Identifying Equivalent Relation Paths in Knowledge Graphs

Sameh K. Mohamed, Emir Muñoz, Vít Nováček, Pierre-Yves Vandenbussche

Language, Data and Knowledge (LDK) Conference. Galway, Ireland. 18~21 June 2017

Agenda

- Knowledge Graphs
- Relation Path Equivalence
- Search of Δ-Equivalences
  - Path extension Extraction
  - Subgraph extraction
  - Building connecting Paths
  - Ranking candidate paths
- Experiments & Results
- Conclusions
- Future work

Knowledge Graphs (KGs)

- Graph representation of knowledge bases, where entities are represented by nodes and relations are represented by edges between them
- Commonly, KGs model facts as Subject-Predicate-Object (SPO) triples.
- A sequence of connected edges in a knowledge graph is called *path*

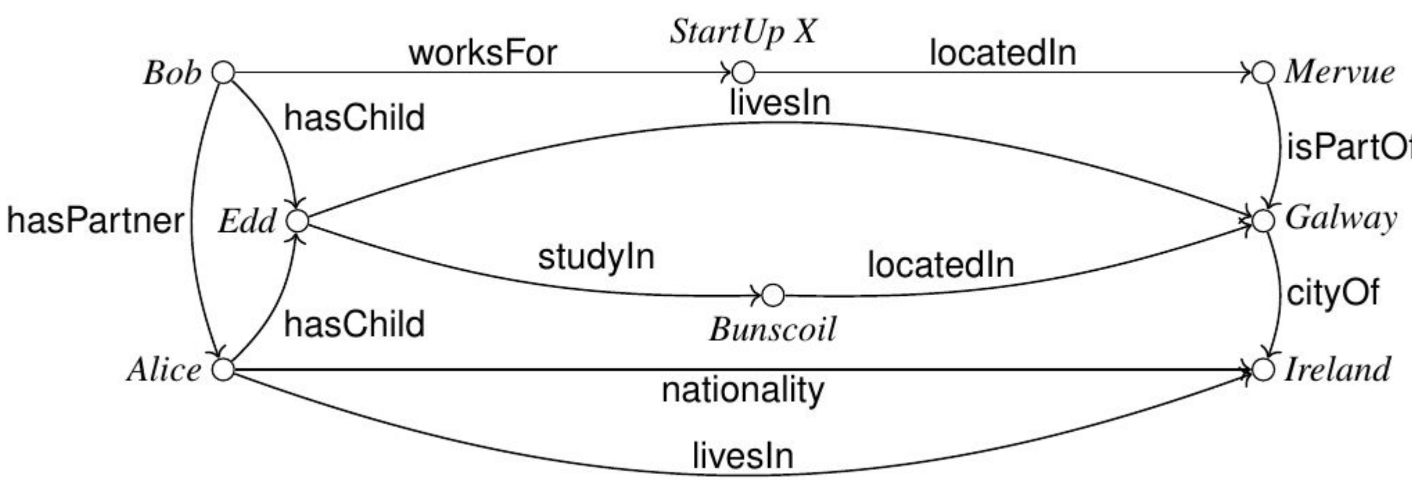


Figure 1: Example of a knowledge graph about people living in Galway, Ireland

Knowledge Graph Paths

- Relation path is a graph path represented as a sequence of its relations.
- Paths can be navigated in both directions using inverses of relations

