IJDC | Conference Paper

T-KAER: Transparency-enhanced Knowledge-Augmented Entity Resolution Framework

Lan Li[1], Liri Fang[1], Yiren Liu[2], Vetle I. Torvik[1], and Bertram Ludäscher[1,3]

[1]School of Information Sciences, University of Illinois Urbana Champaign

[2]Informatics, University of Illinois Urbana Champaign

[3]National Center for Supercomputing Applications (NCSA)

Abstract

Entity resolution (ER) is the process of determining whether two representations refer to the same real-world entity and plays a crucial role in data curation and data cleaning. Recent studies have introduced the KAER framework, aiming to improve pre-trained lan-guage models by augmenting external knowledge. However, identifying and documenting the external knowledge that is being augmented and understanding its contribution to the model's predictions have received little to no attention in the research community. This pa-per addresses this gap by introducing T-KAER, the **T**ransparency-enhanced **K**nowledge-**A**ugmented **E**ntity **R**esolution framework.

To enhance transparency, three Transparency-related Questions (T-Qs) have been pro-posed: T-Q(1): What is the experimental process for matching results based on data in-puts? T-Q(2): Which semantic information does *KAER* augment in the raw data inputs?T-Q(3): Which semantic information of the augmented data inputs influences the predic-tions? To address the T-Qs, T-KAER is designed to improve transparency by documenting the entity resolution processes in log files.

In experiments, a citation dataset is used to demonstrate the transparency components of T-KAER. This demonstration showcases how T-KAER facilitates error analysis from both quantitative and qualitative perspectives, providing evidence on "what" semantic informa-tion is augmented and "why" the augmented knowledge influences predictions differently.

Keywords: Entity Resolution · Pre-trained Language Model · Transparency · Knowledge augmentation · T-KAER

Submitted 9 February 2024 ~ *Accepted* 22 February 2024

Correspondence should be addressed to Lan Li. Email: lanl2@illinois.edu

This paper was presented at International Journal of Digital Curation IDCC24, 19-21 February 2024.

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

Introduction and Overview

The FAIR guiding principles for scientific data aim to ensure that data is *findable*, *accessible*, *interoperable*, and *reusable* (Wilkinson et al., 2016). The concept of research *transparency* follows as one way to improve reusability of data and the reproducibility of results (Nosek et al., 2015; McPhillips et al., 2019). In addition to reproducibility, transparency in the research process is always essential to check research integrity, identify fraud, and track retractions (Lyon, 2016). Essentially, a transparent research process leads to greater trustworthiness by enabling researchers to easily track and verify internal products and understand mechanisms, even without re-running.

In this paper, we explore transparency in entity resolution (ER), the problem of determining whether two separate representations refer to the same real-world entity, regardless of whether they exist within the same database or span across different databases (Christen, 2012). ER helps reconcile data inconsistencies and eliminate duplicates during data integration, playing a crucial role in data curation. Consequently, there is an increasing demand for a reliable and user-friendly entity resolution tool that minimizes the effort required by data curators. ER is also referred to as *deduplication* (Koumarelas et al., 2020). Existing data cleaning tools, such as OpenRefine (OR, 2021), use traditional machine learning algorithms like K-Nearest Neighbors (KNN) to detect duplications. Nowadays, deep learning techniques (Li et al., 2021b), pre-trained language models (PLMs) (Li et al., 2020; Li et al., 2021a; Paganelli et al., 2022), and even large language models (LLMs)(Peeters and Bizer, 2023), have been deployed to tackle entity resolution.

Why augment PLMs with domain knowledge for ER? Many existing methods of entity resolution hypothesize that records of data follow the same known schema (Elmagarmid et al., 2007). However, this is not always the case in real-world applications. Raw data is often collected from sources that are highly heterogeneous and do not share a common schema. The source data can also come from multiple domains, and it may be represented in diverse formats. The complexity involved makes it challenging for data curators to perform entity resolution without specialized knowledge about the data's domain. For instance, a pair of records from the citation domain might include title, author names, venue name, and publication year. The potential challenges for the model to understand citation data include: (1). Not all attributes hold equal importance; title, venue name, and publication year matter more than author names. (2). There are two layers of semantic information in author names: values and order. Even if the spellings for author names are the same, different orders might result in a **not-match**. Additionally, authors' names may be presented in different formats based on the citation type, such as the position of the first name and last name or acronyms of the names. Hence, it is beneficial for entity resolution methods to integrate additional semantic information into the source data.

What are the challenges of using PLMs in ER? As emphasized in (Li et al., 2021c), the responsible management of data requires that algorithms used in entity resolution tasks be explainable: This means, it is particularly critical for comprehending the reasons behind matching entities and, equally important, why certain entities are not considered a match. One shared challenge faced by applications based on these PLMs is their "black-box" nature. Despite certain transformer-based language models having open-access architectures, the lack of transparency in their pretraining data and processes can lead to

a loss of understanding regarding why various data inputs resulted in the final matching results.

