Neural-symbolic Knowledge Representation and Reasoning

Jiaoyan Chen
Lecturer in Department of Computer Science
University of Manchester

ESSAI 2024 Athens

This course is

- introductory
- aimed at general computer scientist

- taught by
  - Uli Sattler - days 1-2
  - **Jiaoyan Chen - days 3-5**

(Hiking in Egina, Greece, 11/2023)

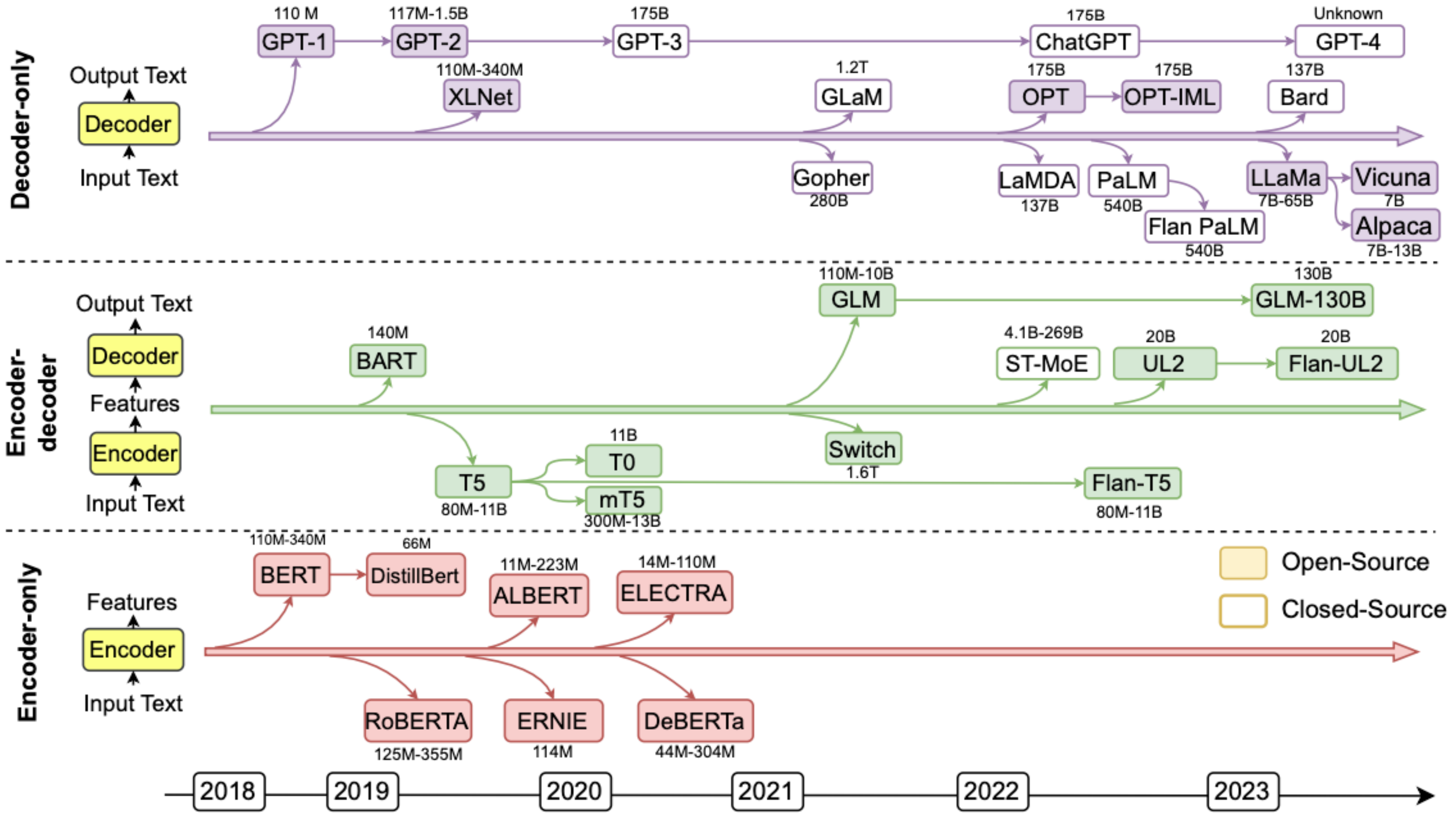
- explores combination/integration/collaboration of
  - Symbolic &
  - **Neural**
    - approaches to knowledge representation, reasoning, ML, …

Overview of this course

Day	Topic	Concepts	Technologies
1	Knowledge Graphs	parsing/serialisation, queries, schemas, validation & reasoning	RDF(S), SPARQL, SHACL,
2	Ontologies	Facts & background knowledge, entailments, reasoning & materialisation	OWL, OWL API, Owlready, Protégé
3	Knowledge Graph Embeddings	Classis Es, variants, inductive inference, literal-aware Es, incremental Es, application	TransE, TransH, TransR, GCN, R-GCN, OntoZSL, RMPI
4	Ontology Embeddings	Geometric embeddings, literal-aware OEs, faithfulness, evaluation & applications	ELEm, Box^{2}EL, OWL2Vec*, LogMap-ML, ZSL, mOWL
5	Language Models & KR, Discussion & Outlook	LM for KR, ontology & KG for LLM	BERTMap, BERTSubs, DeepOnto, ICON, BLINKOut, GraphRAG

Day 5 Language Models & KR

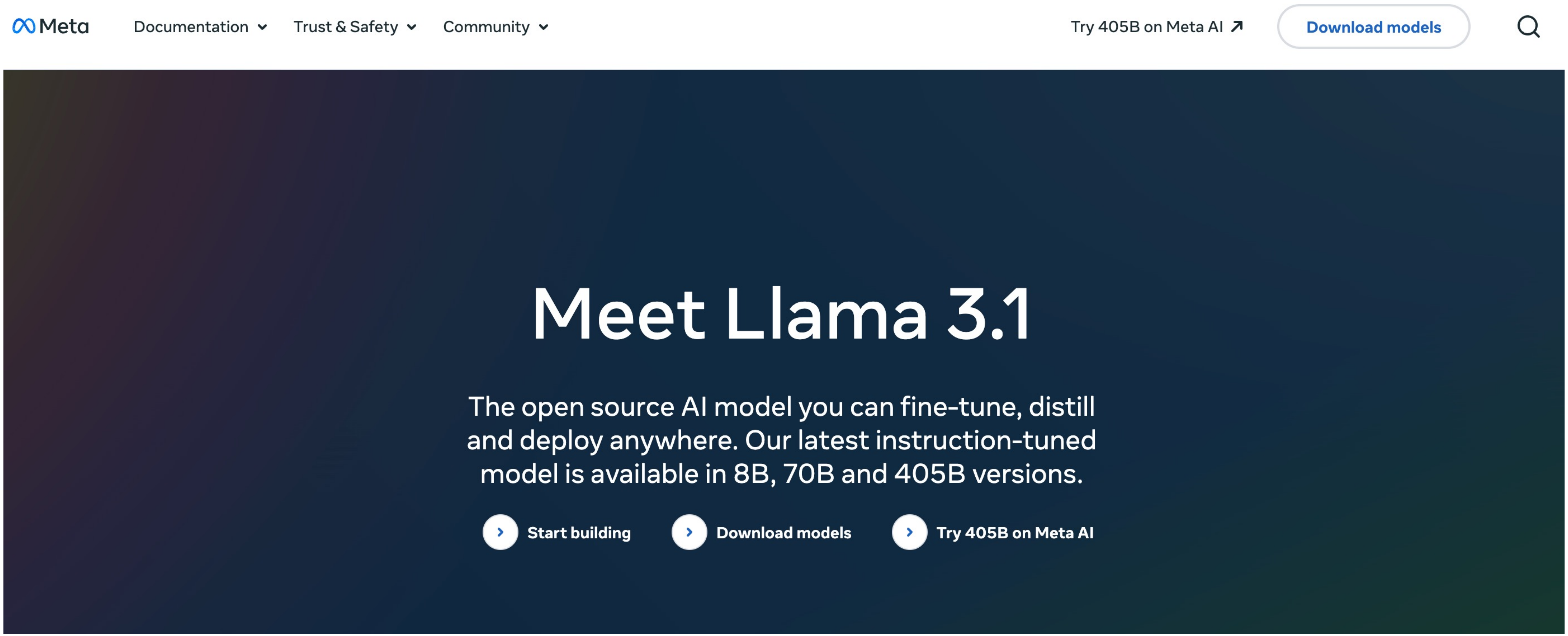
(Large) Language Models



Gemini, LLaMa 2 & 3

. . .

Pan, S.,, et al. "Unifying large language models and knowledge graphs: A roadmap." *IEEE Transactions on Knowledge and Data Engineering* (2024).



https://llama.meta.com/ (released in this week)

MANCHESTER 1824

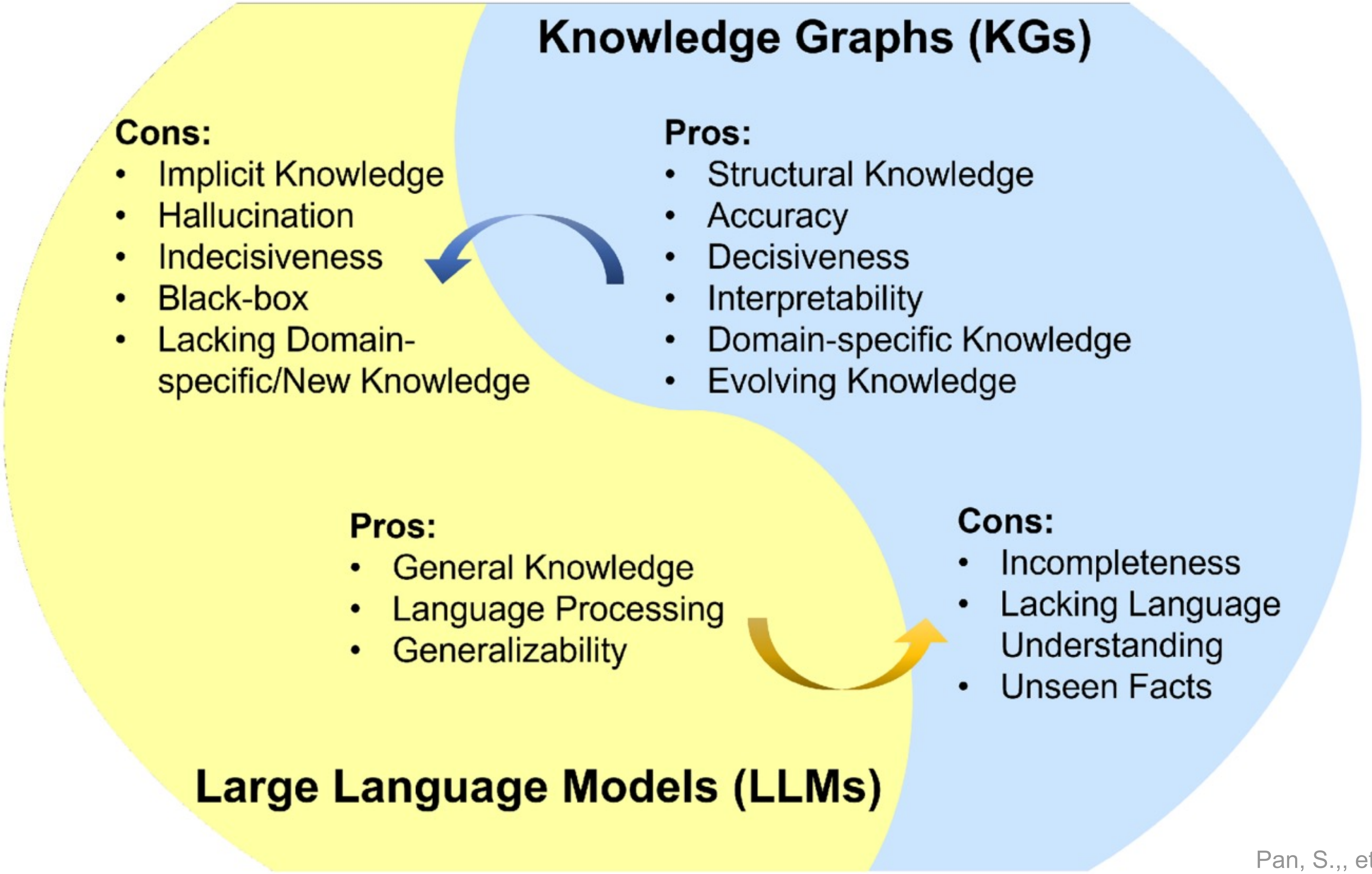
Opportunities and Challenges with LMs

- 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).