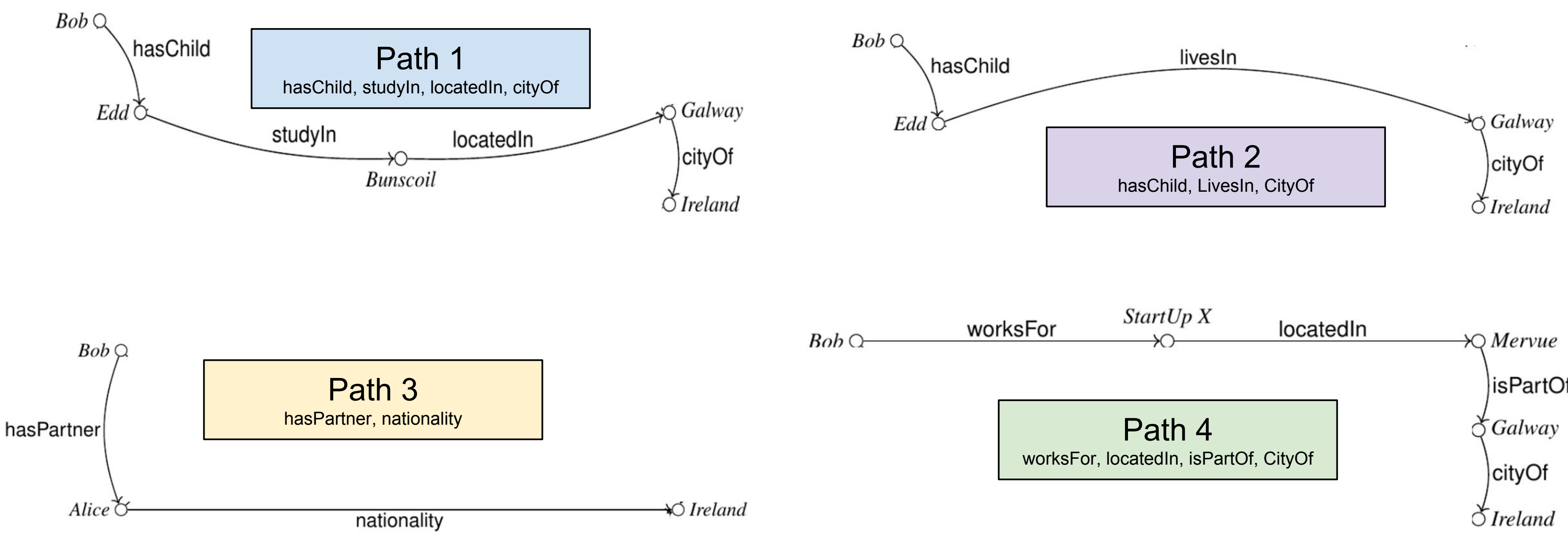


Figure 2: Sample of paths connecting Bob to Ireland from knowledge graph at Fig. 1

Knowledge Graph Paths

- Relation paths are an important feature, that can be used for:
  - Expressing properties of entities
  - Automatic inference of new facts
  - Learning new relation rules

