Using Weak Supervision to Identify Long-Tail Entities

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Yaser Oulabi and Christian Bizer - University of Mannheim, Germany

Motivation: Completing Knowledge Bases

- Cross-Domain Knowledge Bases like DBpedia, Wikidata, or the Google Knowledge Graph are used as background knowledge for tasks such as:
  – Web search
  – Natural language processing
  – Data integration and mining
  – Question answering
- Knowledge bases are more useful **the more complete they are.**
- Cross-domain knowledge bases, e.g. DBpedia, are often derived from Wikipedia and thus do not contain long-tail entities not covered by Wikipedia

Motivation:
Potential Usefulness of Web Tables for Knowledge Base Augmentation

Web Table: a relational HTML table extracted form the Web.

Web tables have been show to have high potential in constructing or completing knowledge bases [Cafarella et al. 2008], [Ritze et al. 2016]

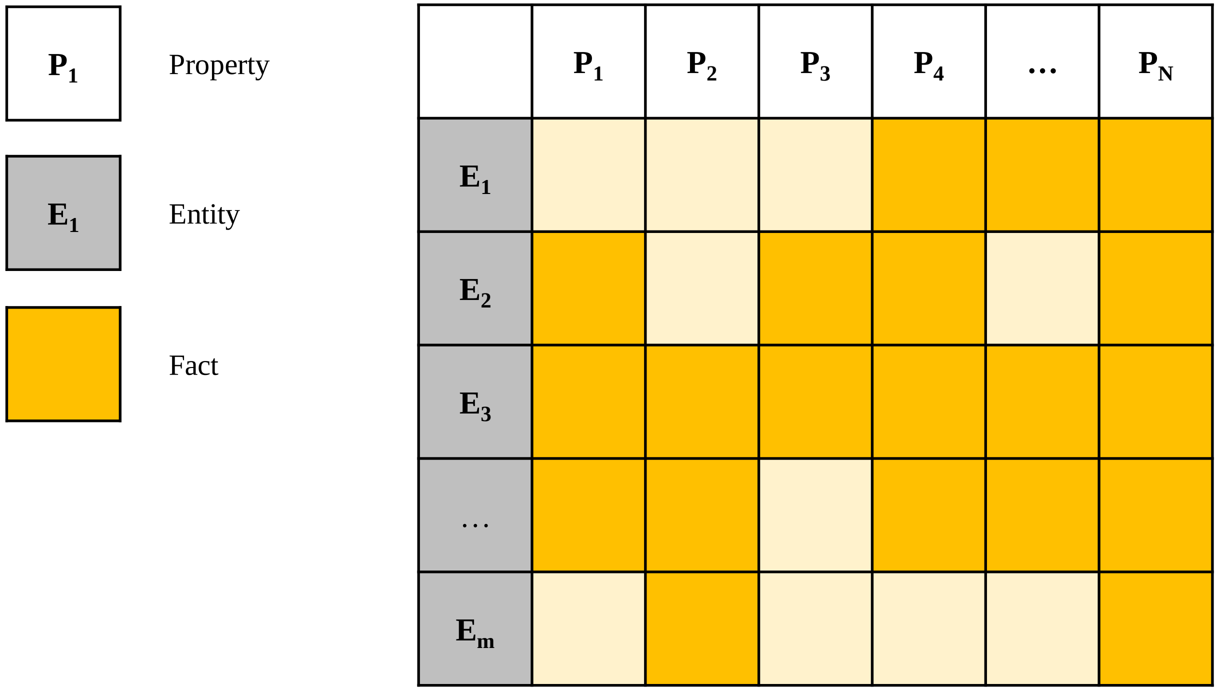
Web Data Commons Web Table Corpus [http://webdatacommons.org/webtables/]

- It consists 91.8 million english-language relational web tables of **varying quality**
- With heterogeneous schemas
- Data about a single entity is found in many web tables
- Entities appear in different combinations in many web tables

