Knowledge Engineering meets (Large) Language Models

Jiaoyan Chen

Lecturer in Department of Computer Science, University of Manchester, UK

Senior Researcher in University of Oxford, UK

Amazon Search Research Talk Series, 12th July 2024

Part I: Symbolic Knowledge Representation

What is an ontology?

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

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

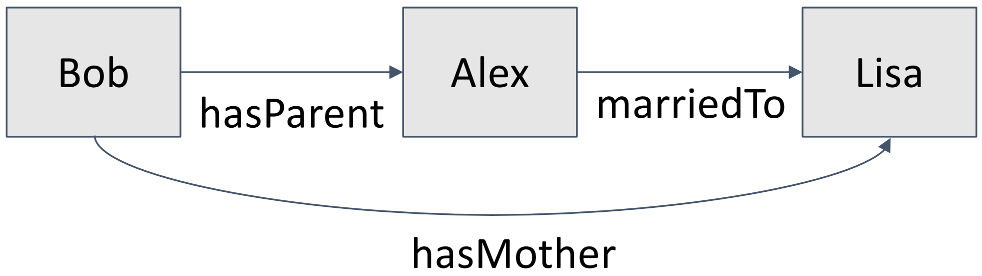
A toy ontology on a family

- Formal
- Explicit
- Shared

How to define formal, explicit and shared ontologies?

