Ontology (Knowledge Graph) Embeddings

Ernesto Jiménez-Ruiz

Lecturer in Artificial Intelligence

Before we start...

Students' module evaluation

- Your feedback is very important.
- Evaluations are anonymous.
- `https://city.surveys.evasysplus.co.uk`
- Access also via **MyMoodle**
- More information on **Student Hub**.

✗ Scores 1, 2, 3 are considered negative.

✓ Scores 4 and 5 are positive.

Drop-in sessions until submission

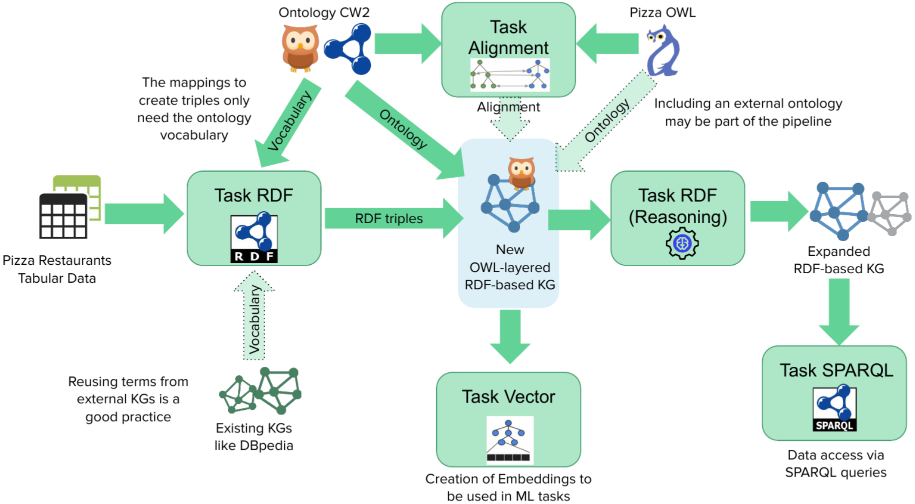
- April 4 (Today, cancelled).
- April 9 (online 10am). Tuesday.
- April 11 (on campus 2-4pm). Thursday.
- April 16 (online 10am). Tuesday.
- April 17 (online 1-3pm). Wednesday.
- April 23 (online 10am-12pm). Tuesday.
- April 30 (online 10am). Tuesday.
- May 2 (on campus 2-4pm). Tuesday.
- May 8 (online 10am). Wednesday.
- May 10 (on campus 3-5pm). Friday.

Where are we? Module organization.

✓ Introduction: Becoming a knowledge scientist.
✓ RDF-based knowledge graphs.
✓ OWL ontology language. Focus on modelling.
✓ SPARQL 1.0 Query Language.
✓ From tabular data to KG.
✓ RDFS Semantics and OWL 2 profiles.
✓ SPARQL 1.1, Rules and Graph Database solutions.
✓ Ontology Alignment.

9. **Ontology (KG) Embeddings and Machine Learning.** (Today)
10. (Large) Language Models and KGs. (Seminar)

The global picture



Coursework part 2

– Sunday, 12 May 2024, 5:00 PM
  - ✓ Tabular Data to Knowledge Graph: 40% (Weeks 2 and 5)
  - ✓ SPARQL and Reasoning: 20% (Weeks 4, 7 and 8)
  - ✓ Ontology Alignment: 10% (Week 9)
  - ✓ Ontology Embeddings: 10% (Week 10)

Hybrid Learning and Reasoning Systems

Motivation:

– Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.
- Limitations of KR systems: **maintenance** and **flexibility** in the inference. *e.g.*, Does $C(a)$ hold if?
  - A `and` (R `some` B) `subClassOf` C. $A(a)$, $B'(b)$, and $R(a, b)$.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Motivation:

- Need of **richer AI** systems, *i.e.*, **semantically sound**, **explainable**, and **reliable**.
- Impressive results in Deep Learning but require **large datasets** and **lack explanation**.
- Limitations of KR systems: **maintenance** and **flexibility** in the inference. *e.g.*, Does $C(a)$ hold if?
  - A `and` (R `some` B) `subClassOf` C. $A(a)$, $B'(b)$, and $R(a, b)$.
- **Solution?** Hybrid Learning and Reasoning Systems.

Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. CoRR abs/2002.06177 (2020)

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods

† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). https://arxiv.org/abs/2102.11965

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods
- Overview of **patterns** for hybrid systems. †

† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). `https://arxiv.org/abs/2102.11965`

Hybrid Learning and Reasoning Systems

- Unification of:
  - **statistical** (data-driven) and
  - **symbolic** (knowledge-driven) methods
- Overview of **patterns** for hybrid systems. †
  - Focus on **Ontology** (knowledge graph) **embeddings** as a component for a hybrid system (*e.g.*, OWL2Vec*).

† Michael van Bekkum et al. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. International Journal of Applied Intelligence (2021). `https://arxiv.org/abs/2102.11965`

Generic Patterns for Hybrid Systems

Focus on Knowledge Graph Embeddings

Introduction: Models

– **Models** are descriptions of entities and their relationships.

Introduction: Models

- **Models** are descriptions of entities and their relationships.
- **Statistical models** represent dependencies between statistical variables.
  - *e.g.*, Examples are (Deep) Neural Networks, Bayesian Networks and Markov Models.