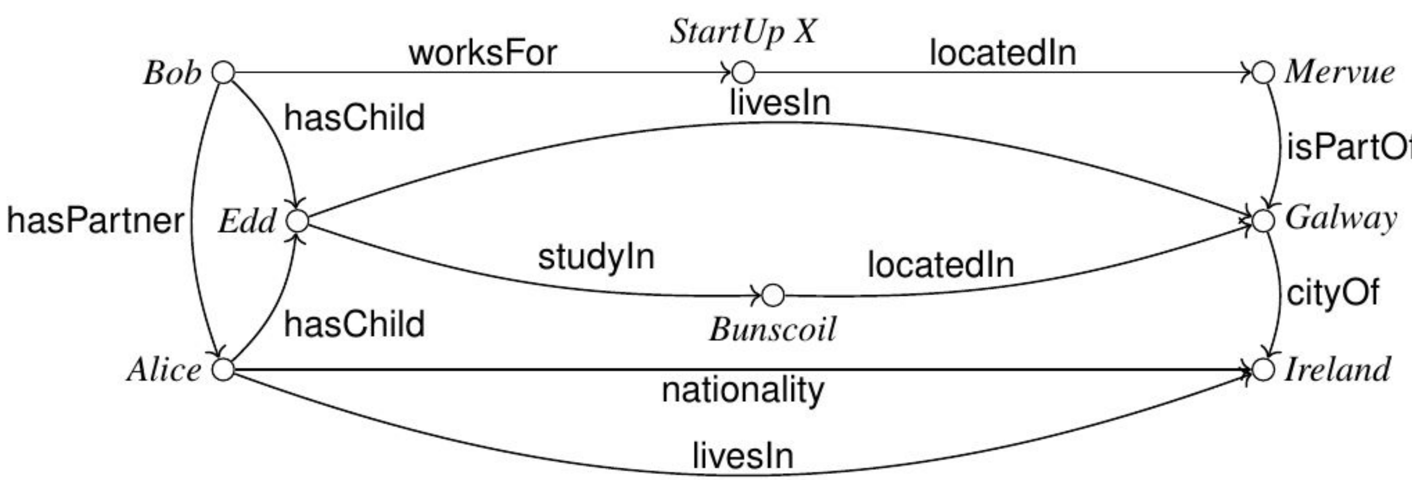


Figure 1: Example of a knowledge graph about people living in Galway, Ireland

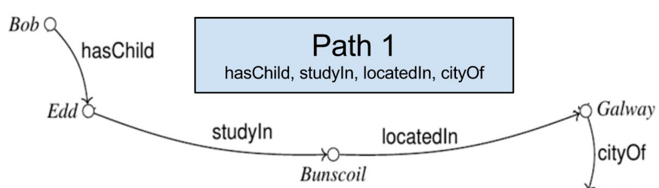
Equivalence of Relation Paths

Two paths are equivalent *if and only if* they have the same path extension.

$$P1 \equiv P2 \textit{ iff } \mathbf{PEXT}(P1) = \mathbf{PEXT}(P2)$$

where PEXT(P) is the set of node pairs that are connected with path P.

PEXT (



) =

(Bob , Ireland)
(e_1 , e_2)
(.... , ...)
(.... , ...)
(e_n, e_k)

Domain | Range

Path 1 have the following semantic:

"A person X who has child studying in a school in a city located in a country Y"

Finding Equivalence of a Relation Path

Search of equivalent relation paths problem
Given: a knowledge graph $\mathcal{G}$, relation path query Q, integer k, depth d **Find:** top-k equivalent relation paths of max. length $2d$ for Q in $\mathcal{G}$ according to a ranking function $Rank_Q(P)$, for all $P \in \mathcal{C}$ set of candidates.

Challenges:

- **Extracting path extension is a complex process**
- **Finding an equivalent relation path require trying combination of all possible path, which is a complex process.**
- **Knowledge incompleteness affect representation of paths in knowledge graphs, that equivalent paths can have different extension**

Proposed Solution:

- **A technique for finding approximate equivalences of a relation path using a sample of path extension instances**

Search Approximate Equivalences of a path (1)

We query approximate equivalences of a relation path using a 4 phases procedure:

1. Extracting sample of path extension
2. Extracting Subgraphs of extension instances
3. Build connecting between domain and range nodes of path extension
4. Ranking Candidate connecting paths

