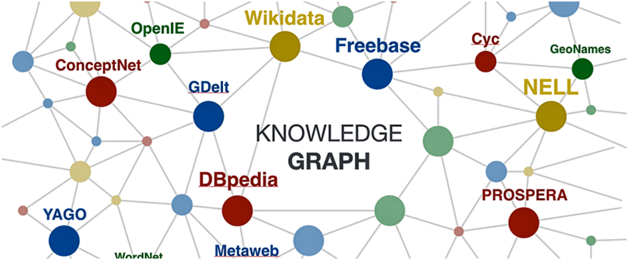
Machine Learning, Reasoning and Knowledge Graphs: a perspective on the usefulness of their interplay

Claudia d'Amato

Computer Science Department
University of Bari "Aldo Moro", Bari, Italy

Journées RoCED (Reasoning on Complex and Evolving Data)
6th July 2021



[1]

Open KG
online with content freely accessible

- BabelNet
- DBpedia
- Freebase
- Wikidata
- YAGO
- ....

Enterprise KG
for commercial usage

- Google
- Amazon
- Facebook
- LinkedIn
- Microsoft
- ....

[1] picture from https://www.csee.umbc.edu/courses/graduate/691/fall19/07/

Applications

- e-Commerce
- Semantic Search
- Fact Checking
- Personalization
- Recommendation
- Medical decision support system
- Question Answering
- Machine Translation
- ...

Research Areas

- Information Extraction
- Natural Language Processing
- Machine Learnig (ML)
- Knowledge Representation
- Web
- Robotics
- ...

Machine Learning & Knowledge Graphs

Two perspectives:

- KG as input to ML
  - **Goal:** improving the performance in many learning tasks, e.g. QA, image classification, instance disambiguation, etc.
- **ML as input to KG**
  - **Goal:** improving the KG itself
    - creating new facts
    - creating generalizations
    - prototyping
    - improving the size, coverage, depth and accuracy of KGs $\rightarrow$ reducing their production costs

What is a Knowledge Graph?

Knowledge Graph: Definition

[a] A graph of data intended to convey knowledge of the real world

- conforming to a graph-based data model
- nodes represent entities of interest
- edges represent potentially different relations between these entities
- data graph potentially enhanced with schema

[a] A. Hogan et al. Knowledge Graphs. arXiv:2003.02320v5 (2020)

KGs: Main Features

- grounded on the Open World Assumption (OWA)
- *ontologies* employed to define and reason about the semantics of nodes and edges
- very large data collections
- suffer of *incompleteness* and *noise*
  - since often result from a complex building process
- RDF, RDFS, OWL represetation languages will be assumed

ML as input to KG

Incompleteness and **noise**

$\Downarrow$

Knowledge Graph Refinement

- *Link Prediction*: predicts missing links between entities
  - regarded as a *learning to rank* problem
- *Triple Classification*: assesses correctness of a statement wrt a KG
  - regarded as a *binary classification* problem

Very Large Data Collections

$\Downarrow$

New scalable Machine Learning methods

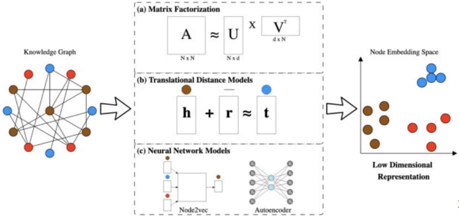
- grounded on *numeric-based approaches*
  - *vector embedding models* largely investigated[2]

Isseus:

- CWA (or LCWA) mostly adopted vs. OWA
- schema level information and reasoning capabilities almost disregarded
- no interpretable models $\Rightarrow$ hard to motivate results

[2] Cai, H. et al.: A comprehensive **survey** of graph embedding: problems, techniques, and applications. IEEE TKDE 30(09), pp. 1616-1637 (2018).

Numeric-based methods consist of series of numbers without any obvious human interpretation

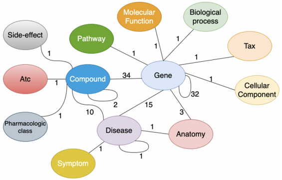


[3]

This may affects:

- the *interpretability* of the results
- the *explainability*
- and thus also somehow the *trustworthiness* of results

DRKG – Drug Repurposing Knowledge Graph



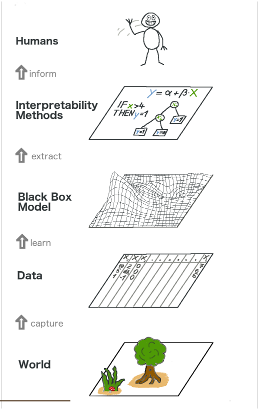
[4]

[3] Picture from D. N. Nicholson et al. Constructing knowledge graphs and their biomedical applications, Computational and Structural Biotechnology Journal, Vol. 18, pp. 1414–1428, (2020) ISSN 2001-0370

[4] Picture from `https://github.com/topics/knowledge-graph-embeddings`

Symbol-based learning methods usually provide

- *interpretable models* generalizing conclusions
    - e.g. trees, rules, logical formulae, etc.
- may be exploited for a better understanding of the provided results
- could be combined with deductive reasoning to make predictions



[5]

[5] Picture from `https://jaipancholi.com/model-interpretability`

Symbol-based learning methods:

- Can be still be applied to KGs? Why doing so?
- If so, is it possible to take into account reasoning capabilities?

