ENTITY UNIFICATION FOR SEMANTIC SEARCH

Albert-Ludwigs-University Freiburg
2013
Anton Stepan

Roadmap

- What is the problem?
- Our Idea
- Algorithm
- Evaluation
- Problems & Improvements

Problem

- **Unification of two or more ontologies** (Triple Datasets)
- Different ontologies with different naming conventions
- Multiple entities with same names
- Which of them belong together?

source1

…
Berlin_1
Berlin_2
Berlin_3
Berlin_4
Berlin_5
Berlin_6
…

source2

…
Berlin_a
Berlin_b
Berlin_c
…

Unification with the help of more information

→ further information about entities

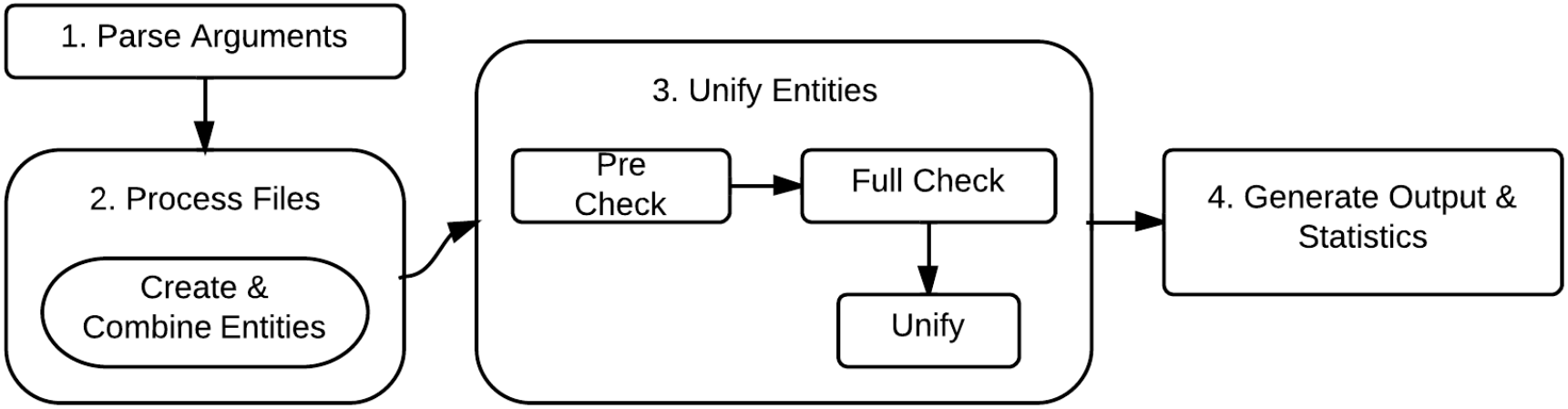
…		
Berlin	located-in	Germany
Berlin	has-longitude	52.31
Berlin	has-latitude	13.24
Berlin	located-in	Berlin,_(District)
Berlin	has-population	3,375,222
…		
Germany	contains	Berlin
…		

Our Algorithm Idea/Approach

- **Modular**
  - → Replaceable sub-parts
  - → tweakable

- **Scores**
  - Different scores for different similarities
  - Tweakable by user / Set focus
    - …without recompiling

