Neural-symbolic Knowledge Representation with Ontology and Knowledge Graph Embeddings

Jiaoyan Chen

Lecturer in Department of Computer Science, University of Manchester

Part-time Senior Researcher in University of Oxford

Seminar in KAUST

What is an ontology?

Knowledge representation of a domain (e.g., concepts/classes, instances/entities, properties, and logical relationships)

$$\begin{aligned}\mathcal{T} = \{&\text{Father} \sqsubseteq \text{Parent} \sqcap \text{Male},\ \text{Mother} \sqsubseteq \text{Parent} \sqcap \text{Female},\\ &\text{Child} \sqsubseteq \exists \text{hasParent.Father},\ \text{Child} \sqsubseteq \exists \text{hasParent.Mother},\\ &\text{hasParent} \sqsubseteq \text{relatedTo}\}\\ \mathcal{A} = \{&\text{Father}(\text{Alex}), \text{Child}(\text{Bob}), \text{hasParent}(\text{Bob}, \text{Alex})\}\end{aligned}$$

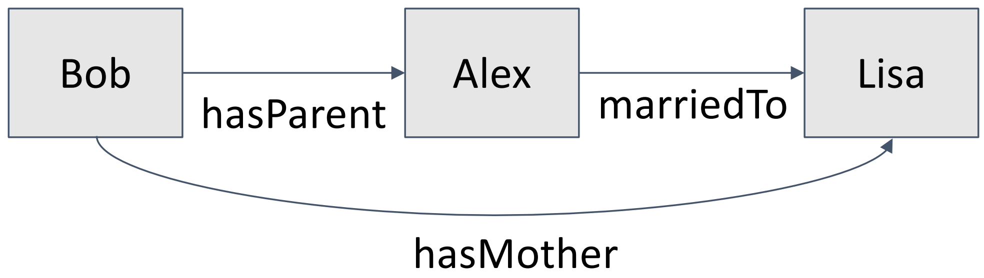
A toy ontology on a family

- Formal
- Explicit
- Shared

What languages for defining formal, explicit and shared ontologies?

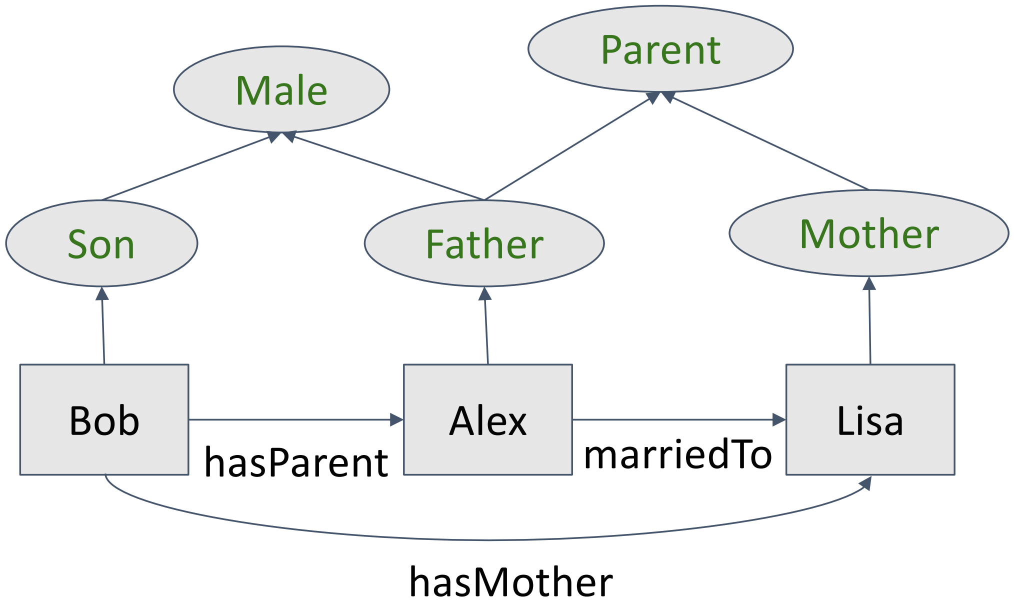
Ontology Languages

- **RDF** (Resource Description Framework)
  - Triple: <Subject, Predicate, Object>
  - Representing facts:
    - E.g., <Bob, hasParent, Alex>



Ontology Languages

- **RDF Schema (RDFS)**
  - Meta data (schema) of instances and facts
    - E.g., hierarchical concepts and properties, property domain and range,

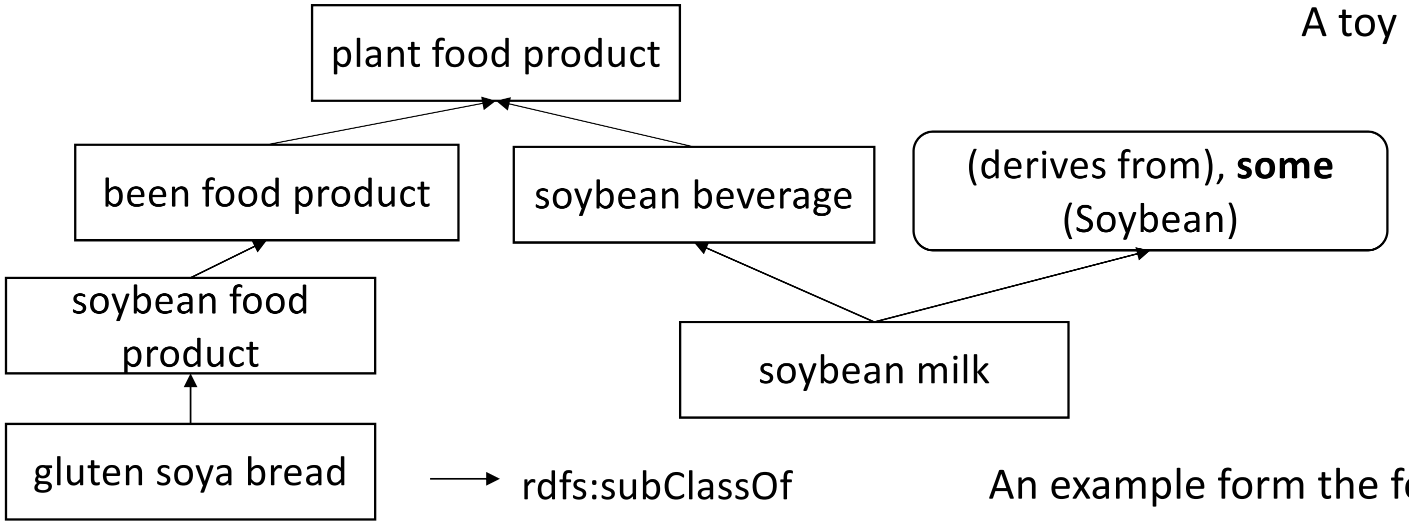


Ontology Languages

- **Web Ontology Language** (OWL)
  - Schema and logical relationships (domain knowledge)
  - Taxonomies and vocabularies

