SchemaLogix: Advancing Interoperability with Machine Learning in Schema Matching

Mohamed Raoui, Mohammed Ennaouri, Moulay Hafid El Yazidi, Ahmed Zellou
ENSIAS, Mohammed V University in Rabat, Morocco

***Abstract*—Schema matching, a fundamental process in data integration, traditionally employs pairwise comparisons to discern semantic correspondences among elements in disparate schemas. However, recent developments underscore the necessity of concurrent matching of interconnected schemas, termed schema alignment, to reconcile heterogeneous elements. This paper presents SchemaLogix, an innovative machine learning-based approach for schema matching. SchemaLogix addresses challenges such as data scarcity and domain-specific constraints through an inventive bootstrapping method, autonomously generating extensive datasets. Furthermore, SchemaLogix capitalizes on inherent alignment context constraints to optimize learning and improve precision across varied schema structures. Additionally, SchemaLogix incorporates user contributions to validate chosen correspondences, refining outputs based on valuable feedback. Empirical evaluations establish SchemaLogix's superiority over traditional methods, achieving an exceptional maximum S1 score of 0.90. These results offer practical insights for real-world applications, substantially advancing data integration and interoperability endeavors.**

***Keywords*—Interoperability; data integration; schema matching; machine learning**

I. Introduction

Schema matching involves the process of identifying semantic connections among attributes of two distinct database structures, which is crucial for facilitating data integration and system compatibility across diverse industries including e-commerce, geospatial analysis, biology, healthcare, and others.

Identifying these connections presents several challenges. First, schema elements, such as attributes representing similar concepts, may have different names across various schemas [1] [2]. Additionally, items sharing common names might actually represent different concepts. Furthermore, corresponding components between two database structures might have divergent structures. Finally, it's possible that in one schema, multiple elements symbolize a concept that would be depicted as a single item in another schema.

For instance, consider the database structures for people's information illustrated in Fig. 1. The objective of schema matching is to identify matches between elements in these schemas. In this case, the left diagram depicts how Person P structures student information within their database, while the right diagram represents the same data within another database schema employed by Person P.

This example encapsulates the inherent challenges of schema matching, where the task extends beyond mere alignment to encompass the reconciliation and harmonization of elements across disparate database structures. In essence, schema matching emerges as a pivotal linchpin in the realm of effective and cohesive information management, demonstrating its profound implications for diverse domains and industries.

Traditionally, schema alignment is typically carried out manually by experts with profound knowledge of database structures and their respective fields. However, even when performed by professionals, this task can be time-consuming, costly, and prone to inaccuracies. Over time, numerous studies and projects have addressed the topic of schema matching, leading to the creation of various articles [3] [4] [5] and the development of multiple prototypes and commercially available solutions. A substantial number of these approaches rely on predefined sets of methods and parameters [6] [7].

Other approaches rely on using machine learning to define specific models designed for each matching task [8] [9]. While heuristics can be effective in some situations, they often require adjustments to produce good results. In contrast, machine learning techniques can adapt to various matching tasks after a significant amount of training data becomes available, although obtaining this data can be challenging.

As the field has advanced, situations have arisen in the alignment of database structures where matching involves multiple data sources, such as databases and query forms [10] [11], forming what can be referred to as a network of patterns. Considering the satisfactory performance observed when applying machine learning methods in pairwise pattern matching scenarios, this study experiments with these methods in the context of pattern matching. However, this introduces challenges, including the need for a substantial volume of annotated data and the handling of imbalanced data sets, where the number of unmatched pairs far exceeds the count of corresponding pairs.

To address these challenges, various approaches are being explored, including utilizing opaque-box pattern schema alignment systems to generate training instances, leveraging network constraints to construct high-quality training sets, and incorporating user reviews to enhance final correspondences. However, these methods may introduce additional time-consuming issues.

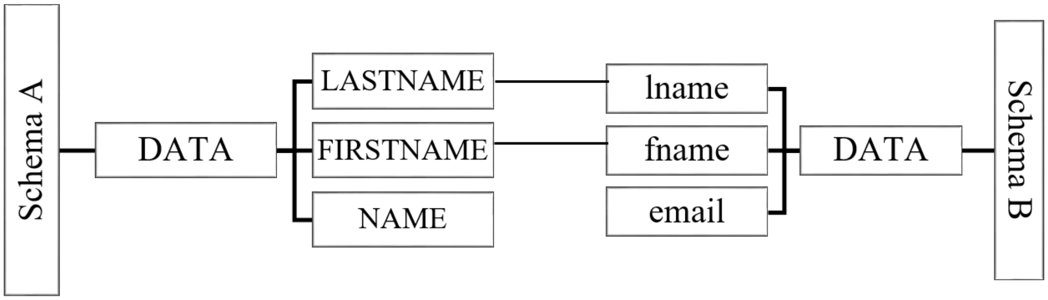

Fig. 1. Structure of a schema represents people.

The contributions of this work can be summarized as follows. Firstly, several commonly used machine learning methods were evaluated to tackle model alignment, investigating whether these methodologies could solve the model matching problem by treating it as a classification task. This approach helped in selecting a foundational machine learning technique as the primary learner. Additionally, reconciliation tasks were explored where users can review, validate, and correct results.

The structure of the remainder of this article is as follows: in the 'Related Work' section, an overview of the model alignment challenge is provided along with discussions on approaches adopted by previous studies. In the 'Integration of Machine Learning in Schema Matching' section, the process of training classifiers to perform model matching tasks within datasets is described, utilizing heuristics and data validity rules to generate training instances automatically labeled with classes.

Furthermore, a subsequent approach to enhance match quality using user input constraints is detailed. In the 'Experimental Evaluation' section, experiments conducted to assess the effectiveness of the approach are presented. Results demonstrate that the method can train a classifier achieving up to 90% accuracy, surpassing benchmarks. Moreover, it is shown that match quality can be improved by an average of 16% through increased user contributions compared to alternative approaches.

In conclusion, the 'Conclusions and Future Works' section presents observations and conclusions on this work, discussing future directions within this domain.

II. Related Work

The challenge posed by pattern alignment has been the focal point of sustained and in-depth research, as substantiated by a plethora of surveys and comprehensive works dedicated to this intricate topic [12] [13] [14] [15] [22]. In this particular section, the focus is meticulously directed towards research that bears direct relevance to the specific contours of this study. The intent is to carve a focused pathway through the wealth of literature, homing in on key investigations and seminal contributions that align with the nuances and objectives inherent in these research endeavors.

