Collaborative Scoping: Self-Supervised Linkability Assessment for Schema Matching

Leonard Traeger
University of Maryland,
Baltimore County, USA
leonard.traeger@umbc.edu

Andreas Behrend
Technical University of Cologne
Cologne, Germany
andreas.behrend@th-koeln.de

George Karabatis
University of Maryland,
Baltimore County, USA
georgek@umbc.edu

Abstract

Schema matching, a critical task for integrating data from diverse sources, seeks to identify correspondences between attributes and tables across different schemas. In multi-source schema scenarios that are highly heterogeneous in volume, schema design, and domain, we observe that only a fraction of tables and attributes are linkable, while many unlinkable ones occupy the search space and worsen the linkage generation. Therefore, we focus on semantic *schema linkability assessment* as a quality pre-processing phase that identifies *linkable* tables and attributes, and prunes unlinkable ones at the same time. This paper introduces *collaborative scoping*, a self-supervised and distributed encoder-decoder framework enabling local schemas to independently scope *streamlined schemas*. Experiments show that collaborative scoping is robust, more efficient, and more effective compared to scoping baselines. This is true for matching scenarios on multi-source schemas that are relatively homogeneous (domain-specific) or highly heterogeneous. In subsequent matching experiments, algorithms that use the *streamlined schemas* as input improve their linkage quality (precision) by up to +80% and F1-measure by up to +20% while still remaining efficient.

Keywords

Data Integration, Schema Matching, Outlier Analysis

1 Introduction

Data is being generated at an unprecedented rate, increasingly migrating to cloud platforms and online data marketplaces. Organizations with historically grown data can improve their data-driven decision-making (e.g., B2B or M&A), without the need for materialized integration [9]. A necessary first step and precursor to integration is identifying the correct linkages between the elements that represent similar semantics across the relevant systems.

For relational systems, tables and attributes of one schema need to be correctly matched to those of another schema. Source-to-target matching is already challenging [2, 17, 37, 38, 49]. When more than two schemas are involved, this process is referred to as *Multi-Source Schema Matching*, and it is known to pose further challenges on efficiency and effectiveness [8, 36, 50]. Matching multiple schemas is necessary for integration purposes, yet no global alignment standard exists [6, 42]. Therefore, organizations typically expose only their metadata in order to identify synergies with other organizations or data markets. However, the underlying data usually remains private and is only available for purchase in data markets.

For example, Figure 1 shows four heterogeneous schemas to be matched that differ in volume, structural design, and domain.

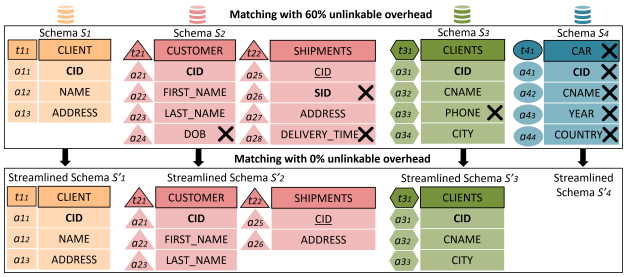


Figure 1: Example of scoping streamlined schemas by distinguishing linkable tables and attributes from unlinkable ones (cross symbol) for multi-source schema matching.

In schema S_1, the table CLIENT and attribute ADDRESS are semantically similar to table CUSTOMER in S_2 and attribute CITY in S_3. Distinguishing relevant from irrelevant tables and attributes for schema matching is a challenge because each schema is constructed with concepts modeled for a specific application, business case, or domain. Consequently, the attributes DOB, SID, DELIVERY_TIME, and PHONE are irrelevant for the other schemas. Matching more than two schemas expands the search space in increasing orders of magnitude with additional attributes and tables that may represent identical or sub-typed (e.g., CITY ≅ ADDRESS) or entirely dissimilar concepts. In the worst case, schemas from a completely unrelated domain such as S_4 (containing Formula One car info) must be still compared to the other schemas, even though the table CAR and attributes CID, CNAME, YEAR, and COUNTRY are unlinkable. *Unlinkable schema elements pose a significant overhead. They not only occupy computational space but also negatively impact the matching quality: When unlinkable elements remain in the matching process, they may transit to overgeneralized schema clusters.* For example, the global schema matching attempt between the schema S_4 with S_1, S_2, and S_3 would lead to false linkages such as YEAR and DOB and missing ones such as ADDRESS and CITY because, element-wise, CITY and COUNTRY are considered to be more similar.

Existing multi-source schema matching solutions fall short by not separating relevant from irrelevant schema portions. By processing them in their original form, every element needs to be searched for potential correspondences in all other schemas. When associated linkages fall below a similarity threshold [9, 10, 17, 23, 50] or when they are not captured in a cardinality-based cluster [21, 27, 33, 37] (Section 2.2), only then they are considered unlinkable. Using a global pruning threshold limits high-quality linkages, especially when matching multiple heterogeneous schemas. In prior work [44], we proposed *Scoping*, a method that adapts outlier detection algorithms (ODAs) for pruning unlinkable schema elements ahead of matching. Although scoping works for domain-specific matching, it is inadequate for heterogeneous schemas as it applies an ODA globally (Section 2.4).

EDBT '26, Tampere (Finland)
© 2025 Copyright held by the owner/author(s). Published on OpenProceedings.org under ISBN 978-3-89318-102-5, series ISSN 2367-2005. Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.

10.48786/edbt.2026.03

In this paper, we propose a new way of pruning unlinkable schema elements ahead of matching: Our approach leverages self-trained encoder-decoder models, which capture the independent semantics of local schemas. Specifically, we identify linkable elements from another schema when they are reconstructed in such a way that an encoder-decoder recognizes local patterns. Hence, our approach tailors source-specific functions and thresholds for the linkability assessment, making it applicable as a robust and self-supervised pre-processing method that prunes away the unlinkable elements and keeps only the linkable subset of the original schemas. Extensive evaluation has shown that matching algorithms benefit from such streamlined schemas as they contain only the linkable subsets of the originals. Our contributions are the following:

- A formal description of the schema *linkability* problem for multi-source schema matching (Section 2).
- A novel methodology that produces streamlined schemas for schema matching using distributed self-supervised encoder-decoder models. The models capture the local schema semantics to assess the linkability of tables and attributes and to prune unlinkable ones. (Section 3).
- A technique that automatically identifies local linkability thresholds. This is an improvement over existing approaches that implement a global similarity or cardinality threshold for matching that are, in essence, unknown and dependent on the schema matching scenario.
- An empirical evaluation of two multi-source schema matching scenarios to convey the practical effectiveness of the proposed algorithmic solution (Section 4). We show that collaborative scoping excels in highly heterogeneous (up to +26%) and domain-specific (up to +8%) matching scenarios compared to scoping baselines. Collaborative scoping remains robust regardless of the schema heterogeneity and is even more efficient because of the reduced complexity.
- An ablation study of collaborative scoping for a Cosine, k-Means, and LSH matcher that empirically validates an improvement in the quality of linkages (up to +80%) and F1-measure (up to +20%) with consistent efficiency gains.

Although our approach mainly targets multi-source schema matching scenarios, it also works well for pruning unlinkable elements for source-to-target matching.

2 Preliminaries

In this section, we provide preliminary definitions relevant to schema linkages. Then, we describe related work on traditional schema matching solutions. Next, we recapture language models and a prior scoping technique using outlier detection algorithms. Table 1 summarizes the notations used.

2.1 Problem Formulation

Schema Linkages: We are given a set of relational schemas $S = (S_1, S_2, \ldots, S_k)$ that are heterogeneous in volume (number of tables and attributes), structural design (level of normalization and attribute atomicity), and domain, that need to be matched. Each schema $S_k = \{t_{k_1}, t_{k_2}, \ldots, t_{k_i}\}$ contains a set of tables and each table $t_{k_i} = \{a_{k_1}, a_{k_2}, \ldots, a_{k_j}\}$ contains a set of attributes. For multi-source matching, the tables and attributes of one schema S_k are aligned with at least another schema S_m. The alignment between heterogeneous schemas is not fully bijective, since they may include one-to-one and one-to-many linkages as well as non-correspondences (unlinkable elements). We define the relevant inter-linkages between the schemas as the set of table pairs and attribute pairs between them: $L(S) = \{(t_{k_i}, t_{m_l}), (a_{k_j}, a_{m_n}), \ldots : t_{k_i} \in S_k \wedge t_{m_l} \in S_m \wedge a_{k_j} \in t_{k_i} \wedge a_{m_n} \in t_{m_l}\}$ where $S_k, S_m \in S$ and $k \neq m$. The table-pairs ($t_{k_i} \cong t_{m_l}$) or attribute-pairs ($a_{k_j} \cong a_{m_n}$) represent a semantic congruence. Note that both binary relationships are symmetric and represent the ground truth.

In order to explain the introduced notion of congruence, we refer to the multi-source schemas depicted in Figure 1. To this end, we seek the following linkage types:

(1) **I**nter-**I**dentical: This linkage type represents identical one-to-one semantics between attribute or table pairs. No further modification of schema elements is necessary apart from lexical normalization (e.g., rename a_{1_2} `NAME` $\Rightarrow$ `CNAME` so it becomes identical to a_{3_2}). Note that the attributes a_{3_2} `CNAME` and a_{4_2} `CNAME` are not inter-identical because of the different underlying semantics (i.e., client name versus car name).

(2) **I**nter-**S**ub-Typed Attributes: These one-to-many linkages represent a relationship between schema attributes with partial information intersection. There are two cases: First, partial information of one attribute links to another (e.g., a split a_{1_3} `ADDRESS` links to a_{3_4} `CITY`). Second, two or more attributes from one schema link to another schema's attribute (e.g., a_{2_2} `FIRST_NAME` and a_{2_3} `LAST_NAME` both link to a_{1_2} `NAME`).

(3) **I**nter-**S**ub-Typed Tables: We consider two tables between different schemas as conceptually similar if they have at least one inter-identical or inter-sub-typed attribute relationship. Tables that contain additional non-matching attributes can be reduced to relevant ones via projection. As an example, projecting $\Pi(a_{2_1}, a_{2_2}, a_{2_3})$ on table t_{2_1} `CUSTOMER` excludes a_{2_4} `DOB` because it is not contained in table t_{1_1} `CLIENT`. This type also includes one-to-many table linkages, e.g., table t_{1_1} `CLIENT` also links to the table t_{2_2} `SHIPMENTS` in S_2 due to the matching customer identifiers and locations. Note that table t_{4_1} `CAR` is not linked to any other table because none of its attributes link to another despite lexical similarities.

Every integration approach aims at finding as many true linkages as possible. The linkages can be used to integrate the schemas with transformations and conjunctive queries (i.e., JOINs and UNIONs), which are out of scope in this work.

Table 1: Symbols and Description.