$$\mathcal{T} = \{\text{Father} \sqsubseteq \text{Parent} \sqcap \text{Male},\ \text{Mother} \sqsubseteq \text{Parent} \sqcap \text{Female},$$
$$\text{Child} \sqsubseteq \exists \text{hasParent.Father},\ \text{Child} \sqsubseteq \exists \text{hasParent.Mother},$$
$$\text{hasParent} \sqsubseteq \text{relatedTo}\}$$
$$\mathcal{A} = \{\text{Father(Alex)}, \text{Child(Bob)}, \text{hasParent(Bob, Alex)}\}$$

A toy ontology on a family



An example form the food ontology FoodOn

Why do we use RDF, RDFS and OWL?

Reason #1: Provide widely used vocabularies for defining all kinds of explicit, formal and shared knowledge

Reason #2: OWL supports Description Logics for representing complex knowledge

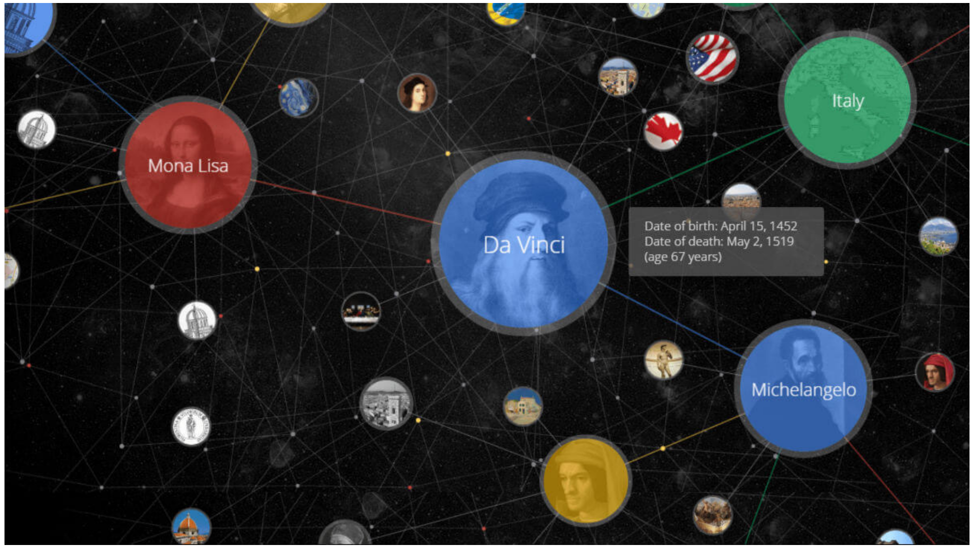
Reason #3: already have been widely deployed

E.g., in Life Sciences: SNOMED Clinical Terms, The Gene Ontology (GO), FoodOn, Human Disease Ontology (DOID), The Orphanet Rare Disease ontology (ORDO)

Chen, J., et al. "Knowledge Graphs for the Life Sciences: Recent Developments, Challenges and Opportunities." *Transactions on Graph Data and Knowledge (TGDK)* (2023).

What is Knowledge Graph?

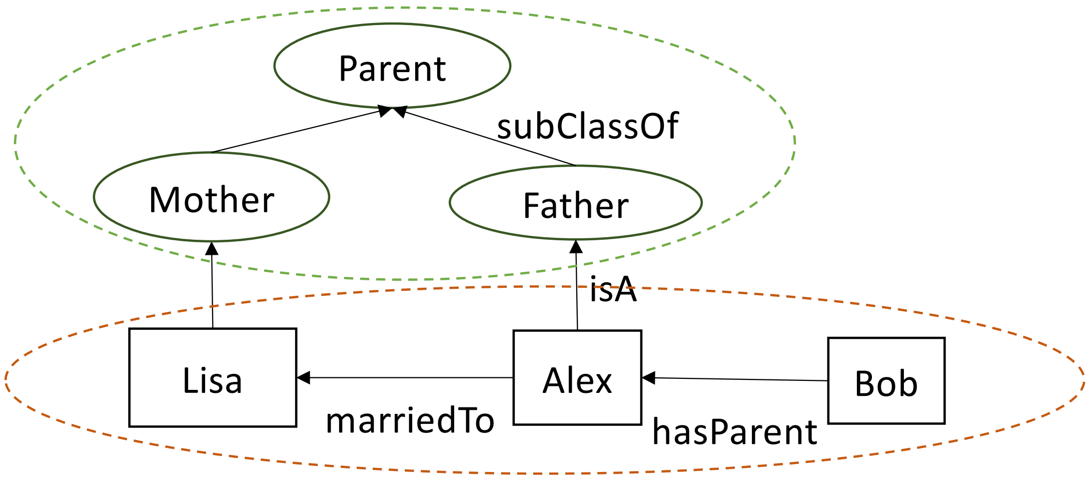
- “Knowledge Graph” was proposed by Google in 2012, referring to its services to enhance its search engine’s results with knowledge gathered from a variety of sources



- Knowledge ≈ Instances + Facts, represented as RDF triples e.g., <Box, hasParent, Alex>
- Linked and graph structured data

Knowledge Graph vs Ontology

- *Database perspective*: ontology as the schema of Knowledge Graph



- *Knowledge Representation Perspective*: an OWL ontology can be regarded as the composition of a TBox (terminologies) and an ABox (assertions); Knowledge Graph can be understood as (a part) of ABox

