Link Prediction in Knowledge Graphs with Concepts of Nearest Neighbours

Sébastien Ferré
Team SemLIS, Univ Rennes, CNRS, IRISA

ESWC 2019
June 4th, Portorož, Slovenia

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

UMR IRISA SemLIS

Introduction

- Knowledge Graphs (KG) are widely used for
  - representation (RDF)
  - reasoning (RDFS, OWL)
  - querying (SPARQL)
- KGs are often incomplete
  and completing them manually is tedious
- Inductive inference by AI means is desirable and feasible
  1. *somebody born in Milano has probably Italian nationality*
  2. *somebody speaking Spanish was probably born in Spain or Latin America*

Introduction

- Knowledge Graphs (KG) are widely used for
  - representation (RDF)
  - reasoning (RDFS, OWL)
  - querying (SPARQL)
- KGs are often incomplete
  and completing them manually is tedious
- Inductive inference by AI means is desirable and feasible
  1. *somebody born in Milano has probably Italian nationality*
  2. *somebody speaking Spanish was probably born in Spain or Latin America*

Introduction

- Knowledge Graphs (KG) are widely used for
  - representation (RDF)
  - reasoning (RDFS, OWL)
  - querying (SPARQL)
- KGs are often incomplete
  and completing them manually is tedious
- Inductive inference by AI means is desirable and feasible
  1. *somebody born in Milano has probably Italian nationality*
  2. *somebody speaking Spanish was probably born in Spain or Latin America*

Link Prediction (aka. KG Completion)

- Definition: predict the missing head or tail of a triple
  - given a KG and an incomplete triple (Pablo, bornIn, ?),
  - predict Spain, Mexico, Peru, ...
- Challenges
  1. complex multi-relational data
  2. many relations to predict (e.g. bornIn)
  3. many possible values for each relation (e.g., all countries)
  4. multi-valued relations (e.g., actor from films to actors)
- Existing approaches
  - latent features: tensor factorization, graph embeddings, ...
    ex: *RESCAL, DistMult, TransE, HolE, ComplEx, R-GCN, ConvE*
    + state-of-the-art performance, – no explanations
  - observed features: paths, graph patterns, ...
    ex: *PRA (random walks), AMIE+ (association rules)*
    + (partial) explanations, – expressivity (mostly constant-free paths)
  - both: – costly learning phase

Link Prediction (aka. KG Completion)

- Definition: predict the missing head or tail of a triple
  - given a KG and an incomplete triple (Pablo, bornIn, ?),
  - predict Spain, Mexico, Peru, ...
- Challenges
  1. complex multi-relational data
  2. many relations to predict (e.g. bornIn)
  3. many possible values for each relation (e.g., all countries)
  4. multi-valued relations (e.g., actor from films to actors)
- Existing approaches
  - latent features: tensor factorization, graph embeddings, ...
    ex: *RESCAL, DistMult, TransE, HolE, ComplEx, R-GCN, ConvE*
    + state-of-the-art performance, **–** no explanations
  - observed features: paths, graph patterns, ...
    ex: *PRA (random walks), AMIE+ (association rules)*
    + (partial) explanations, **–** expressivity (mostly constant-free paths)
  - both: **–** costly learning phase

We propose a new kind of approach

Our approach to link prediction is inspired by k-NN classification

1. Load KG in memory (no training phase)
2. Given an incomplete triple $(e_i, r_k, ?)$
3. Find clusters of entities close to e_i (neighbours)
   - distance = shared graph pattern
   - cluster = Concept of Nearest Neighbours (CNN)
4. With each CNN, perform inferences for the missing entity e_j
5. Merge all CNN-wise inferences with Dempster-Shafer theory

Overview

1. Knowledge Graphs and Graph Patterns
2. Concepts of Nearest Neighbours (CNN)
3. CNN-based Link Prediction
4. Experimental Results
5. Conclusion and Perspectives