Introduction: Models

- **Models** are descriptions of entities and their relationships.
- **Statistical models** represent dependencies between statistical variables.
  - *e.g.*, Examples are (Deep) Neural Networks, Bayesian Networks and Markov Models.
- **Semantic models** represent the implicit meaning and relationships of symbols.
  - *e.g.*, Ontologies and Knowledge Graphs, Rule-based models.

Introduction: Models

- **Models** are descriptions of entities and their relationships.
- **Statistical models** represent dependencies between statistical variables.
  - *e.g.*, Examples are (Deep) Neural Networks, Bayesian Networks and Markov Models.
- **Semantic models** represent the implicit meaning and relationships of symbols.
  - *e.g.*, Ontologies and Knowledge Graphs, Rule-based models.
- **Hybrid models** combine both.

Types of logical inference

– **Deductive inference**

$$\frac{\text{All swans are birds} \quad \text{Daffy is a swan}}{\text{Therefore, Daffy is a bird}}$$

Types of logical inference

– **Deductive inference**

All swans are birds
Daffy is a swan

Therefore, Daffy is a bird

– **Inductive inference**

Daffy is a swan and white
Tweety is a swan and white

Therefore, all swans are white

Types of logical inference

– **Deductive inference**

All swans are birds
Daffy is a swan

Therefore, Daffy is a bird

– **Inductive inference**

Daffy is a swan and white
Tweety is a swan and white

Therefore, all swans are white

– **Abductive inference**

All swans are white
Daffy is white

Therefore, Daffy is a swan

Types of logical inference

– **Deductive inference** (this module)

All swans are birds
Daffy is a swan

Therefore, Daffy is a bird

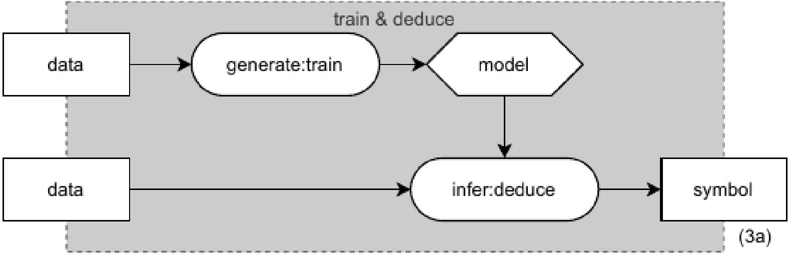
– **Inductive inference** (a bit today)

Daffy is a swan and white
Tweety is a swan and white

Therefore, all swans are white

Ross King. **Automatic Science using Robot Scientists**. `https://www.youtube.com/watch?v=Bj1TwqdLBs0`

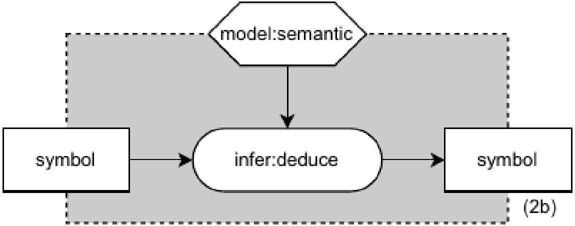
Machine learning pattern (non hybrid)



For example, image classification as in the `http://www.image-net.org/` challenge (symbol = label from WordNet).

infer:deduce $\rightarrow$ inductive inference

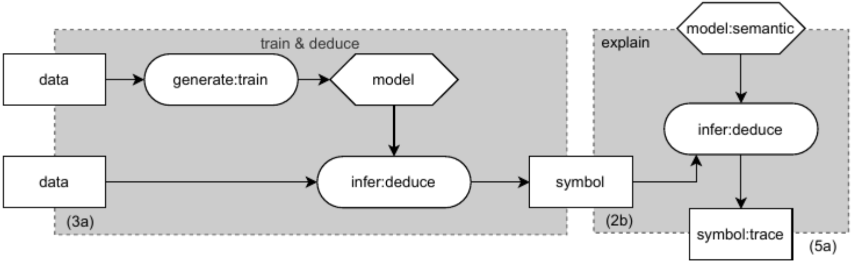
Semantic model pattern (non hybrid)



- Standard reasoning (*e.g.*, classification, class membership).
  - infer:deduce $\rightarrow$ deductive inference
- Rule-mining and ontoloy learning based on symbolic data (*i.e.*, ABox).
  - infer:deduce $\rightarrow$ inductive inference

Hybrid patterns

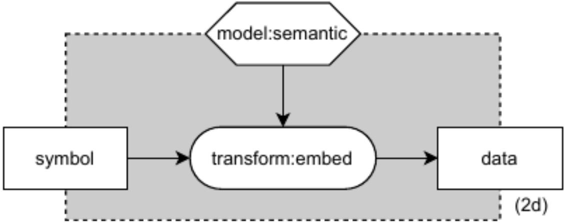
Explainability



The semantic model explains/interprets the prediction. And possibly validates the prediction too.

Human-driven FOL explanations of deep learning. IJCAI 2020
Knowledge graphs as tools for explainable machine learning: A survey. Artificial Intelligence 2022

Knowledge graph embeddings



Symbols are transformed into vectors (*e.g.*, OWL2Vec)

Knowledge Graph Embedding: A Survey of Approaches and Applications. TKDE 2017
OWL2Vec*: Embedding of OWL Ontologies. Machine Learning journal (2021)

Embeddings (definition)

- An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.

https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Embeddings (definition)

- An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.
- One-hot encoding also assign a vector to categories but...
  ✗ has as many dimensions as categories.
  ✗ vectors are not related to each other.