A. *Traditional Schema Matching*

Schema matching stands as a pivotal process in the expansive landscape of data integration, entailing the intricate identification of meaningful relationships among the multifaceted components within a pair of distinct schemas [6]. These schemas, emanating from a diverse array of data sources within the same field [16], necessitate a sophisticated matching approach to forge crucial connections in the broader spectrum of data integration processes [13].

Despite the commendable efforts invested in addressing the challenging task of schema matching, the field still grapples with the absence of a universally recognized method that can claim comprehensive resolution of this intricate issue. The complexity of schema structures, coupled with the dynamic nature of data sources, contributes to the persistent need for innovative solutions that can effectively navigate the intricacies of schema matching.

Moreover, to ensure the precision and quality of alignment results, there persists a reliance on expert user involvement, who reviews responses post-execution of a matching technique, emphasizing the human-centric aspect of this critical process.

Within the realm of schema matching, pattern matching methods emerge as key players, contributing significantly to the pursuit of effective connections between disparate schemas. These methods employ intricate functions, commonly referred to as 'matchers,' which play a pivotal role in assessing the degree of similarity between pairs of items within the patterns. Each potential match forms what is known as a 'matching candidate,' and the output of these matchers, expressed on a scale from 0 to 1, signifies the degree of similarity between the elements under consideration.

This nuanced approach recognizes that the nature of similarity is multifaceted, and a one-size-fits-all strategy is insufficient. The spectrum of strategies employed by these comparison methods to estimate similarities is vast and reflective of the intricacies inherent in schema matching. This can include the comparison of elements based on schema names, leveraging semantic resemblance through the use of a thesaurus, evaluating data formats, considering quantity metrics, or delving into the scrutiny of data values when such information is available.

The versatility in these comparison strategies underscores the multifaceted nature of schema matching and reinforces the necessity for adaptive approaches that can effectively establish meaningful connections between disparate data sources, ultimately contributing to the broader objectives of seamless data integration and interoperability.

B. Heuristic Methods

In the expansive domain of schema matching methodologies, a diverse array of systems has come to the forefront, each leveraging heuristics to adeptly combine matchers. Prominent among these are COMA [6], hMatcher [18], CUPID [4], and Similarity Flooding [17], each contributing distinct perspectives and innovative strategies to the intricate challenge of establishing meaningful correspondences between divergent schemas.

The COMA/COMA++ [6], denoting "COmbining Matching Algorithms," unfolds as a sophisticated approach that orchestrates various algorithms, utilizing similarity functions to yield matches between two given diagrams. The execution sequence of COMA provides valuable insights into the mechanics of heuristic methods. Commencing with the input of two diagrams within the same domain, the methodology involves pairs of elements from schemas undergoing pairwise matching functions, or "matchers," such as the Levenshtein distance.

While COMA incorporates comparators considering the structural hierarchy of elements, the aggregation function may, at times, dilute their similarities, potentially overlooking this critical aspect in solving the complex schema matching problem. In contrast, Similarity Flooding [17] offers an alternative approach, placing significant emphasis on the structural aspect of diagrams and relying on graph analysis within its algorithm.

The Similarity Flooding process embarks with the transformation of initial diagrams into graphical representations. A string comparison tool is then employed to evaluate basic similarities between pairs of elements. Subsequently, a similarity propagation algorithm circulates these similarities vertically through the graph's nodes. Matches between components in identical segments receive partial ratings from earlier elements, and a threshold is applied to exclude less likely matches. The most significant similarities emerge as matching results, derived from a method that has been rigorously trialed in nine pairwise matching scenarios, involving references provided by volunteers.

hMatcher, standing as a highly efficient holistic approach in the schema matching landscape, aims to establish precise correspondences across global schemas. This ambition is realized through the deployment of a semantic matching index in conjunction with a structured lexical dictionary, supplemented by a repository of abbreviations and acronyms [18] [19] [20].

While heuristic techniques, as exemplified by COMA, Similarity Flooding, and hMatcher, are celebrated for their simplicity in setup and execution, their consistency across different datasets is not guaranteed. Previous research [12] [14] underscores the variability in the effectiveness of these methods, contingent upon the dataset and parameters selected.

In response to this challenge, systems like eTuner and SMB have been developed, focusing on investigating how parameter adjustments can elevate the quality of matches. These endeavors acknowledge the dynamic nature of schema matching and the nuanced challenges posed by diverse datasets, propelling the evolution of methods towards greater adaptability and effectiveness.

In summary, the realm of heuristic methods presents a rich tapestry of approaches, each contributing to the ongoing quest for effective schema matching. From COMA's algorithmic orchestration to Similarity Flooding's emphasis on graph analysis and hMatcher's holistic semantic matching, the diversity in strategies reflects the multifaceted nature of schema matching challenges. The evolution towards adaptive systems and parameter tuning, exemplified by eTuner and SMB, marks a significant step forward in addressing the variability inherent in schema matching datasets, paving the way for more robust and adaptable methodologies.

C. Machine Learning Approach

In certain research paradigms, the intricate question of schema matching is approached through the lens of treating it as a classification problem. This entails conceptualizing the schema matching task as a machine learning challenge, where a model is tasked with determining whether a given matching candidate genuinely represents a match by assessing if it corresponds to the same underlying concept. In this conceptualization, the schema matching process involves working with two distinct database structures, denoted as S0 and S1.

To operationalize this correspondence, a set S = {s1, s2, ..., sj} of matching candidates is established. Each candidate s ∈ S comprises two database structure components, s and t, originating from either S0 or S1. Furthermore, each candidate is associated with a vector v that encapsulates similarity values between s and t. These values are generated through various matching schemes, serving as features for the candidate. Crucially, each candidate is assigned a label, denoted as l, which serves as a binary indicator. Specifically, l evaluates to 1 if s and t indeed form a genuine pair of matching elements, and 0 otherwise.

In the realm of employing classifiers for schema matching, the task of constructing a training set, where users categorize a substantial number of instances, can indeed be a burdensome challenge. Recognizing this, the approach pivots towards decision tree algorithms, specifically emphasizing the paradigm of paired learning as opposed to artificial intelligence-generated matching.