Numeric-based learning methods:

- Can be enriched by taking into account schema level information and reasoning capabilities?
- If so, may it be beneficial?

Symbol-based learning methods:

- Can be still be applied to KGs? Why doing so?
- If so, is it possible to take into account reasoning capabilities?

Numeric-based learning methods:

- Can be enriched by taking into account schema level information and reasoning capabilities?
- If so, may it be beneficial?

Symbol-based learning methods for:

- Link Prediction (hits)
- Learning Disjointness Axioms
- Concept Learning

Symbol-based learning methods for:

- Link Prediction (hits)
- Learning Disjointness Axioms
- Concept Learning

Rule Mining for Link Prediction I

Basic Idea: exploit the evidence coming from the assertional data for *discovering hidden knowledge patterns* to be used for link prediction

$$Employee(x) \wedge worksAt(x,z) \wedge workForPrject(x,y) \wedge projectSupervisor(y,x) \Rightarrow isCompanyManagerOf(x,z)$$

- *body*: abstraction of assertions in KG co-occurring (w.r.t. a threshold)
- *head* represents a possibly new triple induced from KG and *body*

Rule Mining for Link Prediction II

Seminal works:

- Völker & Niepert @ ESWC'11; **Galárraga et al. @ WWW'13**
  - *highly scalable*
  - **no schema level information** and **reasoning capability** exploited
- d'Amato et al.@SAC'16, EKAW'16; Minh et al.@GECCO'17, RIVF'19
  - **schema level information** and **reasoning capability** exploited [6]
  - *redundant and inconsistent rules pruned*
  - limited ability to scale

[6] Implemented system publicly available at `https://github.com/tdminh2110/GeneratePatterns/` and `https://github.com/tdminh2110/CheckPatternMultiThreading/`

Symbol-based learning methods for:

- Link Prediction (hits)
- Learning Disjointness Axioms
- Concept Learning

A fine grained schema level information can bring better insight of the data

Disjointness axioms often missing

Problems:

- introduction of noise

$\mathcal{K} = \{ JournalPaper \sqsubseteq Paper,\ ConferencePaper \sqsubseteq Paper,\ ConferencePaper(a), Author(a)\ \}$
$\mathcal{K}$ is Consistent !!!
Cause Axiom: $Author \equiv \neg ConferencePaper$ **missing**

- counterintuitive inferences

$\mathcal{K} = \{ JournalPaper \sqsubseteq Paper,\ ConferencePaper \sqsubseteq Paper,\ ConferencePaper(a)\ \}$

$\mathcal{K} \models JournalPaper(a)$?
Answer: Unknown
Cause Axiom: $JournalPaper \equiv \neg ConferencePaper$ **missing**

- hard collecting negative examples when adopting numeric approaches

Observation: extensions of disjoint concepts do not overlap

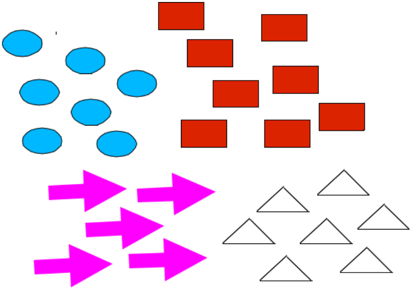
Question: would it be possible to *automatically capture* disjointness axioms by analyzing the data configuration/distribution?

Idea: Exploiting **(Conceptual) clustering methods** for the purpose

Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

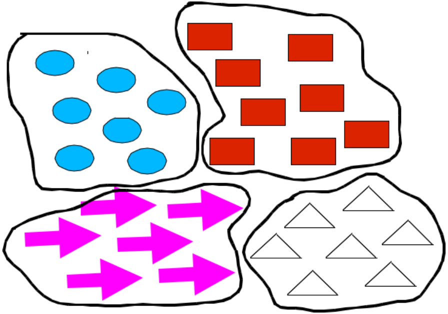
- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

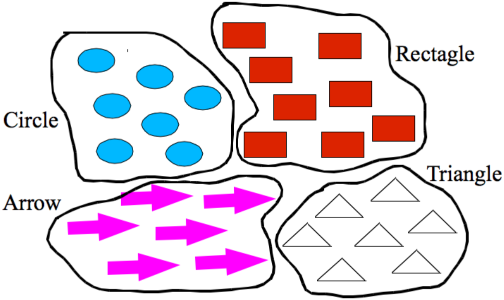
- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



Clustering Methods

Unsupervised inductive learning methods that organize a collection of unlabeled resources into meaningful clusters such that

- intra-cluster *similarity* is high
- inter-cluster *similarity* is low



Observation: extensions of disjoint concepts do not overlap

Question: would it be possible to *automatically capture* them by analyzing the data configuration/distribution?