`https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526`

Embeddings (definition)

- An **embedding** is a function that maps a discrete — categorical — variable (*e.g.*, a KG entity/symbol) to a **vector of numbers**.
- One-hot encoding also assign a vector to categories but...
  ✗ has as many dimensions as categories.
  ✗ vectors are not related to each other.
- Embeddings aim at creating **meaningful low-dimensional continuous vectors**.
  - dbr:london $\rightarrow [0.5 \;\; 0.3 \;\; 0.8 \;\; 1.0 \;\; 0.0 \; ...]$

https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

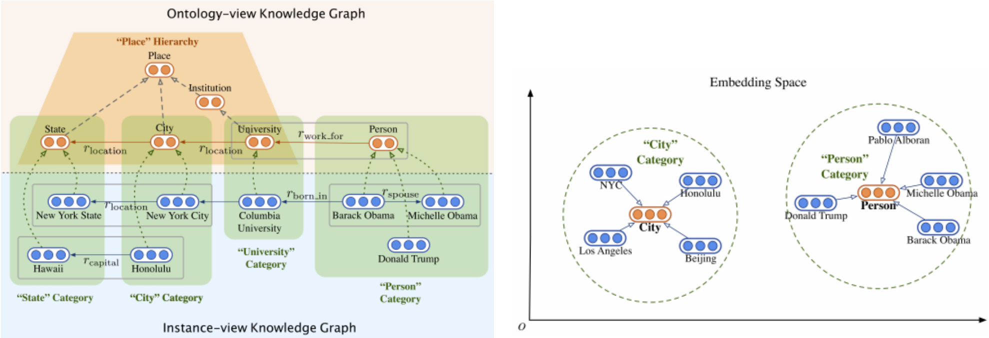
Principal Component-Analysis (PCA)

- An unsupervised dimensionality reduction algorithm.
- Useful....
  - as a tool for visualization (*e.g.*, vector representation in 2-D)
  - for noise filtering
  - for feature extraction and engineering

Figure: From 64 dimensions to 2

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

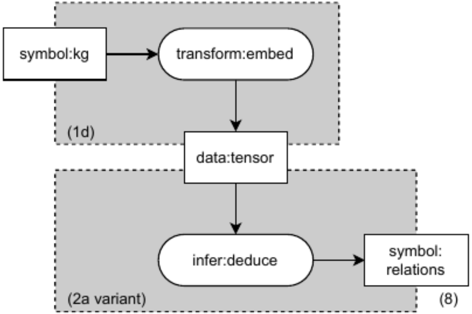
Knowledge graph embeddings (example)



KG Embedding Systems exploit the neighbourhood of an entity to calculate its vector.

Example from: Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. KDD 2019.

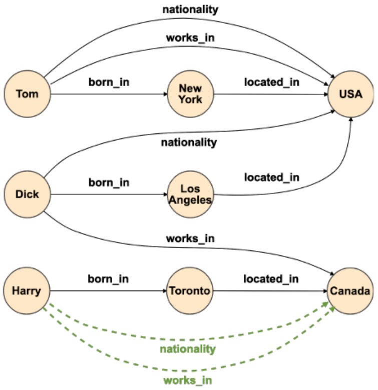
Knowledge Graph Embeddings: Link prediction



– Plausability of a triple `<subject predicate object>` given a scoring function.

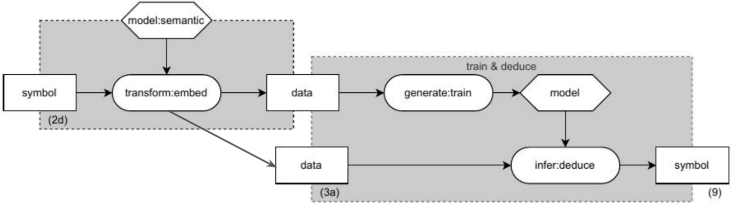
Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. TKDD 2021

Knowledge Graph Embeddings: Link prediction (example)



Example from: Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. TKDD 2021

Learning with (knowledge) embeddings

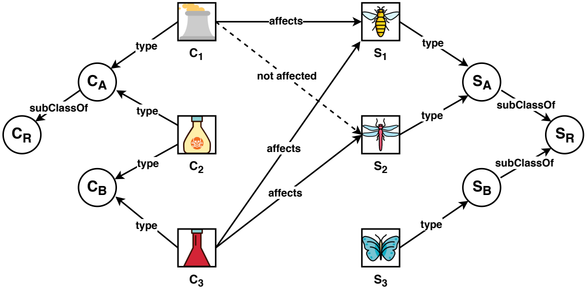


- Applying knowledge graph embeddings in a subsequent classification step.
- Graph Neural Networks over the KG structure.
- Key for zero-shot learning approaches

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.
A Comprehensive Survey on Graph Neural Network. IEEE Transactions on Neural Networks and Learning Systems 2019.
Knowledge-aware Zero-Shot Learning: Survey and Perspective. arXiv:2103.00070. 2021

Learning with (knowledge) embeddings (example)

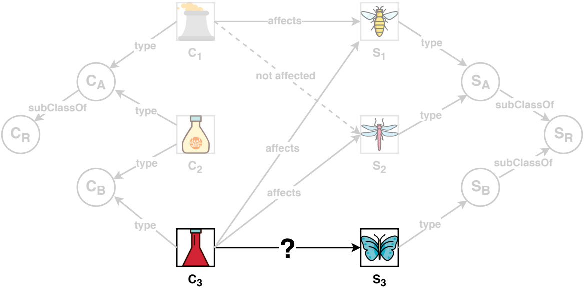
- **Prediction of adverse biological effects** of chemicals via KG embeddings.



Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

Learning with (knowledge) embeddings (example)

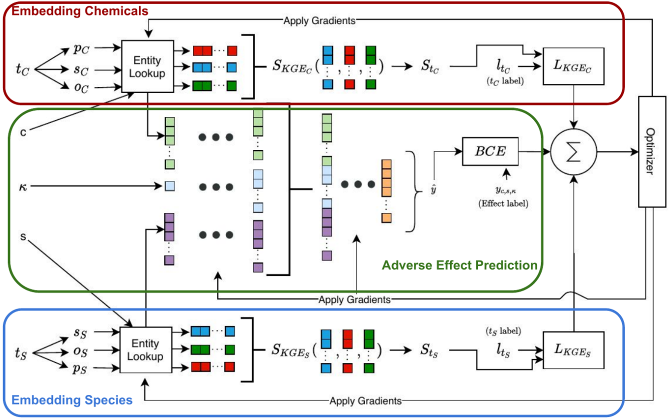
– **Prediction of adverse biological effects** of chemicals via KG embeddings.



KGE are critical for unseen chemicals and species. Also useful for explainability.

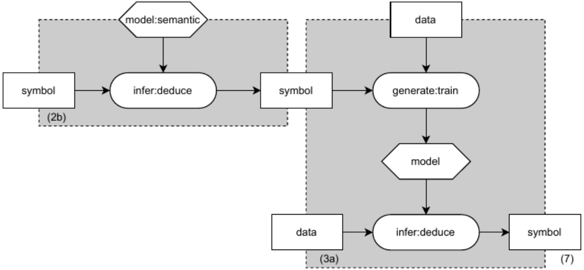
Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

Learning with (knowledge) embeddings (example)



Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Sem Web. 2022.

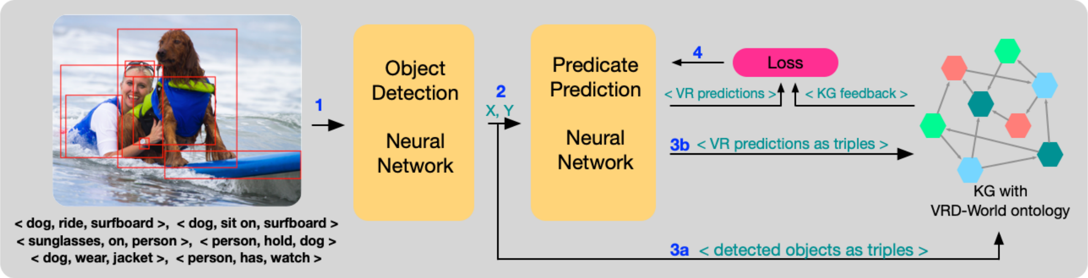
Learning with prior knowledge (i)



- Domain knowledge (*e.g.*, a KG) used to constraint search space during training.
- **Semantic loss function**: impact of the violation of the symbolic knowledge.

A semantic loss function for deep learning with symbolic knowledge. ICML 2018
Logic Tensor Networks. `https://github.com/logictensornetworks/logictensornetworks`

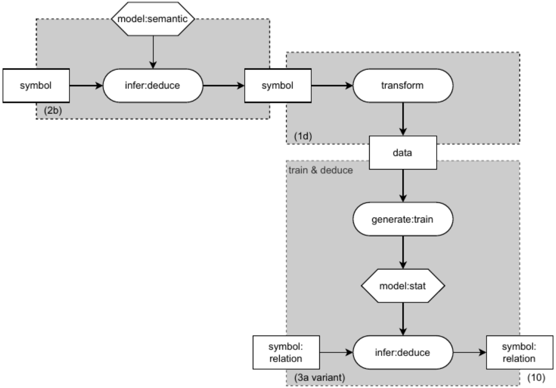
Learning with prior knowledge (ii)



– Penalisation of predictions that violate constrainst in the KG.

D. Herron et al. On the Potential of Logic and Reasoning in Neurosymbolic Systems using OWL-based Knowledge Graphs. Under review, 2024.

Learning to reason

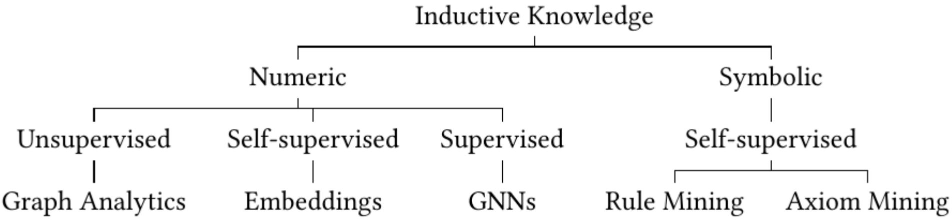


Ontology Reasoning with Deep Neural Networks. JAIR 2021

Inductive Techniques for Knowledge Graphs

Focus on Knowledge Graph Embeddings

Inductive techniques for knowledge graphs



- Input: A Knowledge Graph (symbolic)
- Representation generated: Numeric or Symbolic

Knowledge Graphs. `https://kgbook.org/`. 2021

Graph analytics (unsupervised)

Exploit techniques from **graph theory** and **network analysis**. *e.g.*,:

- **Centrality**: the most important nodes (*i.e.*, concepts, instances) or edges (*i.e.*, properties) of a graph.
- **Community detection**: subgraphs that are densely connected.
- **Connectivity**: how well-connected are the nodes of a graph to identify isolated nodes or subgraphs.

