OntoLM Tutorial@ISWC 2026

The 25th International Semantic Web Conference

Bari, Italy



Chapter 4: Ontology Alignment with DeepOnto

Jiaoyan Chen

MANCHESTER 1824
The University of Manchester

DeepOnto: A Python Package for Ontology Engineering with Deep Learning

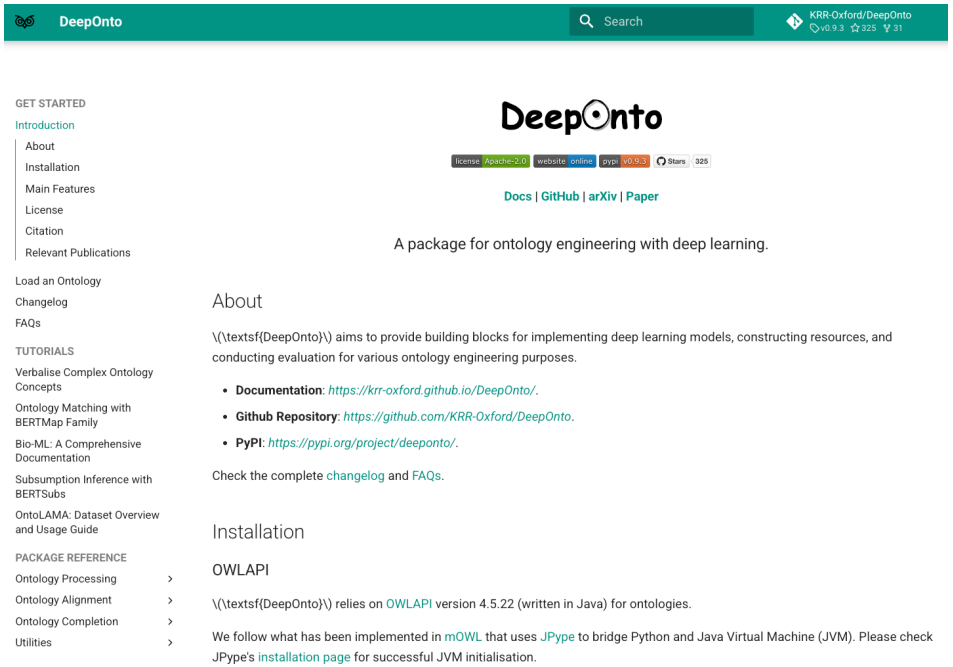
Yuan He [a], Jiaoyan Chen [b,a], Hang Dong [a], Ian Horrocks [a], Carlo Allocca [c], Taehun Kim [c] and Brahmananda Sapkota [c]

[a] *Department of Computer Science, University of Oxford, UK*
E-mails: yuan.he@cs.ox.ac.uk, hang.dong@cs.ox.ac.uk, ian.horrocks@cs.ox.ac.uk
[b] *Department of Computer Science, The University of Manchester,UK*
E-mail: jiaoyan.chen@manchester.ac.uk
[c] *Samsung Research, UK*
E-mails: c.allocca@samsung.com, taehun11.kim@samsung.com, b.sapkota@samsung.com

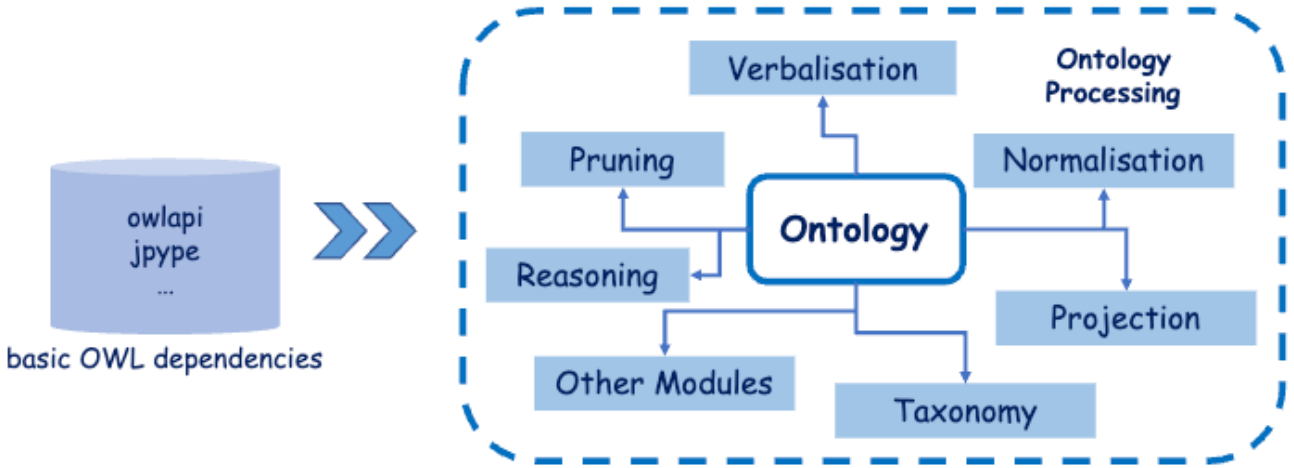
Abstract. Applying deep learning techniques, particularly language models (LMs), in ontology engineering has raised widespread attention. However, deep learning frameworks like PyTorch and Tensorflow are predominantly developed for Python programming, while widely-used ontology APIs, such as the OWL API and Jena, are primarily Java-based. To facilitate seamless integration of these frameworks and APIs, we present DeepOnto, a Python package designed for ontology engineering. The package encompasses a core ontology processing module founded on the widely-recognised and reliable OWL API, encapsulating its fundamental features in a more "Pythonic" manner and extending its capabilities to include other essential components including reasoning, verbalisation, normalisation, projection, and more. Building on this module, DeepOnto offers a suite of tools, resources, and algorithms that support various ontology engineering tasks, such as ontology alignment and completion, by harnessing deep learning methodologies, primarily pre-trained LMs. In this paper, we also demonstrate the practical utility of DeepOnto through two use-cases: the Digital Health Coaching in Samsung Research UK and the Bio-ML track of the Ontology Alignment Evaluation Initiative (OAEI).