Ontology sometimes is regarded as logic equipped Knowledge Graph

Ontology and Knowledge Graph Embedding

- To represent symbols (e.g., entities and relations) in a vector space with their relationships concerned, mainly for being consumed by statistical analysis and machine learning

$$v_{Bob} + v_{hasParent} \approx v_{Alex}$$

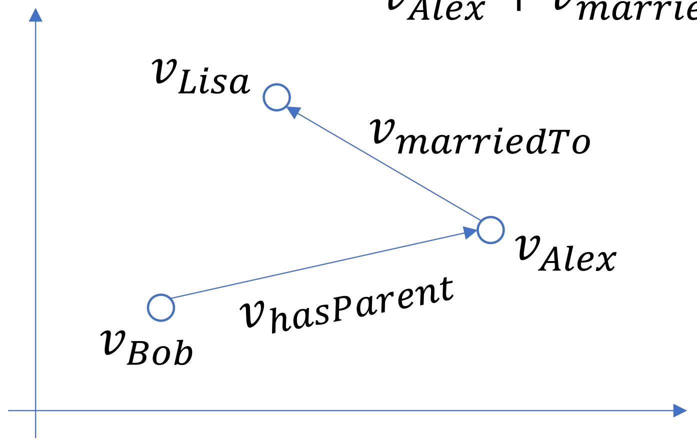
$$v_{Alex} + v_{marriedTo} \approx v_{Lisa}$$

Example: **TransE for RDF triples**

$v_i \in R^n$

<Bob, hasParent, Alex>
<Alex, marriedTo, Lisa>
…

Learning algorithm

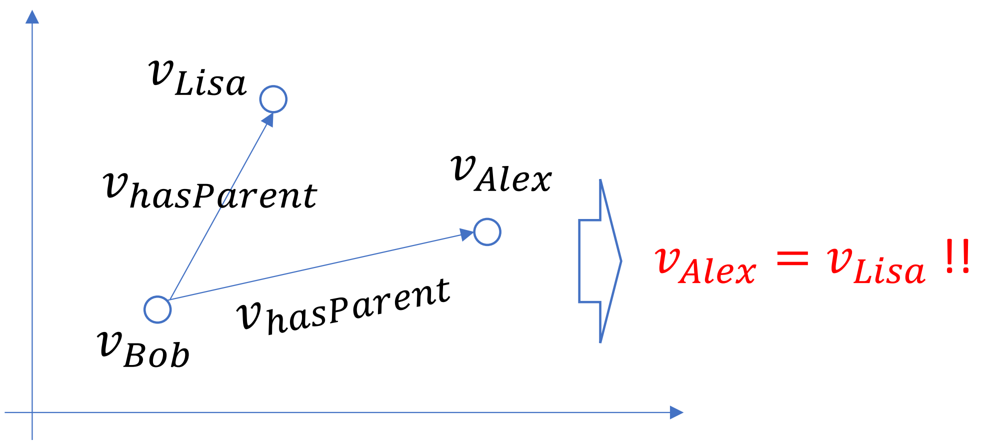


Bordes, A., et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems* 26 (2013).

Ontology and Knowledge Graph Embedding

Limitations of the simple translation-based relation modeling

Cannot deal with **one-to-many, many-to-one and many-to-many relations**



How to embed an OWL (or RDFS) ontology like the family example? Cannot model **concepts and their logical relationships**

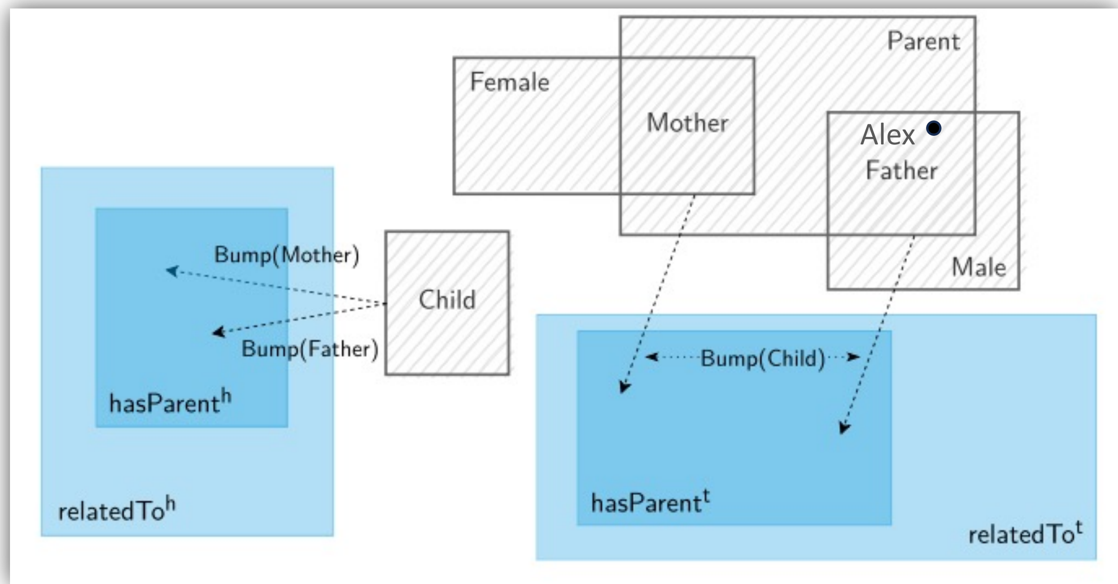
$$\mathcal{T} = \{\text{Father} \sqsubseteq \text{Parent} \sqcap \text{Male},\ \text{Mother} \sqsubseteq \text{Parent} \sqcap \text{Female},$$
$$\text{Child} \sqsubseteq \exists \text{hasParent.Father},\ \text{Child} \sqsubseteq \exists \text{hasParent.Mother},$$
$$\text{hasParent} \sqsubseteq \text{relatedTo}\}$$
$$\mathcal{A} = \{\text{Father(Alex)}, \text{Child(Bob)}, \text{hasParent(Bob, Alex)}\}$$

Wide research for modeling complex relations and graph patterns for embedding KGs: TransR, ComplEx, DistMult, ConvE, RDF2Vec …