Algorithm Outline



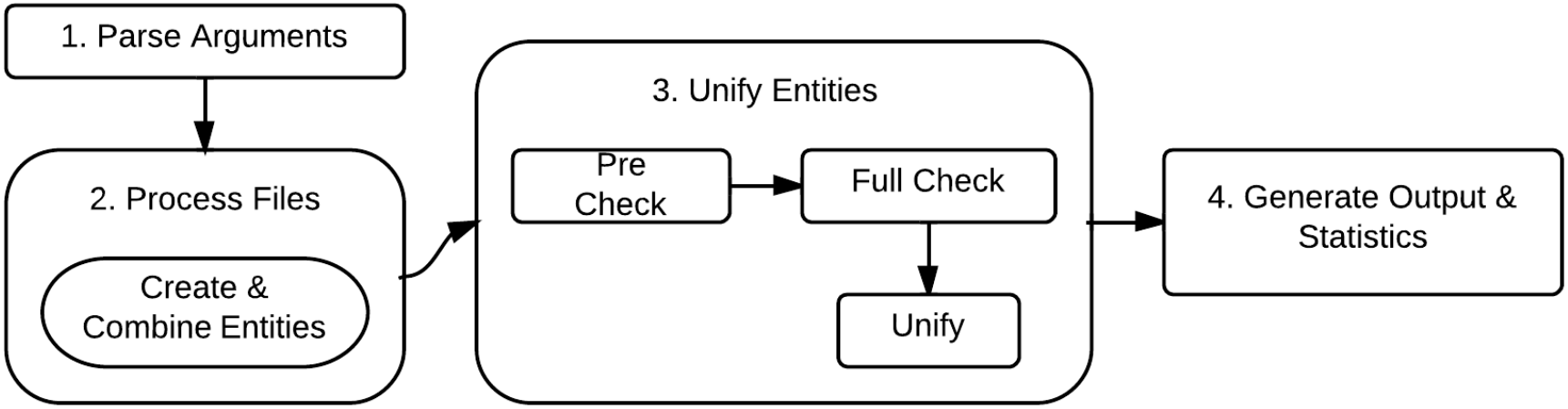
Occuring Problems in Unification Procedure

- Multiple entities with the same name
  → Relation comparison
- Entities with slightly different names
  → Prefix check
- Same entities with different names
  - UTF8, ASCII, …
  - Native names, English names
- Entities with sparse relations
  → Iterations can help

Occuring Problems in Unification Procedure

- Different entities with similar names and similar relations
  → |words|-check
- Relations with different names
  → Relationsmap
- Mistakes in the database
  → scores and thresholds

Algorithm Outline



1. Parse Arguments

- Required
  - Filenames: Input 1 & 2
  - Scores

- Optional
  - Default Folder with config-file
  - Output filename
  - Relationmap (translate relations: „located“ → „located-in“
  - Iterations
  - Debug
  - Generate Example Files (config, relationmap, scores)

2. Process files

Triples: „Subject <tab> Relation <tab> Object“

„Berlin located-in Germany“

„Berlin located-in Berlin,_(District)“

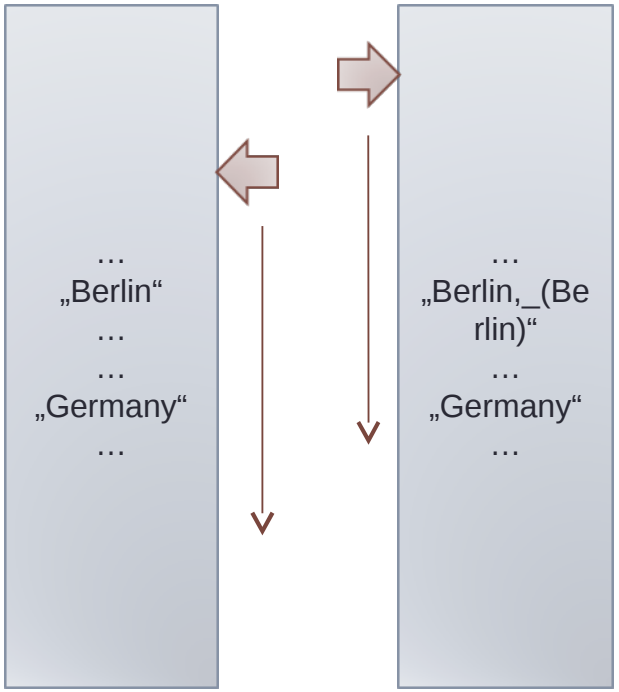
„Freiburg located-in Germany“

- Two Maps: ID → EntityPtr*
  - std::map<std::string, EntityPtr*> map1

- EntityPtr (datastructure)
  - Containing Pointer to real Entity
  - Possible further information