Keywords: Ontology, Ontology Engineering, Deep Learning, Language Model, OWL, Python

Repository: https://github.com/KRR-Oxford/DeepOnto
Documentation: https://krr-oxford.github.io/DeepOnto/
License: Apache License, Version 2.0



DeepOnto Architecture



Ontolog Processing: Recursive Verbalisation

To use the verbaliser, do the following:

```
from deeponto.onto import Ontology, OntologyVerbaliser

# load an ontology and init the verbaliser
onto = Ontology("some_ontology_file.owl")
verbaliser = OntologyVerbaliser(onto)
```

To verbalise a complex concept expression:

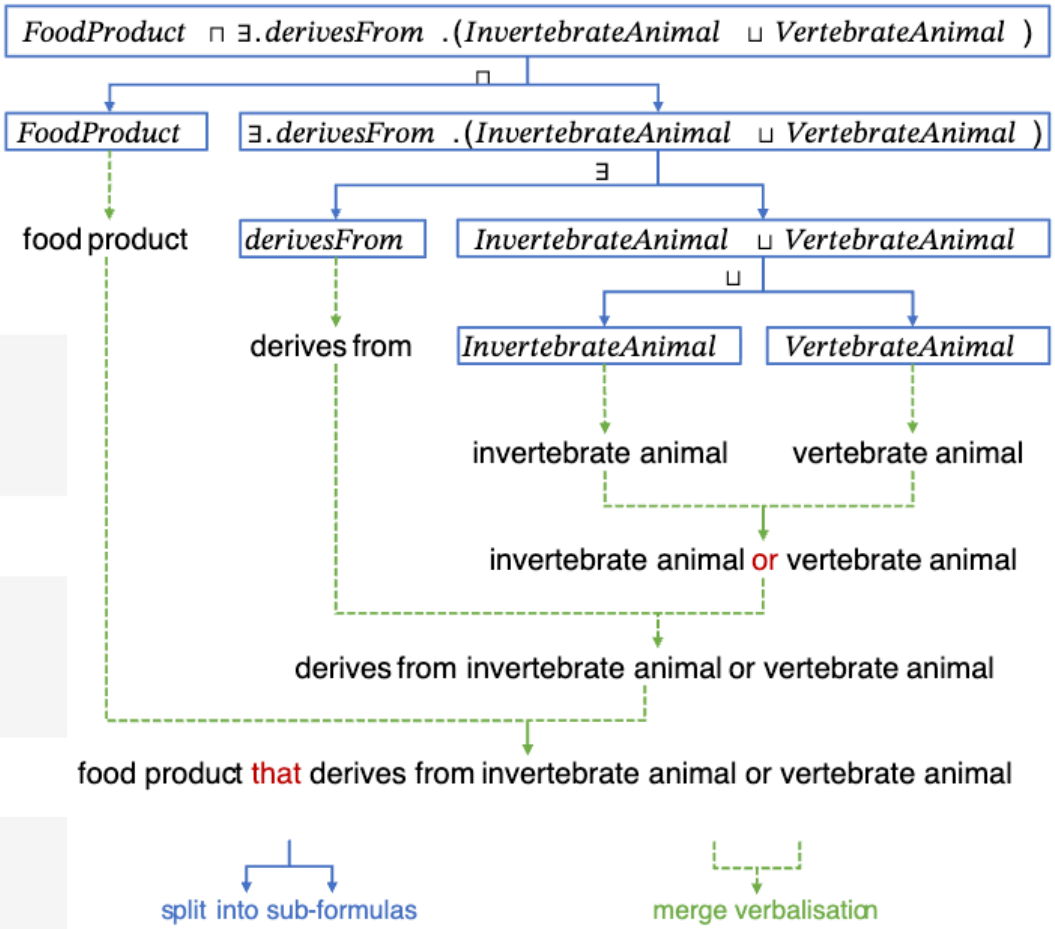
```
# get complex concepts asserted in the ontology
complex_concepts = list(onto.get_asserted_complex_classes())

# verbalise the first complex concept
v_concept = verbaliser.verbalise_class_expression(complex_concepts[0])
```

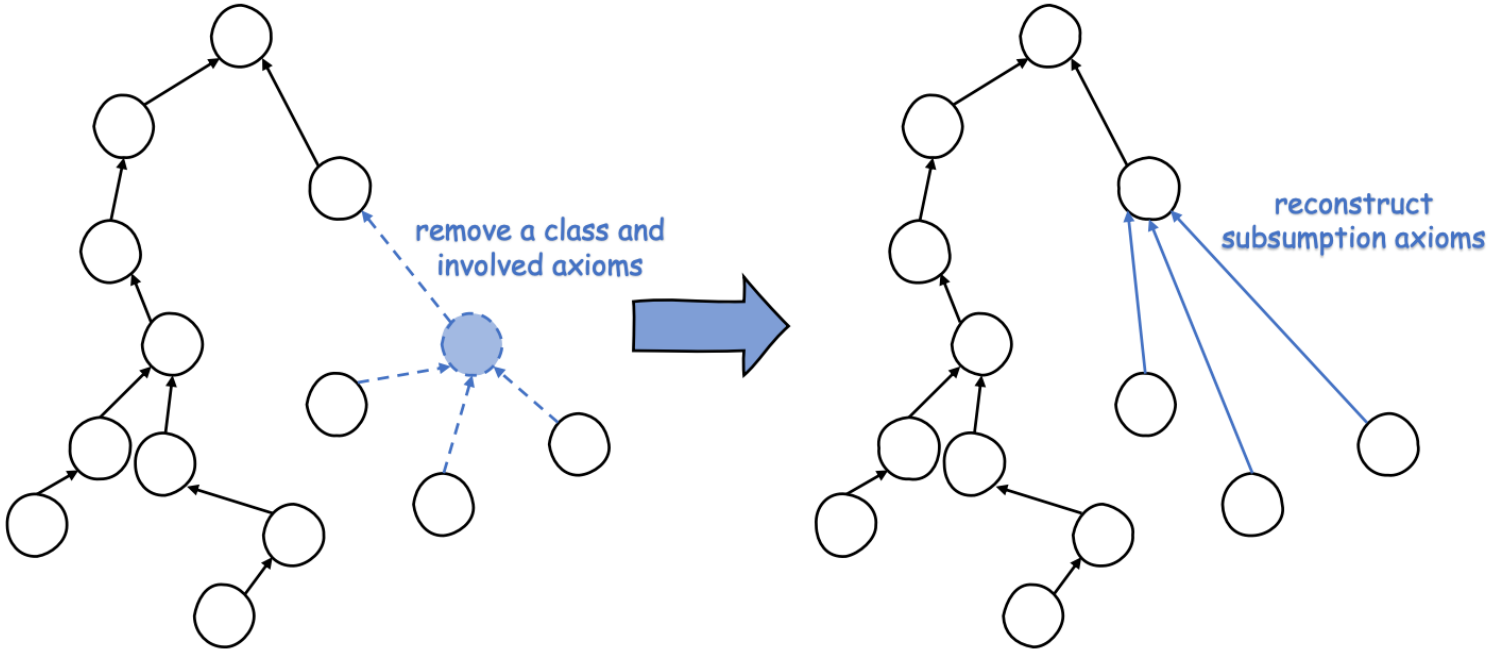
To verbaliser a class subsumption axiom:

```
# get subsumption axioms from the ontology
subsumption_axioms = onto.get_subsumption_axioms(entity_type="Classes")

# verbalise the first subsumption axiom
v_sub, v_super = verbaliser.verbalise_class_subsumption_axiom(subsumption_axioms[0])
```