Embedding OWL Ontologies

$$\mathcal{T} = \{\text{Father} \sqsubseteq \text{Parent} \sqcap \text{Male},\ \text{Mother} \sqsubseteq \text{Parent} \sqcap \text{Female},$$
$$\text{Child} \sqsubseteq \exists \text{hasParent}.\text{Father},\ \text{Child} \sqsubseteq \exists \text{hasParent}.\text{Mother},$$
$$\text{hasParent} \sqsubseteq \text{relatedTo}\}$$
$$\mathcal{A} = \{\text{Father}(\text{Alex}), \text{Child}(\text{Bob}), \text{hasParent}(\text{Bob}, \text{Alex})\}$$

Learning Algorithms



Box^{2}EL for OWL ontologies of Description Logic $\mathcal{EL}^{++}$ (like the family example)

Entity/instance: Point
Concept: Box (center vector & offset vector)
Relation/role: a head box & a tail box
Concept interaction: bump vector

Concept subsumption
Instance membership
Concept intersection
Role inclusion and composition

Existential quantification

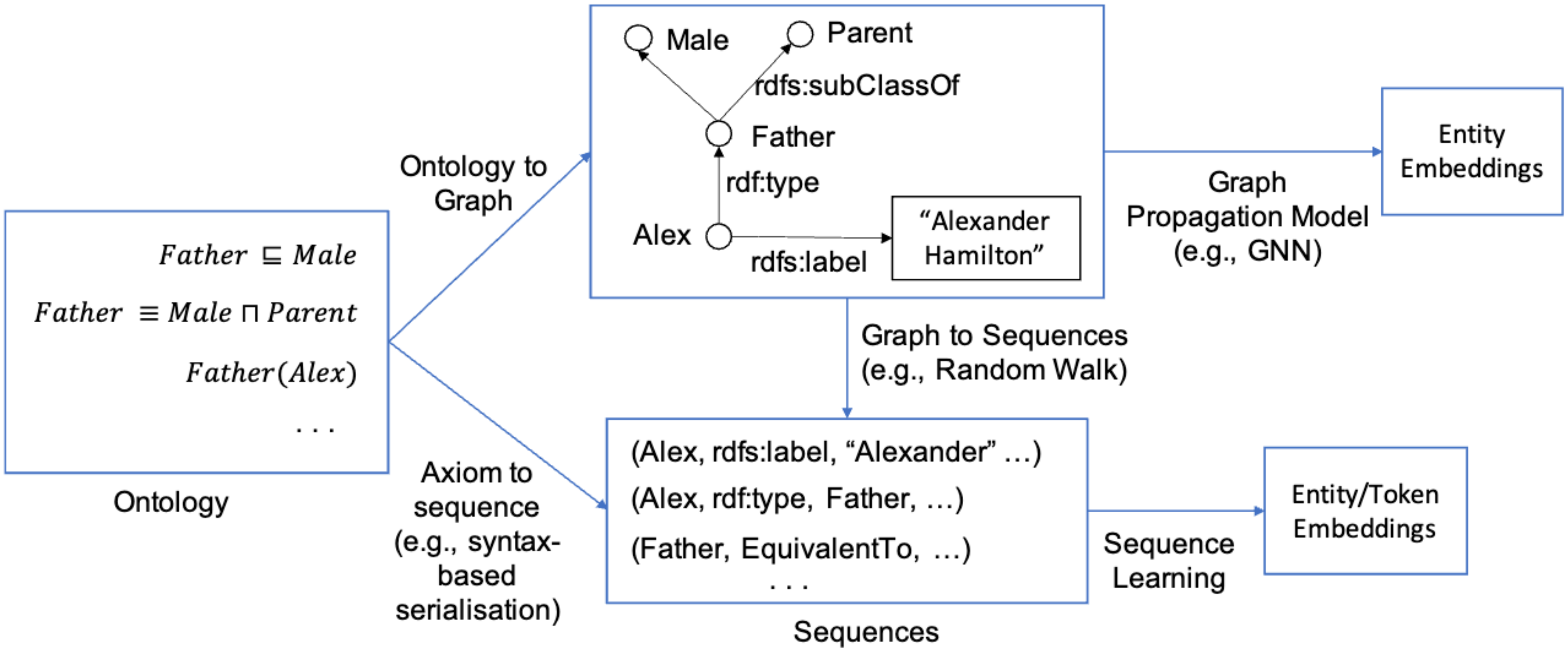
$C \sqsubseteq \exists r.D$: Box(C) ⊗ Bump(D) ⊆ Head(r)
Box(D) ⊗ Bump(C) ⊆ Tail(r)

Jackermeier, M., Chen, J., Horrocks, I.,"Dual Box Embeddings for the Description Logics EL++." The Web Conference 2024.

Paradigms for Ontology Embedding

- Geometric modeling (like Box^{2}EL)
  - Pros: interpretable; sound representation of formal semantics
  - Cons: hard to incorporate informal semantics like textual literals; hard to deal with all the features of OWL
- Sequence modeling
  - Transform axioms and literals into sentences;
  - Train word embedding (sequence learning) models
- Graph propagation
  - Transform axioms to a graph

