Tutorial@ISWC 2026

The 25th International Semantic Web Conference

Bari, Italy

OntoLM: Ontology Embedding, Reasoning and Construction with Large Language Model





The University of Manchester



University of Exeter

Speakers



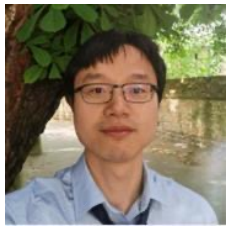
Hui Yang
The University of Manchester

Ontology, KRR,
Neural-symbolic AI



Jiaoyan Chen
The University of Manchester

Knowledge Graph, Ontology, KRR,
AI and LLM with KRR



Hang Dong
University of Exeter

ML for structured data and
knowledge, LLM, Healthcare

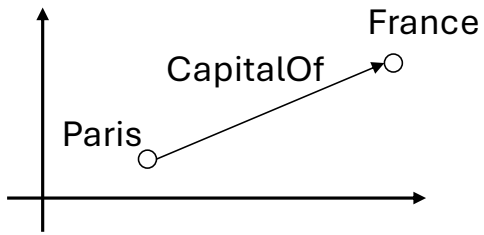
A bit more on myself (Jiaoyan)

- 2016: PhD in Computer Scince in Zhejiang University
  - Semantic Web, ML, spatial data anlaysis
- 2017-2022 (full-time) & 2022 – 2025 (part-time): Senior Researcher in the KRR group in Univeristy of Oxford
  - Knowledge graph, ontology, semantic data annotation, zero-shot learning
  - EPSRC ConCur, Samsung Research UK, eBay, Tencent, Alan Turing Institute, SIRIUS Norway
- 2022 – now (Lecturer / Senior Lecturer): Department of Computer Science, The University of Manchester
  - Knowledge graph, ontology, RAG, LLM, agentic AI
  - EPSRC OntoEm, SNOMED International

Why This Tutorial?

- Knowledge representation learning (embedding)
  - Vector-based (dense) representation of symbolic knowledge
  - Augmentation with machine learning and data mining
  - Reasoning under uncertainty & incompleteness
  - Knowledge curation, and domain applications

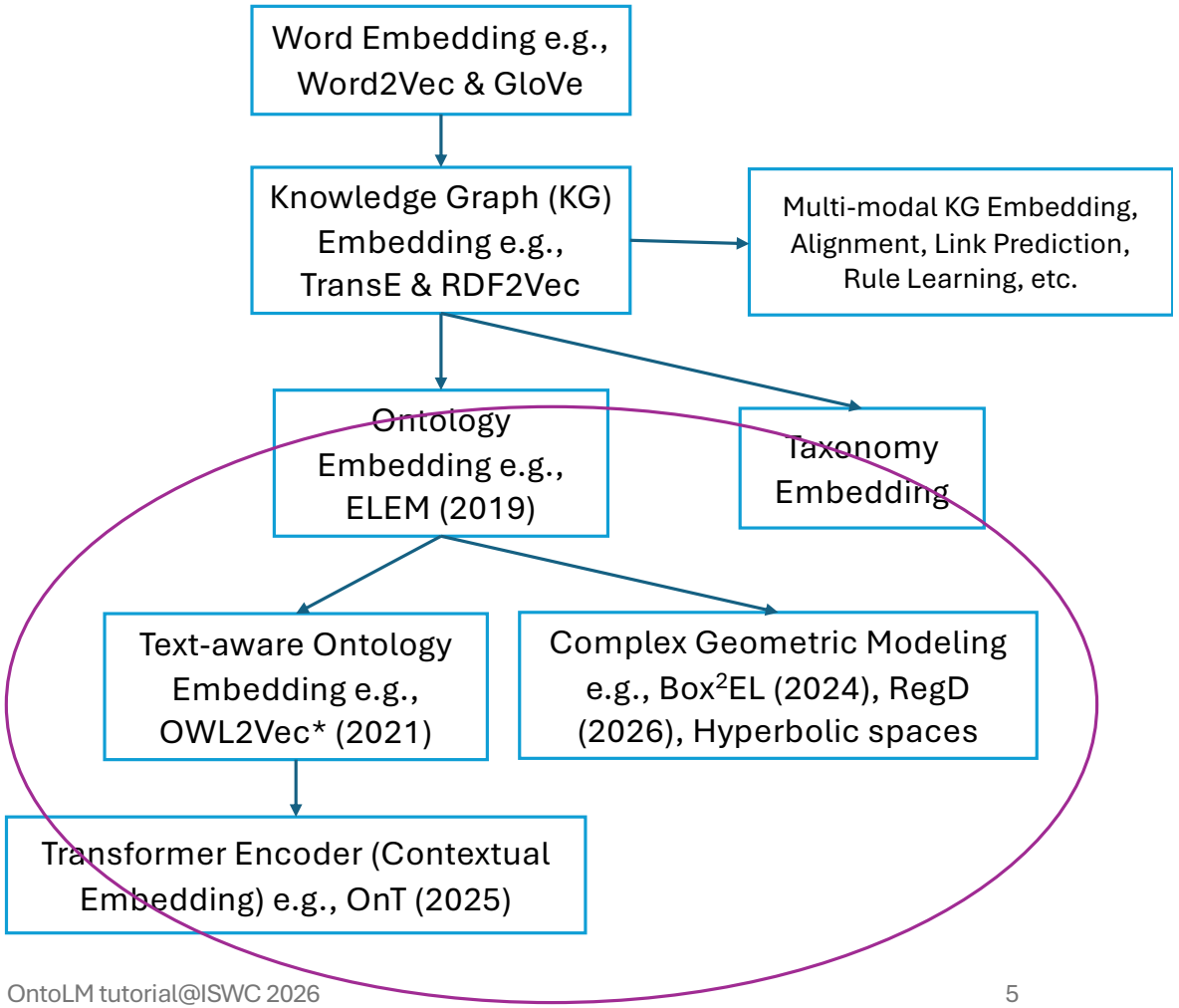
- Word embedding e.g., Word2Vec
  - $V_{king} - V_{queen} \approx V_{man} - V_{woman}$ (distributional hypothesis)
  - $\cos(V_{cat}, V_{kitten}) > \cos(V_{cat}, V_{dog})$ (similarity in usage; cooccurrence)



TransE: modeling relation as a translation mapping (KG embedding)

Why This Tutorial?

- A brief (and incomplete) summary of KG & ontology embedding development



Chen, Jiaoyan, et al. "Ontology embedding: a survey of methods, applications and resources." *IEEE Transactions on Knowledge and Data Engineering* 37.7 (2025): 4193-4212.

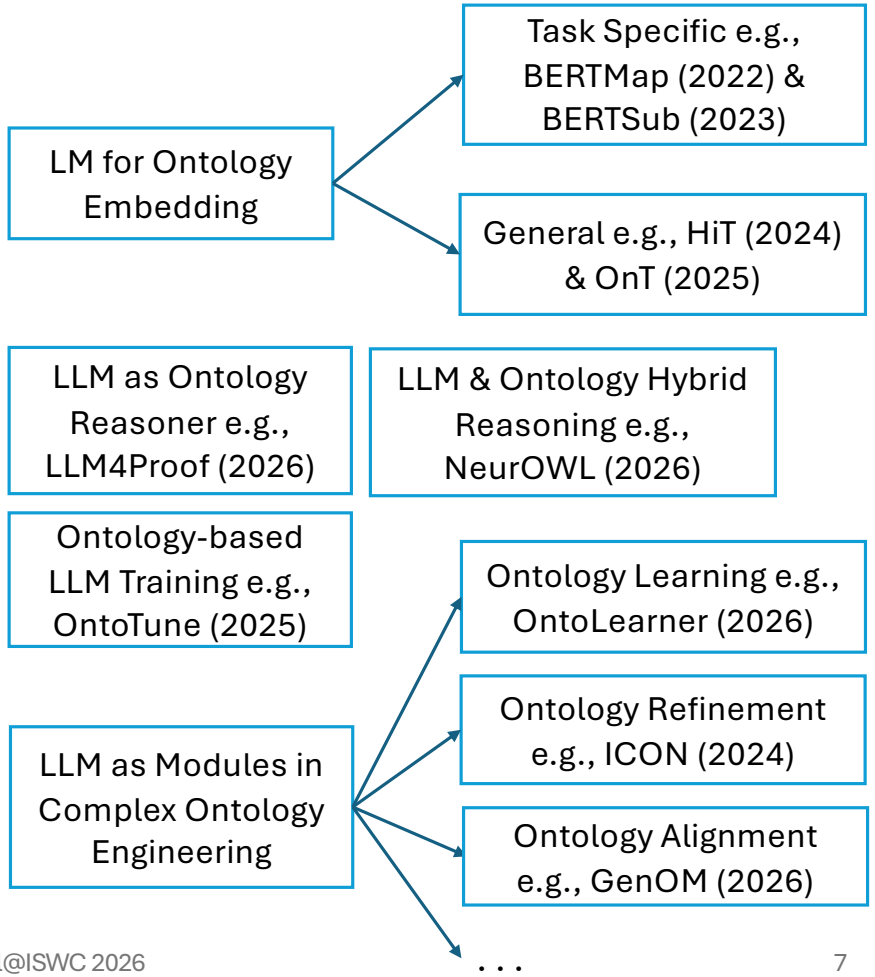
Why This Tutorial?