Idea: Exploiting **(Conceptual) clustering methods** for the purpose

Definition (Problem Definition)

Given

- a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
- a set of individuals (aka entities) $\mathsf{I} \subseteq \mathsf{Ind}(\mathcal{A})$

Find

- n pairwise disjoint clusters $\{\mathsf{C}_1, \ldots, \mathsf{C}_n\}$
- for each $i = 1, \ldots, n$, a concept description D_i that describes C_i, such that:
  - $\forall a \in \mathsf{C}_i : \ \mathcal{K} \models D_i(a)$
  - $\forall b \in \mathsf{C}_j, j \neq i : \ \mathcal{K} \models \neg D_i(b)$.
- Hence $\forall D_i, D_j, i \neq j : \ \mathcal{K} \models D_j \sqsubseteq \neg D_i$.

Learning Disjointness Axioms: Developed Methods

Statistical-based approach

- NAR - exploiting negative association rules *[Fleischhacker et al. @ OTM'11]*
- PCC - exploiting Pearson's correlation coeff. *[Völker at al.@JWS 2015]*

do not exploit any background knowledge and reasoning capabilities

Disjointness axioms learning/discovery can be hardly performed without symbol-based methods

Terminological Cluster Tree

Defined a method [7] for eliciting disjointness axioms *[Rizzo et.al.@ SWJ'21]* [8]

- solving a clustering problem via learning Terminological Cluster Trees
- providing a concept description for each cluster

Definition (Terminological cluster tree (TCT))

A binary logical tree where

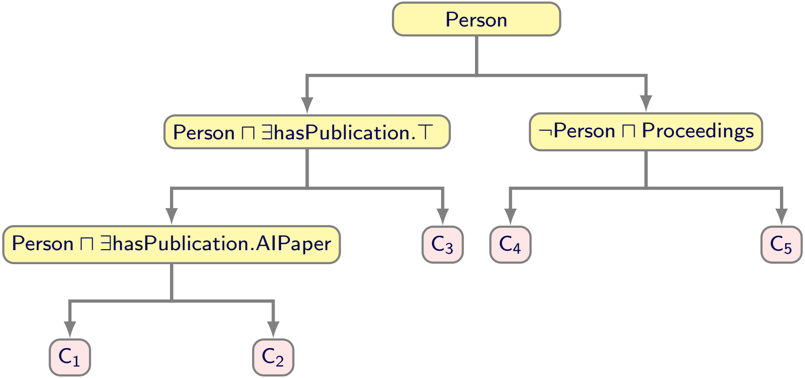
- a leaf node stands for a cluster of individuals C
- each inner node contains a description D (over the signature of $\mathcal{K}$)
- each departing edge corresponds to positive (left) and negative (right) examples of D

[7] Implemented system publicly available at https://github.com/Giuseppe-Rizzo/TCTnew

[8] G. Rizzo, C. d'Amato, N. Fanizzi: An unsupervised approach to disjointness learning based on terminological cluster trees. Semantic Web 12(3): 423-447 (2021)

Example of TCT

Given $\mathsf{I} \subseteq \mathsf{Ind}(\mathcal{A})$, an example of TCT describing the AI research community



Collecting Disjointness Axioms

Given a TCT T:

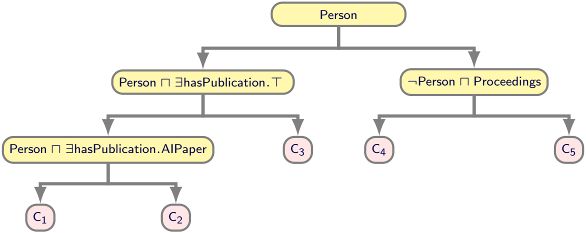
Step I:

- Traverse the T to collect the concept descriptions describing the clusters at the leaves
- A set of concepts CS is obtained

Step II:

- A set of candidate axioms A is generated from CS:
  - an axiom $D \sqsubseteq \neg E$ ($D, E \in$ CS) is generated if
    - $D \not\equiv E$ (or $D \not\sqsubseteq E$ or viceversa - *reasoner needed*)
    - $E \sqsubseteq \neg D$ has not been generated