Ontolog Processing: Pruning

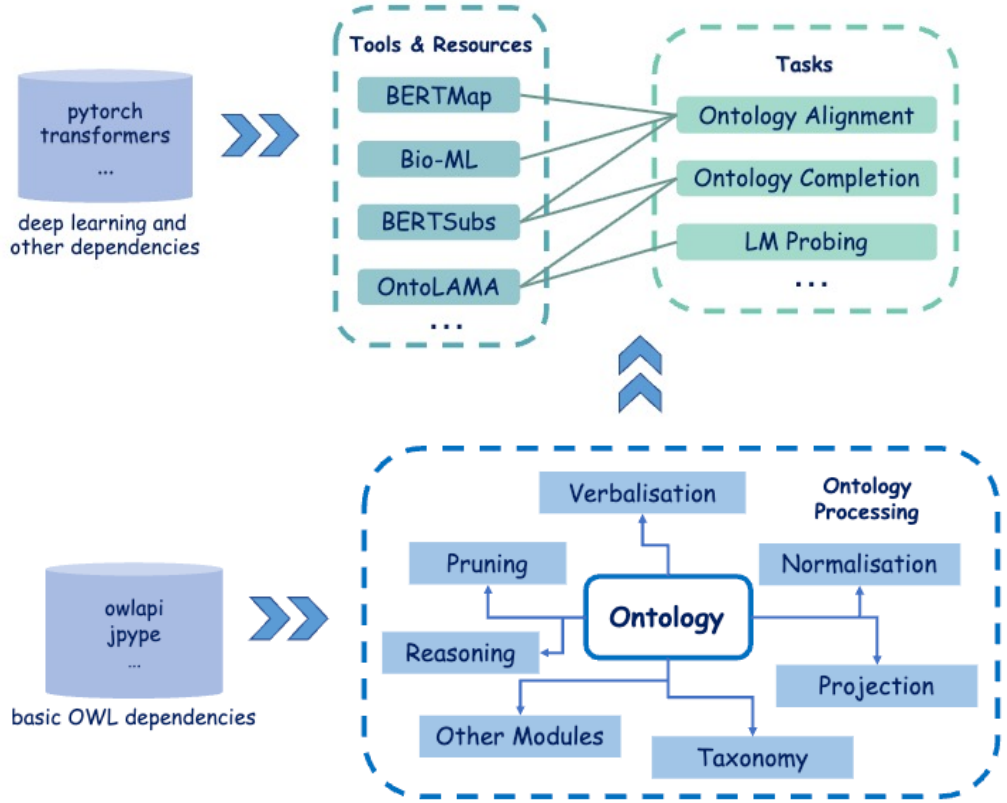


Ontology Processing

- Reasoning: HermiT, ELK
- Projection: OWL ontology into RDF graph with approximation ($C \sqsubseteq \exists r.D \rightarrow (C, r, D)$
- Taxonomy: subsumption between named concepts, asserted subsumptions & inferred subsumptions
- Normalisation (for EL ontologies):
  - Equivalently transform axioms into basic forms:

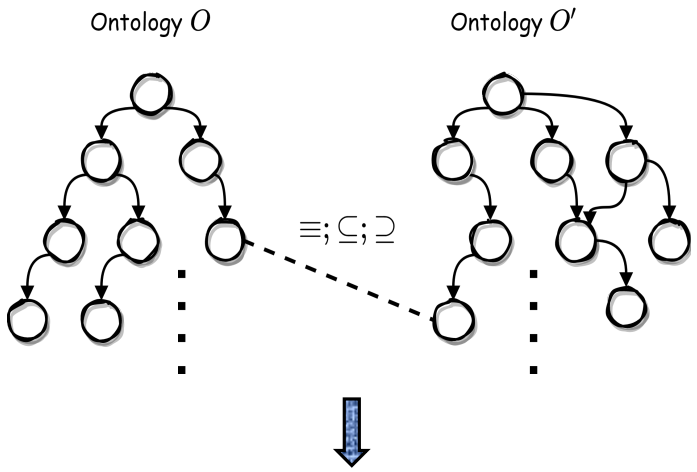
$$C \sqsubseteq D, C \sqcap C' \sqsubseteq D, C \sqsubseteq \exists r.D, \exists r.C \sqsubseteq D, r \sqsubseteq s, \text{and } r \circ r' \sqsubseteq s,$$

DeepOnto Architecture



Ontology Alignment (or Matching)

- Entity:
  - Class
  - Property
  - Instance
- Relationship:
  - Equivalence
  - Subsumption



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

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 $O(n^2)$.

He, Yuan, et al. "BERTMap: a BERT-based ontology alignment system." AAAI 2022

OM Challenges

- Extreme **positive-negative imbalance**:
  - Matched entity pairs are often of ***several orders fewer*** than unmatched ones.
  - High-quality reference data is ***often not available***, e.g., UMLS is considered as a silver standard (not necessarily *complete* or *correct*). This induces two further issues:
    - Results of automatic evaluation ***lack reliability***.
    - Supervised learning setting is often ***unfeasible***.

