Reasoning in Knowledge Graphs

AIB 22

Ricardo Guimarães – University of Bergen
Ana Ozaki – University of Bergen

07.06.2022

Outline

Introduction

RDFS

Datalog

Description Logics

Rule Extraction

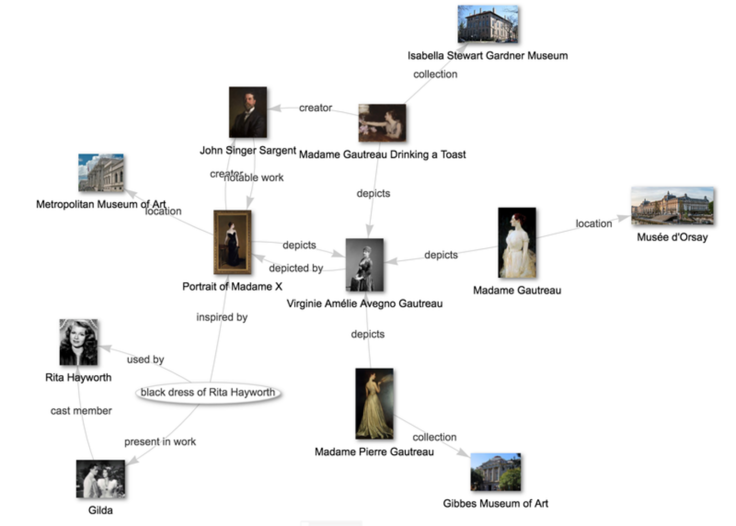
KG Embeddings

KGE and Rule Mining

Concluding Remarks

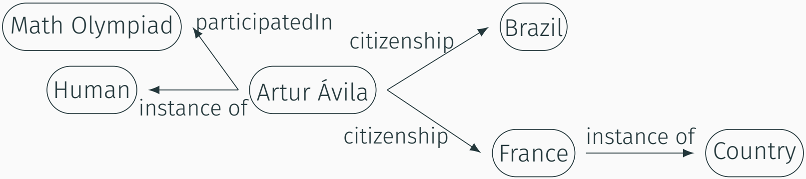
Introduction

Knowledge Graphs



Example Wikidata Query knowledge graph showing Portrait of Madame X
Fuzheado / CC BY-SA 4.0

KG Components



The Same KG in RDF

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <http://www.example.org/> .

# 'Artur Avila' 'is a' 'Human'
:ArturAvila rdf:type :Human .
# 'Artur Avila' 'has country of citizenship' 'France'
:ArturAvila :citizenship  :France .
# 'Artur Avila' 'has country of citizenship' 'Brazil'
:ArturAvila :citizenship :Brazil .
# 'France' 'is a' 'Country'
:France rdf:type :Country .
# 'Artur Avila' 'participated in' 'Math Olympiad'
:ArturAvila :participatedIn :MathOlympiad .
```

Reasoning

- Deriving new information from available data.

Reasoning

- Deriving new information from available data.
- Not necessarily correct or complete.

- Deriving new information from available data.
- Not necessarily correct or complete.
- Many different types and approaches.

Question!

What do you think when you you hear/read “Reasoning in KGs”?

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

Deductive Reasoning

- Apply known *rules* to derive knowledge.

$$\frac{\text{Socrates is Human} \qquad \text{Every Human is Mortal}}{\text{Socrates is Mortal}}$$

Deductive Reasoning

- Apply known *rules* to derive knowledge.
- Usual advantage: reliable and explainable.

$$\frac{\text{Socrates is Human} \qquad \text{Every Human is Mortal}}{\text{Socrates is Mortal}}$$

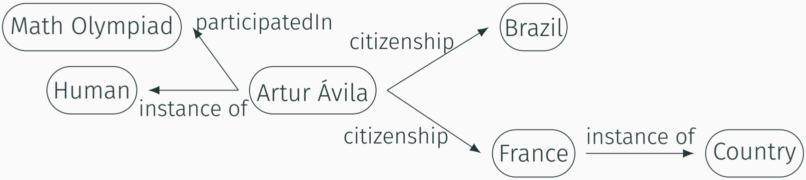
Deductive Reasoning

- Apply known *rules* to derive knowledge.
- Usual advantage: reliable and explainable.
- Usual disadvantage: flexibility and discovering the rules.

$$\frac{\text{Socrates is Human} \qquad \text{Every Human is Mortal}}{\text{Socrates is Mortal}}$$

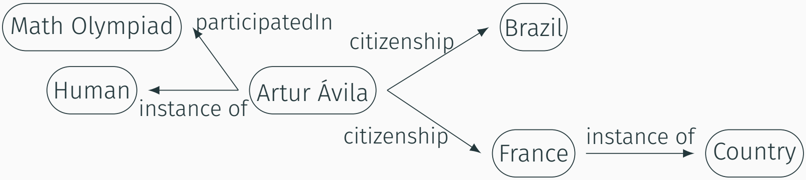
Deductive Reasoning: Example

- If x participated in y then x attended y



Deductive Reasoning: Example

- If x participated in y then x attended y



Conclusion: Artur Ávila attended the Math Olympiad

Inductive Reasoning

- Use patters in observations to derive knowledge.

$$\frac{\text{Duck 1 quacks} \quad \text{Duck 2 quacks} \quad \ldots \quad \text{Duck 1000 quacks}}{\text{Every duck quacks}}$$

Inductive Reasoning

- Use patters in observations to derive knowledge.
- Usual advantage: flexible and simpler to setup.

$$\frac{\begin{array}{r}\text{Duck 1 quacks}\\ \text{Duck 2 quacks}\\ \dots\\ \text{Duck 1000 quacks}\end{array}}{\text{Every duck quacks}}$$

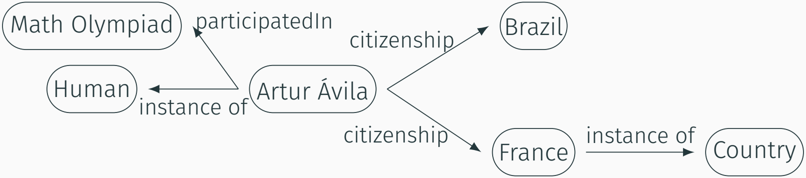
Inductive Reasoning

- Use patters in observations to derive knowledge.
- Usual advantage: flexible and simpler to setup.
- Usual disadvantage: may require many observations and interpretability.

$$\frac{\begin{array}{r}\text{Duck 1 quacks}\\ \text{Duck 2 quacks}\\ \dots\\ \text{Duck 1000 quacks}\end{array}}{\text{Every duck quacks}}$$

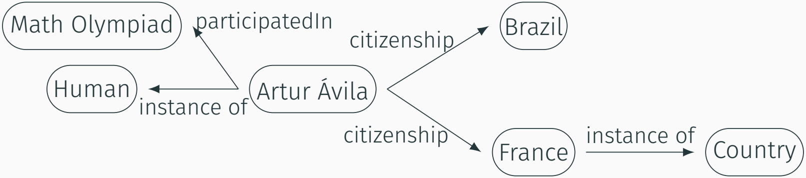
Deductive Reasoning: Example

- Suppose that for 99% of the triples $(\mathsf{x}, participatedIn, \mathsf{y})$, there is one $(\mathsf{x}, attended, \mathsf{y})$



Deductive Reasoning: Example

- Suppose that for 99% of the triples $(\mathsf{x}, participatedIn, \mathsf{y})$, there is one $(\mathsf{x}, attended, \mathsf{y})$



Conclusion: Artur Ávila attended the Math Olympiad (probably)

Abductive Reasoning

- Given the input and an output, find out the "reason".

$$\frac{\begin{array}{r}?\\ \text{Every Human is Mortal}\end{array}}{\text{Socrates is Mortal}}$$

Abductive Reasoning

- Given the input and an output, find out the “reason”.
- How to go from one point to another?

$$\frac{\begin{array}{r}?\\ \text{Every Human is Mortal}\end{array}}{\text{Socrates is Mortal}}$$

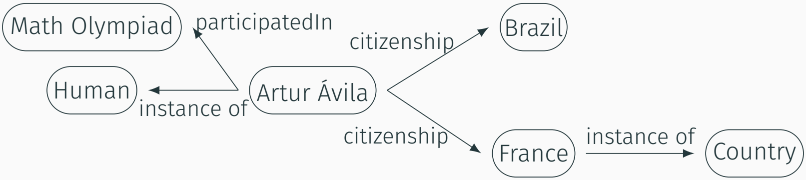
Abductive Reasoning

- Given the input and an output, find out the "reason".
- How to go from one point to another?
- Uses: explanations, repairs

$$\frac{\text{Socrates is Human} \qquad \text{Every Human is Mortal}}{\text{Socrates is Mortal}}$$

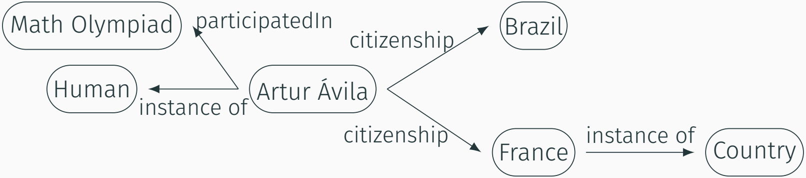
Abductive Reasoning: Example

- Suppose that, when using our KG, a system recognises Brazil as a country. What is a possible cause?



Abductive Reasoning: Example

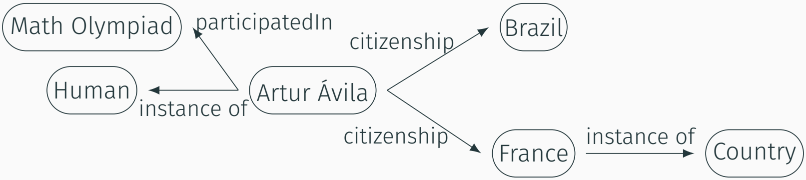
- Suppose that, when using our KG, a system recognises Brazil as a country. What is a possible cause?



One possibility, many ways: find out that the system interprets every “object” of $citizenship$ as a Country.

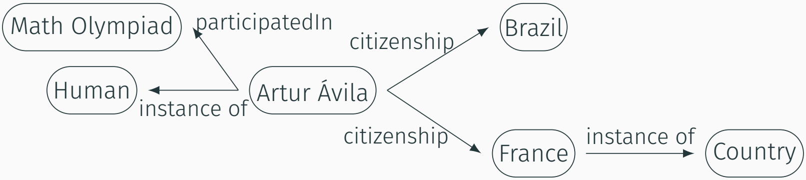
Abductive Reasoning: Example

- Suppose that, when using our KG, a system recognises Brazil is a country. What is a possible cause?



Abductive Reasoning: Example

- Suppose that, when using our KG, a system recognises Brazil is a country. What is a possible cause?



One possibility, many ways: find out that the system interprets every “object” of $citizenship$ as a Country.

Some Reasoning Approaches in KG

Deductive

- RDFS
- Description Logics
- Datalog

Inductive

- KG Embeddings
- Rule Mining
- Graph Neural Networks

RDFS

RDF: Express KGs as triples

RDF: Express KGs as triples

RDFS: Add a Schema to RDF Graphs

RDF: Express KGs as triples

RDFS: Add a Schema to RDF Graphs

- Properties with special semantics (meaning)

RDFS Entailment

RDF: Express KGs as triples

RDFS: Add a Schema to RDF Graphs

- Properties with special semantics (meaning)
- Use this semantics to derive new triples

Class hierarchy