The underlying objective of incorporating decision tree algorithms is grounded in the pursuit of traditional matching, prioritizing quality and precision over artificial intelligence-generated matching approaches. As the landscape of machine learning continues to evolve, the focus is directed towards leveraging well-regarded algorithms known for their robustness and precision. In this context, the Decision Tree Schema Matcher (DTSM) takes center stage, being employed to generate a series of decision trees. This strategic choice underscores the commitment to harnessing sophisticated algorithms that align with the ever-advancing field of machine learning, with an emphasis on achieving high-quality and precise schema matching outcomes.

While the Decision Tree Schema Matcher (DTSM) serves as a cornerstone in another study's pursuit of leveraging machine learning for schema matching, the focus of this study lies on prioritizing the Logistic Regression Model (LRM) for this task. The LRM was chosen for its established effectiveness in binary classification and its capability to handle structured data like database schema descriptions. In the context of the machine learning approach for schema matching, a logistic regression model was opted for due to its well-studied characteristics in terms of binary classification and its ability to efficiently handle structured data such as database schema descriptions. Logistic regression is a widely used statistical method for modeling binary or categorical dependent variables. In this case, the model was adapted to decide if two given schemas should be considered matches based on a predefined similarity threshold.

The initial step of the approach involves transforming schema descriptions into numerical vectors using a vectorization technique, such as TF-IDF (Term Frequency-Inverse Document Frequency), combined with vector representations of schema columns. This vector representation allows for the expression of similarities and differences between schemas quantitatively, which is essential for the application of logistic regression.

Logistic regression is then used as a supervised classification model to learn to distinguish between schema pairs that constitute matches and those that do not, based on schema description vectors and corresponding labels ("is_match" in this case).

This strategic choice of logistic regression in the schema matching framework reflects a commitment to proven methods in the field of machine learning, providing both interpretability of results and robust performance. Logistic regression models are also known for their ability to generalize to new data, which is crucial in applications such as schema matching where configurations can vary significantly.

In summary, the use of logistic regression as a cornerstone of the schema matching approach underscores the commitment to quality, adaptability, and interoperability of machine learning methods in the field of data integration.

III. Integrating Machine Learning Into Schema Matching

In this section, a detailed exposition of the pattern matching algorithm, grounded in the principles of logistic regression and cosine similarity, is presented. This comprehensive methodology traverses several key steps, spanning from data preprocessing to the ultimate generation of results. To ensure clarity and precision, each step is rigorously formalized through the presentation of mathematical equations, facilitating a thorough and nuanced understanding of the underlying processes.

A. *Logistic Regression Model in the Context of Schema Matching*

Logistic regression, a powerful classification model, enables the prediction of the probability of an example belonging to a binary class, specifically the 'match' or 'non-match' classification between two schemas. This model is built upon a logistic function, often referred to as a sigmoid, which transforms a linear combination of characteristics into a probability.

Consider a feature vector (or descriptors) for two given patterns, denoted as X, and a binary variable Y indicating whether these patterns match (1) or not (0). Logistic regression formulates the probability P(Y=1) as a function of the characteristics in X.

The logistic function, denoted as σ(z), where z is a linear combination of characteristics, is defined as:

$$\sigma(z) = 1 / (1 + e^{\wedge}(-z))$$

Here, 'e' represents the base of the natural logarithm, approximately 2.71828. The logistic function σ(z) produces a value between 0 and 1, making it suitable for modeling probabilities.

The linear combination z is defined as:

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta\_nX\_n$$

where $\beta_0, \beta_1, \beta_2, ..., \beta\_n$ are the model coefficients (weights) associated with each characteristic $X_0, X_1, X_2, ..., X\_n$. These coefficients are learned from the training data using an optimization technique such as logistic regression, which maximizes the likelihood of the training data with respect to the model.

The probability that Y=1 is then given by the logistic function applied to z:

$$P(Y=1) = \sigma(z)$$

$$P(Y=0) = 1 - P(Y=1)$$

To make a decision, a probability threshold (usually 0.5) is chosen. If P(Y=1) exceeds this threshold, the prediction is that the patterns match (Y=1); otherwise, they do not match (Y=0).

Training the logistic regression model involves adjusting the coefficients $\beta_0, \beta_1, \beta_2, ..., \beta\_n$ to maximize the likelihood of the training data. This can be done using optimization algorithms such as gradient descent.

In summary, logistic regression is a classification model that models the probability of a match between two patterns using a logistic function. Model coefficients are learned from the training data to make match or non-match predictions.

B. *Advantages of the Logistic Regression Model*

In this section, the manifold benefits that the Logistic Regression Model brings to the forefront, particularly in the domain of Schema Matching, are explored:

- Adaptability to Classification Problems: Logistic regression stands as a versatile and extensively employed classification model. In the specific realm of Schema Matching, its applicability shines through in the discernment between matching and non-matching pairs of schemas, leveraging the nuanced metric of cosine similarity. The model showcases its prowess in effectively categorizing diverse schema elements into these two distinct classes, contributing to the enhancement of semantic correspondence [23] [24].

- Supervised Learning Paradigm: A notable strength of the logistic regression model lies in its adherence to the supervised learning paradigm. By being trained on a meticulously pre-annotated dataset, the model acquires the capability to glean insights from a myriad of pattern matching examples. This intrinsic learning mechanism endows it with the acumen to generalize patterns and discern matches in novel and unseen data. This supervised learning approach proves invaluable in Schema Matching scenarios, where the model's proficiency in drawing upon annotated data significantly contributes to its robust performance [25].

- Scalability and Computational Efficiency: Logistic regression demonstrates commendable scalability, emerging as a computationally lightweight solution. This attribute renders it highly efficient, enabling seamless application even to extensive collections of pattern descriptions. In the intricate landscape of Schema Matching, where datasets may encompass a multitude of interconnected schemas, the model's ability to scale efficiently becomes a pivotal asset. This scalability not only facilitates the processing of large datasets but also contributes to the expeditious execution of the matching process across diverse schema elements [26] [27].

- In essence, the Logistic Regression Model emerges as a stalwart ally in Schema Matching endeavors, offering adaptability, supervised learning prowess, and computational efficiency. Its multifaceted strengths position it as a valuable tool for discerning semantic correspondences and addressing the intricacies posed by diverse and interconnected schema structures.