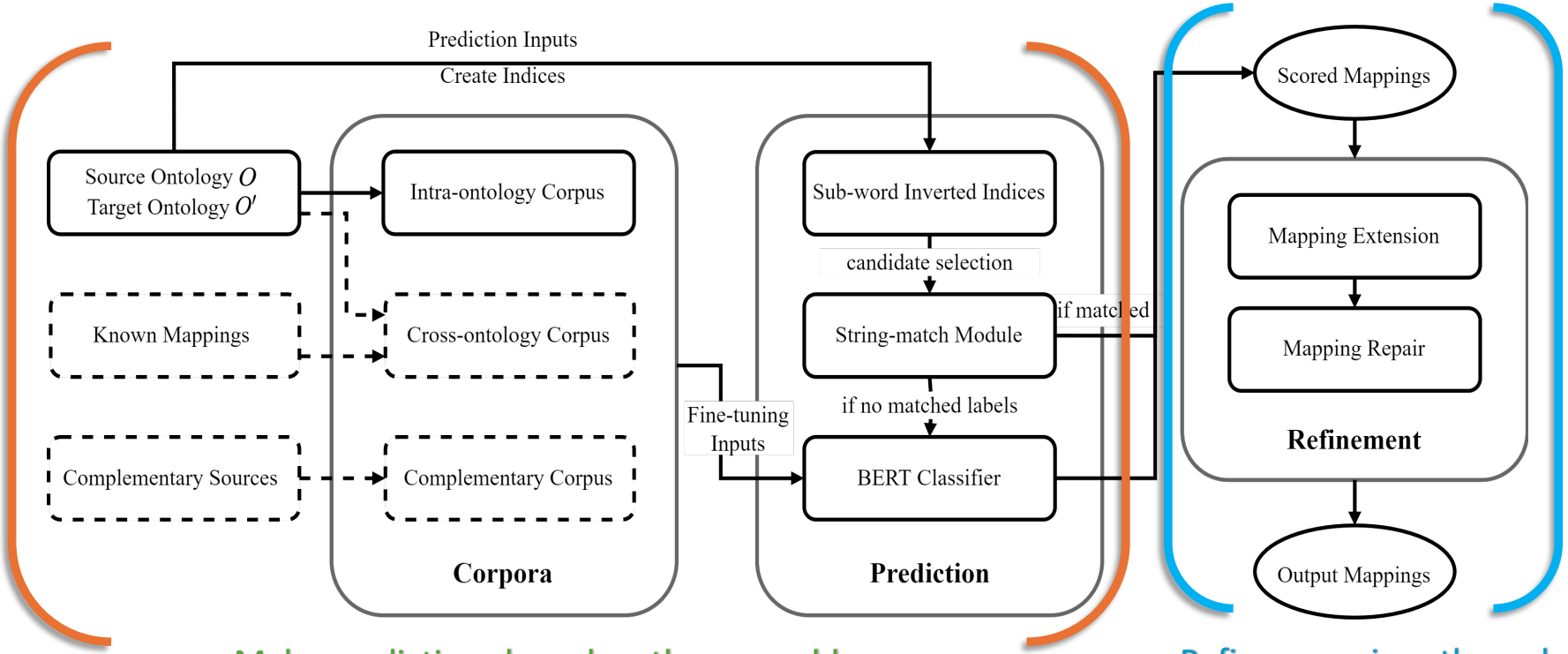
BERT (LM Encoder)

- **B**idirectional **E**ncoder Representations from **T**ransformers:
  - **BERT** can compute dynamic (contextual) embeddings whereas the traditional word embedding systems (e.g., **Word2Vec**, **GloVe**) yield the static (*unified*) word vectors. For example,

> The bank robber was seen fishing on the river bank.

  - *Pre-train & Fine-tune* is a popular learning scheme for language models in NLP.
  - *Pre-trained* BERTs are widely accessible; often, only *fine-tuning* is needed and it takes *2-4 epochs* to achieve convergence.

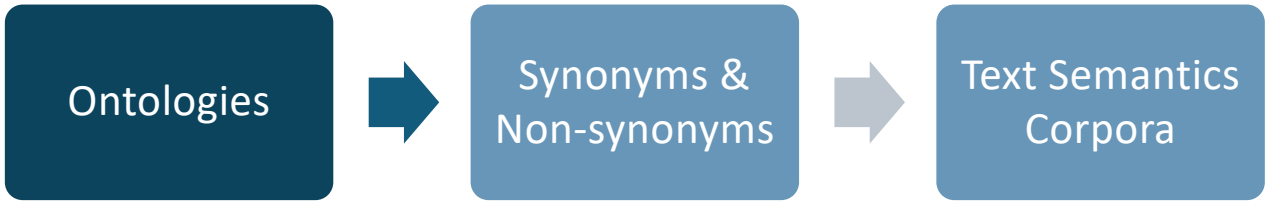
BERTMap Outline



Make predictions based on the ensemble results of class label (text) classification

Refine mappings through extension & repair

Mapping Prediction: Text Semantics Corpora



Synonyms: An ontology class could have multiple *aliases* defined by some annotational properties, e.g., *rdfs:label*, *oboInOwl:hasExactSynonym*.

Note: We considered both *reversed* and *identity* synonyms

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

Mapping Prediction: Text Semantics Corpora

Intra-ontology (from within an ontology)

- As described in the previous slide.

Cross-ontology (from a small portion of given mappings)

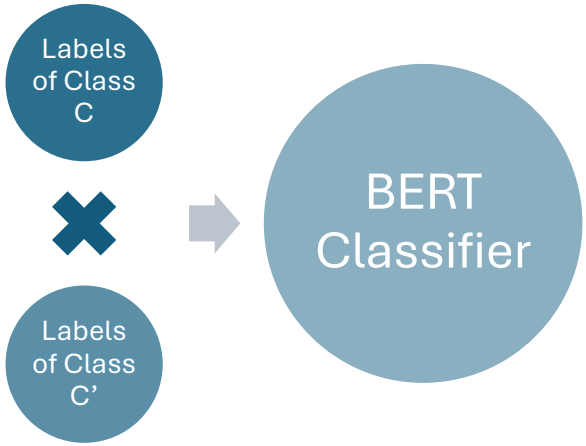
- *Synonyms* are extracted from label pairs of the matched classes.
- *Non-synonyms* are extracted from randomly aligned classes.

