VU VRIJE UNIVERSITEIT AMSTERDAM

Refining Large Identity Graphs using the Unique Name Assumption

Under submission at ESWC 2022 (research track)

Shuai Wang, Joe Raad, Peter Bloem, Frank van Harmelen

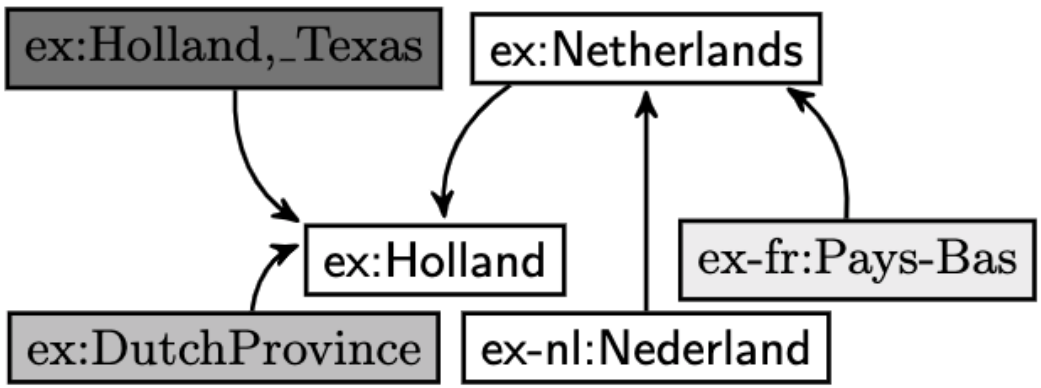
KR&R Group, Vrije Universiteit Amsterdam