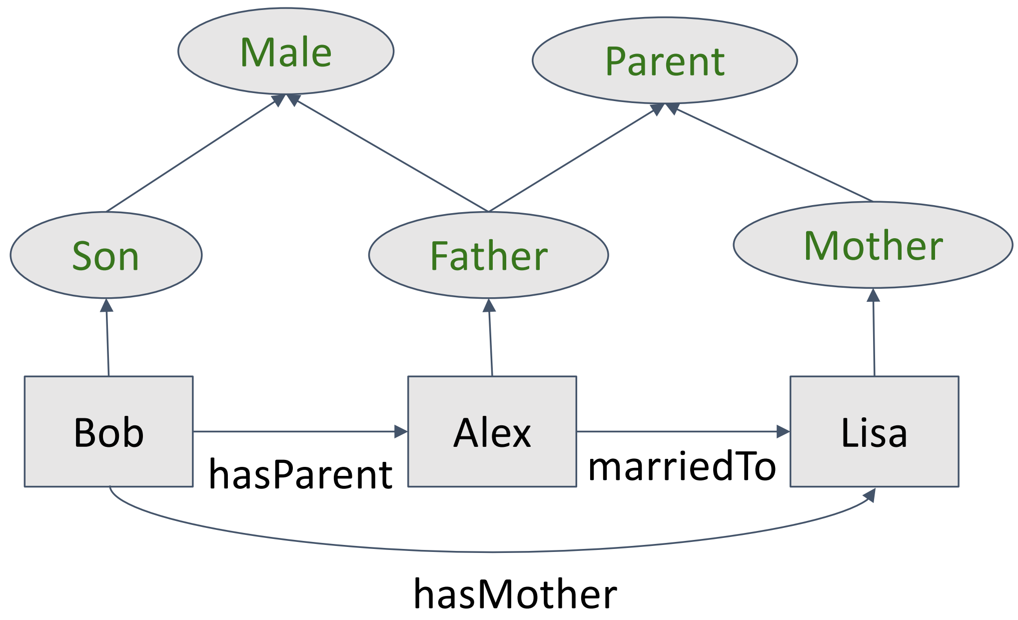
Ontology Languages

- **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,

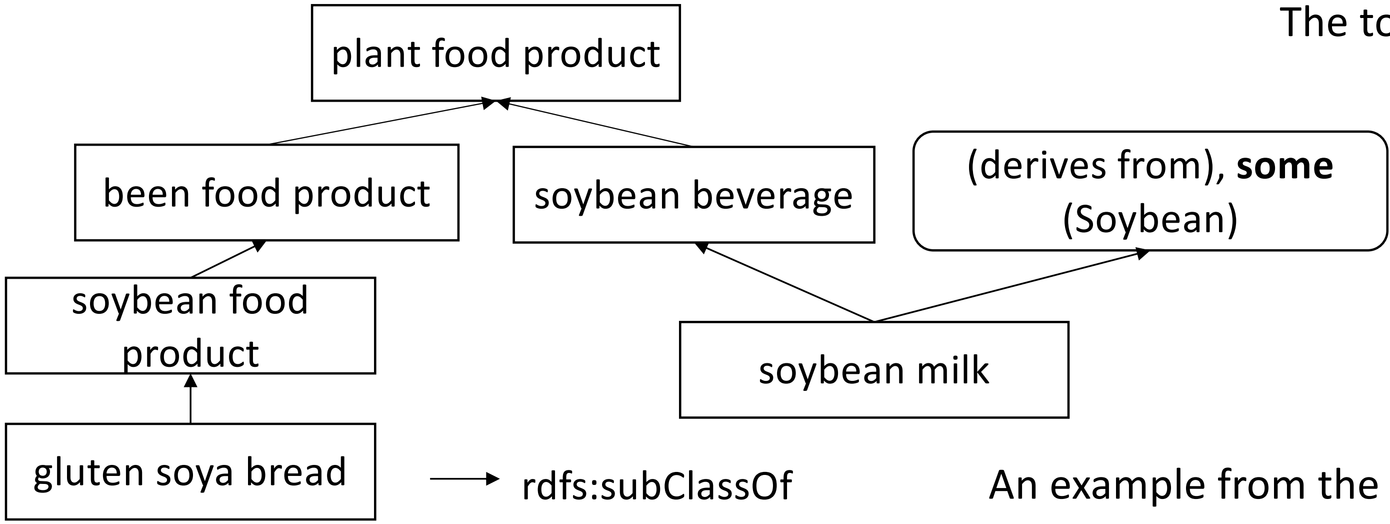


Ontology Languages

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

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

The toy ontology on a family



An example from the food ontology FoodOn

Why do we use RDF, RDFS and OWL?

Reason #1: a bit more semantics; OWL supports Description Logics for representing complex knowledge

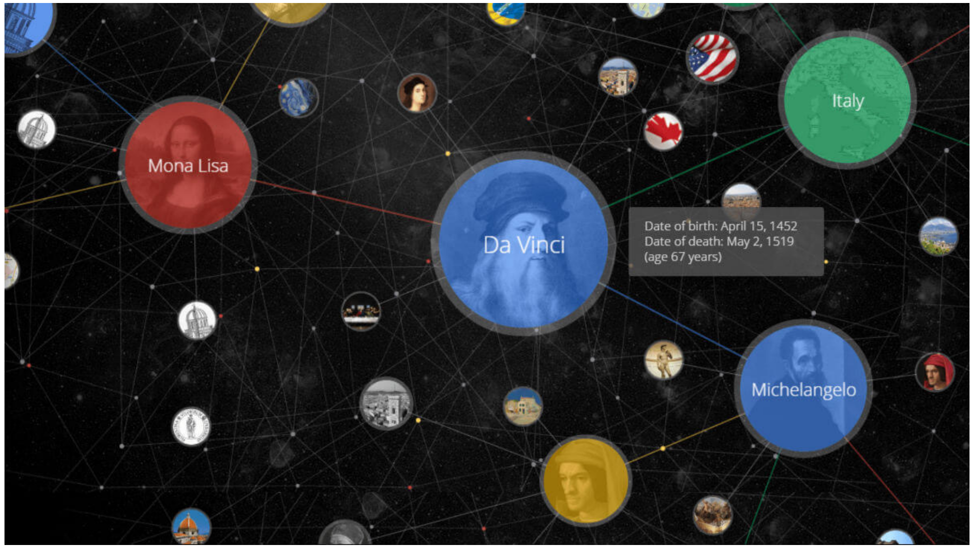
Reason #2: Widely used vocabularies; already have been widely deployed

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

Part II: Sub-symbolic Knowledge Representation with Embeddings

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

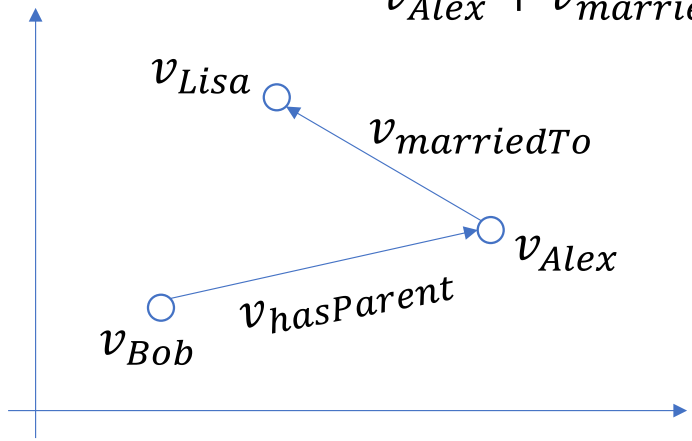
$$v_{Bob} + v_{hasParent} \approx v_{Alex}$$
$$v_{Alex} + v_{marriedTo} \approx v_{Lisa}$$