Symbol	Description
S	Set of multi-source schemas $\{S_1, S_2, \ldots, S_k\}$ to be matched.
S_k	Schema with a set of tables $\{t_{k_1}, t_{k_2}, \ldots, t_{k_i}\}$.
t_{k_i}	Table with a set of attributes $\{a_{k_1}, a_{k_2}, \ldots, a_{k_j}\}$ and metadata on tn_{k_i} table name and $\{an_{k_1}, an_{k_2}, \ldots, an_{k_j}\}$ attribute names.
a_{k_j}	Attribute with metadata on an_{k_j} attribute name, tn_{k_i} table name, d_{k_j} data type, and c_{k_j} constraint.
$L(S)$	All inter-identical and inter-sub-typed linkages between schemas $\{(t_{k_i}, t_{m_l}), (a_{k_j}, a_{m_n}), \ldots : t_{k_i} \in S_k \wedge t_{m_l} \in S_m \wedge a_{k_j} \in t_{k_i} \wedge a_{m_n} \in t_{m_l}\}$ where $S_k, S_m \in S$ and $k \neq m$.
S'	Streamlined schemas $\{S'_1, S'_2, \ldots, S'_k\}$ where each is a subset $S'_k \in S_k$ of the original schema and $L(S') \approx L(S)$.
S^t_k	Metadata based text sequences of attributes and tables in a schema $(e^t_{k_j} \leftarrow T^a(a_{k_j}) \vert a_{k_j} \in t_{k_i} \in S_k) \cup (e^t_{k_i} \leftarrow T^t(t_{k_i}) \vert t_{k_i} \in S_k)$.
$S^{\vec{v}}_k$	Language model encoded schema signatures $(e^{\vec{v}}_{k_i} \leftarrow E(e^t_{k_i}) \vert e^t_{k_i} \in S^t_k))$.

To effectively manage the complex linkage solution space, we propose to decompose it using the notion of linkability that characterizes tables and attributes as linkable or unlinkable within the context of schema matching.

Definition 1. *Linkability*: Any attribute or table in a set of schemas S that occurs in a linkage pair in $L(S)$ is defined as *linkable*. All other attributes and tables not in any of the linkage pairs are denoted as *unlinkable*.

Given the relations in Figure 1, the attribute a_{2_4} DOB in schema S_2 or the whole table t_{4_1} CAR and its attributes $a_{4_1}, a_{4_2}, a_{4_3}$, and a_{4_4} in schema S_4 are considered to be unlinkable (cross symbol). The distinction between linkable and unlinkable schema elements provides a means to use in order to scope streamlined schemas for heterogeneous multi-source matching pipelines.

Definition 2. *Streamlined Schemas*: Given a set of multi-source schemas $S = \{S_1, S_2, \ldots, S_k\}$, we aim to identify the streamlined subset of these $S' = \{S'_1, S'_2, \ldots, S'_k\}$ with $S'_k \subseteq S_k$ which include only linkable tables and attributes. Identifying the set of streamlined schemas S' is an approximation and thus may also include false positive and false negative linkability assignments. The latter has a serious impact on matching because associated inter-linkages will remain undiscovered for the falsely pruned schema elements.

Given the overall number of elements in the schemas $|S|$ and the linkable ones $|S'| \geq 2$, the overhead of unlinkable ones that need to be processed when matching the schemas can be calculated as $\frac{|S|-|S'|}{|S'|} \in \mathbb{R}^+$. For example, the multi-source schema matching scenario in Figure 1 has an unlinkable overhead of $\frac{24-15}{15} = 60\%$.

The goal of our work is to find the best approximation of $L(S')$, which is as close as possible to the schema linkage set $L(S)$ using the original schemas. As a result, we generate streamlined schemas S' that contain only relevant (linkable) elements and, thus, inherently reduce the computational overhead of the subsequent matching pipeline to finally generate higher quality schema linkages, regardless of the applied blocking and matching workflow.

2.2 Related Work

Our proposed method relates to multi-source, schema-based, and target-free relational matching scenarios. Our goal is a distributed and self-supervised pre-processing method for pruning unlinkable schema elements, thus improving the efficiency and effectiveness of traditional matching (ref. Figure 2). Therefore, we now discuss existing matching algorithms, which all require model-specific parameters as well as a global threshold on similarity or cardinality. The threshold values, as well as the parameters, are basically unknown and highly influence the matching process.

Matching with similarity. The long-standing problem of schema matching can be split into schema-based, instance-based, and hybrid. Schema-based matching uses similarity between names, descriptions, synonyms, data types, and constraints of schema elements. For instance-based matching, the actual instance values, patterns, and functional dependencies can be used. Over the years, numerous element-wise algorithms were proposed, e.g., *CUPID* [25], *Similarity Flooding* [28], *COMA* [2], all packaged in the *Valentine* project by Koutras et al. [23]. In such algorithms, unlinkable schema elements are merely pruned when associated linkages fall below a user-defined similarity threshold; the accurate threshold value differs from linkage to linkage and between schemas.

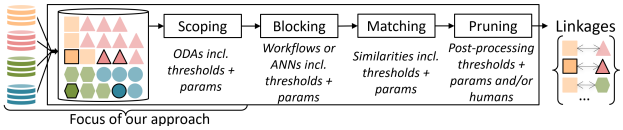


Figure 2: Traditional global matching pipeline.

Exclusively relying on string similarity (e.g., Levenshtein or Fuzzy) between schema names suffers from labeling conflicts. Therefore, recent approaches learn to encode [3, 8, 12, 47], fine-tune [50], or use pre-trained Language Models [17] in order to transform textual descriptions into a *Signature*, a fixed-size numeric embedding (Section 2.3). The values of actual instances (records) may also contribute to the semantics of schema elements. However, matching in the context of privacy-preserving organizations and data markets[1] must resort to the understanding of the linguistics of schema metadata, as access to instance data is limited.

Recently, Shraga and Gal developed *ADnEV* to improve the matching effectiveness by adjusting the similarity matrix between schema elements using a deep neural network [41]. In contrast, our approach scopes streamlined schemas upfront in order to avoid the computational overhead of maintaining linkages that need to be pruned in a post-processing phase. Loster et al. proposed Siamese Neural Networks in order to learn a tailored similarity function that aligns with the characteristics of heterogeneous sources [24]. In contrast to our work, this supervised approach requires linkage annotations that are difficult to obtain. In particular, transferring the models to another matching scenario leads to performance improvements, which motivated our distributed approach. Shraga et al. developed *PoWareMatch* to calibrate a matcher and decision boundary based on temporal matching decisions of an active-learning pipeline [40]. The authors' work suggests that matching improves with threshold calibration, motivating our self-supervised collaborative scoping approach that generates source-specific linkability functions and thresholds but does not require any annotated linkages or linkability labels.

Blocking with cardinality. In general, there is a significant difference between source-to-target $O(|S_{\text{source}}| \cdot |S_{\text{target}}|)$ and multi-source (holistic) schema matching $O(|S_1| \cdot |S_2| \cdot \ldots \cdot |S_k|)$. The goal is to derive linkages that represent groups of semantically related elements, a challenge in efficiency and effectiveness.

He and Chang proposed a holistic ensemble approach that ranks majority voting among matching multiple schemas [15]. Their approach targets large quantities of heterogeneous input schemas but uses sampling techniques to scale holistically. Instead of sampling, a more promising technique to avoid the Cartesian product size of element-wise comparisons is *Blocking*. Here, schema elements that are likely to match are efficiently grouped without affecting linkage completeness (recall) [4, 29, 30, 43]. Blocking utilizes clustering or approximate nearest neighbor search (ANNs), both based on a user-defined cardinality.

Clustering. Papadakis et al. [33] *JedAI* system implements a k-Means module for matching attributes using their names, instances, or combinations [32]. Sahay et al. propose k-Means and Self-organizing map (Kohonen) limited to one-to-one linkages between source-to-target matching [37]. More recently, Khatiwada

[1] AWS marketplace https://aws.amazon.com/marketplace contains 4624 *Data Exchange* offers, of which 268 (5.8%) provide data samples.

et al. propose *ALITE* to bond the clustering cardinality to the Silhouette coefficient [21]. Even if the cluster cardinality is known or self-tuned, all methods still have unlinkable elements in their global matching space, likely leading to more false linkages.

ANNs are built to retrieve the approximately nearby data points (signatures of another schema) for a query item (signature of schema). One famously applied technique is locality-sensitive hashing (LSH) in Data Discovery [11, 22] and Entity Resolution [31, 43] that also suits well to holistic schema matching. For example, Meduri et al. designed *Alfa* that pre-selects linkages for active learning via "semantic blocking" variants (i.e., *SIM*, *CLUSTER*, *LSH*) [27]. However, ANNs require a global top-k cardinality that needs to be searched for every single element, including unlinkable ones.

All related matching approaches use schemas in their original state as input and would benefit from our collaborative scoping approach as a pre-processing module. In Section 4, we show Meduri et al.'s "semantic blocking" variants to be prone to false positive linkages (precision) when schemas contain unlinkable elements.

2.3 Language Models and Schema Signatures

Language models that implement an encoder-decoder architecture, such as Sentence-BERT [34], are trained to transform sequences of words into a fixed-sized latent vector. Formally, given an encoder-based language model E and some input text sequence $t = \{w_1, w_2, \ldots, w_d\}$, first, each word $w_d \in t$ is encoded using word embeddings and consolidated as a matrix set that the encoder transforms via average pooling in order to output a fixed-size vector $\vec{v}$. To this end, we capture the semantic nuances of the textual serializations of tables and attributes by encoding them into signatures. These are used to determine true linkages among the schemas using similarity (e.g., Cosine) with high scores.

Due to limitations on data access for matching organizations and data markets outlined in Section 2.2, we focus on metadata to encode schema-based signatures. For *a*ttributes, we therefore extract the attribute name, table name, data type, and constraint $a_{k_j} = (an_{k_j}, tn_{k_i}, d_{k_j}, c_{k_j})$. For simplicity, the constraint is restricted to `PRIMARY KEY` or `FOREIGN KEY`, the latter without the reference value. Then, the attribute object values are concatenated into a text sequence using the function T^a. As an example, $T^a(a_{1_1})$ from Figure 1 returns $a^t_{1_1}$: `"CID CLIENT NUMBER PRIMARY KEY"`.

We encode the information of tables in a similar way. To this end, we incorporate the table name, [attribute names] $t_{k_i} = (tn_{k_i}, \{an_{k_j} \leftarrow a_{k_j} | \forall a_{k_j} \in t_{k_i}\})$ from the metadata. Then, we generate descriptive text sequences for the tables using the function T^t. For example, the serialization of the table $T^t(t_{1_1})$ results in the text sequence $t^t_{1_1}$: `"CLIENT [CID, NAME, ADDRESS, PHONE]"`.