How to enable Transparency in ER? As mentioned in (Grafberger et al., 2023), "provenance is all you need", suggesting that enabling provenance tracking can automate the detection of many common correctness issues in machine learning pipelines. Specifically, they emphasize the use of *why-provenance* to determine which data inputs were used to compute specific data outputs. Furthermore, we invite the concept of *where-provenance* (Buneman et al., 2001) to describe which semantic information of the augmented data inputs influences the entity resolution results. By explaining three T-Qs related to the augmentation of additional domain knowledge for matching results, we aim to provide a more transparent entity resolution process.

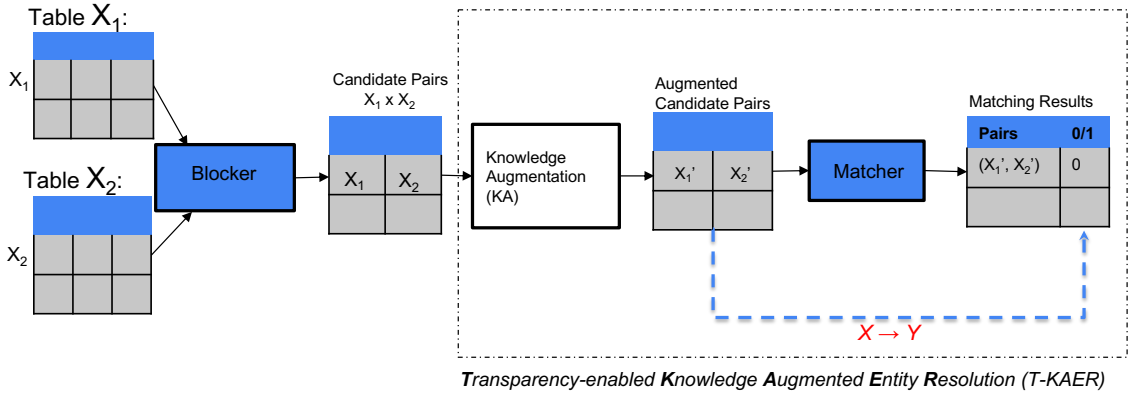

Figure 1. Table X_1 and Table X_2 are a pair of relational tables. Candidate pairs $(x_1, x_2) \in X_1 \times X_2$ are executed by the **Blocker**. Then follows the pipeline of Knowledge Augmented Entity Resolution (*KAER* framework): Augmented candidate pair (x_1', x_2') is processed by KA component. Then it is passed to the **Matcher** and returns the matching results (0: not-match,1:match) to data curators. Transparency is enhanced through providing evidence explaining how the augmented data inputs influence the decision-making process.

Contributions We introduce *T-KAER* [1] (Figure 1) to enhance the transparency for entity resolution process. In summary, this paper makes three main contributions:

- Proposing and addressing three transparency-related questions (T-Qs) to enhance transparency in the Knowledge-Augmented Entity Resolution (KAER (Fang et al., 2023)) framework .
- Designing a provenance-persevering pipeline enables modeling the training information into the structured log files, with the aim of supporting error analysis from both quantitative and qualitative perspectives.
- Conducting experiments on a citation dataset to demonstrate how T-KAER can facilitate error analysis and enhance transparency.

[1] T-KAER is public and freely available from GitHub: https://github.com/idaks/knowledge-augmented-entity-resolution

Notation Definitions and Related Work

Notation of Entity Resolution

The input of the entity resolution task consists of a set $M \subseteq X_1 \times X_2$, where X_1 and X_2 are two sets of data entry collections that contain duplicated entries. Each data entry, $x_1 \in X_1$ or $x_2 \in X_2$, is formatted in $(col_i, val_i)_{1 \leq i \leq \mathcal{N}}$, containing $\mathcal{N}$ column and values. The task discussed in this paper focuses on: for each data entry pair $(x_1, x_2) \in M$, determine whether x_1 and x_2 refer to the same data entity.

KAER: Pre-trained Language Model for Entity Resolution

A few recent works apply transformer-based PLMs to entity resolution tasks. (Paganelli et al., 2022) discover that simply fine-tuning BERT can benefit matching/not-matching classification tasks and recognize the input sequence as a pair of records. Ditto by (Li et al., 2020) is the state-of-the-art entity matching system based on PLMs, i.e., RoBERTa. In addition, Ditto provides a deeper language understanding for entity resolution by injecting domain knowledge, summarizing the key information, and augmenting with more difficult examples for training data. Following the work by (Li et al., 2020), KAER uses RoBERTa as the backbone model.

In summary, KAER uses PLMs for entity resolution and contains three modules for knowledge augmentation: a) column semantic type augmentation, b) entity semantic type augmentation, and c) three options of prompting types. The subsequent sections will describe each module within KAER.