Graph analytics (unsupervised)

Exploit techniques from **graph theory** and **network analysis**. *e.g.*,:

- **Centrality**: the most important nodes (*i.e.*, concepts, instances) or edges (*i.e.*, properties) of a graph.
- **Community detection**: subgraphs that are densely connected.
- **Connectivity**: how well-connected are the nodes of a graph to identify isolated nodes or subgraphs.
- **Path finding**: all possible paths between two nodes.
- **Node similarity**: based on their connection to other nodes (*e.g.*,random-walks techniques).

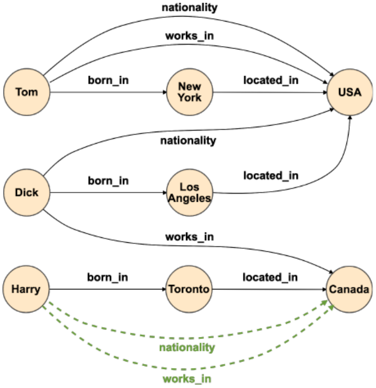
Graph Neural Networks (supervised)

- Machine learning models for **graph-structured data**.
- The **neural network** is **based on the shape and connections** of the (knowledge) graph.
- **End-to-end supervised learning** (*e.g.*, classification).
- Can be used to classify nodes or the graph itself.

A Comprehensive Survey on Graph Neural Network. IEEE Transactions on Neural Networks and Learning Systems 2019.

Symbolic learning (self-supervised)

– Identifies patterns in the data



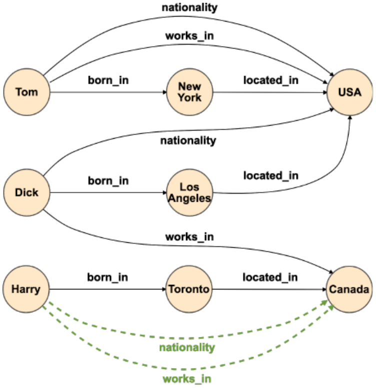
Symbolic learning (self-supervised)

– Identifies patterns in the data

– Learn hypotheses in a symbolic (logical) language. For example:

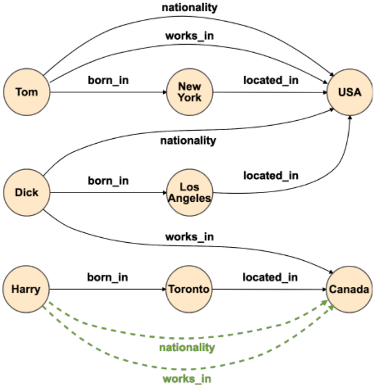
  – As a rule: `nationality(x,z) :- born_in(x,y) ∧ located_in(y,z)`

  – As an OWL 2 axiom: `born_in ∘ located_in SubPropertyOf: nationality`

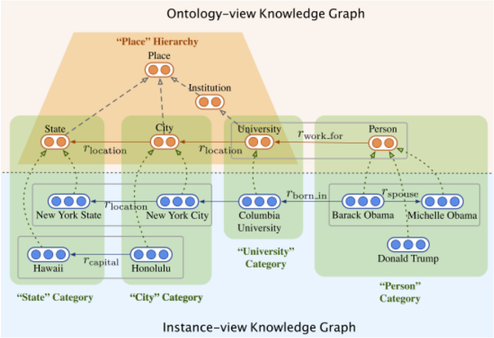


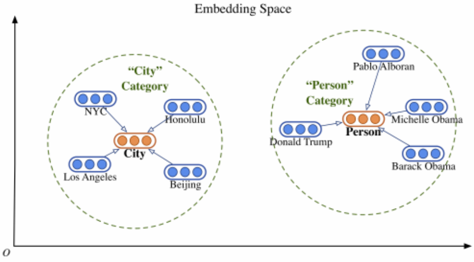
Symbolic learning (self-supervised)

- Identifies patterns in the data
- Learn hypotheses in a symbolic (logical) language. For example:
  - As a rule: `nationality(x,z) :- born_in(x,y) ∧ located_in(y,z)`
  - As an OWL 2 axiom: `born_in ∘ located_in SubPropertyOf: nationality`
- Can help explaining/interpret (link) predictions: *e.g.*, why `nationality(Harry, Canada)`?



Knowledge graph embeddings (self-supervised)





Example from: Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. KDD 2019.

Knowledge graph embeddings (self-supervised)

KGE approaches (excluding those based on language models) typically:

- Receive as input a set of **positive** (the ones in the KG) and **negative triples**.
- Include a **scoring function** that accepts as input the embedding of the elements of a triple (there is an initialization step).