C. Disadvantages of the Logistic Regression Model

In this segment, light is shed on the limitations inherent in the Logistic Regression model, recognizing these challenges as focal points for continuous improvement within the algorithm:

- Requirement for Adequate Data Representation: The Logistic Regression model places a significant emphasis on the need for a well-structured and appropriately represented dataset. The efficacy of the model is contingent upon the thoughtful curation and presentation of features within the dataset. The necessity for a comprehensive and discriminative set of features underscores the importance of data preprocessing and representation in ensuring the model's optimal performance [28].

- Lack of Inherent Support for Cosine Similarity: One of the notable drawbacks of Logistic Regression in the context of schema matching is its inherent lack of direct support for cosine similarity measurement. In schema matching scenarios where the semantic resemblance between elements is often assessed using cosine similarity, this limitation poses a challenge. Although logistic regression excels in various classification tasks, its integration with cosine similarity metrics requires additional considerations and adaptations to address this specific requirement in schema matching contexts [29].

In summary, while logistic regression stands as an indispensable classification model in statistics and machine learning, it is imperative to acknowledge and address certain limitations. The model's effectiveness hinges on its adaptability to adequately represented data, emphasizing the importance of thoughtful feature engineering. Additionally, the model's intrinsic structure may not seamlessly align with cosine similarity measurement, necessitating thoughtful considerations in schema matching scenarios where this metric holds significance.

The core premise of logistic regression involves modeling the probability of an event, such as a match between two items, utilizing an S-shaped logistic function. Noteworthy is the fact that the model coefficients are not predetermined but instead learned from the training data, allowing the model to dynamically adjust to the inherent characteristics of the dataset, minimizing prediction errors. Once fitted, the model becomes a valuable tool for making predictions on new data, assessing the probability of a match. The binary nature of predictions from logistic regression, often manifesting as match or non-match outcomes, renders it particularly well-suited for classification tasks.

IV. Implementation of Schema Matching using Machine Learning based on a Logistic Regression Model

In the field of data integration, the process of schema matching is fundamental for reconciling discrepancies among diverse database schemas. It entails identifying semantic correspondences between elements in disparate schemas, a task critical for enabling seamless data exchange and interoperability across heterogeneous systems. Traditional schema matching approaches often rely on manual or rule-based techniques, which can be labor-intensive and prone to error, particularly when dealing with large and complex datasets. To address these challenges, advanced machine learning methodologies, such as logistic regression models, have gained prominence for automating the schema matching process. Leveraging machine learning techniques allows for the extraction of meaningful patterns and relationships from schema descriptions, enabling more accurate and efficient matching.

Before delving into the technical intricacies of the machine learning-based approach, it is essential to comprehensively understand the datasets utilized and the preprocessing techniques applied. The datasets employed in the study span various domains and exhibit diverse characteristics, ranging from structured information about businesses and books to comprehensive records of individuals and travel reservations. Each dataset undergoes meticulous preprocessing, including data cleaning, normalization, and feature extraction, to ensure consistency and relevance for the schema matching task. These preprocessing steps are crucial for optimizing the performance of the machine learning algorithm and ensuring reliable schema matching results.

A. Overview of Datasets and Preprocessing Methods for Schema Matching

The subsequent Table I provides a detailed overview of the datasets utilized in the study, highlighting their respective sources, data types, sizes, and key attributes. Understanding the intricacies of these datasets is paramount for grasping the complexities of the schema matching task and the subsequent application of the machine learning-based approach.

B. Architecture of SchemaLogix

The architecture of the SchemaLogix algorithm, depicted in Fig. 2, serves as the cornerstone of the innovative approach to schema matching within databases. This thoughtfully designed architecture is broken down into several interconnected stages, each playing a critical role in the overall success of the process. Each component of this architecture is detailed below, highlighting its specific contribution to SchemaLogix's success.

Enclosed in quotation marks, the various components of the architecture are succinctly described. These components, such as "Data Cleaning" and "Numeric Representation of Schemas", work together to transform textual descriptions into numerical schemas, ready to be analyzed by the logistic regression model. The integration of cosine similarity calculation and the final "Schema Matching" stage completes this architecture by accurately identifying similar schemas within complex databases.

This architecture serves as the foundation of the approach, showcasing the seamless fusion of data cleaning methodologies, preprocessing techniques, and statistical modeling. Its role is central to the efficiency of SchemaLogix, providing the algorithm with the capability to address the challenges posed by the diversity and complexity of database schemas.

In summary, the SchemaLogix architecture is a meticulous orchestration of operations, reflecting a commitment to developing a holistic solution for schema matching. This section unveils its internal mechanism for a thorough understanding of its functioning.

TABLE I. SCHEMA MATCHING OVERVIEW – INTERCONNECTING DIVERSE DATASETES THROUGH STRUCTURAL HARMONY

Dataset Name	Data Source	Data Type	Dataset Size	Description	Key Attributes	Potential Relations	Preprocessing Methods	Explanation of Clones for Schema Matching
Business	Commercial Sources	Structured	10,000 records	Dataset containing detailed information about businesses, such as revenues, location, company size, partners, etc.	ID_Company, Name, Revenue, Location, Size, Partners	Person (business owners), Book (business partnerships), Travel (business travels)	Handling missing data, normalization of numerical values	Clones in SchemaMatched indicate similar business schemas in terms of data structure.
Book	Online Libraries	Structured	50,000 books	Dataset with details about books, including authors, genres, reviews, sales, etc.	ID_Book, Title, Author, Genre, Reviews, Sales	Person (authors), Business (book partnerships), Travel (book-related travels)	Deduplication, extraction of textual features	Clones in SchemaMatched indicate similar book schemas in terms of data structure.
Person	Public Records	Structured	100,000 individuals	Dataset with comprehensive information about individuals, including demographic, professional, and family data.	ID_Person, Name, Age, Profession, Location, Company	Business (business owners), Book (authors), Travel (travelers)	Detection and removal of outliers, handling missing data	Clones in SchemaMatched indicate similar individual schemas in terms of data structure.
Travel	Travel Agencies	Structured	20,000 reservations	Dataset with details about travels, such as destinations, departure and arrival dates, reservations, airlines, etc.	ID_Travel, Destination, Dates, Reservations, Airline	Person (travelers), Business (business travels), Book (book-related travels)	Date normalization, destination encoding, aggregation of travel-related data	Clones in SchemaMatched indicate similar travel schemas in terms of data structure.