Column-Level Knowledge Augmentation. Semantic column-type augmentation can inject domain-specific knowledge for columns with/without existing column names. Existing studies (Hulsebos et al., 2019; Suhara et al., 2022) use deep learning approaches to detect semantic data types at the column level. We adapt existing methods: *Sherlock* (Hulsebos et al., 2019) and *Doduo* (Suhara et al., 2022), to perform column semantic typing prediction. *Sherlock* is a multi-input deep neural network that uses multiple feature sets, including embeddings and column statistics, with a multi-layer sub-neural network applied to each column-wise feature set, and the output is fed into a primary neural network. Comparably, *Doduo* is a multi-task learning framework based on PLMs (Suhara et al., 2022). *Doduo* serializes the entire table into a sequence of tokens, i.e., concatenating column values to make a sequence of tokens and feed that sequence as input to the transformer (Suhara et al., 2022).

Entity-Level Knowledge Augmentation. Entity semantic type augmentation leverages the entity linking (Li et al., 2020) method to identify all entity mentions from a given knowledge base (KB) within a given text input. Entity linking (EL) refers to linking entity mentions appearing in natural language text with their corresponding entities in an external knowledge base, e.g., Wikidata. Ayoola et al. (Ayoola et al., 2022) introduced an EL method by fine-tuning a PLM (RoBERTa) over Wikidata, which is used for EL in this study, which will be examined in this work as the entity-level knowledge augmentation methods.

Prompting Types. KAER has employed two different methods of prompting for knowledge augmentation, template-based and constrained tuning. For template-based prompting, text-based templates are utilized to verbalize the domain knowledge as text

input, using connectors such as slash[2] ("/") and space. Second, we have examined prompting with constrained tuning by employing soft-position encoding and visible matrix, informed by the method introduced in KBert (Liu et al., 2020).

T-KAER Introduction

T-KAER enhances the transparency of the entity resolution process by documenting the experimental process into a log file. A list of variables applied and data products generated during the experimental process are recorded (see table **Predicted Data** from Figure 2). In the subsequent sections, we will introduce the main components for T-KAER, and using the datalog queries, a declarative programming language, to show how the system models each transparency-related question (T-Q) with the recorded information.

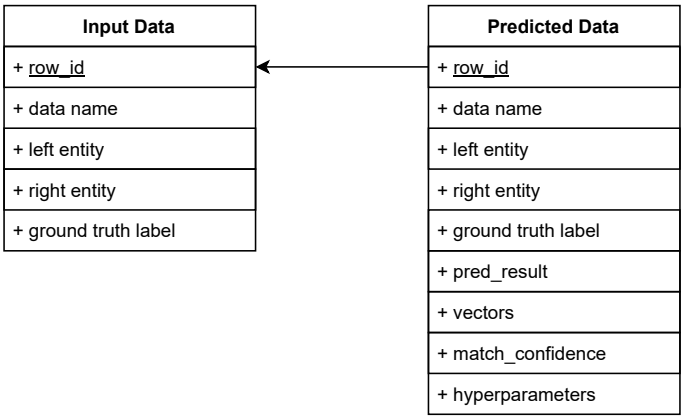

Figure 2. The UML diagram showing the structure of log file in JSON format. The **Input Data** collects information before running the experiment, and the **Predicted Data** appends run-time parameters and computation results to the **Input Data** after the experiment.

Entry Inputs

For each entity pair, (e_1, e_2), the text context of column names and values of e_1 and e_2 are serialized and concatenated as the input for PLMs. The [CLS] token position is used to classify whether e_1 and e_2 refer to the same entity. The loss for optimizing the classification objective is:

$$\ell = -log\ p(y|s(e_1, e_2)) \tag{1}$$

where y denotes whether e_1 and e_2 refer to the same entity, and $s(\cdot,\ \cdot)$ denotes the serialization and transformation of entity pairs with knowledge injection and prompting methods.

$$s(e_i, e_j) ::= \text{[CLS]}\ serialize(e_i)\ \text{[SEP]}\ serialize(e_j)\ \text{[SEP]} \tag{2}$$

where $serialize(\cdot)$ serializes each data entry.

$$\begin{aligned} serialize(e_i) ::= &\text{[COL]}\ f(col_1, pt)\ \text{[VAL]}\ g(val_1, pt)\ \ldots \\ &\text{[COL]}\ f(col_N, pt)\ \text{[VAL]}\ g(val_N, pt) \end{aligned} \tag{3}$$

[2] We use the slash to represent the "or" semantic that is commonly present in the general web corpus.

where $f(col_i, pt)$ denotes the semantic column type injection with prompting method pt, and $g(val_i, pt)$ denotes the EL injection with prompting method pt.