Knowledge graph embeddings (self-supervised)

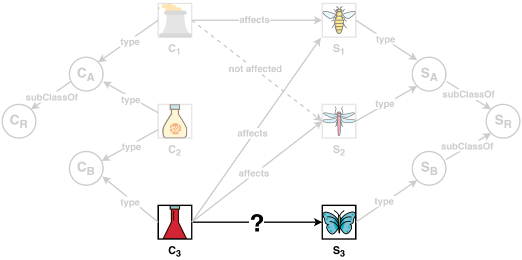
KGE approaches (excluding those based on language models) typically:

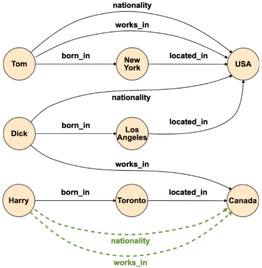
- Receive as input a set of **positive** (the ones in the KG) and **negative triples**.
- Include a **scoring function** that accepts as input the embedding of the elements of a triple (there is an initialization step).
- Learn embeddings so that the score for positive triples is maximized while the score for negative triples is minimized (*i.e.*, **loss function**).
- Compute **similar vectors** for similar nodes (*i.e.*, concepts/instances) and edges (*i.e.*, properties).

Knowledge graph embeddings (applications)

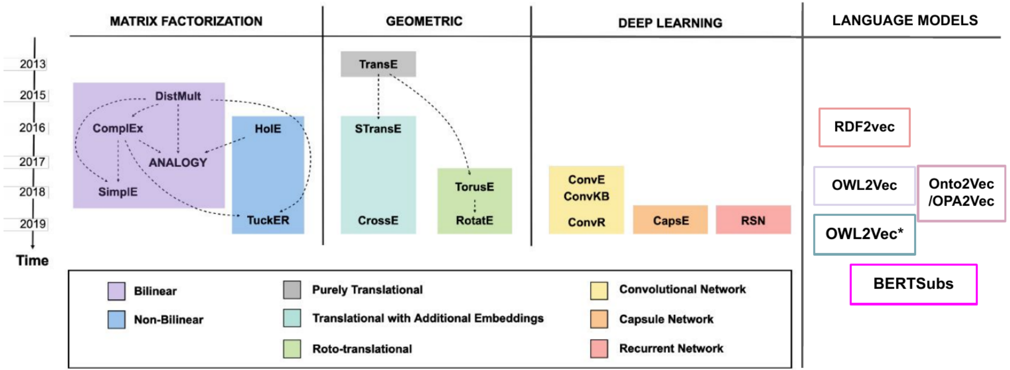
– The computed embedding can be used in a **downstream machine learning task** (*e.g.*, prediction of adverse effect chemical-species).

– The scoring function can be used to evaluate the plausibility of a triple for **link prediction** or **KG completion**.





Knowledge graph embeddings (overview of approaches)



Incomplete list of approaches, adapted from: Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. TKDD 2021

Knowledge graph embeddings (overview of approaches)

– **Translational models**: translate subject entities to object entities via the predicate/relation in the low-dimensional space.

Knowledge graph embeddings (overview of approaches)

- **Translational models**: translate subject entities to object entities via the predicate/relation in the low-dimensional space.
- **Tensor decomposition models**: represent the KG into a one-hot 3-order tensor and apply matrix factorisation/decomposition models to learn entity vectors.

Knowledge graph embeddings (overview of approaches)

- **Translational models**: translate subject entities to object entities via the predicate/relation in the low-dimensional space.
- **Tensor decomposition models**: represent the KG into a one-hot 3-order tensor and apply matrix factorisation/decomposition models to learn entity vectors.
- **Neural models**: unlike previous models they learn embeddings with non-linear scoring functions via a neural model.

Knowledge graph embeddings (overview of approaches)

- **Translational models**: translate subject entities to object entities via the predicate/relation in the low-dimensional space.
- **Tensor decomposition models**: represent the KG into a one-hot 3-order tensor and apply matrix factorisation/decomposition models to learn entity vectors.
- **Neural models**: unlike previous models they learn embeddings with non-linear scoring functions via a neural model.
- **Language models**: perform random walks over the KG to create a document of sentences and leverage existing language models (*e.g.*, word embedding) to learn vectors for each KG entity.

Knowledge graph embeddings (implementations)

- PyKEEN (Python KnowlEdge EmbeddiNgs) with PyTorch: `https://pykeen.github.io/`
- Open Knowledge Embedding implemented with PyTorch: `https://github.com/thunlp/OpenKE`
- Knowledge Embedding implemented with Keras: `https://github.com/NIVA-Knowledge-Graph/KGE-Keras`
- jRDF2Vec: `https://github.com/dwslab/jRDF2Vec`
- pyRDF2Vec: `https://github.com/IBCNServices/pyRDF2Vec`
- OWL2Vec* (python): `https://github.com/KRR-Oxford/OWL2Vec-Star`

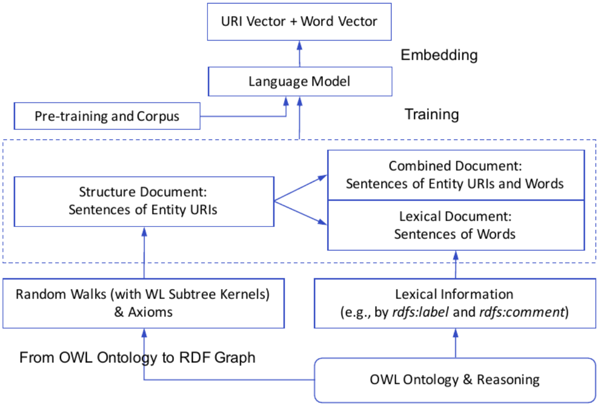
Knowledge graph embeddings (pre-trained)

- OpenKE:
  `http://139.129.163.161/index/toolkits#pretrained-embeddings`
- Drug-drug interaction:
  `https://github.com/rcelebi/GraphEmbedding4DDI/`
- Universal Knowledge Graph Embeddings: `https://embeddings.cc/`

Embedding ontologies with OWL2Vec*

OWL2Vec* Overview

- **projects** the ontology into a graph,
- **walks** the graph,
- creates a **corpus of sentences** according to the walking strategies, and
- generates **embeddings** from that corpus.



