Distribution Agreement

Signature:

Kevin Wu April 10, 2023

CONSchema: Schema matching with semantics and constraints

By

Kevin Wu

Joyce C Ho, Ph.D.
Advisor

Computer Science

Joyce C Ho, Ph.D.
Advisor

Emily Wall, Ph.D.
Committee Member

Shivani Patel, Ph.D.
Committee Member

2023

CONSchema: Schema matching with semantics and constraints

By

Kevin Wu

Joyce C Ho, Ph.D.
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Abstract

CONSchema: Schema matching with semantics and constraints
By Kevin Wu

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.

CONSchema: Schema matching with semantics and constraints

By

Kevin Wu

Joyce C Ho, Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Acknowledgments

This project brought me to the field of research in Computer Science. I started working with Dr. Ho and Jing Zhang starting from the summer of my sophomore year. I didn't know what machine learning was back then but Professor Ho and Jing were very patient and helpful in guiding me through the basics and providing me with a lot of help. As I took more courses in Computer Science at Emory University, I gained more knowledge about machine learning and felt more comfortable conducting the experiment on my own. Throughout this project, I really enjoyed the sense of accomplishment after the completion of a sub-task. I am grateful for everything I was able to achieve and I definitely want to thank my committee member Dr. Wall, Dr. Patel, and my advisor Dr. Wall as well as Jing for all the support.

Contents

List of Figures

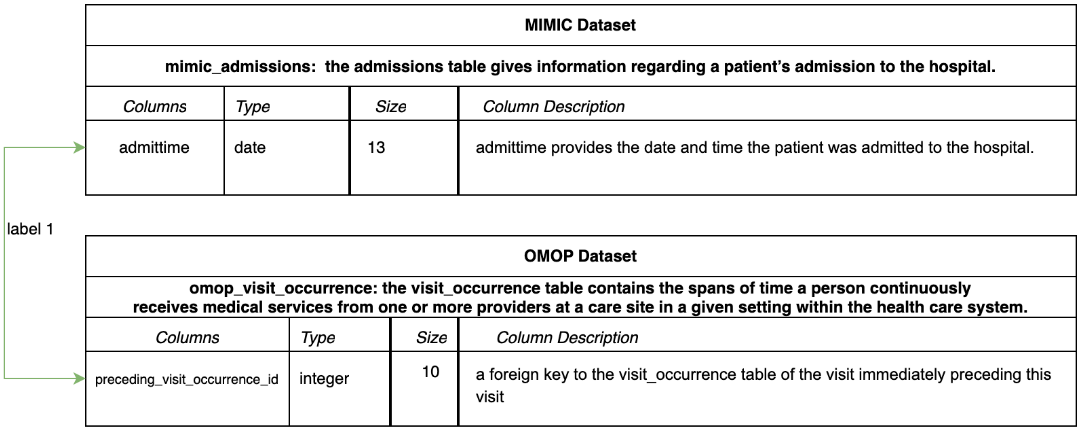
List of Tables

Chapter 1

Introduction

1.1 Motivation

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 (see Figure 1.1 for an identified match between two table attributes). This is important as a wide variety of database systems are used to 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 [11, 33, 37], linking datasets and entities for data discovery [17, 39, 41], identifying related tables in data lakes [41], and merging documents with different formats [35]. Although schema matching is a well-studied field [2], the existing methods still entail significant manual labor or fail to generalize across domains [41]. A recent study found that data scientists still spend more than 80% of their time curating the data for downstream analysis [12].



MIMIC Dataset			
mimic_admissions: the admissions table gives information regarding a patient's admission to the hospital.			
Columns	*Type*	*Size*	*Column Description*
admittime	date	13	admittime provides the date and time the patient was admitted to the hospital.

OMOP Dataset			
omop_visit_occurrence: the visit_occurrence table contains the spans of time a person continuously receives medical services from one or more providers at a care site in a given setting within the health care system.			
Columns	*Type*	*Size*	*Column Description*
preceding_visit_occurrence_id	integer	10	a foreign key to the visit_occurrence table of the visit immediately preceding this visit

Figure 1.1: 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.

Existing automating schema matching methods fall predominantly into three categories based on the level of information: schema-level, instance-level, and hybrid [2]. Schema-level information only entails meta-data information (e.g., columns or attributes in the table) whereas instance-level uses the contents of the schema (e.g., rows or tuples of information). Hybrid-level information uses both schema and instance-level information. 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. We note meta-data and summary statistics pose fewer privacy risks and are often shared in the context of federated databases and privacy-preserving learning [4, 25].

Within the schema-level matching methods, there are often two classifications. Constraint-based approaches [15, 18, 35, 40] rely on database attribute information such as data types (e.g., string, numeric, and character), the range of the values, and keys (e.g., primary, secondary, uniqueness). The other common information is linguistic-based approaches [20, 22, 26, 43], which use meta-data information including the attribute name and any available textual information. Despite the high performance achieved by these methods in various domains, both approaches entail background knowledge to manually define the mapping between the two relations. Such methods 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 domains.

Deep learning has been proposed as a new paradigm for tackling schema-level matching given its success in other applications such as computer vision and natural language processing. DITTO, a state-of-the-art entity matching model, utilizes a pre-trained Transformer-based language model that can solve the entity matching classification problems [27]. It encodes text features in the form of token sequences and introduces optimization methods such as summarizing the long text and emphasizing important information using domain knowledge. However, DITTO may not perform well across different domains, especially with abbreviations and short attribute text. SMAT, another deep learning model, generates a schema-level embedding for the attribute using the element names and descriptions [45]. The attribute embedding uses attention-over-attention to capture the relationships between the attribute name and description, thereby providing a better representation than the vanilla transformer model. These two models demonstrate the potential of deep learning to encode the textual information present in the attribute names and descriptions, yet ignore constraints such as data types, ranges, and key constraints.

This dissertation posits that schema-level matching can be further improved by integrating both constraint and linguistic-based approaches. We introduce CON-Schema to fuse the constraint information such as the data type, range, and key constraints with the textual information by extending the SMAT model (Figure 1.1 shows an example of the schema-level information used for our model). The central insight is that a lightweight classification model (i.e., random forest or multi-layer perception) can then learn the interaction between the attribute similarity and the constraint relatedness, without requiring manual mapping.

Existing strategies for evaluating schema-level matching models also fail to assess the generalizability of the model on unseen elements within the schema. Often, the models have seen training samples involving either the source or target schema elements, thereby offering an optimistic assessment of the predictive performance. We propose a new experimental setting where we evaluate the schema matching models on *unseen elements* to better assess the practical performance of the model. Our experiments on six datasets across various domains not only verify that this is a harder problem but also demonstrate the robustness of CONSchema on unseen data.

1.2 Related Work

We briefly summarize the existing schema matching work focused on schema-level information for relational databases. Instance-level and hybrid-level models require additional privacy-preserving mechanisms for sensitive domains like healthcare and are beyond the scope of this work. We also note the connection between schema matching and data discovery, where the purpose is to identify datasets that can be joined together [7, 19]. Yet, data discovery predominantly focuses on instance- or hybrid-based approaches as the rich profiles used to represent the data are acquired from inspecting the data itself. Existing methods can be classified into 2 different approaches based on the level of information: linguistic and constraint [2].