While instance data samples may become more accessible in data markets, incorporating these into the schema element serialization must be carefully considered for each matching scenario. Given the attributes $a^t_{1_2}$: `NAME CLIENT (Michael Scott)`, $a^t_{2_2}$: `FIRST_NAME CUSTOMER (Michael)`, and $a^t_{2_3}$: `LAST_NAME CUSTOMER (Bluth)`, Sentence BERT captures the semantic similarities between `CLIENT`, `CUSTOMER` as well as `NAME`, `FIRST_NAME`, and `LAST_NAME`. However, including the instance samples (in parentheses) increases the cosine similarity[2] between $a^{\vec{v}}_{1_2} \sim a^{\vec{v}}_{2_2}$ (+5%) but decreases $a^{\vec{v}}_{1_2} \sim a^{\vec{v}}_{2_3}$ (-11%). Overall, including instance samples in the serialization results in less effective matching results [44]. However, effective semantic enrichment strategies exist, such as embedding learning [3, 8, 12], encoder fine-tuning [50], or using LLMs [13, 38, 45, 49]. They can be easily integrated, but they are out of the scope of this paper.

[2]Cosine similarity between Sentence-BERT (all-mpnet-base-v2) encoded signatures.

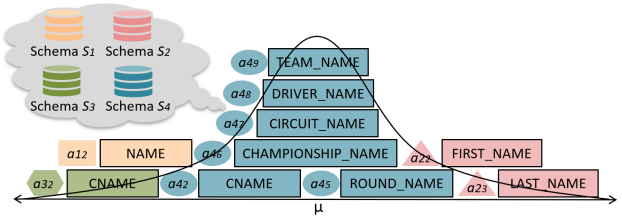


Figure 3: Global example of a normal distribution from collected attributes on names among heterogeneous schemas.

2.4 Scoping Streamlined Schemas

Recently, we introduced an approach called *Scoping* to identify linkable schema elements [44] in a multi-source schema matching scenario. To clarify the differences between Scoping and our new approach, we briefly summarize Scoping before actually introducing *Collaborative Scoping* in Section 3. *Scoping* is a method to generate streamlined schemas by ranking, sorting, and filtering the original schema signatures on linkability:

(1) *Ranking with Outlier Detection Algorithms (ODAs)*: ODAs are designed to identify a normal distribution from a data set in order to identify data points with significant deviations as anomalies. We adopt ODAs to score tables and attributes on linkability across schemas. Using the set of all textually sequenced and encoded signatures from the schemas $S^{\vec{v}} = \{e^{\vec{v}}_{1_1}, \ldots, e^{\vec{v}}_{k_i}\}$, each signature obtains an outlier score. The output is a set of tuples $\{(e_{1_1}, s_{1_1}), \ldots, (e_{k_i}, s_{k_i})\}$, where e_{k_i} is an attribute or table and s_{k_i} its respective outlier score.
(2) *Sorting*: The signature score tuples $[(e_{k_i}, s_{k_i}), (e_{k_j}, s_{k_j}), \ldots]$ are sorted in descending order of the outlier scores $s_{k_i} < s_{k_j}$.
(3) *Scoping*: The sorted signature score tuples are filtered using the relative threshold parameter $p \in (0..1)$. The output is streamlined schemas $S' \subseteq S$ that contain the p portion of schema signatures with lower outlier scores (linkable), leaving the anomalous ones (unlinkable) aside. Scoping with $p = 1$ is equivalent to the original set of input schemas $S' \equiv S$, while $p = 0$ results in empty schemas $S' = \phi$.

It is worth noting that the input size for ODAs is linear in the number of schema elements $|S_1| + |S_2| + \ldots + |S_k|$ and not the Cartesian product size between all possible pairs $|S_1| \cdot |S_2| \cdot \ldots \cdot |S_k|$. Therefore, various ODAs can be used for global scoping. A straightforward method is to compute the signature's standard deviation of the mean μ (Z-score). Alternatively, the density-based approach Local Outlier Factor (LOF) quantifies the local signature distance of a cluster [7]. Recently, encoder-decoder models have received great attention [14, 18, 20, 35]. Self-supervised models, such as Principal Component Analysis (PCA) [39] and Neural Autoencoders [5, 46] (that generalize PCA) function as ODAs by reconstructing signatures to their original state. In general, when a signature is close to the mean, has a low distance to a cluster centroid, or is reconstructed with a low error, then it is considered to be linkable.

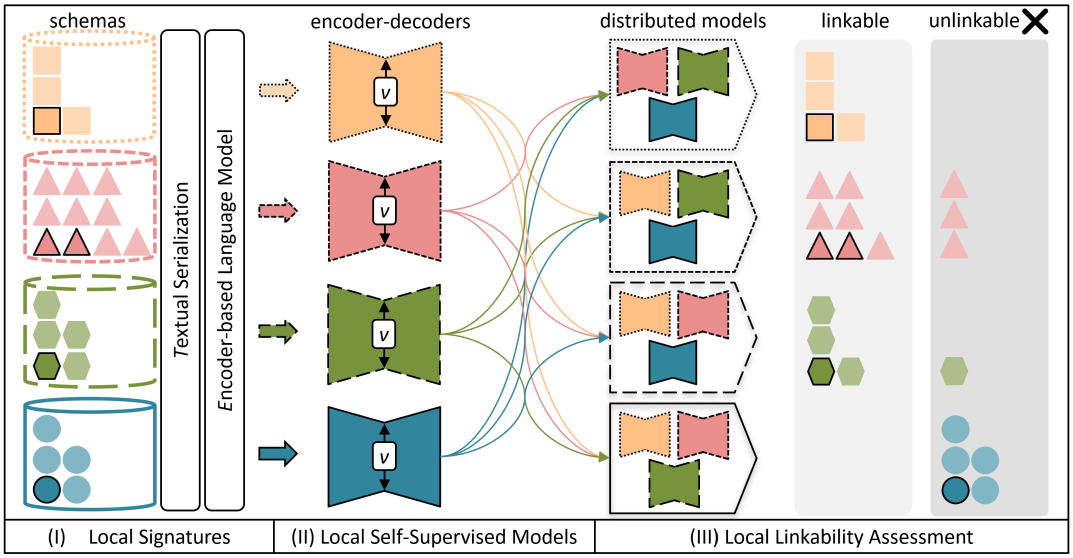


Figure 4: Proposed collaborative scoping framework.

Scoping yields promising results, but it does not work well on schemas that are heterogeneous in volume, design, and domain. As matching heterogeneous schemas may involve an excessive number of unlinkable elements, a single ODA will not be able to capture the linkable ones as the normal distribution. In this case, scoping assesses unlinkable schema elements as linkable (false positives) and linkable schema elements as unlinkable (false negatives). For example, Figure 3 illustrates an excerpt of a normal distribution of attribute signatures from four schemas, of which S_{1-3} represent customer names while S_4 represents an entirely different domain, Formula One data.

Volume. A single ODA assigns equal weight to schema signatures, working under the assumption that the matching schemas have similar volumes of tables and attributes. In reality, they vary in size so that one schema dominates the mean of the normal distribution (e.g., $|S_4| > |S_2| > |S_1| = |S_3|$) with its own elements.

Design. Additionally, the schemas may have varying decompositions of tables and attributes to describe a common concept, e.g., `FIRST_NAME` and `LAST_NAME` (S_2) represent customer `NAME` (S_1) and `CNAME` (S_3). Instead of recognizing similarities across different schema designs, a single ODA would allocate structural differences as anomalous with high outlier scores.

Domain. Lastly, schemas may contain domains or concepts that are completely irrelevant to others (e.g., customer names in S_{1-3} versus Formula One names S_4). A single ODA would have difficulties defining what constitutes the "normal" domain among all heterogeneous schemas. In the worst case, an entirely unrelated schema occupies the most frequent patterns (i.e., Formula One names in S_4). Unfortunately, the unlinkable elements receive low outlier scores while linkable elements receive high ones. Thus, a global scoping approach would generate schema subsets that are ineffective for subsequent matching pipelines.

3 Method

In this section, we introduce collaborative scoping as a robust approach to locally examine the linkability of schema elements on their way toward matching multiple heterogeneous schemas with different volumes, designs, and domains. We first introduce the overall framework, followed by details of the three phases: *(I) Local Signatures*, *(II) Local Self-Supervised Models*, and *(III) Local Linkability Assessment*. Finally, we discuss the computational effort.

Overview. Collaborative scoping is a self-supervised method for pruning unlinkable schema elements. As illustrated in Figure 4, we follow the three sequential phases for generating streamlined schemas, i.e., *(I) Local Signatures*, *(II) Local Self-Supervised Models*, and *(III) Local Linkability Assessment*, for more efficient and effective schema matching pipelines. In contrast to the established matching pipelines (ref. Figure 2), the collaborative scoping approach locally learns the semantics of each input schema. In the first phase, the schemas transform their tables and attributes into signatures. In the second phase, each schema self-supervises an encoder-decoder to capture the local schema semantics based on a global variance parameter that ensures a common degree of generalizability. In the last phase, the schemas locally assess the linkability of tables and attributes using the distributed encoder-decoders. The outputs are streamlined schema subsets to be used for schema matching.

In comparison to the traditional matching pipeline, this self-supervised approach reduces irrelevant (unlinkable) schema elements instead of passing them into matching algorithms. Consequently, we inherently reduce computational overhead for matching (improve efficiency) and avoid an extensive post-pruning of unlinkable tables, attributes, and associated linkages (improve effectiveness).

(I) Local Signatures. In Section 2.3, we describe a centralized approach to process tables and attributes uniformly regardless of schema origins. In order to ensure consistency and compatibility within collaborative scoping among the distributed schemas $S_1, S_2, \ldots, S_k$, the metadata of tables and attributes must be extracted, pre-processed, and uniformly encoded. Therefore, the schemas agree on a global textual serialization (T^t and T^a) and encoder-based language model (E) to transform the local schema serializations into a set of signatures, each being a fixed-size vector of uniform length $S_k^{\vec{v}} = (e_{k_i}^{\vec{v}} \leftarrow E(e_{k_i}^t) | \forall e_{k_i}^t \in S_k^t)$.

(II) Local Self-Supervised Models. In [44], we have shown that encoder-decoders are effective in scoping streamlined schemas

for domain-specific matching scenarios. However, we showed in Section 2.4 that a single global ODA is ineffective for pruning unlinkable elements in heterogeneous schemas. To avoid the outlined problems, we propose to encode-decode the normal distribution of each local schema independently. Note that a self-trained encoder-decoder can also process the signatures of another schema.

Intuitively, if a schema element in S_k is similar to another one in S_m, then the element in S_k also must be recognized by the encoder-decoder of schema S_m. This is true because an encoder-decoder maintains an inherent generalizability that allows the recognition of unseen data associated with a reconstruction error. We propose to train a self-supervised encoder-decoder based on Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) for the signatures $S_k^{\vec{v}}$ in a schema S_k. This way, the semantics of each local schema are summarized into a single encoder-decoder model.

The number of principal components directs the generalization for a PCA-based encoder-decoder and needs to be parameterized. A high number of principal components would overfit so that the variance of the signatures would be entirely reconstructed. On the other hand, a small number of principal components barely captures the variance of the signatures, leading to high reconstruction errors. The parameterization of PCA is a problem because the number of principal components is unknown. Furthermore, they are different for each local schema due to the heterogeneity in volume, design, and domain. Identifying the number of principal components for each local encoder-decoder is closely tied to balancing the generalizability with the risks of overfitting and underfitting the self-supervised patterns of local schema signatures.