Example: **TransE for RDF triples**

$v_i \in R^n$

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

Learning algorithm

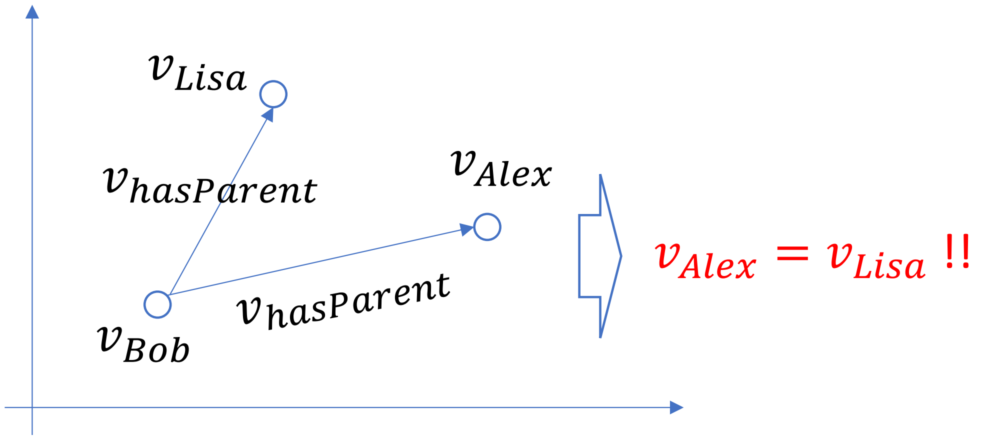


Bordes, A., et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems* 26 (2013).

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**

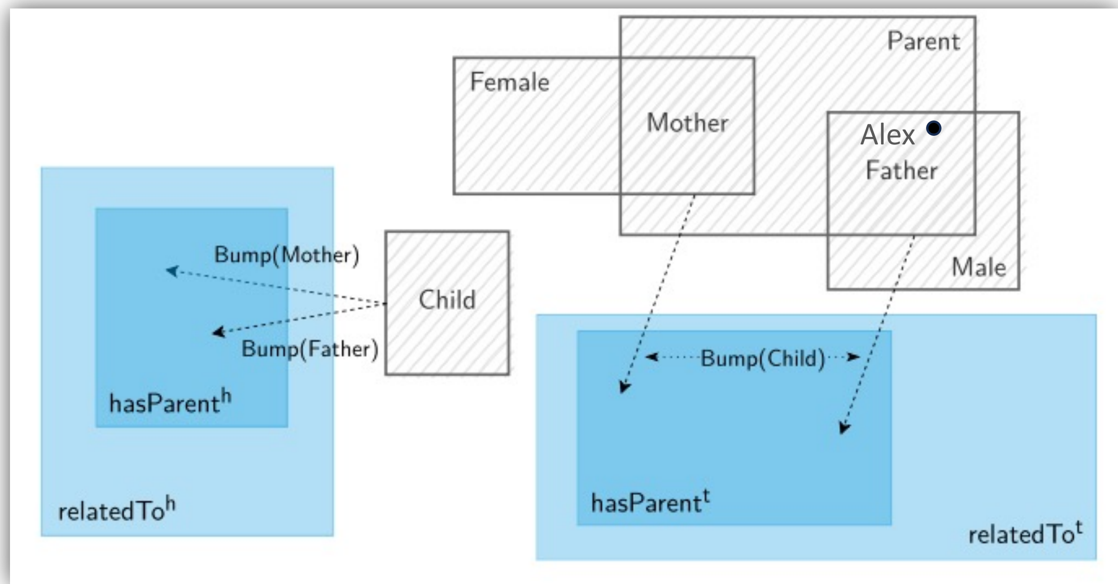
$$\mathcal{T} = \{\text{Father} \sqsubseteq \text{Parent} \sqcap \text{Male},\ \text{Mother} \sqsubseteq \text{Parent} \sqcap \text{Female},$$
$$\text{Child} \sqsubseteq \exists \text{hasParent.Father},\ \text{Child} \sqsubseteq \exists \text{hasParent.Mother},$$
$$\text{hasParent} \sqsubseteq \text{relatedTo}\}$$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