Linguistic-level approaches calculate similarity based on the name of the attributes and/or the description of the attributes. Yu *et. al.* [44] argued that the semantics of attributes can be captured by consulting a prescribed dictionary to obtain the aggregation among fields. However, consulting a synonym lexicon may not fully illuminate the relationships in the case of attribute names that contain abbreviations (e.g., DOB for date of birth, SSN for Social Security number, etc.). Nguyen et al(2019)[32] proposed a probabilistic graphical model to identify the most uncertain mappings and guide the manual validation work. Recent deep learning models have been introduced to perform linguistic matching. ADnEV proposed a deep learning technique to post-process the matching results from other matchers and the results outperformed existing models [38]. However, the reliance on the quality of the matchers can hinder the model's performance. DITTO utilizes pre-trained language models to generate token sequences to accomplish the entity-matching task. It uses optimization techniques such as adding domain knowledge, summarying long text, as well as augmenting the training data to better train the model to handle complex situations. It outperforms other EM models on the EM benchmark datasets in terms of the F1-score. SMAT [45] utilized attention-over-attention to pretrain a language model for the schema attributes, and obtained state-of-the-art 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 [2]. A measure of similarity can be determined by data types and domains, key characteristics (e.g., unique, primary, foreign), and relationships [1, 16, 31]. However, these approaches require a sufficient amount of constraint information to provide a precise match. Several recent works have focused on the hybrid approach which combines constraint-based and instance-based approaches [3, 10] to achieve flexible and more robust matchers. Unfortunately, instance-based approaches can result in privacy leakage. An extension of constraints to incorporate both the internal and external structure as well as the cardinality between the attributes is similar to the constraint-based approach. The idea is to match elements that appear together in a structure [23] and often takes the form of a graph matching problem [35]. Unfortunately, partial matches from sub-schemas can cause problems for structural matchers

[2]. As a result, recent systems have combined a variety of diverse matchers including linguistic and instance-based approaches to achieve better performance [5, 10, 30, 46].

Chapter 2

ConSchema

2.1 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 the example in Figure 1.1, the source target is then 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 the training data, there is an annotated label $L(S_S, S_T)$ where 0 denotes two fields are not related (i.e., not mapped to each other), and 1 denotes two sentences are related (i.e., corresponding attribute-to-attribute matching). Thus the task objective is to classify the semantic relation of each sentence pair with data types to reveal the attribute-to-attribute matching.

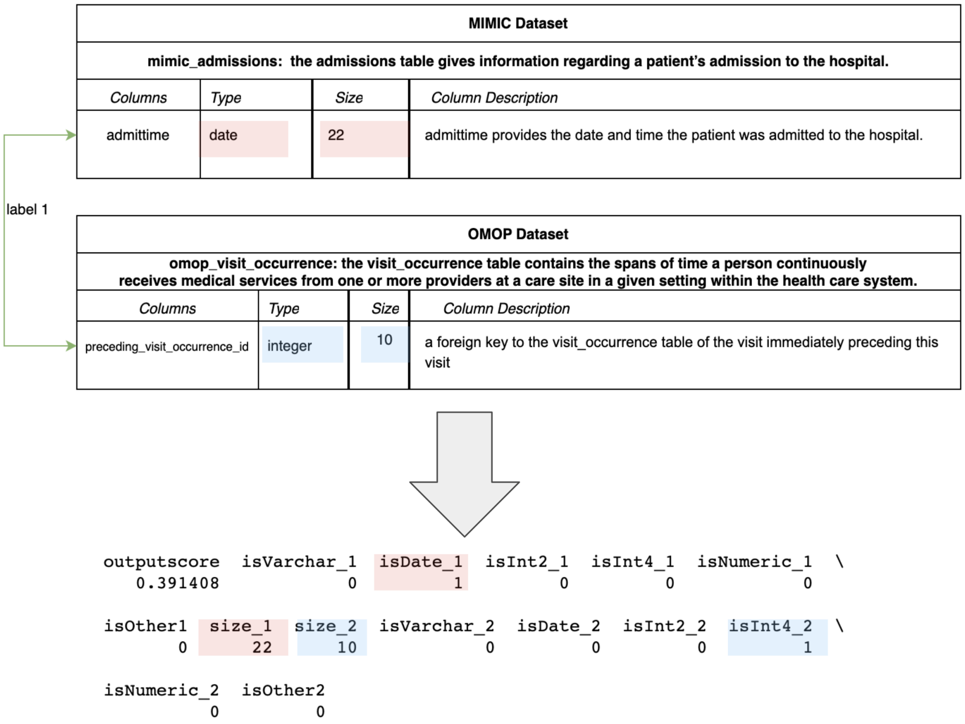
2.2 Model

2.2.1 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 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. SMAT uses a hybrid encoding to represent the word tokens and uses Bidirectional LSTM to understand the hidden semantics. Then, the Attention-over-Attention module is used to compute the attention scores for the source and target by considering the relationship between the words in the attribute and the words in the description. The classification task is performed by connecting the representation to a fully-connected layer and a softmax layer. SMAT is chosen as it has been previously shown to outperform BERT and other schema matching models for various datasets [45]. CONSchema uses the last layer of SMAT to serve as the attribute embedding (a 2-dimensional vector) that captures the semantic similarity between the two attributes.

2.2.2 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 such that a downstream classifier can then learn the importance without requiring previous knowledge. For the purpose of our experiments, we focus on the data types (e.g., varchar, datetime, int, numeric), the data size for the contents (2 versus 128 character length), and the primary and foreign key constraints. To



MIMIC Dataset			
mimic_admissions: the admissions table gives information regarding a patient's admission to the hospital.			
Columns	*Type*	*Size*	*Column Description*
admittime	date	22	admittime provides the date and time the patient was admitted to the hospital.

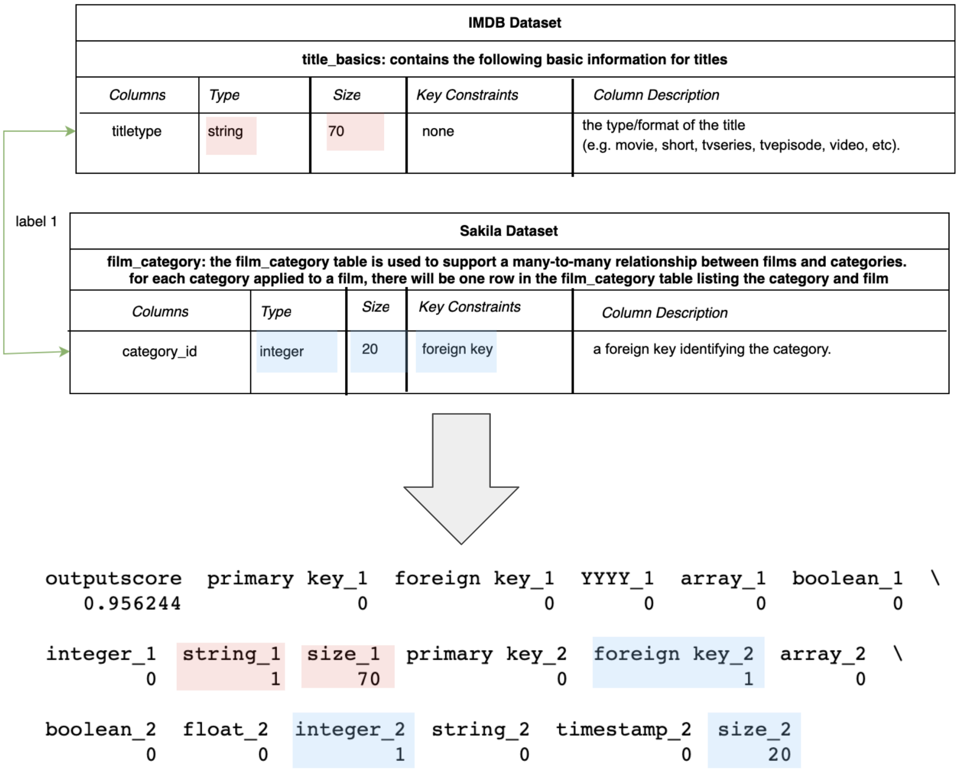
OMOP Dataset			
omop_visit_occurrence: the visit_occurrence table contains the spans of time a person continuously receives medical services from one or more providers at a care site in a given setting within the health care system.			
Columns	*Type*	*Size*	*Column Description*
preceding_visit_occurrence_id	integer	10	a foreign key to the visit_occurrence table of the visit immediately preceding this visit

```
outputscore  isVarchar_1  isDate_1  isInt2_1  isInt4_1  isNumeric_1  \
   0.391408            0         1         0         0            0

isOther1  size_1  size_2  isVarchar_2  isDate_2  isInt2_2  isInt4_2  \
       0      22      10            0         0         0         1

isNumeric_2  isOther2
          0         0
```

Figure 2.1: Illustration of constraint encoding on MIMIC dataset.