	Min	Max	Average	Median
columns	2	713	3.48	3
rows	1	35 640	10.37	2

Columns and Rows Distribution of WDC Web Table Corpus

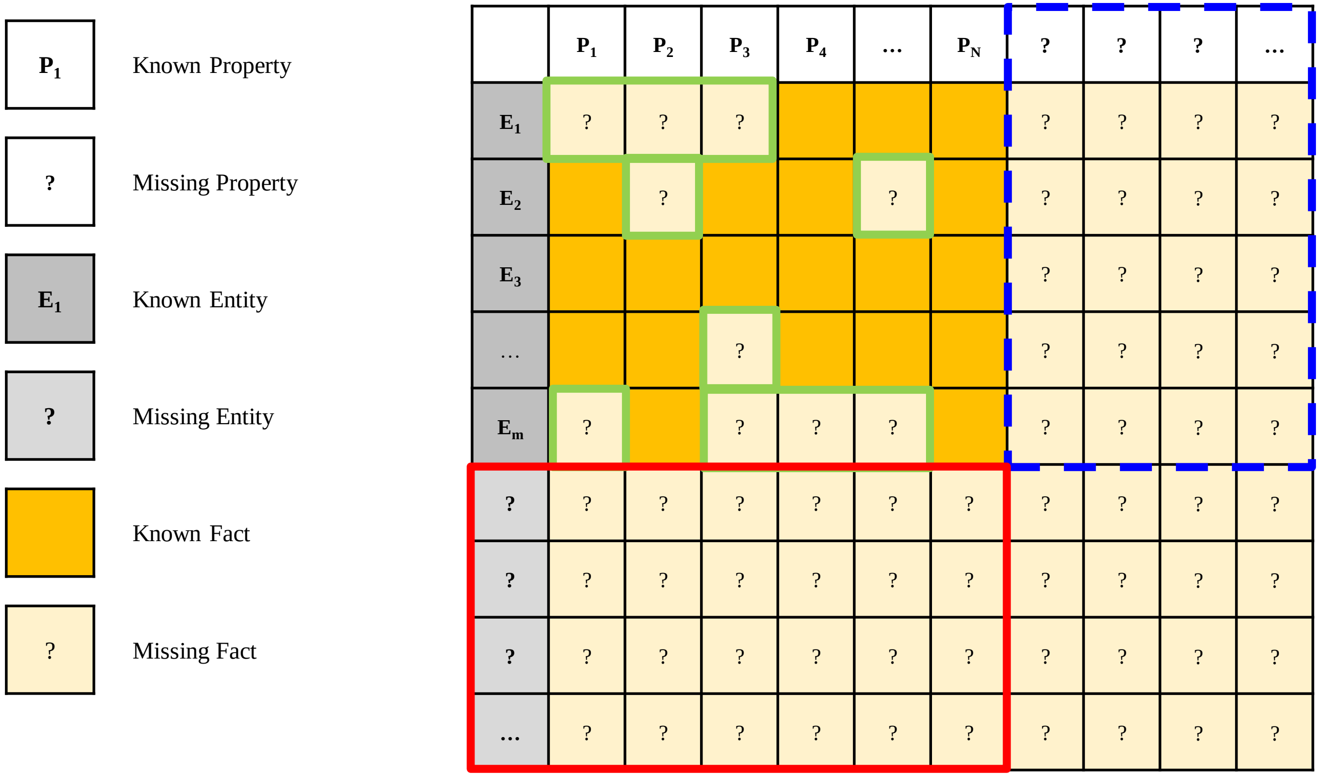
A Class in the Knowledge Base



Viewing a class in the knowledge base as a table

Where is potential for augmentation?

Knowledge Base Augmentation



Slide Filling:

add facts for existing entities and existing properties

Schema Expansion:

add new properties

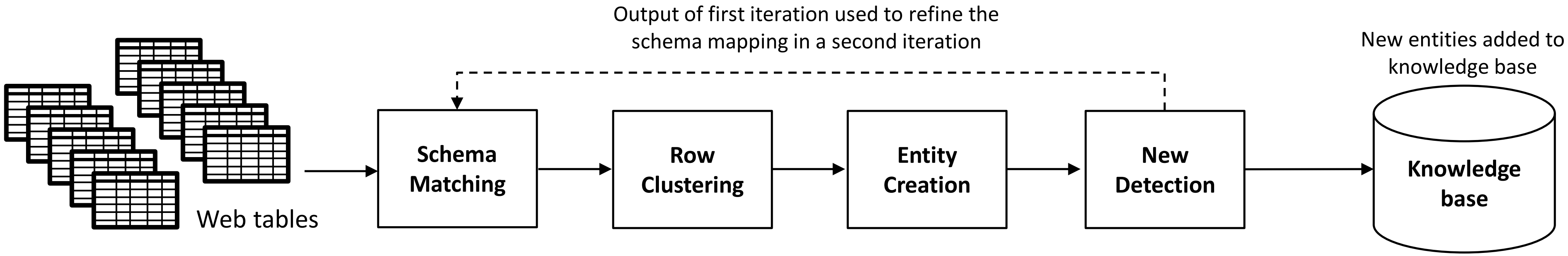
Entity Expansion:

add new entities and their descriptions (facts for existing properties)

our focus

Long-Tail Entity Expansion Pipeline

Oulabi, Y. and Bizer, C. (2019). Extending cross-domain knowledge bases with long tail entities using web table data. *Extending Database Technology 2019*.



Our approach:

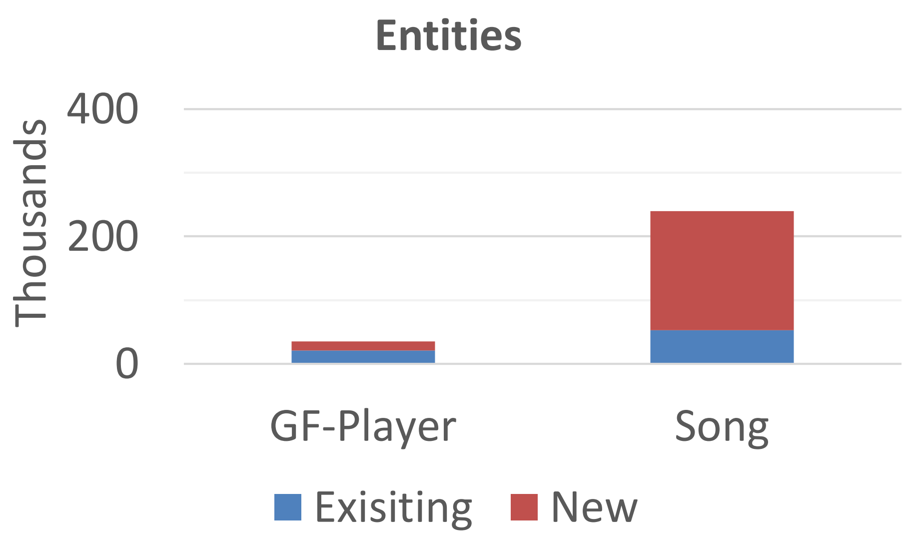
1. **Cluster rows** that describe the same instance together
   Compare two rows with each other
2. **Create entities from row clusters**
3. **Determine which entities describe new instance**
   Compare a created entity with a KB instance

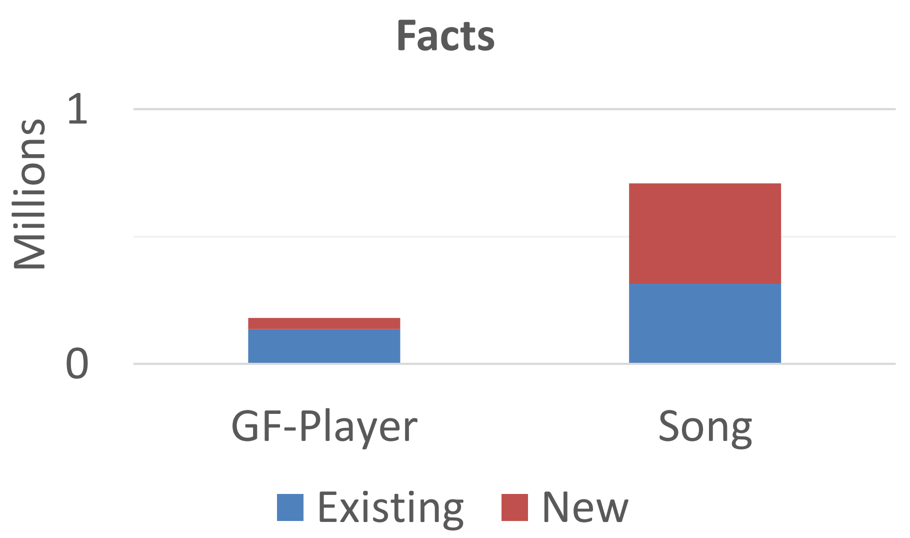
Long-Tail Entity Expansion Pipeline: RESULTS

Oulabi, Y. and Bizer, C. (2019). Extending cross-domain knowledge bases with long tail entities using web table data. *Extending Database Technology 2019*.

