CONSchema: Schema Matching with Semantics and Constraints

Kevin Wu, Jing Zhang, and Joyce C. Ho[(✉)]

Emory University, Atlanta, GA 30322, USA
{kevin.wu2,jing.zhang2,joyce.c.ho}@emory.edu

Abstract. Schema matching aims to establish the correspondence between the attributes of database schemas. It has been regarded as the most difficult and crucial stage in the development of many contemporary database and web semantic systems. Manual mapping is a lengthy and laborious process, yet a low-quality algorithmic matcher may cause more trouble. Moreover, the issue of data privacy in certain domains, such as healthcare, poses further challenges, as the use of instance-level data should be avoided to prevent the leakage of sensitive information. To address this issue, we propose CONSchema, a model that combines both the textual attribute description and constraints of the schemas to learn a better matcher. We also propose a new experimental setting to assess the practical performance of schema matching models. Our results on 6 benchmark datasets across various domains including healthcare and movies demonstrate the robustness of CONSchema.

Keywords: schema matching · constraint matching · semantic matching

1 Introduction

Schema matching in relational databases can be viewed as one of the most essential elements of data integration. The purpose is to identify correspondences among concepts across heterogeneous and potentially distributed data sources. For example, a wide variety of database systems collect similar data and each system has been customized for the company. This results in similar collections of data being stored in different formats, terminologies, and even logically arranged ways. As such, data exchange and integration can be hindered by these customized databases. Thus, schema matching becomes necessary across various domains including sharing health records [17] and merging documents with different formats [1]. Although schema matching is well-studied [1], the existing methods entail significant manual labor or fail to generalize across domains [21].

Given the rising focus on privacy across various sectors such as healthcare, there is a need to focus on schema-level rather than instance- or hybrid levels (i.e., no exchange of information related to instance-level records). Under the schema-level paradigm, only table and attribute information such as the name, description, meta-data, and summary statistics are shared. Meta-data and summary
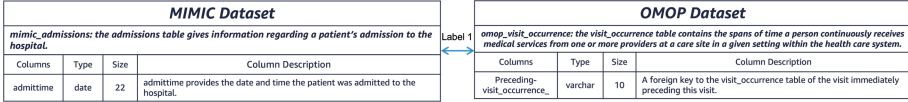
Fig. 1. Example of an identified schema match between the mimic_admission table and admittime source field in MIMIC-III and the omop_visit table and preceding_visit_time field in OMOP using both semantics and constraints.

statistics pose fewer privacy risks and are often shared for federated databases and privacy-preserving learning [4]. Several approaches have been proposed to automate schema-level matching, including constraint-based approaches [3,9,16] and linguistic-based approaches [18,21]. Unfortunately, both approaches entail background knowledge to manually define the mapping between the two relations, assume the content of the elements will be the same across the two schemas, or fail to adequately capture the similarities between the field descriptions. This can yield suboptimal performance for new applications.

Deep learning (DL) has been proposed as a new paradigm for data integration [19] given its success in other applications such as computer vision and natural language processing. DITTO, a state-of-the-art (SOTA) entity matching model, utilizes a pre-trained Transformer-based language model that can solve classification problems with entity matching [13]. However, DITTO may not perform well across different domains. SMAT, another DL model, generates a schema-level embedding using the element names and descriptions to identify the matching relations [21]. These models demonstrate the potential of DL to encode the textual information present in the attribute names and descriptions, yet ignore constraints such as data types, ranges, and key constraints.

We propose CONSchema to fuse the constraint information such as the data type, range, and key constraints with the textual information (Fig. 1 shows an example) to improve DL-based matching. The central insight is a classification model can then learn the interaction between the attribute similarity and the constraint relatedness, without requiring manual mapping. Furthermore, existing strategies do not assess the generalizability of the matching model to unseen elements within the schema. Often, training samples include either the source or target schema elements, thereby offering an optimistic assessment of the predictive performance. We introduce a *unseen partition* experimental setting and our experiments on 6 datasets demonstrate the robustness of CONSchema. The CONSchema is publicly available in the GitHub repository.[1]

2   Related Work

We briefly summarize the existing schema-level matching work focused on relational databases. Instance-level and hybrid-level models require privacy-preserving mechanisms for sensitive domains like healthcare and are beyond the

---

[1] https://github.com/kwu78/CONSchema.

scope of this work. Other related schema matching can be found in the survey [1]. Thus, we focus on existing schema-level matching methods [1].

