Graph-based embeddings in mOWL

Fernando Zhapa-Camacho

<2023-02-13 Mon>

Learning objectives

- Learn about different graph representation of ontologies
- Learn how to use graphs from ontologies in machine learning

Introduction

Ontologies axioms are divided into: ABox
assertions of the world

- `Father(John)`
- `hasChild(John, Mary)`

Introduction

Ontologies axioms are divided into: RBox

- relationships between roles:
- `is_part_of` inverseOf `has_part`
- `negatively_regulates subpPropertyOf regulates`

Introduction

Ontologies axioms are divided into: TBox

- Concept descriptions
- `Mother subClassOf Person`

Graphs from ontologies

- ABox and RBox can be easily transformed into a Knowledge Graph
  - `Father(John)`: (John, instanceof, Father)
  - `hasChild(John, Mary)`: (John, hasChild, Mary)

Graphs from ontologies

- ABox and RBox can be easily transformed into a Knowledge Graph
  - `Father(John)`: (John, instanceof, Father)
  - `hasChild(John, Mary)`: (John, hasChild, Mary)
- What about the TBox?

Graphs from ontologies

- ABox and RBox can be easily transformed into a Knowledge Graph
  - `Father(John)`: (John, instanceof, Father)
  - `hasChild(John, Mary)`: (John, hasChild, Mary)
- What about the TBox?
- Multiple methods to *project* an ontology into a graph

Graphs from ontologies

- ABox and RBox can be easily transformed into a Knowledge Graph
  - `Father(John)`: (John, instanceof, Father)
  - `hasChild(John, Mary)`: (John, hasChild, Mary)
- What about the TBox?
- Multiple methods to *project* an ontology into a graph
- None of them is perfect, there is loss of information

Methods implemented in mOWL

Taxonomy

- Only parses axioms of the form $C \sqsubseteq D$
- C, D are atomic concepts
- Graphs represent the hierarchy of concepts

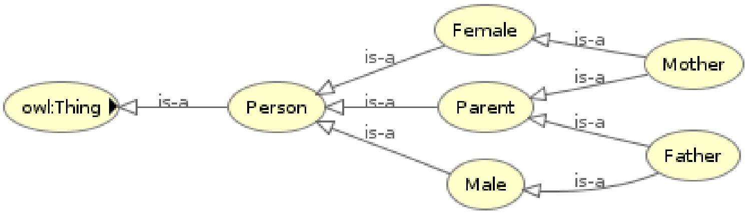

Figure: Family ontology representation

Methods implemented in mOWL

DL2Vec

Condition 1	Condition 2	Triple(s)
$A \sqsubseteq QR_0 \dots QR_m D$ $A \equiv QR_0 \dots QR_m D$	$D := B_1 \sqcup \dots \sqcup B_n \mid B_1 \sqcap \dots \sqcap B_n$	$\langle A, (R_0 \dots R_m), B_i \rangle$ for $i \in 1 \dots n$
$A \sqsubseteq B$		$\langle A, SubClassOf, B \rangle$
$A \equiv B$		$\langle A, EquivalentTo, B \rangle$

Figure: DL2Vec projection rules

Methods implemented in mOWL

OWL2Vec*

Axiom of condition 1	Axiom or triple(s) of condition 2	Projected triple(s)
$A \sqsubseteq \square r.\, D$	$D \equiv B \mid B_1 \sqcup \ldots \sqcup B_n \mid B_1 \sqcap \ldots \sqcap B_n$	$\langle A, r, B \rangle$ or
or		
$\square r.\, D \sqsubseteq A$		
$\exists r.\, \top \sqsubseteq A$ (domain)	$\top \sqsubseteq \forall r.\, B$ (range)	$\langle A, r, B_i \rangle$ for $i \in 1, \ldots, n$
$A \sqsubseteq \exists r.\, \{b\}$	$B(b)$	
$r \sqsubseteq r'$	$\langle A, r', B \rangle$ has been projected	
$r' \equiv r^-$	$\langle B, r', A \rangle$ has been projected	
$s_1 \circ \ldots \circ s_n \sqsubseteq r$	$\langle A, s_1, C_1 \rangle \ldots \langle C_n, s_n, B \rangle$ have been projected	
$B \sqsubseteq A$	–	$\langle B, rdfs{:}subClassOf, A \rangle$
		$\langle A, rdfs{:}subClassOf^-, B \rangle$
$A(a)$	–	$\langle a, rdf{:}type, A \rangle$
		$\langle A, rdf{:}type^-, a \rangle$
$r(a, b)$	–	$\langle a, r, b \rangle$