Opportunities and Challenges with LMs



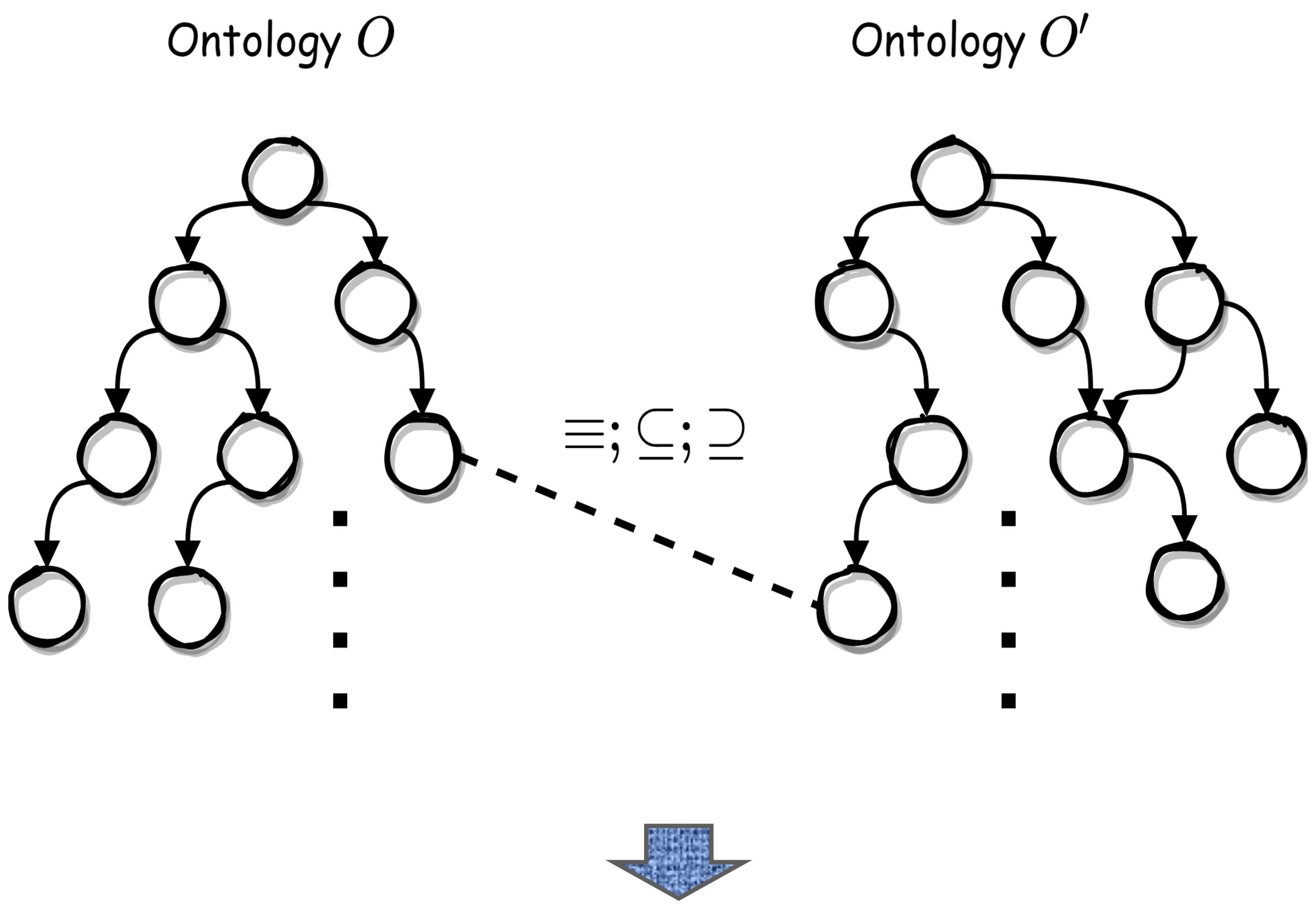
LLMs: Parametric knowledge with uncertainty

KGs/Ontologies: Determined symbolic knowledge

Pan, S.,, et al. "Unifying large language models and knowledge graphs: A roadmap." *IEEE Transactions on Knowledge and Data Engineering* (2024).

BERTMap for Ontology Alignment

- Revisit ontology alignment, a.k.a. Ontology Matching (OM)
  - Systems: LogMap, LogMap-ML, etc.
- Entity:
  - Class
  - Property
  - Instance
- Relationship:
  - Equivalence
  - Subsumption



$$Mapping = \langle e \in O, e' \in O', rel, score \rangle$$

He, Y., et al. "BERTMap: a BERT-based Ontology Alignment System." AAAI 2021.

OM Challenges

- Disambiguation:
  - Naming:
    - E.g., the concept named *muscle layer* in SNOMED-CT is named *muscularis propria* in FMA.
  - Contexts:
    - E.g., in FoodOn, there are two concepts named *mushroom* that are categorized in both *Plant* and *Food.*
- Search space reduction:
  - Traversing all possible mappings takes $O(n^2)$.
- Extreme positive-negative imbalance
- Reference data unavailable

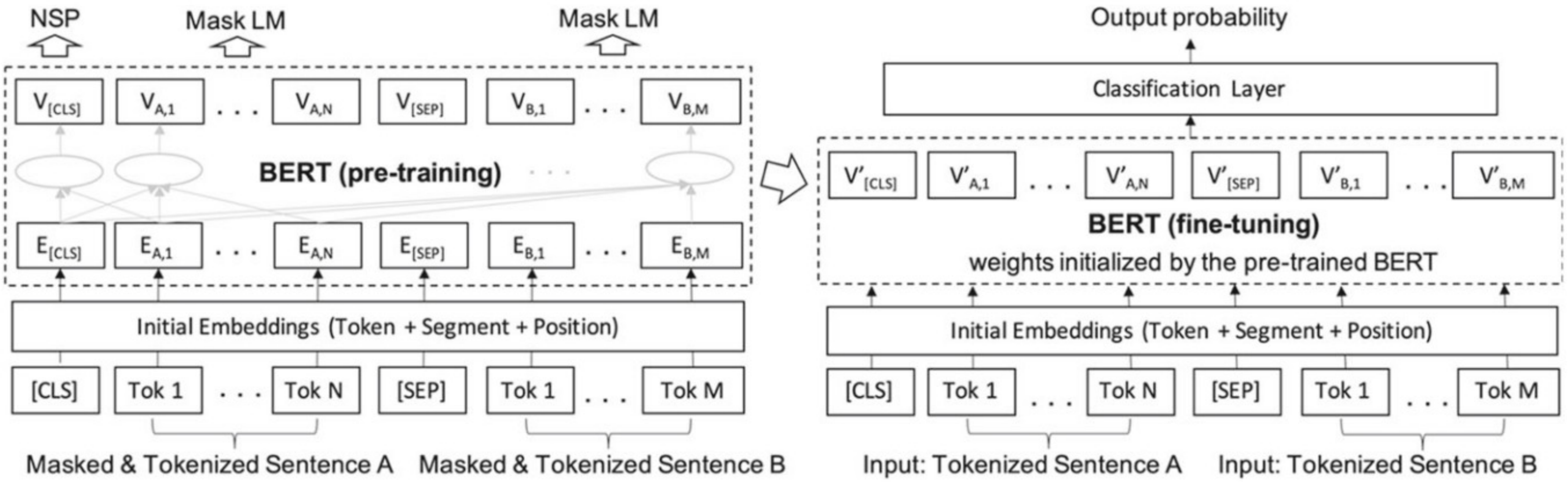
MANCHESTER 1824

BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - Encoder-only
  - BERT can compute contextual embeddings (while Word2Vec is non-contextual)
    - E.g., “The bank robber was seen fishing on the river bank”
  - *Pre-train*
    - Masked token prediction and next sentence prediction
  - *Pre-trained* BERTs is accessible (e.g., from Huggingface) 🤗

Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL 2019*

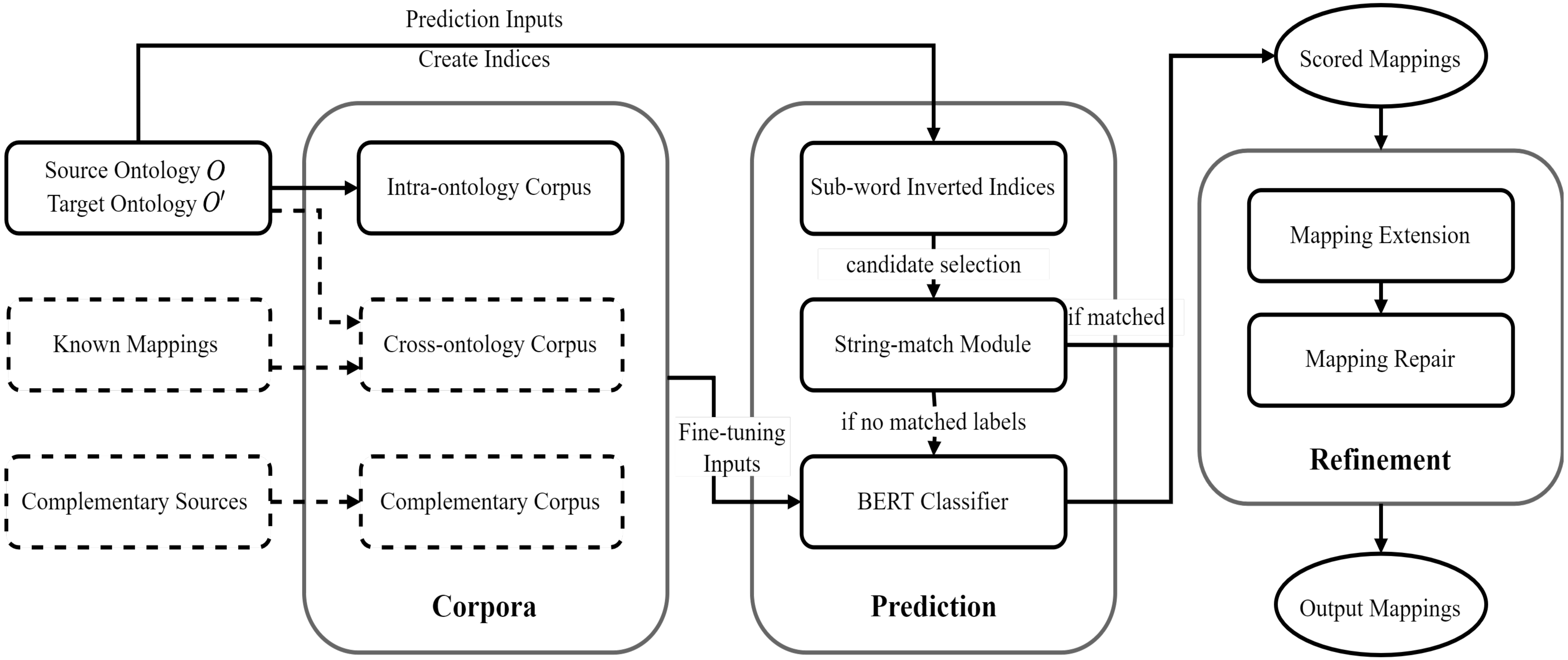
BERT



BERT pre-training

BERT fine-tuning for sentence pair classification

BERTMap System



Collect corpora (pairs of synonyms) as training samples

Fine-tune BERT and predict candidate class pairs

Extend the mappings according to the graph and repair via reasoning

BERT System

Ontologies (and given mappings) → Synonyms & Non-synonyms → Text Semantics Corpora

