parveen, L. Khan, B. Thuraisingham, and S. Shekhar, “Enhanced geographically typed semantic schema matching,” Web Semantics: Science, Services and Agents on the World Wide Web, 9(1), pp. 52-70, 2011.

[27] Y. Jiang, X. Wang, and H. T. Zheng, “A semantic similarity measure based on information distance for ontology alignment” Information Sciences, 278, pp. 76-87, 2014.

[28] K. T. Landauer, W. P. Foltz, and D. Laham, “An introduction to latent semantic analysis,” Discourse processes, 25(2-3), pp. 259-284, 1998.

[29] C. Fellbaum, “WordNet theory and applications of ontology: computer applications,” New York: Springer Science & Business Media, 2010.

[30] L. R. Cilibrasi and P. Vitanyi, “The Google similarity distance” IEEE Transactions on Knowledge and Data Engineering, 19(3), pp. 370 - 383, 2007.

[31] A. Doan, P. Domingos, Y. A. Halevy, “Reconciling schemas of disparate data sources: A machine-learning approach,” Proceedings of the ACM Sigmod Record. ACM, pp. 509-520, 2001.

[32] R. Shu, N. Xing, X. W. Evan, W. Haofen, Y. Qiang and Y. Yong, “A machine learning approach for instance matching based on similarity metrics” Proceedings of the 11th International Semantic Web Conference, Springer Berlin Heidelberg, 2012.

[33] L. S. Wen, and C. Clifton, “SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks” Data & Knowledge Engineering, 33(1), pp. 49-84, 2000.

[34] L. You, L. B. Dong, and Z. M. Wei, “Schema matching using neural network,” Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), IEEE, pp. 743-746, 2005.

[35] Y. Yuan, C. Mengdong, and G. Bin, “An effective content-based schema matching algorithm” Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering (FITME), IEEE Computer Society, pp. 7-11, 2008.

[36] J. Berlin, and A. Motro, “Database schema matching using machine learning with feature selection,” Proceedings of the International Conference on Advanced Information Systems Engineering, Springer, pp. 452-466, 2002.

[37] L. Yan, “An instance-based approach for domain-independent schema matching,” Proceedings of the 46th Annual Southeast Regional Conference (ACM-SE), ACM, pp. 268-271, 2008.

[38] K. Jaewoo, and F. J. Naughton, “On schema matching with opaque column names and data values,” Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, SIGMO, pp. 205-216, 2003.

[39] K. Jaewoo, and F. J. Naughton, “Schema matching using interattribute dependencies,” IEEE Transactions on Knowledge and Data Engineering, 20(10), pp. 1393-1407, 2008.

[40] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, “S-match: an algorithm and an implementation of semantic matching,” Proceedings of the European Semantic Web Symposium, Springer, pp. 61-75, 2004.

[41] J. Partyka, L. Khan, L. and B. Thuraisingham, “Semantic schema matching without shared instances,” Proceedings of the International Conference on the Semantic Computing, 2009, ICSC'09, IEEE, pp. 297-302, 2009.

[42] C. H. E. Cecil, C. L. H. Roger, and L. Ee-Peng, “Instance-based attribute identification in database integration,” The Very Large Data Bases Journal, 12(3), pp. 228-243, 2003.

[43] A. Bilke, and F. Naumann, “Schema matching using duplicates,” Proceedings of the 21st International Conference on Data Engineering (ICDE'05), IEEE , pp. 69-80, 2005.

[44] H. Köhler, X. Zhou, S. Sadiq, Y. Shu, and K. Taylor, “Sampling dirty data for matching attributes,” Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, ACM, pp. 233-246, 2010.

[45] M. A. Soliman, I. F. Ilyas And S. Ben-David, “Supporting ranking queries on uncertain and incomplete data,” The Very Large Data Base Journal, 19(4), pp. 477- 501, 2010.

[46] L. Guoliang, W. Jiannan, Z. Yudian F. J. Michael, “Crowdsourced data management: A survey,” IEEE Transactions on Know. & Data Eng., 28(9), pp. 2296- 2319, 2016.