Contents of Log Files

Figure 2 showcases the information collected in a log file. Before the experiment, the dataset name, and for each row, the row index, entry inputs (left entity and right entity), and the ground truth are recorded in the **Input Data**. After running the experiment, the system will harvest the variables used and internal products computed during the testing process into the **Predict Data**. This includes hyperparameters, embedding vectors for each row of entry inputs by the model, predicted results (**match** or **not-match**), and matching confidence.

Three Transparency-related Questions (T-Qs) Answered by Datalog Queries

In this section, we will model and explain three transparency-related questions (T-Qs)(See Figure 3). We will use Datalog (Ceri et al., 1989) queries to represent the retrieval process. The tables **Input Data** and **Predicted Data** will be used to address the T-Qs.

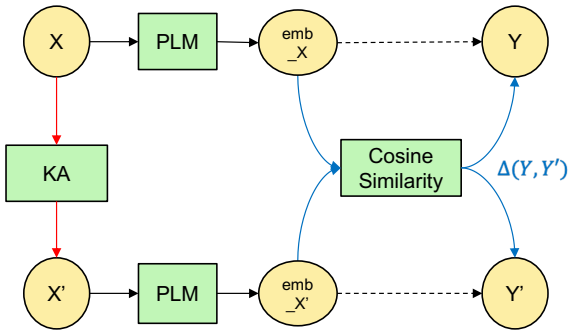

Figure 3. Yellow nodes represent the data and variables, green boxes represent the processes, such as KA for the knowledge augmentation process. Modeling three T-Qs with log files: T-Q(1) - Horizontally, the black path showcases various entry inputs X and X' resulting in predicted results Y and Y'; T-Q(2) - The vertical red path (X → X') tracks how KA methods alter the entry inputs; T-Q(3) - The blue curve path represents the cosine similarity between embeddings, determining the similarity of predicted results.

T-Q(1). What is the experimental process for matching results based on data inputs?

This question requires documenting the experimental process by retrieving entity pairs as entry inputs and predicted results by each method.

$$X\_to\_Y(A, \text{“}Sherlock\text{”}, L, R, G, Y) :- Predicted\_Data(A, \text{“}Sherlock\text{”}, L, R, G, Y, \_, \_, \_). \quad (4)$$

Denote that **Predicted_Data** (recorded in Figure 2) is needed to retrieve left (L) and right entity (R), ground truth label (G), and predicted result (Y) every row and by various methods. Here, the method name is “Sherlock".

T-Q(2). Which semantic information does KAER augment in the raw data inputs?

By exploring this question, we can compare various entry inputs augmented by different methods. As mentioned above, there are two types of knowledge: column level, and entity level (See the examples in Table 1). Various knowledge augmented into the entity pairs (left entity and right entity) will result in the difference between the semantic information contained.

To address T-Q(2), **Input Data** (recorded in Figure 2) is used:

$$delta\_X(A1, \text{“}Sherlock\text{”}, L1, R1, \text{“}Doduo\text{”}, L2, R2) :- Input\_Data(A1, \text{“}Sherlock\text{”}, L1, R1, \_), \\ Input\_Data(A1, \text{“}Doduo\text{”}, L2, R2, \_). \tag{5}$$

Note that L1 and R1 are the left entity and right entity from “original" dataset, while L2 and R2 are from “Sherlock"-augmented dataset. This equation is to help compare entity pairs augmented by different methods.

Table 1. Semantic information added: column semantic typing, and entity linking. In column *column name/ column semantic typing*: the first row is predicted by Sherlock (Hulsebos et al., 2019), in which the *“name”* is augmented with semantic type *“song_ name”*, and the second row is predicted by Doduo (Suhara et al., 2022), in which *“title”* can be augmented with *“computer.software”*. In column *cell value / entity linking*, entity mentions such as *“Illusion"* is annotated as *“single"*, and *“protocol"* is identified as *“computer network protocol"*.

column name (original)	cell value (original)	column name / column semantic typing	cell value / entity linking
name	Illusion (feat . Echosmith) Zedd True Colors Dance, Music, Electronic 2015 Interscope Records 6:30	name / song_name	Illusion / single (feat . Echosmith) Zedd True Colors Dance, Music, Electronic 2015 Interscope Records 6:30
title	the demarcation protocol: a technique for maintaining constraints in distributed database systems vldb j. 1994	title / computer.software	the demarcation protocol / computer network protocol: a technique for maintaining constraints in distributed database systems vldb j. 1994

T-Q(3). Which semantic information of the augmented inputs influences the predictions?

This question aims to explore the impact of augmenting entity pairs using various knowledge augmentation methods on the differences in predicted results. Specifically, by incorporating domain knowledge at the column level, entity level, or both levels, the predicted results may either improve or worsen.

