CITY
UNIVERSITY OF LONDON
EST 1894

# 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.

– Access via e-mail from `evaluations@city.ac.uk` or from MyMoodle

– More information: `https://studenthub.city.ac.uk/student-administration/online-module-evaluation-at-city`

# Additional (potential) MSc Projects

DYAD: `https://www.dyad.net/`

– Helathcare domain.

– KG and graph machine learning.

# Invited talks next week (15+5 min.)

- **Valentina Carapella**
  - Data Scientist at Perspectum (`https://perspectum.com/`)
  - *“KG Use Cases from Medical Imaging Science”*

# Invited talks next week (15+5 min.)

- **Valentina Carapella**
  - Data Scientist at Perspectum (`https://perspectum.com/`)
  - *"KG Use Cases from Medical Imaging Science"*
- **Alexandra Shatova**
  - MSc Data Science at City. Internship at DYAD (`https://www.dyad.net/`).
  - *"Automated KG construction in the medical domain"*

# Invited talks next week (15+5 min.)

- **Valentina Carapella**
  - Data Scientist at Perspectum (`https://perspectum.com/`)
  - *“KG Use Cases from Medical Imaging Science”*
- **Alexandra Shatova**
  - MSc Data Science at City. Internship at DYAD (`https://www.dyad.net/`).
  - *“Automated KG construction in the medical domain”*
- **Vicenzo Cutrona**
  - PhD Student Università degli Studi di Milano - Bicocca.
  - R&D OpenLab @ Corvallis SRL (`https://corvallis.it/`)
  - *“Why semantic table understanding matters! Practical solutions to real-life problems”*

## Where are we?

- ✓ Introduction.
- ✓ RDF-based knowledge graphs.
- ✓ SPARQL 1.0
- ✓ RDFS Semantics and RDF(S)-based knowledge graphs.
- ✓ OWL (2) ontology language. Focus on modelling.
- ✓ Application to Data Science.
- ✓ OWL 2 Profiles, SPARQL 1.1 and Entailment Regimes
- ✓ Ontology (Knowledge Graph) Alignment

# Where are we?

✓ Introduction.

✓ RDF-based knowledge graphs.

✓ SPARQL 1.0

✓ RDFS Semantics and RDF(S)-based knowledge graphs.

✓ OWL (2) ontology language. Focus on modelling.

✓ Application to Data Science.

✓ OWL 2 Profiles, SPARQL 1.1 and Entailment Regimes

✓ Ontology (Knowledge Graph) Alignment

9. **Ontology (Knowledge Graph) Embeddings (today)**.

## Where are we?

✓ Introduction.
✓ RDF-based knowledge graphs.
✓ SPARQL 1.0
✓ RDFS Semantics and RDF(S)-based knowledge graphs.
✓ OWL (2) ontology language. Focus on modelling.
✓ Application to Data Science.
✓ OWL 2 Profiles, SPARQL 1.1 and Entailment Regimes
✓ Ontology (Knowledge Graph) Alignment

9. **Ontology (Knowledge Graph) Embeddings (today)**.
10. Graph Database Solutions and Invited Talks (March 31).

# 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. CoRR abs/2102.11965. Under review (2021)

# 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. CoRR abs/2102.11965. Under review (2021)

# 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 an 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. CoRR abs/2102.11965. Under review (2021)

# 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.

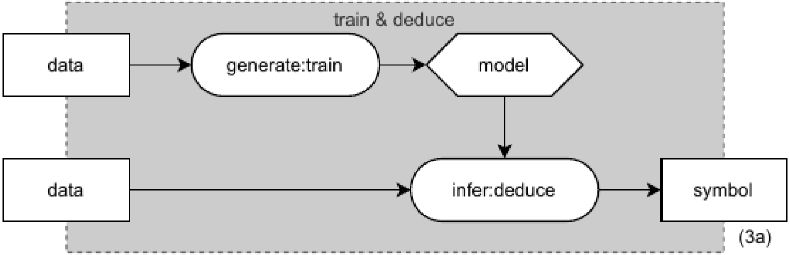
# Introduction: Models

- **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, Knowledge Graphs, Rule-based models.