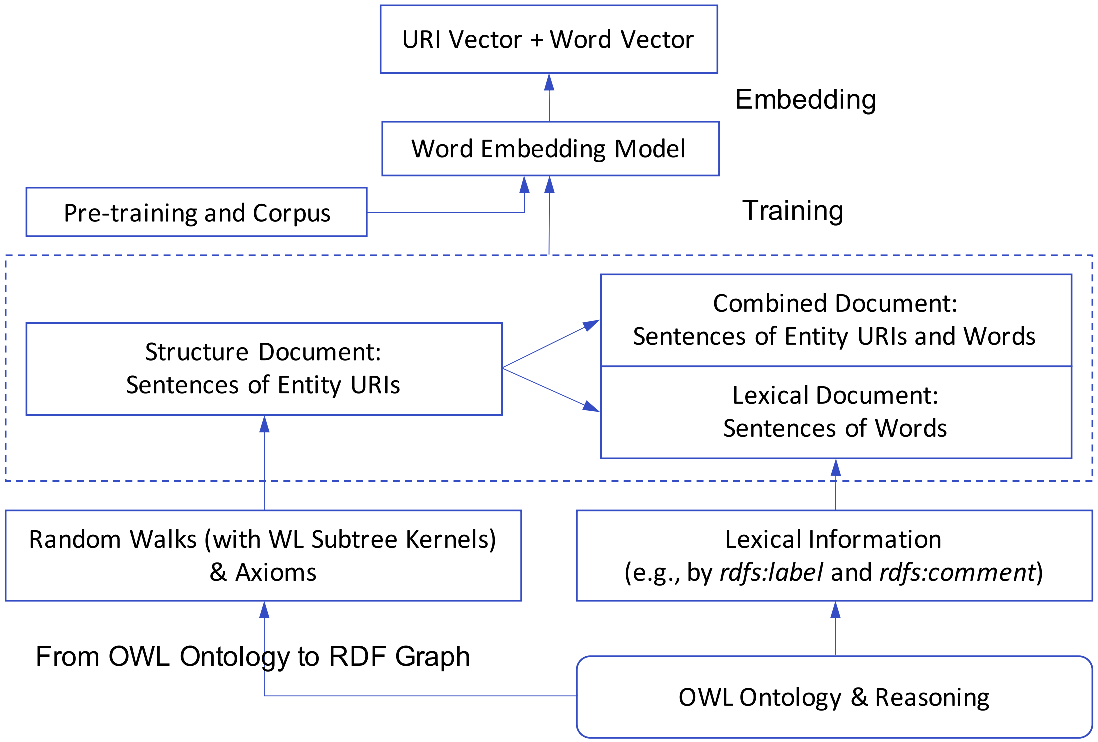
Paradigms for Ontology Embedding



Simple ideas of Sequence Learning & Graph Propagation

A General Ontology Embedding Tool **OWL2Vec***

- **Pipeline & Sequence Modeling**
  - OWL axioms to RDF Graph
    - Two approaches
  - Sequences extracted with literals and random walks over the RDF graph, and axiom serialisation
  - Train Word2Vec embeddings
- Pros & Cons
  - Consider literals; support all kinds of ontology semantics
  - Embed correlations; miss some formal semantics



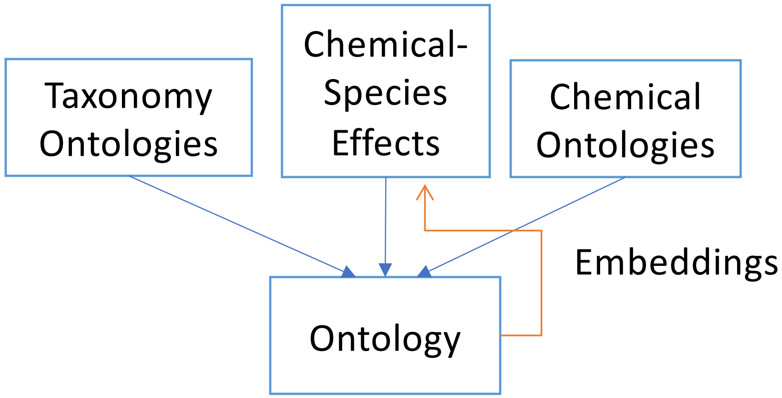
Chen J., et al. "OWL2Vec*: Embedding of OWL ontologies." *Machine Learning* 110.7 (2021): 1813-1845.

Application/Evaluation of Ontology Embeddings

- Link Prediction / Recommendation
  - E.g., protein interaction prediction, ecotoxicological effect prediction

	Model	H@10	H@10 (F)	H@100	H@100 (F)	MR	MR (F)	AUC	AUC (F)
Yeast	ELEm	0.10	0.23	0.50	0.75	247	187	0.96	0.97
	EmEL^{++}	0.08	0.17	0.48	0.65	336	291	0.94	0.95
	BoxEL	0.09	0.20	0.52	0.73	423	379	0.93	0.94
	ELBE	**0.11**	0.26	0.57	0.77	201	154	0.96	0.97
	Box^{2}EL	**0.11**	**0.33**	**0.64**	**0.87**	**168**	**118**	**0.97**	**0.98**
Human	ELEm	**0.09**	0.22	0.43	0.70	658	572	0.96	0.96
	EmEL^{++}	0.04	0.13	0.38	0.56	772	700	0.95	0.95
	BoxEL	0.07	0.10	0.42	0.63	1574	1530	0.93	0.93
	ELBE	**0.09**	0.22	0.49	0.72	434	362	0.97	**0.98**
	Box^{2}EL	**0.09**	**0.28**	**0.55**	**0.83**	**343**	**269**	**0.98**	**0.98**

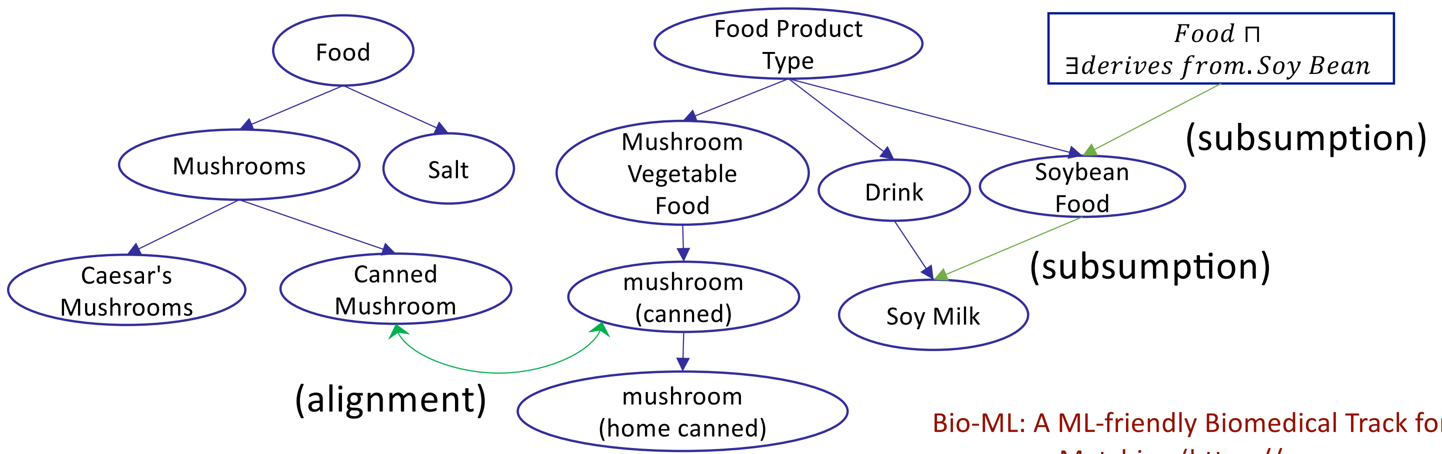
Results of Box^{2}EL on protein-protein interaction prediction on two datasets constructed by the STRING database (ABox) and the Gene ontology (TBox)



Myklebust, Erik B., et al. "Prediction of adverse biological effects of chemicals using knowledge graph embeddings." *Semantic Web* 13.3 (2022): 299-338.

