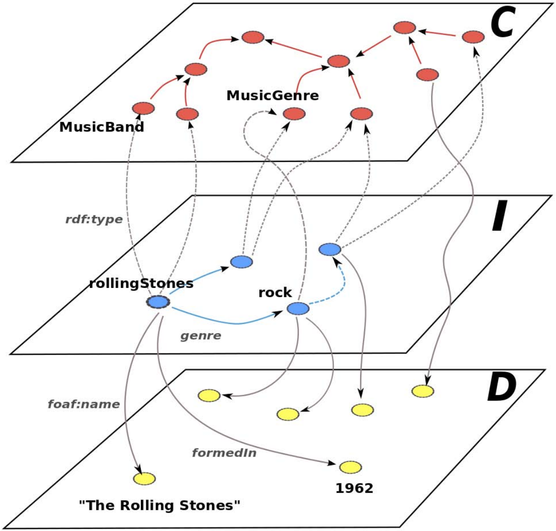
Semantic similarity and machine learning with ontologies

Robert Hoehndorf and Maxat Kulmanov

Graph-based Learning



From Harispe et al., Semantic Similarity From Natural Language And Ontology Analysis, 2015.

How to measure similarity?

- Shortest Path
  - applicable to arbitrary "knowledge graphs"
  - does not capture similarity well over all edge types, e.g., *disjointWith*, *differentFrom*, *opposite-of*, etc.
- Random Walk
  - with or without restart
  - iterated
  - does not consider edge labels $\Rightarrow$ captures only adjacency of nodes
  - scores whole graph with *probability* of being in a state
  - can take multiple seed nodes
    - can be used to find disease genes

Graph-based learning

- feature learning on graphs

Graph-based learning

- feature learning on graphs
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - RDF2Vec: `http://data.dws.informatik.uni-mannheim.de/rdf2vec/`
  - with support for reasoning over ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`

Graph-based learning

- feature learning on graphs
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - RDF2Vec: `http://data.dws.informatik.uni-mannheim.de/rdf2vec/`
  - with support for reasoning over ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`
- Translational knowledge graph embeddings: TransE, TransR, TransE, HolE, etc.
  - analogy- or translation-based
  - `https://github.com/SmartDataAnalytics/PyKEEN`

Graph-based learning

- feature learning on graphs
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - RDF2Vec: `http://data.dws.informatik.uni-mannheim.de/rdf2vec/`
  - with support for reasoning over ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`
- Translational knowledge graph embeddings: TransE, TransR, TransE, HolE, etc.
  - analogy- or translation-based
  - `https://github.com/SmartDataAnalytics/PyKEEN`
- Graph Convolution Neural Networks (not discussed here)

Graph embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be an ontology graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. An ontology graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

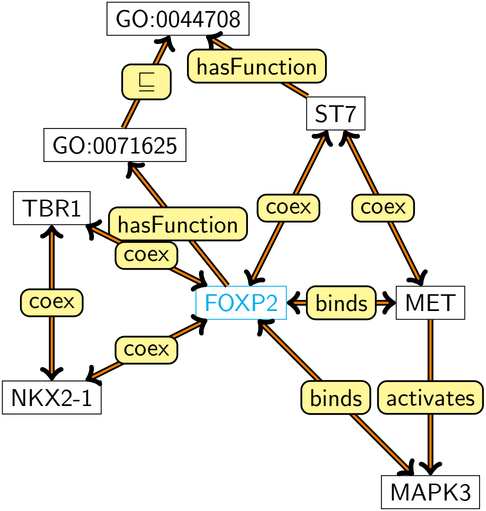
Graph embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be an ontology graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. An ontology graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

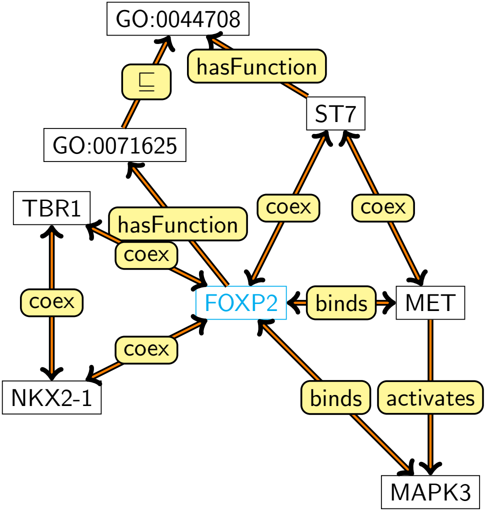
- key idea: preserve *some* structure of the graph in $\mathbb{R}^n$ (under operations in $\mathbb{R}^n$)
- $\mathbb{R}^n$ enables *new* operations (such as many similarity measures)
- useful as *feature* vectors

Random walks



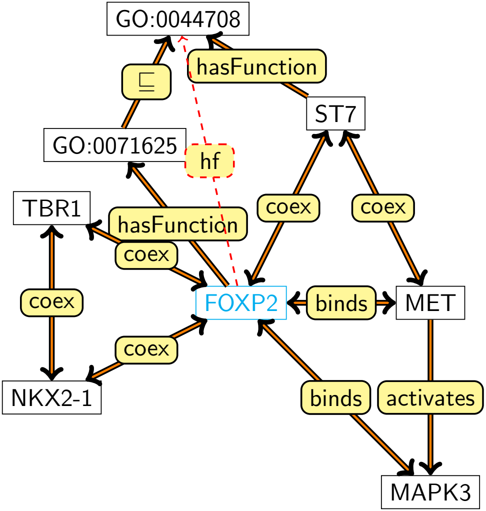
- FOXP2 is characterized by *adjacent* and close nodes and edges
- different edges may “transmit” information differently