In order to achieve interoperability for collaborative scoping among the heterogeneous schemas, we need to maintain a consistent generalization complexity of their encoder-decoders. As we use SVD, we propose to use the explained variance v as a global reference to homogeneously quantify how much of the variance of the local schema signatures is captured by the first n principal components. The global explained variance v enables a common degree of generalization for the encoder-decoders. The ideal value for v is unknown and varies between the matching scenarios. Our domain-specific and heterogeneous experiments have shown that $v = [0.95, 0.6]$ achieves a good balance between precision and recall.

The (II) Local Self-Supervised Models phase is applied to each schema and outlined in **Algorithm 1** with the key observations:

- Line 1-2: Generate local schema signatures with global textual serialization function and language model encoder.
- Line 3-5: Project schema signatures onto the mean and compute full singular value decomposition.
- Line 6-10: Select principal components based on a globally determined explained variance.
- Line 11-13: Encode and decode the schema signatures with the selected principal components.

The output of Algorithm 1 is the local model M_k that constitutes the local schema signature mean μ_m and local principal components PC_m to be reused for the reconstruction of signatures from other schemas. Because each encoder-decoder now captures only its native schema context, scoping the outlier scores with a global threshold is not useful. Consequently, it is not trivial to effectively handle the linkability assessment of tables and attributes in a collaborative context. Instead, our collaborative

Algorithm 1 Local Self-Supervised Models

Input: S_k local schema, T^a, T^t global textual serialization, E global language model encoder, $v \in (1..0)$ global variance
Output: Local model: $M_k = \{\mu_k$ local signature mean, PC_k local principal components, l_k local linkability range$\}$

1: $S_k^t \leftarrow (e_{k_j}^t \leftarrow T^a(a_{k_j}) | a_{k_j} \in t_{k_i} \in S_k) \cup$ $(e_{k_i}^t \leftarrow T^t(t_{k_i}) | t_{k_i} \in S_k)$ //Local serialization.
2: $S_k^{\vec{v}} \leftarrow (e_{k_i}^{\vec{v}} \leftarrow E(e_{k_i}^t) | e_{k_i}^t \in S_k^t))$ //Local signatures.
3: $\mu_k = \text{mean}(S_k^{\vec{v}})$ //Compute local signatures mean.
4: $X_{origin} = S_k^{\vec{v}} - \mu_k$ //Project signatures onto origin.
5: $SV = \{sv_1, sv_2, \ldots\}, PC = \{pc_1, pc_2, \ldots\} = SVD(X_{origin})$ //Compute full **S**ingular **V**alue **D**ecomposition and return **S**ingular **V**alues and **P**rincipal **C**omponents.
6: $EV^{sum} = \sum_{j=1}^{SV} sv_j^2$ //Compute the sum of the squared SV for **E**xplained **V**ariance.
7: $EV \leftarrow (ev_i = \frac{sv_i^2}{EV^{sum}} | \forall sv_i \in SV)$ //Compute EV per PC.
8: $CEV = (ev_1, ev_1 + ev_2, \ldots) \leftarrow \text{CumulativeSum}(EV)$ //Cumulate EV for each added PC.
9: $n_{\text{comp}} \leftarrow \text{GetIndex}(CEV, v) + 1$ //Find PC **n**umber needed to locally explain the variance so that $> v$.
10: $PC_k \leftarrow \{pc_1, pc_2, \ldots\}$ with $pc_l \in PC \wedge l < n$ //Reduce set of all PC to the top-n.
11: $X^Z = X_{origin} \cdot PC_k^T$ //Encode projected signatures.
12: $\hat{X}_{origin} = X^Z \cdot PC_k$ //Decode signatures.
13: $\hat{X} = \hat{X}_{origin} + \mu_k$ //Reverse projection onto origin.
14: $S_k^s \leftarrow \{s_{k_i} = MSE(e_{k_i}^{\vec{v}}, \hat{x}_i) | \forall (e_{k_i}^{\vec{v}}, \hat{x}_i) \in (S_k^{\vec{v}}, \hat{X})\}$ //Compute mean reconstruction error score of original and encoded-decoded signatures.
15: $l_k \leftarrow max(s_{k_i} \in S_k^s)$ //Select maximum outlier score as local linkability range.
16: **return** $M_k = \{\mu_k, PC_k, l_k\}$ //Local model components.

scoping strategy automatically determines a range for assessing the linkability as the third essential component of a local schema model (Line 14-15).

Definition 3. *Local Linkability Range:* Given the encoder-decoder components μ_k, PC_k, and a set of local schema signatures $S_k^{\vec{v}} = \{e_{k_1}^{\vec{v}}, e_{k_2}^{\vec{v}}, \ldots, e_{k_i}^{\vec{v}}\}$, the signature in the set that reconstructs with the largest outlier score s_{k_i} represents the local linkability range l_k of the local model M_k.

Intuitively, a foreign schema element e_{k_i} is considered to be linkable if the encoder-decoder M_m of another schema recognizes it with a reconstruction error that falls into the reconstruction range $[0, \ldots, l_m]$ of its schema elements that it has been trained on. This threshold is relatively strict, as one could allow for more reconstruction errors such as $l_m + \epsilon$. However, the local linkability range l_m is already dependent on the explained variance v, and experiments have shown that further relaxation leads to no improvement in the overall performance.

(III) Local Linkability Assessment. We design a local linkability assessment strategy that does not exchange tables and attributes among the schemas, but the self-trained encoder-decoders. This way, each local schema individually assesses its linkability and streamlines it for schema matching, shown in **Algorithm 2**.

The local schema uses its encoded signature set $S_k^{\vec{v}}$ to recognize linkable schema tables and attributes to generate a streamlined schema. Instead of applying its own model $M_k = \{\mu_k, PC_k, l_k\}$,

the local schema iterates through the set of distributed encoder-decoder models $M = \{M_1, M_2, \ldots, M_m\} \setminus \{M_k\}$ from the schemas intended to match (Line 1-6). Therefore, schema k uses the schema model M_m in order to encode its local signatures $S_k^{\vec{v}}$ in the latent space of schema m. Then, it decodes its salient features using the reverse operation ($\cdot PC_m$ and $+\mu_m$) back into its original state.

Definition 4. *Linkability Assessment:* Given the distributed encoder-decoder model $M_m = \{\mu_m, PC_m, l_m\}$ from another schema and a set of local schema signatures $S_k^{\vec{v}} = \{e_{k_1}^{\vec{v}}, e_{k_2}^{\vec{v}}, \ldots, e_{k_i}^{\vec{v}}\}$, the corresponding tables and attributes in it are linkable when they reconstruct with an outlier score s_{k_i} smaller than the local linkability range l_m. Specifically, the model encoder-decoder function $M_m(e_{k_i}^{\vec{v}})$ filters linkable (true) from unlinkable (false) schema signatures as follows:

$$M_m(e_{k_i}^{\vec{v}}) = MSE(e_{k_i}^{\vec{v}}, ((e_{k_i}^{\vec{v}} - \mu_m) \cdot PC_m^T \cdot PC_m + \mu_m) \leq l_m$$

The output of Algorithm 2 is a streamlined schema S_k' with all tables and attributes considered linkable by one or more of the distributed encoder-decoder models M. The streamlined schema S_k' can be subsequently used by a traditional matching pipeline (ref. Figure 2), avoiding ineffective and inefficient linkages that would have needed to be filtered via subsequent blocking, matching, and post-pruning via user-defined method parameters and thresholds.

Computational Complexity. We intend to match k different schemas $S_1, S_2, \ldots, S_k$, where one schema S_k has a number of signatures (rows) that we denote as $|S_k|$ with the predefined length $|\vec{v}|$ (columns) of the encoder-based language model E.

Scoping. The main computational effort to scope streamlined schemas is calculated with the applied ODA:

- Z-Score: $O(|S| \cdot |\vec{v}|)$
- Local Outlier Factor (LOF) and outlier score computation: $O(|S|^2 \cdot |\vec{v}|) + O(|S| \cdot n)$
- Principal Component Analysis (PCA) with full SVD and reconstruction for outlier scores: $O(|S|^2 \cdot |\vec{v}| + |\vec{v}|^3) + O(|S|)$
- Neural Autoencoder (AE): It is dependent on network configurations such as layers, neurons, and epochs. The time complexity of the simplest network is higher than that of PCA.

With respect to LOF and PCA, the computational complexity of scoping has a constant factor with the signature length $|\vec{v}|$ and predominantly rises in the quadratic sum of signatures $|(S_1 + S_2 + \ldots + S_k)|^2 = |S|^2$ among all schemas intended to match.

Collaborative scoping. Following the same goal with streamlined schemas, we construct one encoder-decoder for every local schema k, which additionally needs to reconstruct its signatures with $|M| = k - 1$ models of the other schemas. The pooled time complexity of the collaborative scoping method with PCA based on Algorithm 1 and 2 advances to the following order

$$\sum_k O(|S_k|^2 \cdot |\vec{v}| + |\vec{v}|^3) + \sum_k O(|S_k| \cdot |M|) = O((|S_1|^2 + |S_2|^2 + \ldots + |S_k|^2) \cdot |\vec{v}| + k \cdot |\vec{v}|^3 + |S| \cdot |M|)$$

because of $|S_1| + |S_2| + \ldots + |S_k| = |S|$. The computational complexity depends on the size $|S_1| + |S_2| + \ldots + |S_k|$ that needs to be additionally factored by the number of k schemas. Note that the higher the number of schemas k gets, the lower the sum of quadratic signatures from local schemas becomes in comparison to the quadratic number of the unified signature set $|S_1|^2 + |S_2|^2 + \ldots + |S_k|^2 < |S|^2$. Furthermore, the computation of the self-supervised encoder-decoder and linkability assessment takes place in parallel at each local schema.