10th January, 2022

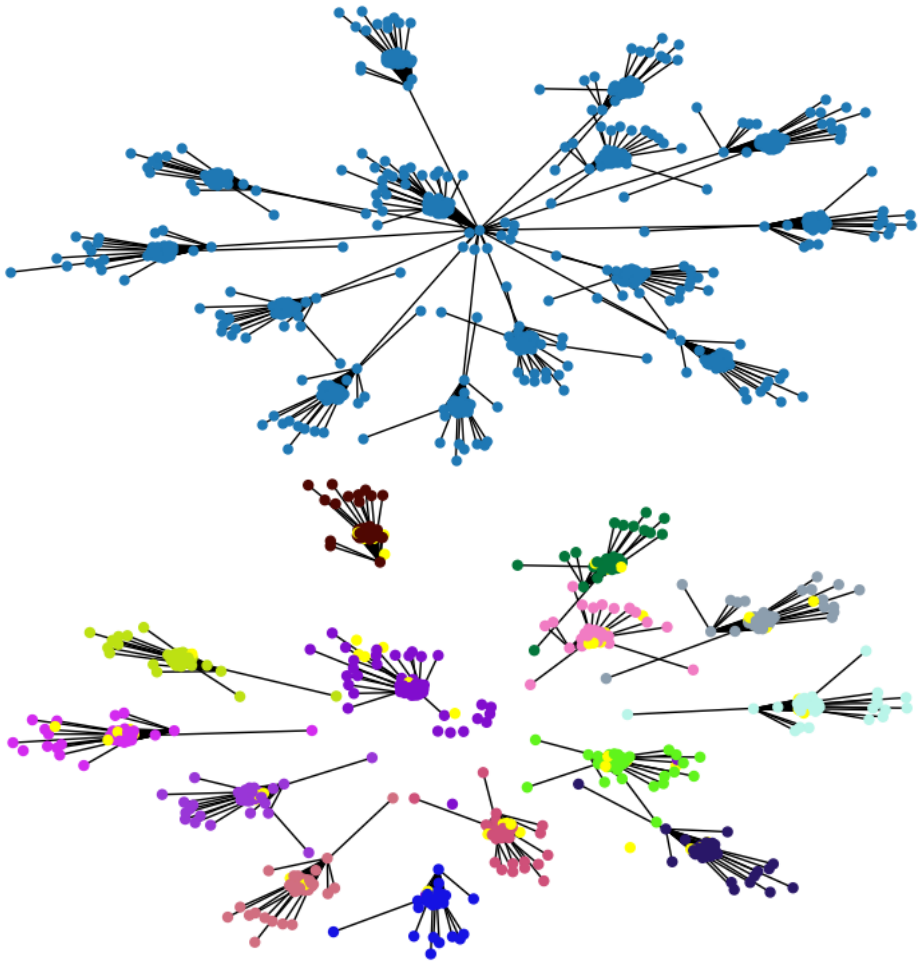
Content

Introduction

- `(A, owl:sameAs, B)`.
- `owl:sameAs` is an equivalence relation : transitive, symmetric, reflexive.
- Error rate: 3 - 4%, or as high as 20%.

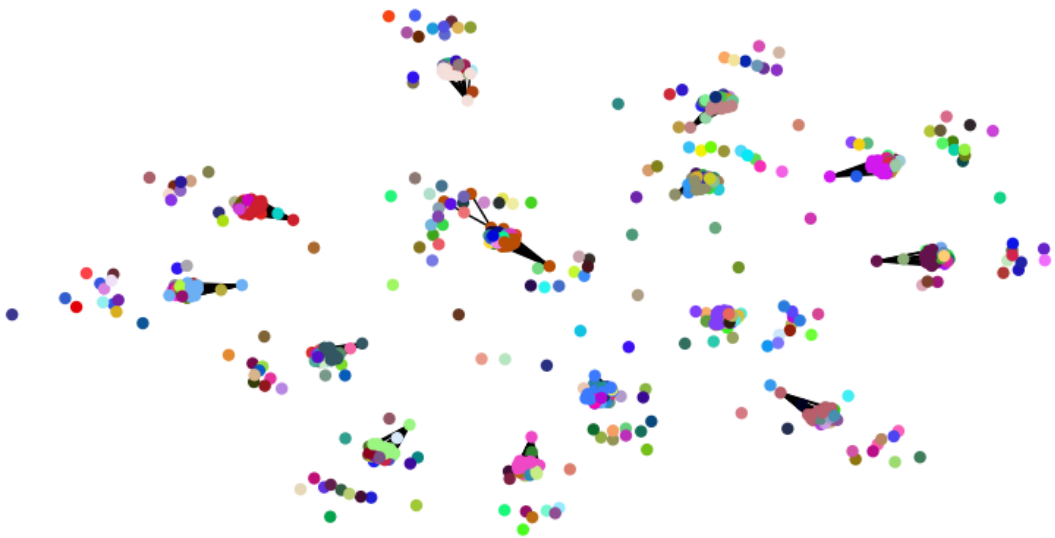


Introduction



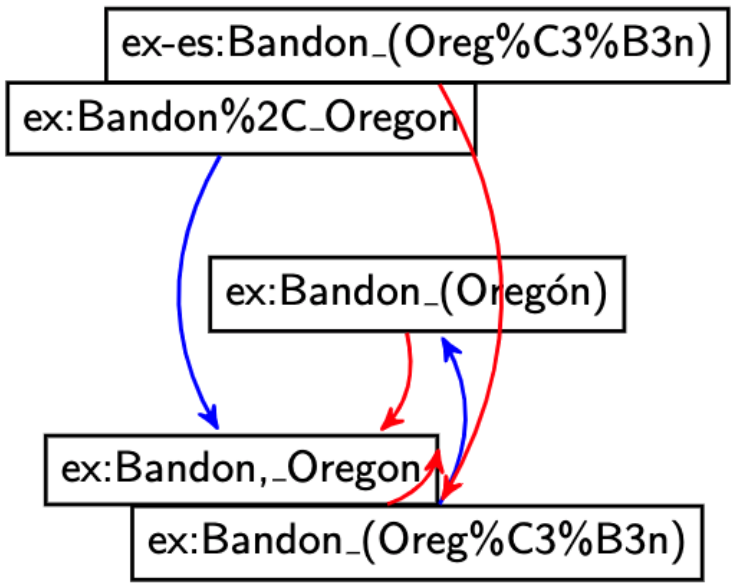
Related Work

- content-based approach
- network-based approach (Louvain)
- inconsistency-based approach



The UNA

- the *naive UNA* (*nUNA*): any two URIs in the same knowledge base cannot refer to the same thing in the real world.
- the *quasi UNA* (*qUNA*) extends this definition by taking the redirect relations (between 6 major hubs) and dead nodes into account.
- we also found entities that only differ in encoding: encoding equivalence.