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 be 1. Key constraints will also be encapsulated using the one-hot encoding mechanism. The raw data size is captured as a numeric element for the size feature. We note that this representation avoids the need to create ad-hoc rules for each domain. Further constraints such as uniqueness, optionality, and functional dependencies can be captured in a similar fashion using the one-hot encoding representation, but such information is not readily available in the datasets used for our experiments.

Figure 2.1 provides an example of the constraint encoding for the MIMIC "admittype" attribute and OMOP "preceding_visit_occurrence_id" attribute (details of the dataset are provided in Chapter 3). For MIMIC, the attribute is a date type with a size of 22, thus the isDate feature for the source (i.e., isDate_2) is set to be 1 while the other data types remain 0 (i.e., isVarchar_2, isInt2_2, isInt4_2). In addition, the



IMDB Dataset				
title_basics: contains the following basic information for titles				
Columns	*Type*	*Size*	*Key Constraints*	*Column Description*
titletype	string	70	none	the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc).

Sakila Dataset				
film_category: the film_category table is used to support a many-to-many relationship between films and categories. for each category applied to a film, there will be one row in the film_category table listing the category and film				
Columns	*Type*	*Size*	*Key Constraints*	*Column Description*
category_id	integer	20	foreign key	a foreign key identifying the category.

```
outputscore  primary key_1  foreign key_1  YYYY_1  array_1  boolean_1  \
   0.956244              0              0       0        0          0

integer_1  string_1  size_1  primary key_2  foreign key_2  array_2  \
        0         1      70              0              1        0

boolean_2  float_2  integer_2  string_2  timestamp_2  size_2
        0        0          1         0            0      20
```

Figure 2.2: Illustration of constraint encoding on IMDB dataset.

size is set to be 22 (i.e., size_2 = 22). Similarly, for the OMOP attribute, since it is an integer of size 10, the vector representation is all zeroes except isInt4_1 = 1 and size_ = 10.

Figure 2.2 provides another example of constraint representation for mapping movies between two databases (IMDB and Sakila). For the "title_type" attribute in the IMDB table "title_basics" (string type of size 70), the constraint representation has isString set to be 1 and the size to 70. Similarly, the corresponding attribute in Sakila, the "category_id" attribute from the "film_category" table, is of type integer with a range of 20 and is a foreign key. Since the attribute is a foreign key, the vector representation then consists of all zeros except for size=20, isInt=1, and isForeignKey=1.

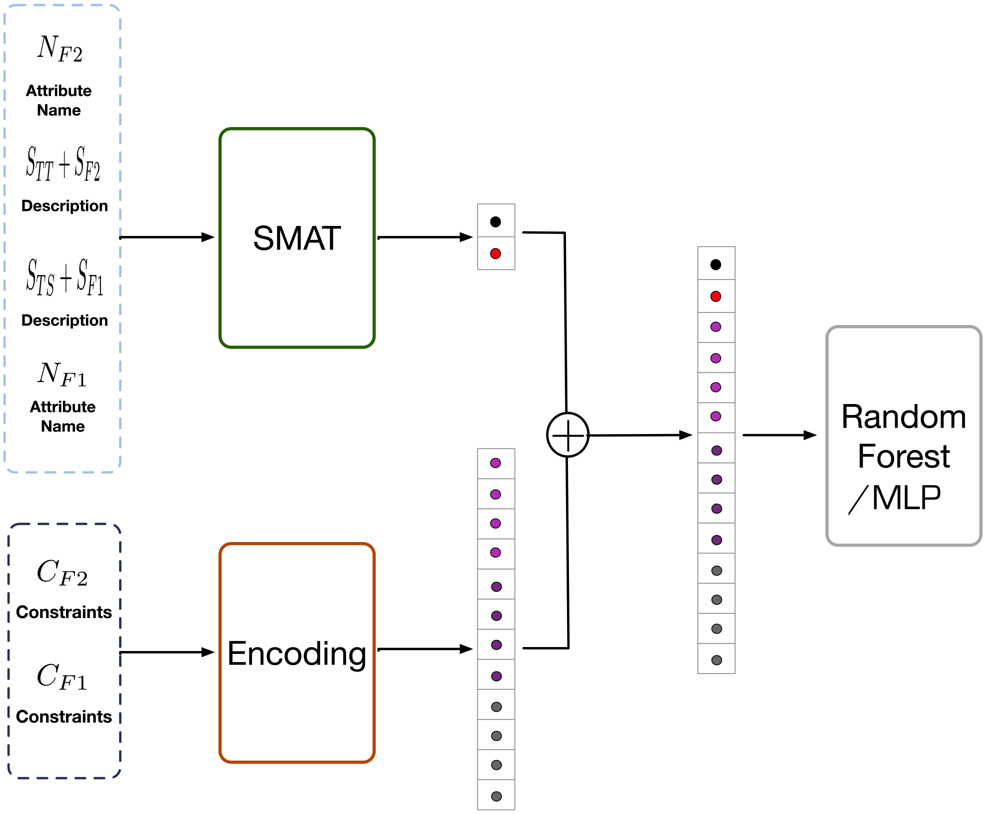


Figure 2.3: Illustration of CONSchema's structure.

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 between the two attributes and is then used with the annotated labels to train a relatively lightweight downstream classifier. For the purpose of our experiments, we explore the use of the random forest (RF) [8] and a simple multi-layer perception (MLP). RF can handle a variety of features and is robust to outliers. It can also find non-linear interactions amongst the features without necessitating a significant number of labeled samples. MLP consists of multiple layers of neurons where the activation function is used before passing the output to the following layer. With increased number of hidden layers, MLP is able to capture more complex information, especially when the input is non-linear. Moreover, both are relatively lightweight compared to an end-to-end deep learning model which will incur significant training and inference overhead. Our preliminary experiments with other simpler machine learning models such as logistic regression did not yield better predictive performances. The whole architecture of CONSchema is shown in Figure 2.3.

Chapter 3

Experiment Setting

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. Under this setting, it is likely that every source attribute occurs in at least 1 pair sample in the training dataset. Thus, evaluation of the test set provides an optimistic assessment of the model performance as partial information on the test pairs has been seen by the model.

In the real world, the algorithm should be able to determine the mapping for an *entirely unseen element*. To mimic this scenario, we propose an unseen partition evaluation strategy. Instead of randomly dividing the dataset based on the pairs, we randomly partition the source attributes and then pair them with all the target attributes. Thus the source attributes that occur in test will never appear in train or validation. As an example from 1.1, the admittime attribute from MIMIC_admission table and all its pairs appear in the training set and would not be seen in the validation or the test set.