Algorithm 2 Local Linkability Assessment

Input: $S_k^{\vec{v}}$ local schema signatures, $M = \{M_1, M_2, \ldots, M_m\} \setminus \{M_k\}$ models of all other local schemas where $M_m = \{\mu_m, PC_m, l_m\}$
Output: Streamlined schema: $S_k' = \{e_{k_1}, e_{k_2}, \ldots, e_{k_i}\}$

1: **for all** $M_m \in M$ **do**
2: $X_{origin} = S_k^{\vec{v}} - \mu_m$ `//Project signatures.`
3: $X^Z = X_{origin} \cdot PC_m^T$ `//Encode signatures.`
4: $\hat{X}_{origin} = X^Z \cdot PC_m$ `//Decode signatures.`
5: $\hat{X} = \hat{X}_{origin} + \mu_m$ `//Reverse projection onto origin.`
6: $S_k^s \leftarrow \{s_{k_i} = MSE(e_{k_i}^{\vec{v}}, \hat{x}_i) | \forall (e_{k_i}^{\vec{v}}, \hat{x}_i) \in (S_k^{\vec{v}}, \hat{X})\}$ `//Compute mean reconstruction error score of original and encoded-decoded signatures.`
7: **for all** $s_{k_i} \in S_k^s$ **do**
8: **if** $s_{k_i} \leq l_m$ **then**
9: $S_k' \leftarrow Append(S_k', e_{k_i})$ `//Append linkable table or attribute signature` $e_{k_i} \in S_k$ `to streamlined schema` S_k'.
10: **end if**
11: **end for**
12: **end for**
13: **return** S_k'

4 Evaluation

In this section, we evaluate our collaborative scoping approach against the traditional matching pipeline as a pre-processing step and show its effectiveness in pruning unlinkable schema elements. We first describe the experimental setup and introduce different evaluation metrics. All experiments are conducted in a Python Jupyter Notebook on an Intel i7-1265U CPU with 32GB memory. The datasets and code can be found at https://github.com/leotraeg/CollaborativeScoping. Overall, we observe that:

(1) Collaborative scoping always outperforms scoping in effectiveness by up to +26%.
(2) Traditional scoping is ineffective for heterogeneous schema matching scenarios compared to domain-specific ones, while collaborative scoping remains highly robust.
(3) In an ablation study, collaborative scoping boosts matching algorithms in precision (PQ) by up to +80% and F1-measure by up to +20% while remaining more efficient in pair comparisons (RR).

4.1 Experimental Setup

Datasets. We conduct experiments that resemble two distinct multi-source schema matching scenarios. The "OC3" dataset contains a domain-specific set of three schemas to store **O**rder-**C**ustomer data from the three different database vendors Oracle[3], MySQL[4], and SAP HANA[5]. Note that even the three domain-specific OC3 schemas have different numbers of tables and attribute atomicity levels, including elements with no correspondences at all that lead to an unlinkable overhead of 103%. The "OC3-FO" dataset extends the domain-specific schemas with tables and attributes from the official **F**ormula **O**ne[6] schema that

[3]Oracle Schema: https://github.com/oracle-samples/db-sample-schemas
[4]MySQL Schema: https://www.mysqltutorial.org/mysql-sample-database.aspx
[5]SAP HANA Schema: https://developers.sap.com/tutorials/hxe-ua-dbfundamentals-sample-project.html
[6]JOLPICA-F1 Formula One Schema: https://github.com/jolpica/jolpica-f1

Table 2: Overview of linkable and unlinkable schema elements in OC3 and OC3-FO dataset.

Schema (S_k)	Tables	Attributes	Linkable	Unlinkable
OC3	**18**	**142**	**79**	**81**
OC-Oracle	7	43	27	23
OC-MySQL	8	59	34	33
OC-HANA	3	40	18	25
OC3-FO	**34**	**253**	**79**	**208**
Formula One	16	111	0	127

Table 3: Overview of Cartesian product size and annotated linkages between schemas for OC3 and OC3-FO dataset.

Schemas ($\lvert S_k\rvert \cdot \lvert S_m\rvert$)	Cartesian Product Table	Cartesian Product Attr.	II*	IS*
OC3	**101**	**6617**	**39**	**36**
Oracle-MySQL	56	2537	14	22
Oracle-HANA	21	1720	10	8
MySQL-HANA	24	2360	15	1
OC3-FO	**389**	**22379**	**39**	**36**

* **I**nter-**I**dentical and **I**nter-**S**ub-typed linkages (ref. Section 2.1).

is completely unrelated, further increasing the overhead of unlinkable schema elements to 263%. For the latter dataset, schema matching is problematic because even the unrelated schema may contain linkable schema elements (e.g., `DRIVER` could be regarded as a `CLIENT` or `EMPLOYEE`). However, because of the different schema semantics, these kinds of tables and attributes should not be linked to the order and customer domain. The overview of the matching scenarios with the schemas and linkability label distribution is summarized in Table 2. The linkability labels derive from the annotated schema linkages $L(S)$, which are shown in Table 3 with the Cartesian product sizes.

Signature details. We extract the metadata of the schemas, create textual sequences of their tables and attributes, and encode these using Sentence-BERT[7] into 768-dimensional signatures (Section 2.3).

Scoping baselines. We compare collaborative scoping with scoping and four ODAs as baselines. Based on Section 2.4, we compute outlier scores via Scoping with the unified set of table and attribute signatures $S^{\vec{v}}$ among all schemas, which are then sorted and filtered using the threshold parameter $p \in (0..1)$ for streamlined schemas. We implement the following ODAs:

- Z-Score: We implement this method using SciPy[8].
- Local-Outlier-Factor (LOF): We use the sklearn neighbors library[9] and the default number of neighbors $n = 20$.
- Principal-Component-Analysis (PCA): We use the NumPy[10] library for the full Singular Value Decomposition (SVD). As the generalizability of the model is unknown, we experiment with three explained variance levels $v = \{0.3, 0.5, 0.7\}$.
- Autoencoder: We use Keras and configure a fully dense network (768|100|10|100|768) to extend the reconstruction complexity to PCA. We use ReLUs with Adam and the mean-squared error (MSE) as the loss function due to its outlier sensitivity. For a stable result, we initialize and train the autoencoder 100 times and sum up each computed outlier score as a variant of ensemble training, each for 50 epochs.

Collaborative scoping details. We use the identical PCA implementation as for the scoping baseline but computed for each set of local schema signatures (i.e., Algorithm 1 and 2) over the range of explained variance $v \in (1..0)$. With respect to the hardware mentioned above and the OC3 and OC3-FO schemas, the computation of the PCA-based encoder-decoder takes less than a second. To determine the best performance results, several encoder-decoders can be constructed with different explained variance values $v \in (1..0)$.

Matching algorithms. We implement *SIM*, *CLUSTER*, and *LSH* for generating linkages based on the "semantic blocking" methods in Meduri et al. [27]. For each algorithm, we select three different threshold values that cover a large, medium, and small portion of the linkage search space. *SIM* enumerates the entire search space (i.e., Cartesian product size in Table 3), which also corresponds to the "Preparation" module in Zhang et al. [50]. Then, those linkages are pruned, not to exceed the threshold values t_{SIM}={0.4, 0.6, 0.8}. The *CLUSTER* method applies k-Means to the signatures of two schemas with k_{Means}={2, 5, 20} clusters. Subsequently, only those linkages are considered true for signatures grouped in identical clusters [37]. Lastly, *LSH* represents a nearest-neighbor search method that we implement with the FAISS library [19]. We build an IndexFlatL2 for each schema that is searched for top-k_{LSH}={1, 5, 20} similar signatures in another schema. Note that *LSH* is also used by the SOTA *DeepBlocker* [43] method for Entity Resolution [31].

4.2 Evaluation Metrics

Scoping. We measure the effectiveness of scoping streamlined schemas by predicting the linkable (true) or unlinkable (false) label for each table and attribute. Following related studies on schema matching [6, 17, 48], we compute accuracy, precision, recall, and F1-score. As we neither know the optimal value for the scoping parameter $p \in (0..1)$ nor the global explained variance $v \in (1..0)$ in collaborative scoping, a common practice[11] in outlier detection is to measure the Area Under the Curve (AUC) between comparative hyperparameter ranges [1, 26]. Accordingly, we measure the **AUC-F1** as a summarizing metric.

Due to the binary class nature of the linkability problem, in which the parameter values heavily influence the scoping performance, we additionally use the AUC of the Receiver Operating Characteristic (**AUC-ROC**). We note that in collaborative scoping, some table and attribute signatures will be reconstructed with an error that is too high to be considered linkable by any other schema encoder-decoder, regardless of the explained variance. In these cases, both the false positive rate (FPR) and true positive rate (TPR) may never reach 100% for any explained variance $v \in (1..0)$. Unfortunately, FPR never reaching 100% negatively affects the AUC-ROC score even though it is a favorable model characteristic. In essence, the quality of a linkability examination

[7] Sentence-BERT pre-trained with all-mpnet-base-v2 (https://huggingface.co/sentence-transformers/all-mpnet-base-v2) is reported as the best general-purpose model (https://www.sbert.net/docs/sentence_transformer/pretrained_models.html).
[8] SciPy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html
[9] LOF: https://scikit-learn.org/stable/api/sklearn.neighbors.html
[10] SVD: https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html
[11] Sklearn evaluation of outlier detection estimators: https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_outlier_detection_bench.html

Table 4: AUC-F1, AUC-ROC, AUC-ROC', and AUC-PR performance of scoping methods with OC3 and OC3-FO schemas.

Methods	ODA	OC3				OC3-FO			
		AUC-F1	AUC-ROC	AUC-ROC'[3]	AUC-PR	AUC-F1	AUC-ROC	AUC-ROC'[3]	AUC-PR
Scoping[1] $p \in (0..1)$	Z-Score	51.64	61.67	64.07	61.93	35.52	55.51	56.39	35.51
	LOF ($n = 20$)	52.52	63.24	66.00	61.79	36.76	56.15	57.79	35.49
	PCA ($v = 0.3$)	55.24	67.71	70.34	67.01	47.04	72.88	74.25	**57.98**
	PCA ($v = 0.5$)	**58.27**	**73.64**	**76.40**	**71.77**	**47.97**	**75.55**	**77.09**	55.22
	PCA ($v = 0.7$)	54.40	66.72	69.23	65.53	40.30	64.11	65.56	39.34
	Autoencoder	55.68	68.90	71.56	67.73	45.79	72.00	73.33	52.54
Collaborative[1] $v \in (1..0)$	PCA	**61.82**	**64.26**	**82.52**	**76.39**	**50.45**	**62.36**	**92.80**	**73.71**
Difference[2]		*+6.10%*	*-12.73%*	*+8.01%*	*+6.44%*	*+5.17%*	*-17.46%*	*+20.37%*	*+27.11%*

[1] The best AUC scores per scoping method are in **bold**.

[2] The percentual improvement or decline between the best AUC scores with scoping and collaborative scoping are in *italic*.

[3] Monotonically sorted and interpolated ROC curve via smoothing[12] to compare model performances with FPR < 100% fairly.

is represented by the fact of how quickly the ROC curve converges to a high TPR, which the overall AUC approximates. In order to diminish the effect of a model never reaching FPR=100%, we monotonically sort and interpolate[12] the ROC curve as ROC' and compute the respective **AUC-ROC'**.

Thirdly, in order to account for the imbalances between the linkable versus unlinkable schema elements (i.e., unlinkable overhead > 100%), we compute the AUC of the precision-recall curve (**AUC-PR**). Due to its focus on the positive class (linkable), we use it as the primary metric to enable a fair comparison between the imbalanced unlinkable overheads with the OC3 and OC3-FO scenarios.

Matching. We evaluate the performance of linkage generation by a matching algorithm A using streamlined schemas S' with the annotated set of ground truth linkages $L(S)$ (ref. Table 3). We focus on match effectiveness with the standard metrics [3, 31, 40] precision as the Pair Quality (PQ) = $|A(S') \cap L(S)|/|A(S')|$, recall as the Pair Completeness (PC) = $|A(S') \cap L(S)|/|L(S)|$, and $F1 = \frac{2 \cdot (PQ \cdot PC)}{(PQ \cdot PC)}$ as the harmonic mean between PC and PQ. We measure the number of comparisons by the matching algorithm with respect to the Cartesian product size of the original schemas as the Reduction Ratio (RR) = $1 - \frac{|A(S')|}{|S_1| \cdot |S_2| \cdot \ldots \cdot |S_k|}$.