Compared to T-Q(2), which retrieves various entry inputs in text, T-Q(3) compares the embedding vectors based on entry inputs generated by PLMs. In detail, the system collects embedding vectors that yield the same correct predicted result and embedding vectors that yield different predicted results. Then the similarity between embedding vectors under each condition is compared to determine whether the former is higher than the latter. The assumption is that embedding vectors can reflect semantic information

Situation I: Various augmentation methods result in the different predicted results:

$$delta\_Y(A1, \text{“}Sherlock\text{”}, L1, R1, Y1, V1, \text{“}Doduo\text{”}, L2, R2, Y2, V2, G) :- \\ Predicted\_Data(A1, \text{“}Sherlock\text{”}, L1, R1, G, Y1, V1, \_, \_), \\ Predicted\_Data(A1, \text{“}Doduo\text{”}, L2, R2, G, Y2, V2, \_, \_), \\ Y1 != Y2. \tag{6}$$

Situation II: Various augmentation methods result in the same and correct predicted results:

$$delta\_Y(A1, \text{“}Sherlock\text{”}, L1, R1, Y1, V1, \text{“}Doduo\text{”}, L2, R2, Y2, V2) :- \\ Predicted\_Data(A1, \text{“}Sherlock\text{”}, L, R, G, Y, \_, \_, \_), \\ Predicted\_Data(A1, \text{“}Sherlock\text{”}, L, R, G, Y, \_, \_, \_), \\ Y1 = Y2, Y1 = G. \tag{7}$$

Experiment and Results Analysis

KAER is evaluated on the Magellan datasets (Das et al., 2022) across various domains. In this section, we will introduce one of the datasets as an example for a case study, namely the *DBLP-ACM* dataset from the citation domain. Then, we will describe the experimental settings and results analysis in both quantitative and qualitative analysis.

Dataset Description

DBLP-ACM dataset comprises four attributes: title, authors, venue, and year. Each entry represents a publication record, with the title indicating the paper or article name, authors containing the author names, and the venue specifying the platform or journal where the publication is released. The data input of each entity pair is the serialized string following Eq. 3. That is, for each entity pair column names and values will be concatenated and serialized into one single string. For instance, 'Entity 1' at row id 1457 from Table 5 will be serialized as: "COL *title* VAL *a formal perspective on the view selection problem* COL *authors* VAL *rada chirkova, dan suciu, alon y. halevy* COL *venue* VAL *vldb j.* COL *year* VAL *2002*" , and 'Entity 2' will be serialized as "COL *title* VAL *a formal perspective on the view selection problem* COL authors VAL *rada chirkova, alon y. halevy, dan suciu* COL venue VAL *very large data bases* COL *year* VAL *2002*". Entities 1 and 2 will be concatenated by space and used as data input for PLMs.

Experimental Settings

As illustrated previously, there are two levels of knowledge augmentation (KA) methods: column level and entity level. Consequently, we established two groups of datasets for experiments: one with column-level KA datasets only and another with combined (combination of column-level and entity-level) KA datasets (See Table 2).

Table 2. Experiments are categorized into two groups: Column-level KA datasets, and Column- entity level KA datasets.

Types	Experiment I (Column Level)	Experiment II (Column Level & Entity Level)
Test I	Matching results by Sherlock and Doduo are different	Matching results by Doduo and Doduo with EL (Entity Linking by ReFinED) are different
Test II	Matching results by Sherlock and Doduo are the same and correct (predicted results by non-KA are wrong)	Matching results by Doduo and Doduo with EL are the same and correct (On the condition that the predicted results by non-KA are incorrect)

Results Analysis

Documenting the experimental process into log files facilitates the retrieval of information to address three T-Qs. In a nutshell, T-Q(1) compares various predicted results, T-Q(2) compares various entry inputs, and T-Q(3) examines the similarity of various embedding vectors. The results analysis will be presented based on the log files from two perspectives: quantitative and qualitative.

Quantitative Analysis

There are two experiments processed for quantitative analysis: (1) Count how many rows fulfill the requirements in Test I and Test II. (2) Compare cosine similarity between embedding vectors based on various entry inputs in Test I and Test II (See conditions for Test I and Test II in Table 2).

Evaluate Performance of KA Methods for Predicted Results: T-Q(1)

This evaluation result (see Table 3) helps address T-Q(1), the number of rows that the predicted matching results are influenced by KA methods. The left table illustrates the performance of column-level augmented methods only, while the right table compares column-level (i.p. KA by Doduo) and combined augmented methods.

In Experiment I, where column semantic typing is injected by both methods, three rows of matching results show improvement. Sherlock correctly predicts three rows that Doduo misclassifies, and Doduo correctly predicts two rows that Sherlock misclassifies.

For Experiment II, both methods enhance the performance of matching result prediction by two rows. The combined method improves five rows of predicted results misclassified by Doduo. Paradoxically, there are four rows correctly predicted by Doduo but misclassified by the combined method.

Table 3. Result analysis on T-Q(1). Left Table: compares prediction results of column-level augmented methods (Sherlock and Doduo). Right Table: compares prediction results of column-level (Doduo) and combined augmented method (Doduo & EL). T: Predicted results equal to the ground truth; F: Predicted results do not equal the ground truth. We compute the number of rows and the ratio (=row count/ total number of rows). The last row for each table represents cases where both KA methods yield true results, given a false prediction by non-KA.