Our experiments evaluate the schema-matching models under both strategies: the existing random partition and our unseen partition. For both scenarios, we maintain a similar partition ratio of 80-10-10 for train, validation, and test, respectively. We measure the performance of the models using precision, recall, and F1 score for the positive class. The calculations for the three measures are as follows.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.2}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

Since the deep learning based models are sensitive to the initialization of the parameters, we train 5 versions of the model using different initialization weights and report the mean value across the 5 models.

3.1 Datasets

We assess the models on the OMAP benchmark, a schema-level matching healthcare dataset [45], and 3 popular schema matching benchmark datasets, IMDB, Real Estate, and Thalia, used for several existing studies [34, 13, 6]. OMAP maps three different healthcare databases to the Observational Medical Outcomes Partnership (OMOP) Common Data Model standard to facilitate evidence-gathering and informed decision-making [33]. The description of each dataset is provided below.

- MIMIC-III [24]: A publicly available intensive care unit (ICU) relational database from the Beth Israel Deaconess Medical Center.

- Synthea [42]: An open-source dataset that captures the medical history of over 1 million Massachusetts synthetic patients.

- CMS DE-SynPUF [9]: A set of realistic claims data generated from 5% of Medicare beneficiaries in 2008.

Table 3.1: Summary statistics of the 6 datasets used in our experiments. The top 4 rows capture the conversion statistics, the next 10 rows represent the data type distribution of the schema, and the last 3 rows provide the character length of the textual descriptions.

	MIMIC	**Synthea**	**CMS**	**Real Estate**	**IMDB**	**Thalia**
# tables	25	12	5	3	23	21
# attributes	240	111	96	76	129	167
# related	129	105	196	66	45	52
# pairs	64080	29637	25632	1323	2350	1002
Varchar/String%	47	77	53	46	33	70
Date/Time%	19	14	14	-	19	14
Bool%	-	-	-	14	3	-
Int2%	9	4	12	-	-	-
Int4%	15	1	21	29	33	16
Float%	-	-	-	11	5	-
Array%	-	-	-	-	7	-
Primary Key%	-	-	-	-	16	-
Foreign Key%	-	-	-	-	20	-
Other%	10	4	0	-	-	-
Min length	64	45	64	4	63	14
Avg length	255	219	232	12	132	22
Max length	688	688	688	20	306	35

- Real Estate [14]: A set of data with information about the houses for sale as well as the sales agents.
- IMDB [36]: A collection of movies and shows information to map between the IMDB dataset and the target schema Sakila can be found in this file.
- Thalia [21]: The Test Harness for the Assessment of Legacy Information Integration (Thalia) is a publicly available set of university course catalogs that are standardized for benchmarking data integration approaches.

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 ($\{N, S_{S+F}, C\}$). The label annotation is based on the final extract, transform and load design. If the table-column in the source schema was mapped to a table-column in the target schema the label is 1, otherwise it is 0. The summary statistics for the 6 datasets are summarized in Table 3.1.

3.2 Baseline Methods

The two versions of CONSchema (i.e., CONSchema-RF and CONSchema-MLP) are compared against four other schema-level matching baseline models.

- **DITTO** [27]. A state-of-the-art entity matching model based on the pre-trained Transformer model. It casts entity matching as a sequence-pair classification problem. For the experiments, only the schema-level information and the associated tokens are used as input.
- **SMAT** [45]: A schema matching model that utilizes attention-over-attention to generate embeddings from the attribute name and description and then feeds the embedding to a multi-layer perceptron to conduct the classification task.[1]
- **CON-RF**: A random forest model that only uses the constraint encoding from Sec. 2.2.2 as an input.
- **CON-MLP**: An multi-layer perceptron model that only uses the constraint encoding from Sec. 2.2.2 as an input.
- **SMAT-RF**: A random forest model that only uses the textual similarity embedding from Sec. 2.2.1 as an input. The main difference between SMAT and SMAT-RF is the classification model (i.e., multi-layer perceptron versus a random forest).

The optimal random forest and multi-layer perceptron hyperparameters are determined using grid search and evaluation on the validation dataset.

[1] Code available at `https://github.com/JZCS2018/SMAT`

Chapter 4

Results

4.1 Random Partition Evaluation

Table 4.1 summarizes the results for the 6 datasets under the common random partition strategy. For illustrative purposes, we present the RF-version of CONSchema(i.e., CONSchema-RF). CONSchema-RF achieves the highest F1 score across all 6 datasets, with an increase ranging from $1.3 - 14.2$. Most notably, our model can offer up to a 2× improvement in precision. CONSchema also yields the best performance across all three metrics for Thalia and Real Estate. However, there is a trade-off in terms of recall for the substantial lift in precision when compared to SMAT. There is a considerable decrease for MIMIC and CMS, whereas there is a slight drop for Synthea and IMDB.

The DITTO and SMAT results on the Real Estate dataset demonstrate that textual embeddings can work even without long descriptions, as it has the smallest average length of characters (12 per attribute). The results also illustrate the importance of using SMAT as the textual similarity embedding module as it achieves better F1 performance than DITTO for all 6 datasets.

Table 4.1: Comparison of precision (P), recall (R), and F1 (F) on the 6 datasets under the random partition strategy. The best performance is **bolded** and the second best is underlined.

Datasets	**DITTO**			**SMAT**			**CONSchema-RF**		
	P	R	F1	P	R	F1	P	R	F1
MIMIC	0.003	0.462	0.006	0.115	**0.846**	0.202	**0.242**	0.550	**0.242**
Synthea	0.007	0.636	0.013	0.244	**0.909**	0.385	**0.527**	0.822	**0.527**
CMS	0.298	0.334	0.315	0.339	**0.950**	0.500	**0.432**	0.633	**0.513**
IMDB	0.626	0.695	0.659	0.687	**0.933**	0.728	**0.778**	0.880	**0.801**
Real Estate	0.872	0.772	0.819	0.914	**0.857**	0.883	**0.971**	**0.857**	**0.910**
Thalia	0.108	0.332	0.163	0.141	0.314	0.191	**0.207**	**0.457**	**0.282**

Table 4.2: 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**			**CONSchema-RF**		
	P	R	F1	P	R	F1	P	R	F1
MIMIC	0.002	0.323	0.004	0.261	0.467	0.284	**0.297**	**0.525**	**0.341**
Synthea	0.004	0.282	0.008	0.409	**0.720**	**0.457**	**0.452**	0.460	0.452
CMS	0.156	0.321	0.210	0.289	**0.821**	0.426	**0.382**	0.811	**0.513**
IMDB	0.149	**0.203**	**0.172**	0.107	0.125	0.110	**0.171**	0.125	0.133
Real Estate	0.138	0.109	0.122	**0.900**	0.167	0.279	0.357	0.267	**0.297**
Thalia	0.117	0.269	0.163	0.120	0.400	0.181	**0.223**	0.720	**0.328**

4.2 Unseen Partition Evaluation

4.2.1 CONSchema-RF

Table 4.2 summarizes the results under the unseen evaluation strategy, where source attributes in the test dataset are guaranteed not to be seen during training. First, we highlight the noticeable performance drop for DITTO across all but Thalia datasets under this evaluation strategy when compared to the results in Table 4.1. SMAT and CONSchema-RF also experience performance degradation, although it is not uniform across all measures and datasets. Recall and precision are consistently impacted on 4 of the 6 datasets except for Synthea and Thalia. 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.

The results in Table 4.2 are generally consistent with the findings on the random

Table 4.3: Comparison of precision (P), recall (R), and F1 (F) on the 6 datasets under the unseen partition evaluation strategy (continued from Table 4.2). The best performance is **bolded** and the second best is underlined.