Collecting Disjointness Axioms: Example



$$
\begin{aligned}
\mathrm{CS} = \{ \quad & \mathrm{Person}, \\
& \mathrm{Person} \sqcap \exists \mathrm{hasPublication}.\top, \\
& \neg(\mathrm{Person} \sqcap \exists \mathrm{hasPublication}.\top) \\
& \mathrm{Person} \sqcap \exists \mathrm{hasPublication}.\mathrm{AIPaper} \\
& \neg \mathrm{Person} \sqcap \mathrm{Proceedings} \cdots \}
\end{aligned}
$$

Axiom1: $Person \sqcap \exists hasPublication.AIPaper \sqsubseteq \neg(\neg Person \sqcap Proceedings)$

Axiom2: $\cdots$

Inducing a TCT

Given the set of individuals I and $\top$ concept

Divide-and-conquere approach adopted

- **Base Case:** test the STOPCONDITION
  - the cohesion of the cluster I exceeds a threshold ν
    - distance between *medoids* below a threshold ν
- **Recursive Step** (STOPCONDITION does not hold):
  - a set S of refinements of the current (parent) description C generated
  - the BESTCONCEPT $E^* \in$ S is selected and installed as *current node*
    - the one showing the *best cluster separation* $\Leftrightarrow$ with max distance between the *medoids* of its positive P and negative N individuals
  - I is SPLIT in:
    - $I_{left} \subseteq$ I $\leftrightarrow$ individuals with the smallest distance wrt the *medoid* of P
    - $I_{right} \subseteq$ I $\leftrightarrow$ individuals with the smallest distance wrt the *medoid* of N
    - *reasoner employed* for collecting P and N

Note: *Number of clusters not required* - obtained from data distribution

Lesson Learnt from experiments I

Experiments performed on ontologies publicly available

- Goal I: Re-discover a target axiom (existing in $\mathcal{K}$)
  - Setting:
    - A copy of each ontology is created removing a target axiom
    - Threshold $\nu = 0.9, 0.8, 0.7$
    - Metrics # discovered axioms and #cases of inconsistency
  - Results:
    - target axioms rediscovered for almost all cases
    - *additional* disjointness *axioms discovered* in a significant number
    - limited number of inconsistencies found

Ontology	TCT 0.9		TCT 0.8		TCT 0.7	
	#inc.	#ax's	#inc.	#ax's	#inc.	#ax's
BioPax	2	53	2	53	3	52
NTN	10	70	9	73	10	75
Financial	0	125	0	126	0	127
GeoSkills	2	345	1	347	4	347
Monetary	0	432	0	432	0	433
DBPedia3.9	45	45	44	44	43	43

Lesson Learnt from experiments II

Goal II:

- Re-discover randomly selected target axioms added according to the **Strong Disjointness Assumption** *[Schlobach et al. @ ESWC 2005]*
  - two sibling concepts in a subsumption hierarchy considered as disjoint
- comparative analysis with statistical-based methods *[Völker at al. @ JWS 2015, Fleischhacker et al. @ OTM'11]*
  - PCC - based on *Pearson's correlation coefficient*
  - NAR - exploiting *negative association rules*
- Setting:
  - A copy of each ontology created removing 20%, 50%, 70% of the disjointness axioms
    - The copy used to induce TCT - $\nu = 0.9, 0.8, 0.7$ - # Run: 10 times
  - **Metrics**: rate of **rediscovered** target axioms, #cases of inconsistency, # addional discovered axioms

Lesson Learnt from experiments III

- Results:
  - *almost all axioms rediscovered*
    - Rate decreases when larger fractions of axioms removed, *as expected*
  - *TCT outperforms PCC and NAR* wrt *additionally discovered axioms* whilst introducing limited inconsistency
    - TCT allows to express complex disjointness axioms
    - PCC and NAR tackle only disjointness between concept names

Exploiting the $\mathcal{K}$ as well as the data distribution improves **disjointness axioms discovery**

Example of axioms

Successfully discovered axioms

- $\text{ExternalReferenceUtilityClass} \sqcap \exists \text{TAXONREF}.\top$
  disjoint with
  xref
- Activity
  disjoint with
  $\text{Person} \sqcap \exists \text{nationality}.\text{United\_states}$
- $\text{Person} \sqcap \text{hasSex}.\text{Male}$ ($\equiv$ Man)
  disjoint with
  $\text{SupernaturalBeing} \sqcap \text{God}$ ($\equiv$ God)

Not discovered axioms

- Actor disjoint with Artefact

(concepts with few instances)

Symbol-based learning methods for:

- Link Prediction (hits)
- Learning Disjointness Axioms
- Concept Learning

Semantic and validating schemata require domain experts for definitions and constraints.

Latent patterns in the data graph could be exploited

Goal: **a)** Learning descriptions for a given concept name / expression

Example : $\mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male}$

b) Learning descriptions for characterizing a given set of individuals

Question: How to learn concept descriptions automatically, given a set of individuals?