```
:Mathematician rdfs:subClassOf :Human .
:Human rdfs:subClassOf :Mammal .
```

$\Rightarrow$

```
:Mathematician rdfs:subClassOf :Mammal .
```

Being an instance of subclass, implies being also an instance of the superclass

```
:Mathematician rdfs:subClassOf :Human .
:artur rdf:type :Mathematician .
```

Rule: rdfs9

Being an instance of subclass, implies being also an instance of the superclass

```
:Mathematician rdfs:subClassOf :Human .
:artur rdf:type :Mathematician .
```

$\Rightarrow$

```
:artur rdfs:type :Human .
```

Rule: rdfs5

Property hierarchy

```
:teachesAt rdfs:subPropertyOf :worksAt .
:worksAt rdfs:subPropertyOf :affiliatedWith .
```

Rule: rdfs5

Property hierarchy

```
:teachesAt rdfs:subPropertyOf :worksAt .
:worksAt rdfs:subPropertyOf :affiliatedWith .
```

$\Rightarrow$

```
:teachesAt rdfs:subPropertyOf :affiliatedWith .
```

Rule: rdfs7

Subproperty in the predicate, implies triple with superproperty

```
:teachesAt rdfs:subPropertyOf :worksAt .
:artur :teachesAt :universitat_zurich .
```

Rule: rdfs7

Subproperty in the predicate, implies triple with superproperty

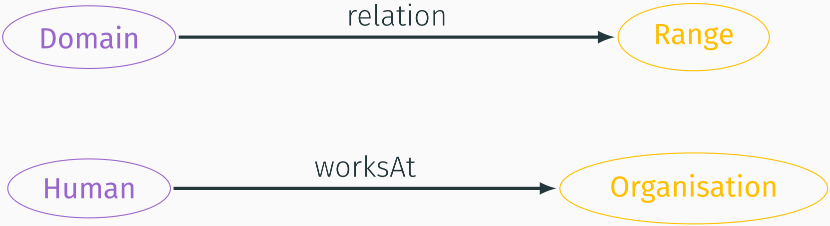
```
:teachesAt rdfs:subPropertyOf :worksAt .
:artur :teachesAt :universitat_zurich .
```

$\Rightarrow$

```
:artur :worksAt :universitat_zurich .
```

Domain and Range

Domain and Range



Rule: rdfs2

From the domain, we get the type

```
:worksAt rdfs:domain :Human .
:artur :worksAt :universitat_zurich .
```

$\Rightarrow$

```
:artur rdf:type :Human .
```

Rule: rdfs3

From the range, we get the type

```
:teachesAt rdfs:range :Organisation .
:artur :worksAt :universitat_zurich .
```

Rule: rdfs3

From the range, we get the type

```
:teachesAt rdfs:range :Organisation .
:artur :worksAt :universitat_zurich .
```

$\Rightarrow$

```
:universitat_zurich rdf:type :Organisation .
```

RDFS Entailment Example

```
:requires rdfs:domain :Food .
:hasIngredient rdfs:subPropertyOf :requires .
:Pasta rdfs:subClassOf :Food .
:lasagna rdf:type :Pasta .
:lasagna :hasIngredient :wheat .
:lasagna :hasIngredient :water .
:wheat rdf:type :Plant .
:hamburguer :hasIngredient :ground_meat .
```

Does the graph entail?

```
:lasagna rdf:type :Food
```

RDFS Entailment Example

```
1 :requires rdfs:domain :Food .
2 :hasIngredient rdfs:subPropertyOf :requires .
3 :Pasta rdfs:subClassOf :Food .
4 :lasagna rdf:type :Pasta .
5 :lasagna :hasIngredient :wheat .
6 :lasagna :hasIngredient :water .
7 :wheat rdf:type :Plant .
8 :hamburguer :hasIngredient :ground_meat .
```

Does the graph entail?

```
:lasagna rdf:type :Food
```

```
# Yes!
:Pasta rdfs:subClassOf :Food # (3)
:lasagne rdf:type :Pasta # (4)
:lasagne rdf:type :Food # (rdfs9) on (3, 4)
```

RDFS Question

```
:requires rdfs:domain :Food .
:hasIngredient rdfs:subPropertyOf :requires .
:Pasta rdfs:subClassOf :Food .
:lasagna rdf:type :Pasta .
:lasagna :hasIngredient :wheat .
:lasagna :hasIngredient :water .
:wheat rdf:type :Plant .
:hamburguer :hasIngredient :ground_meat .
```

Which of these triples the KG entails?