Random walks



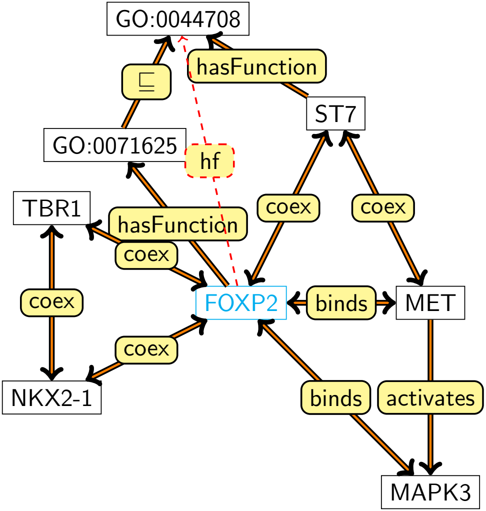
- precompute the deductive closure:
- for all ϕ: if $\mathcal{KG} \models \phi$, add ϕ to $\mathcal{KG}$

Random walks



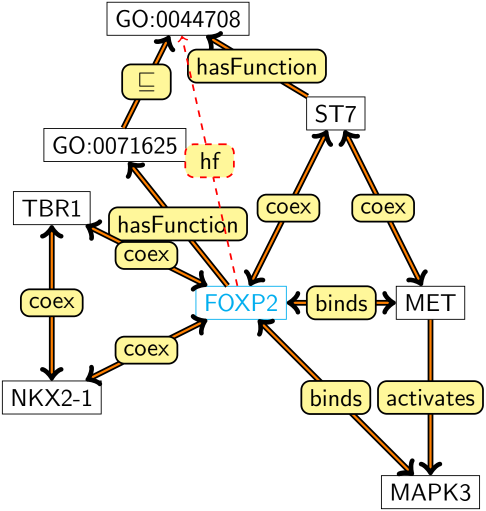
- precompute the deductive closure:
- for all ϕ: if $\mathcal{KG} \models \phi$, add ϕ to $\mathcal{KG}$

Random walks



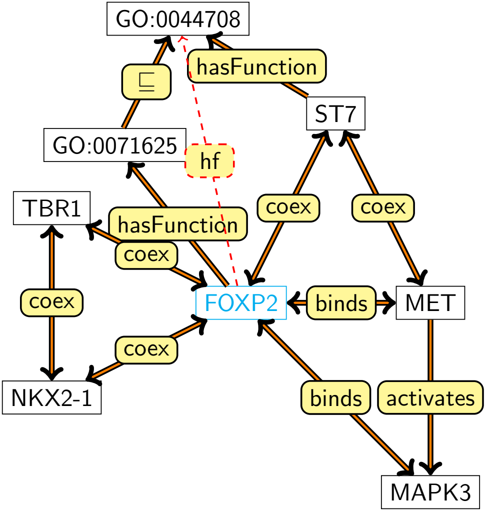
- Exploring the graph:

Random walks



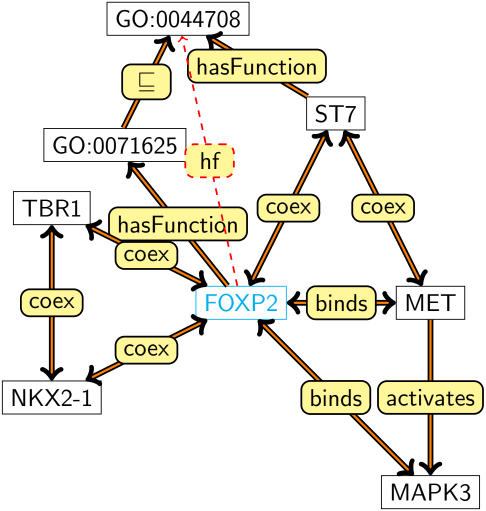
- Exploring the graph:
- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708

Random walks



- Exploring the graph:
- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708

Random walks



- Exploring the graph:
- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Word2Vec and Random Walks

- random walks “flatten” a graph
  - walks capture node neighborhood
  - and generate a “corpus”
- random walks capture graph “structure”
  - in ABox and TBox
  - hub-nodes, communities, etc.
  - determine “importance” of nodes
- embeddings capture co-occurrence
  - similar graph neighborhood $\Rightarrow$ similar co-occurrence $\Rightarrow$ similar vector
- embeddings generate “feature” vectors
  - functions from symbols (words, labels) into $\mathbb{R}^n$