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

Embedding OWL Ontologies

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

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) $\otimes$ Bump(D) $\subseteq$ Head(r)
Box(D) $\otimes$ Bump(C) $\subseteq$ Tail(r)

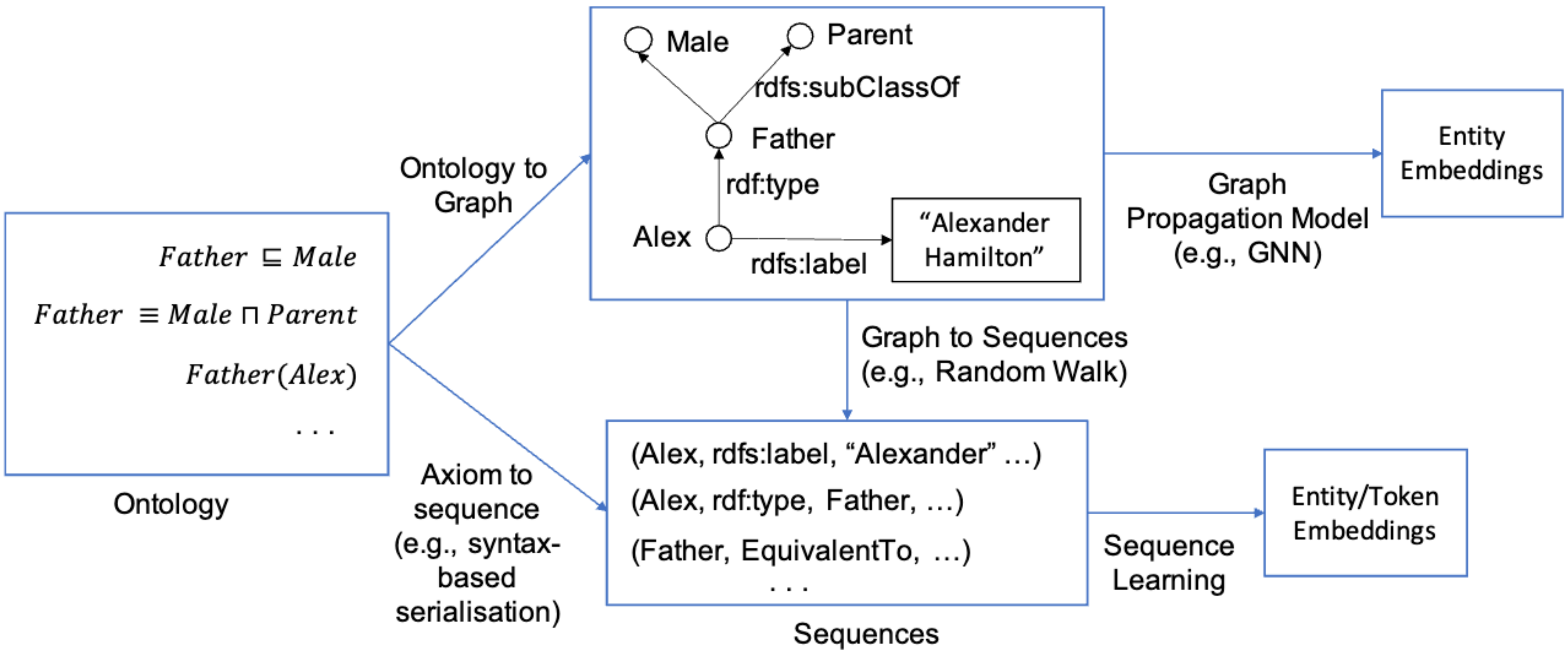
Jackermeier, M., Chen, J., Horrocks, I.,"Dual Box Embeddings for the Description Logics EL++." The Web Conference 2024.

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 into a graph

Chen, J., et al.,"Ontology Embedding: A Survey of Methods, Applications and Resources." https://arxiv.org/abs/2406.10964.