Knowledge Graph = Entities + Relations + Triples

female

male

Diana

Charles

Kate

William

Harry

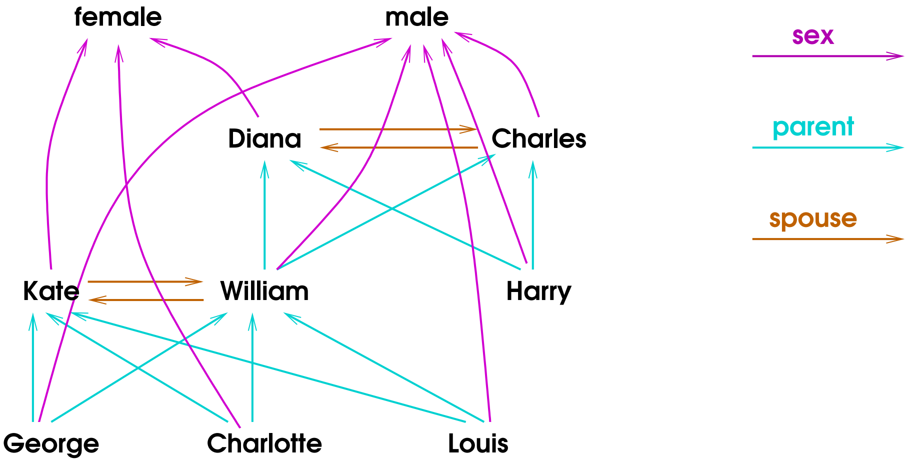
George

Charlotte

Louis

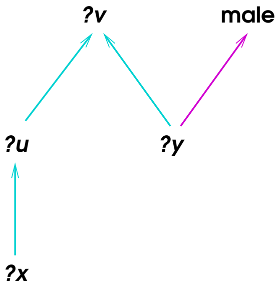
Knowledge Graph = Entities + Relations + Triples

Knowledge Graph = Entities + Relations + Triples



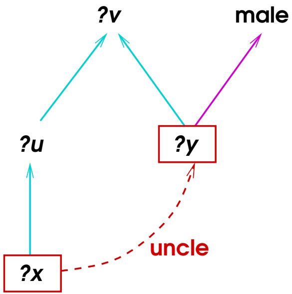
Graph Patterns, Queries, and Answers

Graph Pattern



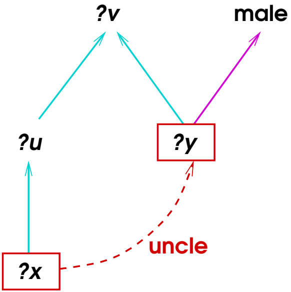
Graph Patterns, Queries, and Answers

Query



Graph Patterns, Queries, and Answers

Query



Answers (6)

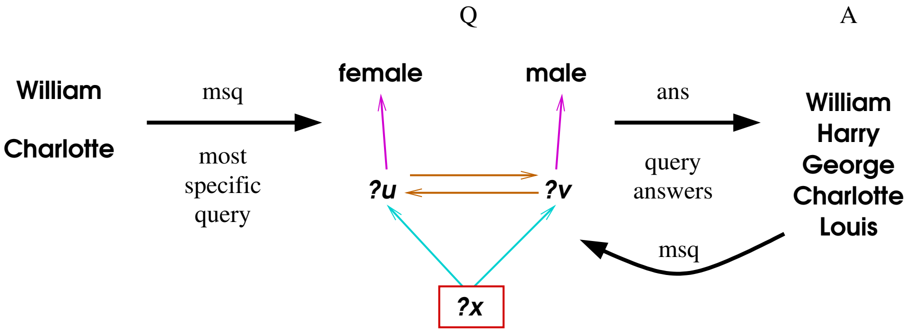
x	y
George	Harry
Charlotte	Harry
Louis	Harry
George	William
Charlotte	William
Louis	William

Overview

1. Knowledge Graphs and Graph Patterns
2. Concepts of Nearest Neighbours (CNN)
3. CNN-based Link Prediction
4. Experimental Results
5. Conclusion and Perspectives

Graph Concepts (akin to Formal Concept Analysis)

Starting from two entities:



A graph concept is a pair (A, Q), satisfying:

- $A = ans(Q)$: extension, set of concept instances
- $Q = msq(A)$: intension, concept description

Conceptual Distance

Definition

The conceptual distance between two entities e_i, e_j is defined as the most specific graph concept that contains them:
$\delta(e_i, e_j) = (A_{ij}, Q_{ij})$ where