Complementary (from an auxiliary ontology)

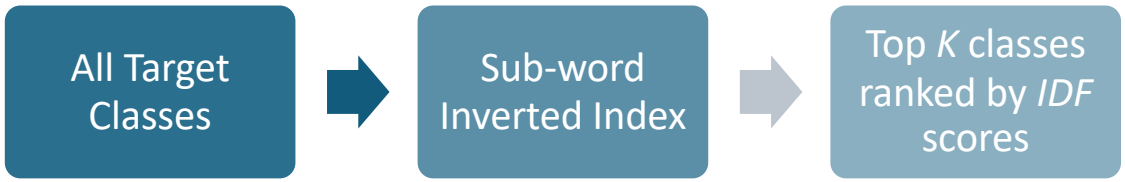
- Augmenting data for training the synonym classifier.

Mapping Prediction: Synonym Classifier

- Our *synonym classifier* comprises a pre-trained BERT model with an extra binary classification layer.
- We fine-tune the BERT classifier on the text-semantics corpora constructed from ontologies.
- 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'.



Mapping Prediction: Candidate Selection

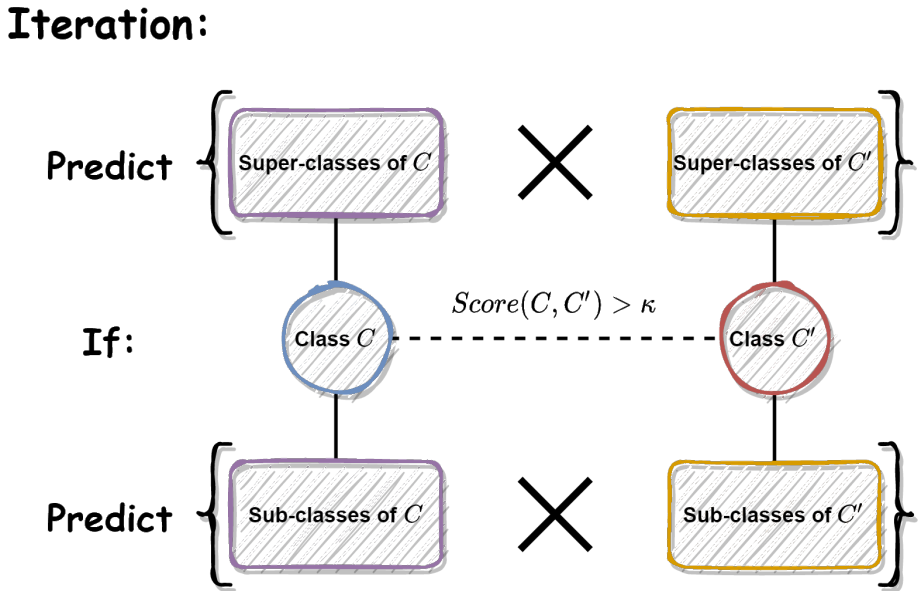


- For each source class *C*
  - Select target classes that share at least one sub-word
  - Get top *K* of them according to IDF score.
- **Outcome**: Reducing the quadratic search space to linear.
- **Limitation**: Miss classes that share no sub-word token with the given class.
- **Advantages**: Various word forms can be captured; **unknown words** are treated as consecutive known sub-words.

BERT's inherent WordPiece tokenizer
e.g., "unbelievable" → ["un", "##believe", "##able"]

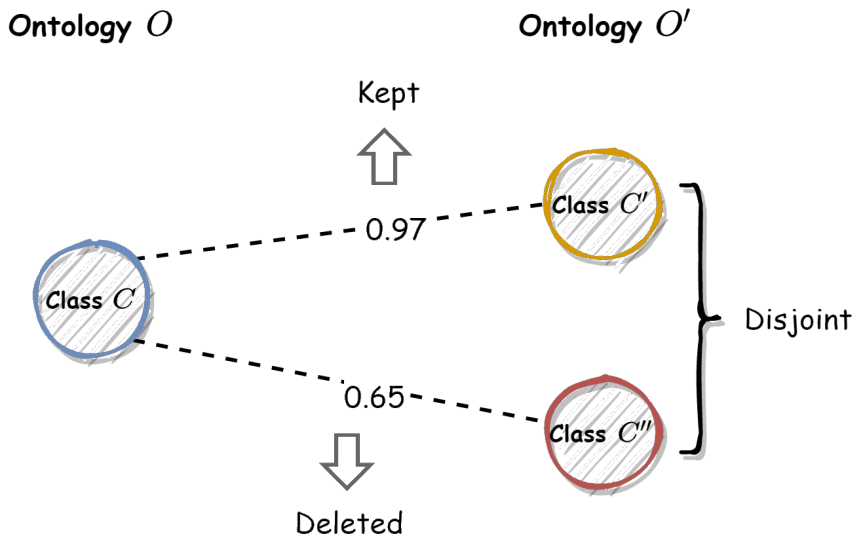
Mapping Refinement: Extension

- **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 (in candidate selection).