Paradigms for Ontology Embedding



Paradigms of Sequence Learning & Graph Propagation

Application/Evaluation of Ontology Embeddings

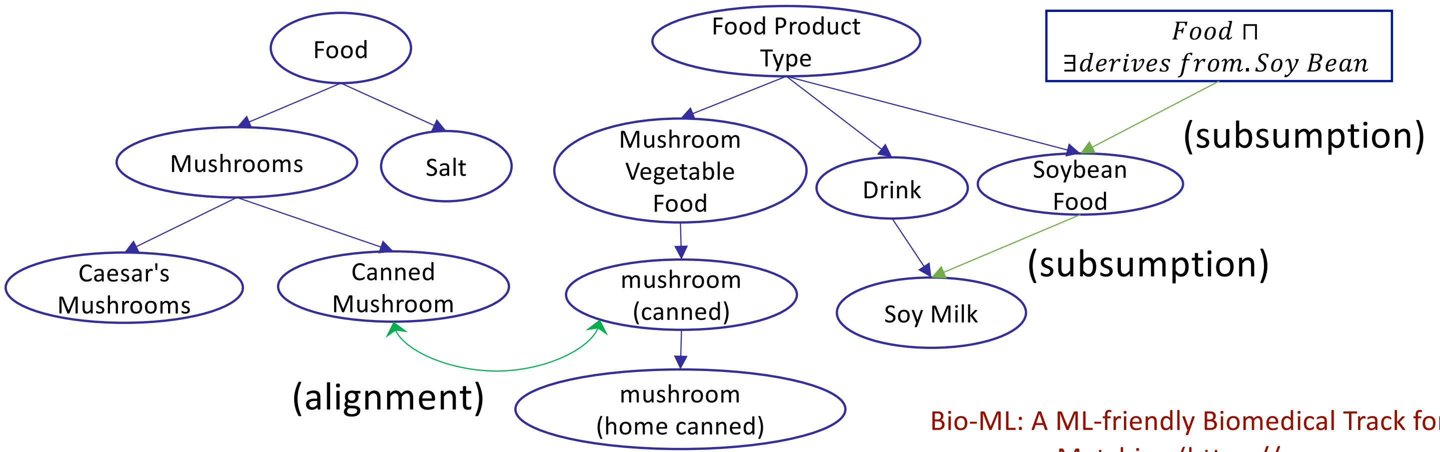
- Link Prediction
  - E.g., protein-protein interaction prediction

	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^2EL	**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^2EL	**0.09**	**0.28**	**0.55**	**0.83**	**343**	**269**	**0.98**	**0.98**

Results of Box^2EL on protein-protein interaction prediction.
the STRING database (ABox) + the Gene ontology (TBox)

Applications and Evaluation of Ontology Embeddings

- Link Prediction
  - E.g., protein-protein interaction prediction, ecotoxicological effect prediction
- Knowledge Engineering
  - E.g., entity alignment, subsumption completion, ontology learning



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
  - E.g., protein-protein interaction prediction, ecotoxicological effect prediction
- Knowledge Engineering
  - E.g., entity alignment, subsumption completion, ontology learning
- 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).

Part III: Parametric Knowledge from Language Models

Challenges and Opportunities from (Large) Language Models

- Language models for neural knowledge representation, and for augmenting knowledge engineering