4.3 Results

We evaluate the collaborative scoping performance compared to the scoping baselines. The AUC scores of the F1, ROC, and PR measures across the OC3 and OC3-FO schemas are reported in Table 4. We plot the best-performing scoping (a, c, e) and collaborative scoping (b, d, f) methods for the OC3 schemas in Figure 5 and OC3-FO schemas in Figure 6. In the first row, we show the accuracy (blue dashed), precision (green dotted), recall (red dash-dotted), and F1 (black solid) on the y-axis for the scoping method (a) with growing $p \in (0..1)$ and collaborative scoping (b) with decreasing $v \in (1..0)$ on the x-axis. In the second row, we show the ROC (black solid) and smoothed ROC' (blue dashed) curves for scoping (c) and collaborative scoping (d). In the last row, we show the PR curves for scoping (e) and collaborative scoping (f).

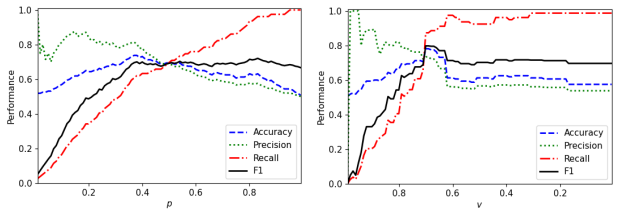
[12]Python library splrep: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.splrep.html (applied smoothing factor $s = 0.2$).

Scoping. As observed in Table 4, scoping with PCA as ODA significantly outperforms the Z-Score and LOF by +13-63% for both OC3 and OC3-FO schemas. PCA ($v = 0.5$) retrieves the best results that surpass the AUC scores of the non-linear autoencoder for both the OC3 and OC3-FO schemas by +4-6%. However, it is evident that the parameter setting of a single ODA influences the outlier scores. Thus, a different v parameterization leads to better or worse scoping performance, such as for the best AUC-PR score for OC3 with PCA ($v = 0.5$) and OC3-FO schemas with PCA ($v = 0.3$).

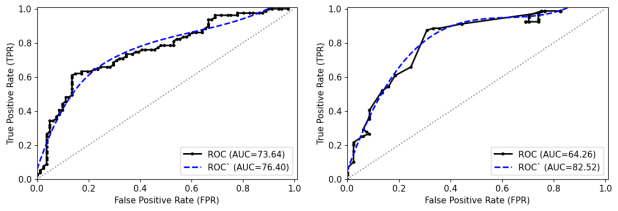
When comparing the performance curves of scoping with PCA ($v = 0.5$) for OC3 (Figure 5 (a)) and OC3-FO (Figure 6 (a)) schemas, we observe a significant drop in precision. Generally, precision declines steadily as p increases in scoping. We attribute the worse performance in precision with OC3-FO schemas to the interference of the Formula One tables and attributes, leading to a considerably lower AUC-F1 (a) score by -18% and AUC-PR (e) score by -19%. Conversely, AUC-ROC (c) remains relatively stable between OC3 and OC3-FO (+3%), highlighting its insensitivity to the linkability class imbalance. Scoping OC3-FO schemas shows a stronger acceleration of recall than for OC3, both stabilizing at $p \approx 0.9$ to 100%.

Notably, all scoping methods, except PCA ($v = \{0.3, 0.5\}$) and the autoencoder, perform worse than randomly guessing (i.e., <50 in AUC-PR) for OC3-FO. We relate this performance decline and change in prediction behavior from OC3 to OC3-FO to the Formula One table and attribute signatures occupying the models' normal distribution mean (ref. Figure 3).

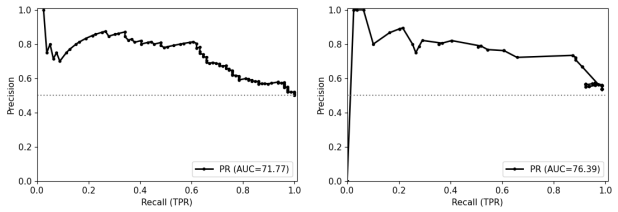
Collaborative scoping. As observed in Table 4, our new approach remains quite robust in terms of AUC-ROC (-3%) and AUC-PR (-4%) for the additional Formula One tables and attributes that cannot be linked. However, this method struggles with maintaining consistent precision and recall, as measured by AUC-F1, showing a decrease of -18% compared to the OC3 and OC3-FO schemas. When comparing the performance curves from the collaborative PCA method for OC3 (Figure 5 (b)) and OC3-FO (Figure 6 (b)) schemas, we observe nearly identical performance trajectories in precision and recall within the explained variance range $1 > v \geq 0.7$. However, performance differences between the two schema-matching scenarios become apparent for $v < 0.7$. We highlight that both the precision and recall curves fluctuate throughout the range of explained variance $v \in (1..0)$ due to



(a) Scoping PCA ($v = 0.5$): F1 (b) Collaborative Scoping PCA: F1

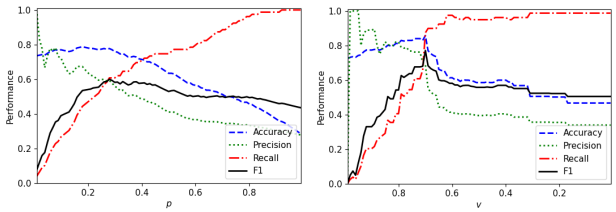


(c) Scoping PCA ($v = 0.5$): ROC (d) Collaborative Scoping PCA: ROC

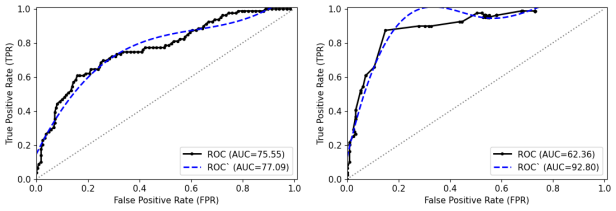


(e) Scoping PCA ($v = 0.5$): PR (f) Collaborative Scoping PCA: PR

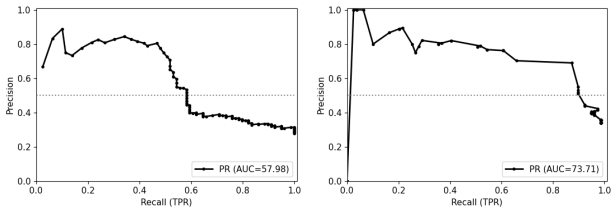
Figure 5: Best performing scoping methods in AUC-F1, AUC-ROC, and AUC-PR for OC3 schemas.



(a) Scoping PCA ($v = 0.5$): F1 (b) Collaborative Scoping PCA: F1



(c) Scoping PCA ($v = 0.5$): ROC (d) Collaborative Scoping PCA: ROC



(e) Scoping PCA ($v = 0.3$): PR (f) Collaborative Scoping PCA: PR

Figure 6: Best performing scoping methods in AUC-F1, AUC-ROC, and AUC-PR for OC3-FO schemas.

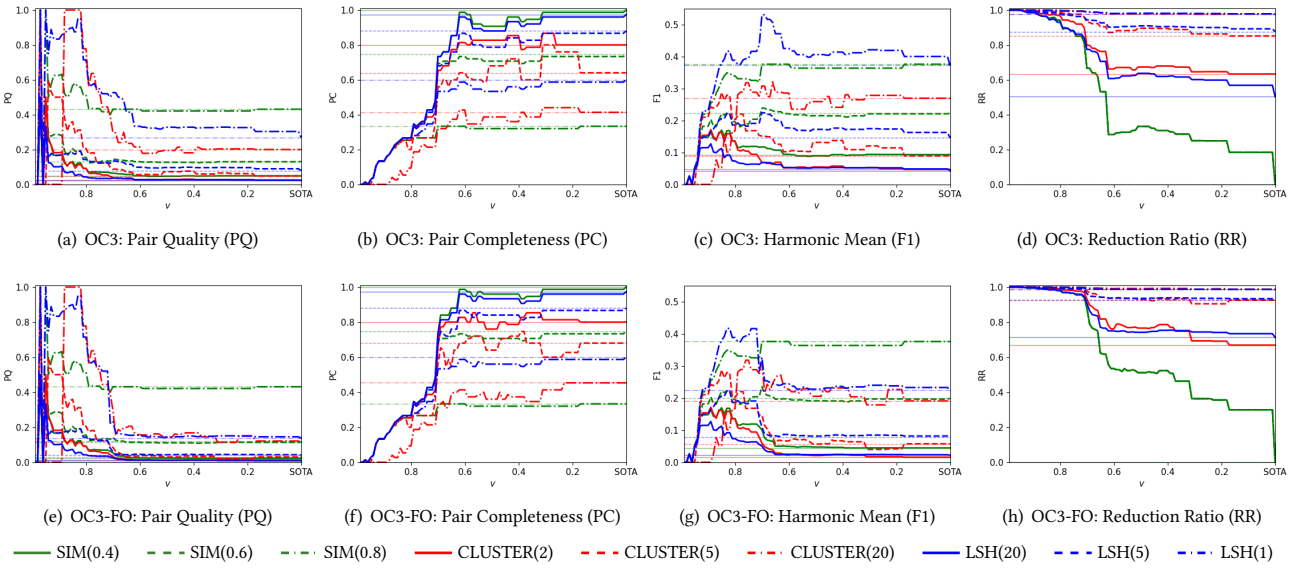
the dependent local linkability range l_k of an encoder-decoder model M_k. Lower explained variance results in underfitting the local encoder-decoders, leading to incorrectly tolerating higher reconstruction errors of unlinkable tables and attributes as linkable. Consequently, precision decreases in trade-off with minor improvements in recall. The AUC-F1 improvement is more significant for the OC3-FO scenario than OC3.

Regarding the AUC-ROC score in Table 4, the collaborative PCA scores for OC3 and OC3-FO schemas are quite similar. At a closer look, the ROC curve for OC3-FO (Figure 6 (b)) rises more steeply at lower FPR ranges compared to the OC3 schemas (Figure 5 (b)). Secondly, the maximum FPR for OC3-FO is $FPR \approx 75\%$, whereas it exceeds $FPR > 80\%$ for OC3 schemas, indicating improved prediction behavior for OC3-FO schemas. Although we interpret the AUC-ROC scores with caution due to the imbalance of label classes not being considered, the smoothed ROC curves suggest a further +13% improvement in AUC-ROC' scores for OC3-FO schemas.