Datasets	**Con-RF**			**SMAT-RF**		
	P	R	F1	P	R	F1
MIMIC	0.249	0.500	0.286	0.168	0.200	0.128
Synthea	0.077	0.300	0.122	0.381	0.220	0.271
CMS	0.208	0.278	0.238	0.374	**0.856**	**0.520**
IMDB	0.053	0.150	0.078	0.022	0.125	0.035
Real Estate	0.177	**0.800**	0.289	0.466	0.167	0.232
Thalia	0.104	**0.960**	0.187	0.114	0.560	0.189

partition evaluation. CONSchema achieves the highest precision across 5 of the 6 datasets and second best for the Real Estate dataset. It yields the best F1 score for MIMIC, Real Estate, Thalia, and CMS while also obtaining the second-best F1 score for Synthea and IMDB. The drop in recall is less than on the random partition evaluation for both MIMIC and CMS when compared to SMAT. Moreover, even under the harder evaluation strategy, CONSchema outperforms DITTO's predictive performance for the easier evaluation strategy.

Table 4.3 summarizes the results for the RF model using only constraints or SMAT embeddings (i.e., Con-RF and SMAT-RF, respectively). The Con-RF 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 MIMIC, Real Estate, and Thalia. This provides evidence that the constraints offer further information that can be utilized to achieve more precise correspondences.

From the results, we observe that RF can result in overfitting and poor generalization of unseen data without the additional constraint representations. Generally, passing the 2-dimension representation from the last layer of SMAT to the RF (SMAT-RF) results in a decrease in performance for MIMIC, Synthea, IMDB, and Real Estate across all three measures. This suggests that the RF model memorizes the training data and thus is unable to generalize well to the unseen data. Surprisingly, CMS and

Thalia are the exceptions to this as the results for SMAT-RF are better than that of SMAT.

As shown in Table 4.2, the F1 score of CONSchema-RF is smaller than SMAT on Synthea but not on the other datasets. To delve into the potential cause of this performance degradation, we analyze the constraint representation itself. The middle rows of Table 3.1 summarize 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 while being the second largest dataset. Given the prevalence of varchar in the dataset, we posit that CONSchema-RF cannot learn the mapping to the other data types as the constraint offers little additional information. On the other hand, Real Estate, IMDB, MIMIC, and CMS have a diversity of data types across the different categories, thereby yielding better representations as observed by the improved F1 score compared to SMAT. For Thalia, its better performance may be a result of the reduced dimension to avoid overfitting. Thalia contains the least amount of samples in our experiments. The constraint statistics also illustrate the importance of appropriately specifying the data type and data range in the database schema. Ambiguous information is likely to hurt CONSchema-RF more than helping it to achieve better results.

4.2.2 CONSchema-MLP

Table 4.4 summarizes the result under unseen partition using CONSchema-MLP and CON-MLP. The results under the MLP classifier illustrate a significant improvement in terms of F-1 score in all datasets except MIMIC compared to the results under RF from Table 4.2. The drop in performance for the MIMIC dataset may be attributed to the low performance of passing solely the constraint features to the MLP classifier as observed in the results in Con-MLP. For MIMIC, using only the constraints as features makes the MLP classifier overestimate the number of positive samples, leading to a

Table 4.4: Comparison of precision (P), recall (R), and F1 (F) on the 6 datasets under the unseen partition evaluation strategy using MLP classifier.

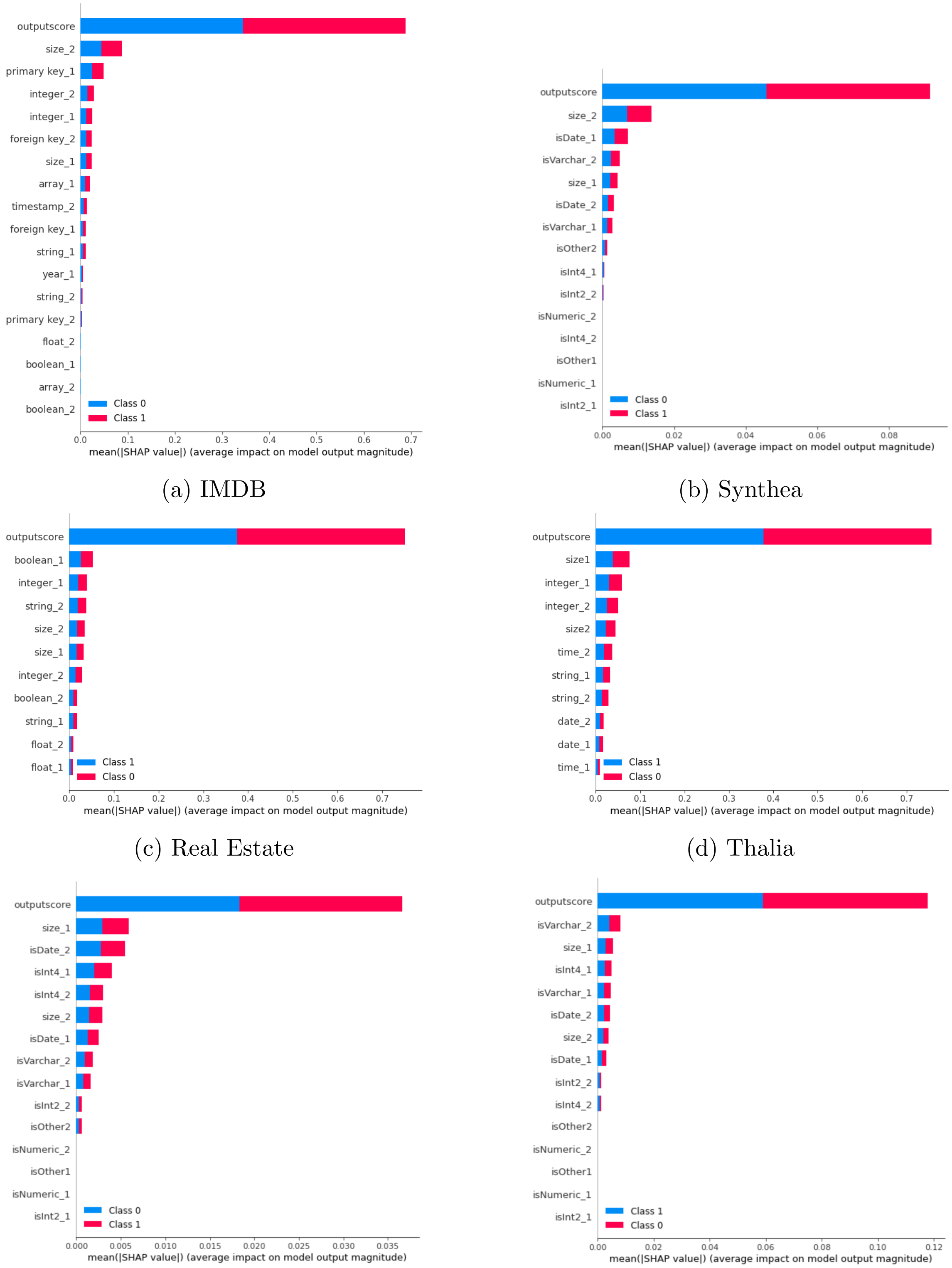
Datasets	**Con-MLP**			**CONSchema-MLP**		
	P	R	F1	P	R	F1
MIMIC	0.041	1.000	0.079	0.247	0.550	0.298
Synthea	0.077	0.300	0.122	0.430	0.680	0.510
CMS	0.214	0.333	0.261	0.416	0.933	0.575
IMDB	0.079	0.375	0.130	0.224	0.150	0.162
Real Estate	0.214	1.000	0.353	0.470	0.400	0.352
Thalia	0.167	1.000	0.286	0.252	0.760	0.374

low precision score (but a perfect recall) and a low F1 score. It is also a general trend in Table 4.4 that Con-MLP achieves a high recall and low precision. The addition of SMAT score helps CONSchema-MLP to grasp the semantic similarity between the columns, therefore boosting the precision of every dataset and making more accurate positive predictions than Con-MLP.