Applications and Evaluation of Ontology Embeddings

- Link Prediction / Recommendation
  - E.g., protein function prediction, ecotoxicological effect prediction
- Knowledge Engineering
  - E.g., entity alignment, subsumption completion



Bio-ML: A ML-friendly Biomedical Track for Equivalence and Subsumption Matching (https://www.cs.ox.ac.uk/isg/projects/ConCur/oaei/)

Applications and Evaluation of Ontology Embeddings

- Link Prediction / Recommendation
  - E.g., protein function prediction, ecotoxicological effect prediction
- Knowledge Engineering
  - E.g., entity alignment, subsumption completion
- Knowledge Retrieval
  - E.g., Retrieval Augmented Generation (RAG)
  - (more to discuss on Ontology & Large Language Models)

https://www.deeplearning.ai/short-courses/knowledge-graphs-rag/

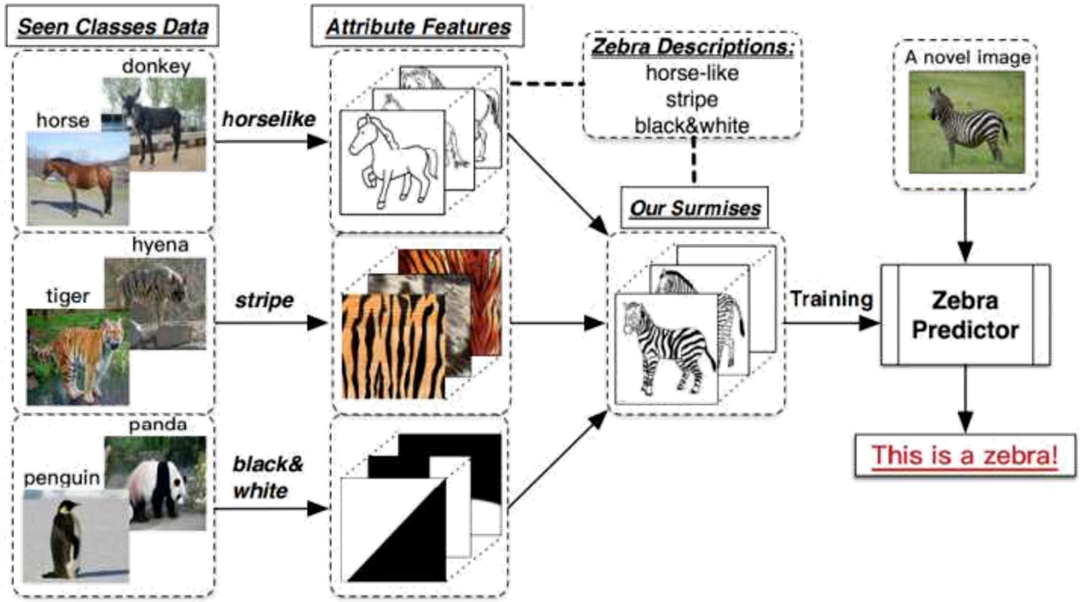
Applications and Evaluation of Ontology Embeddings

- Link Prediction / Recommendation
  - E.g., protein function prediction, ecotoxicological effect prediction
- Knowledge Engineering
  - E.g., entity alignment, subsumption completion
- Knowledge Retrieval
  - E.g., Retrieval Augmented Generation (RAG)
- Augmenting Machine Learning
  - E.g., injecting external knowledge of classes for zero-shot learning

Chen, J, et al. "Zero-Shot and Few-Shot Learning With Knowledge Graphs: A Comprehensive Survey." Proceedings of the IEEE (2023).

Knowledge-driven Zero-shot Learning (ZSL)

- What is ZSL?
  - Predict samples with new classes that have never appeared in training
  - Seen classes vs unseen classes

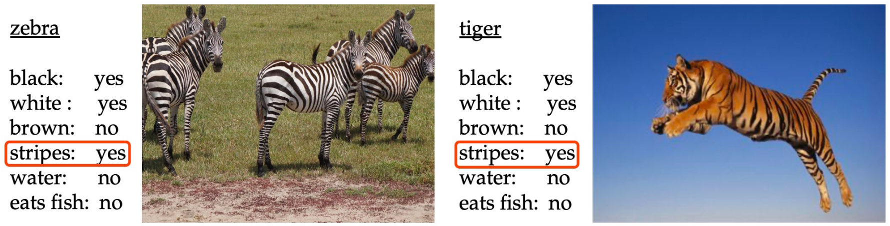


Zero-shot Image Classification

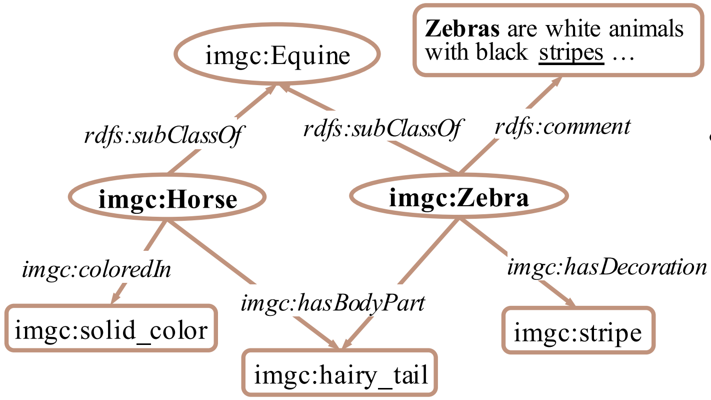
Knowledge-driven Zero-shot Learning (ZSL)

- **External knowledge** (a.k.a. **side information**) model the relationship between classes, thus enabling the transfer of the model from seen classes to unseen classes.