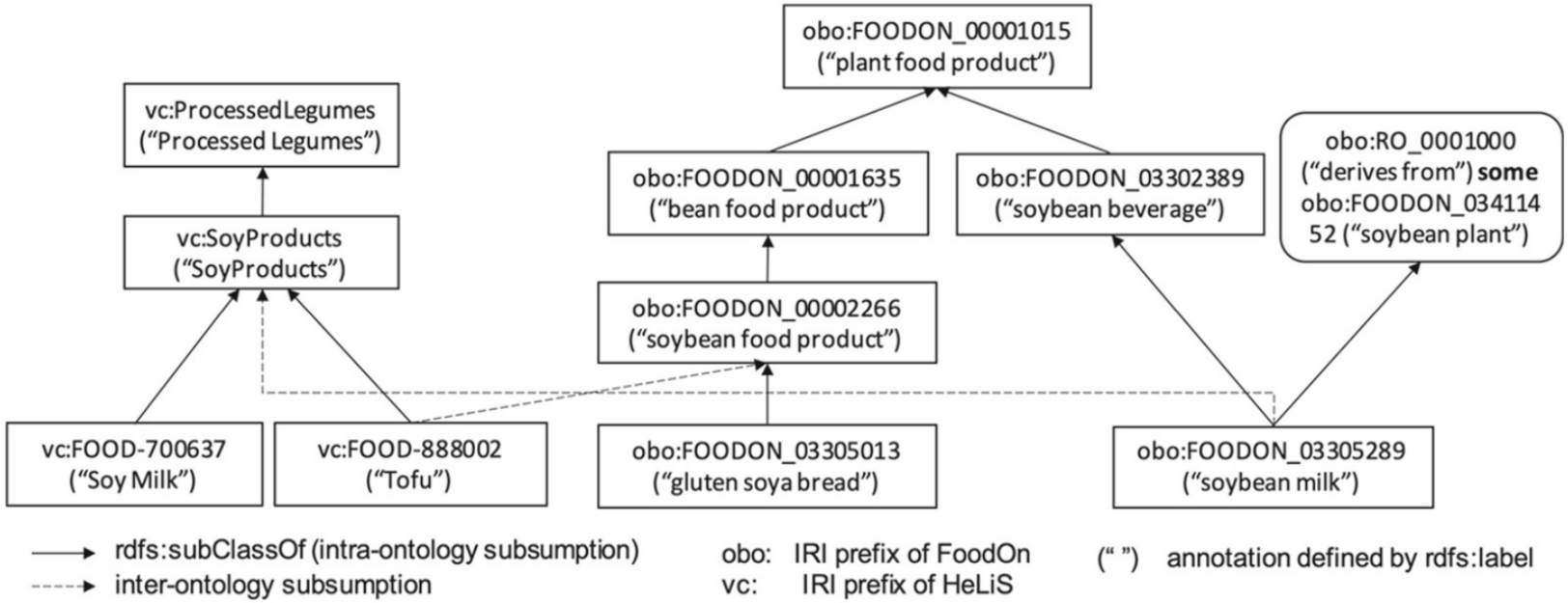
Mapping Refinement: Repair

- **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*.
- **Outcome:** To *improve precision* while keeping the recall as much as possible.



- Adopt the mapping repair tool built by Jimenez-Ruiz et al. (2013).

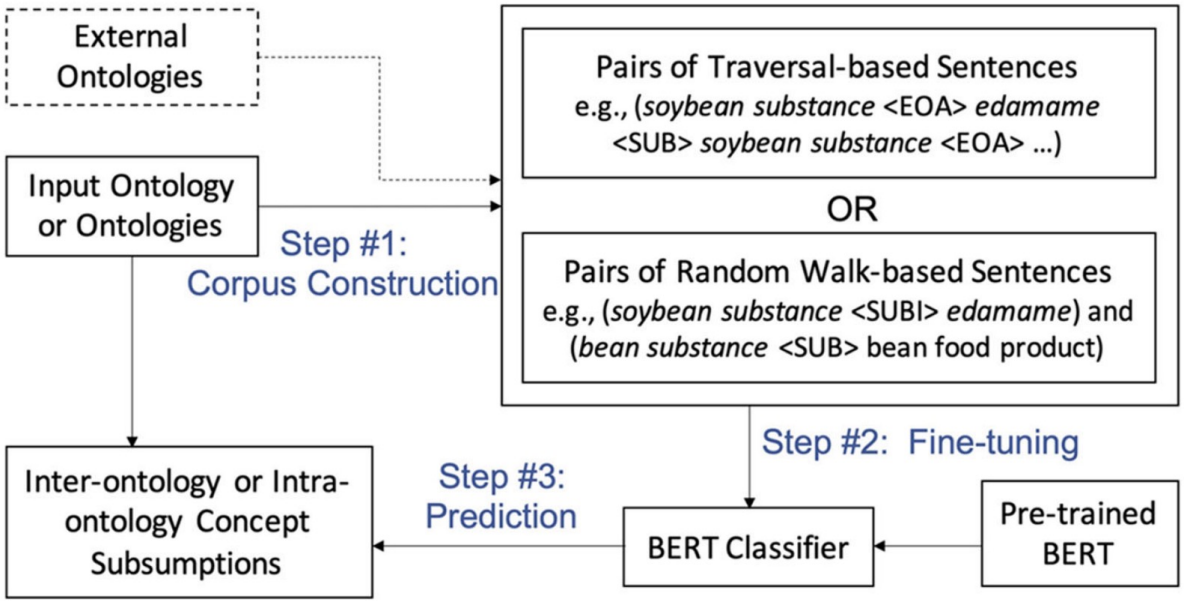
From Equivalence to Subsumption



Segments from the healthy lifestyle ontology HeLiS (Left) and the food ontology FoodOn (Right) with examples of inter-ontology and intra-ontology **class subsumptions**

BERTSubs

- Fine-tune a BERT model with **subsumptions** in the given ontologies
- Different templates for utilizing the context
  - **Class label** alone
  - **Class path**
  - **Class context** (breadth first search)
- Applications
  - OM with subsumption
  - Ontology completion



Chen, Jiaoyan, et al. "Contextual semantic embeddings for ontology subsumption prediction." *World Wide Web* 26.5 (2023): 2569-2591.