Linguistic-level approaches calculate similarity based on the name of the attributes and/or the description of the attributes. Previous heuristic methods [3,7,15] provided decent solutions for schema matching with combinations of matchers. However, the numerical representations of the schema with distance metrics would not handle the semantic heterogeneity. Recent DL models have been introduced to perform linguistic matching. ADnEV proposed to postprocess the matching results from other matchers [18]. Unfortunately, the quality of the matchers impacts the ADnEV performance. SMAT [21] utilized attention-over-attention to pretrain a language model for the attributes, and obtained SOTA performance on several schema-level matching benchmark datasets.

The constraint-based approach relies on the meta-data of the attributes such as the data types and value ranges, uniqueness, optionality, relationship types, and cardinalities. A measure of similarity can be determined by data types and domains, key characteristics (e.g., unique, primary, foreign), and relationships [9,16]. However, precise matching requires rich constraint information. The hybrid approach combining constraint-based and instance-based approaches [2,6] has been popularized to achieve flexible and robust matchers. Unfortunately, instance-based approaches can result in privacy leakage.

3   CONSchema

3.1   Problem Statement

Given two table descriptions S_{TS} and S_{TT}, two attributes' names N_{F1} and N_{F2}, their descriptions S_{F1} and S_{F2}, and their constraints C_{F1} and C_{F2} (i.e., data type, value ranges, primary key, and foreign key) from the source and target schema respectively, we construct two sets of sequences: (1) the source sequence set $S_S = <N_{F1}>, <S_{TS} + S_{F1}>, <C_{F1}>$, and (2) the target sequence set $S_T = <N_{F2}>, <S_{TT} + S_{F2}>, <C_{F2}>$. For Fig. 1, the source example can be constructed as the sequence set "the admissions table gives information regarding a patient's admission to the hospital", "admittime", "admittime provides the date and time the patient was admitted to the hospital", "date" and size 13. For training, there is an annotated label $L(S_S, S_T)$ where 0 and 1 denotes two fields are not related (i.e., not mapped to each other) and related (i.e., corresponding attribute-to-attribute matching), respectively. The task objective is then classifying the relatedness between the two attributes.

3.2   Model

Textual Similarity Embedding. The textual embedding captures the relatedness between the two attributes' names and descriptions. The idea is that the semantic similarity between the two attributes serves as the proxy for relatedness. For example, SMAT constructs two sentence pairs where a sentence consists
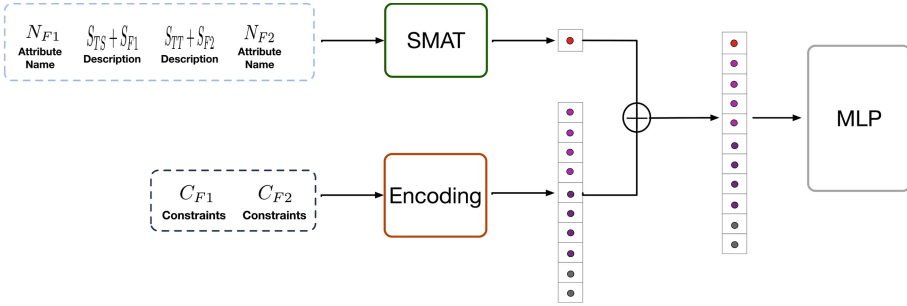
Fig. 2. Illustration of CONSchema's structure.

of the attribute name and description (e.g., $<N_{F1}, S_{TS}+S_{F1}>$). The model then learns the textual similarity between the two sentence pairs and is trained using the labels without encoding domain knowledge explicitly. CONSchema uses the last SMAT layer to serve as the attribute embedding (a 2-dimensional vector) that captures the semantic similarity between the two attributes. SMAT is chosen due to its superior performance for schema matching [21].