- Textual description:
*"**Zebras** are white animals with black stripes, they have larger, rounder ears than horses …"*



- Attribute descriptions, e.g., visual properties of animals

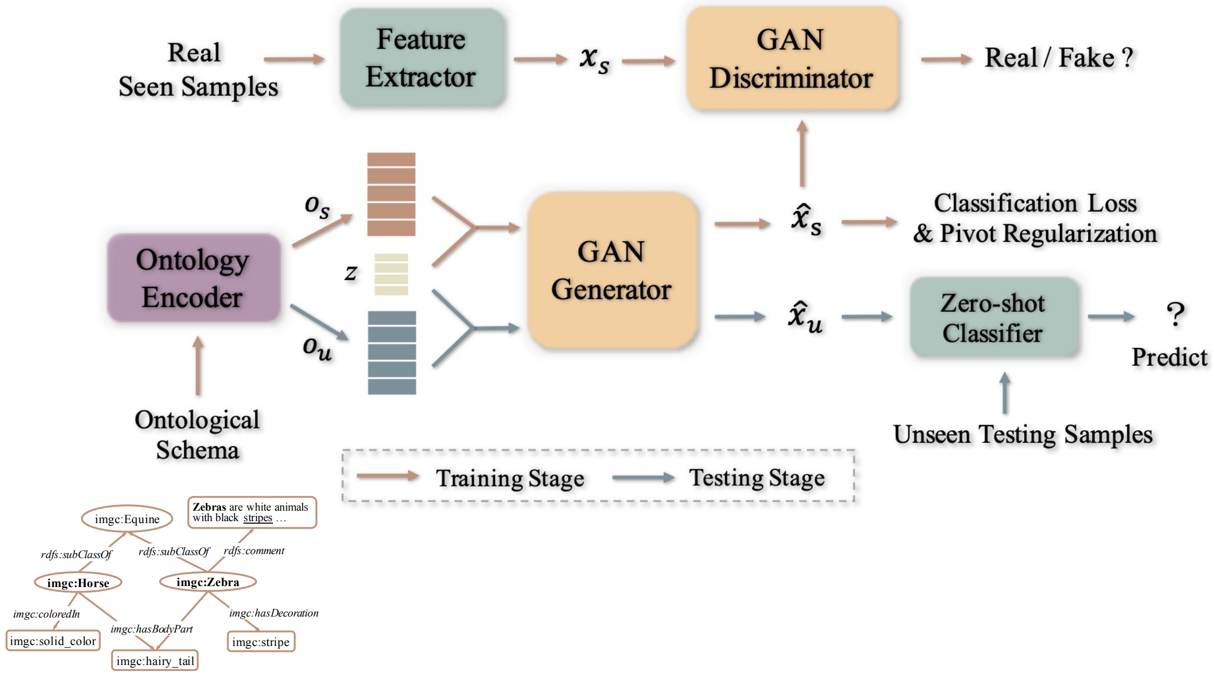


- Relational Facts, Taxonomy, Literals

"$Zebra \sqsubseteq Equine \sqcap \exists hasTexture.Stripes \sqcap \exists hasHabitat.Meadow$ … "
"$hasUncle \equiv hasParent \circ hasBrother$"

- Logics & rules

OntoZSL: Ontology Embeddings and Generative Models for ZSL



- Ontology Encoder: *embed the ontology (taxonomy, facts, textual literals) for vector representations for the classes*
- Feature Extractor: *extract the features of real images*
- Generation Model (GAN Discriminator + Generator): *generate samples for classes conditioned on their embeddings*
- Classifier: *learns classifiers for unseen classes by the generated features*

Geng, Yuxia, et al. "OntoZSL: Ontology-enhanced zero-shot learning." *Proceedings of the Web Conference 2021*.

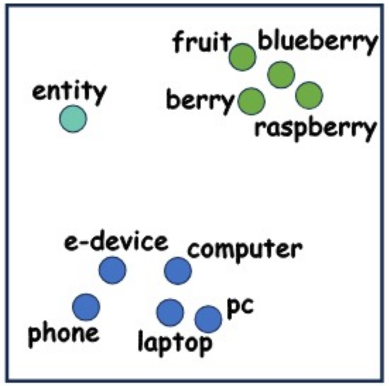
Challenges and Opportunities from Large Language Models

- LLMs for knowledge representation and knowledge engineering

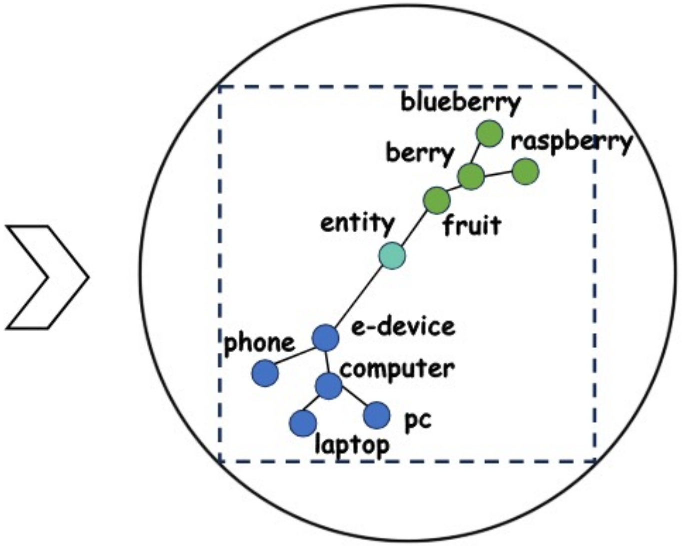
- Knowledge graph & ontology for LLMs

Pan, J., et al. "Large Language Models and Knowledge Graphs: Opportunities and Challenges." *Transactions on Graph Data and Knowledge* (2023).