## Introduction: Models

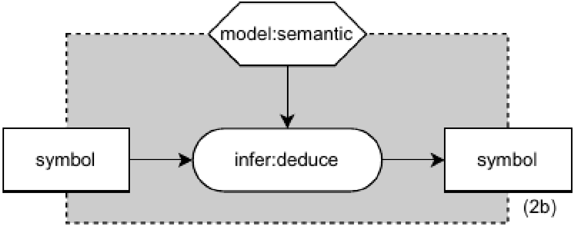
- **Models** are descriptions of entities and their relationships.
- **Statistical models** represent dependencies between statistical variables.
  - *e.g.*, Examples are (Deep) Neural Networks, Bayesian Networks and Markov Models.
- **Semantic models** represent the implicit meaning and relationships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

# Machine learning pattern (non hybrid)



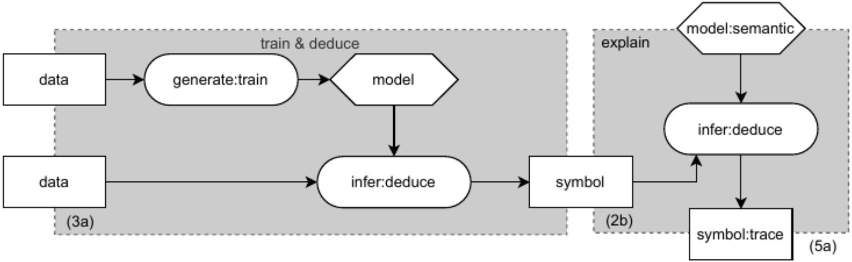

For example, image classification as in the `http://www.image-net.org/` challenge (symbol = label from WordNet).

# Semantic model pattern (non hybrid)




- Standard reasoning (*e.g.*, classification, class membership).
- Rule-mining and ontoloy learning based on symbolic data (*i.e.*, ABox).

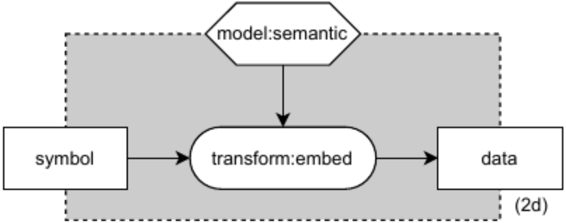
# Explainability




The semantic model explains/interprets the prediction.

Explaining Trained Neural Networks with Semantic Web Technologies: First Steps. NeSy 2017.
Human-driven FOL explanations of deep learning. IJCAI 2020

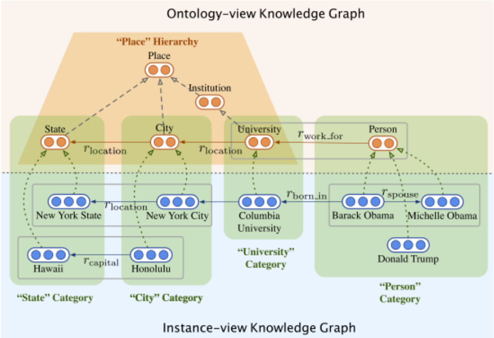
# Knowledge graph embeddings

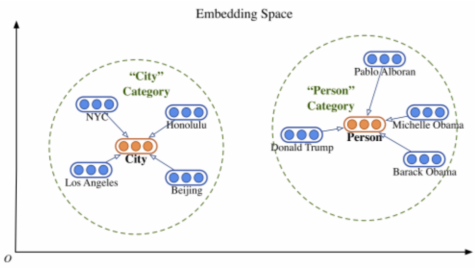



Symbols are transformed into vectors (*e.g.*, OWL2Vec)