```
:hamburguer rdf:type :Food .
:wheat rdf:type :Food .
```

Answer Part 1

```
# Yes!
:hasIngredient rdfs:subPropertyOf :requires # line 2
:hamburguer :hasIngredient :ground_meat . # line 8
:hamburguer :requires :ground_meat . # [T1] (rdfs7) on (2, 8)
:requires rdfs:domain :Food . #  line 1
:hamburguer rdf:type :Food . # (rdfs2) on (1, [T1])
```

Answer Part 2

It is not possible to derive that. The information is not asserted and the only triple with with `wheat` uses the property `hasIngredient` which has no domain.

RDFS Take away

- Simple but effective
- There are other important details (check the specification)
- OWL and SHACL: more power

Question Time: RDFS Question

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

Datalog

Datalog

- Query language
- Deductive Databases
- Modern applications: Ontology-Based Data Access, KGs (Rule Engines)
- Very efficient

Datalog: Signature

Constants: $\mathrm{ArturAvila}, \mathrm{PaulErdos}$

Datalog: Signature

Constants: $\mathrm{ArturAvila}$, $\mathrm{PaulErdos}$

Variables: x, y, z

Datalog: Signature

Constants: $\text{ArturAvila}, \text{PaulErdos}$

Variables: x, y, z

Predicates: $Lecturer/1, WorksAt/2$

Datalog: Atoms and Facts

Atom: Something that can be true or false

Atom: Something that can be true or false

- $\mathit{Lecturer}(\mathrm{ArturAvila})$

Datalog: Atoms and Facts

Atom: Something that can be true or false

- $Lecturer(\mathrm{ArturAvila})$
- $WorksAt(\mathrm{ArturAvila}, x)$

Datalog: Atoms and Facts

Atom: Something that can be true or false

- $Lecturer(\text{ArturAvila})$
- $WorksAt(\text{ArturAvila}, x)$
- $SuppliesPartTo(\text{NVidia}, part, \text{Dell})$

Datalog: Atoms and Facts

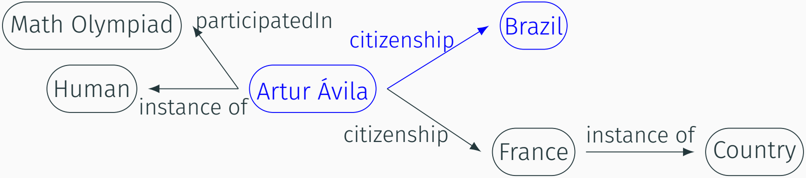
Atom: Something that can be true or false

- $Lecturer(\text{ArturAvila})$
- $WorksAt(\text{ArturAvila}, x)$
- $SuppliesPartTo(\text{NVidia}, part, \text{Dell})$

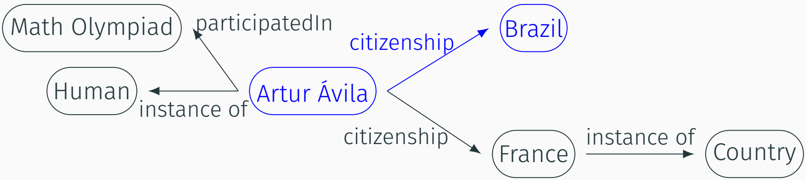
Facts are atoms without variables

- $Award(\text{FieldsMedal})$
- $CapitalOf(\text{Oslo}, \text{Norway})$

Triples as Facts



Triples as Facts



$$Citizenship(\mathrm{ArturAvila}, \mathrm{Brazil})$$

$$\underbrace{Lecturer(x)}_{Head} \leftarrow \overbrace{TeachesAt(x, y), University(x)}^{Body}$$

Datalog: Rules

$$\underbrace{Lecturer(x)}_{Head} \leftarrow \overbrace{TeachesAt(x,y), University(x)}^{Body}$$

- $\underbrace{Worker(x) \leftarrow Lecturer(x)}_{\text{Every lecturer is a worker}}$
- $\underbrace{CoAuthor(x,y) \leftarrow CoAuthor(y,x)}_{\text{coauthorship is symmetrical}}$

Datalog: Safe Rules

Safe Rule
A Datalog rule is safe if every variable in the head, appears in the body.

$+$ $Duck(x) \leftarrow Quacks(x), Swims(x), Flies(x)$

Datalog: Safe Rules

Safe Rule

A Datalog rule is safe if every variable in the head, appears in the body.

- $+$ $Duck(x) \leftarrow Quacks(x), Swims(x), Flies(x)$
- $+$ $Connected(x, z) \leftarrow Connected(x, y), Connected(y, z)$

Safe Rule
A Datalog rule is safe if every variable in the head, appears in the body.

- $+$ $Duck(x) \leftarrow Quacks(x), Swims(x), Flies(x)$
- $+$ $Connected(x, z) \leftarrow Connected(x, y), Connected(y, z)$
- $-$ $WorksAt(x, y) \leftarrow Worker(x)$

Safe Rule
A Datalog rule is safe if every variable in the head, appears in the body.

- $+$ $Duck(x) \leftarrow Quacks(x), Swims(x), Flies(x)$
- $+$ $Connected(x, z) \leftarrow Connected(x, y), Connected(y, z)$
- $-$ $WorksAt(x, y) \leftarrow Worker(x)$
- $-$ $SuppliesPartTo(s, p, c) \leftarrow Buys(c, p), Part(c)$

Question Time: Datalog

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

Datalog: Inference

1. Start from the facts
2. Apply rules to the facts
3. Repeat until no new facts are produced

Datalog: Inference Example – Part 1/2

Does Artur Ávila has an Erdős number?

$$HasErdosNumber(\text{PaulErdos}).$$

$$CoAuthor(\text{ArturAvila}, \text{BarrySimon}).$$

$$CoAuthor(\text{VilmosTotik}, \text{BarrySimon}).$$

$$CoAuthor(\text{VilmosTotik}, \text{PaulErdos}).$$

$$CoAuthor(\text{ArturAvila}, \text{WellingtonDeMelo}).$$

Datalog: Inference Example – Part 1/2

Does Artur Ávila has an Erdős number?

$$HasErdosNumber(\text{PaulErdos}).$$
$$CoAuthor(\text{ArturAvila}, \text{BarrySimon}).$$
$$CoAuthor(\text{VilmosTotik}, \text{BarrySimon}).$$
$$CoAuthor(\text{VilmosTotik}, \text{PaulErdos}).$$
$$CoAuthor(\text{ArturAvila}, \text{WellingtonDeMelo}).$$

$$HasErdosNumber(x) \leftarrow HasErdosNumber(y), CoAuthor(x, y). \quad (R1)$$
$$CoAuthor(x, y) \leftarrow CoAuthor(y, x). \quad (R2)$$

Datalog: Inference Example – Part 2/2

Does Artur Ávila has an Erdős number?

1. With R2 we add the facts that make $CoAuthor$ symmetric, e.g. $CoAuthor(\mathrm{PaulErdos}, \mathrm{VilmosTotik})$.

Datalog: Inference Example – Part 2/2

Does Artur Ávila has an Erdős number?

1. With R2 we add the facts that make $CoAuthor$ symmetric, e.g. $CoAuthor(\mathrm{PaulErdos}, \mathrm{VilmosTotik})$.
2. With R1: we infer that *Vilmos Totik* has an Erdős number.

Datalog: Inference Example – Part 2/2

Does Artur Ávila has an Erdős number?

1. With R2 we add the facts that make $CoAuthor$ symmetric, e.g. $CoAuthor(\mathrm{PaulErdos}, \mathrm{VilmosTotik})$.
2. With R1: we infer that *Vilmos Totik* has an Erdős number.
3. With R1: we infer that *Barry Simon* has an Erdős number (because *Vilmos Totik*) has one.

Datalog: Inference Example – Part 2/2

Does Artur Ávila has an Erdős number?

1. With R2 we add the facts that make $CoAuthor$ symmetric, e.g. $CoAuthor(\mathrm{PaulErdos}, \mathrm{VilmosTotik})$.
2. With R1: we infer that *Vilmos Totik* has an Erdős number.
3. With R1: we infer that *Barry Simon* has an Erdős number (because *Vilmos Totik*) has one.
4. With R1: we infer that *ArturAvila* has an Erdős number (because *Barry Simon*) has one.

Negation: $\neg Duck(x) \leftarrow Human(x)$

Existencial Quantification: $\exists WorksAt(x, y) \leftarrow Worker(x)$

Time: $Arrived(x, t + 1) \leftarrow InTransit(x, t)$

Datalog$^{\pm}$: Example [CGL12]

$$\begin{aligned} Human(x) &\leftarrow Scientist(x) \\ \perp &\leftarrow Scientist(x), Company(x) \\ \exists teachesAt(x, y), University(y) &\leftarrow Lecturer(x) \end{aligned}$$

Description Logics

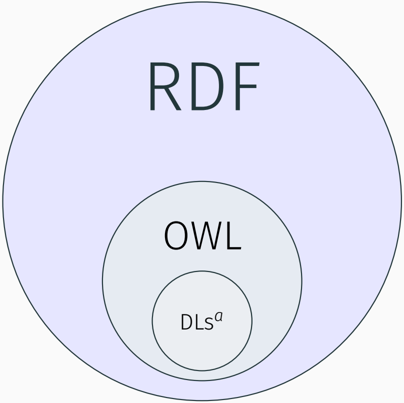
Description Logics (DLs)

- Family of logic-based formalisms
- Goal: Knowledge Representation
- Usually fragments of FOL
- Usually tailored to control the computational complexity of different reasoning problems

RDF: Describe data using graphs

OWL: Describe classes and properties using ontologies

DLs: Logical underpinning of OWL



[a]in general

In DLs the signature is composed by 3 pairwise disjoint sets:

Concept Names (NC): basic "classes" of elements ($Scientist$, $University$)

Role Names (NR): basic "relations" between elements ($worksAt$, $citizenship$)

Individual (NI): names for some of the individuals (artur, Brazil)

Signature

In DLs the signature is composed by 3 pairwise disjoint sets:

Concept Names (NC): basic "classes" of elements ($Scientist$, $University$)
Role Names (NR): basic "relations" between elements ($worksAt$, $citizenship$)
Individual (NI): names for some of the individuals (artur, Brazil)

In OWL, we have:

- Classes as concepts
- Properties as roles
- (Named) Individuals as individual names

$\mathcal{EL}^{\perp}$: A very simple DL

- Each DL offers different ways to combine concepts
- With concepts one can write axioms that work similar to rules or constraints
- $\mathcal{EL}^{\perp}$ is a very simple, efficient and important DL

$\mathcal{EL}^{\perp}$: Everything and Nothing

We can represent:

Everything:

```
owl:Thing
```

$\top$

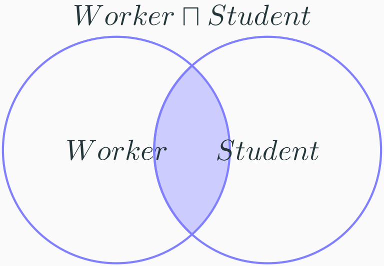
Nothing:

```
owl:Nothing
```

$\perp$

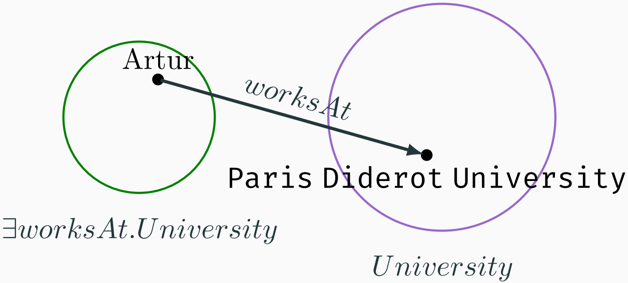
$\mathcal{EL}^{\perp}$: Conjunction

```
Worker and Student
```



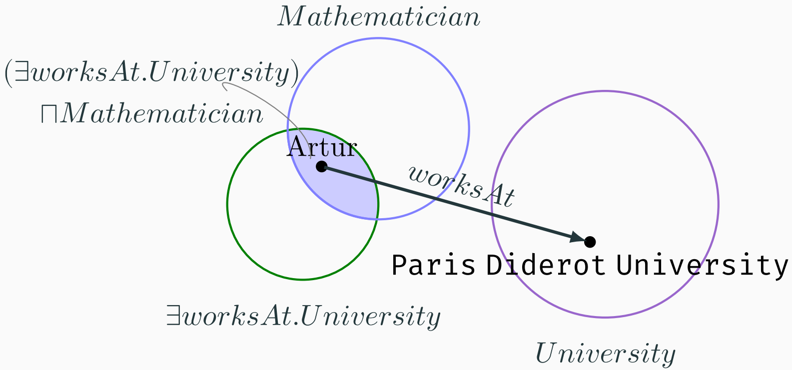
$\mathcal{EL}^{\perp}$: Existencial Restrictions

```
worksAt some University
```



$\mathcal{EL}^{\perp}$: Combining Constructors

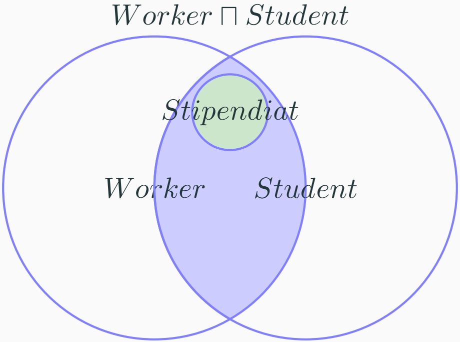
```
(worksAt some University) and Mathematician
```



$\mathcal{EL}^{\perp}$: Concept Inclusions

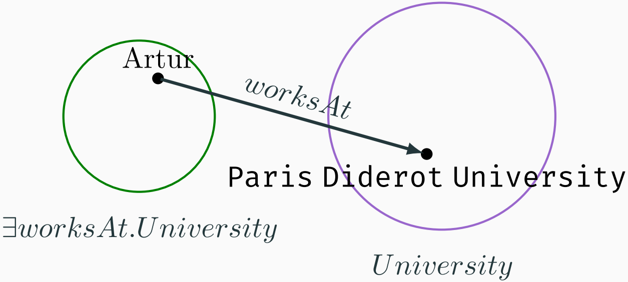
```
Class: Stipendiat
    SubClassOf: Worker and Student
```

$Stipendiat \sqsubseteq Worker \sqcap Student$



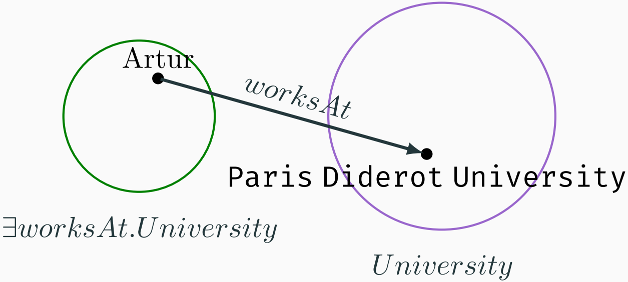
```
Individual: ParisDiderotUniversity
    Types: University
```

$University(\text{ParisDiderotUniv})$