- (Large) Language Models
  - Transformer-based architecture: contextual embedding (**encoder**) of sequences
  - Pre-trained with large-scale corpus (**internal memory**)
  - Generation (**decoder**)
  - Natural language understanding and inference (**performance**)
  - Agentic AI (**thinking**, **planning**, **acting / tool-using**, **adaption**)

Yang, Hui, et al. "Ontology Construction and Reasoning with LLMs: Survey and Perspective." 2026.

Why This Tutorial?

- (Large) Language Models for ontology



Yang, Hui, et al. "Ontology Construction and Reasoning with LLMs: Survey and Perspective." 2026.

What in This Tutorial?

Part I - Ontology Embeddings			
1	15 mins	Introduction and roadmap	Jiaoyan Chen
2	40 mins	Geometric embeddings: from hierarchies to axioms	Hui Yang
3	45 mins	Language model-based embeddings	Hui Yang
30 mins - Conference break			
Part II - Ontology Reasoning and Construction			
4	30 mins	Ontology alignment with DeepOnto	Jiaoyan Chen
5	30 mins	Complex ontology reasoning with LLMs	Hui Yang
6	30 mins	New concept placement in ontologies	Hang Dong
7	10 mins	Discussion and Q&A	All

OntoLM Tutorial | Chapter 2 of 7

ISWC BARI 2026

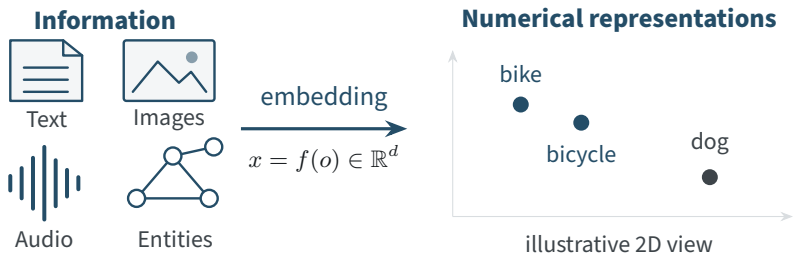
Geometric Embeddings: From Hierarchies to Axioms

Hui Yang
Research Associate
University of Manchester

MANCHESTER 1824
The University of Manchester

University of Exeter

Embeddings map information into vectors



An embedding represents an object as a vector, preserving task-relevant information

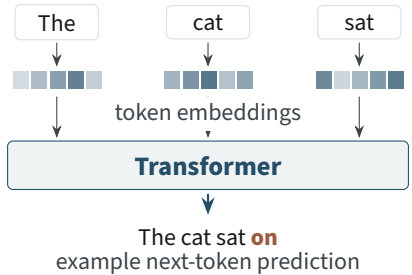
Embeddings support many different tasks

Simple: nearest-neighbour search



- Embed query and product descriptions
- Rank by vector similarity
- Retrieve by meaning, not exact wording

Complex: Transformer language models



- Map each token to a vector
- Use context to update representations
- Generate, translate and summarise

Roadmap: from hierarchies to ontologies

Three parts: hierarchies $\rightarrow$ ontologies $\rightarrow$ open problems

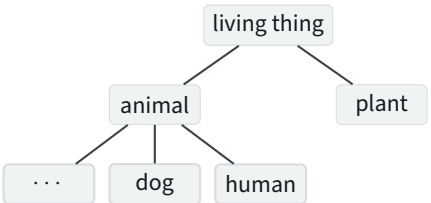
	Part	What it covers	Methods	Time
1	**Hierarchy embeddings**	is-a: asymmetry + transitivity, in low dimensions	order & box embeddings, hyperbolic space & cones, RegD[17]	20 min
2	**EL ontology embeddings**	beyond is-a: conjunction $\sqcap$, existential $\exists r$, plus analysis and training	ELEm[18], ELBE[20], BoxEL[21], Box^{2}EL[22], TransBox[23]	16 min
3	**Other works**	other logics, theoretical analysis, open problems	BoxLitE, FALCON, KB semantics	1 min

1

Hierarchy embeddings

Order and box embeddings
Hyperbolic space
RegD

Our first structured case: a hierarchy



A small taxonomy (example from RegD, KR 2026)

What must the geometry capture?

$C \sqsubseteq D$: "C is a subclass of D" (is-a)

- **Asymmetry**
  dog $\sqsubseteq$ animal, animal $\not\sqsubseteq$ dog
- **Transitivity**
  dog $\sqsubseteq$ animal $\sqsubseteq$ living thing
  $\Rightarrow$ dog $\sqsubseteq$ living thing

- **Where they appear:** plant taxonomy and product categories
  e.g. running shoes $\sqsubseteq$ shoes $\sqsubseteq$ footwear
- **Beyond trees:** a class may have multiple parents (a DAG)

Goal: embed a hierarchy while preserving its is-a relations

Order embeddings[4]: from points to nested regions

in depth

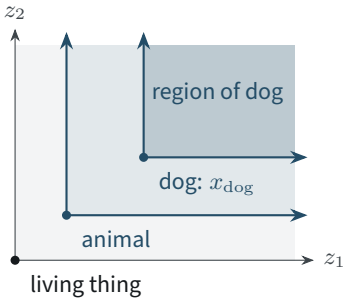
Definition: a region with apex x_u

Each concept u is embedded as an **upper orthant** R_u with apex $x_u \in \mathbb{R}^n_+$:

$$R_u = \{z \in \mathbb{R}^n_+ : z_i \geq (x_u)_i \text{ for all } i\}.$$

Encode $u \sqsubseteq v$ by inclusion:

$$R_u \subseteq R_v.$$



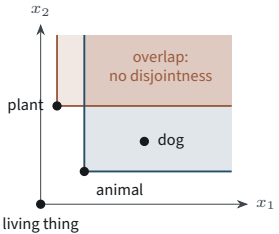
Transitivity comes from set inclusion:

$R_{\text{dog}} \subseteq R_{\text{animal}} \subseteq R_{\text{living thing}} \Rightarrow R_{\text{dog}} \subseteq R_{\text{living thing}}$.

Vendrov et al., ICLR 2016 [4], §2.1; region (cone) view: Vilnis et al., ACL 2018 [5], §3.2

Order embeddings[4]: two limitations

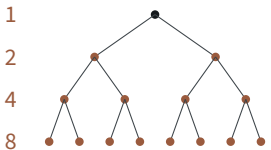
1 No disjointness



Animal and plant always overlap.

2 Crowding

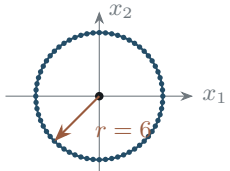
Binary tree (unit edges)



depth 3: $2^3 = 8$ nodes

depth 6: $2^6 = 64$ nodes

2D Euclidean space



64 nodes on the circle

$2\pi r \approx 37.7$

A b-ary tree: b^r **nodes** at depth r (unit edges).

In $\mathbb{R}^n$: sphere area $\propto r^{n-1}$.

Order embeddings: Vendrov et al., ICLR 2016 [4]; no disjointness: Vilnis et al., ACL 2018 [5], §4.2; crowding: Nickel & Kiela, NeurIPS 2017 [7], §2

τBox[16]: extending order embeddings with boxes

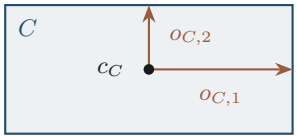
Box definition

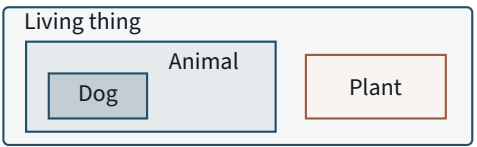
Each concept C is an axis-aligned box:

$$\mathrm{Box}(C) = [\, c_C - o_C,\ c_C + o_C \,].$$

$c_C \in \mathbb{R}^n$: **centre**

$o_C \in \mathbb{R}^n_{\geq 0}$: **offset** (half-width in each dimension)





What boxes fix

Containment: encode is-a.

Disjointness: encode empty intersection.

Remaining limitation

Euclidean space still suffers from **crowding**.