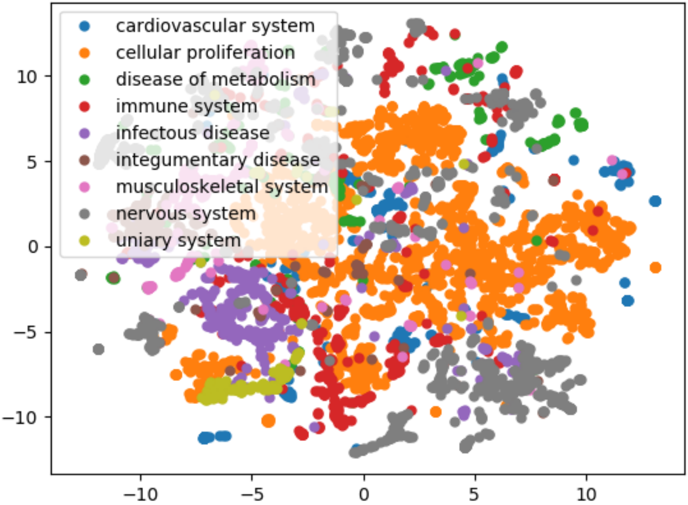
What to do with embeddings?

- useful for edge prediction, similarity, clustering, as feature vectors
  - supervised: edge prediction (e.g., SVM, ANN)
    - e.g.: find a function $f : \mathbb{R}^n \times \mathbb{R}^n \mapsto [0, 1]$ s.t. $\sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$ (RMSE) is minimized for a set of true labels y_k
  - unsupervised: clustering, similarity, visualization
    - cosine similarity (for L2-normalized features)
    - Word2Vec embeddings capture similarity between co-occurrence vectors

Visualizing feature vectors: dimensionality reduction

- project *n*-dimensional vectors in 2D (or 3D) space
- and color with some known labels
  - high-level/general classes in an ontology work great
- PCA or t-SNE
- `https://lvdmaaten.github.io/tsne/`

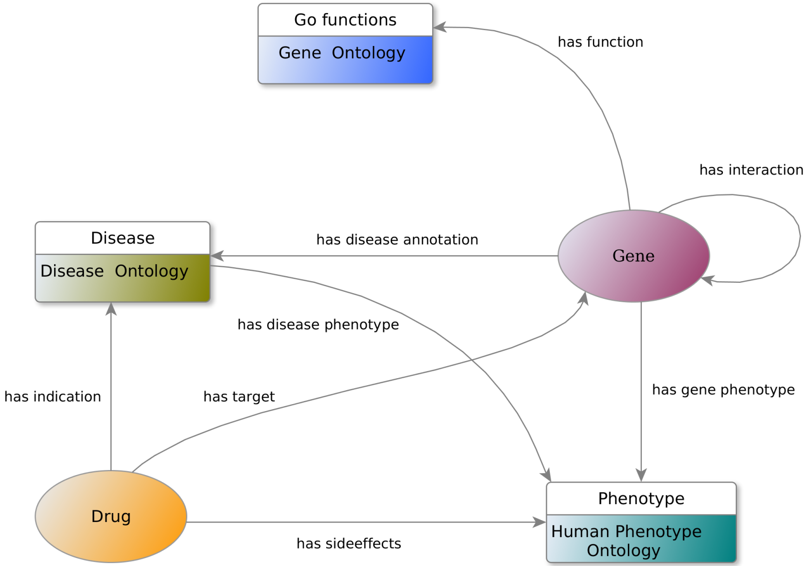
Visualizing feature vectors



Features: supervised learning

- feature vectors represent graph neighborhood of nodes
  - adjacent nodes and edges
  - ontology classes (asserted & inferred)
- useful in supervised prediction tasks
- relation prediction:
  - input: two features vectors (from embedding function)
  - output: 0 or 1 (relation or not)
  - training data: positive and negative cases
    - $R(x, y)$ and $\neg R(x, y)$
    - $R(x, y)$ and not provable $R(x, y)$

Features: supervised learning



Features: supervised learning

Object property	Source type	Target type	Without reasoning		With reasoning	
			F-measure	AUC	F-measure	AUC
has target	Drug	Gene/Protein	0.94	0.97	0.94	0.98
has disease annotation	Gene/Protein	Disease	0.89	0.95	0.89	0.95
has side-effect*	Drug	Phenotype	0.86	0.93	0.87	0.94
has interaction	Gene/Protein	Gene/Protein	0.82	0.88	0.82	0.88
has function*	Gene/Protein	Function	0.85	0.95	0.83	0.91
has gene phenotype*	Gene/Protein	Phenotype	0.84	0.91	0.82	0.90
has indication	Drug	Disease	0.72	0.79	0.76	0.83
has disease phenotype*	Disease	Phenotype	0.72	0.78	0.70	0.77

Ontologies, graphs, and text

The forkhead-box P2 (FOXP2) gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between FOXP2 gene polymorphism and clinical symptoms in schizophrenia.

Ontologies, graphs, and text

The forkhead-box P2 (FOXP2) gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between FOXP2 gene polymorphism and clinical symptoms in schizophrenia.

- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Ontologies, graphs, and text

The :FOXP2 gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between :FOXP2 gene polymorphism and clinical symptoms in schizophrenia.

- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Tools and resources

- RDF2Vec: random walks on RDF + Word2Vec
- RDF2Vec: Weisfeiler-Lehmann kernel on RDF
- `https://datalab.rwth-aachen.de/embedding/RDF2Vec/`

Tools and resources

- RDF2Vec: random walks on RDF + Word2Vec
- RDF2Vec: Weisfeiler-Lehmann kernel on RDF
- `https://datalab.rwth-aachen.de/embedding/RDF2Vec/`
- Walking RDF+OWL: random walks on RDF + Elk + Word2Vec
  - inference
- `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`

Some limitations

- “word”-based (Word2Vec):
  - semantics is reduced to co-occurrence (in ABox/TBox statements)
  - “disjointWith” vs. “part-of” vs. “subClassOf”

Jupyter excercise

- Open the Jupyter notebook `graph.ipynb`
- Follow the examples in the first part of the notebook (random walks)
- If you don't have a powerful CPU in your laptop (with multiple cores), you may want to lower the number of iterations (`n_iter`) during TSNE
- some of the code will take a while to run
  - if things are too slow, you can keep it running while we continue or complete this after the tutorial
- (some notes on parameters and hyperparameters...)

Translating embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be a knowledge graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

Translating embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be a knowledge graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

Graph as edgelist: set of (s, p, o) statements

Translating embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be a knowledge graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

Graph as edgelist: set of (s, p, o) statements
Idea: $\mu(s) + \mu(p) \approx \mu(o)$

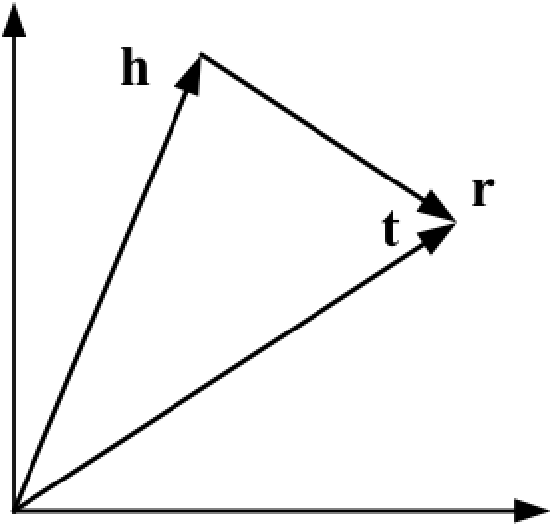
Translating embeddings

Definition

Let $KG = (V, E, L; \vdash)$ be a knowledge graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a set of labels Lab to vertices and edges, and an inference relation $\vdash$. A knowledge graph embedding is a function $f_\eta : L(V) \cup L(E) \mapsto \mathbf{R}^n$.

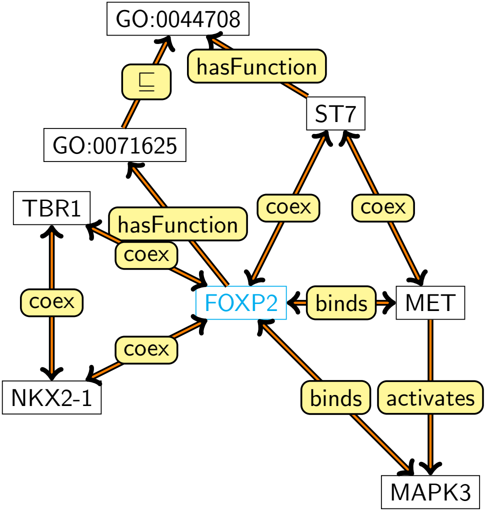
Graph as edgelist: set of (s, p, o) statements
Idea: $\mu(s) + \mu(p) \approx \mu(o)$
Minimize: $\sum_t \|\mu(s) + \mu(p) - \mu(o)\|$ (chose your norm, usually L2)