As FPR can remain well below 100% in collaborative scoping throughout the explained variance $v \in (1..0)$ range, so may the recall/true positive rate (TPR) be affected (recall (b) or TPR (d) in Figure 5 and 6). For example, in collaborative scoping applied to both OC3 and OC3-FO schemas at $v \leq 0.3$, the reconstruction of the encoded attribute text sequence `ORDERDATE ORDERS DATE` from schema $S_{\text{OC-MySQL}}$ ends up not passing the model $M_{\text{OC-Oracle}}$ as linkable even though its schema $S_{\text{OC-Oracle}}$ contains a linkable attribute with the text sequence `ORDER_DATETIME ORDERS DATE`. Due to their semantic similarity, these two attributes are annotated as an inter-sub-typed linkage in the ground truth set $L(S)$. However, the little nuanced differences (i.e., _ and `TIME`) encode a deviating signature that causes this false negative at any given v. While this case has only occurred to a single element in both schema matching scenarios, this is a limitation of collaborative scoping. Pruning false negatives ahead of matching will cause associated true linkages to remain undiscovered.

Comparison. Collaborative scoping outperforms all scoping baselines (except AUC-ROC discussed in Section 4.2). For the domain-specific schemas in OC3, collaborative scoping yields +6% higher scores for AUC-F1 and AUC-PR and performs +8% better in AUC-ROC' compared to the best-performing scoping method with PCA ($v = 0.5$). For the more challenging OC3-FO matching scenario with a 2.6× unlinkable overhead, collaborative scoping further excels in AUC-F1 by +5%, AUC-ROC' by +21%, and AUC-PR by +27%.

Notably, the performance curves for collaborative scoping (Figure 5 and 6 (b)) fluctuate throughout the explained variance ranges $v \in (1..0)$ compared to scoping (Figure 5 and 6 (a)) that monotonically increase and decrease with the relative threshold $p \in (0..1)$. The fluctuating performance in collaborative scoping stems from each local encoder-decoder M_k maintaining its own linkability range l_k guided by the explained variance v rather than the global threshold p in scoping. Carefully considering AUC-ROC' alongside the improved AUC-F1 and AUC-PR performances, collaborative scoping preserves the local schema context and, therewith, remains robust in pruning unlinkable schema elements across both matching scenarios regardless of the different overheads.



(a) OC3: Pair Quality (PQ) (b) OC3: Pair Completeness (PC) (c) OC3: Harmonic Mean (F1) (d) OC3: Reduction Ratio (RR)

(e) OC3-FO: Pair Quality (PQ) (f) OC3-FO: Pair Completeness (PC) (g) OC3-FO: Harmonic Mean (F1) (h) OC3-FO: Reduction Ratio (RR)

Figure 7: Ablation study for matching OC3 & OC3-FO schemas with collaborative scoping on PQ, PC, F1, and RR.

Ablation study with matching algorithms. We select collaborative scoping over scoping as a more effective and efficient pre-processing method (analysis in Section 3). Figure 7 shows the PQ, PC, F1, and RR performance (y-axis) for matching OC3 (a-d) and OC3FO (e-h) schemas with collaborative scoping S' over the explained variance range $v \in (1..0)$ (x-axis). As an ablation study, we also apply the matching algorithms on the original schemas S without any pre-processing (represented as SOTA in x-axis=0), with each matching algorithm's performance drawn as a thin horizontal baseline. *SIM* is shown in green with the cosine threshold values t_{SIM} = {0.4 (solid), 0.6 (dashed), 0.8 (dash-dot)}, *CLUSTER* in red with the cluster numbers k_{Means} = {2 (solid), 5 (dashed), 20 (dash-dot)}, and *LSH* in blue with top-k_{LSH} = {1 (solid), 5 (dashed), 20 (dash-dot)} lines.

Pair Quality. With collaborative scoping, all matching algorithms generate significantly fewer false positive linkages. Hence, a global similarity or cardinality threshold alone is not able to prune false positive linkages effectively. For the variance values $v > 0.7$, the PQ reaches up to 100% and remains consistent for both OC3 (a) and OC3-FO (e) schemas, showing its robustness to heterogeneous schema matching scenarios. At $v > 0.6$, *CLUSTER(20)* improves by up to +80%, *LSH(20)* by up to +70%, and *SIM(0.8)* by up to +30%. At $v < 0.6$, the PQ performance saturates on par with SOTA while all *LSH* parameterizations consistently outperform it. *Thus, collaborative scoping considerably boosts PQ for all values of v.*

Pair Completeness. Collaborative scoping primarily enhances PQ, whereas the PC metric reflects the trade-off risk of pruning linkable elements ahead of matching. At $v \approx 0.6$ and again at $v < 0.35$, all matching algorithms achieve near-SOTA performance (within 1% difference) for both OC3 (b) and OC3-FO (f) scenarios. Only *SIM(0.4)* followed by *LSH(20)* retain enough linkages to achieve near full PC, but this comes with the cost of large expansions in the search space. Notably, *CLUSTER(2)* and *CLUSTER(5)* perform better than SOTA at certain variance values $v < 0.6$. *However, this optimized PC score is achieved at the cost of a lower PQ score, which is typically not desired.*

F1. In general, most matching algorithms benefit from collaborative scoping and improve F1. Thus, streamlining schemas before matching results in a considerable gain of *PQ* compared to the minor losses in *PC*. Matching heterogeneous OC3-FO schemas is more error-prone compared to matching homogeneous OC3 schemas. In particular, *LSH(1)* improves F1 with up to +15% for OC3 (c) and +20% for OC3-FO (g) schemas at $v < 0.95$. The only two exceptions are the *SIM(0.8)* and *SIM(0.6)* parameterizations that only reach on-par with SOTA at $v < 0.7$ as they perform an exact search through the expensive Cartesian product space.

Reduction Ratio. We consistently reduce the number of comparisons using the streamlined schemas compared to using the original ones. The higher the global explained variance, the fewer schema elements are assigned as linkable, reducing the search space. The boost in efficiency has more impact on matching algorithms that extend the linkage search space from medium to large magnitudes. Notably, even the lowest variance value $v = 0.01$ prunes 9.37% (15) elements in OC3 (d) and 19.86% (57) in OC3-FO (h) schemas. Given the PQ and PC graphs, with the exception of one, all these schema elements are irrelevant comparison candidates (true negative unlinkable elements) pruned ahead of matching via collaborative scoping.

4.4 Discussion

Setting variance value. The global variance v is a key determinant in collaborative scoping that automatically generates an independent linkability function and threshold for each local schema. Our experiments indicate that when $v \in [0.95, 0.6]$, collaborative scoping effectively prunes unlinkable schema elements while balancing high precision with high recall. The method demonstrated robustness for pruning the different overheads of unlinkables by achieving identical performance results for $v > 0.7$ for the domain-specific (OC3: 103%) and heterogeneous (OC3-FO: 263%) scenarios. As a pre-processing phase, we believe setting v as a global model parameter for local pruning is advantageous because it enables determining the pair quality and completeness of linkages. In contrast, post-matching with a global similarity or cardinality threshold cannot account for the

local schema semantics, such as the mismatching of the Formula One schema in Figure 1 (e.g., client `CITY` ≠ car `COUNTRY`).

Pre-processing trade-off. Collaborative scoping requires each schema to independently self-train a PCA-based encoder-decoder with $O(|S_k|^2)$ as the additional cost. While avoiding element-wise comparisons, all schema elements must be passed through the encoder-decoders of the other schemas $O(|S| \cdot |M|)$. However, the number of encoder-decoder pass operations is relatively small compared to the Cartesian product size $O(|S_1| \cdot |S_2| \cdot \ldots \cdot |S_k|)$ (ref. Table 3). Specifically, that is 4.76% (320) for OC3 and 3.78% (861) for OC3-FO schemas. In general, the higher the number of schemas, the lower this percentage.

Limitations. First, collaborative scoping does not guarantee retaining all linkable elements, and sometimes may allow false negatives. Schema elements classified as unlinkable need to be carefully evaluated, particularly for matching scenarios with a strict requirement for Pair Completeness. Secondly, our approach heavily relies on schema metadata, which, in general, is obtainable but may sometimes be incomplete or may not exist. Regardless, our approach avoids matching unlinkable schemas without any meaningful semantics provided, making collaborative scoping a fault-tolerant pruning solution.

5 Conclusion

In this paper, we discussed *schema linkability* as a practical precursor problem for multi-source schema matching, facing the challenge of distinguishing linkable from unlinkable schema elements. Our *collaborative scoping* method aims to solve this problem by capturing the independent *semantics* of a schema with *self-supervised encoder-decoder models*. Subsequently, the models of other schemas are used in order to locally assess the schema linkability and scope a *streamlined schema*. This approach significantly differs from matching methods that apply global similarity or cardinality thresholds, which are locally self-tuned with collaborative scoping. Evaluations show that our approach remains robust for pruning unlinkable elements when matching heterogeneous schemas and performs better than prior work. This way, the linkage quality and F1 performance of schema matching significantly improve. At the same time, the number of element-wise comparisons is inherently reduced. As part of future work, we plan to extend encoder-decoders in order to recognize non-linear signature patterns and experiment with the overall applicability in entity resolution and ontology alignment [16].

Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. Leonard Traeger was partially supported by the Technology Catalyst Fund (TCF24KAR11131049602) by UMBC and PLan_CV (03FHP109) by the German Federal Ministry of Research, Technology and Space (BMFTR) and Joint Science Conference (GWK).

Artifacts

The OC3-FO dataset with the extracted and serialized schemas, the annotated linkages used to derive the linkability labels, the executable Python Jupyter notebook `collaborative_scoping.ipynb`, and the reported experimental results are publicly available and accessible in the GitHub repository https://github.com/leotraeg/CollaborativeScoping. The `README.md` file provides a description of the datasets along with a quick-start guide to the implementation and performance metrics.

References

[1] Charu C. Aggarwal. 2017. An Introduction to Outlier Analysis. In *Outlier Analysis*, Charu C. Aggarwal (Ed.). Springer International Publishing, Cham, 1–34. doi:10.1007/978-3-319-47578-3_1

[2] David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. 2005. Schema and ontology matching with COMA++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, Baltimore Maryland, 906–908. doi:10.1145/1066157.1066283

[3] Daniel Ayala, Inma Hernández, David Ruiz, and Erhard Rahm. 2022. LEAPME: Learning-based Property Matching with Embeddings. *Data & Knowledge Engineering* 137 (Jan. 2022), 101943. doi:10.1016/j.datak.2021.101943

[4] Fabio Azzalini, Songle Jin, Marco Renzi, and Letizia Tanca. 2021. Blocking Techniques for Entity Linkage: A Semantics-Based Approach. *Data Science&Engineering* 6.1 (March 2021), 20–38. doi:10.1007/s41019-020-00146-w

[5] Dor Bank, Noam Koenigstein, and Raja Giryes. 2021. Autoencoders. http://arxiv.org/abs/2003.05991 arXiv:2003.05991.