Idea: Regard the problem as a *supervised concept learning* task

Supervised Concept Learning:

- Given a training set of positive and negative examples for a concept name,
- *construct* a *description* that will accurately classify whether future examples are positive or negative.

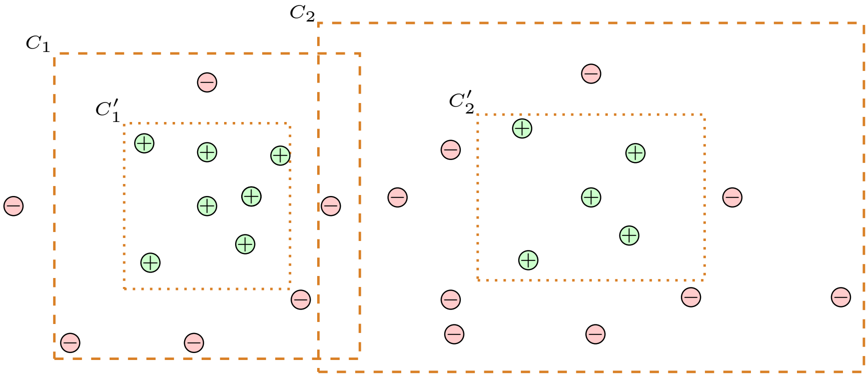
Definition (Problem Definition)

- *Given*
  - a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
  - a subset *pos* of individuals as positive examples of C
  - a subset *neg* of individuals as negative examples of C
- *Learn*
  - a DL concept description D so that
  - the individuals in *pos* are instances of D while those in *neg* are not

Developed Methods for Supervised Concept Learning

- **Separate-and-conquer approach**
  - YinYang *[Iannone et al. @ Appl. Intell. J. 2007]*
  - DL-FOIL *[Fanizzi et al. @ ILP 2008, Rizzo et al. @ FGCSJ 2020]*
  - DL-Learner *[Lehmann et al. @ MLJ 2010, SWJ 2011]*
- **Divide-and-conquer approach**
  - TermiTIS *[Fanizzi et al. @ ECML 2010, Rizzo et al. @ ESWC 2015, Rizzo et al. @ Aprox. Reas. J. 2018]*

DL-FOIL - Separate and Conquer: Example



C_1 = MasterStudent $\quad C'_1$ = MasterStudent $\sqcap \exists$worskIn.$\top$

C_2 = BachelorStudent $\quad C'_2$ = BachelorStudent $\sqcap \exists$worskIn.$\top$

On Evaluating the Learnt Concept Descriptions

- Publicly available ontologies considered
- A number (30) of satisfiable randomly generated concepts considered
- Positive and negative examples collected for each concept by using a deductive reasoner
- Running concept learning[9] on the collected positive and negative examples
- Inductive classification performed on the learnt concept descriptions

ontology	match rate	commission error rate	omission error rate	induction rate
BioPax	**76.9** ± 15.7	**19.7** ± 15.9	**7.0** ± 20.0	**7.5** ± 23.7
NTN	**78.0** ± 19.2	**16.1** ± 4.0	**6.4** ± 8.1	**14.0** ± 10.1
Financial	**75.5** ± 20.8	**16.1** ± 12.8	**4.5** ± 5.1	**3.7** ± 7.9

[9] Implemented system and datasets publicly available at https://bitbucket.org/grizzo001/dlfocl/src/master/

Examples of Learned Concept Descriptions with DL-FOIL

BioPax

induced:

```
Or( And( physicalEntity protein) dataSource)
```

original:

```
Or( And( And( dataSource externalReferenceUtilityClass)
ForAll(ORGANISM ForAll(CONTROLLED phys icalInteraction)))
protein)
```

NTN

induced:

```
Or( EvilSupernaturalBeing Not(God))
```

original:

```
Not(God)
```

Financial

induced:

```
Or( Not(Finished) NotPaidFinishedLoan Weekly)
```

original:

```
Or( LoanPayment Not(NoProblemsFinishedLoan))
```

Symbol-based learning methods:

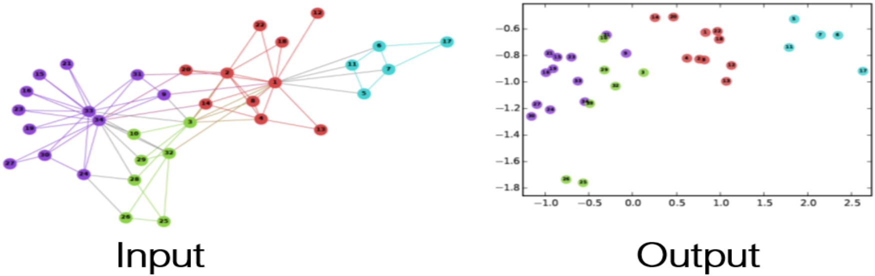
- Can be still be applied to KGs? Why doing so?
- If so, is it possible to take into account reasoning capabilities?