The Challenge

- need a large gold standard (so no reliable evaluation)
- no redirect graphs
- no graphs about encoding equivalence
- no definition about provenance
- no UNA definitions has been validated at large scale

Provenance

let $\eta(e_i)$ be the sources of an entity e_i.

Explicit sources: the files where there are triples of `rdfs:isDefinedBy`*.

Implicit label-like sources: the files where there are triples of `rdfs:label`*.

Implicit comment-like sources: the files where there are triples of `rdfs:comment`*.

* or any equivalent relation or sub-properties

internal UNA (iUNA)

iUNA: two different IRIs within the same namespace should refer to distinct real-world entities only when they are defined in the same source.
Exemptions:

- redirects
- encoding equivalence
- exceptions while resolving the IRI:
  - dead node
  - not found
  - unresolvable
  - redirects until reaching some error or not found
  - or has timeout error while resolving

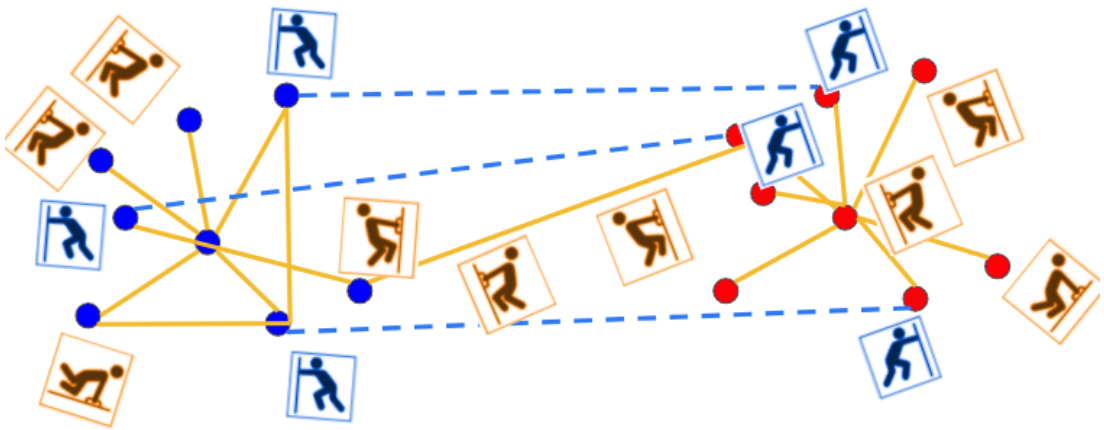
The Problem

Issue A very large subgraph about `owl:sameAs` (550 million nodes in LOD-a-lot).

Task remove as few edges as possible.

Complexity = APX-hard (has a polynomial-time approximation).