Predicted Results		Row Count	Ratio
Sherlock	Doduo		
T	F	3	0.0012
F	T	2	0.0008
T	T	3	0.0012

Predicted Results		Row Count	Ratio
Doduo	Doduo & EL		
T	F	4	0.0016
F	T	5	0.0020
T	T	2	0.0008

Evaluate Semantic Information Contained in Embedding Vectors: T-Q(3)

The evaluation results (see Table 4) address T-Q(3), illustrating how internal products, embedding vectors generated by PLMs to reflect the semantic information in entry inputs, lead to either identical or different predicted results. We calculate the cosine similarity for embeddings in Test I and Test II, comparing the average similarity for each test.

The left table 4 showcases the column-level methods, while the right table compares column-level and combined methods. In both experiments, the average cosine similarity in Test II is higher than in Test I. Embedding vectors in Test II are derived from entry inputs through various augmentation methods, resulting in the same prediction result. On the other hand, embedding vectors in Test I are generated from entry inputs leading to different prediction results. The higher cosine similarity between embeddings represents they contain more similar semantic information. This aligns with our assumption that "Data inputs resulting in the same predicted results (Test II) exhibit more similar embeddings (with similar semantic information) compared to those resulting in different predicted results (Test I)". This holds true on average.

Table 4. Result analysis on T-Q(3). Left Table: compares prediction results of column-level augmented methods (Sherlock and Doduo). Right Table: compares predictions of column-level (Doduo) combined augmented methods (Doduo & EL). T: Predicted results equal to the ground truth; F: Predicted results do not equal to the ground truth. We calculate the average of cosine similarity (Avg.) for embeddings for Test I and Test II.

Test ID	Predicted Results		Ground Truth	Avg
	Sherlock	Doduo		
Test I	T	F	1	0.4980
	F	T	1	
	T	F	0	
	F	T	0	
Test II	T	T	1	0.6475
	T	T	0	

Test ID	Predicted Results		Ground Truth	Avg
	Doduo	Doduo & EL		
Test I	T	F	1	0.2749
	F	T	1	
	T	F	0	
	F	T	0	
Test II	T	T	1	0.5796
	T	T	0	

Qualitative Analysis

To gain a deeper understanding of how KA methods influence entity resolution results, we select certain entity pairs from the quantitative analysis and proceed with qualitative analysis. The chosen entity pairs from the original dataset are listed in Table 5. Next, we manually complete the entity resolution tasks for the selected rows, highlighting potential challenges. Subsequently, we present the predicted results on those rows using various KA methods and provide explanatory reasons through error analysis based on T-KAER. Therefore, the log files enable us to explain T-Q(1) and T-Q(2) explicitly in these case studies.

Table 5. Selected rows from the original **DBLP-ACM** dataset for error analysis. 'Entity 1' is from source DBLP. 'Entity 2' is from source ACM, highlighted in grey.

entry	row id	title	authors	venue	year
Entity 1	2437	the mariposa distributed database management system	jeff sidell	sigmod record	1996
Entity 2	2437	mariposa : a wide-area distributed database system	michael stonebraker , paul m. aoki , witold litwin , avi pfeffer , adam sah , jeff sidell , carl staelin , andrew yu	the vldb journal – the international journal on very large data bases	1996
Entity 1	2407	a formal perspective on the view selection problem	rada chirkova , dan suciu , alon y. halevy	vldb j.	2002
Entity 2	2407	a formal perspective on the view selection problem	rada chirkova , alon y. halevy , dan suciu	very large data bases	2001
Entity 1	1457	reminiscences on influential papers	hector garcia-molina , patricia g. selinger , tomasz imielinski , david maier , jeffrey d. ullman , richard t. snodgrass	sigmod record	1998
Entity 2	1457	reminiscences on influential papers	richard snodgrass	acm sigmod record	1998

Entity Resolution Finished Manually

Entity pairs at row 2437 (the first two rows in Table 5) are evidently not a match, as the values for title and venue differ significantly. Additionally, the authors are not identical. In the case of entity pairs at row 2407, determining a match might be challenging without domain knowledge in citation data, as the order of author names and the publication year are crucial. Therefore, despite these two entities sharing the same title, the same set of author names, and a similar meaning of venue name (i.e., 'vldb j.' is the acronym for 'very large data bases'), entity pairs at row 2407 are marked as **not-match**. For entity pairs at row 1457, we observe that the values for key columns "title," "venue," and "year" are the same, but the author names are not exactly identical. It turns out that this record contains scientific commentaries on various works by different authors. Therefore, the single editor name: *Richard Snodgrass* is sufficient to represent other authors. Consequently, the last pair of entities refers to the same citation.