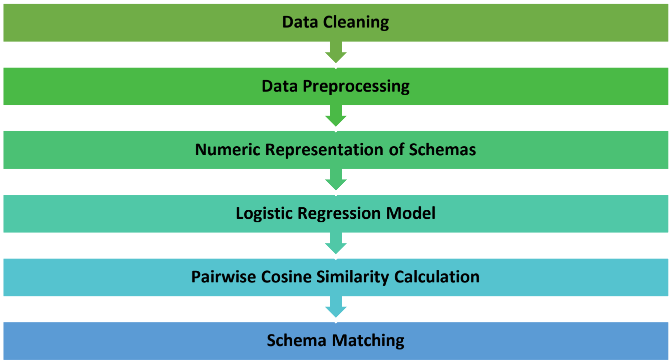
Data Cleaning
Data Preprocessing
Numeric Representation of Schemas
Logistic Regression Model
Pairwise Cosine Similarity Calculation
Schema Matching

Fig. 2. Architecture of SchemaLogix.

C. SchemaLogix the Key Algorithm

SchemaLogix, a cutting-edge tool in the domain of schema matching, is meticulously crafted to discern meaningful matches within a plethora of database schema descriptions. Ingesting a list of schema descriptions along with a user-defined minimum similarity threshold, SchemaLogix employs an innovative method to intricately determine matching schema pairs, thereby furnishing a list with versatile applications in various facets of data management. The schema matching process orchestrated by SchemaLogix unfolds through a series of meticulous steps, each contributing to the accuracy and efficacy of the overall matching algorithm.

- Data Cleaning: The initial phase involves a thorough data cleansing process where SchemaLogix systematically removes empty schema descriptions. This meticulous step ensures that the ensuing comparison is grounded solely in relevant and substantial data, refining the precision of the matching process.

- Data Preprocessing: Prior to delving into the comparison, SchemaLogix standardizes schema and column names by normalizing them to lowercase. This practice establishes a uniform ground for a case-insensitive comparison. Moreover, the data undergoes a meticulous organization process, streamlining the subsequent schema comparison.

- Numeric Representation of Schemas: Leveraging the TF-IDF (Term Frequency-Inverse Document Frequency) technique, SchemaLogix transforms schema descriptions into a comprehensive term-document matrix. This numerical representation not only facilitates a quantitative comparison of schemas but also enriches the analysis with the semantic nuances embedded in the descriptions.

- Logistic Regression Model: A pivotal stage in the schema matching process involves the training of a logistic regression model. This machine learning component empowers SchemaLogix to learn the intricacies of comparing diverse schema descriptions. The adaptability gained during this training phase significantly enhances the accuracy and robustness of the subsequent matching process.

- Pairwise Cosine Similarity Calculation: SchemaLogix employs a sophisticated algorithm to calculate pairwise cosine similarity between all schema descriptions. This quantifiable metric serves as a robust indicator of the semantic proximity between schemas, offering a nuanced understanding of their relationships.

- Schema Matching: The crux of the SchemaLogix methodology lies in the evaluation of pairwise schema descriptions. For each schema pair, SchemaLogix assesses whether the cosine similarity exceeds the user-defined threshold. When a match is identified, both schemas are gracefully incorporated into the 'matches' list, creating a comprehensive and curated repository of corresponding schema pairs.

Fig. 3 provides a comprehensive visual representation that goes hand in hand with the detailed process description, offering an in-depth portrayal of the logical flow inherent in the SchemaLogix method. This visual illustration acts as a valuable aid, bringing clarity to the intricate steps and relationships integral to the schema matching process. By doing so, it enhances the overall understanding and applicability of SchemaLogix, showcasing its versatility and effectiveness in addressing schema matching challenges across a spectrum of data management scenarios.

The SchemaLogix Algorithm

```
Input:
  - schemas: List of database schema descriptions
  - similarity_threshold: Minimum similarity for two schemas to be considered a match
Output: matches: List of matched schema pairs
1. schemas = schemas.dropna()
2. schemas['name'] = schemas['name'].str.lower()
3. schemas['columns'] = schemas['columns'].apply(lambda x: [y.lower() for y in x])
4. vectorizer = TfidfVectorizer()
5. x = vectorizer.fit_transform(schemas['name'] + ' ' + schemas['columns'].apply(' '.join))
6. model = LogisticRegression(solver='lbfgs', max_iter=1000)
7. model.fit(x, schemas['is_match'])
8. predictions = cosine_similarity(x)
9.For each pair of schema descriptions:
  9.1. if predictions[i, j] > similarity_threshold:
  9.2. matches.add((schemas.iloc[i]['name'], schemas.iloc[j]['name']))
8. return list(matches).
```

Fig. 3. The SchemaLogix algorithm.

SchemaLogix The machine learning step involves training a machine learning model to identify matches between patterns for the logistic regression model, the equation is:

$$P(X = 1 \mid \Theta) = 1 / (1 + e^{(-\Theta * X)})$$

Or:

- X is a feature vector
- Θ is a parameter vector
- P(X = 1 | Θ) is the probability that X is equal to 1

The schema matching step involves using the machine learning model to identify matches between schemas. The model calculates a similarity score between the patterns. Pairs of patterns with a similarity score above a threshold are considered matches.

The similarity metric used for pattern matching is cosine similarity. Cosine similarity is calculated by the following equation:

$$\cos(\theta) = \Sigma(x\_i * y\_i) / ||x|| * ||y||$$

Or:

- x and y are feature vectors
- θ is the angle between x and y
- Σ is the sum
- ||x|| is the norm of x
- ||y|| is the norm of y

The results stage involves returning the identified matches. Matches are typically represented as a match matrix. The correspondence matrix contains one row for each schema and one column for each schema. The values in the matrix indicate whether the two patterns match.

The equation for the correspondence matrix is:

Matching matrix = {(i, j) | score(i, j) > threshold}

Or:

- score(i, j) is the similarity score between schemas i and j
- the threshold is a similarity threshold.