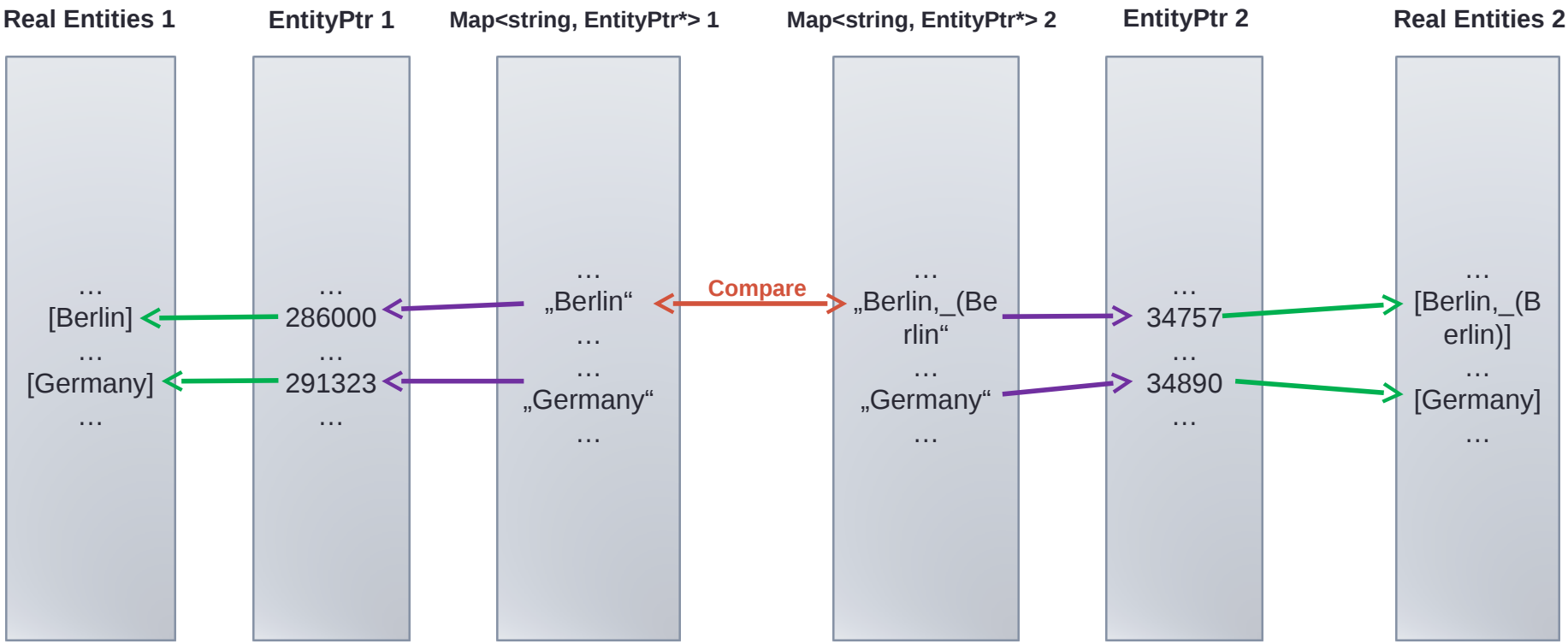
3. Unify

- Pre Check
  - Possible equal?
  - Prefixcheck + |Words|-check
- Full Check
  - Comparing relations
  - Computing scores
- Unify
  - if ($Score_{OVERALL}$ > Threshold)
  - Reallocating EntityPtr
  - Merging relations