Error Analysis Supported by T-KAER

Case Study I: Entity Resolution Results by Column-Level KA Methods are True.

Entity pairs at row 2437 are correctly predicted as **not-match** by both Sherlock and

Doduo. The entry inputs augmented by these two methods are as follows (see Table 6). Note that the column semantic types (CST) predicted by the same method for Entity 1 and Entity 2 are even slightly different, which is mainly due to the separate training of two datasets (i.e., DBLP and ACM datasets). Overall, the column semantic types predicted by Doduo are more precise than those predicted by Sherlock. For instance, values in the column “authors" are recognized as “people.person," and the year is annotated as “time.event." Despite the poor CST predicted by Sherlock, the entity resolution results from both methods are correct. One reason is that most of the strings in the sequence are different, so it does not significantly impact the predicted results, regardless of whether the augmented knowledge is correct or not.

Table 6. Augmented Entity pairs at row 2437: The CST augmented by these two methods is highlighted in **bold**.

Entity Pair	entry_sherlock	entry_doduo
Entity 1	COL title **symbol** VAL the mariposa distributed database management system COL authors **area** VAL jeff sidell COL venue **person** VAL sigmod record COL year **education** VAL 1996	COL title **business.industry** VAL the mariposa distributed database management system COL authors **people.person** VAL jeff sidell COL venue **organization.organization** VAL sigmod record COL year **time.event** VAL 1996
Entity 2	COL title **result** VAL mariposa : a wide-area distributed database system COL authors **category** VAL michael stonebraker , paul m. aoki , witold litwin , avi pfeffer , adam sah , jeff sidell , carl staelin , andrew yu COL venue **code** VAL the vldb journal – the international journal on very large data bases COL year **education** VAL 1996	COL title **business.industry** VAL the mariposa distributed database management system COL authors **people.person** VAL jeff sidell COL venue **organization.organization** VAL sigmod record COL year **time.event** VAL 1996

Case Study II: Entity Resolution Results Predicted by Column-Level KA Methods are Different.

These two records are incorrectly predicted as **match** by Sherlock, but correctly predicted as **not-match** by Doduo. While comparing the entry inputs, there are sig-nificant differences between the column semantic types augmented by Sherlock and Doduo. Column “authors" is predicted as “area" by Sherlock, correctly recognized as “people.person" by Doduo. For column “venue", Sherlock predicts the column val-ues as “person", on the contrary, Doduo labels it as “organization.organization" and “book.periodical". Finally, the column “year" is annotated as “education" by Sherlock, as “time.event" by Doduo instead. Consequently, knowledge augmented by Doduo is more precise than Sherlock.

Case Study III: Entity Resolution Results Predicted by Column-Level KA and Combined KA Methods are Different

This case study compares the predicted results based on entry inputs augmented by column-level knowledge, i.p., by Doduo, and entry inputs augmented by combined knowledge, by Doduo and entity linking methods. These two records are incorrectly predicted as **not-match** by Doduo, but correctly predicted as **match** by combined knowledge. Initially, before knowledge injection, the two records seemed different due

Table 7. Augmented Entity pairs at row 2407: The CST augmented by these two methods is highlighted in **bold**.

Entity Pair	entry_sherlock	entry_doduo
Entity 1	COL title **symbol** VAL a formal perspective on the view selection problem COL authors **area** VAL rada chirkova , dan suciu , alon y. halevy COL venue **person** VAL vldb j. COL year **education** VAL 2002	COL title **business.industry** VAL a formal perspective on the view selection problem COL authors **people.person** VAL rada chirkova , dan suciu , alon y. halevy COL venue **organization.organization** VAL vldb j. COL year **time.event** VAL 2002
Entity 2	COL title **result** VAL a formal perspective on the view selection problem COL authors **category** VAL rada chirkova , alon y. halevy , dan suciu COL venue **code** VAL very large data bases COL year **education** VAL 2001	COL title **business.industry** VAL a formal perspective on the view selection problem COL authors **people.person** VAL rada chirkova , alon y. halevy , dan suciu COL venue **book.periodical** VAL very large data bases COL year **time.event** VAL 2001

to distinct values in the author fields, and Doduo alone couldn't reflect the similarity. By adding the knowledge from entity linking, the dataset was enriched with additional labels such as "scientist" for "hector garcia-molina" and "professional" for "richard snodgrass". This additional layer of knowledge was able to improve the performance of the entity resolution model.

Table 8. Augmented Entity pairs at row 1457: The CST augmented by Doduo is highlighted in **bold**, and the knowledge augmented by entity linking (EL) is in red.