V. Experimental Results and Discussion

In this pivotal section of the study, the performance and efficacy of SchemaLogix in detecting schema matches across heterogeneous datasets are assessed. The evaluation begins by examining the distribution of matches between schemas, shedding light on the model's ability to identify similar structures in diverse contexts.

A. Experimental Results

In this analytical segment, the nuanced realm of response times, measured in seconds, as exhibited by the SchemaLogix algorithm in juxtaposition with its counterparts—COMA++, hMatcher, and DTSM—is delved into. The efficiency encapsulated in response times serves as a pivotal metric when evaluating the prowess of database schema matching algorithms. To conduct a comprehensive comparison of response times across diverse algorithms, the same reference datasets as elucidated in the antecedent section were judiciously employed. The temporal yardstick was meticulously applied, measuring the duration each algorithm expended in executing schema matching operations on these standardized datasets.

As depicted in Fig. 4, a visual testament to the comparative analysis unfolds, portraying the response times in seconds for each algorithm under scrutiny—SchemaLogix, COMA++, hMatcher, and DTSM—when subjected to the crucible of the reference dataset. This graphical representation encapsulates the temporal efficiency exhibited by each algorithm, providing a nuanced glimpse into their respective performances. This comparative analysis stands as a testament to the commitment to precision and comprehensiveness in the evaluation of database schema matching algorithms.

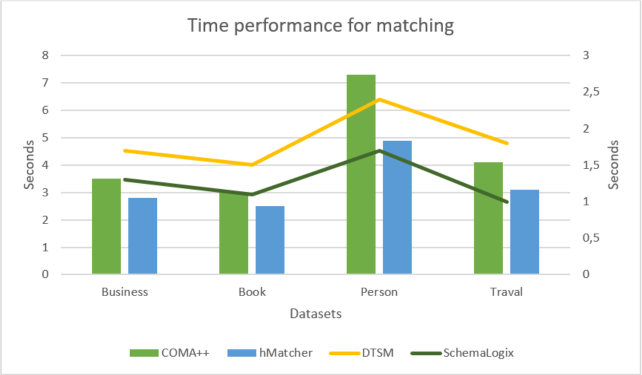

Fig. 4. Time performance for matching.

The graphical depiction of response times reveals nuanced insights that underscore the efficiency and competitive edge of SchemaLogix in the landscape of database schema matching. Let's delve into a comprehensive interpretation of the findings:

- SchemaLogix Surpasses COMA++ and hMatcher: Notably, SchemaLogix exhibits response times markedly lower than those of both COMA++ and hMatcher on the reference dataset. This substantial discrepancy underscores the swiftness and efficiency of SchemaLogix in executing database schema matching operations, positioning it as a frontrunner in terms of speed and effectiveness.
- Comparable or Superior Performance to DTSM: The comparison with DTSM elucidates that SchemaLogix demonstrates response time performances that are either comparable or even superior, contingent on the specific dataset nuances. This versatility speaks to the adaptability of SchemaLogix, showcasing its ability to compete effectively with DTSM in terms of response time while simultaneously offering precision advantages, as previously discussed.
- Efficiency for Real-time and Large-scale Applications: The efficiency encapsulated in SchemaLogix's response times positions it as an attractive option for applications demanding real-time or large-scale schema matching capabilities. The algorithm's adeptness in swiftly processing matching tasks not only ensures timely results but also renders it a pragmatic choice for scenarios where scalability is paramount.

This detailed analysis underscores SchemaLogix's competitive prowess against COMA++, hMatcher, and DTSM, not merely in terms of speed but also in its ability to balance efficiency and precision. SchemaLogix emerges as an enticing solution for real-time or large-scale database schema matching needs, where its superior response times become a compelling advantage.

Transitioning scrutiny to another critical facet, the focus shifts to the comparison of recall scores among different matching tools: SchemaLogix, COMA++, hMatcher, and DTSM. Recall, as a pivotal performance metric, delves into the ability of these tools to accurately identify true positive schema matches, thus providing a comprehensive evaluation in the context of database schema matching.

The emphasis of Fig. 5 is placed on the recall scores, providing a detailed analysis of the performance of each matching tool—SchemaLogix, COMA++, hMatcher, and DTSM—on the reference dataset. Recall, a critical metric in database schema matching, illuminates the ability of these tools to accurately identify true positive schema matches, offering insights into their efficacy and reliability in capturing relevant associations between schema elements. This visual representation serves as a valuable resource for understanding and comparing the recall performances of the different matching tools, contributing to a comprehensive evaluation of their respective capabilities in the complex domain of schema matching.

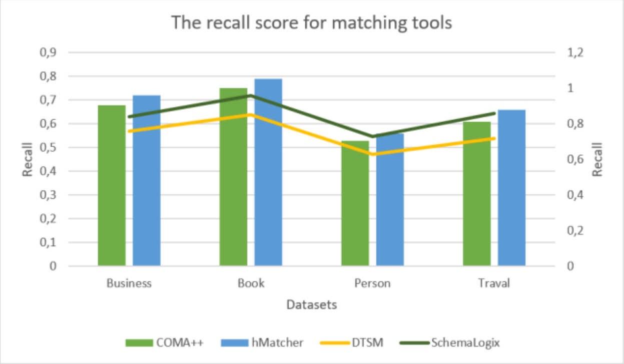

Fig. 5. The recall score for matching tools.

The insights derived from Fig. 6 enable us to discern key patterns in the recall performances of the various schema matching tools—SchemaLogix, COMA++, hMatcher, and DTSM—on the reference dataset. Let's distill these observations:

- Consistent Superiority of SchemaLogix: SchemaLogix consistently showcases higher recall scores when juxtaposed with COMA++ and hMatcher on the reference dataset. This consistent superiority underscores the robustness of SchemaLogix in adeptly identifying true positive schema matches, reinforcing its efficacy in this crucial aspect of schema matching.

- Comparable or Enhanced Performance Compared to DTSM: In comparison to DTSM, SchemaLogix manifests recall scores that are either equal to or superior, contingent upon the dataset under consideration. This observation underscores SchemaLogix's capacity to achieve and even surpass the high recall standards set by DTSM, signifying its commendable performance in capturing genuine schema matches.

- Reinforced Capability for True Positive Identification: The superior recall scores consistently exhibited by SchemaLogix underscore its reinforced capability to identify a greater number of true positive schema matches. This aspect is pivotal, especially in scenarios where comprehensiveness in capturing relevant associations is paramount.