Knowledge Graph Embedding: A Survey of Approaches and Applications. TKDE 2017
OWL2Vec*: Embedding of OWL Ontologies. CoRR abs/2009.14654 (2020)

# Knowledge graph embeddings (example)

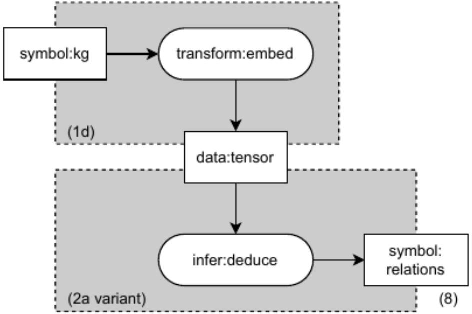



KG Embedding Systems exploit the neighbourhood of an entity to calculate its vector.

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

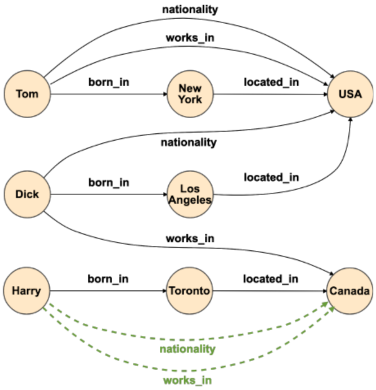
# Knowledge Graph Embeddings: Link prediction




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

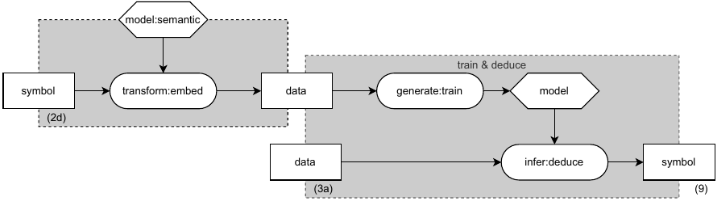
Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. TKDD 2021

# Knowledge Graph Embeddings: Link prediction (example)




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

# Learning with (knowledge) embeddings

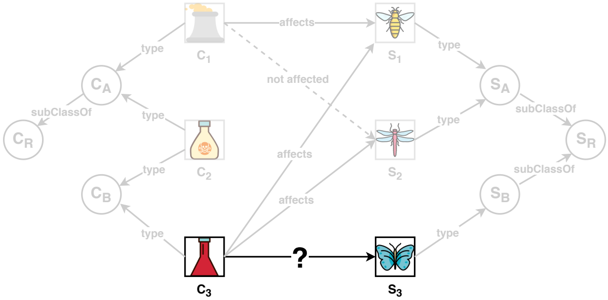



- Applying knowledge graph embeddings in a subsequent classification step.
- Graph Neural Networks over the KG structure.
- Key for zero-shot learning approaches

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Under review. 2021.
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

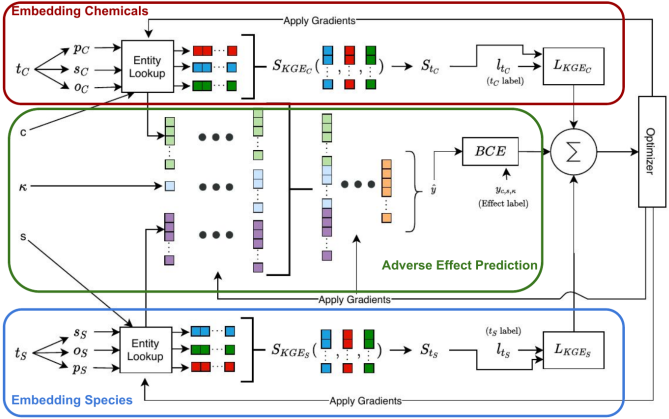
# Learning with (knowledge) embeddings (example)

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



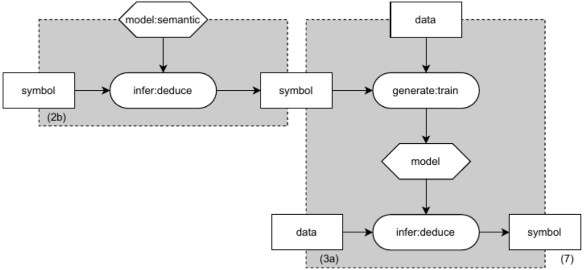

Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Under review. 2021.