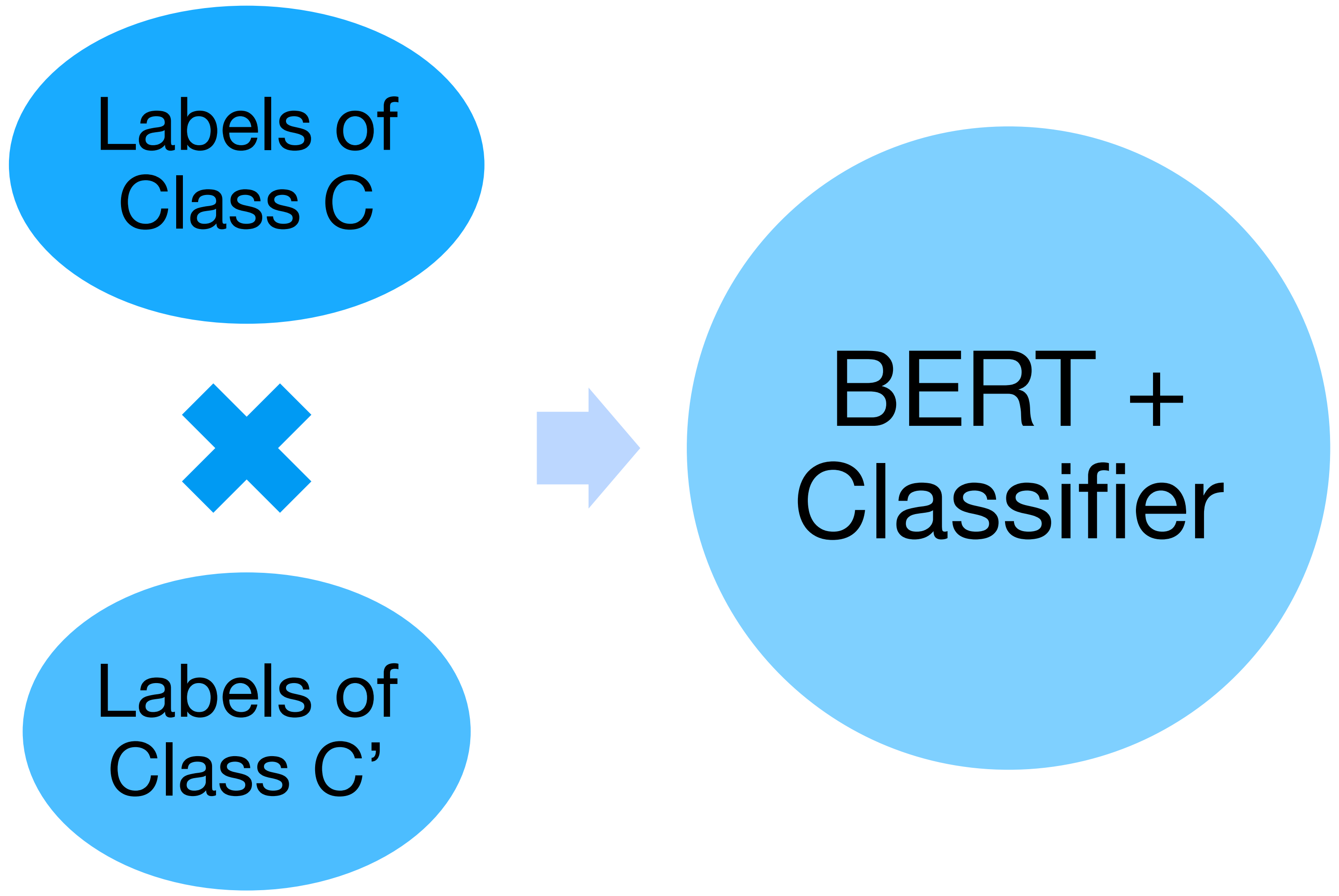
Synonyms: An ontology class could have multiple *aliases* defined by some annotational properties, e.g., *rdfs:label*, *oboInOwl:hasExactSynonym*. Two matched classes have labels of synonyms.

Non-synonyms: retrieved from *either* label pairs of two random classes (*soft*) *or* label pairs of logically disjoint classes (*hard*).

BERT System

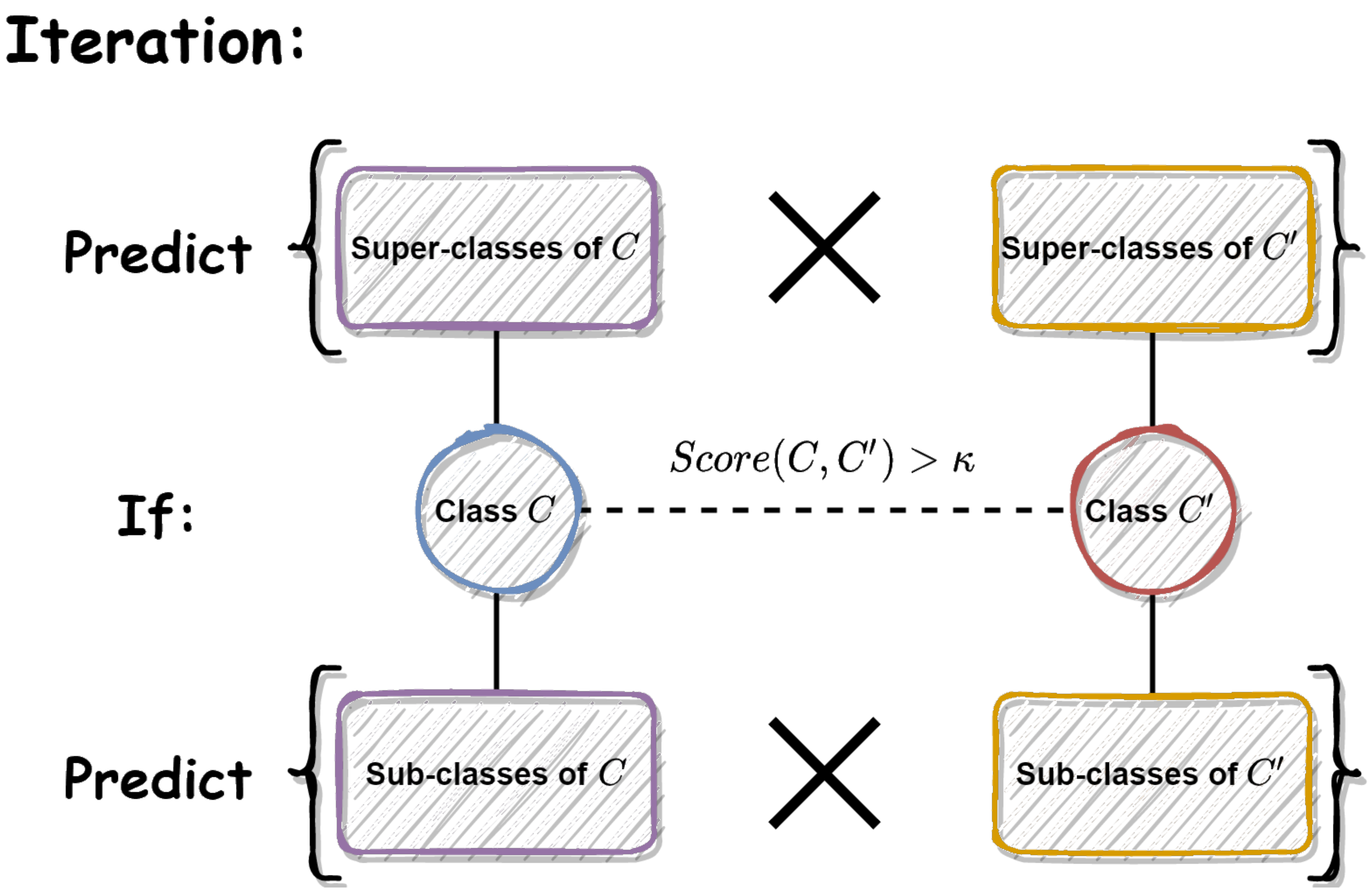
1. Model (BERT + classifier) is fine-tuned with the synonyms and non-synonyms

2. The similarity score for class *C* and *C’* is computed using the **average of the synonym scores** of the paired labels of *C* and *C’*.



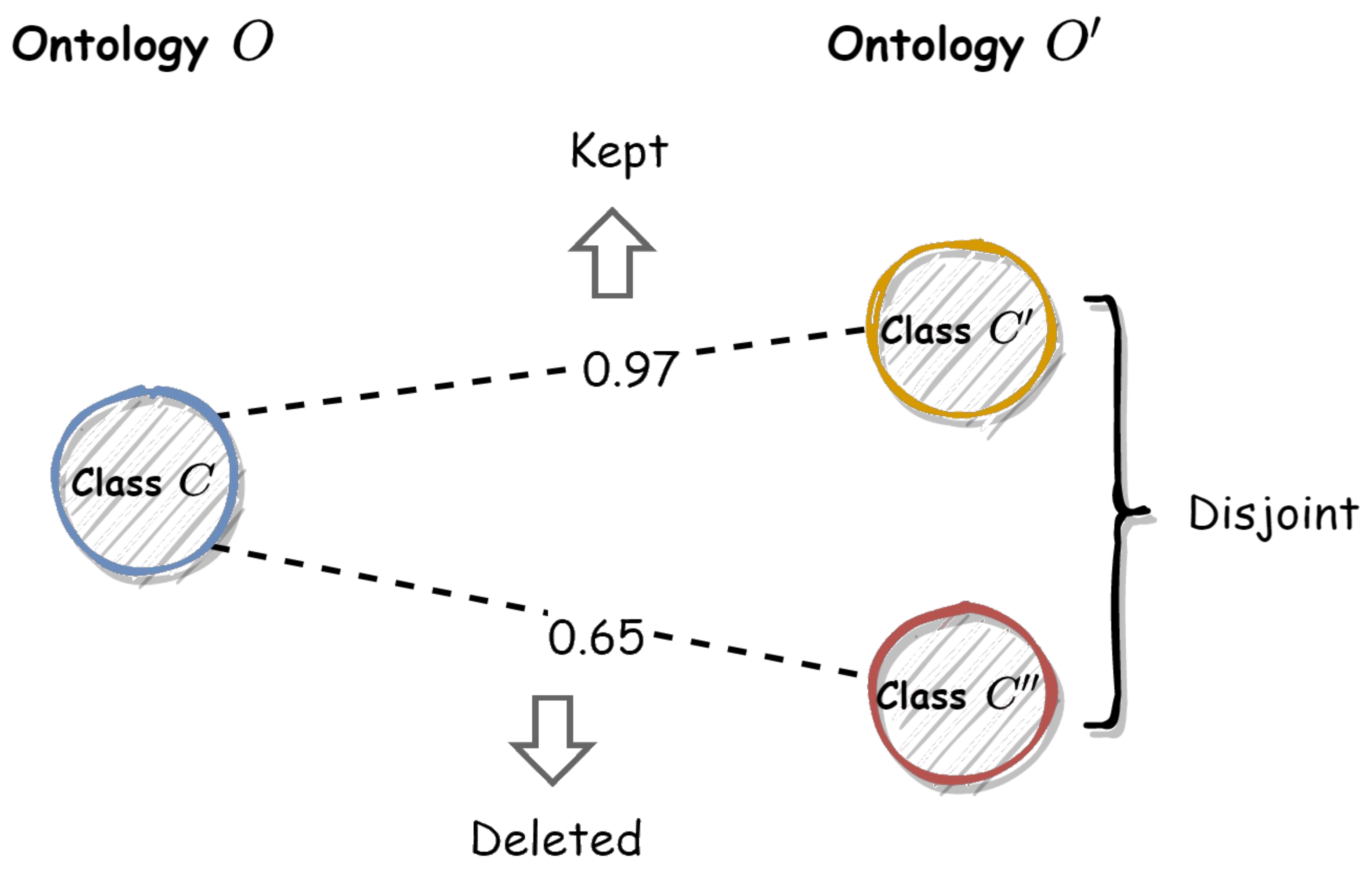
BERT System

- Mapping extension
  - ***Locality Principle***: *Semantically related* classes of the matched classes are likely to be matched.
  - ***Outcome***: To *recall more mappings*, especially the ones that violate the assumption of token sharing (missed in candidate selection based on indices).