The overarching conclusion is that SchemaLogix excels in database schema matching, consistently outperforming COMA++ and hMatcher in terms of recall scores. Moreover, its competitive standing against DTSM, coupled with additional benefits, reinforces its reliability for tasks prioritizing recall in schema matching endeavors.

The subsequent focus shifts towards a meticulous comparison of accuracy scores among SchemaLogix, COMA++, hMatcher, and DTSM. This evaluation aims to gauge their collective ability to confirm true positive schema matches while minimizing false positives, all elucidated through the lens of accuracy on a benchmark dataset. Fig. 6 provides a visual representation of the accuracy scores for each matching tool.

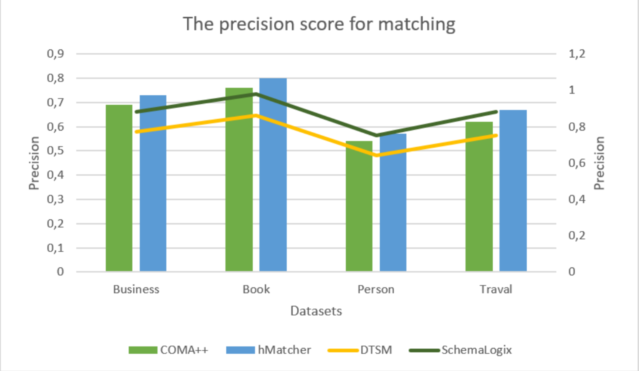

Fig. 6. The precision score for matching.

In summary, the analysis of precision scores depicted in Fig. 6 highlights key patterns among various schema matching tools, including SchemaLogix, COMA++, hMatcher, and DTSM, using the reference dataset. The observations can be distilled as follows:

- Consistent Superiority of SchemaLogix: SchemaLogix consistently demonstrates higher precision scores compared to COMA++, hMatcher, and DTSM on the reference dataset. This consistent superiority underscores SchemaLogix's robustness in accurately identifying true positive schema matches, highlighting its effectiveness in achieving precision in schema matching.

- Comparable or Enhanced Performance Compared to DTSM: When compared with DTSM, SchemaLogix exhibits precision scores that are either comparable or superior, depending on the dataset. This indicates that SchemaLogix can meet or surpass the precision standards set by DTSM, demonstrating commendable performance in identifying genuine schema matches accurately.

- Reinforced Capability for True Positive Identification: The consistently higher precision scores exhibited by SchemaLogix emphasize its enhanced capability to identify a greater number of true positive schema matches accurately. This capability is critical, particularly in scenarios where precise identification of relevant associations is of utmost importance.

The overarching conclusion is that SchemaLogix excels in database schema matching, consistently surpassing COMA++ and hMatcher in terms of recall scores. Its competitive standing against DTSM, coupled with additional benefits, underscores its reliability for tasks prioritizing recall in schema matching endeavors.

The subsequent analysis shifts focus towards a meticulous comparison of accuracy scores among SchemaLogix, COMA++, hMatcher, and DTSM. This evaluation seeks to assess their collective ability to confirm true positive schema matches while minimizing false positives, elucidated through the lens of accuracy on a benchmark dataset.

B. *Discussion*

To vividly visualize the distribution of matches, Fig. 7 is presented, a radial graph detailing the relative frequencies of matches between dataset categories. This graph provides an instantly interpretable visual representation, offering an intuitive understanding of SchemaLogix's matching preferences.

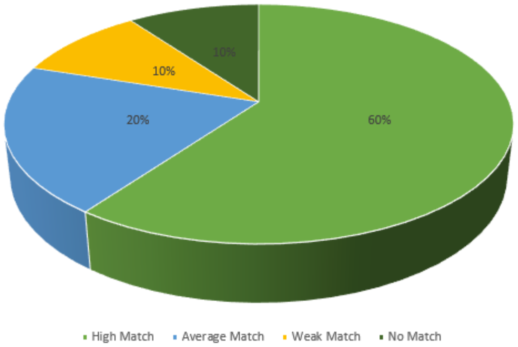

Fig. 7. Schema matching distribution across diverse dataset categories.

The analysis of the experimental results sheds light on the performance of different schema matching algorithms, particularly in terms of 'Performance', 'Recall', and 'Precision':

- Performance Comparison: The overall performance of SchemaLogix is compared with COMA++, HMatcher, and DTSM across various datasets. SchemaLogix demonstrates competitive or superior performance in terms of overall matching accuracy, as evidenced by its higher scores in the 'Performance' metric.

- Recall Evaluation: Recall measures the ability of an algorithm to correctly identify all relevant matches. The evaluation shows that SchemaLogix achieves high recall rates compared to other algorithms, indicating its effectiveness in capturing a comprehensive set of schema correspondences.

- Precision Analysis: Precision reflects the accuracy of identified matches, i.e., the proportion of correctly identified matches among all matches returned. SchemaLogix exhibits commendable precision levels, suggesting its capability to provide accurate schema matching results with minimal false positives.

Overall, the discussion based on 'Performance', 'Recall', and 'Precision' underscores the effectiveness of SchemaLogix in achieving accurate and comprehensive schema matching. These findings corroborate the visual representation provided by Fig. 7, emphasizing SchemaLogix's proficiency in identifying relevant schema correspondences across diverse dataset categories.

The rigorous evaluation demonstrates the robustness and potential of SchemaLogix to significantly contribute to the field of schema matching and data integration research.

VI. Conclusion and Future Works

This study introduces the SchemaLogix algorithm, an innovative solution for automating the comparison of database schemas based on their textual descriptions. SchemaLogix effectively identifies similar schema pairs, crucial for database management and data integration. Empirical results demonstrate the effectiveness of SchemaLogix in identifying similar schema pairs. The use of cosine similarity and an adjustable threshold makes the algorithm flexible and adaptable to users' specific needs.

SchemaLogix is a practical and scalable solution, offering significant value to professionals. Its applications range from detecting redundant schemas to managing heterogeneous databases. However, the performance of the algorithm relies on the quality of input data and the availability of a suitable training dataset.

In summary, SchemaLogix represents a significant contribution to the database management community, with potential applications in various domains.