Intuition pull & push

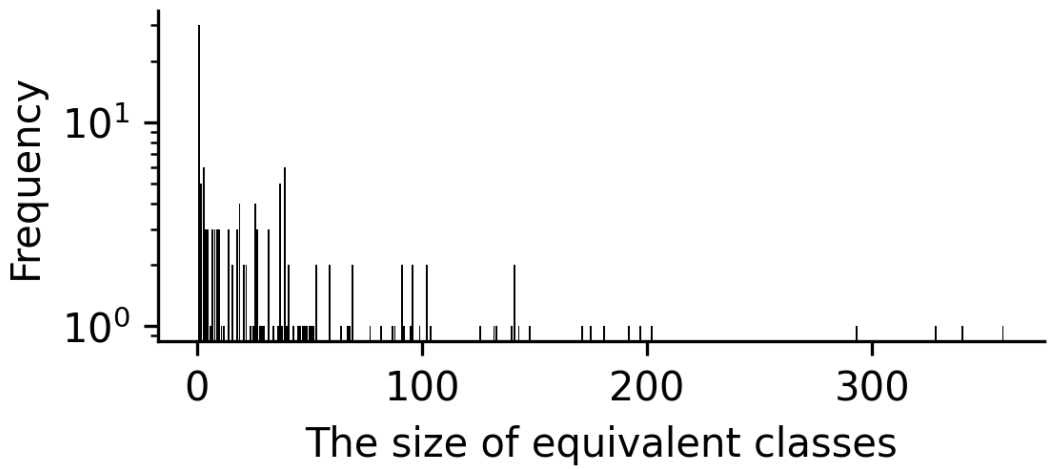


Research Questions

- **RQ1:** how can we define UNA for large integrated knowledge graphs?
- **RQ2:** how do we validate the definitions proposed?
- **RQ3:** can UNA give a reliable indication of identity errors in practise?
- **RQ4:** can we define an efficient algorithm for the refinement of the identity graphs?
- **RQ5:** is it possible to improve the results using additional information from the graph?

The Gold Standard

- 8,394 manually annotated entities in 28 files
- a total of 232,311 `owl:sameAs` links
- 11.75% entities are ‘unknown’
- the error rate is between 1.58% and 9.98%



Validating UNA

Does the UNA hold using label-like sources?

- nUNA: 93.50%
- qUNA: 94.43%
- iUNA: 94.11%

Using comment-like sources:

- nUNA: 97.46%
- qUNA: 96.77%
- iUNA: 97.09%

Yes, very much so!