BERTMap System

- Mapping repair
  - ***Issue***: Reasoning over each input ontology may lead to *inconsistency* after alignment.
  - ***Solution***: Remove a minimal set of mappings (*a.k.a. diagnosis*) to achieve *consistency,* using (approximate) OWL reasoning
  - ***Outcome****:* To *improve precision* while keeping the recall as much as possible.



(Example of repair)

Jiménez-Ruiz, Ernesto, et al. "Evaluating Mapping Repair Systems with Large Biomedical Ontologies." *Description Logics* 13 (2013): 246-257.

BERTMap Evaluation

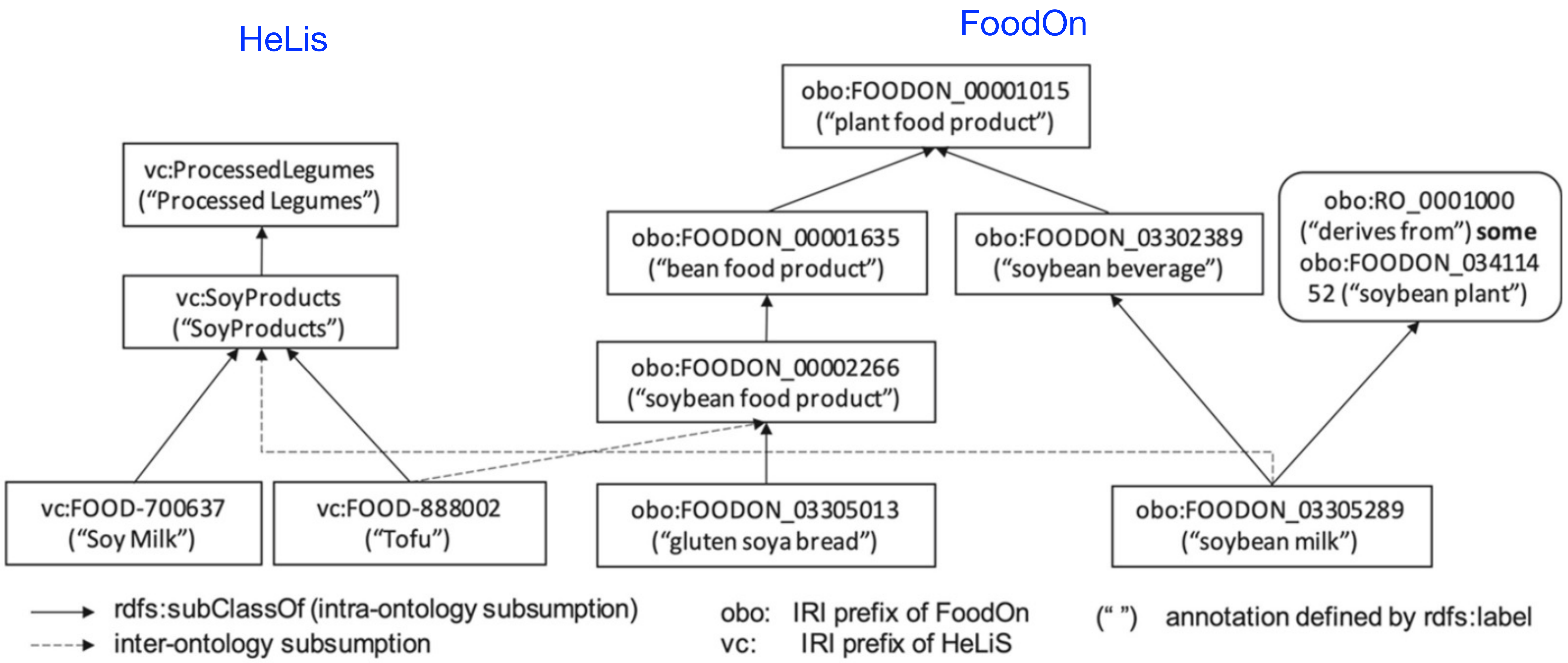
		Hyper-params	Unsupervised			Semi-supervised		
			90% Test Mappings			70% Test Mappings		
	System	$\{\tau, \lambda\}$	Precision	Recall	Macro-F1	Precision	Recall	Macro-F1
Ablations	io	(tgt2src, 0.999)	0.705	0.240	0.359	0.649	0.239	0.350
	io+ids	(tgt2src, 0.999)	0.835	0.347	0.490	0.797	0.346	0.483
	io+cp	(src2tgt, 0.999)	0.917	0.750	0.825	0.895	0.748	0.815
	io+ids+cp	(src2tgt, 0.999)	0.910	0.758	0.827	0.887	0.755	0.816
	io+ids+cp (ex)	(src2tgt, 0.999)	0.896	0.771	0.829	0.869	0.771	0.817
	io+ids+cp (ex+rp)	(src2tgt, 0.999)	0.905	0.771	**0.833**	0.881	0.771	0.822
	io+co	(src2tgt, 0.997)	NA	NA	NA	0.937	0.564	0.704
	io+co+ids	(src2tgt, 0.999)	NA	NA	NA	0.850	0.714	0.776
	io+co+cp	(src2tgt, 0.999)	NA	NA	NA	0.880	0.779	0.826
	io+co+ids+cp	(src2tgt, 0.999)	NA	NA	NA	0.899	0.774	0.832
	io+co+ids+cp (ex)	(src2tgt, 0.999)	NA	NA	NA	0.882	0.787	0.832
	io+co+ids+cp (ex+rp)	(src2tgt, 0.999)	NA	NA	NA	0.892	0.786	**0.836**
Baselines	string-match	(combined, 1.000)	0.987	0.194	0.324	0.983	0.192	0.321
	edit-similarity	(combined, 0.920)	0.971	0.209	0.343	0.963	0.208	0.343
	LogMapLt	NA	0.965	0.206	0.339	0.956	0.204	0.336
	LogMap	NA	0.935	0.685	0.791	0.918	0.681	0.782
	AML	NA	0.892	0.757	0.819	0.865	0.754	0.806
	LogMap-ML*	NA	0.944	0.205	0.337	0.928	0.208	0.340

Results on the SNOMED-FMA task of OAEI
#classes: 3696-6488
#reference mappings: 2686 (10% for validation)

BERTMap Evaluation

- Ablations of BERTMap settings:
  - *Unsupervised* settings can already perform rather well
  - *Semi-supervised* > unsupervised
  - The *complementary corpus* is extremely useful
  - Mapping *extension* and *repair* consistently boost the performance
- Comparisons to baseline models:
  - BERTMap > LogMap, AML and LogMap-ML

BERTSubs for Subsumption Prediction

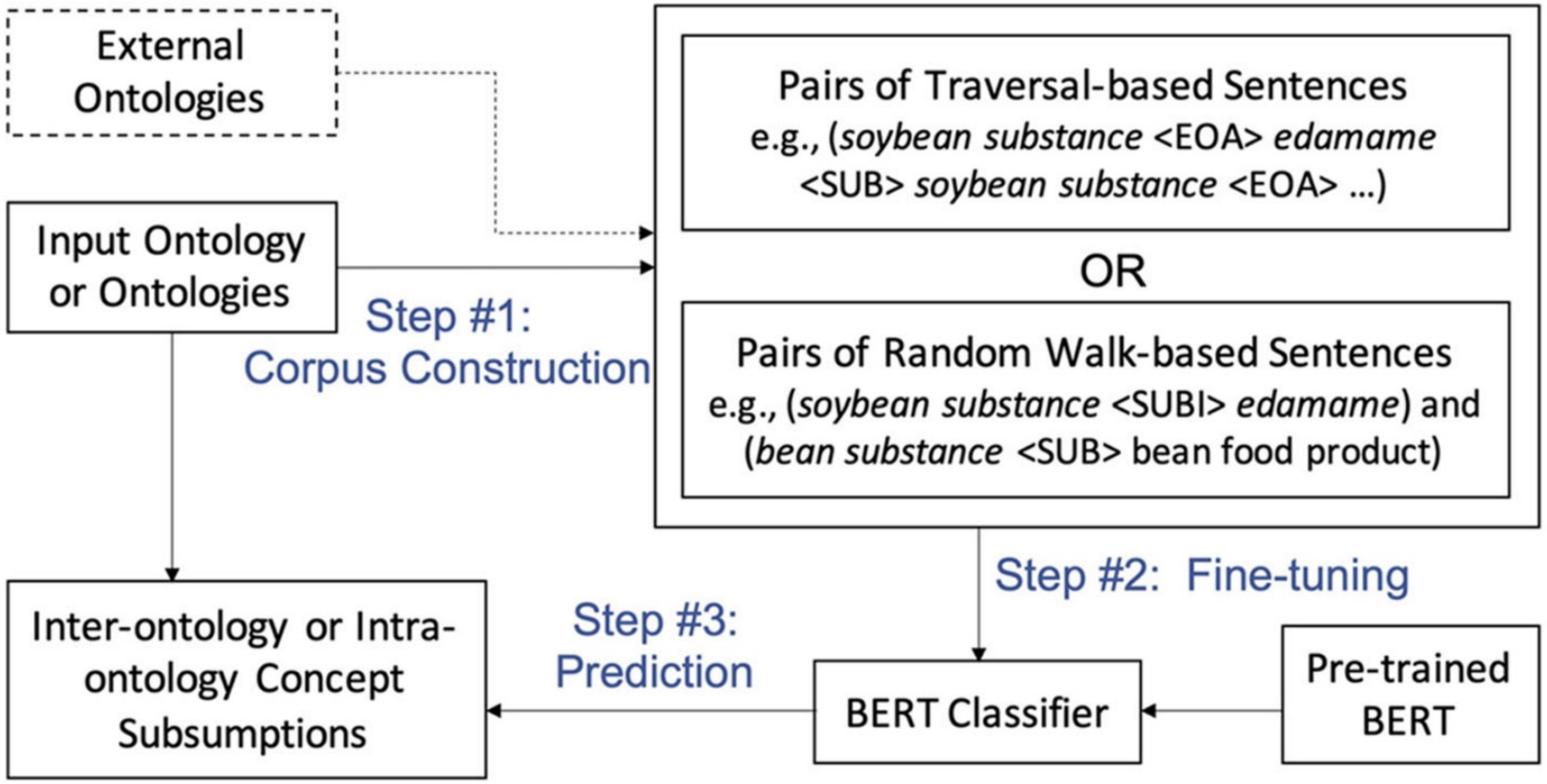


Example of inter-ontology and intra-ontology **class subsumptions**
Super class could be complex class (logical expressions)

Chen, J., et al. "Contextual semantic embeddings for ontology subsumption prediction." *World Wide Web* (2023): 1-23.