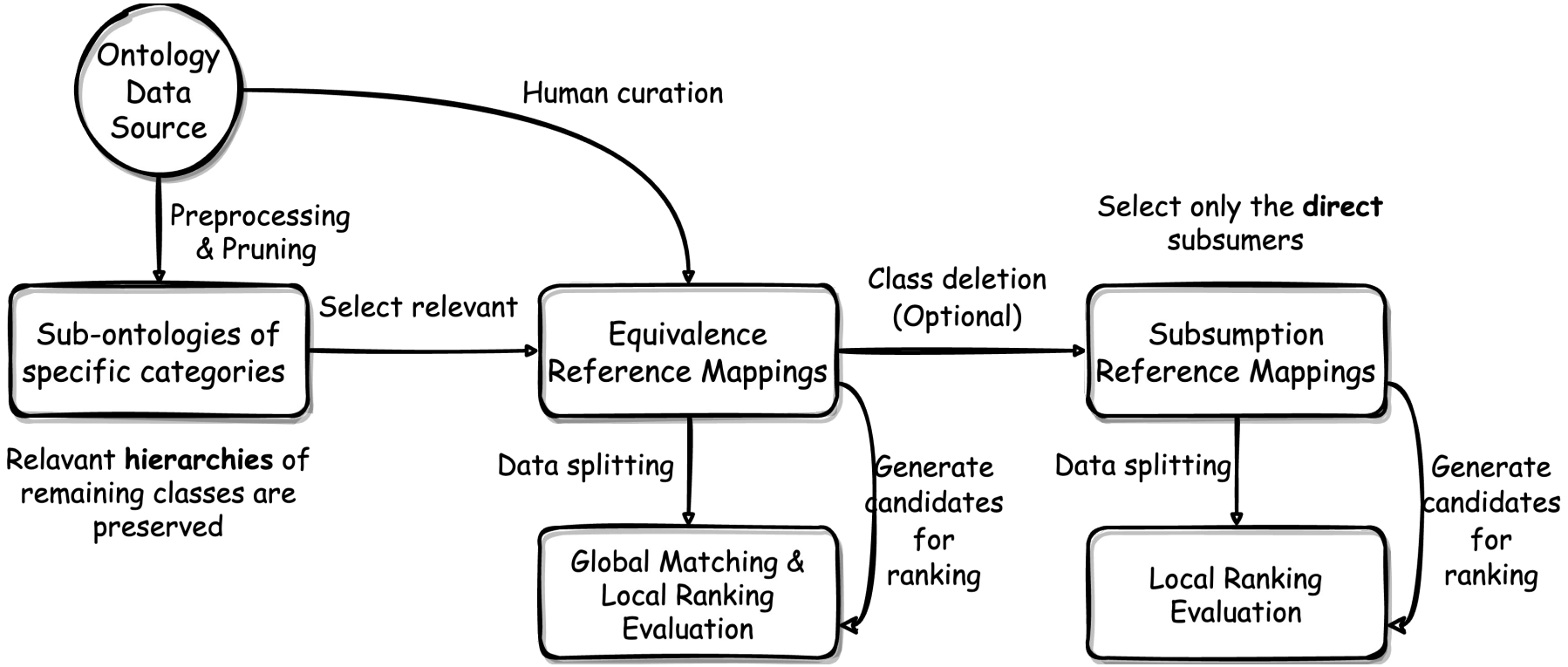
Bio-ML: New OM Benchmarks

- Motivation

Challenges for OM systems	Limitations of Existing OM Data
• **Variety** of naming scheme => synonyms; naming styles • **Ambiguity** => similar naming but in different contexts • **Scalability** => naïve traversal of OM takes O(n^2) • **Inconsistency** => merging ontologies often lead to logical conflicts	• Lack of **high-quality** gold standard mappings • Lack of a **unified** evaluation framework • Often **limited** to equivalence matching • Lack of support for **machine learning-based** systems.

He, Yuan, et al. "Machine learning-friendly biomedical datasets for equivalence and subsumption ontology matching." ISWC 2022.

Resource Construction



Resource Construction

For better quality of reference mappings:

- Industrial collaboration with human expertise (Mondo)
- Large-scale crowd-sourcing (UMLS)

For scalable ontologies, apply pruning by:

- Keeping concepts of a fine-grained category (disease, anatomy, etc.)
- Preserving relevant hierarchies (*rdfs:subClassOf*) of remaining concepts

To incorporate not just equivalence matching:

- Construct subsumption reference mappings from equivalence reference mappings
- Delete one of the classes in an equivalence mapping used for constructing any subsumption mappings => more challenging

Evaluation Protocol

- *Global matching* evaluation using traditional P/R/F1 metrics
  - To evaluate **overall** performance of OM systems
  - High-recall systems could be misjudged on **incomplete** reference mappings

- *Local ranking* evaluation using ranking metrics such as Hits@K and MRR.
  - To evaluate OM systems' ability on distinguishing the **correct mapping** from selective negative candidates.
  - Still informative on **incomplete** reference mappings
  - More **efficient** way of comparing or developing **ML-based** systems

Evaluation Protocol

For fairly comparing both ML-based and non-ML-based systems:

- A *validation* set for tuning hyper-parameters of ML-based systems or adjusting configurations of non-ML-based systems.
- A *testing* set for final evaluation.
- A *training* set (optional) for (semi-)supervised ML-based systems.

Bio-ML Datasets

Source	Task	Category	#Classes	#RefMaps (equiv)
Mondo	OMIM-ORDO	Disease	9,642-8838	3,721
Mondo	NCIT-DOID	Disease	6,835-8,848	4,684
UMLS	SNOMED-FMA	Body	24,182-64,726	7,256
UMLS	SNOMED-NCIT	Pharm	16,045-15,250	5,803
UMLS	SNOMED-NCIT	Neoplas	11,271-13,956	3,804

Dataset Statistics for Equivalence Ontology Matching

Bio-ML Datasets

Source	Task	Category	#Classes	#RefMaps (subs)
Mondo	OMIM-ORDO	Disease	9,642-8,735	103
Mondo	NCIT-DOID	Disease	6,835-5,113	3,339
UMLS	SNOMED-FMA	Body	24,182-59,567	5,506
UMLS	SNOMED-NCIT	Pharm	16,045-12,462	4,225
UMLS	SNOMED-NCIT	Neoplas	11,271-13,790	213

Dataset Statistics for Subsumption Ontology Matching

OAEI Bio-ML

- Bio-ML datasets are used for the OAEI Bio-ML track from 2022
  - 5 tasks
  - Both equivalence and subsumption matching
  - https://krr-oxford.github.io/OAEI-Bio-ML/



Bio-ML Track of OAEI 2025

Equivalence Matching Results for SNOMED-FMA (Body)

	Unsupervised (No Train Maps)					Semi-supervised (Optional 30% Train Maps)					
System	Precision	Recall	F-score	MRR	Hits@1	Use Train Maps	Precision	Recall	F-score	MRR	Hits@1
BERTMap‡	0.979	0.662	0.790	0.944	0.920	✔	0.970	0.669	0.792	0.965	0.947
BERTMapLt‡	0.979	0.655	0.785	0.892	0.865	✘	0.970	0.662	0.787	0.892	0.865
BioGITOM*						✔	0.962	0.886	0.923		
BioSTransMatch*	0.128	0.384	0.192	0.633	0.513	✔	0.357	0.661	0.464	0.855	0.798
HybridOM*	0.870	0.722	0.790	0.907	0.861	✘	0.825	0.725	0.772	0.907	0.861
LogMap†	0.744	0.407	0.526			✘	0.673	0.411	0.511		
LogMapBio†	0.827	0.577	0.680			✘	0.770	0.577	0.660		
LogMapLt†	0.970	0.542	0.696			✘	0.958	0.542	0.693		
Matcha†	0.887	0.502	0.641	0.950	0.935	✘	0.846	0.502	0.630	0.950	0.935

Bio-ML Track of OAEI 2025

Equivalence Matching Results for SNOMED-NCIT (Pharm)