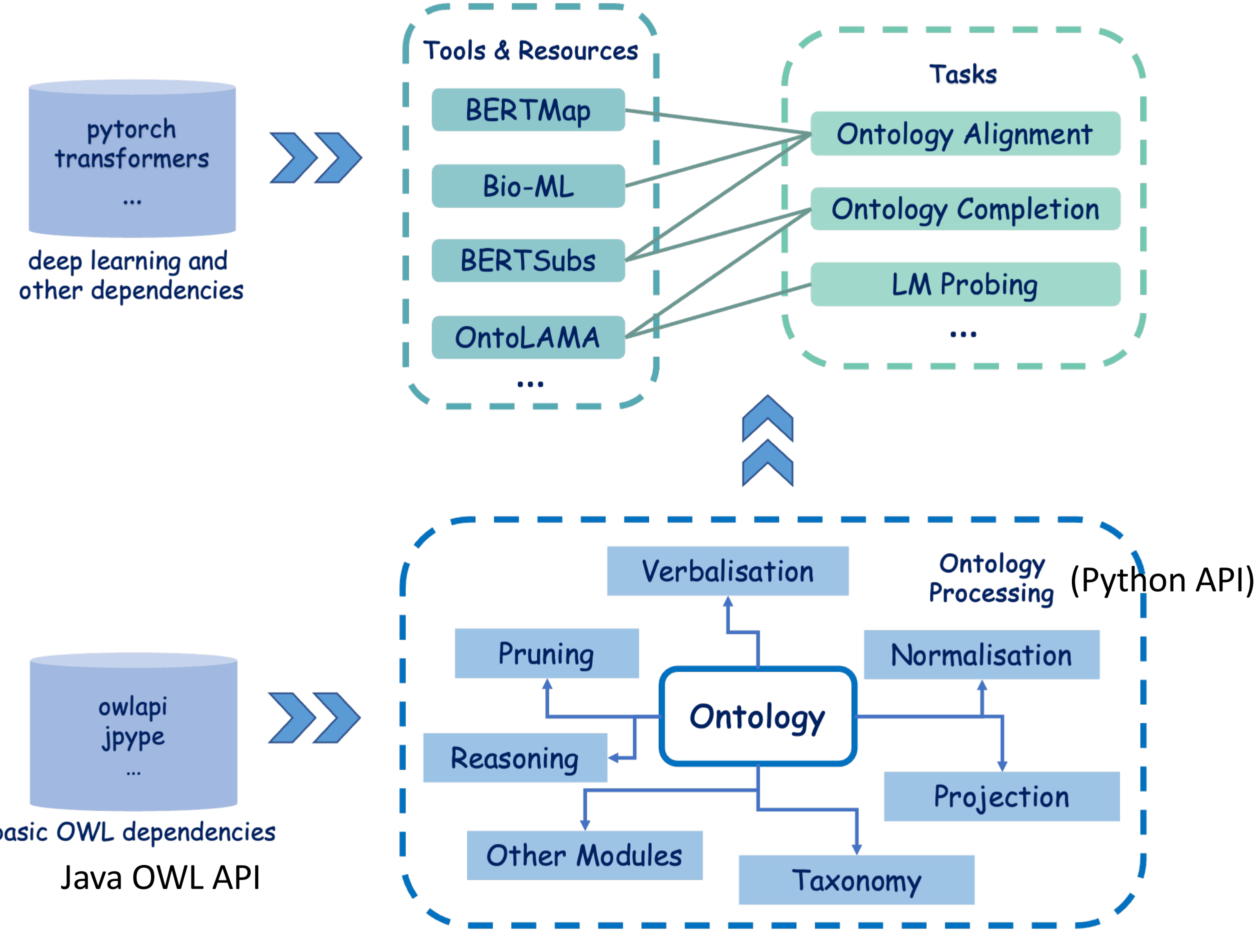
- Knowledge graph & ontology for LLMs

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

An LM-based Ontology Engineering Library

DeepOnto

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



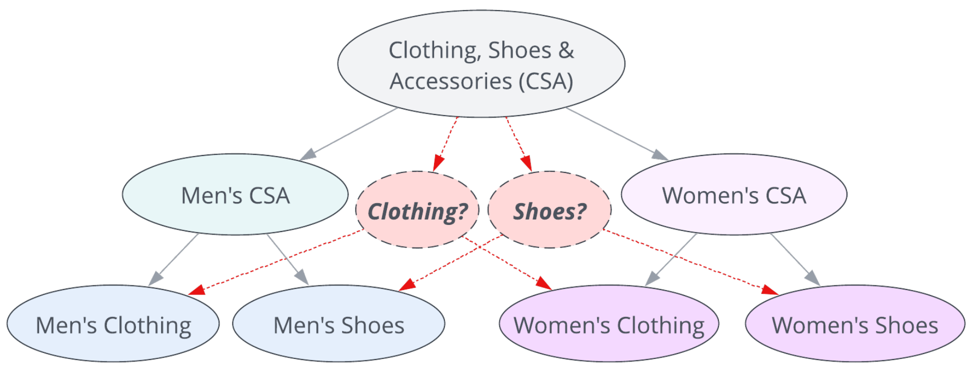
He, Y., et al. "DeepOnto: A Python package for ontology engineering with deep learning." *Semantic Web Journal* (2024).