Translating embeddings

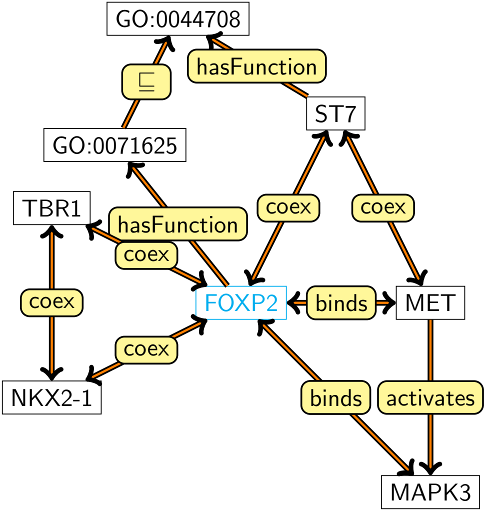


Entity and Relation Space

Translating embeddings

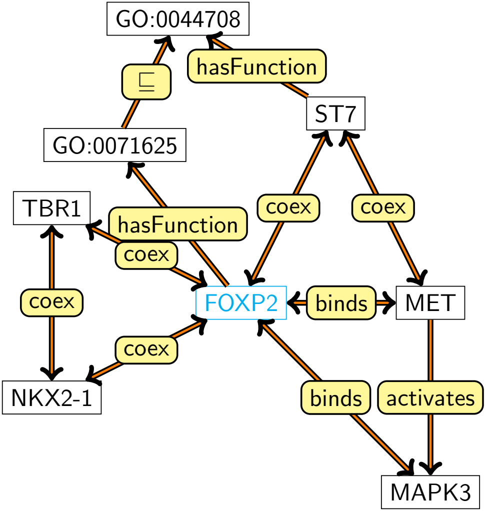


Translating embeddings



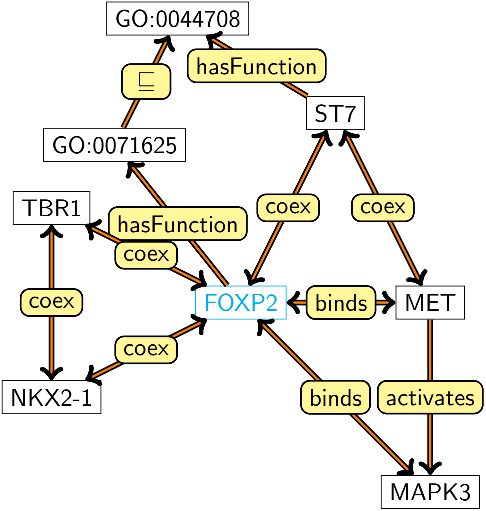
- FOXP2 + binds = MET

Translating embeddings



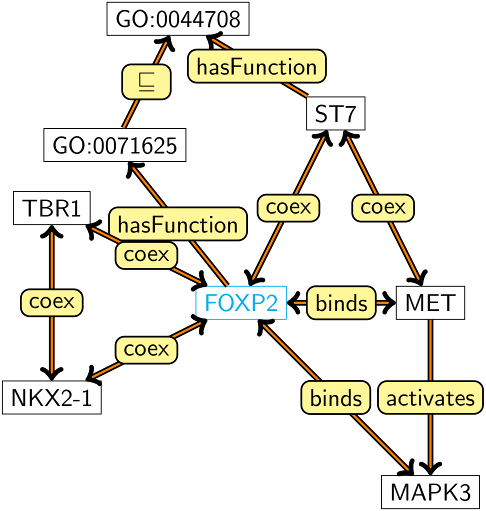
- FOXP2 + binds = MET
- MET + activates = MAPK3

Translating embeddings



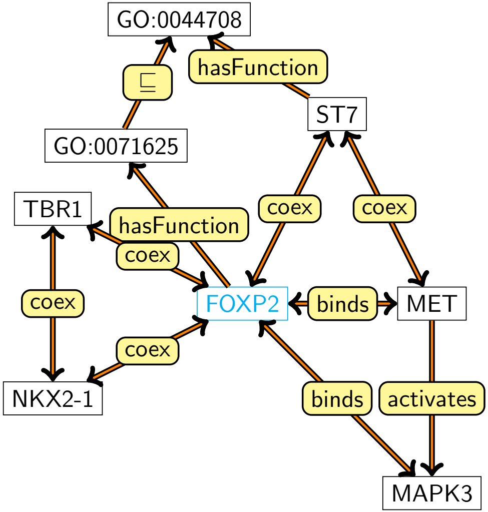
- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2

Translating embeddings



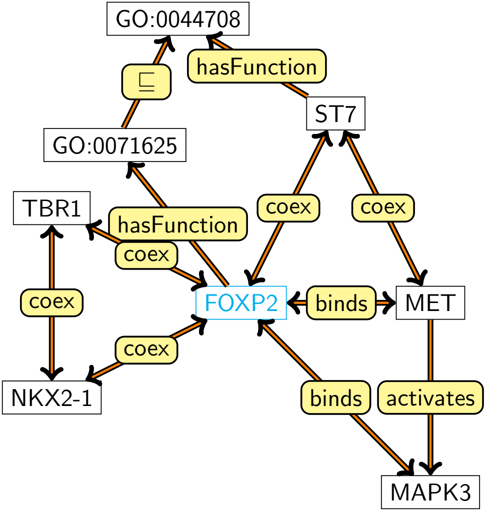
- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2
- ST7 + hasFunction = `GO:0044708`

Translating embeddings



- FOXP2 + binds = MET
- MET + activates = MAPK3
- MET + binds = FOXP2
- ST7 + hasFunction = `GO:0044708`
- ...

Translating embeddings



- FOXP2 + binds - MET = 0
- MAP + activates - MAPK3 = 0
- MET + binds - FOXP2 = 0
- ST7 + hasFunction - `GO:0044708` = 0
- ...

Translating embeddings

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$
  $\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$
  $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop
  $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$
  $S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b
  $T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets
  for $(h, \ell, t) \in S_{batch}$ **do**
    $(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet
    $T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$
  end for
  Update embeddings w.r.t. $\displaystyle\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$

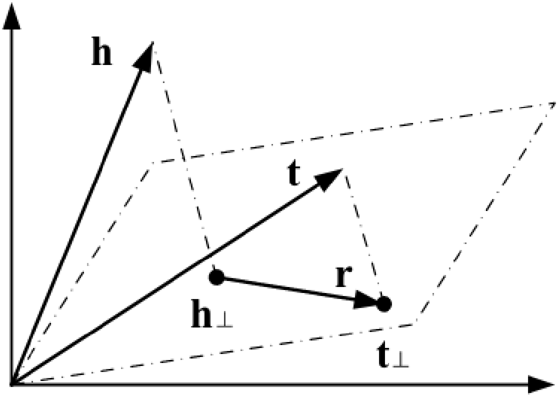
end loop

Bordes et al. (2013). Translating Embeddings for Modeling Multi-relational Data.