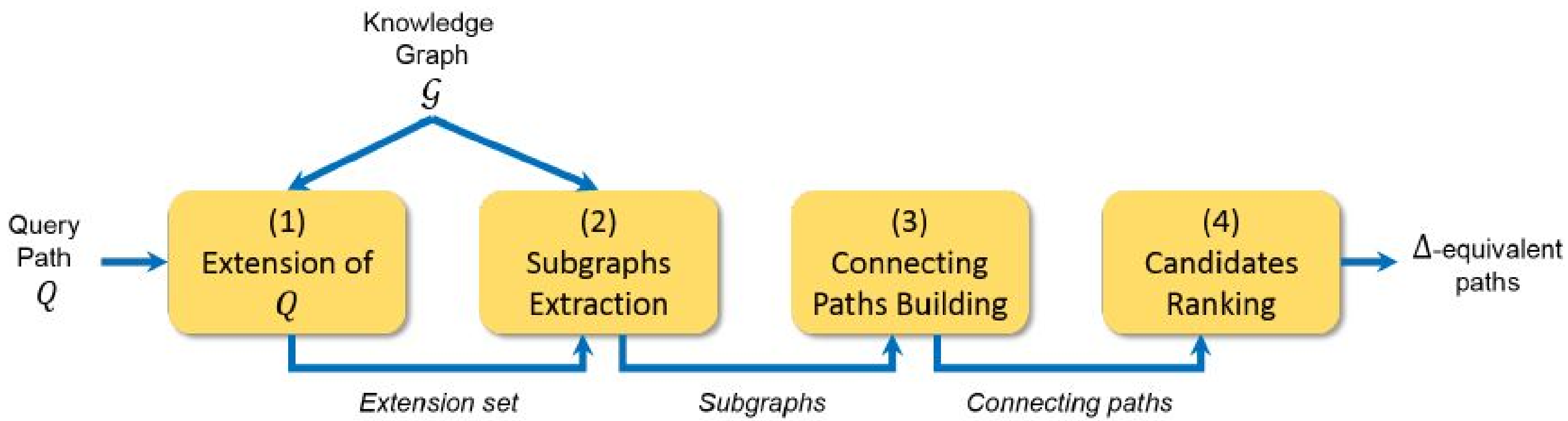
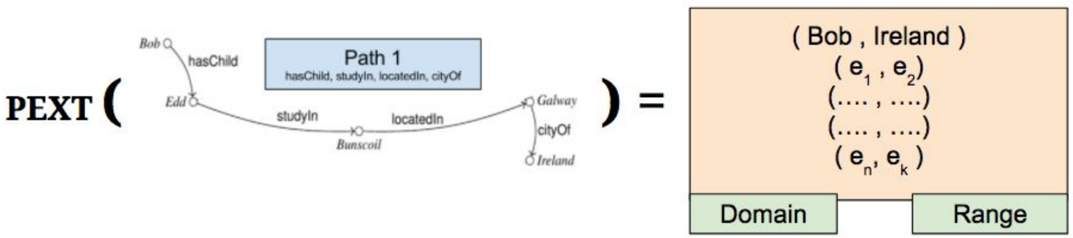
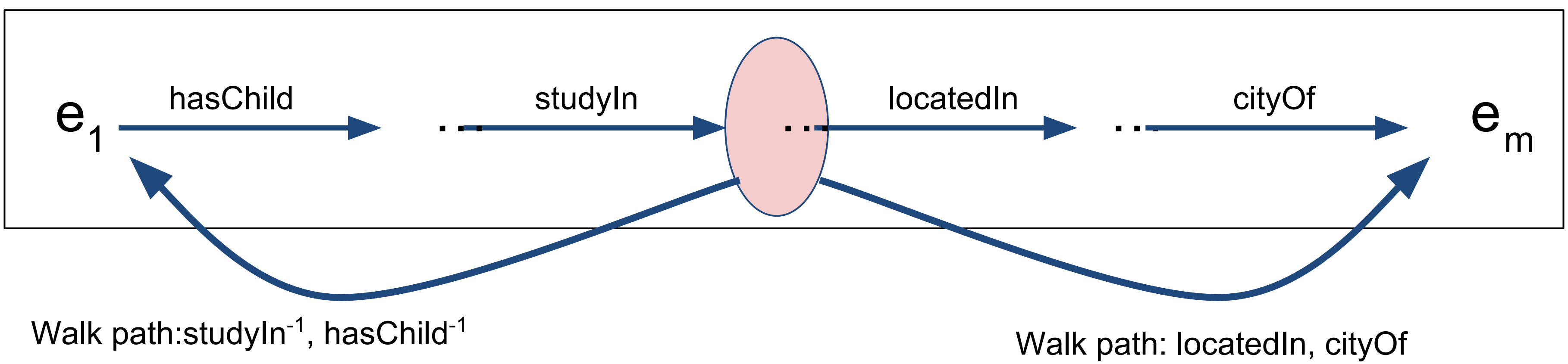


Figure 2: Flow diagram of the process of retrieving approximate equivalences of a query path "Q"

Search Approximate Equivalences of a path (2)

(1) **Extracting sample of path extension**
- **Finding middle path entities**
- **Walk from middle points to both domain and range nodes**
- **Combine reached domain and range nodes starting from same middle node.**





Search Approximate Equivalences of a path (3)

(2) Extracting subgraphs of path extension instances
- We use constrained Depth-First-Search (DFS)