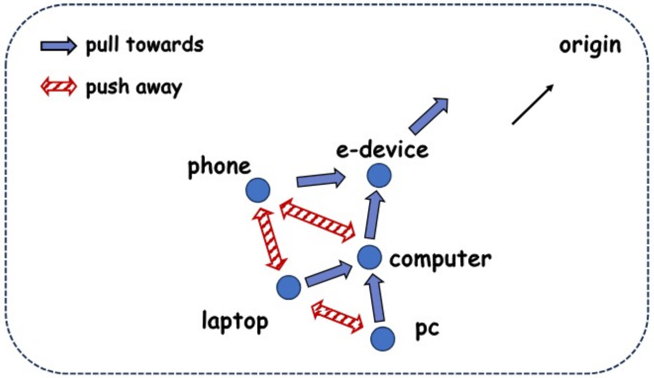
Language Model as Hierarchy Encoder (HiT)



Concept's Text Embedding in Euclidean Space by a Pre-trained LM



Concept's Text Embedding in Poincare´ Ball Space by a PLM re-trained on an ontology



The idea for PLM re-training

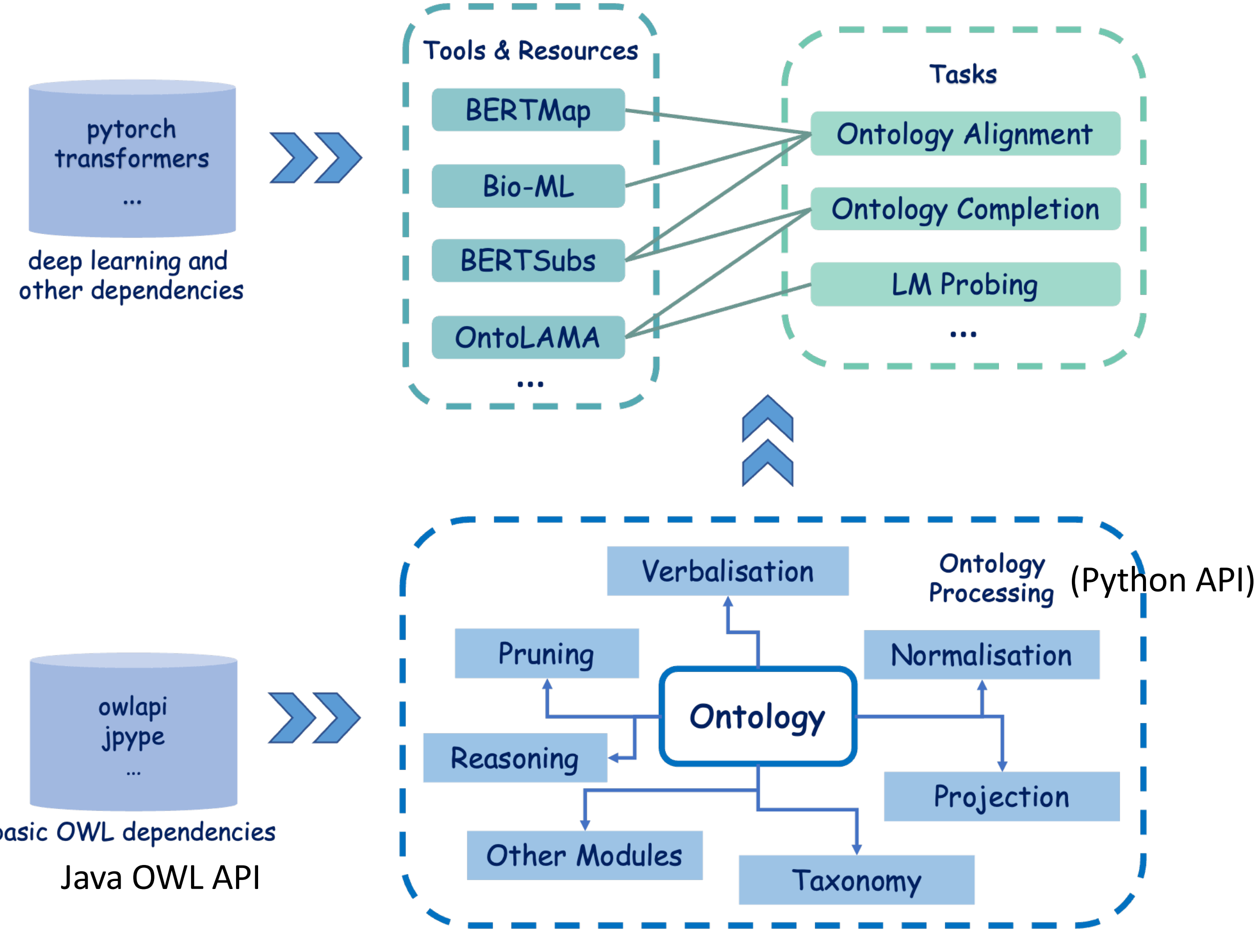
(Ongoing Work)

He, Yuan, et al. "Language Models as Hierarchy Encoders." arXiv preprint arXiv:2401.11374 (2024).

An LM-based Ontology Engineering Library

DeepOnto

https://github.com/KRR-Oxford/DeepOnto



He, Y., et al. "DeepOnto: A Python package for ontology engineering with deep learning." *Semantic Web Journal* (2024).

Several tools implemented in DeepOnto

- **BERTMap: A BERT−Based Ontology Alignment System** by fine-tuning pre-trained language models (PLMs) by synonyms (AAAI 2022)
- **BERTSubs: ontology subsumption prediction** by prompts for encoding concept contexts and PLM fine-tuning (World Wide Web Journal 2023)
- **Machine Learning-Friendly Biomedical Datasets for Equivalence and Subsumption Ontology Matching** (ISWC 2022)
- **OntoLAMA: a Tool of Language Model Analysis** for Ontology Subsumption Inference (Findings of the ACL 2023)
- **ICON: taxonomy completion with missing common parents** (The Web Conference 2024)
- More in our TODO list; **External contributions are very welcomed**

How to augment Large Language Models?

- Knowledge Graphs
- Ontologies
- Tables / Data Lakes
- …

Pre-train / Fine-tune / Edit

Retrieval Augmented Generation (RAG)

Support (Large) Language Models

- Capture domain knowledge
- Reason with private knowledge
- Deal with evolving knowledge
- Add explanations / citations
- Rely on less samples / training
- …

Open Questions

- How will LLMs change knowledge representation and knowledge engineering?
- What roles can knowledge representation/engineering play in the era of LLMs?
- Can knowledge embeddings support RAG and how?

Thanks for your attention

Q&A