UNIFY Step 0 - comparison

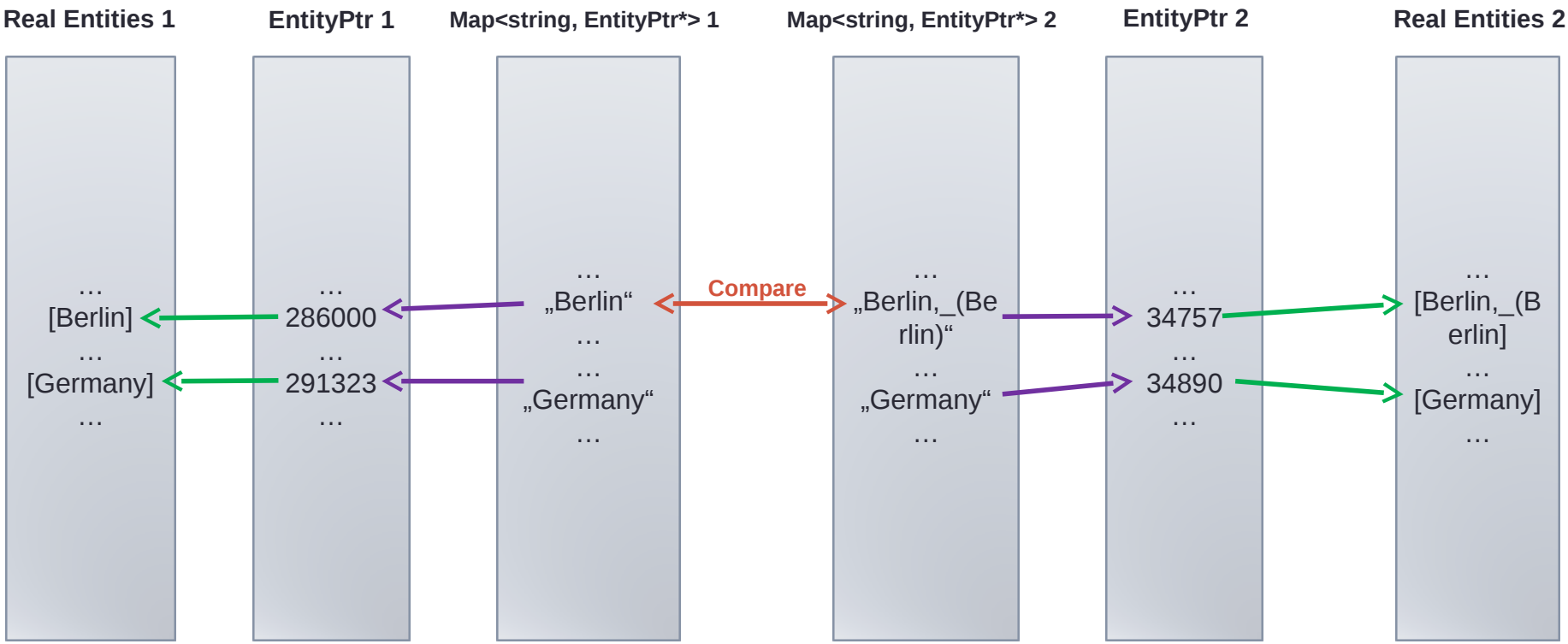
- **Goal:** Unification of „*Berlin*“ and „*Berlin,_(Berlin)*“



→ Relations of „Berlin“ and „Berlin,_(Berlin)“ were compared and $score_{OVERALL}$ is bigger than threshold.