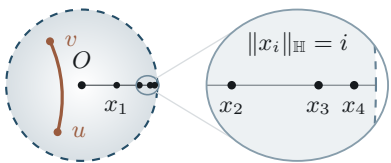
Box embeddings: Vilnis et al., ACL 2018 [5]; τBox: Boratko et al., NeurIPS 2021 [16]

Hyperbolic space: the Poincaré ball

Definition: the Poincaré ball $\mathbb{B}^n = \{x \in \mathbb{R}^n : \|x\| < 1\}$

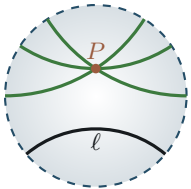
$$d_{\mathbb{H}}(u, v) = \operatorname{arcosh}\left(1 + \frac{2\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\right)$$

Shortest paths



Orange curve: shortest path from u to v
$\|x\|_{\mathbb{H}} := d_{\mathbb{H}}(O, x)$ (distance from O)
Boundary: **infinitely far**

Parallel lines

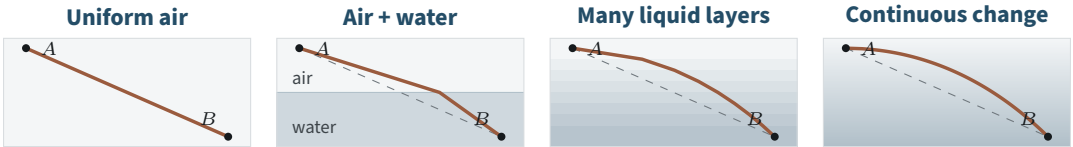


Through P: **infinitely many** lines never meet ℓ
(Euclidean space: exactly one)

Poincaré ball model; distance as in Nickel & Kiela, NeurIPS 2017 [7], §3, Eq. 1

Why can a “straight line” look curved?

An optical analogy for why a shortest route can look curved.



Orange: fastest light path

Dashed: straight shortcut

Light: fastest route

Changing speed can make the fastest route bend.

Hyperbolic space: the same intuition

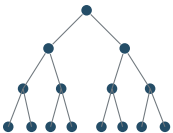
In the Poincaré ball picture, “density” increases towards the boundary, so shortest paths can look curved.

> An intuitive motivation only: not an equivalent model or a definition of hyperbolic space.

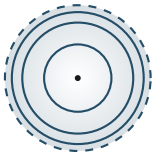
Optical analogy: Feynman Lectures I, Ch. 26; Poincaré model: Nickel & Kiela, 2017 [7]

Why hyperbolic space? Room that grows like a tree

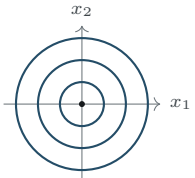
Crowding comes from $\mathbb{R}^n$: trees grow exponentially, Euclidean room only polynomially



Tree
2^ℓ nodes at level ℓ



Poincaré ball
$L(r) = 2\pi \sinh r$



Euclidean plane
$L(r) = 2\pi r$

$\times$ Euclidean $\mathbb{R}^n$: polynomial

Sphere of radius r: area $\propto r^{n-1}$.

Hyperbolic (curvature -1): exponential

Sphere area $\propto \sinh^{n-1} r \sim e^{(n-1)r}$.

Nickel & Kiela, NeurIPS 2017 [7], Section 2

Any tree fits in two hyperbolic dimensions

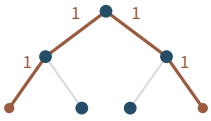
A tree as a **distance space**: vertices are points, path lengths are distances

Theorem (Sarkar, GD 2011 [8])

For every finite weighted tree T and every $\varepsilon > 0$, there are an embedding $f : V(T) \to \mathbb{H}^2$ and a global scale $\tau > 0$ such that

$$\tfrac{1}{1+\varepsilon}\, d_T(u,v) \;\le\; \tfrac{1}{\tau}\, d_{\mathbb{H}}\big(f(u), f(v)\big) \;\le\; d_T(u,v).$$

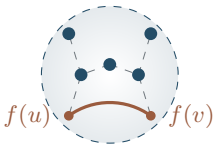
(a) Tree metric $(V(T), d_T)$



$d_T(u,v) = 1+1+1+1 = 4$

f

(b) Embedding in $\mathbb{H}^2$



$d_{\mathbb{H}}(f(u), f(v))$

Sarkar, GD 2011 [8]. Schematic mapping, not the theorem's exact construction.

Poincaré embeddings[7]: an asymmetric score

Score function ($\alpha > 0$)

$$\mathrm{score}(u \sqsubseteq v) = -\big(1 + \underbrace{\alpha(\|v\|_{\mathbb{H}} - \|u\|_{\mathbb{H}})}_{\text{asymmetry term}}\big) \quad \underbrace{d_{\mathbb{H}}(u, v)}_{\text{hyperbolic distance}}$$

$\|x\|_{\mathbb{H}} := d_{\mathbb{H}}(O, x)$: **hyperbolic norm** (distance from the origin).

- $d_{\mathbb{H}}(u, v) = d_{\mathbb{H}}(v, u)$: distance alone gives no direction.
- With the asymmetry term:

$$\|v\|_{\mathbb{H}} < \|u\|_{\mathbb{H}} \quad \Longrightarrow \quad \mathrm{score}(u \sqsubseteq v) > \mathrm{score}(v \sqsubseteq u).$$

The score favours **parents closer to the origin** than their children.

Hyperbolic-radius variant of Nickel & Kiela, NeurIPS 2017 [7], §4.3, Eq. 7

Poincaré embeddings[7]: WordNet results

Reconstruct the transitive closure of WordNet nouns: 743k is-a edges among 82k nouns

Method	Dimension	Space	MAP ↑	Mean rank ↓
Poincaré embeddings[7]	5	Hyperbolic (Poincaré ball)	**0.82**	**4.9**
Euclidean point baseline	200	Euclidean	0.17	1157

MAP ↑

How well related nouns are ranked above unrelated ones.
1 is best.

Mean rank ↓

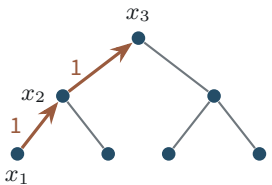
Average position of a related noun.
Lower is better.

> 40× fewer dimensions, far better reconstruction

Nickel & Kiela, NeurIPS 2017 [7], §4.1 and Table 1

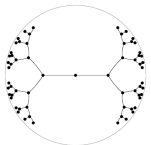
Hyperbolic embeddings keep distances, not order

Sarkar's theorem preserves the tree metric.
It does not define an is-a relation.



Arrows: child $\to$ parent





Poincaré disk

Preserved: distances

Sarkar: distortion at most $1 + \varepsilon$.

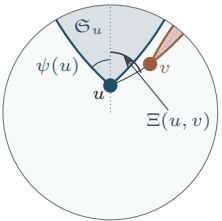
Not preserved: transitive

x_1 is a child of x_2, x_2 is a child of x_3.

$\not\Rightarrow$ x_1 is a child of x_3.

Sarkar, GD 2011 [8]. Visualisation: Stanford DAWN

Entailment cones[14]: hierarchy through nested regions



$\mathfrak{S}_u$ (blue), $\mathfrak{S}_v$ (orange) $\quad v \notin \mathfrak{S}_u$.

Cone geometry

Apex u Half-aperture $\psi(u)$ around the outward radial direction.

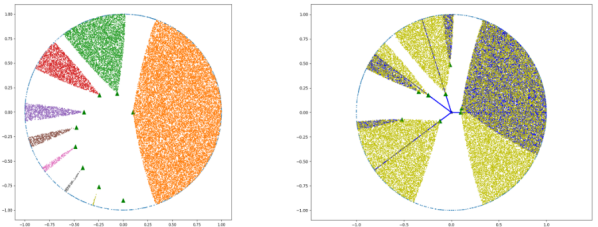
$$\psi(u) = \arcsin\left(K\frac{1-\|u\|^2}{\|u\|}\right), \quad K > 0$$

Entailment test

$\Xi(u, v)$: angle from the outward radius to the geodesic toward v.

$$E(u, v) = \max\big(0, \Xi(u, v) - \psi(u)\big)$$

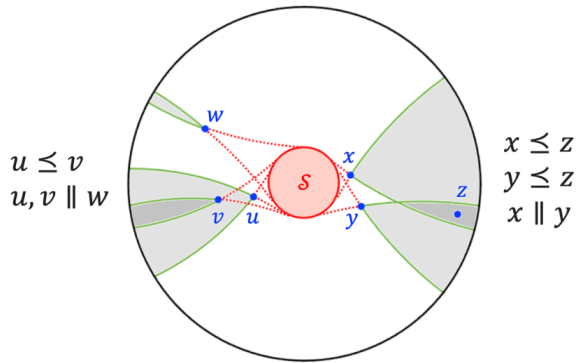
$E = 0$ inside $\quad E > 0$ outside $\quad$ Nesting gives transitivity.