```
Individual: Artur
    Facts: worksAt ParisDiderotUniv
```

$University(\mathrm{ParisDiderotUniv})$



Exercise: $\mathcal{EL}^\perp$

Question
Which properly describes “human parents must have a human child”?

1. $Parent \sqsubseteq Human \sqcap \exists hasChild.Human$
2. $Human \sqcap Parent \sqsubseteq Human \sqcap \exists hasChild.\top$
3. $Human \sqcap Parent \sqsubseteq \exists hasChild.Human$

Question Time: $\mathcal{EL}^{\perp}$

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

Reasoning Problems in DLs

Classification: Which concept names imply which?

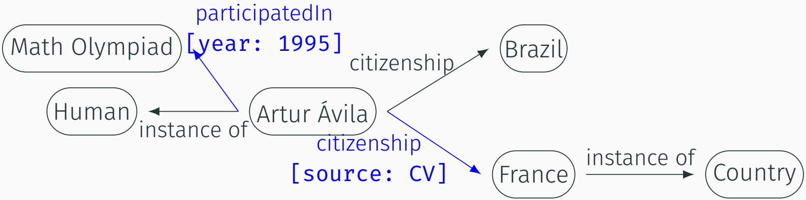
Concept Satisfiability: Is a concept name equivalent to $\perp$?

Instance Checking: Does an individual belong to a concept?

Inconsistency Checking: Does my ontology have an intepretation?

...

What About Annotations?



Attributed DLs

DLs tailored for KGs with annotations!

Attributed DLs

DLs tailored for KGs with annotations!

In attributed $\mathcal{EL}^\perp$, we can express the following:

- $participatedIn(\mathrm{ArturAvila}, \mathrm{MathOlympiad})@[year : 1995]$
  (Artur Avila participated in the Math Olympiad of 1995)

Attributed DLs

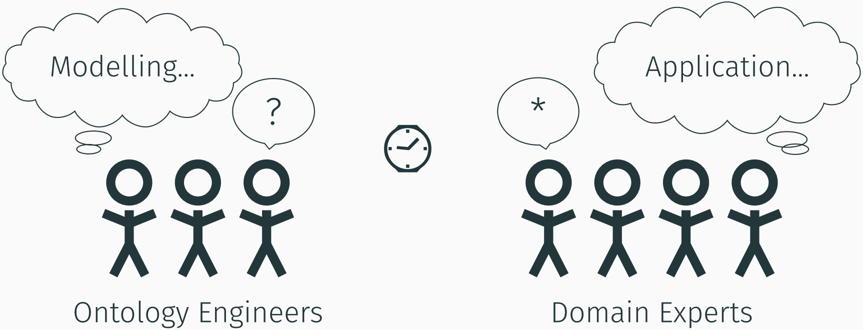
DLs tailored for KGs with annotations!

In attributed $\mathcal{EL}^{\bot}$, we can express the following:

- $participatedIn(\mathrm{ArturAvila}, \mathrm{MathOlympiad})@[year : 1995]$
  (Artur Avila participated in the Math Olympiad of 1995)
- $X : \lfloor reference : + \rfloor (PhD@[reference : X.reference] \sqsubseteq \exists educatedAt@\lfloor reference : X.reference \rfloor.University)$
  (Those with a PhD with a “source”, must have been educated at some university according to the same “source”)

Rule Extraction

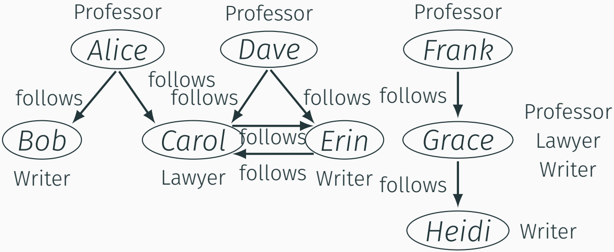
Building Ontologies: a Challenge



Mining Rules

Formal Concept Analysis (FCA) : theoretical guarantees

Association Rule Mining (ARM) : performance


Professor
Alice
Professor
Dave
Professor
Frank
follows
follows
follows
follows
follows
Professor
Bob
Carol
follows
Erin
Grace
Lawyer
Writer
Lawyer
follows
Writer
follows
Writer
Heidi
Writer

Attributes: A set of concept expressions (e.g. $\exists follows.Lawyer$)

CIs: An $\mathcal{EL}^{\perp}$ formula (e.g. $Professor \sqsubseteq \exists follows.Lawyer$)

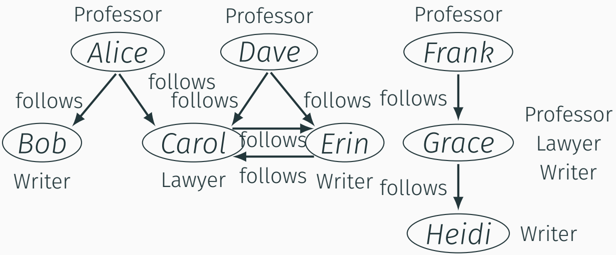
Base: A set of CIs (e.g.
$\{Professor \sqsubseteq \exists follows.Lawyer, Lawyer \sqcap \exists follows.\top, ... \}$).

Role Depth

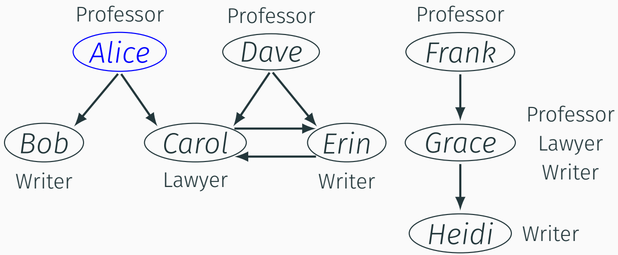
Maximum nesting of $\exists$:

Concept	Role Depth
$Person$	0
$Doctor \sqcap Professor$	0
$\exists worksAt.Hospital$	1
$(\exists worksAt.Hospital) \sqcap (\exists hasChild.\top)$	1
$\exists knows(Doctor \sqcap \exists hasChild.\top)$	2

Model-Based Most Specific Concepts

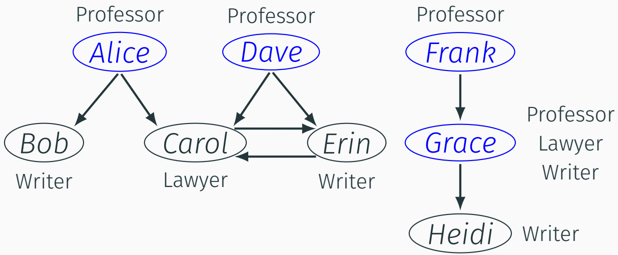


Model-Based Most Specific Concepts



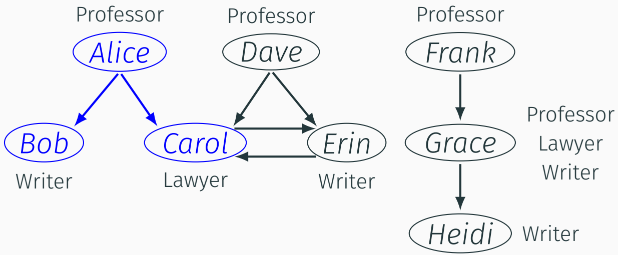
d	mmsc $(\{Alice\}, \mathcal{I}, d)$	mmsc $(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	

Model-Based Most Specific Concepts



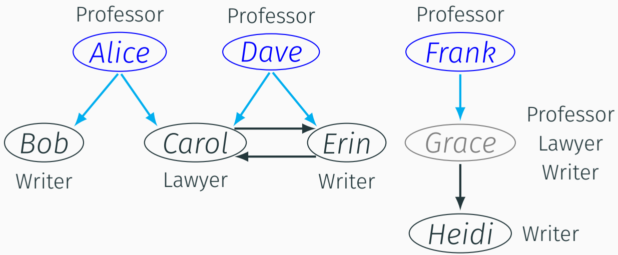
d	mmsc $(\{Alice\}, \mathcal{I}, d)$	mmsc $(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$

Model-Based Most Specific Concepts



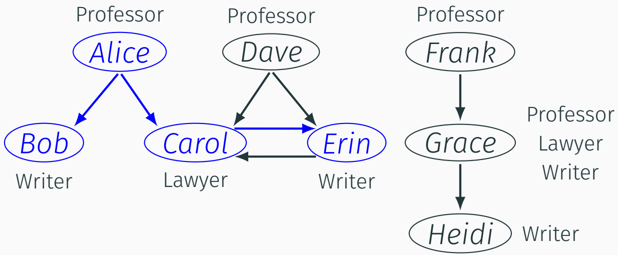
d	$\mathrm{mmsc}\,(\{Alice\}, \mathcal{I}, d)$	$\mathrm{mmsc}\,(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	

Model-Based Most Specific Concepts



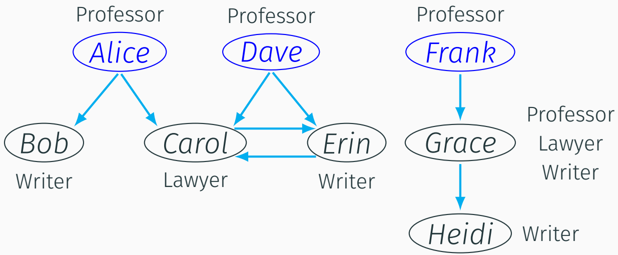
d	$\mathrm{mmsc}\,(\{Alice\}, \mathcal{I}, d)$	$\mathrm{mmsc}\,(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	$\{Alice, Dave, Frank\}$

Model-Based Most Specific Concepts



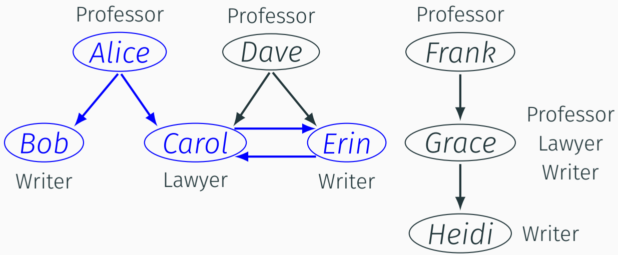
d	$\mathsf{mmsc}(\{Alice\}, \mathcal{I}, d)$	$\mathsf{mmsc}(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	$\{Alice, Dave, Frank\}$
2	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.W)$	

Model-Based Most Specific Concepts



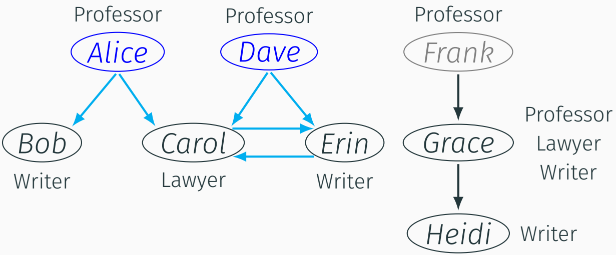
d	mmsc $(\{Alice\}, \mathcal{I}, d)$	mmsc $(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	$\{Alice, Dave, Frank\}$
2	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.W)$	$\{Alice, Dave, Frank\}$

Model-Based Most Specific Concepts



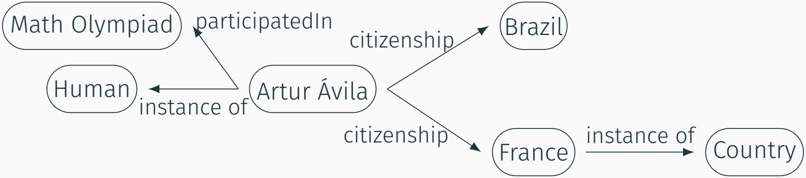
d	$\mathsf{mmsc}(\{Alice\}, \mathcal{I}, d)$	$\mathsf{mmsc}(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	$\{Alice, Dave, Frank\}$
2	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.W)$	$\{Alice, Dave, Frank\}$
3	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.(W \sqcap \exists f.L))$	

Model-Based Most Specific Concepts



d	mmsc $(\{Alice\}, \mathcal{I}, d)$	mmsc $(\{Alice\}, \mathcal{I}, d)^{\mathcal{I}}$
0	P	$\{Alice, Dave, Frank, Grace\}$