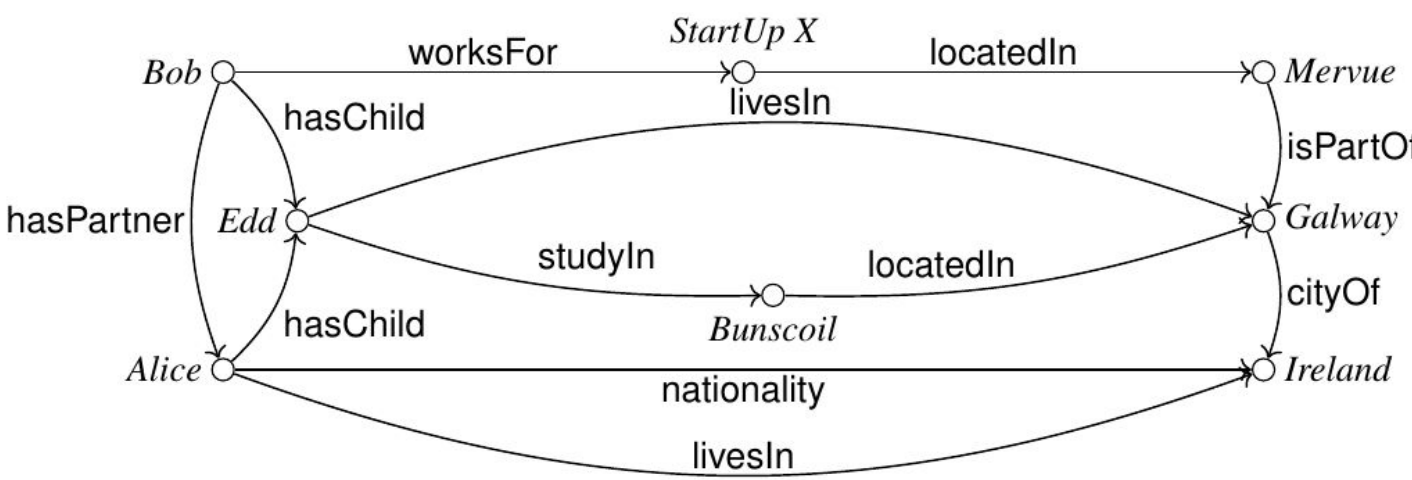


Figure 1: Example of a knowledge graph about people living in Galway, Ireland

Search Approximate Equivalences of a path (3)

(3) Build connecting paths between domain and range nodes

Algorithm 2 (ConnectingPaths: CONNECTING PATHS EXTRACTION)

Input: v_1, v_2 nodes, depth d, knowledge graph $\mathcal{G}$

Output: $\mathcal{C}$ list of connecting paths between v_1 and v_2

$\mathcal{G}_1, \mathcal{G}_2 \leftarrow \mathsf{Graph}^d(\mathcal{G}, v_1), \mathsf{Graph}^d(\mathcal{G}, v_2)$

$T_1, T_2 \leftarrow \{v \mid u, v \in V_{\mathcal{G}_1} \wedge \mathbf{I}_{\mathcal{G}_1}(P(u \rightsquigarrow v))\}, \{w \mid y, w \in V_{\mathcal{G}_2} \wedge \mathbf{I}_{\mathcal{G}_2}(P(y \rightsquigarrow w))\}$

for $t \in T_1 \cap T_2$ **do**

  for $P_1 \in \{P \mid u \in V_{\mathcal{G}_1} \wedge \mathbf{I}_{\mathcal{G}_1}(P(u \rightsquigarrow t))\}$ **do**

    for $P_2 \in \{P \mid v \in V_{\mathcal{G}_2} \wedge \mathbf{I}_{\mathcal{G}_2}(P(v \rightsquigarrow t))\}$ **do**

      $\mathcal{C}.append(P_1 \oplus \mathsf{Inverse}(P_2))$

return $\mathcal{C}$

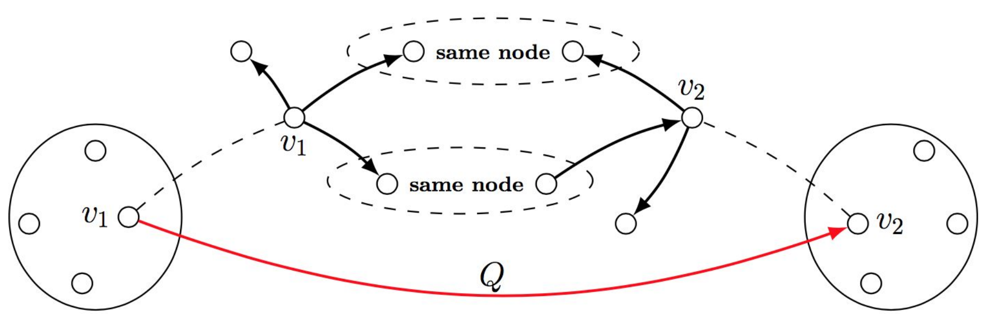


Fig. 3: Generation of connecting paths from the path extension of query Q.