$\square$ is one of: $\geq, \leq, =, \exists, \forall$. A, B, B_i and C_i are atomic concepts (classes), s_i, r and r' are roles (object properties), r^- is the inverse of a relation r, a and b are individuals (instances), $\top$ is the top concept (defined by *owl:Thing*)

Figure: OWL2Vec* projection rules

Limitations of projection methods

- Some are not complete (Taxonomical projection):
  - $(C \sqcap D) \sqsubseteq E \rightarrow \textit{undefined}$

Limitations of projection methods

- Some are not complete (Taxonomical projection):
  - $(C \sqcap D) \sqsubseteq E \rightarrow$ *undefined*
- Some are not invertible (OWL2Vec*):
  - $(C \sqsubseteq \exists R.D) \rightarrow (C, R, D)$
  - $(C \sqsubseteq \forall R.D) \rightarrow (C, R, D)$
  - Inverse of $(C, R, D) \rightarrow$??

We have graphs. Next step?

- Ontology has been transformed into a graph
- Graphs as input for a machine learning model
- mOWL supports two ways to *embed* a graph:
  - Random-walk based embeddings
  - Translational models (and more)

Embedding with random walks

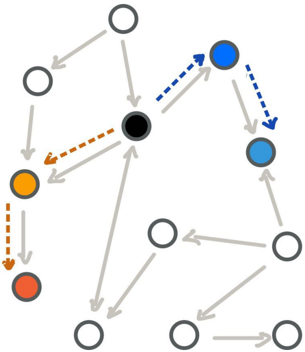

Figure: Random walks

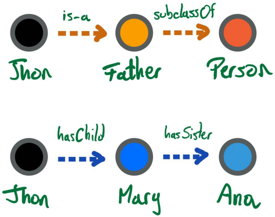

Figure: Sequences generated

Embedding with random walks

- Generated sentences become input in a language processing model:
  - Word2Vec
  - Transformers

Embedding with random walks

- Word2Vec embeddings preserve co-occurrence
- Generated embeddings can be used to compute similarity between entities E_1 and E_2
- Similarity can be computed using the dot product

$$|v(E_1) \cdot v(E_2)| = 1 \rightarrow \textit{similar}$$

$$|v(E_1) \cdot v(E_2)| = 0 \rightarrow \textit{dissimilar}$$

Embedding with random walks

- Word2Vec embeddings preserve co-occurrence
- Generated embeddings can be used to compute similarity between entities E_1 and E_2
- Similarity can be computed using the dot product

$$|v(E_1) \cdot v(E_2)| = 1 \rightarrow \textit{similar}$$

$$|v(E_1) \cdot v(E_2)| = 0 \rightarrow \textit{dissimilar}$$

- Protein-protein interactions
- Gene-disease associations

Translational models

- Graphs are composed by triples *head*, *relation*, *tail*
- Translational models consider *relation* to be a translation operation between *head* and tail

$$||h + r - t|| \approx 0$$

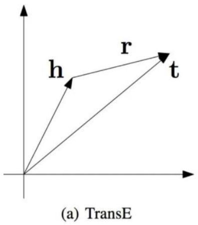

Figure: TransE

Translational models

- The *score* of a triple is given by $d(h, r, t) = ||h + r - t||$, where the lower d is, the more plausible the triple to hold true.
- TransE is the most representative model.

$$\mathcal{L} = \left[\gamma + d(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t}) - d\left(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t}'\right)\right]_+$$

- $d(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t})$ is the score of a *positive* triple
- $d\left(\boldsymbol{h} + \boldsymbol{r}, \boldsymbol{t}'\right)$ is the score of a *negative* triple
- γ is a margin parameter

Translational models

- Other models have appeared by changing

$$h + r \approx t$$

into

$$f(h) + f(r) \approx f(t)$$

- Some variations of TransE are:
    - TransH
    - TransR
    - TransD

Translational Embeddings:

- We can use Translational models to predict new triples:

$$(C, R, D$$

- Those triples can be transformed into axioms:

$$C \sqsubseteq \exists R.D$$