Cone regions and an embedded hierarchy

Ganea et al., ICML 2018 [14], Eqs. 24–28, 33; Fig. 2. Left: $K = 0.2$, $\|u\| = 0.25$; $\|u\|$ is the Euclidean coordinate norm.

Shadow cones[15]: inspired by light and shadow



Original figure: $u \preceq v$ corresponds to $v \sqsubseteq u$.

- **Different construction:** cones inspired by light and shadow.
- **Same principle:** hierarchy through nested regions.

Yu et al., Shadow Cones: A Generalized Framework for Partial Order Embeddings, ICLR 2024 [15], Fig. 1.

Remaining limitations, and our question

× **Remaining limitations of hyperbolic methods**

- Specific cone constructions: incompatible with other region embeddings
- Hyperbolic computation, Riemannian optimisation and high numerical precision (e.g. FP64 / double precision)

Riemannian SGD: Bonnabel, IEEE TAC 2013 [10]; Bécigneul & Ganea, ICLR 2019 [11]. Precision: Sala et al., ICML 2018 [9]; Yu & De Sa, NeurIPS 2019 [12]; Mishne et al., ICML 2023 [13].

Can **Euclidean regions** get **hyperbolic expressivity**?

Remaining limitations, and our question

× **Remaining limitations of hyperbolic methods**

- Specific cone constructions: incompatible with other region embeddings
- Hyperbolic computation, Riemannian optimisation and high numerical precision (e.g. FP64 / double precision)

Riemannian SGD: Bonnabel, IEEE TAC 2013 [10]; Bécigneul & Ganea, ICLR 2019 [11]. Precision: Sala et al., ICML 2018 [9]; Yu & De Sa, NeurIPS 2019 [12]; Mishne et al., ICML 2023 [13].

Can **Euclidean regions** get **hyperbolic expressivity**?

RegD[17]: a Euclidean framework

1. **Plug-and-play**: works with existing region methods (balls, boxes)
2. **Euclidean realisation**: hyperbolic expressivity, standard computation

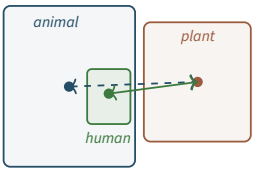
Goal: Euclidean regions + hyperbolic expressivity

What Euclidean regions lack: P1 and P2

Two key properties of hyperbolic space that Euclidean regions lack

P1: layer discrimination

A descendant can be arbitrarily farther from an outside concept than its ancestor



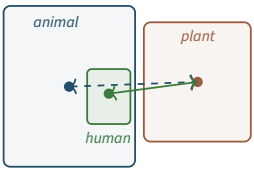
Euclidean centres: no gap guaranteed

What Euclidean regions lack: P1 and P2

Two key properties of hyperbolic space that Euclidean regions lack

P1: layer discrimination

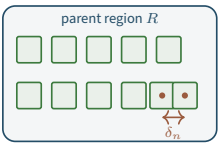
A descendant can be arbitrarily farther from an outside concept than its ancestor



Euclidean centres: no gap guaranteed

P2: child distinguishability

A parent can hold arbitrarily many children, pairwise far apart



Euclidean packing: children crowd together

Yang & Chen, KR 2026 [17], §3

RegD[17] (1): depth dissimilarity

in depth

Definition: depth dissimilarity for regions R_1, R_2

$$d_{\mathrm{dep}}(R_1, R_2) = g\left(\frac{\|P(R_1) - P(R_2)\|_p^p}{f(R_1)f(R_2)}\right)$$

P: parameters | $f(R) > 0$, region size $\to 0 \Rightarrow f(R) \to 0$ | g increasing, $g(0) = 0$.

E.g. $R = B(\mathbf{c}, r)$: $P(R) = (\mathbf{c}, r), \quad f(R) = r, \quad g(x) = x.$

Hyperbolic distance in the Poincaré ball $\mathbb{B}^n$

$$d_{\mathbb{H}}(u, v) = \operatorname{arcosh}\left(1 + \frac{2\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\right)$$

Yang & Chen, KR 2026 [17], §3.1, Prop. 1; Nickel & Kiela, NeurIPS 2017 [7], Eq. 1.

RegD[17] (1): depth dissimilarity

in depth

Definition: depth dissimilarity for regions R_1, R_2

$$d_{\text{dep}}(R_1, R_2) = g\left(\frac{\|P(R_1) - P(R_2)\|_p^p}{f(R_1)f(R_2)}\right)$$

P: parameters | $f(R) > 0$, region size $\to 0 \Rightarrow f(R) \to 0$ | g increasing, $g(0) = 0$.

E.g. $R = B(\mathbf{c}, r)$: $P(R) = (\mathbf{c}, r), \quad f(R) = r, \quad g(x) = x.$

Proposition 1: exact hyperbolic distance

For **Euclidean balls**, suitable f, g (with $p = 2$) make depth dissimilarity **exactly equal to hyperbolic distance** in one higher dimension.

Yang & Chen, KR 2026 [17], §3.1, Prop. 1; Nickel & Kiela, NeurIPS 2017 [7], Eq. 1.

RegD[17] (1): what depth dissimilarity guarantees

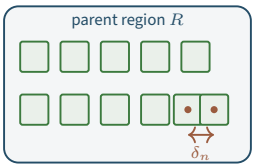
in depth

Theorem 1: d_{dep} realises P1 and P2 (balls and boxes alike)

P1: every small enough $R' \subseteq R_2$ has $d_{\mathrm{dep}}(R_1, R') > d_{\mathrm{dep}}(R_1, R_2) + \Delta$, for any $\Delta > 0$.

P2: every region contains arbitrarily many children, pairwise $d_{\mathrm{dep}} > M$, for any $M > 0$.

P2 example: fit more children, yet keep them distinguishable



Euclidean centres crowd together.

$$d_{\mathrm{dep}}(R_i, R_j) = g\left(\frac{\|P(R_i) - P(R_j)\|_p^p}{f(R_i) f(R_j)}\right)$$

Step 1. Shrink to fit. Make regions small enough to fit any number N inside the parent.

Step 2. Shrink again to separate. Even with tiny parameter gaps, shrink sizes further $\Rightarrow$ pairwise $d_{\mathrm{dep}} > \delta_n$.

Yang & Chen, KR 2026 [17], §3.1, Theorem 1. P2 example uses $g(x) = x$.

RegD[17] (2): boundary dissimilarity

Boundary dissimilarity: is R_2 inside R_1?

$$d_{\mathrm{bd}}(R_1, R_2) = \begin{cases} -\min_{\mathbf{x}_2 \in R_2} d(R_1^c, \mathbf{x}_2), & R_2 \subseteq R_1, \\ \max_{\mathbf{x}_2 \in R_2 \setminus R_1} d(R_1, \mathbf{x}_2), & \text{otherwise.} \end{cases}$$

Inside: minus the cost to move part of R_2 out of R_1.

Outside: the cost to move all of R_2 into R_1.

Yang & Chen, KR 2026 [17], §3.2, Def. of boundary dissimilarity and ball example.

RegD[17] (2): boundary dissimilarity

Boundary dissimilarity: is R_2 inside R_1?

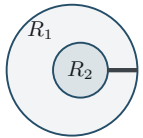
$$d_{\mathrm{bd}}(R_1, R_2) = \begin{cases} -\min_{\mathbf{x}_2 \in R_2} d(R_1^c, \mathbf{x}_2), & R_2 \subseteq R_1, \\ \max_{\mathbf{x}_2 \in R_2 \setminus R_1} d(R_1, \mathbf{x}_2), & \text{otherwise.} \end{cases}$$

Inside: minus the cost to move part of R_2 out of R_1.

Outside: the cost to move all of R_2 into R_1.

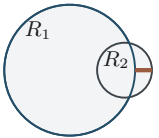
Example: balls $R_i = B(\mathbf{c}_i, r_i)$

$d_{\mathrm{bd}}(R_1, R_2) = \|\mathbf{c}_1 - \mathbf{c}_2\|_2 + r_2 - r_1$



$d_{\mathrm{bd}}(R_1, R_2) < 0$

contained



$d_{\mathrm{bd}}(R_1, R_2) > 0$

violation

$d_{\mathrm{bd}}(R_1, R_2) \le 0 \iff R_2 \subseteq R_1$ (zero: contained and touching).

Yang & Chen, KR 2026 [17], §3.2, Def. of boundary dissimilarity and ball example.