- $Q_{ij} = msq(\{e_i, e_j\})$: what they have in common
- $A_{ij} = ans(Q_{ij})$: which entities range between them

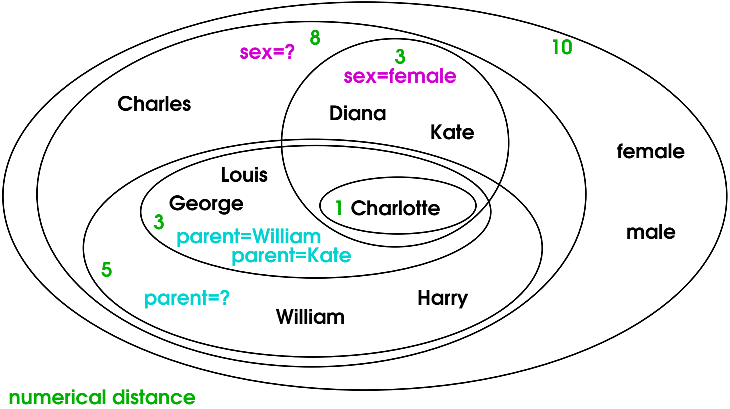
- δ is a symbolic distance
  - distances are partially ordered (concept inclusion)
- δ verifies distance axioms (positivity, symmetry, triangular ineq.)
  - with bottom concept as zero
  - with concept union as addition
- numerical measures can be derived
  - $dist(e_i, e_j) = |ext(\delta(e_i, e_j))|$: distance as number of answers
  - $sim(e_i, e_j) = |int(\delta(e_i, e_j))|$: similarity as size of the query

Concepts of Nearest Neighbours (CNN)

Given a knowledge graph $K = (E, R, T)$, and an entity $e \in E$:

$$CNN(e, K) = \{\delta(e, e') \mid e' \in E\}$$

Example for $e = Charlotte$ (6 CNNs)



Algorithmic and Practical Aspects

[see ESWC'18 paper on approximate query answering]

- $CNN(e, K)$ are computed by incrementally partitioning E
  - triples describing e are used as discriminating features
  - PRO: the number of clusters is bounded by $|E|$
- the partitioning algorithm is anytime
  - only coarser partition if stopped before completion
- previous experiments have shown greater efficiency compared to
  - computing conceptual distances with each entity
  - applying query relaxation to the description of e

Overview

1. Knowledge Graphs and Graph Patterns
2. Concepts of Nearest Neighbours (CNN)
3. CNN-based Link Prediction
4. Experimental Results
5. Conclusion and Perspectives

Inference for each CNN

Adaptation of k-NN classification:

- Let $(e_i, r_k, ?)$ an incomplete triple (missing tail e_j)
  e_i plays the role of the instance to be classified
  e_j plays the role of the target class
- For each CNN $\delta_l = (A_l, Q_l = x \leftarrow P_l(x)) \in CNN(e_i, K)$
  - A_l: neighbours, similar entities
  - $d_l = |A_l|$: distance
    the lower the distance, the better
- For each entity $e_j \in E$
  - $B_j = \{x \in E \mid (x, r_k, e_j) \in T\}$: instances of the target class
  - $\phi_{l,j} = \frac{|A_l \cap B_j|}{|A_l|}$: confidence of inference rule $P_l(x) \rightarrow (x, r_k, e_j)$
    the higher the confidence, the better
- A ranking of tail entities can be derived from distances and confidences

Merging Inferences with Dempster-Shafer Theory

How to merge inferences produced by each CNN?