OWL2Vec*: Embedding of OWL Ontologies. Machine Learning journal 2021.

OWL2Vec*: ontology projection

Approximation of an OWL 2 ontology into an RDF graph.

Axiom of Condition 1	Axiom or Triple(s) of Condition 2	Projected Triple(s)
$A \sqsubseteq \square r.D$ or $\square r.D \sqsubseteq A$	$D \equiv B \mid B_1 \sqcup \ldots \sqcup B_n \mid B_1 \sqcap \ldots \sqcap B_n$	$\langle A, r, B \rangle$ or $\langle A, r, B_i \rangle$ for $i \in 1, \ldots, n$
$\exists r.\top \sqsubseteq A$ (domain)	$\top \sqsubseteq \forall r.B$ (range)	
$A \sqsubseteq \exists r.\{b\}$	$B(b)$	
$r \sqsubseteq r'$	$\langle A, r', B \rangle$ 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, $\top$ is the top concept.

OWL2Vec*: sentence generation via random walks

Strategies:

- Random walks
- Weisfeiler Lehman (WL) kernel, which assign identifiers to subgraphs and includes them into the walk.

Structure Document Sentences

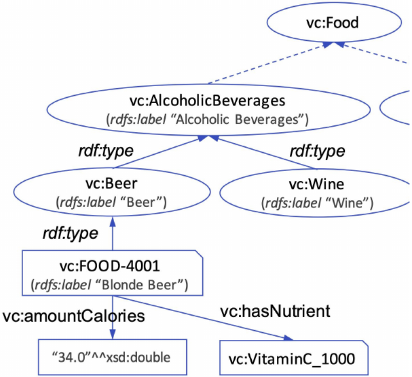
(vc:Beer, rdf:type, vc:FOOD-4001, vc:hasNutrient, vc:VitaminC_1000)

Lexical Document Sentences

("beer", "type", "blonde", "beer", "has", "nutrient", "vitamin", "c")

Combined Document Sentences

(vc:FOOD-4001, "has", "nutrient", "vitamin", "c")
OR
("blonde", "beer", "has", "nutrient", vc:VitaminC_1000)



OWL2Vec*: language model and embeddings

- OWL2Vec* relies on the **Word2vec** as neural **language model**.
- Word2vec learns **embeddings** for all the elements in the documents (*i.e.*, both **words** and **URIs**)

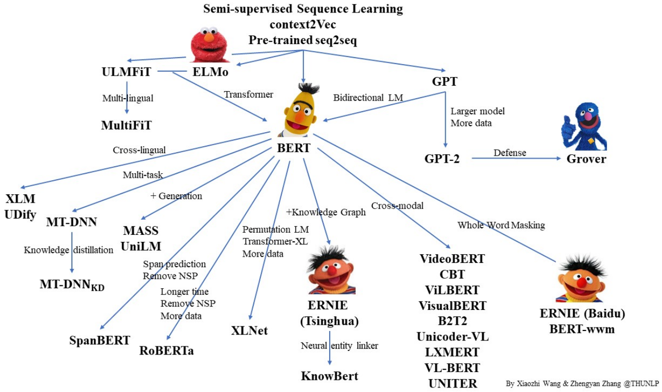
OWL2Vec*: language model and embeddings

- OWL2Vec* relies on the **Word2vec** as neural **language model**.
- Word2vec learns **embeddings** for all the elements in the documents (*i.e.*, both **words** and **URIs**)
- The embeddings of the ontology entities can be calculated via their **URI embedding** or via the **word embeddings** of their labels.
  - The URI `vc:FOOD-4001` (Blonde Beer) has a vector.
  - As well as the words “`blonde`” and “`beer`”.

OWL2Vec*: language model (present and future)

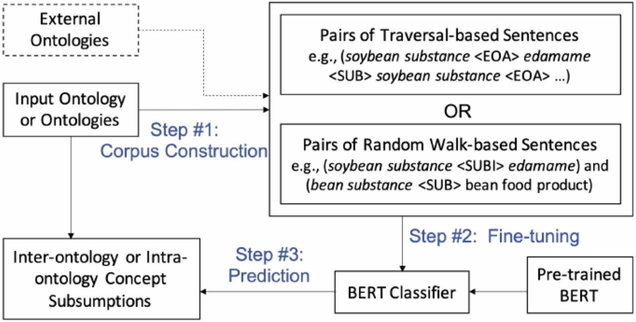
Other language models could be used in OWL2Vec*:

`https://github.com/thunlp/PLMpapers`



Ontology embeddings beyond OWL2Vec*

BERTSubs: contextual semantic embeddings

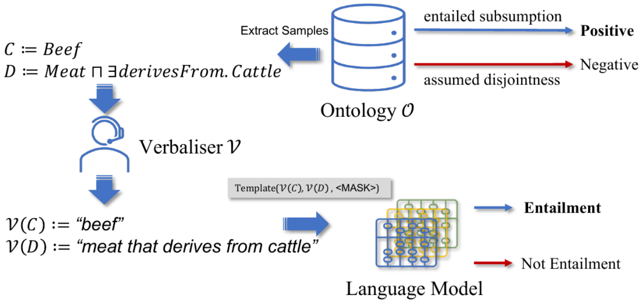


(*) Considering the sentence "the bank robber was seen on the river bank"; unlike word2vec, BERT computes different embeddings for the two occurrences of "bank" which have different meanings.