4.3 Explaining CONSchema Matching Decisions

To better understand the predictions of CONSchema-RF, 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 [29] 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. Although RF provides feature importance, SHAP has been shown to have the following useful properties: local accuracy, missingness, and consistency [28]. Moreover, SHAP provides the capability to tease out the feature differences with respect to the class.

The SHAP analysis is performed on the best-performing model from the 5 different versions. Figure 4.1 provides the summary plots of the 6 datasets where the features are sorted in descending order of their overall impact on the model output. The Y-



(a) IMDB

(b) Synthea

(c) Real Estate

(d) Thalia

(e) Mimic

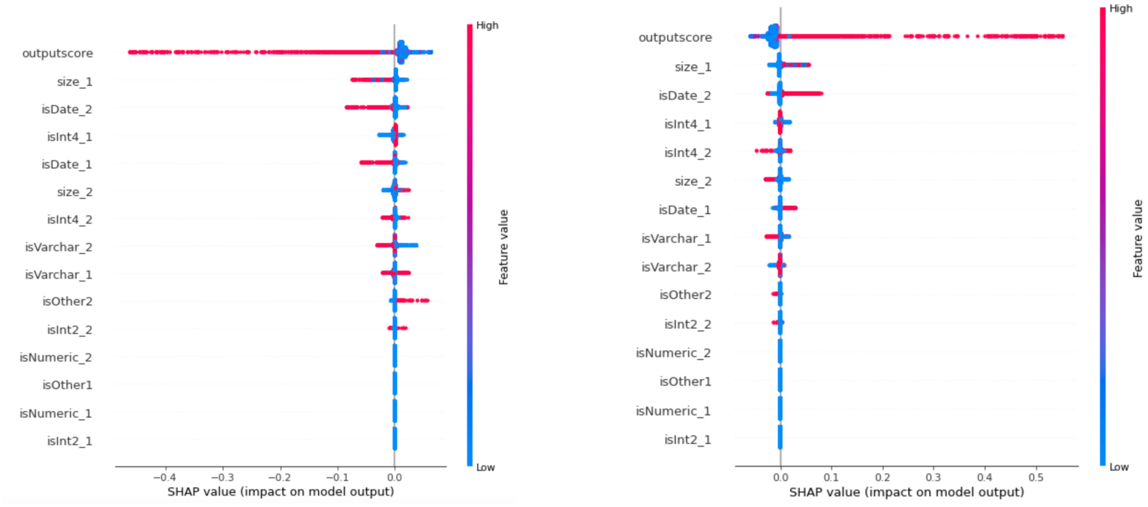
(f) CMS

Figure 4.1: Illustration of the SHAP values to explain the impact of the features in CONSchema-RF.

axis labels with 1 (i.e., size_1) represent the constraints of the source dataset, whereas the 2 denote 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 4.2.

For the IMDB dataset (Figure 4.1a), we observe that the size of the data, the array data type, and whether or not the entry is a primary key all play an important role in making the prediction. IMDB has the richest constraints compared to all the other dataset, therefore CONSchema is able to outperform SMAT in terms of F1 and precision. On the Real Estate dataset (Figure 4.1c), the data types play an important role as boolean, integer, and string have the top SHAP values outside of the SMAT score. This illustrates that diversity of constraints can also improve performance, especially with respect to recall. However, in the process of improving recall, CONSchema sacrifices on precision (as SMAT achieves a high precision with low recall). In Thalia (Figure 4.1d), we observe that the data types that matter are integer and time as the string type is common and thus offers limited information.

SHAP also allows us to assess the impact of the features on individual training instances. Figure 4.2 provides more details of the Shapley values for the positive and negative correspondences from the MIMIC dataset. In the case of both positive and negative, the output score plays an essential role in the prediction as they typically have high Shapley values (i.e., redder). More interesting is the difference in importance in the constraint type between the two classes. For the positive label (Figure 4.2b), the model learns to differentiate based on whether the data type on the source schema is an integer (isint4_1) versus the data type of the target schema is a date (isDate_2) or integer (isint4_2). This can be contrasted with the negative label (Figure 4.2a), where the determination is based on whether both data types are date. The summary plot also illustrates some of the limitations of a constraint-based only approach, as having matching data types does not necessarily mean the two attributes



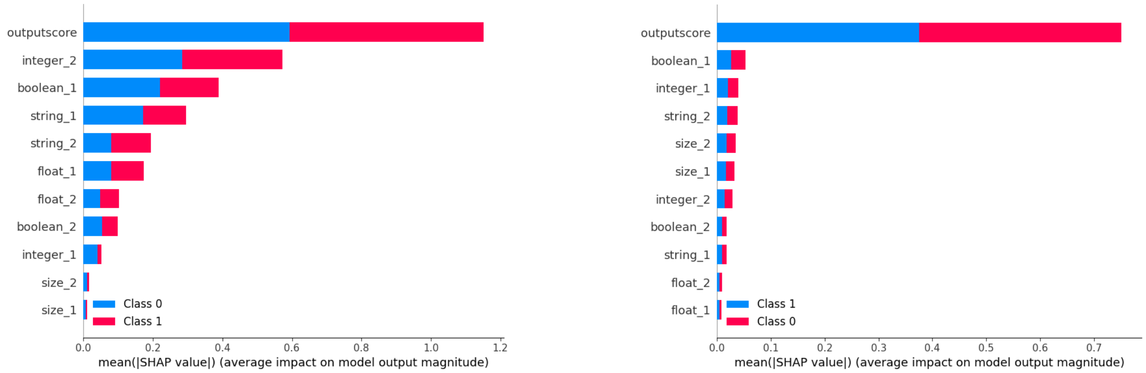
(a) Negative

(b) Positive

Figure 4.2: Illustration of the SHAP values for the positive and negative instances in MIMIC.

should be matched. Instead, fusing the semantic similarity and the constraints allows CONSchema to better identify whether there is a correspondence.

Figure 4.3 demonstrates the difference between the SHAP scores of CONSchema using Random Forest (RF) and Multi-Layer Perceptron (MLP). As observed in Figure 4.3b for Random Forest, the outputscore feature (i.e., SMAT score) has high importance whereas the other features have minimal impacts on the predictions. For the model using Multi-Layer Perceptron (Figure 4.3a), all the constraints except the size features contribute noticeably to making the positive and negative predictions. Since MLP involves non-linear activation functions as well as multiple layers of hidden layers, it is able to learn the weights of each feature through training and validation. Random Forest depends significantly on the SMAT score to make its predictions while MLP is able to take advantage of the data type constraints that yield more accurate results as shown in Table 4.2 and 4.4.



(a) MLP

(b) RF

Figure 4.3: Illustration of the SHAP values for the Real Estate dataset using MLP and RF.

4.4 CMS Case Study

To better understand the benefits and limitations of CONSchema, we performed a qualitative study on the CMS dataset by assessing three different scenarios. The first is a positive pair that 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 mapping over SMAT even though the description is not quite similar (claim diagnosis code to foreign key referring to death). However, the type and constraint size provide some indication that they 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 but SMAT does not. We observe that the text (the start date of a claim and the start time of a medical condition) and the two constraints (date types) are similar, which causes CONSchema to predict a higher probability of a match.