Entity Pair	entry_doduo	entry_doduo_EL
Entity 1	COL title **business.industry** VAL reminiscences on influential papers COL authors **people.person** VAL hector garcia-molina , patricia g. selinger , tomasz imielinski , david maier , jeffrey d. ullman , richard t. snodgrass COL venue **organization.organization** VAL sigmod record COL year **time.event** VAL 1998	COL title **business.industry** VAL reminiscences on influential papers COL authors **people.person** VAL hector garcia-molina (scientist) , patricia g. selinger , tomasz imielinski , david maier , jeffrey d. ullman , richard t. snodgrass COL venue **organization.organization** VAL sigmod (album) record COL year **time.event** VAL 1998 (periodic process)
Entity 2	COL title **business.industry** VAL reminiscences on influential papers COL authors **people.person** VAL richard snodgrass COL venue **book.periodical** VAL acm sigmod record COL year **time.event** VAL 1998	COL title **business.industry** VAL reminiscences on influential papers COL authors **people.person** VAL richard snodgrass (professional) COL venue **book.periodical** VAL acm sigmod (artificial physical object) record COL year **time.event** VAL 1998 (periodic process)

Conclusions

We present T-KAER, a framework that enhances transparent entity resolution tasks by documenting the experimental process in log files. We conduct a case study on a citation dataset, both quantitative and qualitative analyses are processed on the log files. T-KAER allows us to address three transparency-related questions, elucidating: (1) How diverse entry inputs lead to variations in the performance of predicted results; (2) What distinct semantic information is contained in entry inputs; (3) How the internal products, specifically embeddings generated by pre-trained language models (PLMs), can accurately represent the semantic information derived from entry inputs.

References

Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., and Pierleoni, A. (2022). Refined: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 NAACL: Human Language Technologies: Industry Track, NAACL 2022*, pages 209–220.

Buneman, P., Khanna, S., and Wang-Chiew, T. (2001). Why and where: A characterization of data provenance. In *International conference on database theory*, pages 316–330. Springer.

Ceri, S., Gottlob, G., Tanca, L., et al. (1989). What you always wanted to know about datalog(and never dared to ask). *IEEE transactions on knowledge and data engineering*, 1(1):146–166.

Christen, P. (2012). The Data Matching Process. In *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, pages 23–35. Springer, Berlin, Heidelberg.

Das, S., Doan, A., G. C., P. S., Gokhale, C., Konda, P., Govind, Y., and Paulsen, D. (2022). The magellan data repository. https://sites.google.com/site/anhaidgroup/projects/data.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Fang, L., Li, L., Liu, Y., Torvik, V. I., and Ludäscher, B. (2023). Kaer: A knowledge augmented pre-trained language model for entity resolution. *Knowledge Augmented Methods for Natural Language Processing workshop in conjunction with AAAI 2023*.

Grafberger, S., Groth, P., and Schelter, S. (2023). Provenance tracking for end-to-end machine learning pipelines. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1512–1512.

Hulsebos, M., Hu, K. Z., Bakker, M. A., Zgraggen, E., Satyanarayan, A., Kraska, T., Demiralp, Ç., and Hidalgo, C. A. (2019). Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on KDD 2019*, pages 1500–1508. ACM.

Koumarelas, I., Jiang, L., and Naumann, F. (2020). Data Preparation for Duplicate Detection. *Journal of Data and Information Quality*, 12(3):15:1–15:24.

Li, B., Miao, Y., Wang, Y., Sun, Y., and Wang, W. (2021a). Improving the Efficiency and Effectiveness for BERT-based Entity Resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13226–13233. Number: 15.

Li, X., Talburt, J. R., Li, T., and Liu, X. (2021b). When entity resolution meets deep learning, is similarity measure necessary? In *Advances in Artificial Intelligence and Applied Cognitive Computing*, pages 127–140. Springer.

Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W.-C. (2020). Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1):50–60.

Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., and Tan, W.-C. (2021c). Deep entity matching: Challenges and opportunities. *Journal of Data and Information Quality (JDIQ)*, 13(1):1–17.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020). K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI*, volume 34, pages 2901–2908.

Lyon, L. (2016). Transparency: The emerging third dimension of open science and open data. *LIBER Quarterly: The Journal of the Association of European Research Libraries*, 25(4):153–171.

McPhillips, T., Willis, C., Gryk, M. R., Nuñez-Corrales, S., and Ludäscher, B. (2019). Reproducibility by Other Means: Transparent Research Objects. In *15th International Conference on eScience*, pages 502–509.

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., et al. (2015). Promoting an open research culture. *Science*, 348(6242):1422–1425.

OR (2021). Openrefine: A free, open source, power tool for working with messy data. github.com/OpenRefine.

Paganelli, M., Buono, F. D., Baraldi, A., and Guerra, F. (2022). Analyzing how BERT performs entity matching. *Proceedings of the VLDB Endowment*, 15(8):1726–1738.

Peeters, R. and Bizer, C. (2023). Using chatgpt for entity matching. *arXiv preprint arXiv:2305.03423*.

Suhara, Y., Li, J., Li, Y., Zhang, D., Demiralp, c., Chen, C., and Tan, W.-C. (2022). Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 1493–1503. Association for Computing Machinery.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3:160018.