RegD[17] (3): the score function

Poincaré embeddings[7]: points in $\mathbb{B}^n$

$$-\mathrm{score}(u \sqsubseteq v) \;=\; \big(1 + \underbrace{\alpha\,(\|v\| - \|u\|)}_{\text{direction: norm}}\big) \cdot \underbrace{d_{\mathbb{H}}(u, v)}_{\text{hyperbolic distance}}$$

RegD[17]: regions in $\mathbb{R}^n$

$$E(u \sqsubseteq v) \;=\; \underbrace{d_{\mathrm{bd}}(R_v, R_u)}_{\text{direction: inclusion}} \;+ \lambda \cdot \underbrace{d_{\mathrm{dep}}(R_v, R_u)}_{\text{hyperbolic-like distance}}$$

Yang & Chen, KR 2026 [17], §3.3 and Table 1; Nickel & Kiela, NeurIPS 2017 [7], Eq. 7

RegD[17] (3): the score function

Poincaré embeddings[7]: points in $\mathbb{B}^n$

$$-\text{score}(u \sqsubseteq v) \;=\; \big(1 + \underbrace{\alpha\,(\|v\| - \|u\|)}_{\text{direction: norm}}\big) \cdot \quad \underbrace{d_{\mathbb{H}}(u, v)}_{\text{hyperbolic distance}}$$

RegD[17]: regions in $\mathbb{R}^n$

$$E(u \sqsubseteq v) \;=\; \underbrace{d_{\text{bd}}(R_v, R_u)}_{\text{direction: inclusion}} \;+ \lambda \cdot \quad \underbrace{d_{\text{dep}}(R_v, R_u)}_{\text{hyperbolic-like distance}}$$

- Lower is better.
- Predict $u \sqsubseteq v$ when $E \leq t$ (threshold tuned on validation).
- λ **tunes the distance term**: $\lambda = 0 \rightarrow$ pure inclusion
  $\lambda > 0 \rightarrow$ adds hyperbolic behaviour.

Yang & Chen, KR 2026 [17], §3.3 and Table 1; Nickel & Kiela, NeurIPS 2017 [7], Eq. 7

RegD[17] experiments (1): DAG inference

Train on basic edges, predict held-out inferred edges

F1 (%)	dims	best baseline	RegD[17]
Mammal	2 / 5	57.7 / 69.4	64.9 / 71.8
WordNet Noun	5 / 10	45.2 / 52.2	58.4 / 59.1
MCG	5 / 10	48.4 / 55.5	50.7 / 58.5
Hearst	5 / 10	41.6 / 46.8	42.8 / 49.6

- Baselines: entailment / shadow cones, order emb., τBox, ELBE, ELEm
- RegD best in **8/8** settings
- Largest gain $+13.2$ (Noun, $d{=}5$)

Yang & Chen, KR 2026 [17], Table 2

Hierarchy embeddings: a comparison

Method	Representation	Transitivity	Disjointness	No crowding (low dim.)	Euclidean computation
Order embeddings[4]	orthants in $\mathbb{R}^n_+$	✓	×	×	✓
τBox[16]	boxes in $\mathbb{R}^n$	✓	✓	×	✓
Poincaré embeddings[7]	points in $\mathbb{B}^n$	×	×	✓	×
Hyperbolic cones	cones in $\mathbb{H}^n$	✓	✓	✓	×
RegD[17]	any regions in $\mathbb{R}^n$	partial*	✓	✓	✓

* Full transitivity only for $\lambda = 0$ (pure inclusion, but then no hyperbolic room).

Summary of the methods above; RegD: Yang & Chen, KR 2026 [17], §3.3

2

EL-family ontology embeddings

EL-family normal forms
From points to regions
ELEm $\rightarrow$ boxes $\rightarrow$ TransBox

EL-family normal forms

From hierarchies to ontology axioms between concepts

Normal forms (templates)

NF1 $A \sqsubseteq B$

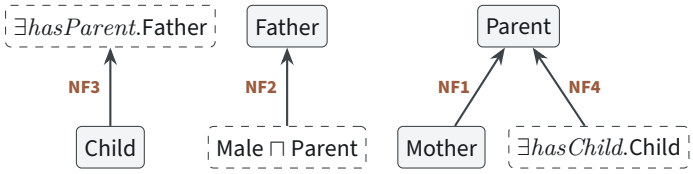
NF2 $A_1 \sqcap A_2 \sqsubseteq B$

NF3 $A \sqsubseteq \exists r.B$

NF4 $\exists r.B \sqsubseteq A$

A, A_i, B: concept names
r: role

Family example: a hierarchy of (complex) concepts



Arrows: $\sqsubseteq$ (sub $\rightarrow$ super)
Dashed boxes: complex concepts

Beyond basic EL: two role-axiom forms used here

Role inclusion: $r \sqsubseteq s$
Example: $hasMother \sqsubseteq hasParent$

Role-chain inclusion: $r_1 \circ r_2 \sqsubseteq s$
Example: $hasParent \circ hasParent \sqsubseteq hasGrandparent$

Normal forms: Baader et al., An Introduction to Description Logic, 2017 [1], Ch. 6; family ontology: Yang et al., WWW 2025 [23], Fig. 4a

Compare KG embeddings to ontology embeddings

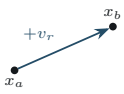
Basic idea: extend KG embeddings from **points** to **regions**

KG embedding (TransE[2])

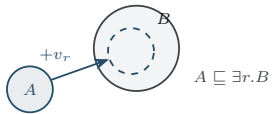
- Tuples $r(a, b)$ between **instances**
- Instance $\mapsto$ point

Ontology embedding

- Axioms $A \sqsubseteq B$ between **concepts**
- Concepts **include** each other $\Rightarrow$ concept $\mapsto$ **region**



KG: points



ontology: regions

TransE: Bordes et al., NeurIPS 2013 [2]

ELEm[18]: balls and translations

in depth

Concept $A \mapsto \mathrm{Ball}(A) := B(c_A, r_A)$ (centre, radius)
Role $r \mapsto$ translation vector v_r

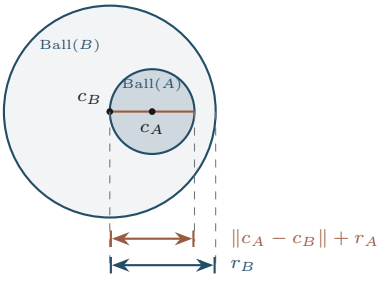
NF1 $A \sqsubseteq B$**:** $\mathrm{Ball}(A) \subseteq \mathrm{Ball}(B)$

$$\mathrm{Ball}(A) \subseteq \mathrm{Ball}(B) \iff \|c_A - c_B\| + r_A \leq r_B$$

Loss: $[\,\|c_A - c_B\| + r_A - r_B - \gamma\,]_+$ (margin γ)

Encode: $(\exists r.B)^{\mathcal{I}} = \mathrm{Ball}(B) - v_r$.

NF3 $A \sqsubseteq \exists r.B$: $\mathrm{Ball}(A) + v_r \subseteq \mathrm{Ball}(B)$.
NF4 $\exists r.B \sqsubseteq A$: $\mathrm{Ball}(B) - v_r \subseteq \mathrm{Ball}(A)$.



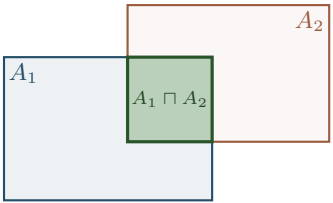
orange $\leq$ blue $\Rightarrow$ ball containment

NF2 $A_1 \sqcap A_2 \sqsubseteq B$: $\mathrm{Ball}(A_1) \cap \mathrm{Ball}(A_2)$ is generally not a ball $\Rightarrow$ **approximate** loss.
Zero loss need not imply $\mathrm{Ball}(A_1) \cap \mathrm{Ball}(A_2) \subseteq \mathrm{Ball}(B)$.

Kulmanov et al., IJCAI 2019 [18], §3.2, Eqs. 1–4

ELBE[20]: intersectional closure

brief



The box picture from ELBE

Roles use translations: $(\exists r.B)^{\mathcal{I}} = \mathrm{Box}(B) - v_r$.

NF2 $A_1 \sqcap A_2 \sqsubseteq B$

$$\mathrm{Box}(A_1) \cap \mathrm{Box}(A_2) \subseteq \mathrm{Box}(B)$$

Conjunction is represented exactly:

the intersection of two boxes is a box (or empty).

Peng et al., ELBE, arXiv 2022 [20], §4.2–4.3, Eqs. 2–6

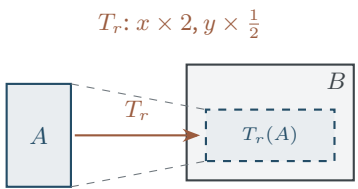
BoxEL[21]: roles as affine maps

Role definition (invertible case)

$$T_r(x) = D_r x + b_r, \quad D_r = \mathrm{diag}(d_{r,1}, \ldots, d_{r,n}), \quad d_{r,i} > 0; \qquad r^{\mathcal{I}} = \{(x, T_r(x)) : x \in \mathbb{R}^n\}.$$

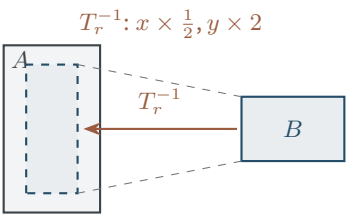
Encode $\exists r.B$ as $T_r^{-1}(\mathrm{Box}(B))$: take the preimage of B under the role map.