Several tools implemented in DeepOnto

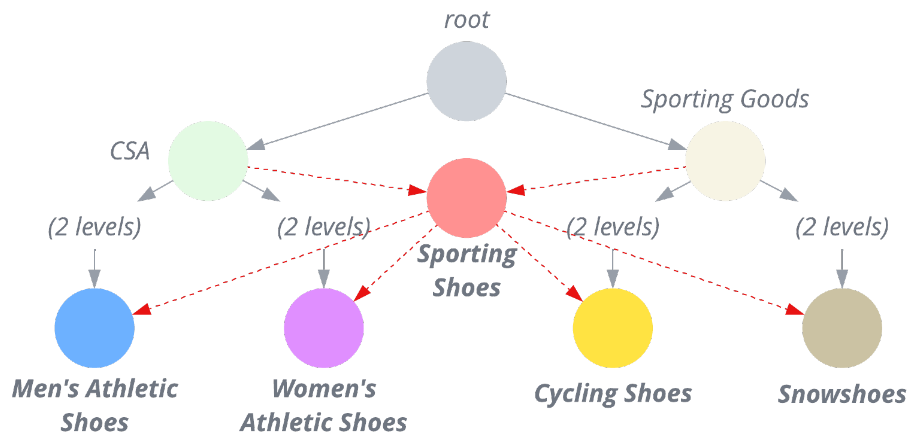
- **BERTMap: A BERT−Based Ontology Alignment System** by fine-tuning pre-trained language models (PLMs) by synonyms (AAAI 2022)
- **BERTSubs: ontology subsumption prediction** by prompts for encoding concept contexts and PLM fine-tuning (World Wide Web Journal 2023)
- **Machine Learning-Friendly Biomedical 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**

Implicit Taxonomy Completion

- Taxonomies of e.g., e-commerce have “holes”



Example 1: Concepts that should have existed



Example 2: Concepts bridging multiple branches of the taxonomy

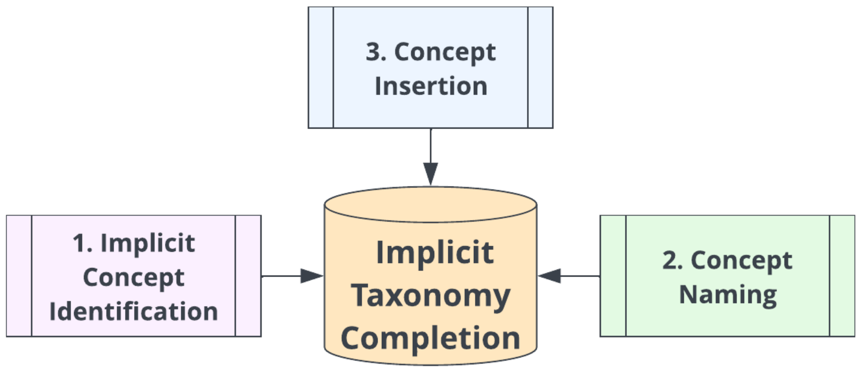
TAXONOMY CURATION

ICON

IMPLICIT CONCEPT INSERTION

Anatomy of the task

1. Identify the implicit concepts (BERT Embedding + nearest neighbour search with contrastive learning)
2. Generate the label for each implicit concept (text summarisation with T5 + prompts)
3. Find the parents and children for each implicit concept (classification with BERT fine-tuning & traversal algorithms)