Constraint Encoding. The key idea behind CONSchema is to fuse the schema constraints (i.e., C_{F1} and C_{F2}) to the textual embedding. This is done by encoding the constraints into a numerical vector format where each column represents a different constraint such that a downstream classifier can then learn the importance without requiring previous knowledge. For our experiments, we focus on the data types (e.g., varchar, datetime, int, numeric), the data size for the contents (2 versus 128 characters), and key constraints. To represent the data type, we use a one-hot encoding where the value is 1 for the corresponding feature and 0 elsewhere. For example, if the attribute type is a String, then the isString feature will be set to 1. Similarly, key constraints are encapsulated using the one-hot encoding mechanism (e.g., isPrimaryKey, isForeignKey). The raw data size serves as a numeric element.

For the Fig. 1 scenario, the "admittime" attribute is a date type with a size of 22, thus the isDate feature is set to be 1 while the other data types remain 0 (i.e., isVarchar, isInt2, isInt4). In addition, the size is set to 22 (i.e., size = 22). For the OMOP attribute, since it is a varchar of size 10, the vector representation is all 0 except isVarchar = 1 and size = 10. This representation avoids the need to create ad-hoc rules for each domain. Other constraints such as uniqueness, optionality, and functional dependencies can be captured in a similar fashion, but such information is not available in the 6 datasets used for our experiments.

Final Classification. The textual similarity embedding and the constraint encoding representations are concatenated together to create the final vector representation. This fused vector encapsulates the semantic relation and the constraints of the two attributes and is then used with the annotated labels to train a

Table 1. Summary statistics of the 6 datasets used in our experiments. The columns under # capture the conversion statistics for table, attribute, and related, respectively. The next 5 columns under % represent the data type distribution where B/I2/I4 denotes boolean, int2, and int4; Fl/Arr/Oth represents float, array, and other; and PK/SK denotes primary and secondary keys. The last 3 columns provide the character length of the textual descriptions.

| | # | | | % | | | | | Length | | |
	Tab.	Attr.	Rel.	String	Date	B/I2/I4	Fl/Arr/Oth	PK/SK	Min	Avg	Max
MIMIC [11]	25	240	129	47	19	-/9/15	-/-/10	-/-	64	255	688
Synthea [20]	12	11	105	77	14	-/4/1	-/-/4	-/-	45	219	688
CMS [5]	5	96	196	53	14	-/12/21	-/-/-	-/-	54	232	688
Real Estate [8]	3	76	66	46	-	14/-/29	11/-/-	-/-	4	12	20
IMDB [12]	23	129	45	33	19	3/-/33	5/7/-	16/20	63	132	306
Thalia [10]	21	167	52	70	14	-/-/16	-/-/-	-/-	14	22	35

multi-layer perceptron (MLP).[2] MLP can capture non-linear interactions (especially with an increasing number of hidden layers) and is relatively lightweight compared to an end-to-end DL model that will incur significant training and inference overhead. Moreover, the MLP seamlessly integrates with existing DL frameworks, offering adaptability and unlocking the potential of the models. In this work, the MLP consists of two linear layers and a ReLU layer. A softmax is then applied to the outputs of MLP in order to obtain a prediction of schema matching. Figure 2 shows the entire architecture of CONSchema.

4   Experiments

Our experiments are designed to evaluate the accuracy and robustness of the model to *unseen* attributes. Existing evaluation strategies involves randomly partitioning the attribute pairs into training, validation, and test datasets. Thus, the source attribute likely occurs in at least 1 pair sample in the training dataset and provides an optimistic assessment of the model performance as partial information on the test pairs has been seen by the model. Instead, we introduce an unseen partition evaluation strategy. We randomly partition the source attributes and pair them with all the target attributes, ensuring source attributes in test never appear in training or validation, named as {dataset}_S. The same strategy is applied on the target attributes and denoted as {dataset}_T.