Search Approximate Equivalences of a path (4)

(4) Ranking candidate relation paths

Rank extracted connecting paths using the following tri-criteria ranking function:

$$Rank_Q(P) = \alpha \underbrace{\frac{|\mathsf{PEXT}_{\mathcal{G}}(P)|}{|\mathsf{PEXT}_{\mathcal{G}}(Q)|}}_{\textsc{Cr-1}} + \beta \underbrace{\frac{\sigma(P)}{max\{\sigma(P_i) : P_i \in \mathcal{C}\}}}_{\textsc{Cr-2}} + \gamma \underbrace{\frac{|Q| - |P|}{max\{|Q|, |P|\}}}_{\textsc{Cr3}},$$

Similarity with Query Path	Frequency Among Candidate Paths	Path Length

where α , β , γ are configurable parameters.

Experimental setup

GOAL: Find ranked Δ-equivalences for a given query path

We experimented our approach on 4 datasets:

- NELL
- DBpedia
- YAGO3
- WordNet

Table 1: Statistics of knowledge graphs used in our experiments.

	NELL	DBpedia	YAGO3	WordNet
#Entities - $\lvert V \rvert$	1.2M	1.2M	2.6M	10K
#Relations - Σ_E	520	644	36	18
#Triples - $\lvert E \rvert$	3.8M	4M	5.5M	141K

Results (1)

- Queries from DBpedia dataset

Query	$Rank_Q(P)$	CR-1	CR-2	CR-3
Query B.1: ⟨wasBornIn,isLocatedIn⟩	Time ca. 118 min.		1000[b] instances	
⟨wasBornIn,isLocatedIn,isLocatedIn^{-1},isLocatedIn⟩	1.276	0.776	1.000	-0.500
⟨isCitizenOf⟩	1.025	0.016	0.001	0.500
⟨isPoliticianOf⟩	0.517	0.013	0.001	0.500
⟨livesIn⟩	0.514	0.007	0.000	0.500
⟨hasGender,hasGender^{-1},isPoliticianOf⟩	0.507	0.332	0.243	-0.333
Query B.2: ⟨actedIn,directed^{-1}⟩	Time ca. 193 min.		1000[b] instances	
⟨actedIn,isLocatedIn,isLocatedIn^{-1},directed^{-1}⟩	1.360	0.860	1.000	-0.500
⟨hasGender,hasGender^{-1}⟩	0.587	0.583	0.004	0.000
⟨actedIn,actedIn^{-1},actedIn,directed^{-1}⟩	0.524	0.674	0.350	-0.500
$\langle\epsilon\rangle^a$	0.518	0.018	0.000	0.500
⟨isMarriedTo⟩	0.503	0.003	0.000	0.500

Results (2)

- Queries from Yago dataset

Query	$Rank_Q(P)$	CR-1	CR-2	CR-3
Query C.1: ⟨artist,bandMember⟩	Time ca. 58 min.		1000[b] instances	
⟨artist,associatedMusicalArtist^{-1},associatedBand,bandMember⟩	1.435	0.935	1.000	-0.500
⟨artist,associatedBand^{-1},associatedMusicalArtist,bandMember⟩	1.429	0.935	0.994	-0.500
⟨artist,associatedBand^{-1},associatedMusicalArtist,associatedBand^{-1}⟩	0.953	0.524	0.929	-0.500
⟨artist,associatedMusicalArtist^{-1},associatedBand,associatedMusicalArtist^{-1}⟩	0.952	0.524	0.928	-0.500
⟨genre,instrument,instrument^{-1},genre^{-1}⟩	0.736	0.432	0.804	-0.500
Query C.2: ⟨academicAdvisor,almaMater⟩	Time ca. 80 min.		335 instances	
⟨academicAdvisor,birthPlace,birthPlace^{-1},almaMater⟩	1.080	0.580	1.000	-0.500
⟨academicAdvisor,deathPlace,birthPlace^{-1},almaMater⟩	0.846	0.575	0.771	-0.500
⟨almaMater⟩	0.641	0.121	0.020	0.500
⟨academicAdvisor,deathPlace,deathPlace^{-1},almaMater⟩	0.587	0.620	0.467	-0.500
⟨notableStudent^{-1},almaMater⟩	0.540	0.459	0.081	0.000

Results (3)

