Splitting Tuples of Mismatched Entities

Wenfei Fan[1 2 3] Ziyan Han[1] Weilong Ren*[2] Ding Wang[4] Yaoshu Wang[2] Min Xie[2] Mengyi Yan[1]
[1]Beihang University [2]Shenzhen Institute of Computing Sciences [3]University of Edinburgh [4]Tsinghua University

A. Introduction

1. A real mismatch in IMDB: As shown in the figure below, the tuple t_s denotes a record of two mismerged European directors named "Noemi Schneider" in IMDB; in contrast, the tuple t_c is a director record with erroneous attribute values.

information from the Swiss director | information from the German director | erroneous values

tid	name	nationality	born	college	film	filmFestival	festCity	festCountry
t_s	Noemi Schneider	Swiss	2013	null	Sturm	DOK.fest	Munich	null
t_c	Noemi Schneider	Japanese	1986	ZHdK	Sturm	Landshut Short Film Festival	Landshut	Germany

Questions:
- ➢ How can we decide whether a tuple with conflict values should be split or corrected?
- ➢ To split a tuple, how should we distribute its values to the right entities?
- ➢ How can we fill in missing values for the tuples resulted from splitting?

2. Other real mismatches
- ❑ Wikidata (e.g., Joseph de Cambis (Q3185827))
- ❑ DBLP (e.g., authors with the same names can be mismerged)

B. Problem Definition

- Input: A schema R, a relation D of R, and a knowledge graph G.
- Output: The split $TS(t)$ for all tuples t in D, by referencing G.

C. Contribution

1. A scheme
- ❑ Scheme SET (Splitting EnTities). It takes G as input, and (1) splits mismatched entities and (2) corrects tuples with errors.

2. Extending REEs
- ❑ REE+. It extends Entity Enhancing Rules (REE) to REE+; by employing REE+, SET splits mismatched tuples and corrects errors in a uniform process of logic deduction, ML correlation and data extraction.

3. Detecting mismatched entities
- ❑ An ML model M_c. We train an ML model M_c that assesses the correlation of attribute values.
- ❑ Mismatch detection. By embedding an ML model M_c in REE+, SET decides whether a tuple with conflicts to split or to correct.
- ❑ Initial split. For a tuple to split, it decomposes it into multiple tuples, each denotes a distinct entity, by referencing knowledge graph G.

4. Splitting tuples
- ❑ Tuple (further) split. For a tuple to split, it distributes the un-assigned attribute values to right entities.
- ❑ Tuple correction. For tuples with errors, it resolves the conflicts by enforcing REE+s and accumulating/referencing a set Γ of validated facts (ground truth).
- ❑ Church-Rosser property. We show that under certain conditions on the ML models M in REE+, the chase is Church-Rosser.

5. Deducing missing values
- ❑ An ML model M_d. It trains an ML model M_d for imputing missing attribute values.
- ❑ Three imputation strategies. SET fill in the missing values of the split tuples by supporting three strategies: logic deduction, data extraction from knowledge graphs, and ML prediction.

6. Experimental study on real-life data
- ❑ Accuracy. On average,
  - its F_1-score is 0.92 by combining logic deduction, ML correlation models and data extraction from knowledge graphs.
  - It is more accurate than all the baselines, by 31.8%, 8.3% and 39.5% for deciding what tuples to split/correct, assigning attribute values to the split tuples, and imputing missing value, respectively.
  - It outperforms rule-based methods and ML-based methods by 35.5% and 30.3% respectively.
- ❑ Efficiency. It takes $1{,}481$s on a dataset of $1{,}057{,}217$ tuples, with a single machine.