Numeric-based learning methods:

- Can be enriched by taking into account schema level information and reasoning capabilities?
- If so, may it be beneficial?

KG Embedding Models...

KGE models[10] convert data graph into an optimal low-dimensional space



[11]

Graph structural information and *properties* preserved as much as possible

[10] Cai, H. et al.: A comprehensive **survey** of graph embedding: problems, techniques, and applications. IEEE TKDE 30(09), pp. 1616-1637 (2018).

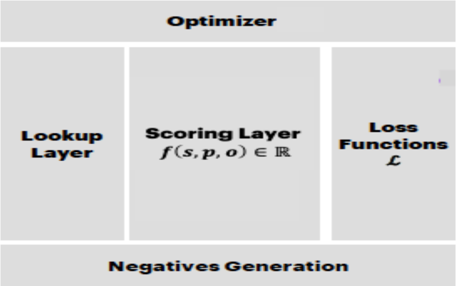
[11] Picture from `https://laptrinhx.com/node2vec-graph-embedding-method-2620064815/`

...KG Embedding Models

Goal

Learning embeddings s.t.

- score of a valid (positive) triple is higher than
- the score of an invalid (negative) triple



[12]

[12] Picture from "ECAI-20 Tutorial: Knowledge Graph Embeddings: From Theory to Practice"

Idea: Enhance KGE through Background Knowledge Injection

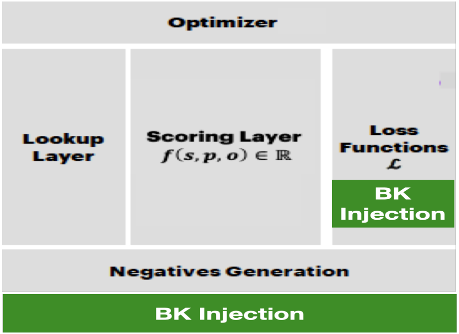
By two components:

Reasoning: used for generating negative triples

Axioms: `domain`, `range`, `disjointWith`, `functionalProperty`;

BK Injection: defines constraints on functions, corresponding to the considered axioms, *guiding the way embedding are learned*

Axioms: `equivClass`, `equivProperty`, `inverseOf` and `subClassOf`.



Other KG Embedding Methods Leveraging BK

- Jointly embedding KGs and logical rules *Guo, S. et al. @ ACL 2016*
  - triples represented as atomic formulae
  - rules represented as complex formulae modeled by t-norm fuzzy logics
- Adversarial training exploiting Datalog clauses encoding assumptions to regularize neural link predictors *[Minervini, P. et al. @ UAI 2017]*

A specific form of BK required, not directly applicable to KGs

An approach to learn embeddings exploiting BK

[d'Amato et al. @ ESWC 2021][13]

TRANSOWL

TransE

TRANSROWL TRANSROWL^R

TransR

Could be applied to more complex KG embedding methods
with additional formalization

[13] C. d'Amato, N. F. Quatraro, N. Fanizzi: Injecting Background Knowledge into Embedding Models for Predictive Tasks on Knowledge Graphs. ESWC 2021: 441-457 (2021)

TransOWL...

TransOWL maintains TransE setting

TransE[14] learns the vector embedding by minimizing

Margin-based loss function

$$L = \sum_{\substack{\langle s,p,o\rangle \in \Delta \\ \langle s',p,o'\rangle \in \Delta'}} \left[\gamma + f_p(\mathrm{e}_s, \mathrm{e}_o) - f_p(\mathrm{e}_{s'}, \mathrm{e}_{o'})\right]_+$$

where $[x]_+ = \max\{0, x\}$, and $\gamma \geq 0$

Score function

similarity (negative L_1 or L_2 distance) of the translated subject embedding $(\mathrm{e}_s + \mathrm{e}_p)$ to the object embedding e_o:

$$f_p(\mathrm{e}_s, \mathrm{e}_o) = -\|(\mathrm{e}_s + \mathrm{e}_p) - \mathrm{e}_o\|_{\{1,2\}}.$$

[14] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Proceedings of NIPS 2013 (2013)