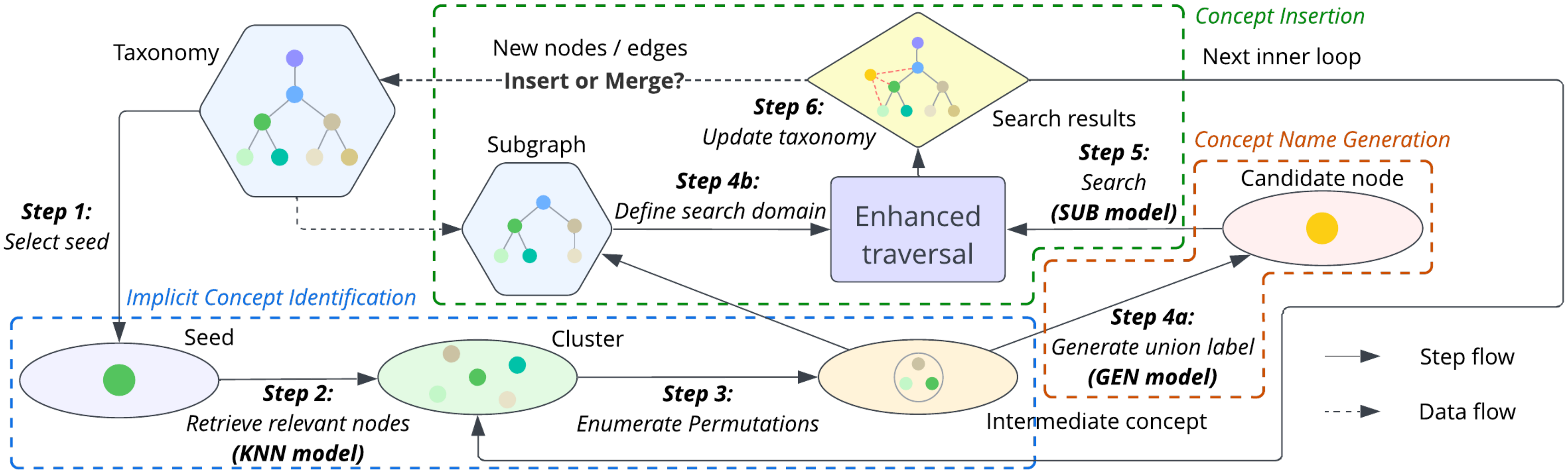


Shi, J., et al. "Taxonomy Completion via Implicit Concept Insertion." *The Web Conference* 2024.

TAXONOMY CURATION

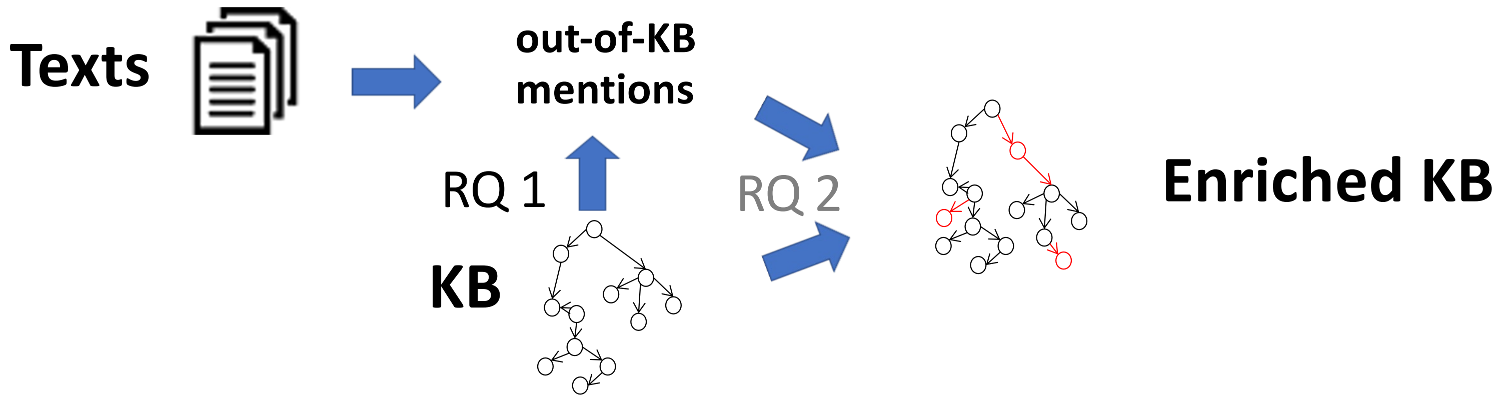
ICON

IMPLICIT CONCEPT INSERTION



New Concepts from Text for Ontology Completion

- RQ1: How to identify out-of-KB mentions, i.e., NIL entity uncaptured by a Knowledge Base (ontology or knowledge graph), from texts?
- RQ2: How to insert out-of-KB mentions as new entities into a Knowledge Base?



Two-step Framework

- Stage 1: Candidate generation
  - Retrieval K candidates with BM2.5 or a BERT-based bi-encoder trained with contrastive learning (a max-margin triplet loss)
  - Candidates are matched entities + NIL of mentions for RQ1, and edges for new concept insertion for RQ2

- Stage 2: Candidate ranking
  - Classification of K candidates (fine-tuning an encoder-only PLM e.g., BERT for multi-label classification or using a decoder-only LLM with Prompts)

Dong, H., et al. ”Reveal the Unknown: Out-of-Knowledge-Base Mention Discovery with Entity Linking." *CIKM 2023*.
Dong, H., et al. “A Language Model based Framework for New Concept Placement in Ontologies.” ESWC 2024.

Augment Large Language Models

- Knowledge Graphs
- Ontologies
- Tables / Data Lakes
- ...
- **Embeddings**

Symbolic and neural knowledge representations

LM Pre-train / Fine-tune for retrieval

Retrieval Augmented Generation (RAG)

Reasoning and generation

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

Thanks for your attention