# Learning with (knowledge) embeddings (example)




Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings. Under review. 2021.

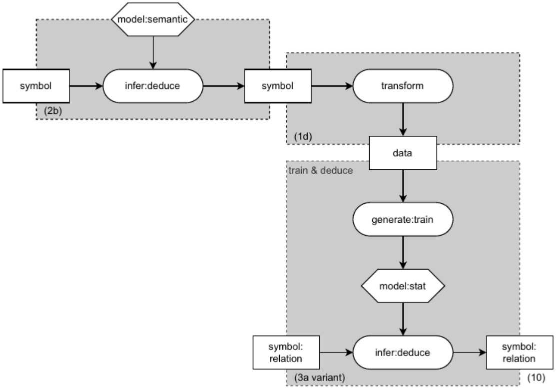
# Learning with prior kowledge




- 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 to reason

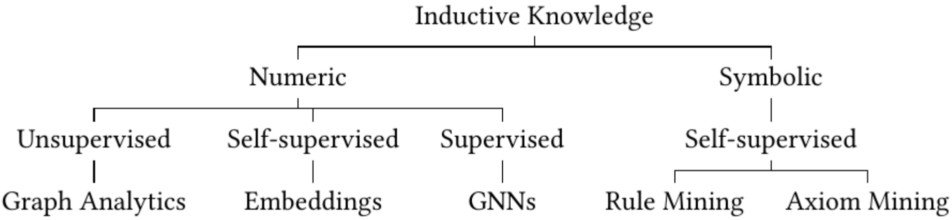



Ontology Reasoning with Deep Neural Networks. JAIR 2021

# Inductive Techniques for Knowledge Graphs

## Focus on Knowledge Graph Embeddings

# Inductive techniques for knowledge graphs




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

Knowledge Graphs. arXiv:2003.02320. 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).

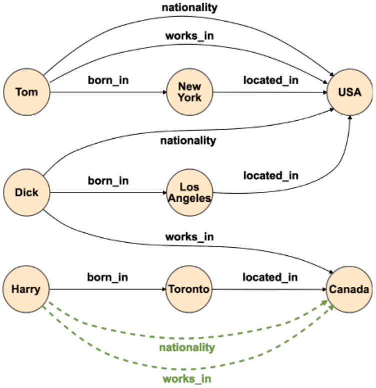
# Graph Neural Networks (supervised)

- 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.

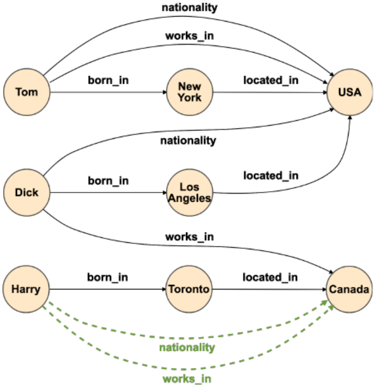
# Symbolic learning (self-supervised)

– Identifies patterns in the data



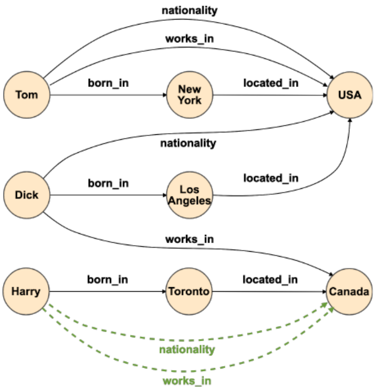
# Symbolic learning (self-supervised)