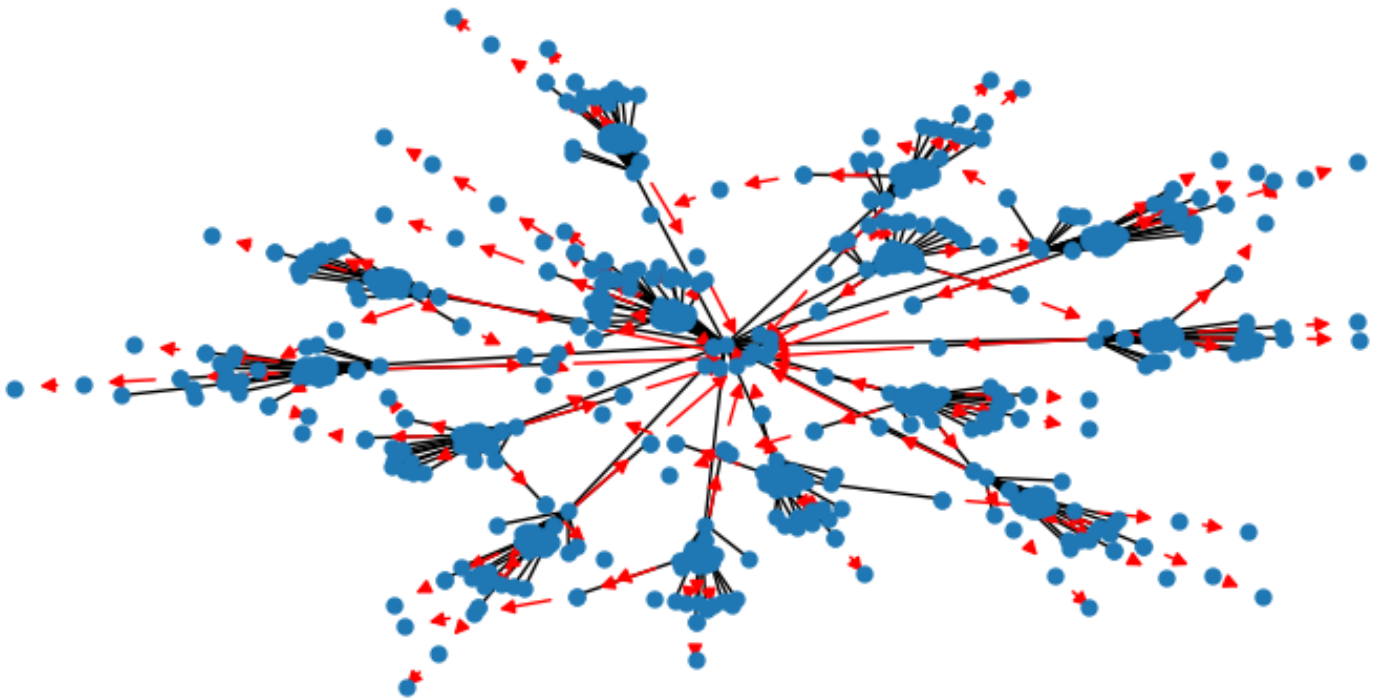
Reliability

RQ3: Can the UNA give a reliable indication of identity errors in practice?
Baseline (label-like sources): error rate of random pairs: 47-68%.

- nUNA: 61.79% pairs violate; error: 33.31 - 49.89%.
- qUNA: 41.23% pairs violate; error: 33.28 - 51.87%.
- iUNA: 0.78% pairs violate; error: 6.10 - 36.69%.

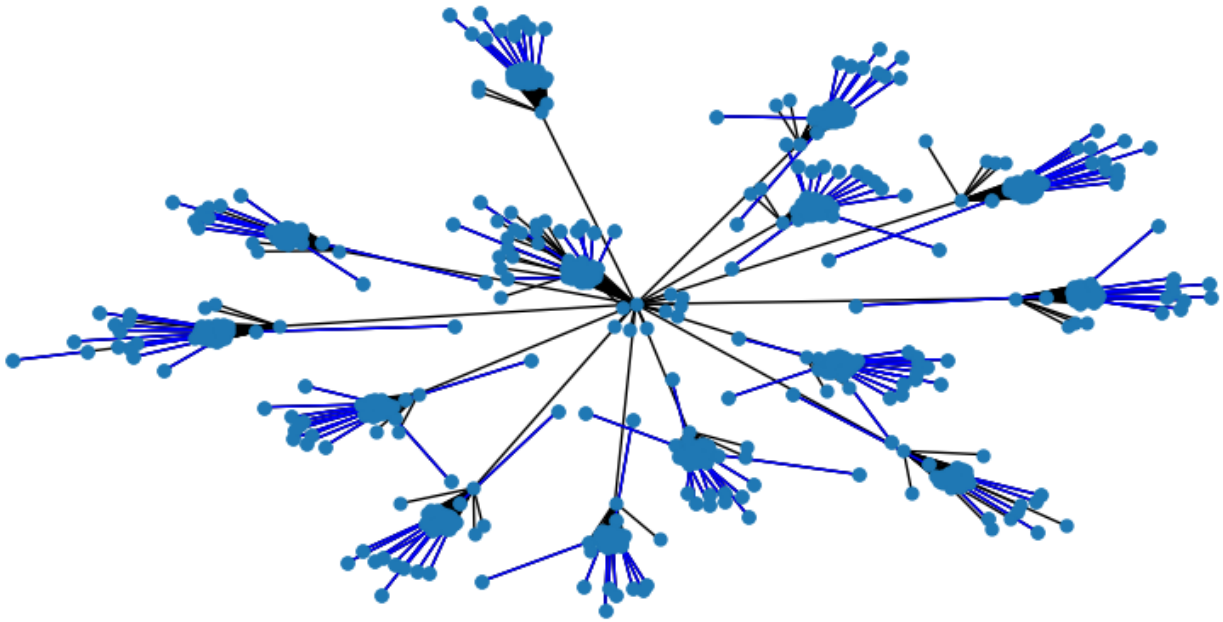
Reliability: Redirect

Among them existing edges: error rate 1.47 - 7.69%.
Others: error rate 4.29 - 6.32%.