...TransOWL

- Derive *further triples to be considered for training* via schema axioms
  - `equivClass`, `equivProperty`, `inverseOf` and `subClassOf`
- More complex loss function
  - adding a number of terms consistently with the constraints

$$
\begin{aligned}
L \; = & \overbrace{\sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r,t'\rangle \in \Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+}^{\textrm{TransE } \textit{loss function}} + \sum_{\substack{\langle t,q,h\rangle \in \Delta_{\mathsf{inverseOf}} \\ \langle t',q,h'\rangle \in \Delta'_{\mathsf{inverseOf}}}} [\gamma + f_q(t,h) - f_q(t',h')]_+ \\
& + \sum_{\substack{\langle h,s,t\rangle \in \Delta_{\mathsf{equivProperty}} \\ \langle h',s,t'\rangle \in \Delta'_{\mathsf{equivProperty}}}} [\gamma + f_s(h,t) - f_s(h',t')]_+ + \sum_{\substack{\langle h,\mathsf{typeOf},l\rangle \in \Delta \cup \in \Delta_{\mathsf{equivClass}} \\ \langle h',\mathsf{typeOf},l'\rangle \in \Delta' \cup \Delta'_{\mathsf{equivClass}}}} [\gamma + f_{\mathsf{typeOf}}(h,l) - f_{\mathsf{typeOf}}(h',l')]_+ \\
& + \sum_{\substack{\langle h,\mathsf{subClassOf},p\rangle \in \Delta_{\mathsf{subClass}} \\ \langle h',\mathsf{subClassOf},p'\rangle \in \Delta'_{\mathsf{subClass}}}} [(\gamma - \beta) + f(h,p) - f(h',p')]_+
\end{aligned}
$$

where $q \equiv r^-$, $s \equiv r$ (properties), $l \equiv t$ and $t \sqsubseteq p$ (classes) and $f(h,p) = \|\mathrm{e}_h - \mathrm{e}_p\|$

TransROWL...

TransROWL

- adopts the same approach of TransOWL
- *is derived from* TransR

TransE ⇒ poor modeling *reflexive* and *non* 1-to-1 relations (e.g. typeOf)
TransR[15] ⇒ more suitable to handle such specificity

TransR adopts TransE *loss function*

Score function
preliminarily projects e_s and e_o to the different d-dimensional space of the relational embeddings e_p through a suitable matrix $M \in \mathbb{R}^{k \times d}$:

$$f'_p(e_s, e_o) = -\|(Me_s + e_p) - Me_o\|_{\{1,2\}}.$$

[15] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI 2015 Proceedings. (2015)

...TransROWL

- TransOWL loss function adopted plus weighting parameters
  - equivClass, equivProperty, inverseOf and subClassOf
- TransR score function adopted

$$
\begin{aligned}
L = & \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r,t'\rangle \in \Delta'}} [\gamma + f'_r(h,t) - f'_r(h',t')]_+ + \lambda_1 \sum_{\substack{\langle t,q,h\rangle \in \Delta_{\text{inverseOf}} \\ \langle t',q,h'\rangle \in \Delta_{\text{inverseOf}'}}} [\gamma + f'_q(t,h) - f'_q(t',h')]_+ \\
& + \lambda_2 \sum_{\substack{\langle h,s,t\rangle \in \Delta_{\text{equivProperty}} \\ \langle h',s,t'\rangle \in \Delta_{\text{equivProperty}'}}} [\gamma + f'_s(h,t) - f'_s(h',t')]_+ + \lambda_3 \sum_{\substack{\langle h,\text{typeOf},l\rangle \in \Delta \cup \Delta_{\text{equivClass}} \\ \langle h',\text{typeOf},l'\rangle \in \Delta' \cup \Delta'_{\text{equivClass}}}} [\gamma + f'_{\text{typeOf}}(h,l) - f'_{\text{typeOf}}(h',l')]_+ \\
& + \lambda_4 \sum_{\substack{\langle t,\text{subClassOf},p\rangle \in \Delta_{\text{subClass}} \\ \langle t',\text{subClassOf},p'\rangle \in \Delta_{\text{subClass}'}}} [(\gamma - \beta) + f'(t,p) - f'(t',p')]_+
\end{aligned}
$$

where

- $q \equiv r^-$, $s \equiv r$ (properties), $l \equiv t$ and $t \sqsubseteq p$ (classes)
- the parameters λ_i, $i \in \{1, \ldots, 4\}$, weigh the influence that each function term has during the learning phase

TransROWL^R...

TransROWL^R adopts axiom-based regularization of *the loss function*, as for TransE^R [16]

- by adding specific constraints to the loss function rather than
- explicitly derive additional triples during training

TransE^R adopt TransE *score* and *loss function*
adds to the loss function *axiom-based regularizers* for inverse and equivalent property constraints

Loss function

$$L = \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ (h',r',t') \in \Delta'}} [\gamma + f_r(h,t) - f_r(h',t')]_+ + \lambda \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|r + q\| + \lambda \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|r - p\|$$

where $\mathcal{T}_{\text{inverseOf}}$ $\mathcal{T}_{\text{equivProp}}$ set of inverse properties and equivalent properties

[16] P. Minervini, L. Costabello, E. Muñoz, V. Novácek, P. Vandenbussche: Regularizing knowledge graph embeddings via equivalence and inversion axioms. ECML PKDD Proc. LNAI, vol. 10534, pp. 668–683 (2017)