Our experiments evaluate the schema-matching models under our unseen partition using a ratio of 80-10-10 for train, validation, and test, respectively. The hidden units and learning rate sweep are done over the ranges [24, 36, 48, 64] and [1e−5, 5e−5, 1e−4], respectively. Since the DL-based models are sensitive to the initialization of the parameters, we train 5 versions of the model using different initial weights and report the mean value across the 5 initializations.

---

[2] We explored other models such as random forest and logistic regression and the results follow similar trends with MLP providing the largest performance boost.

Table 2. Comparison of precision (P), recall (R), and F1 (F) on the 6 datasets under the unseen partition evaluation strategy. The best performance is **bolded** and the second best is underlined.

Datasets	DITTO			SMAT			Con-MLP			CONSchema		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
MIMIC_S	0.002	0.323	0.004	**0.261**	0.467	0.284	0.041	**1.000**	0.079	<u>0.247</u>	<u>0.550</u>	**0.298**
MIMIC_T	0.001	0.285	0.002	**0.137**	0.650	**0.226**	0.008	**1.000**	0.016	<u>0.122</u>	<u>0.750</u>	<u>0.209</u>
Synthea_S	0.004	0.282	0.008	<u>0.409</u>	**0.720**	<u>0.457</u>	0.077	0.300	0.122	**0.430**	<u>0.680</u>	**0.510**
Synthea_T	0.003	0.314	0.006	<u>0.259</u>	<u>1.000</u>	<u>0.411</u>	0.256	1.000	0.408	**0.345**	**1.000**	**0.513**
CMS_S	0.156	0.321	0.210	<u>0.289</u>	<u>0.821</u>	<u>0.426</u>	0.214	0.333	0.261	**0.430**	**0.933**	**0.575**
CMS_T	0.008	<u>0.763</u>	0.015	<u>0.097</u>	0.200	<u>0.089</u>	0.009	0.316	0.017	**0.140**	**0.316**	**0.194**
IMDB_S	<u>0.149</u>	<u>0.203</u>	**0.172**	0.107	0.125	0.110	0.079	**0.375**	0.130	**0.224**	0.150	<u>0.162</u>
IMDB_T	0.056	**0.867**	**0.355**	<u>0.092</u>	0.422	0.150	0.018	0.333	0.058	**0.143**	<u>0.444</u>	<u>0.216</u>
Real Estate_S	0.138	0.109	0.122	**0.900**	0.167	0.279	0.214	**1.000**	**0.353**	<u>0.470</u>	<u>0.400</u>	<u>0.352</u>
Real Estate_T	**0.613**	<u>0.533</u>	**0.553**	<u>0.084</u>	0.500	0.111	0.065	0.500	0.115	0.082	**0.667**	<u>0.146</u>
Thalia_S	0.117	0.269	0.163	0.120	0.400	0.181	<u>0.167</u>	**1.000**	0.286	**0.252**	<u>0.760</u>	**0.374**
Thalia_T	0.052	<u>0.880</u>	0.095	0.109	0.431	0.164	<u>0.116</u>	**1.000**	<u>0.208</u>	**0.176**	0.600	**0.273**

Datasets. We assess the models on the OMAP benchmark, a schema-level matching healthcare dataset [21] mapping 3 databases to the Observational Medical Outcomes Partnership (OMOP) Common Data Model standard to facilitate evidence-gathering and informed decision-making [17], and 3 popular schema matching benchmark datasets, IMDB, Real Estate, and Thalia, used for several existing studies [12]. For each dataset, the element table name with its descriptions, attribute column name with its descriptions, attribute data type, and attribute key constraints are used to construct the sequence. The label annotation is based on the final ETL design, where a 1 denotes the table-column in the source schema was mapped to a table-column in the target schema. The summary statistics for the 6 datasets are summarized in Table 1.

Baseline Methods. CONSchema is compared against 3 matching models: (1) **DITTO** [13], a SOTA entity matching model based on the pre-trained Transformer model that matches using a sequence-pair classification problem; (2) **SMAT** [21], a SOTA schema matching model generating embeddings from the attribute name and description and then feeding the embedding to a MLP to conduct the classification task; and (3) **CON-MLP**, an MLP model using only the constraint encoding as an input. The optimal hyperparameters are determined using grid search and evaluation on the validation dataset.