NF3 $A \sqsubseteq \exists r.B$



$$T_r\big(\mathrm{Box}(A)\big) \subseteq \mathrm{Box}(B)$$

NF4 $\exists r.B \sqsubseteq A$



$$T_r^{-1}\big(\mathrm{Box}(B)\big) \subseteq \mathrm{Box}(A)$$

Xiong et al., ISWC 2022 [21], §4.1, §4.4, Eqs. 9–10; adapted from Fig. 2(f), illustrative scales

Beyond functions: roles can be many-to-many

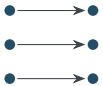
One transformation per role

ELEm[18] **/ ELBE**: one translation vector

$$x_b = x_a + v_r$$

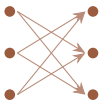
BoxEL[21]: one affine map per role.

$\Rightarrow$ Each input has **one output**.



One output per input

each input $\mapsto$ exactly one output



Many-to-many

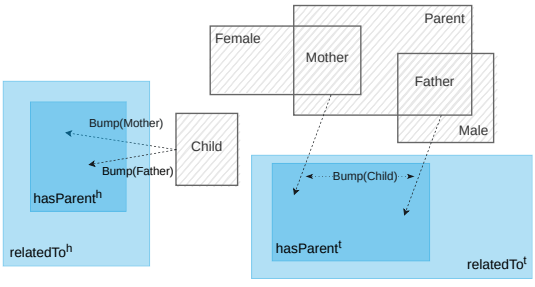
each input $\mapsto$ several outputs,
each output $\leftarrow$ several inputs
e.g. colleagueOf, hasParent

Next: **Box**^{2}**EL**[22] $\rightarrow$ head / tail boxes + bumps
TransBox[23] $\rightarrow$ boxes of translations

ELEm: Kulmanov et al., 2019 [18]; ELBE: Peng et al., 2022 [20]; BoxEL: Xiong et al., 2022 [21].

Box^{2}EL[22]: concept boxes and role position boxes

Each concept: a box and one learned bump.
Box^{2}EL: $A \mapsto (\mathrm{Box}(A), b_A)$



Stripes: concept boxes Blue: role head / tail boxes
Arrows: bumps

Each role: head and tail boxes.
$r \mapsto (H_r, T_r)$.

NF3 training: $A \sqsubseteq \exists r.B$

$$\mathrm{Box}(A) + b_B \subseteq H_r$$
$$\mathrm{Box}(B) + b_A \subseteq T_r$$

NF4 training: $\exists r.B \sqsubseteq A$

Step 1. Cover the existential region:
For any $x \in (\exists r.B)^{\mathcal{I}}$, NF3's head condition gives

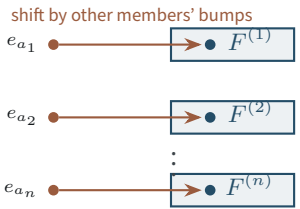
$$x + b_B \in H_r \;\Rightarrow\; x \in H_r - b_B.$$

Step 2. Require containment in A:
$H_r - b_B \subseteq \mathrm{Box}(A)$ (sufficient condition).

BoxE: the source of Box^{2}EL's point-and-bump idea

$F(a_1, \ldots, a_n)$ is a relation fact: F names the relation, and each a_i is a participating entity.

- **Example:** $F = \text{TeamMembers}$
  Each a_i is one team member.
- **Each member a_i:** a point e_{a_i} and one learned bump b_{a_i}.
- **Relation F:** one box $F^{(i)}$ for each argument position i.



One box per argument position

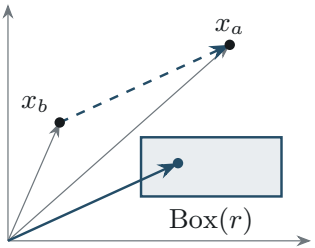
$F(a_1, \ldots, a_n)$: $e_{a_i} + \sum_{j \neq i} b_{a_j} \in F^{(i)}$ for every i. Other members' bumps only.

Abboud et al., NeurIPS 2020 (BoxE), §4, Eq. 1

TransBox[23]: roles as boxes of translations

in depth

A single translation vector → a **region of possible translations**



$r(a, b)$ iff $x_a - x_b \in \mathrm{Box}(r)$

A box for each role

Pair (a, b) satisfies r:
its displacement lies inside the role box

Many valid displacements

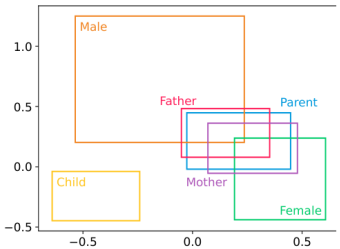
The same individual can relate to several others.

⇒ **many-to-many roles**

Concepts = regions of individuals
Roles = regions of translations

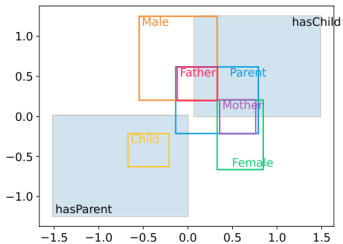
TransBox[23]: an embedding example

Family ontology (TransBox paper): a child has a mother *and* a father, and Male and Female are disjoint



(b) ELBE

ELBE: Father and Mother overlap



(d) TransBox

TransBox: they are disjoint (roles are the **large** shaded boxes)

[screenshot] Yang et al., WWW 2025 [23], Fig. 4 (b, d)

TransBox[23]: sparse intersections

Box are sparse in high dimensions spaces.

Challenge: Theorem 4.5

Centers: $[-1, 1]^n$ Offsets: $(0, a_0]^n$ ($a_0 < 1$).

$$P(B \cap B' \neq \emptyset) \leq \left(a_0 - \tfrac{1}{4}a_0^2\right)^n \quad \Longrightarrow \quad n = 50,\ a_0 \uparrow 1 :\ P < 6 \times 10^{-7}.$$

Strategy: Extend the space.

Add an empty coordinate value:

$$\mathbb{R}^n \longrightarrow (\mathbb{R} \cup \{\emptyset\})^n$$

$$\widetilde{B} = \prod_{j=1}^{n} (I_j \cup \{\emptyset\})$$

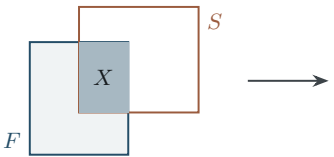
- If one axis is disjoint, its coordinate becomes $\{\emptyset\}$. Other axes may overlap.
- An adapted loss for compute the inclusion in this case.

Yang et al., WWW 2025 [23], §4.4–4.5, Thm. 4.5

Analysis (1): composing a complex concept

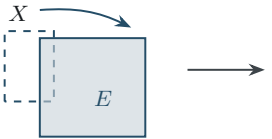
We can build **Person** $\sqcap \exists \mathit{hasChild}.(\textbf{Female} \sqcap \textbf{Student})$ in three steps

1 Intersect



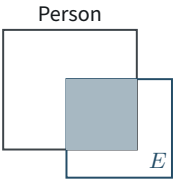
$X = \mathrm{Box}(F) \cap \mathrm{Box}(S)$
Female students

2 Apply the role



$E = X \oplus \mathrm{Box}(\mathit{hasChild})$
Those with a female student child

3 Intersect again



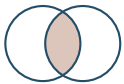
$\mathrm{Box}(\mathsf{Person}) \cap E$
The complete concept

Closure: every step returns a box $\Rightarrow$ any nesting can be built
Methods differ in steps 1, 2

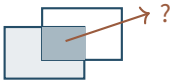
Yang et al., WWW 2025 [23], §3–4; $\oplus$: Minkowski sum; F = Female, S = Student

Analysis (2): conjunctions and NF4

Method	1 $C \sqcap D$	2 $\exists r.B$	How step 2 is built
ELEm[18]	✗	✓	translate a ball
ELBE[20]	✓	✓	translate a box
BoxEL[21]	✓	✓	affine preimage of a box
Box^{2}EL[22]	◐	✓	NF4: $H_r - b_B \subseteq \mathrm{Box}(A)$
TransBox[23]	✓	✓	box sum $\oplus$



ELEm, step 1: ball ∩ ball is a lens



Box^{2}EL, step 1: box intersection exists, but no bump for $C \sqcap D$

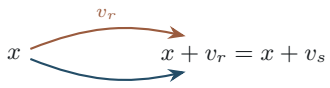
Kulmanov et al., 2019 [18]; Peng et al., 2022 [20]; Xiong et al., 2022 [21]; Jackermeier et al., 2024 [22]; Yang et al., WWW 2025 [23], §3

Analysis (3): how the methods handle role axioms

Compare the geometric condition with what each role representation can express.