[6] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegrakis. 2011. On Evaluating Schema Matching and Mapping. In *Schema Matching and Mapping*, Zohra Bellahsene, Angela Bonifati, and Erhard Rahm (Eds.). Springer, Berlin, Heidelberg, 253–291. doi:10.1007/978-3-642-16518-4_9

[7] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. *ACM SIGMOD Record* 29, 2 (June 2000), 93–104. doi:10.1145/335191.335388

[8] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1335–1349. doi:10.1145/3318464.3389742

[9] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A Data Discovery System. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, Paris, 1001–1012. doi:10.1109/ICDE.2018.00094

[10] Raul Castro Fernandez, Essam Mansour, Abdulhakim A. Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping Semantics: Linking Datasets Using Word Embeddings for Data Discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, Paris, 989–1000. doi:10.1109/ICDE.2018.00093

[11] Martin P Christensen, Aristotelis Leventidis, Matteo Lissandrini, Laura Di Rocco, Renée J. Miller, and Katja Hose. 2025. Fantastic Tables and Where to Find Them: Table Search in Semantic Data Lakes. doi:10.48786/EDBT.2025.32

[12] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. TURL: Table Understanding through Representation Learning. *SIGMOD Rec.* 51, 1 (June 2022), 33–40. doi:10.1145/3542700.3542709

[13] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Pena, Aécio Santos, Cláudio Silva, and Eden Wu. 2025. Large Language Models for Data Discovery and Integration: Challenges and Opportunities. *Data Engineering* (2025), 3. http://sites.computer.org/debull/A25mar/A25MAR-CD.pdf#page=5

[14] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. 2019. Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. doi:10.48550/arXiv.1904.02639 arXiv:1904.02639.

[15] Bin He and Kevin Chen-Chuan Chang. 2005. Making holistic schema matching robust: an ensemble approach. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, Chicago Illinois USA, 429–438. doi:10.1145/1081870.1081920

[16] Yuan He, Jiaoyan Chen, Hang Dong, Ernesto Jiménez-Ruiz, Ali Hadian, and Ian Horrocks. 2022. Machine Learning-Friendly Biomedical Datasets for Equivalence and Subsumption Ontology Matching. In *The Semantic Web – ISWC 2022*, Ulrike Sattler, Aidan Hogan, Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato (Eds.). Vol. 13489. Springer International Publishing, Cham, 575–591. doi:10.1007/978-3-031-19433-7_33 Series Title: Lecture Notes in Computer Science.

[17] Benjamin Hättasch, Michael Truong-Ngoc, Andreas Schmidt, and Carsten Binnig. 2022. It's AI Match: A Two-Step Approach for Schema Matching Using Embeddings. doi:10.48550/arXiv.2203.04366 arXiv:2203.04366.

[18] Ihab F. Ilyas and Theodoros Rekatsinas. 2022. Machine Learning and Data Cleaning: Which Serves the Other? *Journal of Data and Information Quality* 14, 3 (July 2022), 13:1–13:11. doi:10.1145/3506712

[19] Jeff Johnson, Matthijs Douze, and Herve Jegou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (July 2021), 535–547. doi:10.1109/TBDATA.2019.2921572

[20] Firuz Kamalov and Ho Hon Leung. 2020. Outlier Detection in High Dimensional Data. *Journal of Information & Knowledge Management* 19, 01 (March 2020), 2040013. doi:10.1142/S0219649220400134 Publisher: World Scientific Publishing Co..

[21] Aamod Khatiwada, Roee Shraga, Wolfgang Gatterbauer, and Renée J. Miller. 2022. Integrating Data Lake Tables. *Proc. VLDB Endow.* 16, 4 (Dec. 2022), 932–945. doi:10.14778/3574245.3574274

[22] Enas Khwaileh and Yannis Velegrakis. 2025. Dataset Discovery using Semantic Matching. doi:10.48786/EDBT.2025.52

[23] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating Matching Techniques for Dataset Discovery. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. 468–479. doi:10.1109/ICDE51399.2021.00047 ISSN: 2375-026X.

[24] Michael Loster, Ioannis Koumarelas, and Felix Naumann. 2021. Knowledge Transfer for Entity Resolution with Siamese Neural Networks. *Journal of Data and Information Quality* 13, 1 (March 2021), 1–25. doi:10.1145/3410157

[25] Jayant Madhavan, Philip A Bernstein, and Erhard Rahm. 2001. Generic Schema Matching with Cupid. *VLDB* (2001).

[26] Henrique O. Marques, Lorne Swersky, Jörg Sander, Ricardo J. G. B. Campello, and Arthur Zimek. 2023. On the evaluation of outlier detection and one-class classification: a comparative study of algorithms, model selection, and ensembles. *Data Mining and Knowledge Discovery* 37, 4 (July 2023), 1473–1517. doi:10.1007/s10618-023-00931-x

[27] Venkata Vamsikrishna Meduri, Abdul Quamar, Chuan Lei, Xiao Qin, and Berthold Reinwald. 2024. Alfa: active learning for graph neural network-based semantic schema alignment. *The VLDB Journal* 33, 4 (July 2024), 981–1011. doi:10.1007/s00778-023-00822-z

[28] S. Melnik, H. Garcia-Molina, and E. Rahm. 2002. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *Proceedings 18th International Conference on Data Engineering*. IEEE Comput. Soc, San Jose, CA, USA, 117–128. doi:10.1109/ICDE.2002.994702

[29] Franziska Neuhof, Marco Fisichella, George Papadakis, Konstantinos Nikoletos, Nikolaus Augsten, Wolfgang Nejdl, and Manolis Koubarakis. 2024. Open benchmark for filtering techniques in entity resolution. *The VLDB Journal* 33, 5 (Sept. 2024), 1671–1696. doi:10.1007/s00778-024-00868-7

[30] George Papadakis, Marco Fisichella, Franziska Schoger, George Mandilaras, Nikolaus Augsten, and Wolfgang Nejdl. 2022. How to reduce the search space of Entity Resolution: with Blocking or Nearest Neighbor search? http://arxiv.org/abs/2202.12521 arXiv:2202.12521.

[31] George Papadakis, Marco Fisichella, Franziska Schoger, George Mandilaras, Nikolaus Augsten, and Wolfgang Nejdl. 2023. Benchmarking Filtering Techniques for Entity Resolution. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. 653–666. doi:10.1109/ICDE55515.2023.00389 ISSN: 2375-026X.

[32] George Papadakis, Ekaterini Ioannou, Themis Palpanas, Claudia Niederée, and Wolfgang Nejdl. 2013. A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces. *IEEE Transactions on Knowledge and Data Engineering* 25, 12 (Dec. 2013), 2665–2682. doi:10.1109/TKDE.2012.150 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[33] George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, and Manolis Koubarakis. 2020. Domain- and Structure-Agnostic End-to-End Entity Resolution with JedAI. *ACM SIGMOD Record* 48, 4 (Feb. 2020), 30–36. doi:10.1145/3385658.3385664

[34] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. doi:10.48550/ARXIV.1908.10084 Version Number: 1.

[35] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. 2021. A Unifying Review of Deep and Shallow Anomaly Detection. *Proc. IEEE* 109, 5 (May 2021), 756–795. doi:10.1109/JPROC.2021.3052449 arXiv:2009.11732.

[36] Alieh Saeedi, Lucie David, and Erhard Rahm. 2021. Matching Entities from Multiple Sources with Hierarchical Agglomerative Clustering:. In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SCITEPRESS - Science and Technology Publications, 40–50. doi:10.5220/0010649600003064

[37] Tanvi Sahay, Ankita Mehta, and Shruti Jadon. 2020. Schema Matching using Machine Learning. In *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*. 359–366. doi:10.1109/SPIN48934.2020.9071272 ISSN: 2688-769X.

[38] Eitam Sheetrit, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. ReMatch: Retrieval Enhanced Schema Matching with LLMs. doi:10.48550/arXiv.2403.01567 arXiv:2403.01567.

[39] Jonathon Shlens. 2014. A Tutorial on Principal Component Analysis. doi:10.48550/arXiv.1404.1100 arXiv:1404.1100.

[40] Roee Shraga and Avigdor Gal. 2021. PoWareMatch: a Quality-aware Deep Learning Approach to Improve Human Schema Matching. doi:10.48550/arXiv.2109.07321 arXiv:2109.07321.

[41] Roee Shraga, Avigdor Gal, and Haggai Roitman. 2020. ADnEV: cross-domain schema matching using deep similarity matrix adjustment and evaluation. *Proc. VLDB Endow.* 13, 9 (May 2020), 1401–1415. doi:10.14778/3397230.3397237

[42] Michael Stonebraker and Ihab F Ilyas. 2018. Data Integration: The Current Status and the Way Forward. *IEEE* Data Eng. Bull., 41 (2018), 3–9.

[43] Saravanan Thirumuruganathan, Han Li, Nan Tang, Mourad Ouzzani, Yash Govind, Derek Paulsen, Glenn Fung, and AnHai Doan. 2021. Deep learning for blocking in entity matching: a design space exploration. *Proceedings of the VLDB Endowment* 14, 11 (July 2021), 2459–2472. doi:10.14778/3476249.3476294

[44] Leonard Traeger, Andreas Behrend, and George Karabatis. 2025. Collective Scoping: Streamlining Entity Sets Towards Efficient and Effective Entity Linkages. *SN Computer Science* 6, 3 (Feb. 2025), 238. doi:10.1007/s42979-025-03734-7

[45] Leonard Traeger, Andreas Behrend, and George Karabatis. 2025. SEALM: Semantically Enriched Attributes with Language Models for Linkage Recommendation:. In *Proceedings of the 27th International Conference on Enterprise Information Systems*. SCITEPRESS - Science and Technology Publications, Porto, Portugal, 39–50. doi:10.5220/0013217700003929

[46] Yasi Wang, Hongxun Yao, and Sicheng Zhao. 2016. Auto-encoder based dimensionality reduction. *Neurocomputing* 184 (April 2016), 232–242. doi:10.1016/j.neucom.2015.08.104

[47] Alexandros Zeakis, George Papadakis, Dimitrios Skoutas, and Manolis Koubarakis. 2023. Pre-Trained Embeddings for Entity Resolution: An Experimental Analysis. *Proceedings of the VLDB Endowment* 16, 9 (May 2023), 2225–2238. doi:10.14778/3598581.3598594

[48] Xiaocan Zeng, Pengfei Wang, Yuren Mao, Lu Chen, Xiaoze Liu, and Yunjun Gao. 2024. MultiEM: Efficient and Effective Unsupervised Multi-Table Entity Matching. IEEE Computer Society, 3421–3434. doi:10.1109/ICDE60146.2024.00264

[49] Zezhou Huang, Guo, Jia, and Wu, Eugene. 2024. Transform Table to Database Using Large Language Models. https://tabular-data-analysis.github.io/tada2024/papers/TaDA.6.pdf

[50] Yunjia Zhang, Avrilia Floratou, Joyce Cahoon, Subru Krishnan, Andreas C. Müller, Dalitso Banda, Fotis Psallidas, and Jignesh M. Patel. 2023. Schema Matching using Pre-Trained Language Models. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, Anaheim, CA, USA, 1558–1571. doi:10.1109/ICDE55515.2023.00123