J. Chen et al. Contextual Semantic Embeddings for Ontology Subsumption Prediction. World Wide Web Journal 2023

OntoLAMA: Language Model Analysis for Ontologies

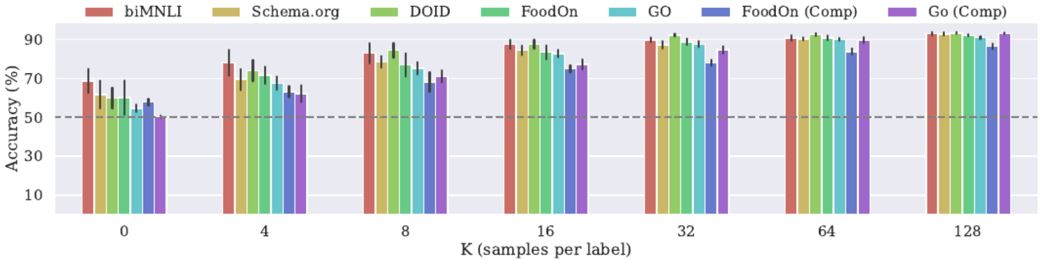
- To what extent LMs infer ontology semantics?
- **Prompt-based Inference** using RoBERTa in a K-shot setting.



Y. He et al. Language Model Analysis for Ontology Subsumption Inference. ACL Findings 2023.
https://arxiv.org/abs/2302.06761

OntoLAMA: Language Model Analysis for Ontologies

- To what extent LMs infer ontology semantics?
- **Prompt-based Inference** using RoBERTa in a K-shot setting.



Y. He et al. Language Model Analysis for Ontology Subsumption Inference. ACL Findings 2023.
`https://arxiv.org/abs/2302.06761`

Ontology Embeddings applications

Ontology Embedding applications

- **Class subsumption** and **class membership predictions** (as in OWL2Vec* paper and ♠)

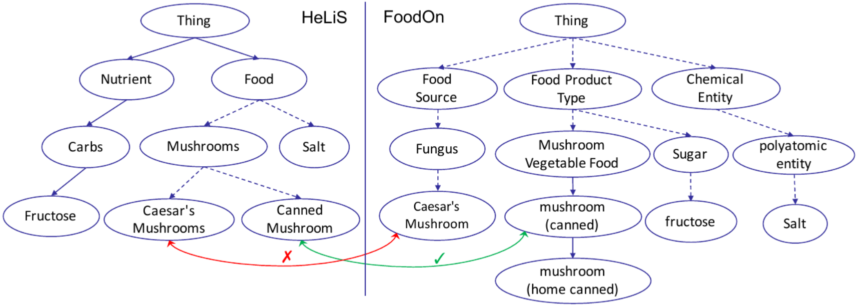
- Embedding of chemicals and species to **predict adverse effects**.

- **Ontology alignment** (Samsung UK project with food ontologies) †

- **Ontology clustering** in life sciences ontologies to be applied in an Information Retrieval task. ‡

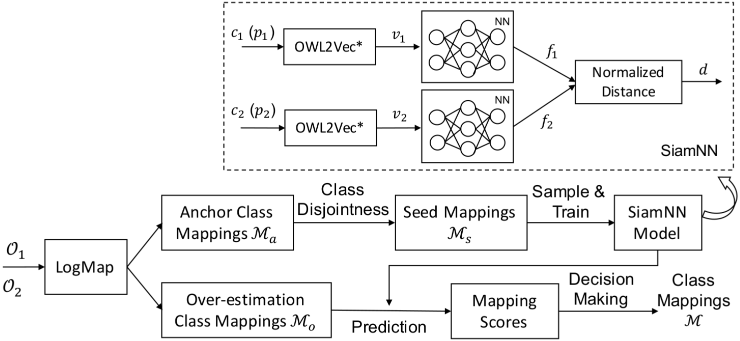
† J. Chen et al. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. ESWC 2021
‡ A. Ritchie et al. Ontology Clustering with OWL2Vec*. DeepOntonNLP ESWC Workshop 2021.
♠ J. Chen et al. Contextual Semantic Embeddings for Ontology Subsumption Prediction. World Wide Web Journal 2023

Ontology Embeddings applications: alignment (Samsung Research UK)



J. Chen et al. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. ESWC 2021

Ontology Embeddings applications: alignment (Samsung Research UK)



J. Chen et al. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. ESWC 2021

Ontology Embeddings applications: clustering

Embedding of the Gene Ontology and its 3 branches: biological process, cellular component, and molecular function

A. Ritchie et al. Ontology Clustering with OWL2Vec*. DeepOntonNLP ESWC Workshop 2021.

Acknowledgements

– OWL2Vec* developers and collaborators: Jiaoyan Chen, Pan Hu, Ole Magnus Holter, Denvar Antonyrajah, and Ian Horrocks.

– Special acknowledgement:
  – **Jiaoyan Chen**, University of Manchester
  – **Yuan He** and **Hang Dong**, University of Oxford
  – **Ole Magnus Holter**, University of Oslo

Laboratory Session

OWL2Vec* in practice

- Execute OWL2Vec* over the `Pizza` and `FoodOn` ontologies.
- Compute similarity among words and entities.
- Perform clustering and visualize results.