Method	Role	$r \sqsubseteq s$	$r_1 \circ r_2 \sqsubseteq s$
ELEm, ELBE[18,20]	translation v_r	◐ only $v_r = v_s$	◐ only $v_{r_1} + v_{r_2} = v_s$
BoxEL[21]	affine map T_r	◐ only $T_r = T_s$	◐ only $T_{r_2} \circ T_{r_1} = T_s$
Box^{2}EL[22]	head, tail boxes	✓ $H_r \subseteq H_s, T_r \subseteq T_s$	◐ $H_{r_1} \subseteq H_s$, $T_{r_2} \subseteq T_s$
TransBox[23]	box of translations	✓ $\mathrm{Box}(r) \subseteq \mathrm{Box}(s)$	◐ $\mathrm{Box}(r_1) \oplus \mathrm{Box}(r_2) \subseteq \mathrm{Box}(s)$

ELEm / ELBE: one vector per role



$r \sqsubseteq s$ forces the two maps to coincide.

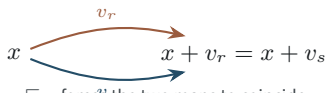
Jackermeier et al., WWW 2024 [22], §3.1–3.2; Yang et al., WWW 2025 [23], §4.1, Prop. 4.2; Mondal et al., AAAI-MAKE 2021 [19], §3.2.3 (EmEL++)

Analysis (3): how the methods handle role axioms

Compare the geometric condition with what each role representation can express.

Method	Role	$r \sqsubseteq s$	$r_1 \circ r_2 \sqsubseteq s$
ELEm, ELBE[18,20]	translation v_r	◐ only $v_r = v_s$	◐ only $v_{r_1} + v_{r_2} = v_s$
BoxEL[21]	affine map T_r	◐ only $T_r = T_s$	◐ only $T_{r_2} \circ T_{r_1} = T_s$
Box^{2}EL[22]	head, tail boxes	✓ $H_r \subseteq H_s, T_r \subseteq T_s$	◐ $H_{r_1} \subseteq H_s$, $T_{r_2} \subseteq T_s$
TransBox[23]	box of translations	✓ $\mathrm{Box}(r) \subseteq \mathrm{Box}(s)$	◐ $\mathrm{Box}(r_1) \oplus \mathrm{Box}(r_2) \subseteq \mathrm{Box}(s)$

ELEm / ELBE: one vector per role



$r \sqsubseteq s$ forces the two maps to coincide.

Can't handle transitive role ($r \circ r \sqsubseteq r$)

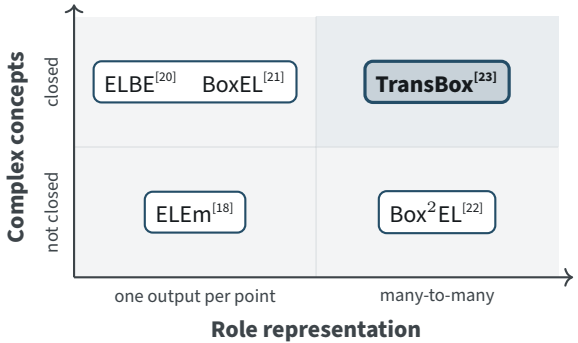
Box^{2}EL: $H_r \subseteq H_r, T_r \subseteq T_r$
Always true ⇒ every role is transitive.

TransBox: $\mathrm{Box}(r) \oplus \mathrm{Box}(r) \subseteq \mathrm{Box}(r)$
$\Rightarrow \mathrm{Box}(r) = \{\mathbf{0}\}$ or $\mathbb{R}^n$.

Jackermeier et al., WWW 2024 [22], §3.1–3.2; Yang et al., WWW 2025 [23], §4.1, Prop. 4.2; Mondal et al., AAAI-MAKE 2021 [19], §3.2.3 (EmEL++)

Analysis (4): capabilities at a glance

Two separate capabilities: composing complex concepts vs. many-to-many roles



One output per point

Translation or affine map:
each point $\mapsto$ **one** point

$\Rightarrow$ a child cannot reach both a mother and a father box

Many-to-many

Box^{2}EL: head / tail boxes
TransBox: a **box of translations**.

Right column: many-to-many roles and role axioms. **Training: see backup.**

ELEm [18], ELBE [20], BoxEL [21], Box^{2}EL [22], TransBox [23] (2019–2025)

3

Other works

Beyond the EL family
Theoretical analysis
Open problems

Other works: models and theory at a glance

brief

Different supported constructors — not a strict inclusion chain

DL-Lite$_{\mathcal{H}}$

BoxLitE[32] KR'26
convex training (SOCP)
✓ weakly faithful

EL family

DF-$\mathcal{EL}^{++}$ [33] KDD'26
fuzzy truth degrees

DELE[27] NAI'26
closure-aware negatives

$\mathcal{ALC}$ ($\neg$, $\sqcup$, $\forall$)

FALCON[24] preprint
fuzzy models

CatE[25] NAI'25
concept lattice + order emb.

Axis-aligned cones[26] JAIR'23
$\neg$ via polar cones

Theoretical analysis and results across ontology families

KB embedding semantics[29] KR'24
faithfulness, entailment closure, expressiveness

Strong faithfulness for ELH[30] TGDK'24
existence proof for convex geometric models

Discussion: hierarchy embeddings

Hyperbolic methods work well, but

- hard to implement: Riemannian geometry needs manifold-aware optimisation and numerical care.
- hard to reuse: special constructions (e.g. cones) bring method-specific order tests.

RegD[17]: a first step, hyperbolic in $\mathbb{R}^n$

- Transitivity is only partly supported: $\lambda = 0$ gives exact inclusion. $\lambda > 0$ adds depth but weakens that guarantee.
- Open score design: a sum $d_{\mathrm{bd}} + \lambda d_{\mathrm{dep}}$, a product (as in Poincaré), or another combination?

Hyperbolic optimisation: Mishne et al., ICML 2023 [13]; cones: Ganea et al., ICML 2018 [14]; RegD: Yang & Chen, KR 2026 [17], §3.3

Discussion: ontology embeddings

Problems

1. **Role axioms**: a role hierarchy is still weak.
   In particular, role chains $r_1 \circ r_2 \sqsubseteq s$.
2. **Beyond the EL family**: little work on $\mathcal{ALC}$, and mostly theoretical.
   Negation is the main problem
3. **Real-world applications**: move beyond axiom-ranking benchmarks.
   E.g., clinical ontology completion and ontology-guided biological prediction.
4. **Lack of transfer abilities**

Next (Chapter 3): language models bring in the labels

Ontology-embedding survey [3]; role-method comparison above; broader logics: FALCON [24], CatE [25].

References (1/2): foundations and hierarchies

[1] Baader, Horrocks, Lutz, Sattler. An Introduction to Description Logic. Cambridge University Press, 2017.

[2] Bordes, Usunier, García-Durán, Weston, Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. NeurIPS 2013.

[3] Chen, Mashkova, Zhapa-Camacho, Hoehndorf, He, Horrocks. Ontology Embedding: A Survey of Methods, Applications and Resources. IEEE TKDE, 2025.

[4] Vendrov et al. Order-Embeddings of Images and Language. ICLR 2016.

[5] Vilnis, Li, Murty, McCallum. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. ACL 2018.

[7] Nickel & Kiela. Poincaré Embeddings for Learning Hierarchical Representations. NeurIPS 2017.

[8] Sarkar. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. GD 2011.

[9] Sala, De Sa, Gu, Ré. Representation Tradeoffs for Hyperbolic Embeddings. ICML 2018.

[10] Bonnabel. Stochastic Gradient Descent on Riemannian Manifolds. IEEE Trans. Automatic Control, 2013.

[11] Bécigneul & Ganea. Riemannian Adaptive Optimization Methods. ICLR 2019.

[12] Yu & De Sa. Numerically Accurate Hyperbolic Embeddings Using Tiling-Based Models. NeurIPS 2019.

[13] Mishne, Wan, Wang, Yang. The Numerical Stability of Hyperbolic Representation Learning. ICML 2023.

[14] Ganea, Bécigneul, Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. ICML 2018.

[15] Yu et al. Shadow Cones: A Generalized Framework for Partial Order Embeddings. ICLR 2024.

[16] Boratko et al. Capacity and Bias of Learned Geometric Embeddings for Directed Graphs (τBox). NeurIPS 2021.

[17] Yang & Chen. RegD: Hierarchical Embeddings via Dissimilarity between Arbitrary Euclidean Regions. KR 2026.

References (2/2): ontology embeddings and theory

[18] Kulmanov et al. EL Embeddings: Geometric Construction of Models for the Description Logic EL++ (ELEm). IJCAI 2019.