- Queries from NELL dataset

Query	$Rank_Q(P)$	CR-1	CR-2	CR-3
Query A.1: ⟨riverEmptiesIntoRiver,riverFlowsThroughCity⟩	Time ca. 53.2 min.		519 instances	
⟨cityLiesOnRiver^{-1},generalizations,generalizations^{-1}⟩	1.342	0.688	0.987	-0.333
⟨riverFlowsThroughCity,generalizations,generalizations^{-1}⟩	1.337	0.688	0.982	-0.333
⟨cityLiesOnRiver^{-1},generalizations,generalizations^{-1},generalizations^{-1}⟩	1.192	0.692	1.000	-0.500
⟨riverFlowsThroughCity,generalizations,generalizations^{-1},generalizations^{-1}⟩	1.189	0.692	0.997	-0.500
⟨riverEmptiesIntoRiver,cityLiesOnRiver^{-1}⟩	1.015	0.996	0.019	0.000
Query A.2: ⟨athletePlaysForTeam,teamHomeStadium,stadiumLocatedInCity⟩	Time ca. 35 min.		326 instances	
⟨athletePlaysForTeam,generalizations,generalizations^{-1},citySportsTeams^{-1}⟩	1.404	0.770	0.884	-0.250
⟨athletePlaysForTeam,generalizations,generalizations^{-1},teamPlaysInCity⟩	1.353	0.764	0.839	-0.250
⟨teamMember^{-1},generalizations,generalizations^{-1},citySportsTeams^{-1}⟩	1.264	0.739	0.775	-0.250
⟨athletePlaysForTeam,teamPlaysAgainstTeam^{-1},teamPlaysAgainstTeam^{-1}, citySportsTeams^{-1}⟩	1.249	0.531	0.969	-0.250
⟨teamMember^{-1},generalizations,generalizations^{-1},teamPlaysInCity⟩	1.244	0.733	0.761	-0.250

Conclusions

Our proposed technique for identifying equivalences of a relation path that achieves the following:

- Address the complexity of finding strict equivalences by using samples of path extensions

- Query and rank approximately equivalent paths depending on multiple ranking criteria.

Future work

1. Association rule mining by querying singular relation paths

2. Knowledge embedding to latent feature models using path similarities

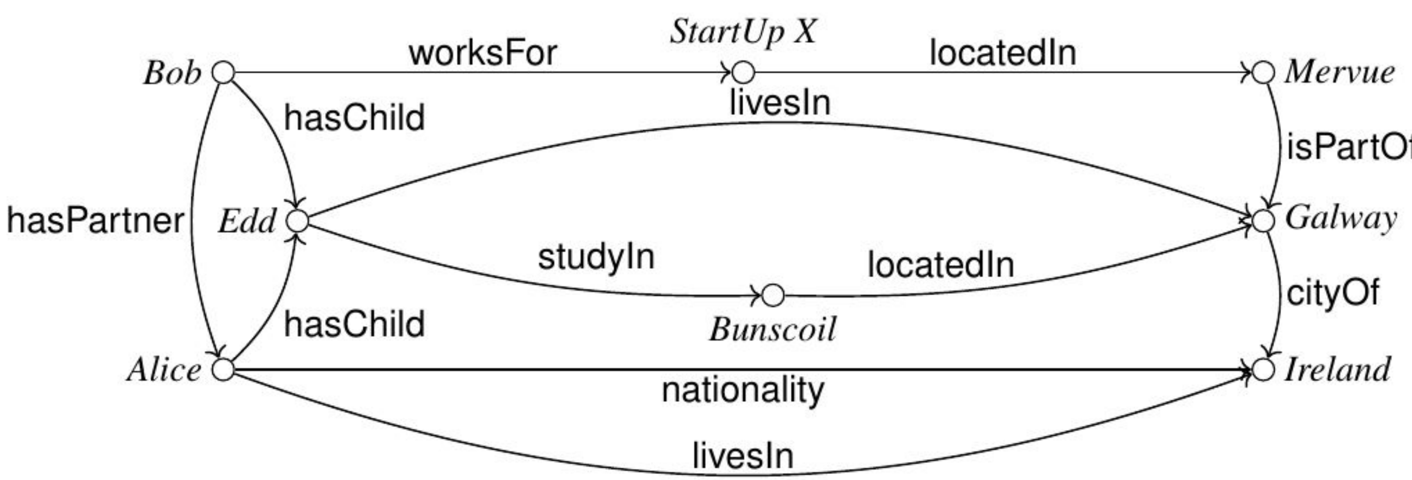


Figure 1: Example of a knowledge graph about people living in Galway, Ireland

Questions ?

Thank you