1	$P \sqcap \exists f.W \sqcap \exists f.L$	$\{Alice, Dave, Frank\}$
2	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.W)$	$\{Alice, Dave, Frank\}$
3	$P \sqcap \exists f.W \sqcap \exists f.(L \sqcap \exists f.(W \sqcap \exists f.L))$	$\{Alice, Dave\}$

Exercise: MMSC



What is the MMSC for “Artur Avila” with depth 2?

(A) $Human \sqcap \exists participatedIn.\top \sqcap \exists citizenship.Country$

(B) $\exists instanceOf.Human \sqcap \exists participatedIn.MathOlympiad \sqcap \exists citizenship.Brazil \exists citizenship.(France \sqcap \exists instanceOf.Country)$

Question Time: MMSC

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

The Attributes

The set of attributes $M_{\mathcal{J}}$ contains:

- $\bot$
- Person, Professor, Lawyer and Doctor
- $\exists r.\mathsf{mmsc}\,(X, \mathcal{J}, d)$ for every $X \subseteq \Delta^{\mathcal{J}}$ and role name r (e.g. $\exists follows.\mathsf{mmsc}\,(\{Alice, Bob\}, \mathcal{J}, d)$)

The Attributes

The set of attributes $M_{\mathcal{J}}$ contains:

- $\bot$
- Person, Professor, Lawyer and Doctor
- $\exists r.\text{mmsc}\,(X, \mathcal{J}, d)$ for every $X \subseteq \Delta^{\mathcal{J}}$ and role name r (e.g. $\exists follows.\text{mmsc}\,(\{Alice, Bob\}, \mathcal{J}, d)$)

Special concepts:

$$\Lambda_{\mathcal{J}} = \left\{ \bigsqcap U \mid U \subseteq M_{\mathcal{J}} \right\}$$

For instance, $Person \sqcap \exists knows.\text{mmsc}\,(\{Alice, Bob\}, \mathcal{J}, d) \in \Lambda_{\mathcal{J}}$

Assembling the Ontology

- Naïve idea: check for each pair of attributes C, D if $\mathcal{J} \models C \sqsubseteq D$
- Ideally obtain a non-redudant one
- Drawbacks of FCA strategy: many attributes and sensitivity to noise

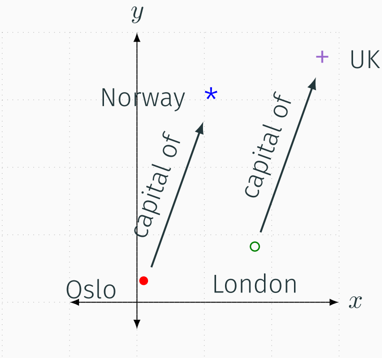
KG Embeddings

Representation Learning

- Subarea of Machine Learning
- Automatically build representations for data
- Reason with the represention
- Preserve and reveal patterns

Vector Space Models

- Represent things as "vectors"
- Space: $\mathbb{R}^n, \mathbb{C}^n$
- Word embeddings, graph embeddings, ...



Knowledge Graph Embeddings

- Map entities and relationships to a VSM
- Same space for entities and relations?
- Maintain the view of KG as set of triples
- One embedding for each entity or more?

Knowledge Graph Embeddings

- Map entities and relationships to a VSM
- Same space for entities and relations? It depends...
- Maintain the view of KG as set of triples
- One embedding for each entity or more? It depends...

KGE Design Goals

- Computational performance
- Full expressivity: can always find a way to separate true from false triples (given suitable data)
- Represent relation patterns (inverse, symmetry/anti-symmetry, composition)
- Sometimes we also want to express ontological constraints (e.g. concept and role hierarchies)

Notation for KGE

- Triple: (h, r, t) (or (s, p, o))

Notation for KGE

- Triple: (h, r, t) (or (s, p, o))
- Arbitrary embedding function: emb
- Embeddings: $\mathrm{emb}(h) = \mathbf{h}$, $\mathrm{emb}(t) = \mathbf{t}$, $\mathrm{emb}(r) = \mathbf{r}$

Notation for KGE

- Triple: (h, r, t) (or (s, p, o))
- Arbitrary embedding function: emb
- Embeddings: $\mathrm{emb}(h) = \mathbf{h}$, $\mathrm{emb}(t) = \mathbf{t}$, $\mathrm{emb}(r) = \mathbf{r}$
- Set of all entities: $\mathcal{E}$
- Set of all relations: $\mathcal{R}$

Notation for KGE

- Triple: (h, r, t) (or (s, p, o))
- Arbitrary embedding function: emb
- Embeddings: $\mathrm{emb}(h) = \mathbf{h}$, $\mathrm{emb}(t) = \mathbf{t}$, $\mathrm{emb}(r) = \mathbf{r}$
- Set of all entities: $\mathcal{E}$
- Set of all relations: $\mathcal{R}$
- Set of all triples in the KG: $\mathcal{K}$
- Set of all true triples: $\mathcal{W}$
- Set of all false triples: $\mathcal{W}^c$

Notation for KGE

- Triple: (h, r, t) (or (s, p, o))
- Arbitrary embedding function: emb
- Embeddings: $\mathrm{emb}(h) = \mathbf{h}$, $\mathrm{emb}(t) = \mathbf{t}$, $\mathrm{emb}(r) = \mathbf{r}$
- Set of all entities: $\mathcal{E}$
- Set of all relations: $\mathcal{R}$
- Set of all triples in the KG: $\mathcal{K}$
- Set of all true triples: $\mathcal{W}$
- Set of all false triples: $\mathcal{W}^c$
- Set of all (possible) entities with a legal embedding: $\mathcal{E}^*$

Basic Components of a KG Embedding Model

- Embedding space(s) ($\mathbb{S}_{\mathcal{E}}$, $\mathbb{S}_{\mathcal{R}}$)
- Energy/score function (emb)
- Loss function (loss)
- Optimisation algorithm
- False triple generation

Energy Functions

- Measure how much a triple "looks" correct
- Higher energy $\rightsquigarrow$ less likely to be true
- Many different types of energy function
- Ideally:

$$f(h,r,t)\begin{cases}\text{low} & \text{if } (h,r,t)\in\mathcal{W}\\ \text{high} & \text{if } (h,r,t)\notin\mathcal{W}\end{cases}$$