5   Results

5.1   Unseen Partition Evaluation

Table 2 summarizes the results under our unseen evaluation strategy, where source ({dataset}_S)/ target ({dataset}_T) attributes in the test dataset are
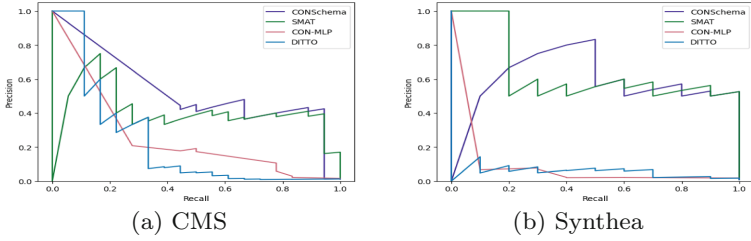
Fig. 3. Precision-Recall Curves for two of the datasets.

guaranteed not to be seen during training. In comparison with the random partition evaluation for MIMIC, Synthea, and CMS in [21], we observe that the recall for both DITTO (i.e., 0.462, 0.40, 0.636) and SMAT (i.e., 0.846, 0.950, and 0.909 respectively) is lower. This suggests the performance under random splits tends to overestimate the recall performance as having seen some of the pairings with the attributes can help the model generalize better on the test set.

We observe that CONSchema achieves the highest precision across 4 of the 6 datasets and second best for MIMIC and Real Estate on two different partitions. It also yields the best F1 score for MIMIC, Synthea, CMS, and Thalia, and the second-best F1 score for Real Estate and IMDB. Furthermore, the F1 scores for CONSchema are all better than SMAT for the 6 datasets, except **MIMIC_T**. This provides evidence that the constraints offer further information to more accurately identify the correspondences. Moreover, there are no huge differences on CONSchema between target and source partitions demonstrating its robustness.

Semantic embeddings can work even without long and well-formed textual descriptions. As shown in Table 1, the real estate database attributes are short (average of 12 characters). Examples of the textual descriptions include "water", "agent name", "type", and "firm city" which correspond to the attributes "water", "agent_name", "type", and "firm_city". From Table 2, we observe that DITTO and SMAT, which rely only on textual descriptions, can achieve reasonable performance compared to the longer counterparts such as CMS and Synthea.

The Con-MLP results illustrate the importance of our constraint vector representation. Without any textual similarity information, the model achieves better F1 scores across all but IMDB datasets when compared with DITTO. The F1 score is also better than SMAT for IMDB, Real Estate, and Thalia. To better understand the benefits of the constraint representation, Table 1 summarizes the mean, or frequency, of each encoded column for its corresponding dataset. We observe that Synthea has the highest proportion of the *Varchar* datatype when compared to the other datasets. Thus, CON-MLP is unable to achieve high recall or F1 as the constraint representation offers little information. In contrast, Real Estate, IMDB, MIMIC, and Thalia have a diversity of data types, thereby yielding improved recall scores compared to SMAT. The constraint statistics also illustrate the importance of appropriately specifying the data type and data
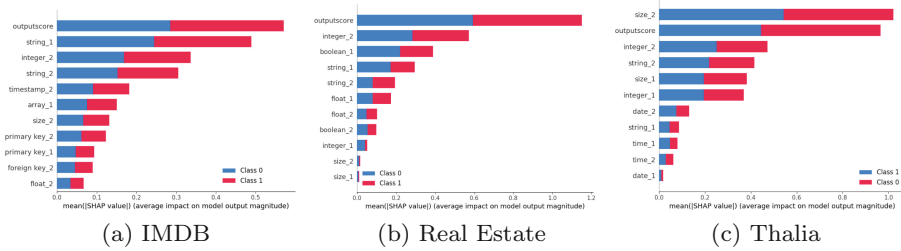
Fig. 4. Illustration of the SHAP values to explain the impact of the features in CON-Schema.

range in the database schema. Ambiguous information is likely to hurt CON-Schema more than helping it to achieve better results.