BERTSubs Framework

- Fine-tune a BERT model with subsumptions in the given ontologies
- Different templates as input for utilizing the context
    - Class label alone
    - Class path
    - Class context (breadth first search)



BERTSubs Evaluation

- Similar setting as ontology completion of OWL2Vec*
  - Ranking-based metrics
    - MRR, Hits@K
  - Negative candidates
    - Neighboring concepts of the ground truth
    - or existential restrictions sharing relation or class as the ground truth

Method	FoodOn				GO			
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
TransE	0.479	0.332	0.654	0.816	0.320	0.192	0.444	0.605
TransR	0.508	0.367	0.674	0.827	0.354	0.218	0.497	0.647
DistMult	0.509	0.369	0.678	0.821	0.344	0.216	0.471	0.612
HAKE	0.488	0.349	0.658	0.800	0.416	0.295	0.541	0.654
Text-aware TransE	0.572	0.429	0.734	0.869	0.518	0.357	0.718	0.863
Text-aware^{+} TransE	0.567	0.434	0.730	0.860	0.515	0.354	0.716	0.856
Word2Vec	0.562	0.426	0.717	0.866	0.416	0.284	0.549	0.721
Onto2Vec	0.591	0.451	0.762	0.875	0.428	0.291	0.570	0.751
OPA2Vec	0.607	0.464	0.782	0.892	0.434	0.294	0.585	0.760
OWL2Vec*	0.628	0.502	0.797	0.900	0.462	0.328	0.596	0.787
BERTSubs (IC)	0.635	0.483	**0.832**	0.931	0.586	0.408	**0.825**	**0.937**
BERTSubs (PC)	**0.636**	**0.491**	0.829	0.932	**0.606**	**0.453**	0.806	0.927
BERTSubs (BC)	0.618	0.459	0.824	**0.935**	0.578	0.429	0.767	0.907

Results of intra-ontology named subsumption prediction

BERTSubs Evaluation

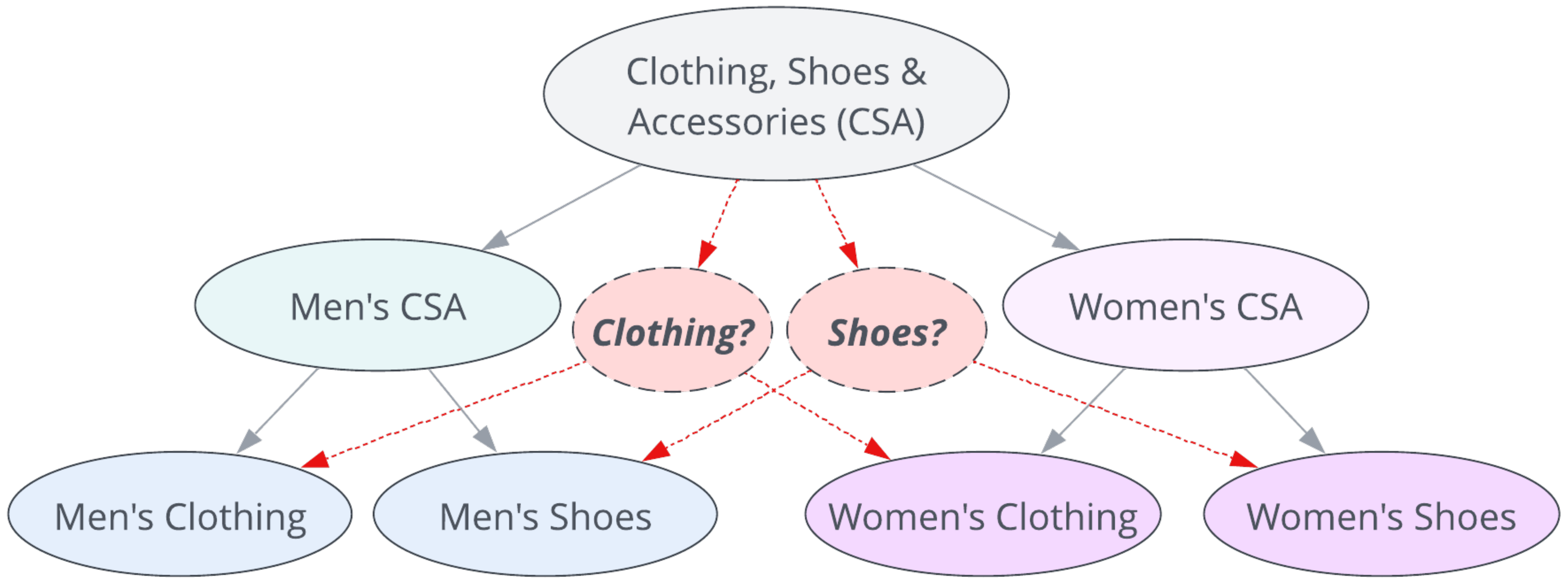
Method	NCIT-DOID				HeLiS-FoodOn			
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
Word2Vec	0.444	0.320	0.575	0.722	0.541	0.415	0.712	0.810
Onto2Vec	0.485	0.351	0.637	0.784	0.592	0.465	0.725	0.842
OPA2Vec	0.488	0.367	0.641	0.784	0.588	0.449	0.731	0.870
OWL2Vec*	0.506	0.378	0.650	0.784	0.610	0.501	0.753	0.839
BERTSubs (IC)	0.695	0.574	0.854	0.935	0.619	0.449	0.858	0.936
BERTSubs (PC)	0.707	0.588	0.863	0.934	0.629	0.481	0.842	0.927
BERTSubs (BC)	0.693	0.565	0.851	0.929	0.589	0.440	0.767	0.875