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.

4.5 Precision Recall Analysis

To better understand the trade-off in precision and recall, Figure 4.4 and Figure 4.5 plot the precision-recall curve for MIMIC and Synthea datasets. As can be observed for MIMIC, the precision of SMAT is lower than CONSchema-RF for recall < 0.25. However, for recall between 0.25 and 0.5, SMAT precision is identical to ConSchema-RF, before dropping below CONSchema-RF and CONSchema-MLP for the remaining recall values. Although there isn't a significant difference, the two CONSchema methods using RF and MLP demonstrate the usefulness of adding constraints to the features. It can be observed on this curve that CONSchema-MLP does not perform as well as CONSchema-RF in MIMIC dataset as its precision lies generally below CONSchema-RF except for recall between 0.15 and 0.25. Part of the reason is that as observed in Table 4.4, the precision score for CON-MLP is the worst in MIMIC among all the dataset. It suggests that the constraints themselves have limited importance when performing classification using Multi-Layer Perceptron

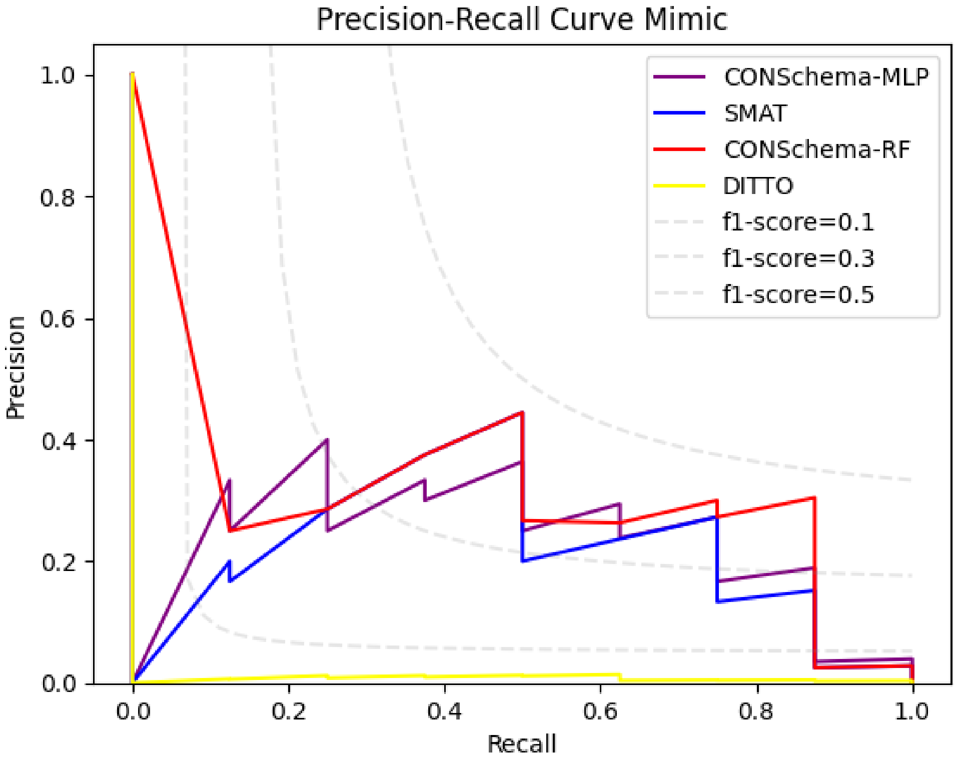


Figure 4.4: Precision Recall Curve for the MIMIC dataset.

compared to Random Forest.

A similar trend can be observed for the Synthea dataset shown in Figure 4.5 as well. For recall between 0.2 and 0.3, ConSchema-RF and CONSchema-MLP outperform SMAT in terms of precision. However, for recall between 0.3 and 0.5, CONSchema-MLP performs significantly better than all other models. For recall larger than 0.5, there is no consistent trend as for some points SMAT provides better precision whereas, for others, CONSchema-RF and CONSchema-MLP yield the same or higher precision. These results highlight that at the lower recall rates, the constraints help better differentiate the positive matches from the negative matches when using textual embeddings. This can also be seen using the Con-RF and Con-MLP results, which can yield comparable precision at lower recall rates.

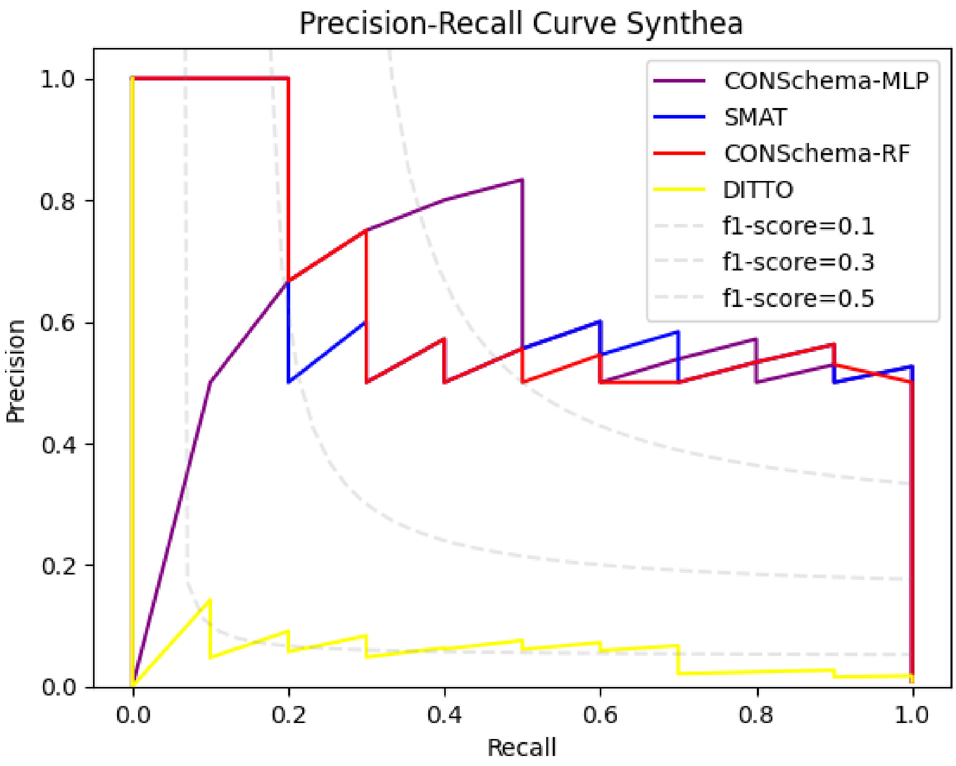


Figure 4.5: Precision Recall Curve for the Synthea dataset.

Chapter 5

Conclusion

This dissertation proposes CONSchema, a model that incorporates schema constraints to improve the predictive performance of an existing schema-level matching model. As it does not require instance-level information and avoids directly encoding domain knowledge regarding the source and target systems, CONSchema can be used for privacy-sensitive applications. We also propose a new evaluation strategy to better understand the generalizability of existing models. The experiments on 6 datasets illustrate that CONSchema can serve as the new state-of-the-art model for schema-level matching tasks.

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. 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. 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.

Bibliography

[1] Bogdan Alexe, Mauricio Hernández, Lucian Popa, and Wang-Chiew Tan. Mapmerge: Correlating independent schema mappings. *Proceedings of the VLDB Endowment*, 3(1-2):81–92, 2010.