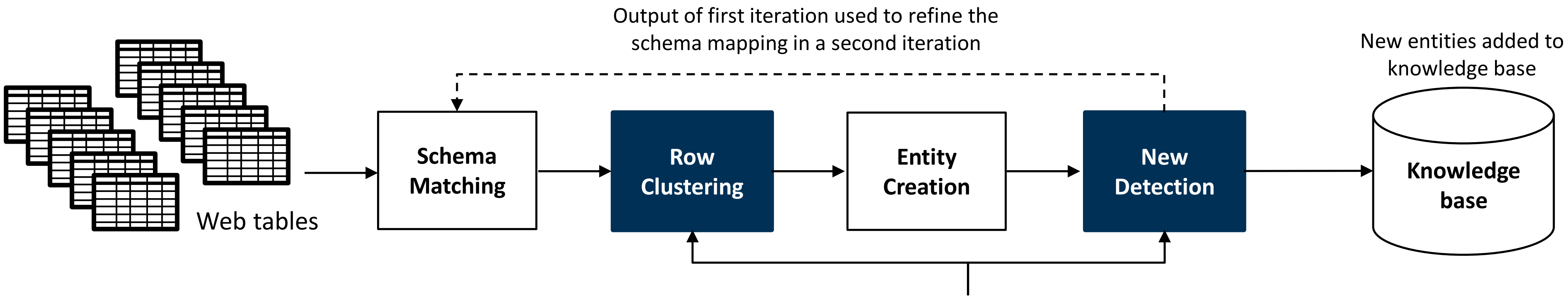
UNIVERSITY OF MANNHEIM
Data and Web Science Group

Class	Total rows	Existing entities in KB	New entities from WT	New facts from WT	N. entities accuracy	N. facts accuracy
GF-Player	648,741	30,074	13,983 (+67%)	43,800 (+32%)	0.60	0.95
Song	2,173,536	40,455	186,943 (+356%)	393,711 (+125%)	0.70	0.85





Some Components Require Supervision



These components make use of class-specifically trained entity matching methods (random forest classifier)

Label type	GF-Player	Song	Settlement	Sum
Row pair	1,298	231	2,768	4,297
Entity-instance-pair	80	34	51	165
New entity classification	17	63	23	103
Sum	**1,395**	**328**	**2,842**	**4,565**

Number of labels in **T4LTE** (http://webdatacommons.org/T4LTE/)

- To train the models we **need** positive and negative entity matching pairs
- We train the models using the **T4LTE gold standard** (Web Tables For Long-Tail Entity Extraction), which we **manually annotated** for evaluation and training in the task of long-tail entity extraction

Problem: Manually Annotating Class-Specific Training Data Is Not Viable

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Knowledge bases cover many classes
- Creating thousands of manually labeled entity matches for each class limits the applicability of automatic knowledge base expansion from web data
- We need an alternative to manually labeled entity matches