	Unsupervised (No Train Maps)					Semi-supervised (Optional 30% Train Maps)					
⇕System	⇕Precision	⇕Recall	⇕F-score	⇕MRR	⇕Hits@1	⇕Use Train Maps	⇕Precision	⇕Recall	⇕F-score	⇕MRR	⇕Hits@1
BERTMap‡	0.971	0.585	0.730	0.969	0.951	✔	0.898	0.715	0.796	0.971	0.953
BERTMapLt‡	0.981	0.574	0.724	0.849	0.773	✘	0.973	0.569	0.718	0.849	0.773
BioGITOM*						✔	0.983	0.713	0.827		
BioSTransMatch*	0.584	0.844	0.690	0.943	0.918	✔	0.845	0.860	0.852	0.957	0.943
HybridOM*	0.916	0.889	0.902	0.964	0.936	✘	0.884	0.886	0.885	0.964	0.936
LogMap†	0.966	0.607	0.746			✘	0.952	0.603	0.738		
LogMapBio†	0.928	0.611	0.737			✘	0.899	0.606	0.724		
LogMapLt†	0.996	0.599	0.748			✘	0.994	0.594	0.743		
Matcha†	0.987	0.607	0.752	0.936	0.921	✘	0.982	0.601	0.746	0.936	0.921

Bio-ML Track of OAEI 2025

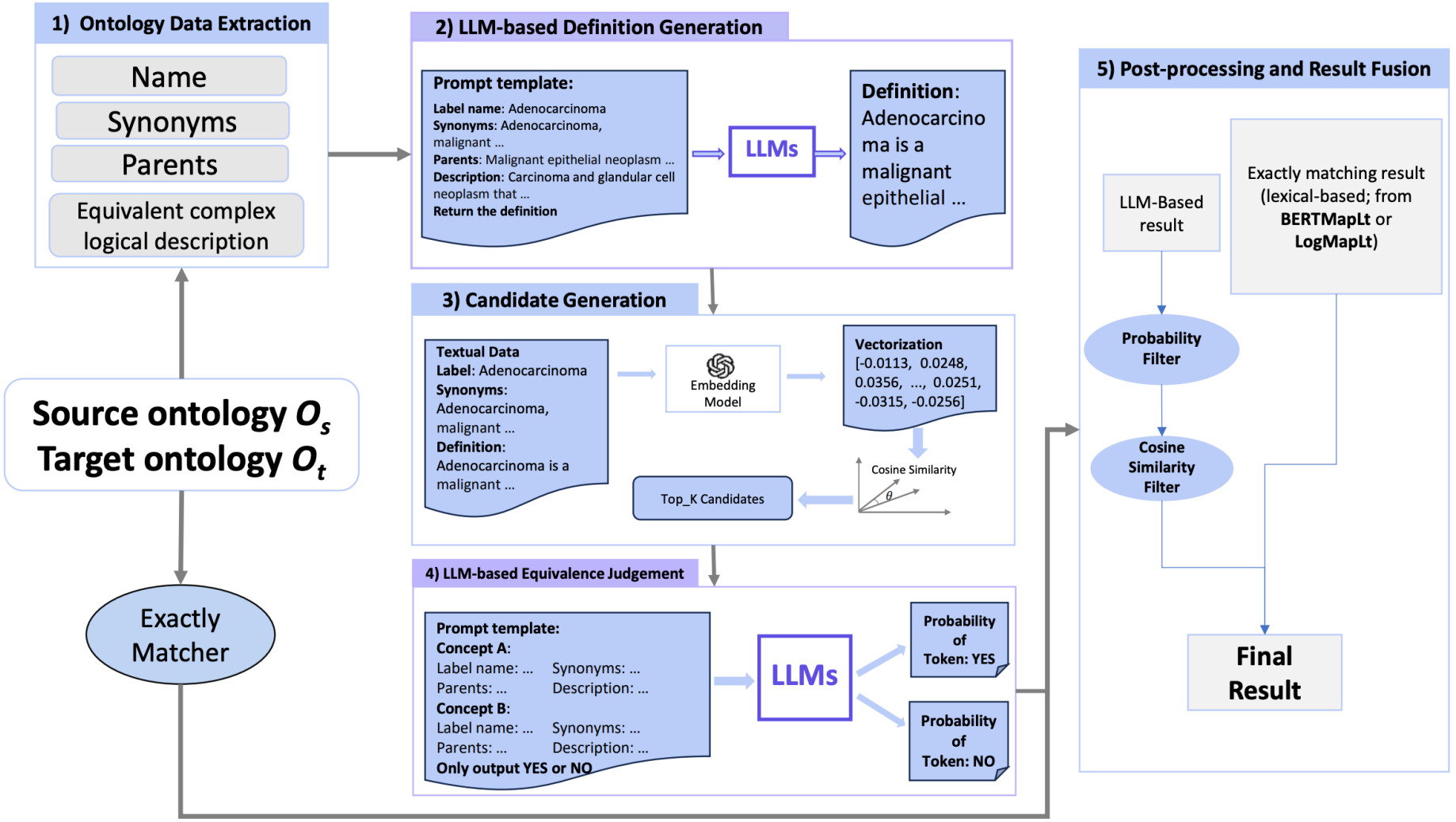
Subsumption Matching Results for SNOMED-FMA (Body)

⇕System	⇕Use Training Maps	⇕MRR	⇕Hits@1	⇕Hits@5	⇕Hits@10
BERTSubs (IC)‡	✔	0.596	0.466	0.768	0.908
BERTSubs (IC)‡	✘	0.479	0.305	0.692	0.878
OWL2Vec* + RF‡	✘	0.465	0.293	0.684	0.818
SORBETMtch*	✔	0.516	0.311	0.821	0.941
Word2Vec + RF‡	✘	0.356	0.210	0.509	0.694
Word2Vec + RF (del)‡	✘	0.305	0.164	0.443	0.639

GenOM

- How to utilise the generation capabilities and internal memory of LLMs?
  - GenOM: generate natural language definition before alignment

Song, Yiping, Jiaoyan Chen, and Renate A. Schmidt. "GenOM: ontology matching with description generation and large language models." *World Wide Web* 29.3 (2026): 29.



GenOM Architecture

Discussion / Future Directions

- LLM support for DeepOnto
  - E.g., GenOM and NeurOWL integration
- Agentic AI for OM
  - AgenticMap (ongoing)
- From OM to ontology **reuse**
  - Sub-ontology extraction
  - Web-scale ontology repository
  - Ontology reuse in agentic AI (Skills? MCP?)