Results of inter-ontology named subsumption prediction

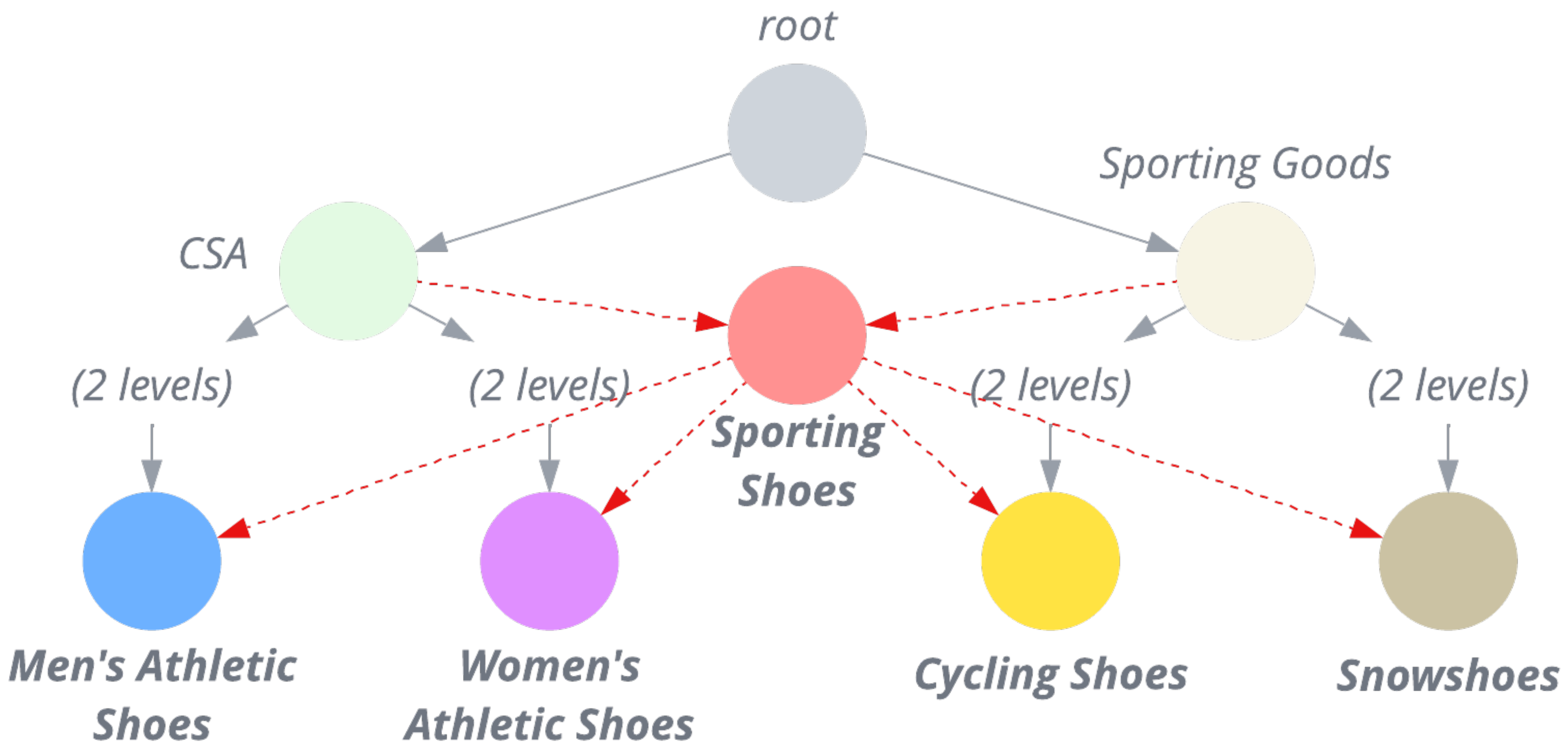
ICON for Implicit Concept Insertion

TAXONOMY CURATION
ICON
IMPLICIT CONCEPT INSERTION

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



Example 1: Concepts that should have existed

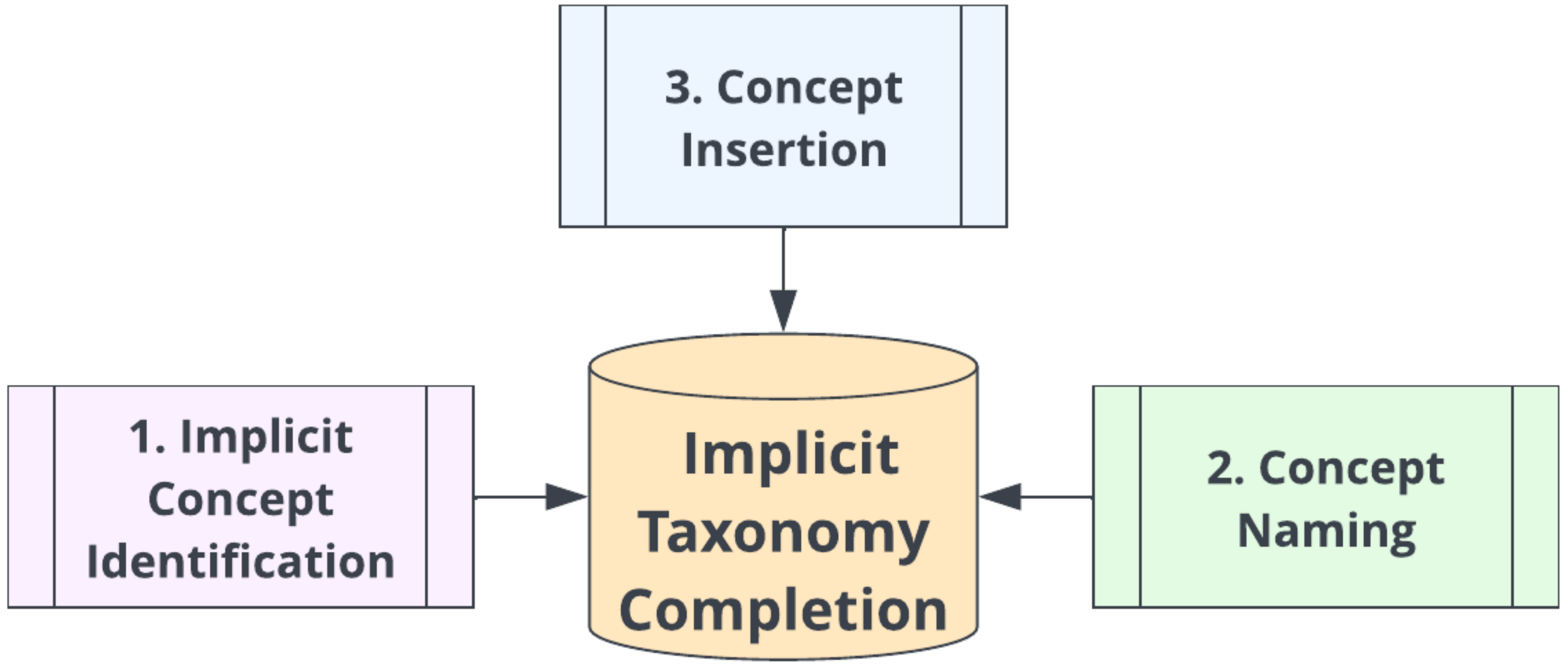


Example 2: Concepts bridging multiple branches of the taxonomy

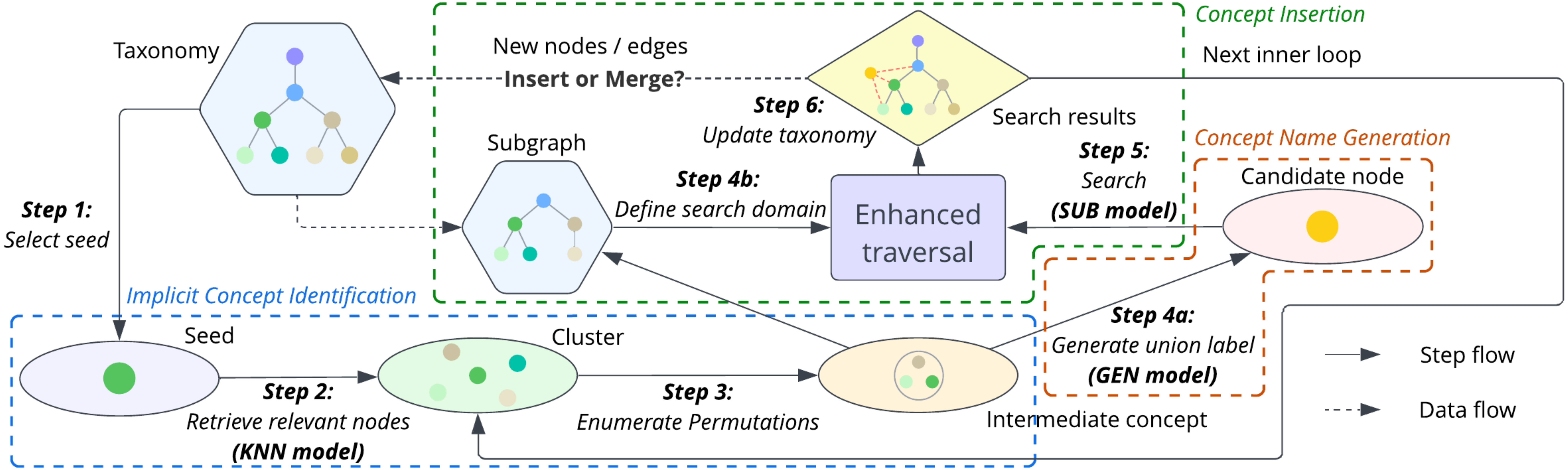
Shi, J., et al. ”Taxonomy Completion via Implicit Concept Insertion." *The Web Conference* 2024.

ICON Framework

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



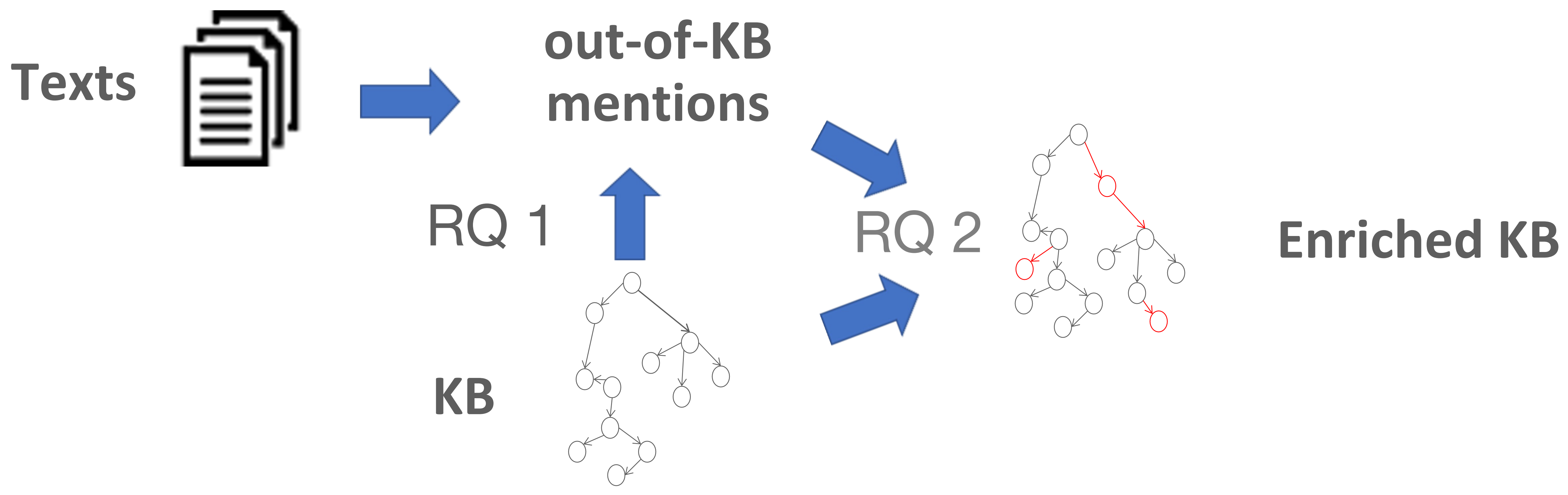
ICON Framework



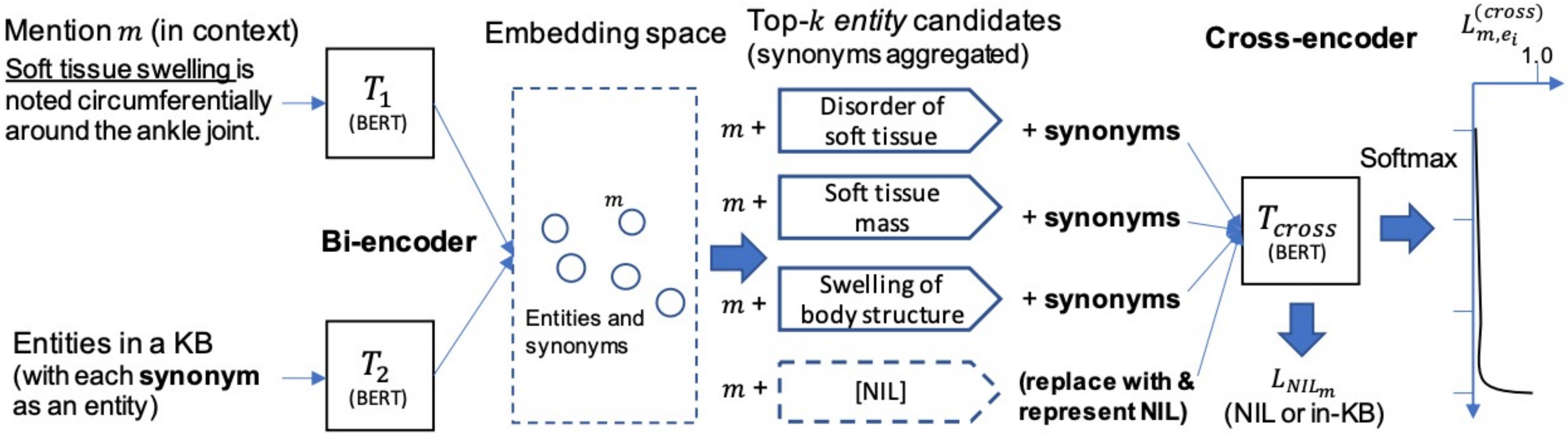
An iterative workflow with the outer loop (steps 1 & 2) and the inner loop (steps 3-6)

New Concept 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?
  - A.k.a. entity linking with NIL
- RQ2: How to insert out-of-KB mentions as new entities into a Knowledge Base?



BLINKOut for Entity Linking with NIL

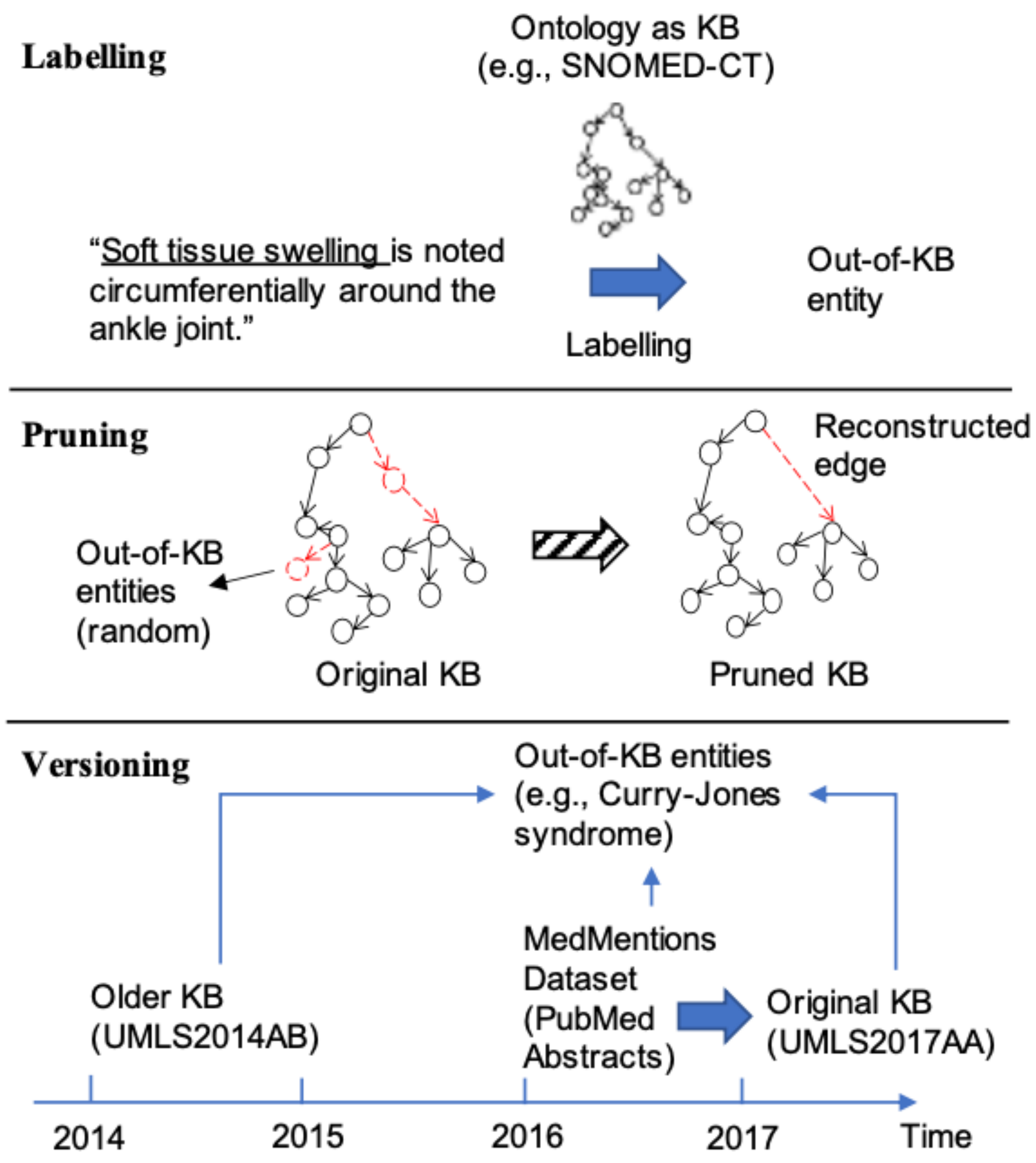


The architecture of BLINKOut includes

- A bi-encoder based on BERT and contrastive learning for ranking candidate entities
- A cross-encoder for classification of candidate entities (including NIL)

Dong, H., et al. "Reveal the Unknown: Out-of-Knowledge-Base Mention Discovery with Entity Linking." *CIKM 2023*.