Reliability: Encoding Equivalence

Among them existing edges: error rate 2.21 - 8.50%.
Others: error rate 1.16% and 14.83%.



Algorithm

1. compute the minimum spanning forest (to reduce the load)
2. sample some edges from the original graph
3. assign an integer to each node
4. the (weighted) clauses are equivalence relations between these integers
5. positive weights for existing edges
6. negative weights for pairs violating the (chosen) UNA.
7. repeat until no such pair found or no edge removed

Algorithm

Algorithm 1: partition

Input: an identity graph G, a graph of redirect G^R, a graph of equivalence under various encodings G^E, a weighting scheme w
Result: status s, a set of edges removed A, the graph of partitions G_P
initiate A as an empty set;
let status s, removed_edges A = partition_iter(G, G^R, G^E, w);
if s *is error* **then**
    return ('error', $\emptyset$, G)
while $|A|$ *is not increasing (no new edge to remove)* **do**
    let H be the new graph of G with A removed;
    obtain H_{ccs}, the graphs for each connected component of G' ;
    foreach $H_{cc} \in H_{ccs}$ **do**
        obtain the corresponding subgraphs H^R_{cc}, H^E_{cc} from G^R, G^E respectively;
        (s', A') = partition_iter(H_{cc}, H^R, H^E, w);
        if s' *is not 'error'* **then**
            $A := A \cup A'$
        else
            return ('error', A)
remove A from G to get G_P;
return ('success', A, G_P).

Algorithm 2: partition_iter

Input: a graph of connected component G_{cc}, a graph of redirect G^R_{cc}, a graph of equivalence under various encodings G^E_{cc}, a weighting scheme w
Result: s, a set of edges removed A_{cc}
obtain a set of pairs P violating the UNA;
if $|P| \leq 1$ **then**
    return ('success', $\emptyset$)
initiate an SMT solver o;
\# hard clauses ;
foreach *entity* s *in* G_{cc} **do**
    we introduce an integer variable I_s and assert hard clauses $(0 \leq I_s)$ and $(I_s \leq M)$
\# soft clauses ;
let F be the minimum spanning forest of G_{cc};
randomly sample a small portion of B edges in G_{cc} and add to F;
obtain G'^R_{cc} the undirected graph of the (directed) graph G^R_{cc};
foreach *pair* $f = (s, t)$ *in* $F \cup P$ **do**
    initiate a soft clause c_f according to w
foreach *pair* $r = (s, t)$ *in* $G'^R_{cc} \cup G^E_{cc}$ **do**
    if s *and* t *are reachable* **then**
        initiate/update the weight of a soft clause c_r according to w
add all soft and hard clauses to o;
let s be the result of o after solving and m be the model (if any);
if *status* s *is 'unknown'* **then**
    return ('error', $\emptyset$)
else
    check each edge f of F, against the model m;
    let A_{cc} be the set of all removed edges of F;
    return ('success', A_{cc}).

Algorithm: Weights

- the weighting scheme: $w = (f_G, f_R, f_E, f_P)$
- the weight of an edge c_e:
  $f_G(c_e) + f_R(c_e) + f_E(c_e) + f_P(c_e)$
- two weighting schemes for evaluation: w_1 and w_2

Evaluation

- Why precision-recall doesn't work anymore?
- a new measure

$$\Omega(G') = \sum_{C \in G'_{ccs}} \sum_{Q_e \in E(C)} \frac{|Q_e|}{|V|} \frac{|Q_e|}{|O_e|} \frac{|Q_e|}{|C|}.$$