...TransROWL^R

- TransR score function adopted
- *additional regularizers needed* for `equivalentClass` and `subClassOf` axioms
- *further constraints on the projection matrices* associated to relations

Loss function

$$
\begin{aligned}
L \;=\; & \sum_{\substack{\langle h,r,t\rangle \in \Delta \\ \langle h',r',t'\rangle \in \Delta'}} [\gamma + f'_r(h,t) - f'_r(h',t')]_+ \\
& + \lambda_1 \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|r + q\| \; + \lambda_2 \sum_{r \equiv q^- \in \mathcal{T}_{\text{inverseOf}}} \|M_r - M_q\| \\
& + \lambda_3 \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|r - p\| \; + \lambda_4 \sum_{r \equiv p \in \mathcal{T}_{\text{equivProp}}} \|M_r - M_p\| \\
& + \lambda_5 \sum_{e' \equiv e'' \in \mathcal{T}_{\text{equivClass}}} \|e' - e''\| \; + \lambda_6 \sum_{s' \subseteq s'' \in \mathcal{T}_{\text{subClass}}} \|1 - \beta - (s' - s'')\|
\end{aligned}
$$

Additional term for projection matrices required for `inverseOf` and `equivProp` triples to favor the equality of their projection matrices

Lesson Learnt from Experiments... I

Goal: Assessing the benefit of exploiting BK

- Comparing[17] TransOWL, TransROWL, TransROWL^R over to the original models TransE and TransR as a baseline

Perfomances tested on:

- Link Prediction task
- Triple Classification task

KGs adopted:

KG	*#Triples*	*#Entities*	*#Relationships*
DBpedia15K	180000	12800	278
DBpedia100K	600000	100000	321
DBpediaYAGO	290000	88000	316
NELL[18]	150000	68000	272

[17] All methods implemented as publicly available systems `https://github.com/Keehl-Mihael/TransROWL-HRS`

[18] equivalentClass and equivalentProperty missing; limited number of typeOf-triples; abundance of subClassOf-triples

...Lesson Learnt from Experiments I

- **Each dataset randomly partitioned** into *training* (70%), *validation* (10%) and *test* (20%) sets
- Learning rate: 0.001; minibatch dimension: 50; entity/relation vector dimension = 100; epochs: $\{250, 500, 1000\}$
- Both Filtered and Raw setting adopted
- TransROWL hyperparameters λ_i:
  - inverseOf $\lambda_1 = 1$; equivalentProperty $\lambda_2 = 1$; equivalentClass $\lambda_3 = 0.1$; subClassOf $\lambda_4 = 0.01$;
- TransROWL^R hyperparameters λ_i:
  - $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0.1$;

Link Prediction I

Measured the performance:

- considering *all properties but typeOf*
- *typeOf only* (focussing on *Type Prediction*)
- standard metrics adopted i.e. Mean Rank (MR), Hits@10 (H@10)

Type Prediction (typeOf only)

- Best performance achieved by TransROWL, in most of the cases, especially in terms of H@10
- TransOWL outperforms its baseline TransE only for the case *Type Prediction* (typeOf only)

Link Prediction II

Link Prediction other properties (no typeOf)

- TransROWL, TransROWL^R and TransR resulted more suitable for link prediction problems
  - TransROWL and TransROWL^R outperformed TransE and TransOWL, in most of the cases
- TransROWL, TransROWL^R outperfermed TransR most of the cases
  - when not (only in terms of MR), close runner-ups

As for NELL, the models showed lower performances wrt the baselines

- NELL was aimed at testing in condition of larger incompleteness
  - `equivalentClass` and `equivalentProperty` **missing**
  - low number of typeOf-triples per entity

Triple Classification I

Measured the performance:

- considering *all properties but typeOf* and *typeOf only*
- standard metrics: accuracy, precision, recall, false positive rate (FPR)

Results:

- Overall TRANSROWL and TRANSROWL^R achieve the best performance
  - with a few exceptions, particularly in terms of FPR
- TRANSROWL slightly superior performance of TRANSROWL^R
- TRANSOWL showed a general improvement over TRANSE,
  - especially in terms of FPR (for `typeOf` problems) and
  - in terms of accuracy and recall on two datasets (for no `typeOf`)
- NELL turned out to be more difficult for the models (oscillating performances)

Conclusions

Conclusions:

- Symbol-based learning methods necessary for supplementing schema level information
- Exploiting BK to learn embeddings models may improve link prediction and triple classification results
- Deductive reasoning essential for the full usage of BK

Further Research Directions:

- scalability of symbol-based learning methods to be improved
- more robust KB embedding solutions in case of KG incompleteness need to be developed (case of NELL)
- integrate further reasoning approaches (e.g. common sense reasoning, defeasible reasoning)

Thank you

Nicola Fanizzi   Giuseppe Rizzo   Nicola Flavio Quatraro