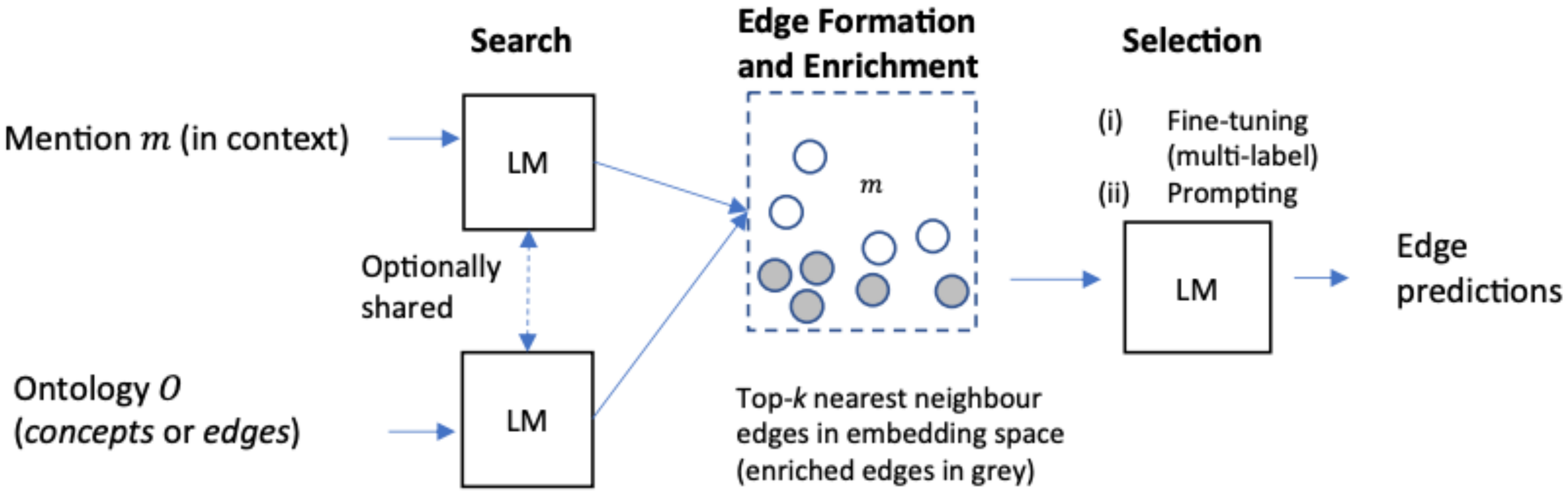
BLINKOut Evaluation Datasets



← Three strategies for datasets construction

Insertion of New Concept from Text

- Problem:
  - Given a new mention from the text, find out an edge -- its parent (named or complex class) and its child in the ontology for insertion
- Similar architecture, but:
  - Search for edges & concepts with the bi-encoder
  - Enrich the edges via the graph
  - Use LLM & prompts or fine-tuning for the cross-encoder



Dong, H., et al. "A Language Model based Framework for New Concept Placement in Ontologies." ESWC 2024.

DeepOnto: A Library for Ontology Engineering

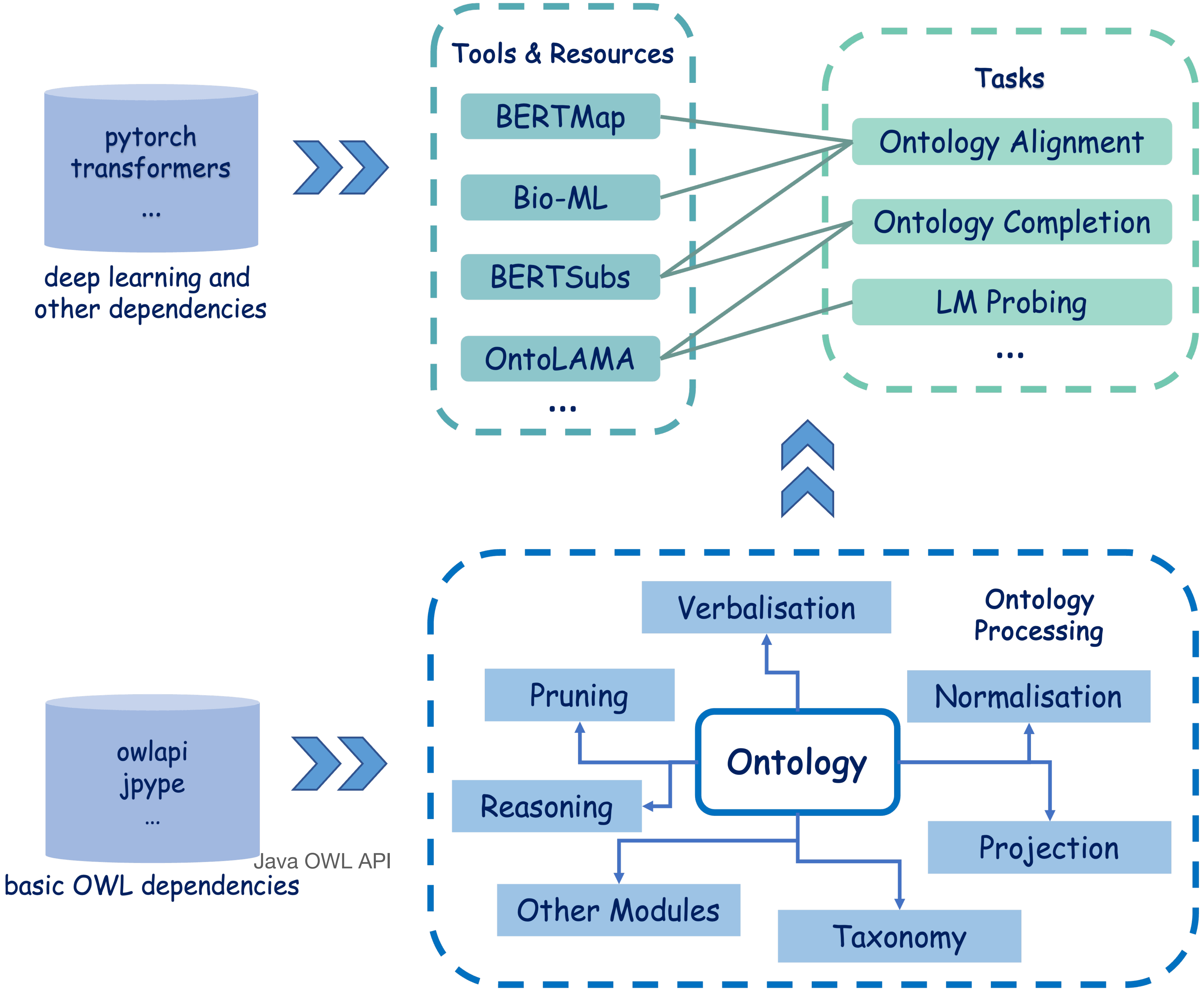
DeepOnto

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

- Python interface for **more compact interaction with deep learning libraries**
- Ontology processing APIs for **fostering deep learning and NLP techniques in ontology engineering**
- Ontology engineering **tools and resources implemented with our APIs, deep learning and LMs**

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

DeepOnto Framework



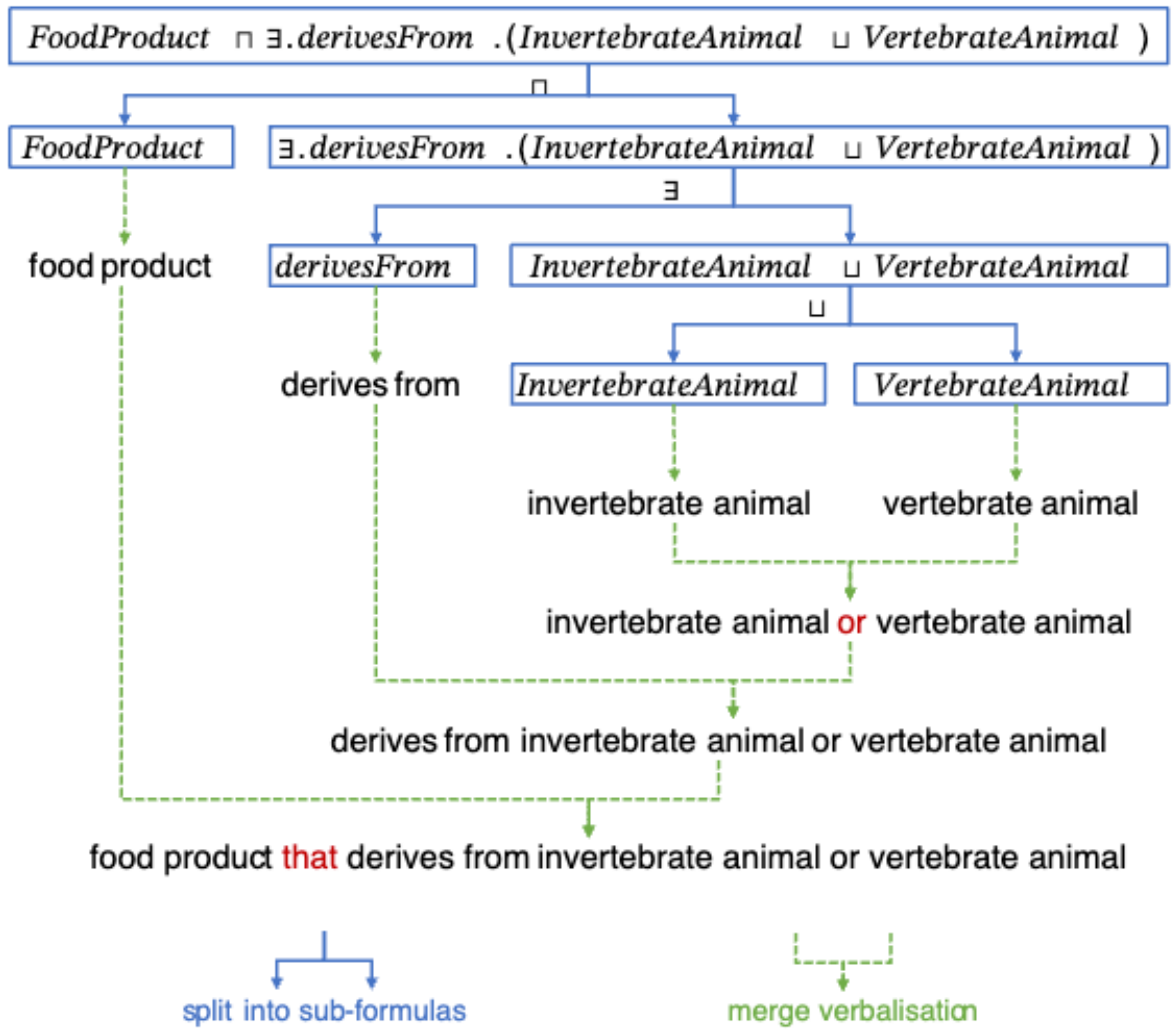
← BERTMap, BERTSubs

← Bio-ML: resources for evaluating ML-based ontology alignment systems

← OntoLAMA: probing the knowledge and reasoning capabilities of LMs

← these functions have been introduced in our previous lecturers

DeepOnto (Verbalization)



← An example of transforming a complex class into natural language description

DeepOnto Document

DeepOnto | Search | KRR-Oxford/DeepOnto v0.9.1 176 11

GET STARTED
- Introduction
- Load an Ontology
  - Loading Ontology
  - Acessing Ontology Entities
  - Ontology Reasoning
  - Feature Requests
- Changelog
- FAQs

TUTORIALS
- Verbalise Complex Ontology Concepts
- Ontology Matching with BERTMap Family
- Bio-ML: A Comprehensive Documentation
- Subsumption Inference with BERTSubs
- OntoLAMA: Dataset Overview and Usage Guide

PACKAGE REFERENCE
- Ontology Processing
- Ontology Alignment
- Ontology Completion
- Utilities

Basic Usage of Ontology

DeepOnto extends from the OWLAPI and implements many useful methods for ontology processing and reasoning, integrated in the base class `Ontology`.