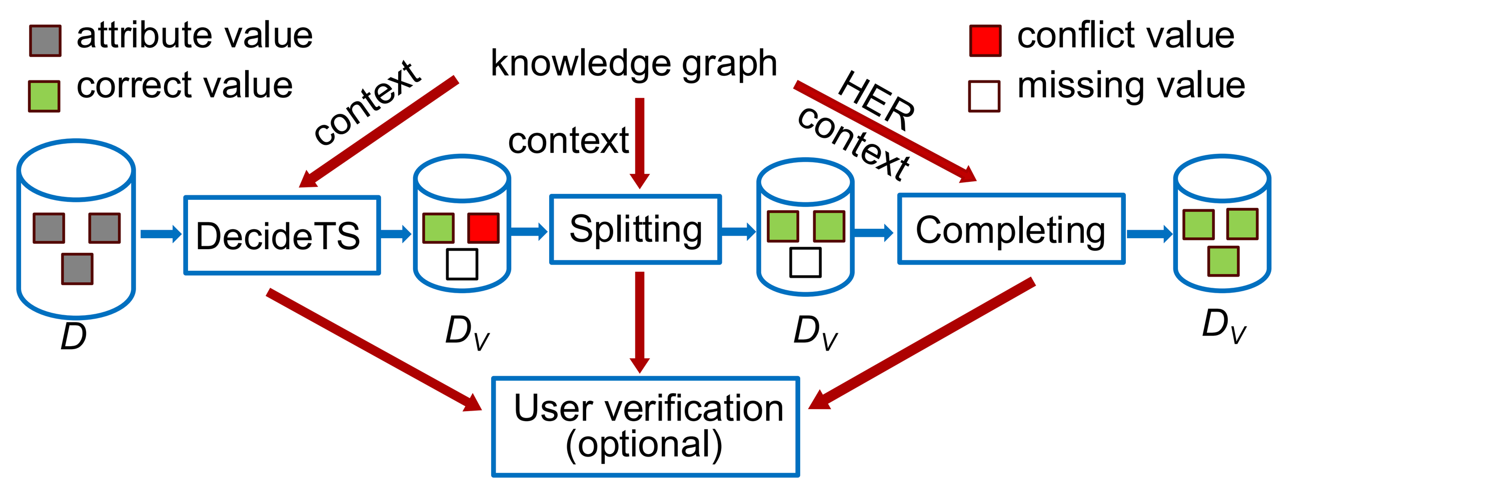
D. REE+, scheme and models

1. REE+
We extend REEs by supporting the following predicates defined over a database schema R and a knowledge graph G.

$$p := \text{vertex}(x, G) \mid \text{HER}(t, x) \mid \text{match}(t.A, x.\rho) \mid t[A] = \text{val}(x.\rho) \mid M_c(t[\bar{A}], t[B]) \geq \delta \mid M_c(t[\bar{A}], t[B] = c) \geq \delta \mid t[B] = M_d(t[\bar{A}], B)$$

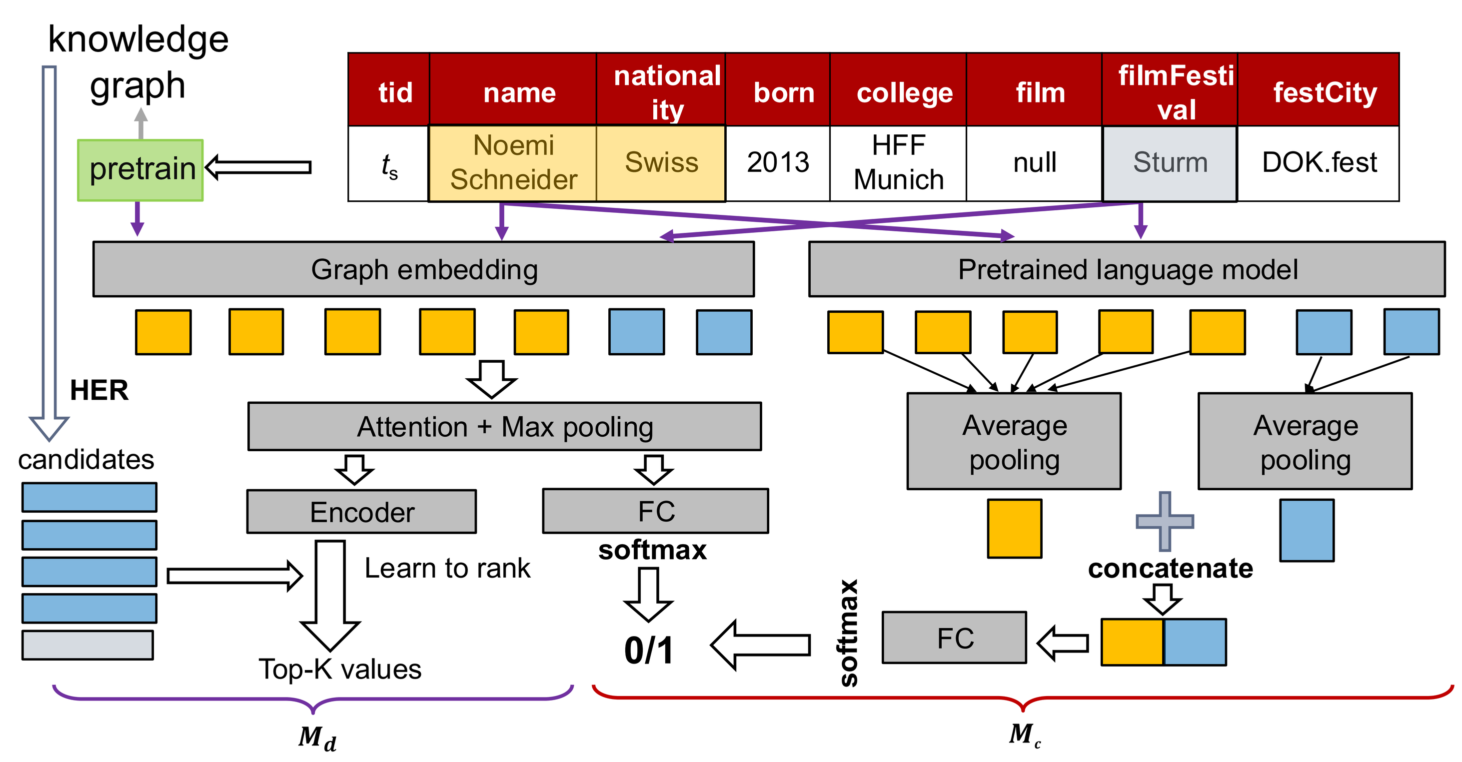
- x in vertex(x, G) is a variable denoting a vertex in knowledge graph G, referred to as a variable bounded by vertex(x, G).
- If x is bounded by vertex(x, G) and t is bounded by R(t), HER(t, x) is a Boolean function that returns true if tuple t and vertex x refer to the same entity.
- If ρ is a label path and if x and t are bounded as above, match$(t.A, x.\rho)$ checks whether the path ρ from vertex x encodes the A-attribute of tuple t.
- If t and x are bounded as above and match$(t.A, x.\rho)$ returns true, $t[A] = \text{val}(x.\rho)$ indicates that the A-attribute of t takes the value (label) of the last vertex v on the match of ρ from vertex x.
- M_c is an ML model that checks the strength of the correlation between (partial) tuple $t[\bar{A}]$ and the B-attribute value $t[B]$, and δ is a strength threshold.
- M_d is an ML model that given a partial tuple $t[\bar{A}]$, predicts a value for its B-attribute.