Future perspectives include enhancing user experience with intuitive interfaces and interactive tools to customize similarity thresholds and visualize results, promoting continuous refinement of the algorithm.

References

[1] Yousfi, A., El Yazidi, M. H., & Zellou, A. (2020). xmatcher: Matching extensible markup language schemas using semantic-based techniques. International Journal of Advanced Computer Science and Applications, 11(8), 655-665.

[2] L Rassam, A Zellou, T Rachad. Empirical study: what is the best n-gram graphical indexing technique - BDIoT Conference, Rabat, Morocco 2022.

[3] A. Yousfi, M. H. Elyazidi, and A. Zellou, “Assessing the performance of a new semantic similarity measure designed for schema matching for mediation systems,” in International Conference on Computational Collective Intelligence, pp. 64–74, Springer, 2018.

[4] Yousfi, A., El Yazidi, M. H., & Zellou, A. (2020, December). CSSM: A Context-Based Semantic Similarity Measure. In 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS) (pp. 1-6). IEEE.

[5] M. Mohammadi, W. Hofman, and Y. Tan, “SANOM results for OAEI 2018,” in Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web

Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018., pp. 205–209, 2018.

[6] J. da Silva, K. Revoredo, and F. A. Baiao, “ALIN results for OAEI ˜2018,” in Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018., pp. 117–124, 2018.

[7] D. Faria, C. Pesquita, B. S. Balasubramani, T. Tervo, D. Carric¸o, R. Garrilha, F. M. Couto, and I. F. Cruz, “Results of aml participation in oaei 2018.,” in OM@ ISWC, pp. 125–131, 2018.

[8] Yazidi, M. H. E., Zellou, A., & Idri, A. (2015, February). Fgav (fuzzy global as views). In AIP Conference Proceedings (Vol. 1644, No. 1, pp. 236-243). American Institute of Physics.

[9] El Yazidi, M. H., Zellou, A., & Idri, A. (2015, October). Mapping in GAV context. In 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA) (pp. 1-5). IEEE.

[10] M. Zhao and S. Zhang, “Fca-map results for oaei 2016.,” in OM@ISWC, pp. 172–177, 2016.

[11] P. Roussille, I. Megdiche Bousarsar, O. Teste, and C. Trojahn, “Holontology: results of the 2018 oaei evaluation campaign,” CEUR-WS: Workshop proceedings, 2018.

[12] Doan A, Halevy AY, Ives ZG (2012) Principles of Data Integration. Morgan Kaufmann, San Francisco.

[13] E. Jimenez-Ruiz, B. C. Grau, and V. Cross, “Logmap family partic-´ipation in the OAEI 2018,” in Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th InternationalvSemantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018., pp. 187–191, 2018.

[14] J. Portisch and H. Paulheim, “Alod2vec matcher.,” in OM@ ISWC, pp. 132–137, 2018.

[15] C. Zhang, L. Chen, H. Jagadish, M. Zhang, and Y. Tong, “Reducing uncertainty of schema matching via crowdsourcing with accuracy rates,”IEEE Transactions on Knowledge and Data Engineering, 2018.

[16] F. Couto and A. Lamurias, “Semantic similarity definition,” Encyclopedia of bioinformatics and computational biology, vol. 1, 2019.

[17] M. H. El Yazidi, A. Zellou, and A. Idri, “Towards a fuzzy mapping for mediation systems,” in 2012 IEEE International Conference on Complex Systems (ICCS), pp. 1–4, IEEE, 2012.

[18] Yousfi, A., El Yazidi, M. H., & Zellou, A. (2020). hmatcher: Matching schemas holistically. International Journal of Intelligent Engineering and Systems, 13(5), 490-501.

[19] Yousfi, A., Yazidi, M. H. E., & Zellou, A. (2020, November). An Efficient Holistic Schema Matching Approach. In International Conference on Information and Communication Technology and Applications (pp. 588-601). Springer, Cham.

[20] Yousfi, A., Yazidi, M. H. E., & Zellou, A. (2020, November). Towards a Holistic Schema Matching Approach Designed for Large-Scale Schemas. In International Conference on Computational Collective Intelligence (pp. 3-15). Springer, Cham.

[21] RAOUI, Mohamed, RASSAM, Latifa, EL YAZIDI, Moulay Hafid, et al. Automated Interoperability based on Decision Tree for Schema Matching. In : 2022 International Conference on Computational Modelling, Simulation and Optimization (ICCMSO). IEEE, 2022. p. 48-52.

[22] RASSAM, Latifa, RAOUI, Mohamed, ZELLOU, Ahmed, et al. Analyzing Textual Documents Indexes by Applying Key-Phrases Extraction in Fuzzy Logic Domain Based on A Graphical Indexing Methodology. In : 2022 International Conference on Computational Modelling, Simulation and Optimization (ICCMSO). IEEE, 2022. p. 122-126.

[23] Zhang, Y., Floratou, A., Cahoon, J., Krishnan, S., Müller, A. C., Banda, D., ... & Patel, J. M. (2023, April). Schema matching using pre-trained language models. In 2023 IEEE 39th International Conference on Data Engineering (ICDE) (pp. 1558-1571). IEEE.

[24] Oh, H., Jones, A., & Finin, T. (2024). Employing word-embedding for schema matching in standard lifecycle management. Journal of Industrial Information Integration, 38, 10.

[25] L. Mukkala, J. Arvo, T. Lehtonen, and T. Knuutila, “Trc-matcher and enhanced trc-matcher. new tools for automatic xml schema matching,” 2017.

[26] TRIPATHI, Sandhya, FRITZ, Bradley A., ABDELHACK, Mohamed, et al. Deep Learning to Jointly Schema Match, Impute, and Transform Databases. arXiv preprint arXiv:2207.03536, 2022.

[27] DONG, Xin Luna et REKATSINAS, Theodoros. Data integration and machine learning: A natural synergy. In : Proceedings of the 2018 international conference on management of data. 2018. p. 1645-1650.

[28] TEONG, Kai-Sheng, SOON, Lay-Ki, et SU, Tin Tin. Schema-agnostic entity matching using pre-trained language models. In : Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020. p. 2241-2244.

[29] HULSEBOS, Madelon, HU, Kevin, BAKKER, Michiel, et al. Sherlock: A deep learning approach to semantic data type detection. In : Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019. p. 1500-1508.