This page gives typical examples of how to use `Ontology`. There are other more specific usages, please refer to the documentation by clicking `Ontology`.

Loading Ontology

`Ontology` can be easily loaded from a local ontology file by its path:

```
from deeponto.onto import Ontology
```

Importing `Ontology` will require the setting of `JAVA_HOME` environment variable if it does not find the JVM, and JVM memory allocation (defaults to `8g`; if `nohup` is used to run the program in the backend, use `nohup echo "8g" | python command`):

```
Please enter the maximum memory located to JVM: [8g]: 16g

16g maximum memory allocated to JVM.
JVM started successfully.
```

Loading an ontology from a local file:

```
onto = Ontology("path_to_ontology.owl")
```

It also possible to choose a reasoner to be used:

```
onto = Ontology("path_to_ontology.owl", "hermit")
```

Tip

For faster (but incomplete) reasoning over larger ontologies, choose a reasoner like `"elk"`.

Hands on tutorial and document

Day 5 Discussion & Outlook

Augmenting LLMs

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

Symbolic and neural knowledge representations

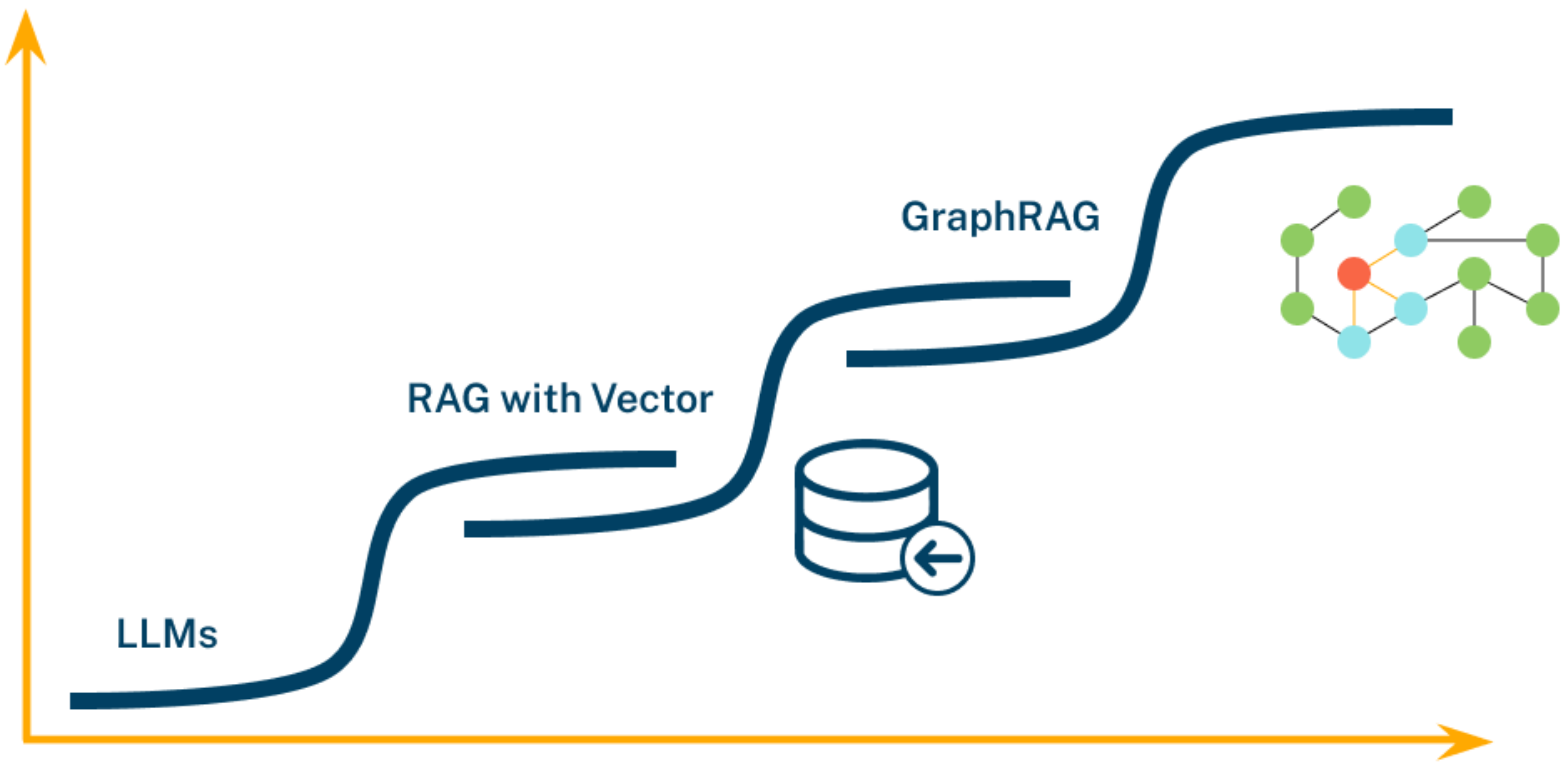
LM Pre-train / Fine-tune for retrieval

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
- …

From RAG to GraphRAG



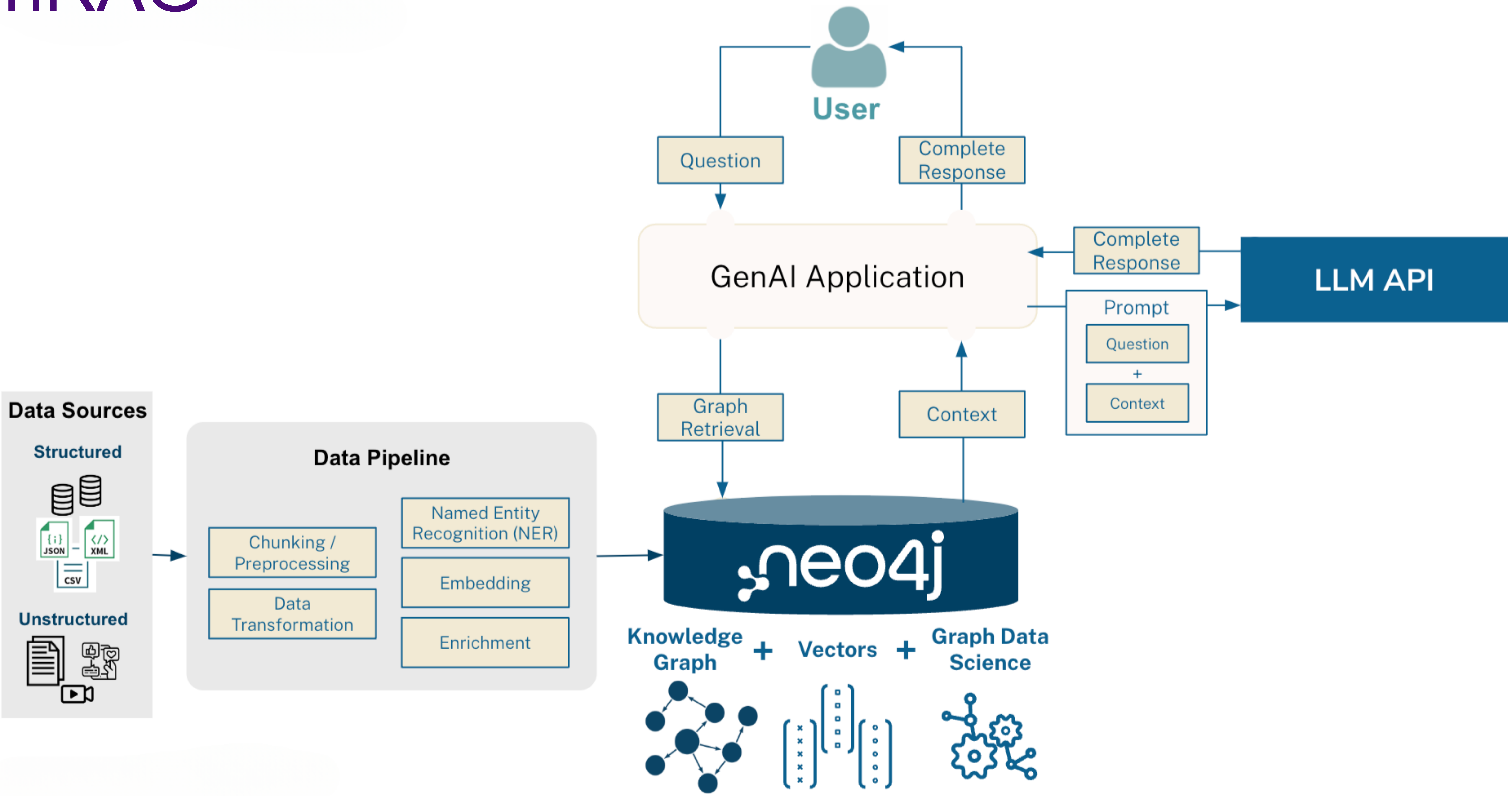
From https://neo4j.com/blog/graphrag-manifesto/

Vector-based RAG: Retrieval based on the similar of embeddings

GraphRAG: RAG with the Retrieval path including a (knowledge) graph

- Higher accuracy & more useful answers (running time)
- Improved data understanding, faster iteration (development)
- Explainability, security, access control, etc. (governance)

GraphRAG



From https://neo4j.com/blog/graphrag-manifesto/

Challenges & Discussion

- Knowledge representation for LLM and RAG
  - How to represent heterogeneous data and knowledge for supporting retrieval?
    - Transformation, linking, embedding, …
  - How to formally manage the evidences?
    - Integration, consistency checking, entailment, …
  - How to understand the inference of LLM?
    - Benchmark, attribution/explainability, …
  - How to turn LLMs for reasoning?
    - Instruct tuning, alignment, regularization, human guidance, …

Challenges & Discussion

- Knowledge representation with vectors and parameters
  - What kind of complex knowledge can be efficiently represented in the Euclidean (or non-Euclidean) space? And how?
    - The full semantics of OWL 2? And with literals?
  - Can we embed ontologies and KGs with LLM parameters, or a mixture of vectors and LLM parameters?
    - Formal semantics by vectors in a geometric space and literals by LLM parameters?

Challenges & Discussion

- Knowledge engineering in the LLM era
  - How to manage a mixture of multi-modal data and knowledge?
    - Ontology, KG, Data Lake, Image, Vectors, Parameters, …
  - More tools for knowledge and data curation?
    - Schema inference, KG/ontology completion, Integration, …

Summary

- Background: (Large) LMs
- Knowledge Representation and engineering with LMs
  - BERTMap, BERTSubs
  - ICON, BLINKOut
  - DeepOnto (system)
- Outlook
  - RAG, GraphRAG

Q&A

The End of Day 5