Which KGE models do you know?

Fact-Based: KGs as pure triples

Translation-based: TransE, TransD, RotatE

Tensor Factorisation: RESCAL, DistMult, ComplEx

Neural Networks: R-GCNs, ConvE, ConvKB

Description-Based: Triples with extra information

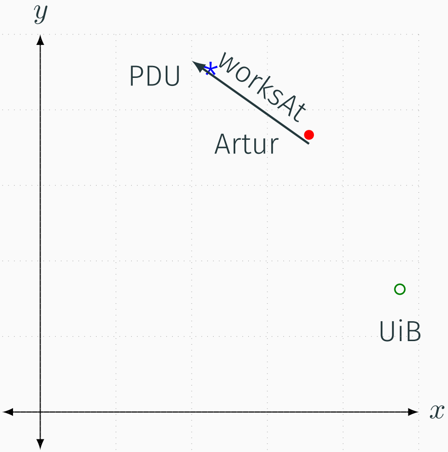
Text-based: TKRL

Path-based: PTransE

Others: temporal, provenance, types

Example of Energy Function: TransE [Bor+13]

$f_{\text{TransE}}(h, r, t) = ||(\mathbf{h} + \mathbf{r}) - \mathbf{t}||_\ell$[1]



[1] $||.||_\ell$ is either the L_1 or the L_2 norm

Example of Energy Function: TransE [Bor+13]

$f_{\text{TransE}}(h, r, t) = ||(\mathbf{h} + \mathbf{r}) - \mathbf{t}||_\ell$[1]

$$\mathbf{Artur} = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \mathbf{worksAt} = \begin{bmatrix} -0.31 \\ 0.22 \end{bmatrix} \quad \mathbf{PDU} = \begin{bmatrix} 0.45 \\ 0.89 \end{bmatrix}$$

$$\begin{aligned} f_{\text{TransE}}(Artur, worksAt, PDU) &= \left|\left| \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} + \begin{bmatrix} -0.31 \\ 0.22 \end{bmatrix} - \begin{bmatrix} 0.45 \\ 0.89 \end{bmatrix} \right|\right|_2 \\ &= \left|\left| \begin{bmatrix} 0.05 \\ 0.04 \end{bmatrix} \right|\right|_2 = 0.06 \end{aligned}$$

[1] $||.||_\ell$ is either the L_1 or the L_2 norm

Example of Energy Function: TransE [Bor+13]

$f_{\text{TransE}}(h, r, t) = ||(\mathbf{h} + \mathbf{r}) - \mathbf{t}||_{\ell}$[1]

$$\mathbf{Artur} = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \mathbf{worksAt} = \begin{bmatrix} -0.31 \\ 0.22 \end{bmatrix} \qquad \mathbf{UiB} = \begin{bmatrix} 0.95 \\ 0.32 \end{bmatrix}$$

$$\begin{aligned} f_{\text{TransE}}(Artur, worksAt, UiB) &= \left|\left| \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} + \begin{bmatrix} -0.31 \\ 0.22 \end{bmatrix} - \begin{bmatrix} 0.95 \\ 0.32 \end{bmatrix} \right|\right|_2 \\ &= \left|\left| \begin{bmatrix} -0.55 \\ 0.61 \end{bmatrix} \right|\right|_2 = 0.82 \end{aligned}$$

[1] $||.||_{\ell}$ is either the L_1 or the L_2 norm

Loss Function

- The actual function that will be optimised
- Idea: penalise low scores for true triples and high scores for false triples
- Many possible types: margin-based, cross-entropy and variants

Negative Examples with Incomplete Data

- Most KGs are incomplete
- Open World Assumption
- How do we sample $\mathcal{W}^c$?

Corrupting Triples

$$\begin{aligned}\text{corrupt}(h,r,t) = &\{(h,r,t') \mid t' \in \mathcal{E} \text{ and } (h,r,t') \notin \mathcal{K}\} \cup \\ &\{(h',r,t) \mid h' \in \mathcal{E} \text{ and } (h',r,t) \notin \mathcal{K}\}\end{aligned}$$

$$\mathcal{K}^c = \bigcup_{(h,r,t)\in\mathcal{K}} \text{corrupt}(h,r,t)$$

Corrupted Triples: Example

$$
\begin{aligned}
\mathcal{E} =& \{Alice, Bob, Bergen, UiB, Google\} \\
\mathcal{R} =& \{locatedIn, worksAt\} \\
\mathcal{K} =& \{(Alice, worksAt, UiB)\}
\end{aligned}
$$

Corrupted Triples: Example

$$
\begin{aligned}
\mathcal{E} =& \{Alice, Bob, Bergen, UiB, Google\} \\
\mathcal{R} =& \{locatedIn, worksAt\} \\
\mathcal{K} =& \{(Alice, worksAt, UiB)\}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^c =& \{(Alice, worksAt, Alice),\ (Alice, worksAt, Bob), \\
& (Alice, worksAt, Bergen), (Alice, worksAt, Google)\} \cup \\
& \{(Bob, worksAt, UiB),\ (Bergen, worksAt, UiB), \\
& (UiB, worksAt, UiB), (Google, worksAt, UiB)\}
\end{aligned}
$$

Example of Loss Function: TransE [Bor+13]

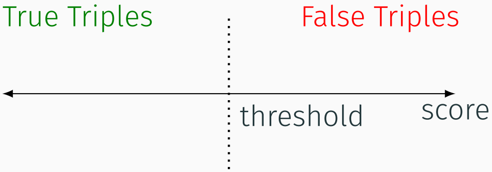
$$\text{loss} = \sum_{(h,r,t)\in\mathcal{K}} \sum_{(h',r',t')\in\mathcal{K}^c} \max(\gamma + f(h',r',t') - f(h,r,t), 0)$$

Full Expressivity [FRP19]

A model is fully expressive if given any assignment of truth values to all triples, there exists an assignment of values to the embeddings of the entities and relations that accurately separates $\mathcal{W}$ from $\mathcal{W}^c$ using model's score function.

Full Expressivity: Illustration

You can always find a threshold such that:



TransE: Expressivity Failure

How would you prove that TransE is not fully expressive?

TransE: Expressivity Failure

How would you prove that TransE is not fully expressive?

A counter-example is enough! TransE does not handle symmetric relations well:

TransE: Expressivity Failure

How would you prove that TransE is not fully expressive?

A counter-example is enough! TransE does not handle symmetric relations well:

$$T^{+} = \{(Alice, friends, Bob), (Bob, friends, Alice)\}$$
$$T^{-} = \{(Alice, friends, Alice), (Bob, friends, Bob)\}$$

$$\mathbf{friends} \rightsquigarrow \vec{0}$$

TransE: Weaknesses

- Symmetric relations
- 1-N, N-1 and N-N relations

KG Embedding: Research Directions

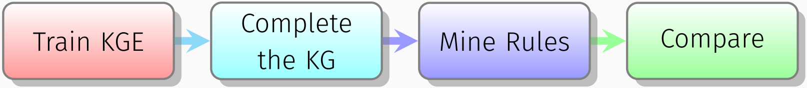
- Temporal KGEs
- Handle type hierarchy