– 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`

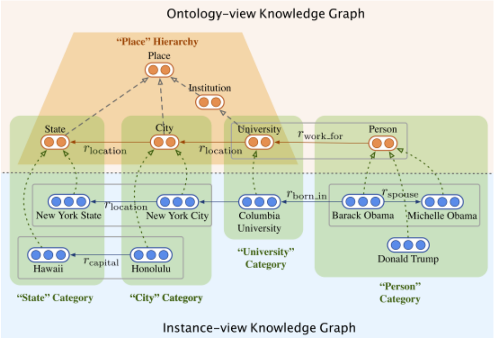


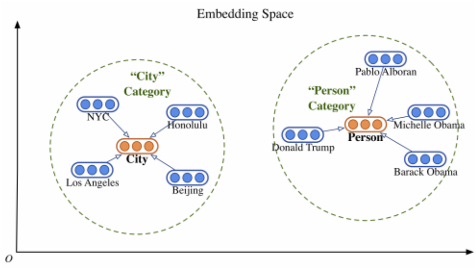
# Symbolic learning (self-supervised)

– 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)

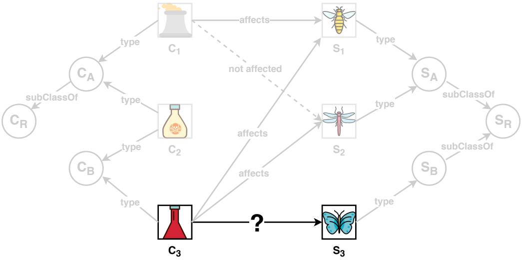
KGE approaches (excluding those based on language models) typically:

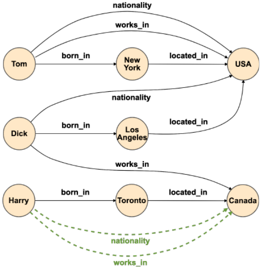
- Receive as input a set of **positive** (the ones in the KG) and **negative triples**.
- Include a **scoring function** that accepts as input the embedding of the elements of a triple (there is an initialization step).
- Learn embedding so that the score for positive triples is maximized while the score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

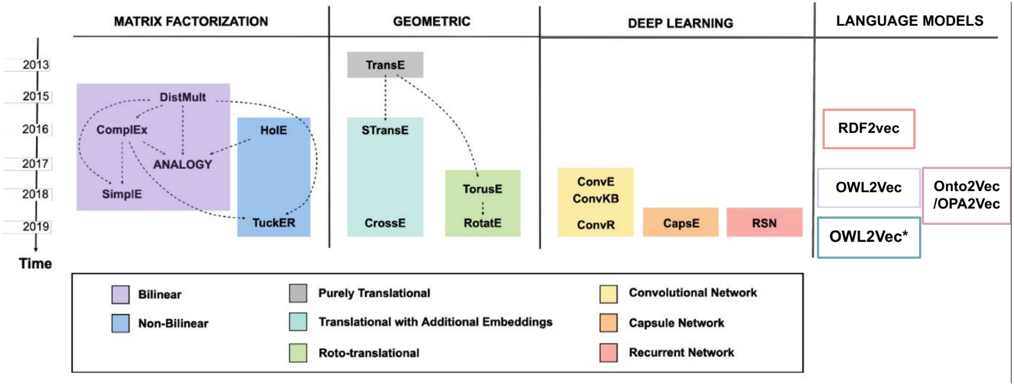
– The computed embedding can be used in a **downstream machine learning task** (*e.g.*, prediction of adverse effect chemical-species).