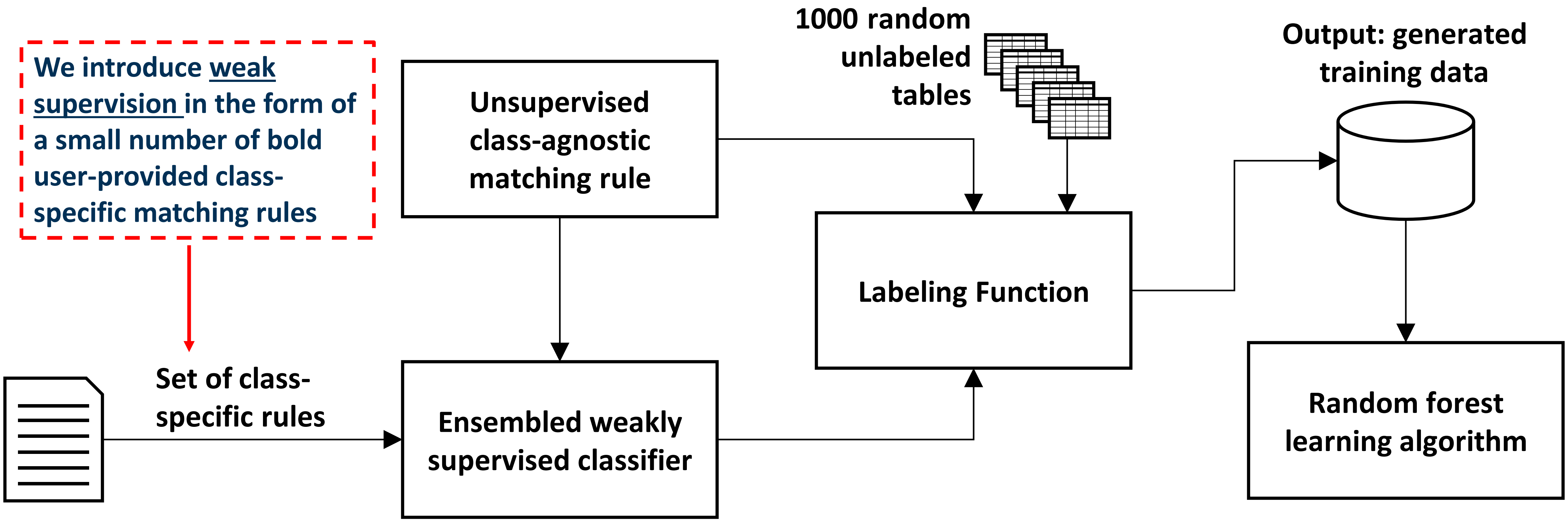
Weak Supervision & Data Programming

Weak supervision: reduce labeling effort by using supervision that is more abstract or noisier compared to traditional manually labeled high-quality training examples (strong supervision). [Ratner2017]

Data programming: paradigm, where experts are tasked with codifying any form of weak supervision into labeling functions. These functions are then employed within a broader system to generate training data by assigning labels and confidence scores to unlabeled data. [Ratner 2016]

Methodology

Overall Methodology



Summary: Similarity Features

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Class-agnostic features

- LABEL
- BOW
- PHI[1]
- SAME_TABLE[1]
- TYPE[2]
- POPULARITY[2]

Class-specific features

- ATTRIBUTE, e. g.

  ATTRIBUTE::draftPick
  ATTRIBUTE::musicalArtist
  ATTRIBUTE::postalCode

- IMPLICIT_ATT, e. g.

  IMPLICIT_ATT::team
  IMPLICIT_ATT::album
  IMPLICIT_ATT::country

[1] Row clustering only
[2] New detection only

Unsupervised Class-Agnostic Matching Rule

- Aggregate similarity features using a **weighted average**
- Weights are **equal for all classes** (assigned based on our judgement)
- Class-specific features (ATTRIBUTE and IMPLICIT_ATT) are transformed into class-agnostic by averaging
- We classify pairs as matching or non-matching **using a threshold (0.5)**
- **Classification confidence** is equal relative distance to the threshold

User-Provided Class-Specific Matching Rules

Rules are easy to create:

We restrict the rule format to conjuncts of equality tests, expressed using the schema of the knowledge base.

Rules are bold:

Provided rules must be accurate, regardless of their coverage

Small rule sets are sufficient:

We create per class only four rules

Rules for the Class: GridironFootballPlayer

$$(\text{draftYear} = \text{Equal}) \wedge (\text{draftPick} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{LABEL} = \text{Equal}) \wedge (\text{birthDate} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{draftYear} = \text{Unequal}) \rightarrow \text{Non-Match}$$

$$(\text{draftPick} = \text{Unequal}) \rightarrow \text{Non-Match}$$

Rules for the Class: Song

$$(\text{LABEL} = \text{Equal}) \wedge (\text{artist} = \text{Equal}) \wedge (\text{releaseDate} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{LABEL} = \text{Equal}) \wedge (\text{artist} = \text{Equal}) \wedge (\text{album} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{artist} = \text{Unequal}) \rightarrow \text{Non-Match}$$