To better understand the trade-off in precision and recall, Fig. 3 plots the precision-recall curve for two datasets. For CMS, the precision of CONSchema is consistently higher than SMAT until the higher recall rates. DITTO and CON-MLP have comparable precision at the lower recall and DITTO drops below CON-MLP for recall > 0.3. For Synthea, we observe slightly different dynamics where at the lower recall (<0.2), SMAT outperforms CONSchema in terms of precision. However, for recall between 0.2 and 0.5, CONSchema outperforms SMAT significantly in terms of precision. For recall > 0.5, the two methods yield similar precision. The plots suggest for midrange recall, the constraints are particularly helpful to differentiate the positive matches. The plots also suggest that solely using constraints can generate comparable precision at lower recall rates.

5.2   Explaining CONSchema Matching Decisions

To better understand the predictions of CONSchema, we investigate the importance of the features and how they differ across the three datasets. Our analysis is based on the SHapley Additive exPlanations (SHAP) framework [14] to better understand the impact with respect to the label. SHAP is a popular explainable artificial intelligence framework that is model-agnostic. It is an additive feature attribution method and explains the change in the expected model prediction when conditioning on that feature. The SHAP analysis is performed on the best-performing model from the 5 different versions.

Figure 4 provides the summary plots for 3 of the 6 datasets where the features are sorted in descending order of their overall impact on the model output. The Y-axis labels with 1 (i.e., size_1) represent the constraints of the source dataset, whereas the 2 represent the constraints of the target dataset. The plots illustrate that the SMAT output score is one of the most important features across all datasets, which is not surprising given the results from Table 2. However, the importance of the constraints on the precision is illustrated both for IMDB and Thalia. For IMDB (Fig. 4a), which has the richest constraint diversity, we observe

that the data type (string versus integer) is almost equivalent in importance to SMAT. Similarly, for Thalia (Fig. 4c), the size of the target data type is more important than SMAT. On Real Estate (Fig. 4b), we observe boolean, integer, and string all have top SHAP values. This further illustrates the importance of constraint diversity towards improving performance.

5.3   CMS Case Study

We also performed a qualitative study on the CMS dataset by assessing three different scenarios. The first positively maps the CMS icd9_dgns_cd attribute from table inpatientclaims (varchar type of size 100 with a description of "claim diagnosis code 1 - claim diagnosis code 10") to the OMOP cause_source_concept_id element from table death (int4 type of size 10 with a description of "a foreign key to the concept that refers to the code used in the source. note this variable name is abbreviated to ensure it will be allowable across database platforms."). CONSchema correctly identifies the match over SMAT even though the descriptions are dissimilar as the constraints indicatethey might be potentially related.

The second is a negative pair where the CMS clm_from_dt attribute from the inpatientclaims table (date type of size 13 with the description "claims start date") does not map to the OMOP condition_start_datetime element from the condition occurrence table (date type of size 296 with the description "the date and time when the instance of the condition is recorded"). CONSchema incorrectly identifies a match whereas SMAT does not. We observe both the text (the start date of a claim and the start time of a medical condition) and the constraints are similar, thus leading to an incorrect conclusion by CONSchema.

The last scenario is a positive pair that matches the CMS sp_cncr attribute from the beneficiary summary table (int2 type of size 5 with the description "chronic condition: cancer") and the OMOP cohort_definition_id element from the cohort table (int4 type of size 10 with description "a foreign key to a record in the cohort definition table containing relevant cohort definition information"). Both SMAT and CONSchema incorrectly classify this sample as the textual description of OMOP is too broad (no text related to the chronic condition cancer), and the constraint type encoding does not convey enough information.

6   Conclusion

This paper proposes CONSchema, a model that incorporates schema constraints and textual descriptions to achieve better schema-level matching. As it does not utilize instance-level information and avoids directly encoding domain knowledge regarding the source and target systems, CONSchema can be used for privacy-sensitive applications. Moreover, our constraint encoding can encompass categorical-style features (type of data, type of constraint) and numeric representations (size of data) common across a variety of relational database schemas. We also propose an evaluation strategy to better understand the generalizability of existing models and demonstrate the robustness on 6 datasets.