– 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`

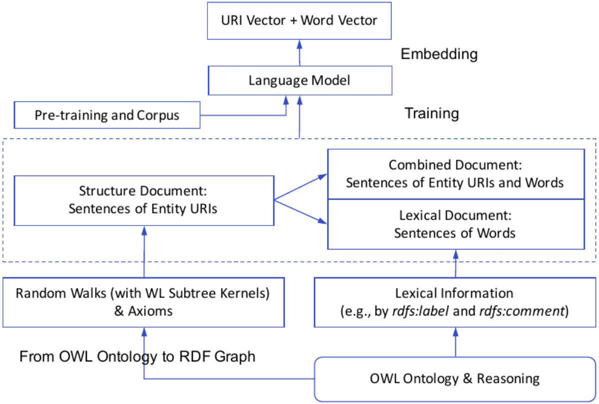
## Knowledge graph embeddings (pre-trained)

- OpenKE:
  http://139.129.163.161/index/toolkits#pretrained-embeddings
- KGvec2go: http://www.kgvec2go.org/
- Drug-drug interaction:
  https://github.com/rcelebi/GraphEmbedding4DDI/

# 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. CoRR abs/2009.14654 (2020)

# 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) |
|---|---|---|
| $A \sqsubseteq \Box r.D$ or $\Box r.D \sqsubseteq A$ | $D \equiv B \mid B_1 \sqcup \ldots \sqcup B_n \mid B_1 \sqcap \ldots \sqcap B_n$ | $\langle A, r, B \rangle$ 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 | |
| $B \sqsubseteq A$ | – | $\langle B, rdfs{:}subClassOf, A \rangle$ $\langle A, rdfs{:}subClassOf^-, B \rangle$ |
| $A(a)$ | – | $\langle a, rdf{:}type, A \rangle$ $\langle A, rdf{:}type^-, a \rangle$ |
| $r(a, b)$ | – | $\langle a, r, b \rangle$ |

$\Box$ 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**

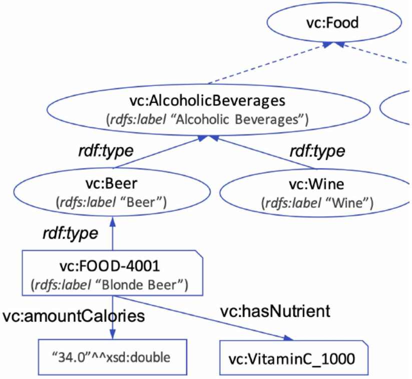
*(vc:Beer, rdf:type, vc:FOOD-4001, vc:hasNutrient, vc:VitaminC_1000)*

**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**)

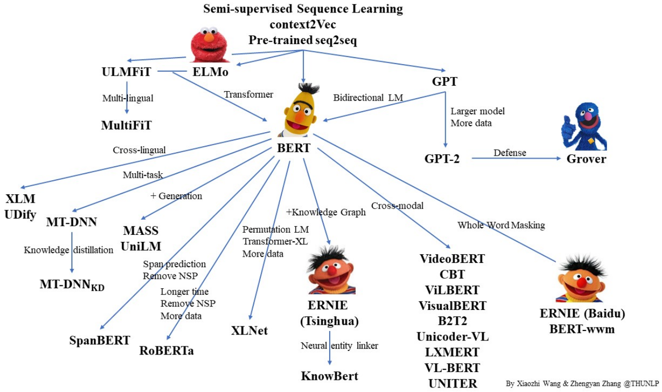
# OWL2Vec*: language model and embeddings

- 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 (future)

Other language models could be used in OWL2Vec*:

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



# OWL2Vec*: applications

- **Class subsumption** and **class membership predictions** (as in OWl2Vec* paper)
- 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*. Submitted 2021.

# OWL2Vec*: clustering

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


A. Ritchie. Ontology Clustering with OWL2Vec*. Submitted 2021.

# Acknowledgements

– OWL2Vec* developers and collaborators.

– Specially **Jiaoyan Chen**, University of Oxford

# Laboratory Session

## OWL2Vec* in practice

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