UNIFY Step 1 – merge flag & ID

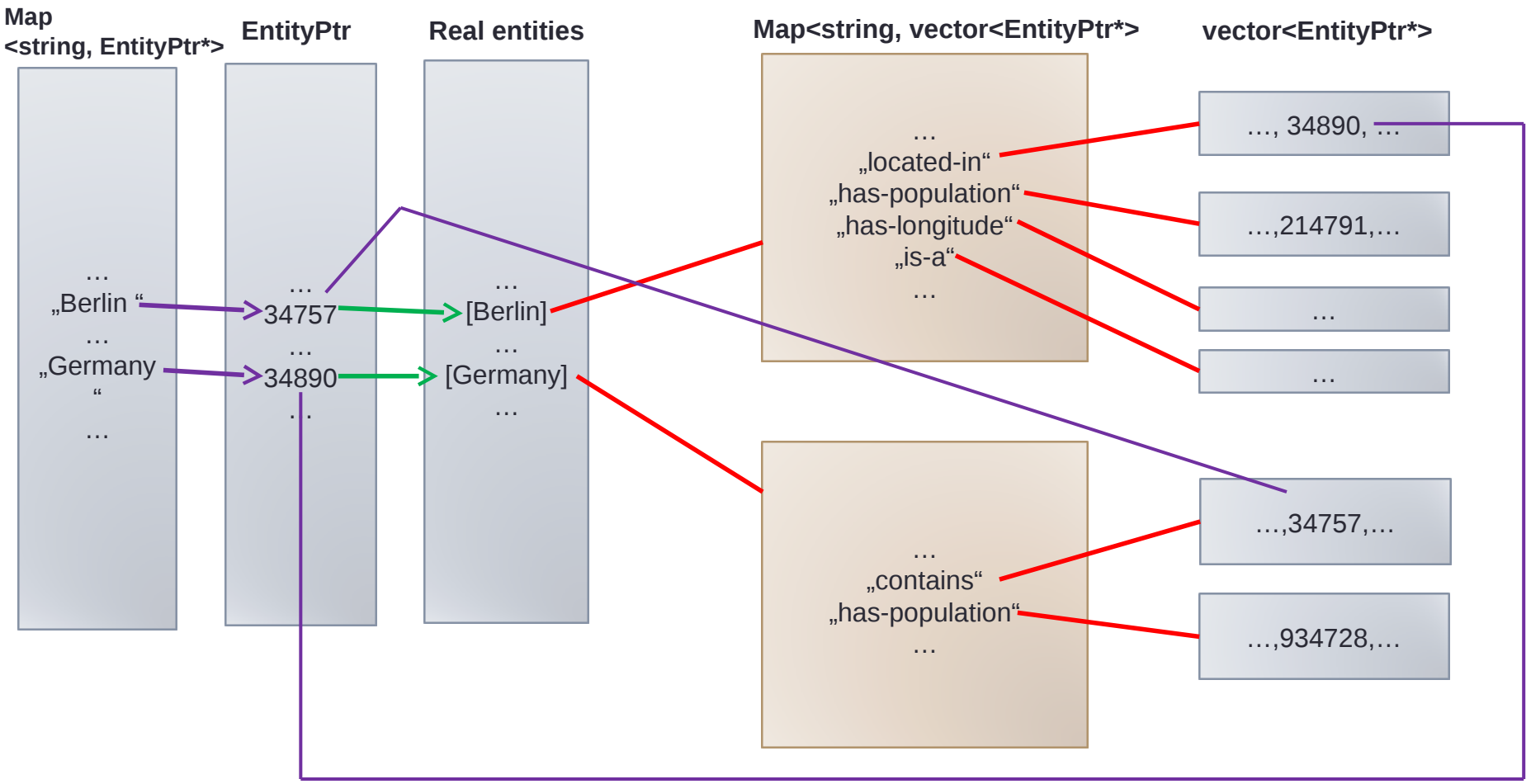
- **Goal:** Unification of „*Berlin*“ and „*Berlin,_(Berlin)*“



→ **Set merge flag to true & add ID**

```
map[„Berlin“]→getPtr()→setMerged(true);
```