[2] Ali A Alwan, Azlin Nordin, Mogahed Alzeber, and Abedallah Zaid Abualkishik. A survey of schema matching research using database schemas and instances. *International Journal of Advanced Computer Science and Applications*, 8(10):2017, 2017.

[3] Paolo Atzeni, Luigi Bellomarini, Paolo Papotti, and Riccardo Torlone. Meta-mappings for schema mapping reuse. *Proc. VLDB Endow.*, 12(5):557–569, January 2019.

[4] Leonardo Guerreiro Azevedo, Elton Figueiredo de Souza Soares, Renan Souza, and Márcio Ferreira Moreno. Modern federated database systems: An overview. *ICEIS (1)*, pages 276–283, 2020.

[5] Philip A Bernstein, Jayant Madhavan, and Erhard Rahm. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11):695–701, 2011.

[6] Roger Blake. A survey of schema matching research. Technical report, College of Management Working Papers and Reports, 2007.

[7] Alex Bogatu, Alvaro AA Fernandes, Norman W Paton, and Nikolaos Konstantinou. Dataset discovery in data lakes. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 709–720. IEEE, 2020.

[8] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[9] Centers for Medicare & Medicaid Services. Cms 2008-2010 data entrepreneurs' synthetic public use file (de-synpuf). `https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF`, 2011. Accessed: 2019-04-06.

[10] Chen Chen, Behzad Golshan, Alon Y Halevy, Wang-Chiew Tan, and AnHai Doan. Biggorilla: An open-source ecosystem for data preparation and integration. *IEEE Data Eng. Bull.*, 41(2):10–22, 2018.

[11] Douglas A Colquhoun, Amy M Shanks, Steven R Kapeles, Nirav Shah, Leif Saager, Michelle T Vaughn, Kathryn Buehler, Michael L Burns, Kevin K Tremper, Robert E Freundlich, et al. Considerations for integration of perioperative electronic health records across institutions for research and quality improvement: the approach taken by the multicenter perioperative outcomes group. *Anesthesia and analgesia*, 130(5):1133, 2020.

[12] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. The data civilizer system. In *8th Biennial Conference on Innovative Data Systems Research, CIDR 2017*, 2017.

[13] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, and Pedro Domingos. imap: Discovering complex semantic matches between database

schemas. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 383–394, 2004.

[14] AnHai Doan. *Learning to map between structured representations of data.* University of Washington, 2002.

[15] Ronald Fagin, Laura M Haas, Mauricio Hernández, Renée J Miller, Lucian Popa, and Yannis Velegrakis. Clio: Schema mapping creation and data exchange. In *Conceptual modeling: foundations and applications*, pages 198–236. Springer, 2009.

[16] Ronald Fagin, Phokion G Kolaitis, Lucian Popa, and Wang-Chiew Tan. Schema mapping evolution through composition and inversion. In *Schema matching and mapping*, pages 191–222. Springer, 2011.

[17] Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Seeping semantics: Linking datasets using word embeddings for data discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 989–1000. IEEE, 2018.

[18] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1190–1201. IEEE, 2019.

[19] Javier de Jesús Flores Herrera, Sergi Nadal Francesch, and Óscar Romero Moral. Towards scalable data discovery. In *Advances in Database Technology: EDBT 2021, 24th International Conference on Extending Database Technology: Nicosia, Cyprus, March 23-26, 2021: proceedings*, pages 433–438. OpenProceedings, 2021.

[20] Alon Halevy, Ema Nemes, Xin Dong, Jayant Madhavan, and Jun Zhang. Similarity search for web services. In *Proceedings of the 30th VLDB Conference*, pages 372–383, 2004.

[21] J. Hammer, M. Stonebraker, and O. Topsakal. Thalia: Test harness for the assessment of legacy information integration approaches. In *21st International Conference on Data Engineering (ICDE'05)*, pages 485–486, 2005.

[22] Bin He and Kevin Chen-Chuan Chang. Statistical schema matching across web query interfaces. In *Proc. of SIGMOD*, pages 217–228, 2003.

[23] Sumit Jain and Sanjay Tanwani. Schema matching technique for heterogeneous web database. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, pages 1–6. IEEE, 2015.

[24] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[25] Alan F Karr, William J Fulp, Francisco Vera, S Stanley Young, Xiaodong Lin, and Jerome P Reiter. Secure, privacy-preserving analysis of distributed databases. *Technometrics*, 49(3):335–345, 2007.

[26] Mohamed Salah Kettouch, Cristina Luca, Mike Hobbs, and Sergiu Dascalu. Using semantic similarity for schema matching of semi-structured and linked data. In *2017 Internet technologies and applications (ITA)*, pages 128–133. IEEE, 2017.

[27] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*, 2020.

[28] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[29] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[30] Sabine Massmann, Salvatore Raunich, David Aumüller, Patrick Arnold, Erhard Rahm, et al. Evolution of the coma match system. *Ontology Matching*, 49:49–60, 2011.

[31] Giansalvatore Mecca, Paolo Papotti, and Donatello Santoro. Schema mappings: From data translation to data cleaning. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, pages 203–217. Springer, 2018.

[32] Quoc Viet Hung Nguyen, Matthias Weidlich, Thanh Tam Nguyen, Zoltán Miklós, Karl Aberer, and Avigdor Gal. Reconciling matching networks of conceptual models. Technical report, École polytechnique fédérale de Lausanne, 2019.

[33] Observational Health Data Sciences and Informatics. *The book of OHDSI*. Independently published, 2019.

[34] Laurel Orr, Magdalena Balazinska, and Dan Suciu. Sample debiasing in the themis open world database system. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 257–268, New York, NY, USA, 2020. Association for Computing Machinery.

[35] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[36] Sakila introduction. `https://dev.mysql.com/doc/sakila/en/sakila-introduction.html`, Retrieved December 22.

[37] Fahad Ahmed Satti, Musarrat Hussain, Jamil Hussain, Syed Imran Ali, Taqdir Ali, Hafiz Syed Muhammad Bilal, Taechoong Chung, and Sungyoung Lee. Unsupervised semantic mapping for healthcare data storage schema. *IEEE Access*, 9:107267–107278, 2021.

[38] Roee Shraga, Avigdor Gal, and Haggai Roitman. Adnev: cross-domain schema matching using deep similarity matrix adjustment and evaluation. *Proc. of the VLDB*, 13(9):1401–1415, 2020.

[39] Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Synthesizing entity matching rules by examples. *Proceedings of the VLDB Endowment*, 11(2):189–202, 2017.

[40] Balder ten Cate, Phokion G Kolaitis, Kun Qian, and Wang-Chiew Tan. Active learning of gav schema mappings. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 355–368, 2018.

[41] Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. Data curation with deep learning. In *EDBT*, pages 277–286, 2020.

[42] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 08 2017.

[43] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In *Proc. of SIGMOD*, pages 95–106, 2004.

[44] Clement Yu, Wei Sun, Son Dao, and David Keirsey. Determining relationships among attributes for interoperability of multi-database systems. In *[1991] Proceedings. First International Workshop on Interoperability in Multidatabase Systems*, pages 251–257. IEEE, 1991.

[45] Jing Zhang, Bonggun Shin, Jinho D Choi, and Joyce C Ho. Smat: An attention-based deep learning solution to the automation of schema matching. In *European Conference on Advances in Databases and Information Systems*, pages 260–274. Springer, 2021.

[46] Huimin Zhao and Sudha Ram. Combining schema and instance information for integrating heterogeneous data sources. *Data & Knowledge Engineering*, 61(2):281–303, 2007.