- Incorporate constraints (e.g. subproperty)

KGE and Rule Mining

What rules can we learn (mine) with KGEs?

- Can we mine new rules by completion KGs?
- Is the choice of KGE model important?
- Does the performance on KG completion matter?

General Idea [JGO22]



TransE

- Translational
- Simple
- Issues with symmetric and non 1 to 1 relations

DistMult

- Tensor Factorisation
- Simple
- Issues with asymmetric relations

ComplEx

- Tensor Factorisation
- Needs more parameters than DistMult
- Can handle more “patterns”

Rule Mining with AMIE 3 [LGS20]

- Extract rules similar to Datalog rules
- Many heuristics and optimisations to navigate all possible rules
- Search guided by metrics:

  Head Coverage: proportion of instantions of the read correctly predicted

  PCA Confidence: proportion of correct and incorrect predictions, adjusted due to KG incompleteness

WN18RR

- Classical dataset (WordNet)
- Restricted version (6 relations)
- $\geq 88k$ triples

Family KG

- Based on Wikidata5M
- Restricted version (6 relations)
- $\approx 250k$ triples

Question Time: KG Completion

Which model you think was the best on the completion task?

`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

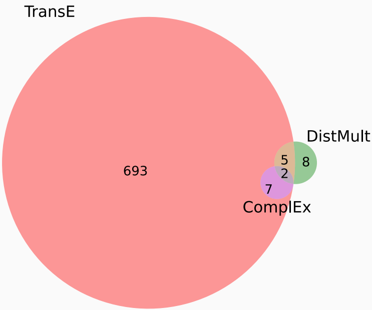
KG Completion Evaluation: WN18RR

Dataset	WN18RR KG				
Model	MR	MRR	Hits@K		
			1	3	10
Random	495.32	0.01	0.00	0.00	0.01
TransE	**34.29**	0.60	0.51	**0.66**	**0.76**
DistMult	152.37	**0.62**	**0.59**	0.63	0.66
ComplEx	139.36	0.59	0.57	0.60	0.63

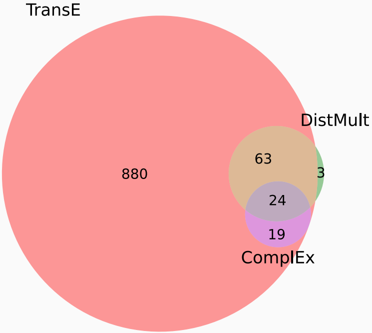
KG Completion Evaluation: Family KG

Dataset	Family KG				
Model	MR	MRR	Hits@K		
			1	3	10
Random	498.72	0.00	0.00	0.00	0.10
TransE	**2.59**	0.93	0.88	0.97	**0.99**
DistMult	7.45	0.98	**0.99**	**0.99**	**0.99**
ComplEx	4.64	**0.99**	0.98	**0.99**	**0.99**

Number of Rules



WN18RR



Family KG

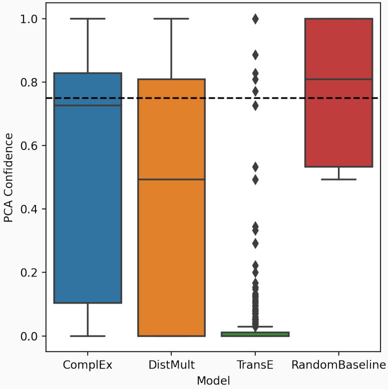
Question Time: Best Rules

Which model you think was the best on PCA Confidence?

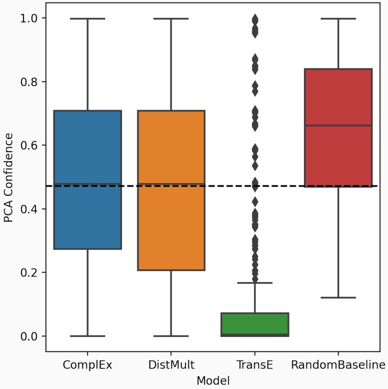
`https://www.menti.com/n27nnq4pi8`

`https://www.menti.com` and use the code: 56 97 85 68

PCA Confidence Results



(a) Original WN18RR KG



(b) Original family KG

Some (Bad) TransE-only Rules

- $spouse(x, y) \Rightarrow father(x, y)$
- $child(z, x) \land mother(z, y) \Rightarrow spouse(x, y)$
- $spouse(x, y) \Rightarrow child(x, y)$

TransE: many bad rules (very low PCA Confidence on the original data)

Random Baseline: Did not learn new rules

Examples of Plausible New Rules

- $father(x, y) \wedge mother(x, y) \Rightarrow child(x, y)$
- $relative(z, y) \wedge spouse(x, z) \Rightarrow relative(x, y)$

What happened with TransE?

- Many symmetric relations in both datasets
- As expected, they collapsed to the null vector

Take-home Messages

- KG Completion may increase the number of rules learned
- The KGE model affects the rules mined significantly
- What about the other way around?

Concluding Remarks

Recap

- Reasoning
- Deductive: RDFS, Datalog and DLs
- Inductive: FCA, Rule Mining and KGEs

Concluding Remarks

- Many different forms and approaches to reasoning
- A tool for each job
- Still in search of the “Holy Grail”
- There is much more to it still (some you will see in the research school!)
- Active area: combining deductive and inductive approaches

Thank you!

Questions?

E-mail: ricardo dot guimaraes at uib dot no

Website: `https://rfguimaraes.github.io`

License

References

References i

[Bor+13] Antoine Bordes et al. 'Translating embeddings for modeling multi-relational data'. In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.

[CGL12] Andrea Calì, Georg Gottlob and Thomas Lukasiewicz. 'A general Datalog-based framework for tractable query answering over ontologies'. In: *Journal of Web Semantics* 14 (July 2012), pp. 57–83. DOI: `10.1016/j.websem.2012.03.001`.

[FRP19] Bahare Fatemi, Siamak Ravanbakhsh and David Poole. 'Improved Knowledge Graph Embedding Using Background Taxonomic Information'. In: *AAAI*. AAAI Press, 2019, pp. 3526–3533.

[GO22] Ricardo Guimarães and Ana Ozaki. 'Reasoning in Knowledge Graphs (Invited Paper)'. en. In: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI: `10.4230/OASICS.AIB.2022.2`.

[Gui+21] Ricardo Guimarães et al. 'Mining EL Bases with Adaptable Role Depth'. In: *AAAI*. AAAI Press, 2021, pp. 6367–6374.

[JGO22] Johanna Jøsang, Ricardo Guimarães and Ana Ozaki. 'On the Effectiveness of Knowledge Graph Embeddings: a Rule Mining Approach'. In: (June 2022). arXiv: `2206.00983 [cs.LG]`.

[LGS20] Jonathan Lajus, Luis Galárraga and Fabian Suchanek. 'Fast and Exact Rule Mining with AMIE 3'. In: *The Semantic Web*. Springer International Publishing, 2020, pp. 36–52. DOI: `10.1007/978-3-030-49461-2_3`.