Some properties of TransE

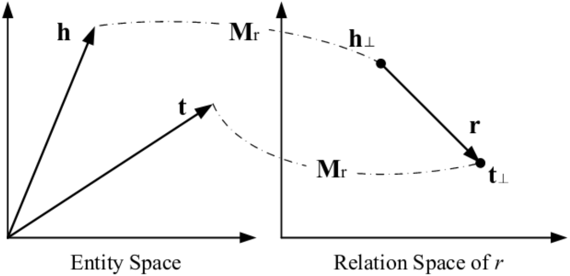
- graph-based
  - works well on RDF graphs
  - and ontology graphs
- 1:1 relations only
  - not suitable for hierarchies (1-N relations)
  - not suitable for N-N relations
  - no transitive, symmetric, reflexive relations

Translating embeddings



Entity and Relation Space

Translating embeddings



(c) TransR.

Translating embeddings

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\|(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\|_2^2$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1$ $\|\mathbf{w}_r^\top\mathbf{r}\|/\|\mathbf{r}\|_2\le\epsilon, \|\mathbf{w}_r\|_2=1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$ $\|\mathbf{M}_r\mathbf{h}\|_2\le1, \|\mathbf{M}_r\mathbf{t}\|_2\le1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$ $\|(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\|_2\le1$ $\|(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\|_2\le1$
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$ $\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$	$-\|\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\|_{1/2}^2$ $-\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_{1/2}^2$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$ $\|\mathbf{M}_r(\theta_r)\mathbf{h}\|_2\le1, \|\mathbf{M}_r(\theta_r)\mathbf{t}\|_2\le1$ $\|\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\|_2\le1, \|\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\|_2\le1$
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\theta_r\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2-\theta_r^2)^2$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$-(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)^\top\mathbf{M}_r(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|)$	$\|\mathbf{h}\|_2\le1, \|\mathbf{t}\|_2\le1, \|\mathbf{r}\|_2\le1$ $\|\mathbf{M}_r\|_F\le1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge0$
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ $\boldsymbol{\Sigma}_h,\boldsymbol{\Sigma}_t\in\mathbb{R}^{d\times d}$	$\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$ $\boldsymbol{\mu}_r\in\mathbb{R}^d, \boldsymbol{\Sigma}_r\in\mathbb{R}^{d\times d}$	$-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$ $-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$ $\boldsymbol{\mu}=\boldsymbol{\mu}_h+\boldsymbol{\mu}_r-\boldsymbol{\mu}_t$ $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_t$	$\|\boldsymbol{\mu}_h\|_2\le1, \|\boldsymbol{\mu}_t\|_2\le1, \|\boldsymbol{\mu}_r\|_2\le1$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_t\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_r\le c_{max}\mathbf{I}$
TransG [46]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\sigma_h^2\mathbf{I})$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma_t^2\mathbf{I})$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$	$\boldsymbol{\mu}_r^i\sim\mathcal{N}(\boldsymbol{\mu}_t-\boldsymbol{\mu}_h,(\sigma_h^2+\sigma_t^2)\mathbf{I})$ $\mathbf{r}=\sum_i\pi_r^i\boldsymbol{\mu}_r^i\in\mathbb{R}^d$	$\sum_i\pi_r^i\exp\left(-\frac{\|\boldsymbol{\mu}_h+\boldsymbol{\mu}_r^i-\boldsymbol{\mu}_t\|_2^2}{\sigma_h^2+\sigma_t^2}\right)$	$\|\boldsymbol{\mu}_h\|_2\le1, \|\boldsymbol{\mu}_t\|_2\le1, \|\boldsymbol{\mu}_r^i\|_2\le1$
UM [56]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	—	$-\|\mathbf{h}-\mathbf{t}\|_2^2$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$
SE [57]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$	$-\|\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\|_1$	$\|\mathbf{h}\|_2=1, \|\mathbf{t}\|_2=1$

Wang et al. Knowledge Graph Embedding: A Survey ofApproaches and Applications.

PyKEEN

- Python package to generate knowledge graph embeddings
- supports many different graph embedding types: TransE, TransR, TransD, RESCAL, etc.
- hyperparameter optimization ("HPO") and evaluation included
- `https://github.com/SmartDataAnalytics/PyKEEN`

Some limitations

- graph-based (same as random walks):
  - ontologies are not graphs!
  - converting ontologies to graphs loses information
  - no axioms, no definitions
- (this also holds for Graph Convolutional Networks, which are not covered here)

Jupyter excercise

- run the PyKEEN part of `graph.ipynb`
- again: this may take a while
- you can also explore `https://github.com/SmartDataAnalytics/PyKEEN`
- try to expand the notebook to predict “new” relations
  - using numpy directly, or PyKEEN’s predictions methods
- Change the TSNE to work only on enzymes (don’t include the GO classes, etc.)

How to overcome the semantic gap?

- none of the models discussed above are truly "semantic"
  - all syntactic
  - graph-based or based on axioms

How to overcome the semantic gap?

- none of the models discussed above are truly "semantic"
  - all syntactic
  - graph-based or based on axioms
- what do we actually mean by "semantics"?