UNIFY Step 2 – unify relations



UNIFY Step 2 – unify relations

map1[„Berlin“]->getPtr()->relations

…
„located-in“
„has-population“
„has-longitude“
„is-a“
…

map2[„Berlin,_(Berlin)“]->getPtr()->relations

…
„located-in“
„has-population“
„has-latitude“
„is-a“
…

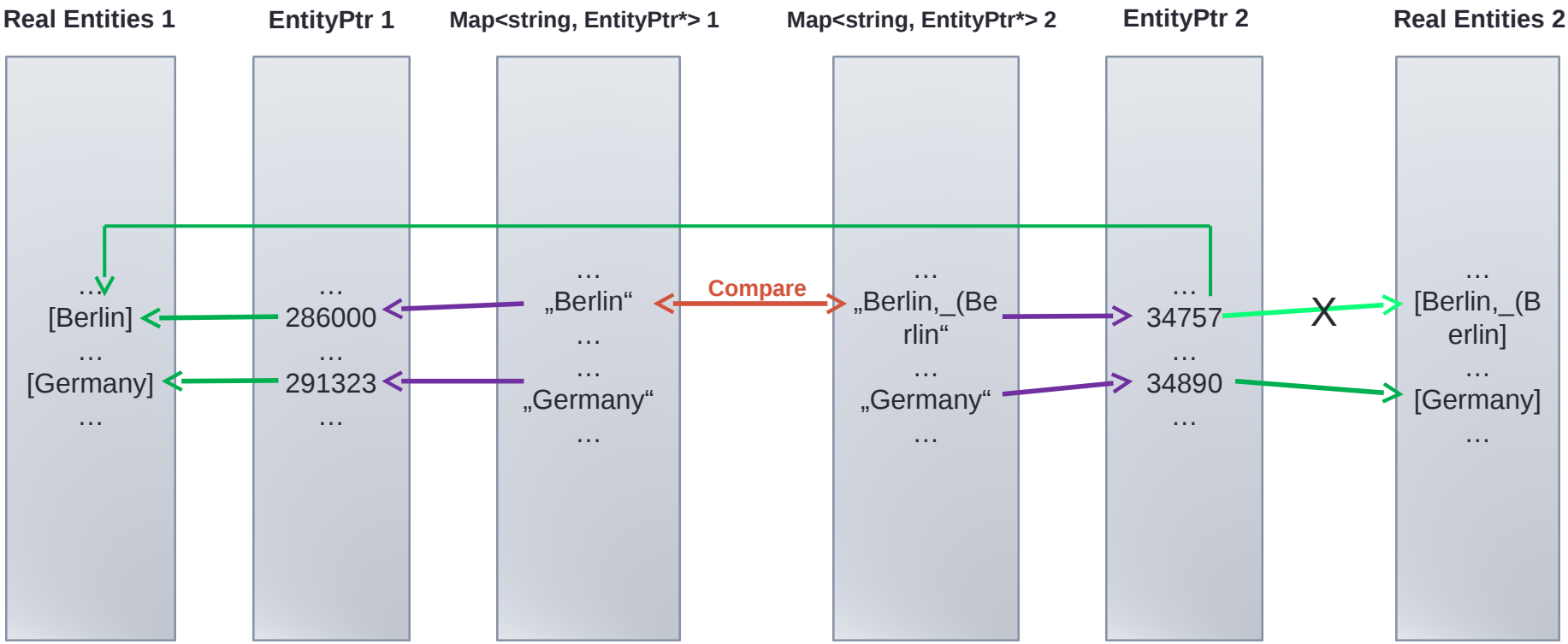
- Each entity E has a relation set R_E
- all triples: *E relationname Object*

- $R_E = \{(r_i.name, f(r_i)) : r_i \in relations_{out}(E)\}$
- with r_i is the set of relation targets, i.e. $f(r_i) = \{ y : (E, y) \in R_i\}$