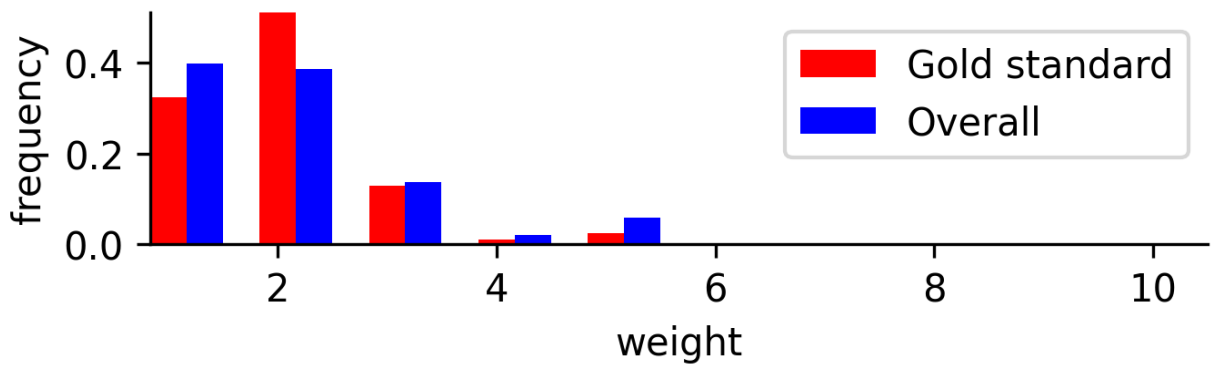
- C iterates over all connected components in G'
- $E(C)$ is a partitioning of the nodes in C by equivalent class
- O_e is the set of all entities in G' referring to e.

Evaluation

	training set				evaluation set			
	precision	recall	Ω	$\lvert A\rvert$	precision	recall	Ω	$\lvert A\rvert$
Louvain algorithm	0.020	**0.759**	0.084	39,302	0.039	**0.660**	0.083	43,642
qUNA-label-w_1	0.300	0.061	0.587	**14**	**0.417**	0.006	0.607	57
qUNA-label-w_2	0.237	0.083	0.618	88	0.167	0.004	0.576	53
qUNA-comment-w_1	**0.324**	0.031	0.595	**14**	0.244	0.004	0.562	**24**
qUNA-comment-w_2	0.236	0.104	0.614	91	0.199	0.021	0.591	79
iUNA-label-w_1	0.186	0.077	0.605	101	0.086	0.026	0.585	35
iUNA-label-w_2	0.168	0.108	**0.619**	262	0.065	0.016	**0.617**	175
iUNA-comment-w_1	0.187	0.053	0.609	91	0.146	0.009	0.575	42
iUNA-comment-w_2	0.084	0.003	0.618	114	0.072	0.026	0.610	130

Evaluation

- weight: number of sources regarding each triple ($0 < w < 650k$).
- DBpedia disambiguation nodes: corresponding to Wikipedia disambiguation pages.



Evaluation: improving the results

	training set				evaluation set			
	precision	recall	Ω	$\|A\|$	precision	recall	Ω	$\|A\|$
iUNA-label-w_2	0.168	0.108	**0.619**	262	0.065	0.016	0.617	175
iUNA-label-w_2+weight	0.217	0.108	0.610	233	0.050	0.015	0.614	162
iUNA-label-w_2+disambiguation	0.221	0.135	0.615	264	0.098	**0.030**	**0.642**	191
qUNA-comment-w_1	0.324	0.031	0.595	**14**	**0.244**	0.004	0.562	**24**
qUNA-comment-w_1+weight	0.159	0.016	0.579	17	0.111	0.002	0.575	27
qUNA-comment-w_1+disambiguation	**0.412**	**0.163**	0.573	209	0.133	0.005	0.578	43

List of Contributions

1. **UNA**: validated and checked the reliability of nUNA, qUNA, and iUNA.
2. **Algorithm** but does not scale to 177k.
3. **Datasets**: redirect, weights, encoding equivalence, disambiguation, etc.
4. **Gold standard**
5. **Results**: and how we improved it using additional information.

Discussion

- 211,348 out of 232,311 edges (90.98%) are about DBpedia entities between different languages
- 177k nodes in the largest weakly connected component.
- only 5 have different label-like or comment-like sources: UNA is not about the source of errors.
- Next: evaluate using more methods
- Next: Deep Learning on the identity graph!

Thank You for your attention!

Contact: shuai.wang@vu.nl