How to overcome the semantic gap?

- none of the models discussed above are truly "semantic"
  - all syntactic
  - graph-based or based on axioms
- what do we actually mean by "semantics"?
  - formal definition of "truth" relies on "models"

How to overcome the semantic gap?

- none of the models discussed above are truly "semantic"
  - all syntactic
  - graph-based or based on axioms
- what do we actually mean by "semantics"?
  - formal definition of "truth" relies on "models"
  - universal algebra over formal languages (with signature Σ)

Description Logic EL++

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
generalized concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role inclusion	$r_1 \circ \ldots \circ r_n \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \ldots \circ r_n^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

Models

- Interpretations and Σ-structures
- Model $\mathfrak{A}$ of a formula ϕ: ϕ is true in $\mathfrak{A}$ ($\mathfrak{A} \models \phi$)
- Theory T: set of formulas
- $\mathfrak{A}$ is a model of T if $\mathfrak{A}$ is a model of all formulas in T
- Ontologies are (special kinds of) theories

EL Embeddings

- given a theory/ontology T with signature $\Sigma(T)$
- aim: find $f_e : \Sigma(T) \mapsto \mathbb{R}^n$ s.t. $f_e(\Sigma(T))$ is a model of T ($f_e(\Sigma(T)) \models T$)

EL Embeddings

- given a theory/ontology T with signature $\Sigma(T)$
- aim: find $f_e : \Sigma(T) \mapsto \mathbb{R}^n$ s.t. $f_e(\Sigma(T))$ is a model of T ($f_e(\Sigma(T)) \models T$)
- more general: find an algorithm that maps symbols (signatures) into $\mathbb{R}^n$ so that the *semantics* of the symbol (expressed through axioms and explicit in model structures) is preserved
  - or: the embedding function *is* an interpretation function

EL Embeddings

- given a theory/ontology T with signature $\Sigma(T)$
- aim: find $f_e : \Sigma(T) \mapsto \mathbb{R}^n$ s.t. $f_e(\Sigma(T))$ is a model of T ($f_e(\Sigma(T)) \models T$)
- more general: find an algorithm that maps symbols (signatures) into $\mathbb{R}^n$ so that the *semantics* of the symbol (expressed through axioms and explicit in model structures) is preserved
    - or: the embedding function *is* an interpretation function
- any consistent $\mathcal{EL}^{++}$theory has infinite models

EL Embeddings

- given a theory/ontology T with signature $\Sigma(T)$
- aim: find $f_e : \Sigma(T) \mapsto \mathbb{R}^n$ s.t. $f_e(\Sigma(T))$ is a model of T ($f_e(\Sigma(T)) \models T$)
- more general: find an algorithm that maps symbols (signatures) into $\mathbb{R}^n$ so that the *semantics* of the symbol (expressed through axioms and explicit in model structures) is preserved
  - or: the embedding function *is* an interpretation function
- any consistent $\mathcal{EL}^{++}$theory has infinite models
- any consistent $\mathcal{EL}^{++}$theory has models in $\mathbb{R}^n$ (Loewenheim-Skolem, upwards; compactness)

Key idea

- for all $r \in \Sigma(T)$ and $C \in \Sigma(T)$, define $f_e(r)$ and $f_e(C)$
- $f_e(C)$ maps to points in an open n-ball such that $f_e(C) = C^{\mathcal{I}}$: $C^{\mathcal{I}} = \{x \in \mathbb{R}^n | \, \|f_e(C) - x\| < r_e(C)\}$
  - these are the *extension* of a class in $\mathbb{R}^n$
- $f_e(r)$ maps a binary relation r to a vector such that $r^{\mathcal{I}} = \{(x, y) | x + f_e(r) = y\}$
  - that's the TransE property for *individuals*
- use the axioms in T as constraints

Algorithm

- normalize the theory:
  - every $\mathcal{EL}^{++}$theory can be expressed using four normal forms (Baader et al., 2005)
- eliminate the ABox: replace each individual symbol with a singleton class: a becomes $\{a\}$
- rewrite relation assertions $r(a, b)$ and class assertions $C(a)$ as $\{a\} \sqsubseteq \exists r.\{b\}$ and $\{a\} \sqsubseteq C$
  - something to remember for the next class-vs-instance discussion?
- normalization rules to generate:
  - $C \sqsubseteq D$
  - $C \sqcap D \sqsubseteq E$
  - $C \sqsubseteq \exists R.D$
  - $\exists R.C \sqsubseteq D$

Algorithm: loss functions

$$
\begin{aligned}
& loss_{C \sqsubseteq D}(c, d) = \\
& \max(0, \|f_\eta(c) - f_\eta(d)\| + r_\eta(c) - r_\eta(d) - \gamma) \\
& + | \|f_\eta(c)\| - 1| + | \|f_\eta(d)\| - 1|
\end{aligned} \tag{1}
$$

Algorithm: loss functions

Let $h = \frac{r_\eta(c)^2 - r_\eta(d)^2 + \|f_\eta(c) - f_\eta(d)\|^2}{2\|f_\eta(c) - f_\eta(d)\|}$, then the center and radius of the smallest n-ball containing the intersection of $\eta(C)$ and $\eta(D)$ are $f_\eta(c) + \frac{h}{\|f_\eta(c) - f_\eta(d)\|}(f_\eta(d) - f_\eta(c))$ and $\sqrt{r_\eta(c)^2 - h^2}$.