- → unification of relations = unification of two sets

UNIFY Step 3 – Reallocating

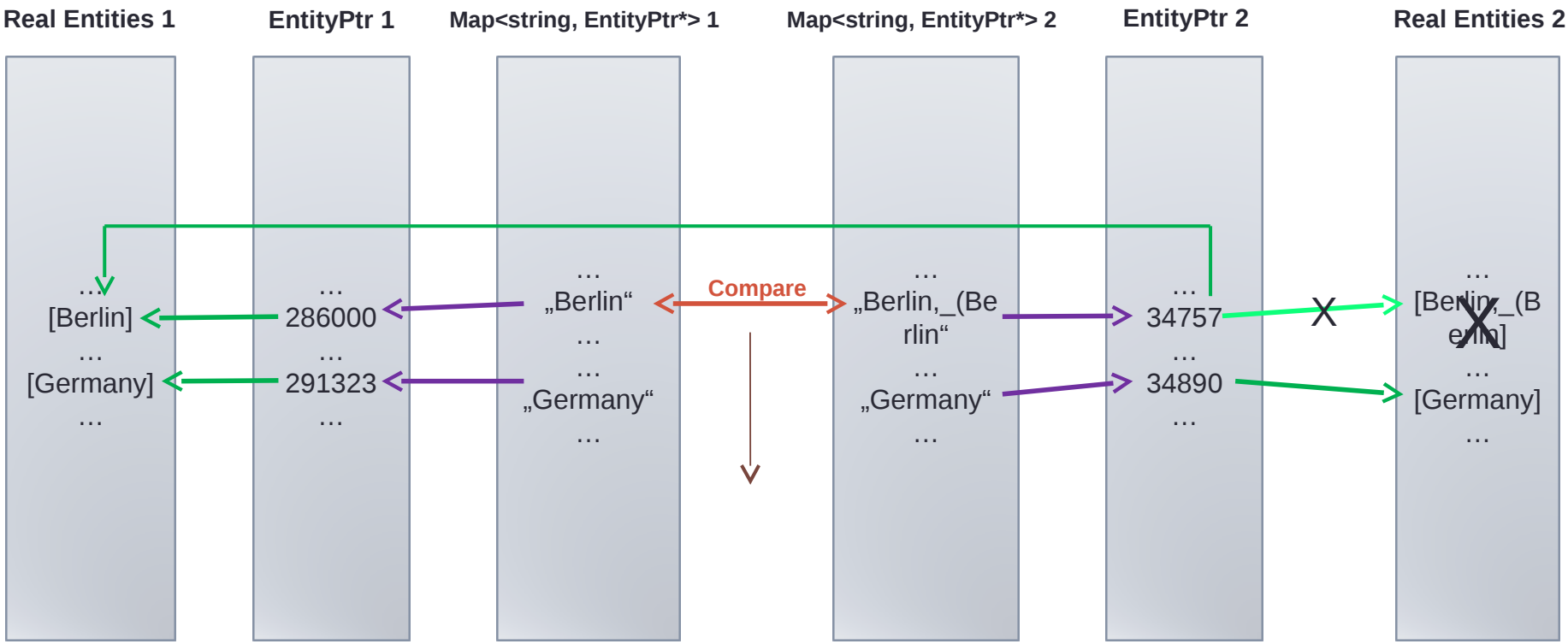
- **Goal:** Unification of „*Berlin*“ and „*Berlin,_(Berlin)*“



Reallocation the EntityPtr of „*Berlin,_(Berlin)*“
→ All relations with target [Berlin,_(Berlin)] now also point to [Berlin]

UNIFY Step 4 – Deleting [Berlin,…]

- **Goal:** Unification of „*Berlin*“ and „*Berlin,_(Berlin)*“



Evaluation

- Two datasets based on Geonames and Freebase

Dataset	#Lines	#Entities	Filesize
Geonames	813,489	383,421	37 MB
Freebase	4,710,584	3,006,213	244 MB

- Result

ID	Debug	Iterations	Avg. Elapsed Time (Unification Phase)	Unification Count	Unification percentage
1	Off	1	15.21 s	161,746	42.18 %
2	Off	2	22.68 s	197,500	51.50 %
3	Off	3	27.98 s	203,694	53.12 %
4	Off	20	64.44 s	205,897	53.69 %
5	On	1	2.22 min	161,746	42.18 %
6	On	2	5.13 min	197,500	51.50 %

Problems & Improvements

- Different entity names
  - „Nordrhein-Westfalen“ VS „North Rhine-Westphalia”

  → Entity-Translation-Map
- Same name with different meaning
  - Geonames
    - “Freiburg” <the city>
    - “Freiburg Region” <the region>
  - Freebase
    - “Freiburg im Breisgau” <the city>
    - “Freiburg” <the region>
  - City and Region share same information
- Special Places

Live Demo