There are several limitations of our work. First, the F1 scores are too low to be used in practice. Yet, the improvement in precision can facilitate less manual matching by prioritizing the predicted positive cases. Next, utilizing one-hot encoding to represent the constraints can yield sparse inputs for large number of constraints. This can be addressed by utilizing an auto-encoder to reduce variations or inconsistencies in the constraints originating from diverse data sources. Another limitation is the need for sufficient labels. We posit that contrastive learning techniques and data augmentation approaches may reduce the need for annotations and improve predictive performance. We also note a stronger evaluation strategy is to use the zero-shot learning framework where the model is not trained on any of the source or target attributes, and leave this for future work. Finally, CONSchema has only been demonstrated for relational schemas and should be extended to encompass a variety of data (e.g., nested data models and unstructured data) and data discovery tasks.

References

1. Alwan, A.A., Nordin, A., Alzeber, M., Abualkishik, A.Z.: A survey of schema matching research using database schemas and instances. Int. J. Adv. Comput. Sci. Appl. **8**(10), 2017 (2017)
2. Atzeni, P., Bellomarini, L., Papotti, P., Torlone, R.: Meta-mappings for schema mapping reuse. Proc. VLDB Endow. **12**(5), 557–569 (2019)
3. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with coma++. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 906–908 (2005)
4. Azevedo, L.G., de Souza Soares, E.F., Souza, R., Moreno, M.F.: Modern federated database systems: an overview. ICEIS **1**, 276–283 (2020)
5. Centers for Medicare & Medicaid Services: CMS 2008–2010 data entrepreneurs' synthetic public use file (de-synpuf) (2011)
6. Chen, C., Golshan, B., Halevy, A.Y., Tan, W.C., Doan, A.: Biggorilla: an open-source ecosystem for data preparation and integration. IEEE Data Eng. Bull. **41**(2), 10–22 (2018)
7. Do, H.H., Rahm, E.: Coma-a system for flexible combination of schema matching approaches. Proc. VLDB, 610–621 (2002)
8. Doan, A.: Learning to map between structured representations of data (2002)
9. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Schema mapping evolution through composition and inversion. In: Schema Matching and Mapping, pp. 191–222 (2011)
10. Hammer, J., Stonebraker, M., Topsakal, O.: Thalia: test harness for the assessment of legacy information integration approaches. In: Proceedings of ICDE, pp. 485–486 (2005)
11. Johnson, A.E., et al.: Mimic-iii, a freely accessible critical care database. Sci. Data **3**, 160035 (2016)
12. Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., Neumann, T.: How good are query optimizers, really? Proc. VLDB Endow. **9**(3), 204–215 (2015)

13. Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.C.: Deep entity matching with pre-trained language models. arXiv preprint abs/2004.00584 (2020)
14. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Proceedings of NeurIPS, pp. 4765–4774 (2017)
15. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: vldb. vol. 1, pp. 49–58 (2001)
16. Mecca, G., Papotti, P., Santoro, D.: Schema mappings: from data translation to data cleaning. In: A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, pp. 203–217 (2018)
17. Observational Health Data Sciences and Informatics: The book of OHDSI (2019)
18. Shraga, R., Gal, A., Roitman, H.: Adnev: cross-domain schema matching using deep similarity matrix adjustment and evaluation. Proc. VLDB **13**(9), 1401–1415 (2020)
19. Thirumuruganathan, S., Tang, N., Ouzzani, M., Doan, A.: Data curation with deep learning. In: EDBT, pp. 277–286 (2020)
20. Walonoski, J., et al.: Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. JAMIA **25**(3), 230–238 (2017)
21. Zhang, J., Shin, B., Choi, J.D., Ho, J.C.: Smat: an attention-based deep learning solution to the automation of schema matching. In: Proceedings of ADBIS, pp. 260–274 (2021)