Algorithm: loss functions

$$
\begin{aligned}
&loss_{C \sqsubseteq \exists R.D}(c, d, r) = \\
&\max(0, \|f_\eta(c) + f_\eta(r) - f_\eta(d)\| + r_\eta(c) - r_\eta(d) - \gamma) \\
&+ |\, \|f_\eta(c)\| - 1| + |\, \|f_\eta(d)\| - 1|
\end{aligned} \tag{2}
$$

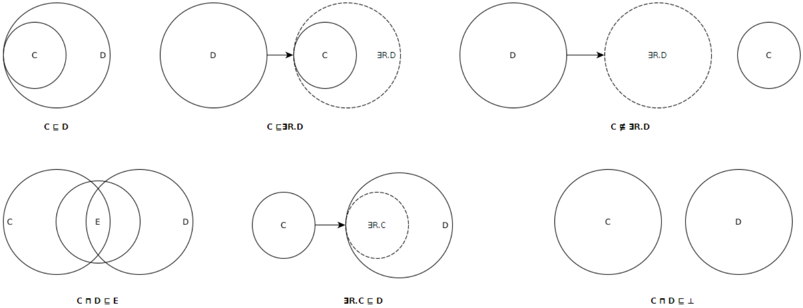
Algorithm: loss functions

$$
\begin{aligned}
&loss_{\exists R.C \sqsubseteq D}(c, d, r) = \\
&\max(0, \|f_\eta(c) - f_\eta(r) - f_\eta(d)\| - r_\eta(c) - r_\eta(d) - \gamma) \\
&+ |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1|
\end{aligned} \tag{3}
$$

Algorithm: loss functions

$$
\begin{aligned}
&loss_{C \sqcap D \sqsubseteq \perp}(c, d, e) = \\
&\max(0, r_\eta(c) + r_\eta(d) - \|f_\eta(c) - f_\eta(d)\| + \gamma) \\
&+ |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1|
\end{aligned} \tag{4}
$$

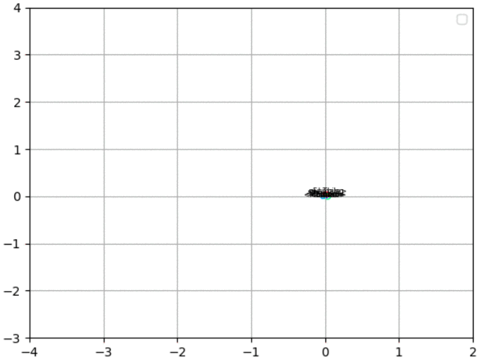
Algorithm: loss functions



EL Embeddings

$$
\begin{align}
Male &\sqsubseteq Person \tag{5}\\
Female &\sqsubseteq Person \tag{6}\\
Father &\sqsubseteq Male \tag{7}\\
Mother &\sqsubseteq Female \tag{8}\\
Father &\sqsubseteq Parent \tag{9}\\
Mother &\sqsubseteq Parent \tag{10}\\
Female \sqcap Male &\sqsubseteq \bot \tag{11}\\
Female \sqcap Parent &\sqsubseteq Mother \tag{12}\\
Male \sqcap Parent &\sqsubseteq Father \tag{13}\\
\exists hasChild.Person &\sqsubseteq Parent \tag{14}\\
Parent &\sqsubseteq Person \tag{15}\\
Parent &\sqsubseteq \exists hasChild.\top \tag{16}
\end{align}
$$

EL Embeddings



- model with $\Delta = R^n$
- support quantifiers, negation, conjunction,...

Jupyter excercise

- Run the new Docker image `coolmaksat/embeddings:latest`
- `docker run -i -t -p 8888:8888 coolmaksat/embeddings /bin/bash -c "jupyter notebook --notebook-dir=/usr/src/app/ --ip='0.0.0.0' --port=8888 --no-browser --allow-root"`

Summary

- ontologies contain background knowledge that is useful as background knowledge:
  - axioms
  - natural language (definitions, labels, synonyms)

Summary

- ontologies contain background knowledge that is useful as background knowledge:
  - axioms
  - natural language (definitions, labels, synonyms)
- feature learning (deep learning) on ontologies encodes this background knowledge
  - using ontology graphs, axioms, or model structures

Open research questions

Where is our semantics, in the machine learning model or the axioms?

- implicit or explicit?
- hidden or interpretable?
- example: transitive relations
- combination of both?

Open research questions

How do we evaluate our models and methods?

- random splits (standard in machine learning)
- time-based splits
- other?

Open research questions

What is the interface between knowledge representation and learning?

- *How* to represent knowledge affects learning outcomes
- representation patterns and learning $\Rightarrow$ specific algorithms?

Acknowledgements

Lots of help from students:

- Mona Alshahrani
- Fatima Smaili

and colleagues:

- Mehdi Ali & Hajira Jabeen (PyKEEN)
- Michel Dumontier
- Andreas Karwath
- Paul Schofield

and KAUST (Bio2Vec grant, award no. URF/1/3454-01-01).