$$(\text{releaseYear} = \text{Unequal}) \rightarrow \text{Non-Match}$$

Rules for the Class: Settlement

$$(\text{country} = \text{Equal}) \land (\text{postalCode} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{LABEL} = \text{Equal}) \land (\text{isPartOf} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{LABEL} = \text{Equal}) \land (\text{postalCode} = \text{Equal}) \rightarrow \text{Match}$$

$$(\text{country} = \text{Unequal}) \rightarrow \text{Non-Match}$$

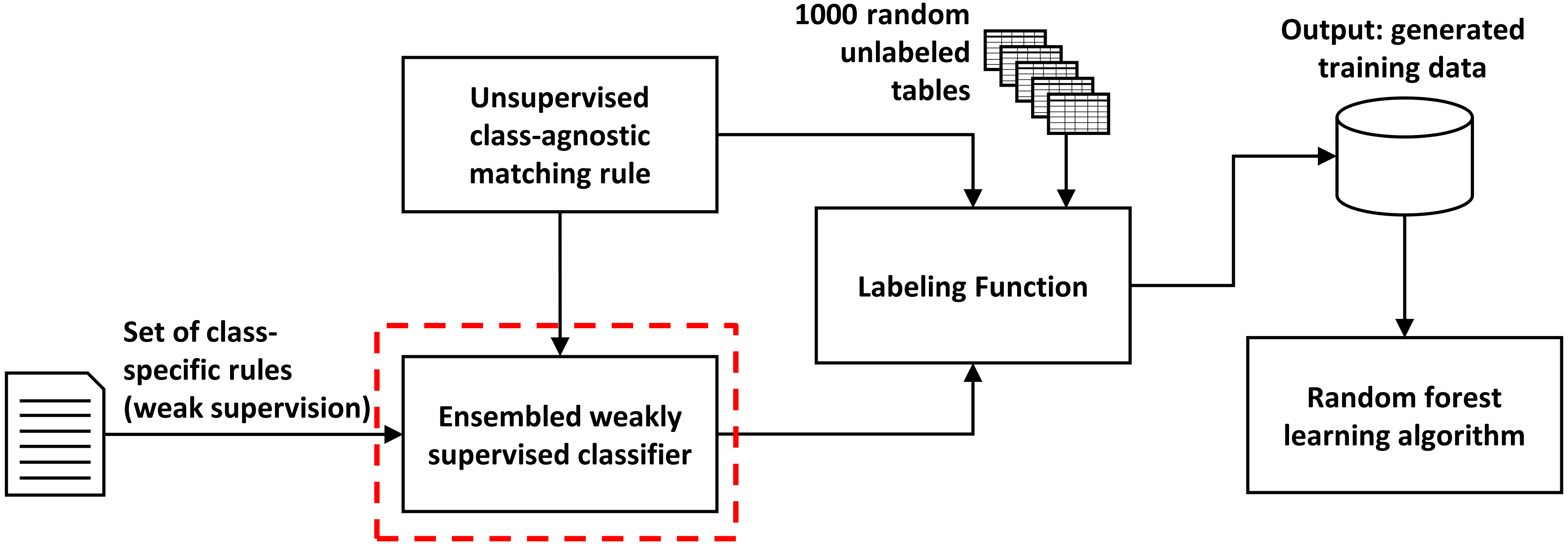
Rule Execution

Rules are executed using the class-specific ATTRIBUTE and IMPLICT_ATT features, which return **a similarity score per property**.

Using these scores:

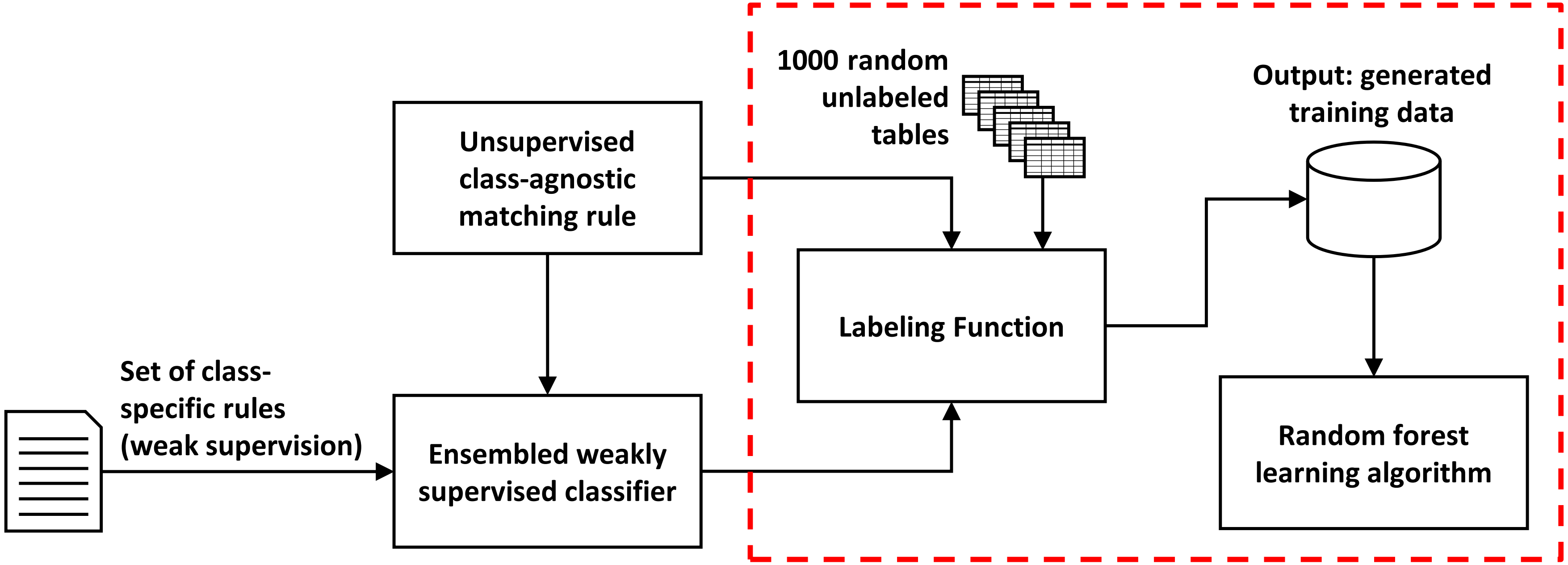
1. we determine **when a rule fires**
2. we determine **the confidence** of the classification

Ensembling



- **Ensemble** rules unsupervised matching rule **to increase coverage**.
- When multiple rules fire, we consider the one with **highest confidence**.
- We **average** the output of the fired rule and the unsupervised model and return a classification (along with a confidence score).

Bootstrapping



Given a **labeling function**:

1. We select 1000 random tables form the web corpus to annotate
2. We select **row-pairs** and **entity-instance-pairs** using label blocking (Lucene)
3. Label pairs using labeling function as either matching and non-matching pairs
4. Using the labeled pairs as training examples we train a random forest classifier (Labeled training examples are weighted by their classification confidence)

Experiments

Experimental Setup

- We evaluate our approaches on T4LTE Gold Standard
- It uses **DBpedia** as the target knowledge base to be extended
- We evaluate
  1. row clustering performance
  2. new detection performance
  3. end-to-end performance
- We compare our approaches with strong supervision
  (Using 3-fold CV throughout all experiments)

Row Clustering Performance

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
Unsupervised	0.76	0.86	0.80	0.90	0.65	0.86
Weak supervision	0.83	0.89	0.86	0.93	0.81	0.84
+ Bootstrapping	0.83	0.90	0.86	0.89	0.83	0.86
Strong supervision	0.86	0.90	0.88	0.91	0.84	0.90

New Detection Performance

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
Unsupervised	0.87	0.76	0.80	0.82	0.68	0.89
Weak supervision	0.87	0.81	0.83	0.82	0.78	0.89
+ Bootstrapping	0.87	0.90	0.87	0.87	0.85	0.90
Strong supervision	0.82	0.94	0.87	0.88	0.92	0.81

End-To-End Performance

Method	Average P	Average R	Average F1	GF-Player F1	Song F1	Settlement F1
Unsupervised	0.71	0.71	0.69	0.76	0.50	0.82
Weak supervision	0.72	0.77	0.74	0.76	0.63	0.82
+ Bootstrapping	0.72	0.86	0.78	0.81	0.72	0.80
Strong supervision	0.73	0.93	0.81	0.84	0.78	0.81