- Adapting Denoeux's work on k-NN classification
  - based on Dempster-Shafer theory
  - each CNN is seen as an evidence for several tails e_j
  - assuming that tails are independent (1-N relations)
- Defining a Basic Belief Assignment for each CNN δ_l, and tail e_j
  - $m_{l,j}(\{e_j\}) = \alpha_0 \, \phi_{l,j} \, e^{-d_l}$ mass of evidence
- Merging belief for each tail e_j over all CNNs

$$Bel_j = Bel_j(\{e_j\}) = 1 - \prod_l (1 - m_{l,j}(\{e_j\}))$$

- Ranking all entities in $e_j \in E$ by decreasing belief Bel_j

Overview

1. Knowledge Graphs and Graph Patterns
2. Concepts of Nearest Neighbours (CNN)
3. CNN-based Link Prediction
4. Experimental Results
5. Conclusion and Perspectives

Examples of correct inferences

On two datasets, with timeout = 1s:

- On FB15k-237 (standard link prediction benchmark)
  - film "Dragon Ball Z: Bojack Unbound" $\xrightarrow{\text{has language}}$ Japanese
    26 CNNs, best explanation: *produced in Japan, and starring Toshiyuki Morikawa*
  - person "Tabu" $\xrightarrow{\text{lives in}}$ Mumbai
    32 CNNs, best explanation: *got "Filmware Award for Best Actress"*
- On (subset of) Mondial
  - mountain "Matterhorn" $\xrightarrow{\text{is located in}}$ Switzerland
    best explanation: *mountain in the Alps, also located in Italy (like "Monte Roza")*
  - "Lagen" $\xrightarrow{\text{is located in}}$ Norway
    best explanation: *it is the estuary of a river located in Norway*
    ⇒ suggests a generalization:
    IF a river is located in country X THEN its estuary is located in X

Examples of correct inferences

On two datasets, with timeout = 1s:

- On FB15k-237 (standard link prediction benchmark)
  - film "Dragon Ball Z: Bojack Unbound" $\xrightarrow{\text{has language}}$ Japanese
    26 CNNs, best explanation: *produced in Japan, and starring Toshiyuki Morikawa*
  - person "Tabu" $\xrightarrow{\text{lives in}}$ Mumbai
    32 CNNs, best explanation: *got "Filmware Award for Best Actress"*
- On (subset of) Mondial
  - mountain "Matterhorn" $\xrightarrow{\text{is located in}}$ Switzerland
    best explanation: *mountain in the Alps, also located in Italy (like "Monte Roza")*
  - "Lagen" $\xrightarrow{\text{is located in}}$ Norway
    best explanation: *it is the estuary of a river located in Norway*
    $\Rightarrow$ suggests a generalization:
    IF a river is located in country X THEN its estuary is located in X

Experimental Results

Comparison of ranking measures (MRR, Hits@{1,3,10}) between:

- *Freq*: decreasing value frequency (naive baseline)
- latent embeddings: TransE, DistMult, HolE, ComplEx, R-GCN
- association rules: AMIE+
- CNN: our approach with several timeouts

	FB15k-237 (300K triples)				Mondial (10K triples)			
Approche	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Freq	*.236*	*.175*	*.253*	*.356*	*.142*	*.069*	*.159*	*.309*
AMIE+	.143	.096	.155	.241	.179	.127	.208	.281
DistMult	.191	.106	.207	.376	-	-	-	-
ComplEx	.201	.112	.213	.388	-	-	-	-
HolE	.222	.133	.253	.391	-	-	-	-
TransE	.233	.147	.263	.398	-	-	-	-
R-GCN	.248	.153	.258	.414	-	-	-	-
ConvE	.325	.237	.356	.501	-	-	-	-
CNN 0.1s	.264	.198	.284	.395	.327	.271	.355	.433
CNN 1s	.286	.215	.311	.428	.320	.267	.344	.431

Overview

1 Knowledge Graphs and Graph Patterns

2 Concepts of Nearest Neighbours (CNN)

3 CNN-based Link Prediction

4 Experimental Results

5 Conclusion and Perspectives

Conclusion

We have proposed a symbolic approach to link prediction

- that is competitive with latent-based approaches
- that provides explanations for each inference
  - local explainability (not global explainability)
- that avoids the training phase (instance-based learning)
- that has a controllable runtime (anytime algorithm)
  - timeout is the only significant hyperparameter

In short

An adaptation of k-NN classification to knowledge graphs with conceptual distances.

Perspectives

- To extend graph patterns
  ex: inequalities; richer filters on numbers, strings, dates
- To optimize the computation of CNNs
  ex: partitioning strategies, parallelization
- To explore other kinds of inference
  ex: analogical inference, structured prediction, ...
- To evaluate on other datasets and tasks

The End

Thanks for listening !