2. Scheme
The workflow of SET is shown as follows.



- ❑ DecideTS. For each t in D, SET detects conflicts in a single tuple (e.g., a film and filmFestival) and across tuples (e.g., different countries for the same city), with M_c. For each detected t, SET creates a set $TS(t)$ of split tuples $\{t_1, \ldots, t_k\}$ based on conflicting attributes, such that each t in $TS(t)$ denotes a distinct entity. When $|TS(t)| = 1$, t is erroneous and is corrected without splitting.
- ❑ Splitting. For each t in $TS(t)$ to split or correct, SET resolves conflicts and distributes attribute values of t to the right entities with M_c by chasing $TS(t)$ with REE+.
- ❑ Completing. SET then fills in missing values of tuples in $TS(t)$ with M_d by applying REE+s.
- ❑ User verification (optional). SET presents tuples in $TS(t)$ to users for confirmation.

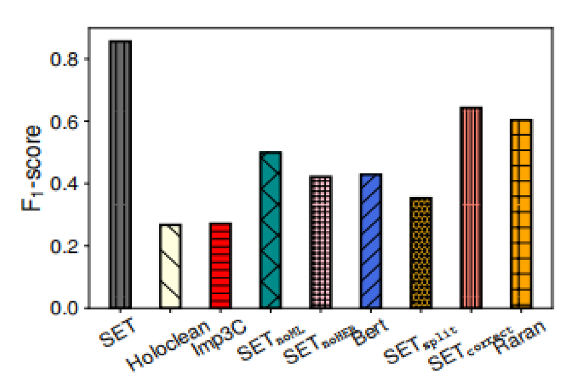
3. Network structure of M_c and M_d



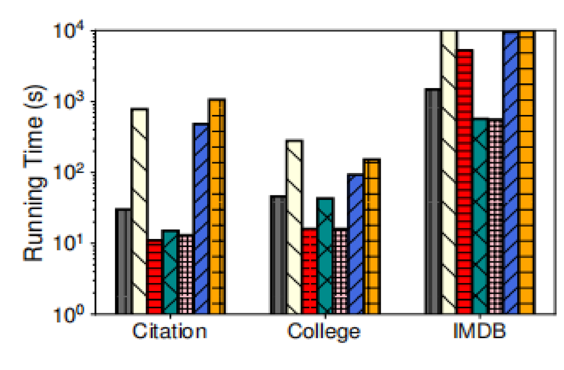
- ❑ Graph pretraining. We pretrain graph embeddings on a knowledge graph G, so that we can implicitly learn rich contextual information (e.g., DOK.fest held in Munich) from pretrained embedding.
- ❑ Context-aware embedding. We model $I_t = (t[\bar{A}], t[B])$ as a sequence by concatenating attribute values) and design encoders to obtain two representations of I_t via graph embeddings and language models, respectively. After a softmax layer, we combine the classifications and generate a confidence score by incorporating semantics.

E. Experiments

- ❑ Real-life Datasets. Citation, College, Person and IMDB.
- ❑ Baselines. Bert, Raha+Baran, Holoclean and Imp3C.
- ❑ Measurements. F1-score and execution time



Citation (overall F1)



Datasets (Time)