Bootstrapping and Matching Rules

	Row Clustering	New Detection
Pairs To Be Labeled	2.8m	1.27m
Matching Pairs	275k	26k
Positive Rules Firings	37k (13%)	13k (50%)
Non-Matching pairs	2.54m	1.27m
Negative Rule Firings	500k (20%)	150k (12%)

Importance of Ensembling

UNIVERSITY OF MANNHEIM
Data and Web Science Group

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
MR Unensembled	0.43	0.05	0.09	0.00	0.14	0.14
+ Bootstrapping	0.47	0.58	0.34	0.14	0.74	0.15
MR Ensembled	0.72	0.77	0.74	0.76	0.63	0.82
+ Bootstrapping	0.72	0.86	0.78	0.81	0.72	0.80

Discussion & Conclusion

Weak Supervision Using Bold Rules

- **Little effort** is required for creating rules
- Rules could be **mined** from or tested on the knowledge base
- **Ensembling** provides **full coverage**
- **Limitation:** requires web tables to describe entities using useful knowledge base attributes

Bootstrapping

Using bootstrapping we can learn a model that **outperforms** the labeling function from which it was bootstrapped.

The trained random forest:

- can exploit **more class-specific similarity features**
- is **more expressive** than the unsupervised model or the matching rules

Conclusion

- Approach substitutes **thousands of manually labeled entity matches** with a **small set of user-provided bold class-specific matching rules** when training a supervised learning algorithm.
- Enables cross-domain long-tail entity extraction with little supervision effort
- **Potential for bootstrapping active learning:**
  - We can reduce effort spent on learning initial models considerably
  - Learned models can be refined by labeling individual selected examples

Thanks for Listening

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Links:

- **Web Tables for Long-Tail Entity Extraction**
  http://webdatacommons.org/T4LTE/
- **Extracting Long Tail Entities from Web Tables for Augmenting Cross-Domain Knowledge Bases -**
  http://data.dws.informatik.uni-mannheim.de/expansion/LTEE/

References:

- **[Cafarella2008]** Cafarella, M. J., Halevy, A. Y., Zhang, Y., Wang, D. Z., and Wu, E. (2008). Uncovering the relational web. WebDB ’08.
- **[Oulabi2019]** Oulabi, Y. and Bizer, C. (2019). Extending cross-domain knowledge bases with long tail entities using web table data. EDBT ’19.
- **[Ritze2016]** Ritze, D., Lehmberg, O., Oulabi, Y., and Bizer, C. (2016). Profiling the potential of web tables for augmenting cross-domain knowledge bases. WWW’16.
- **[Ratner2016]** Ratner, A. J., Sa, C. D., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. NIPS ’16.
- **[Ratner2017]** Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. Proc. VLDB Endow., 11(3):269–282.

Row Clustering Performance

UNIVERSITY OF MANNHEIM
Data and Web Science Group

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
Unsupervised	0.76	0.86	0.80	0.90	0.65	0.86
+ Bootstrapping	0.78	0.88	0.83	0.89	0.73	0.86
Weak supervision	0.83	0.89	0.86	0.93	0.81	0.84
+ Bootstrapping	0.83	0.90	0.86	0.89	0.83	0.86
Strong supervision	0.86	0.90	0.88	0.91	0.84	0.90

New Detection Performance

UNIVERSITY OF MANNHEIM
Data and Web Science Group

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
Unsupervised	0.87	0.76	0.80	0.82	0.68	0.89
+ Bootstrapping	0.86	0.86	0.85	0.86	0.78	0.90
Weak supervision	0.87	0.81	0.83	0.82	0.78	0.89
+ Bootstrapping	0.87	0.90	0.87	0.87	0.85	0.90
Strong supervision	0.82	0.94	0.87	0.88	0.92	0.81

End-To-End Performance

UNIVERSITY OF MANNHEIM
Data and Web Science Group

	Average			GF-Player	Song	Settlement
Method	**P**	**R**	**F1**	**F1**	**F1**	**F1**
Unsupervised	0.71	0.71	0.69	0.76	0.50	0.82
+ Bootstrapping	0.71	0.81	0.74	0.79	0.60	0.82
Weak supervision	0.72	0.77	0.74	0.76	0.63	0.82
+ Bootstrapping	0.72	0.86	0.78	0.81	0.72	0.80
Strong supervision	0.73	0.93	0.81	0.84	0.78	0.81