[19] Mondal et al. EmEL++: Embeddings for EL++ Description Logic. AAAI-MAKE 2021.

[20] Peng et al. Description Logic EL++ Embeddings with Intersectional Closure (ELBE). arXiv 2202.14018, 2022.

[21] Xiong et al. Faithful Embeddings for EL++ Knowledge Bases (BoxEL). ISWC 2022.

[22] Jackermeier, Chen, Horrocks. Dual Box Embeddings for the Description Logic EL++ (Box^{2}EL). WWW 2024.

[23] Yang, Chen, Sattler. TransBox: EL++-closed Ontology Embedding. WWW 2025.

[24] Hinnerichs et al. FALCON: Scalable Reasoning over Inconsistent ALC Ontologies. arXiv:2208.07628v5, 2024.

[25] Zhapa-Camacho & Hoehndorf. Lattice-Based ALC Ontology Embeddings with Saturation (CatE). Neurosymbolic AI, 2025.

[26] Özçep, Leemhuis, Wolter. Embedding Ontologies in the Description Logic ALC by Axis-Aligned Cones. JAIR 2023.

[27] Mashkova, Zhapa-Camacho, Hoehndorf. DELE: Deductive EL++ Embeddings for Knowledge Base Completion. Neurosymbolic AI, 2026.

[29] Bourgaux et al. Knowledge Base Embeddings: Semantics and Theoretical Properties. KR 2024.

[30] Lacerda et al. Strong Faithfulness for ELH Ontology Embeddings. TGDK 2024.

[32] Lourenço, Morgan, Ozaki, Pavlović, Sallinger. BoxLitE: A Faithful Knowledge Base Embedding Based on Convex Optimization. KR 2026.

[33] Zhao. Fast and Faithful: Scalable Neuro-Symbolic Learning and Reasoning with Differentiable Fuzzy EL++. KDD 2026.

OntoLM Tutorial | Chapter 3 of 7

Language Model-Based Embeddings

Hui Yang
Research Associate
University of Manchester

The University of Manchester

University of Exeter

0

Why language models?

A concept comes with textual information

New class: “soybean milk” → candidate parents?

From structure alone

- Class ID: identity, little meaning
- No / few links ⇒ little to learn from

From its language

Label: soybean milk

Description: a beverage made from soybeans

Useful clues: beverage, soybean, made from

soybean milk ⇢ **soybean food product** / **beverage**

> **Text adds evidence** when structural links are sparse.

Two families of language-based methods

Two routes to embeddings from text

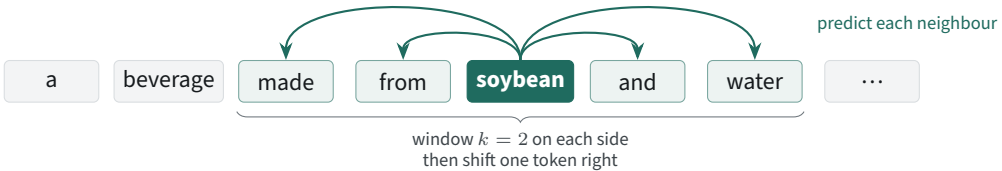
	1. Corpus-based methods	**2. Pre-trained Transformers**
Route	learn token vectors	adapt a text encoder
Core idea	ontology → context → Word2Vec[34]	text → contextual encoder → task training
Representation	fixed vector per word / entity token	context-dependent representation
Methods	RDF2Vec[39], Onto2Vec[40], OPA2Vec[41], OWL2Vec[35], **OWL2Vec*[42]**	**BERTSubs[43]**: joint pair scoring **HiT[47] / OnT[48]**: reusable concept vectors

1

Corpus-based methods: ontology as a corpus

Word2Vec[34] in one slide: predict nearby tokens

Skip-gram: use the centre word to predict tokens in a short window



How it learns

- Form pairs: (soybean, from), (soybean, water)
- Raise real-neighbour scores
  Lower sampled ones
- Shared neighbours often yield nearby vectors



after training: illustrative word vectors

No vector-arithmetic target is imposed.

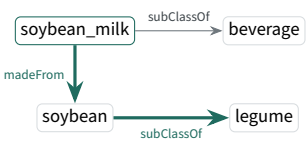
For ontologies: a corpus of class IDs, relations and words

Mikolov et al., NeurIPS 2013 [34]; illustrative sentence and vectors

Corpus-based methods: different training corpora

Each method: ontology → its own "sentences" → Word2Vec

1. Walks over an RDF graph




```
soybean_milk madeFrom soybean
soybean_milk subClassOf legume
```

RDF2Vec[39] (2016): neighbourhood as context
OWL2Vec[35] (2019): OWL → graph, biased walks

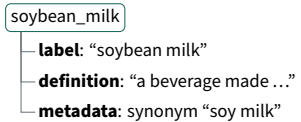
2. Axioms as sentences




```
soybean_milk SubClassOf madeFrom
some soybean
```

Onto2Vec[40] (2018): asserted + inferred axioms
OPA2Vec[41] (2019): axioms + annotations

3. Annotations as text



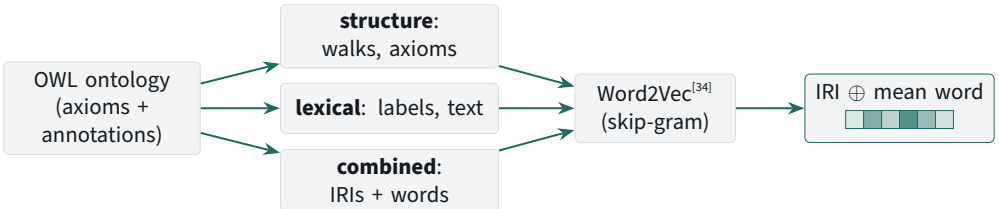

```
soybean_milk label soybean milk
```

OPA2Vec[41] (2019): + PubMed pre-training
OWL2Vec*[42] (2021): **1 + 2 + 3** together (next)

Ristoski & Paulheim 2016 [39]; Smaili et al. 2018/2019 [40,41]; Holter et al. 2019 [35]; Chen et al. 2021 [42]; illustrative

OWL2Vec*[42]: combine structure and language

Three documents → Word2Vec → entity features



- **Structure**: walks + axioms
- **Lexical**: labels + annotations
- **Combined**: one entity IRI + surrounding words

Scoring: $[v_C; v_D]$ → a separate classifier

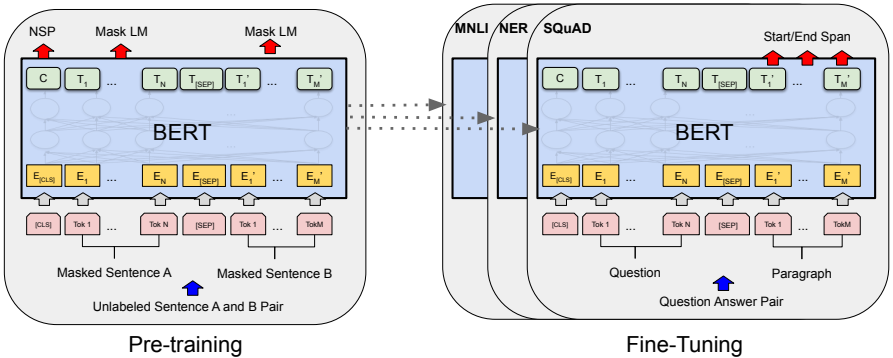
Chen et al., Machine Learning 2021 [42], §3; redrawn

2

Pre-trained Transformer methods

BERT + pair classification
Sentence-BERT + structured embeddings

BERT[36]: pre-train, then fine-tune

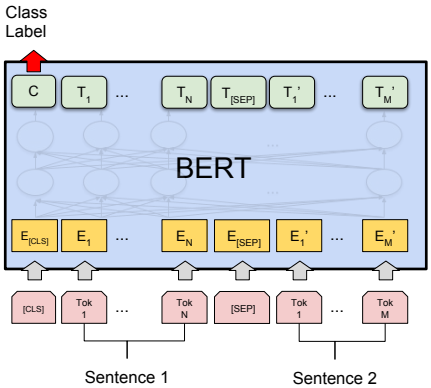


- **Pre-training**: masked tokens (MLM) + next-sentence prediction (NSP)
- **Fine-tuning**: update encoder + task head on labelled task data

Devlin et al., NAACL 2019 [36], Fig. 1; original figure, cropped

BERT[36]: read a sentence pair jointly

Bidirectional attention across both texts → one pair prediction



How to read the figure

- `[SEP]`: separates the two texts
- Tokens attend across both texts
- C: final `[CLS]` representation
- Classification head reads C

Devlin et al., NAACL 2019 [36], Fig. 4(a); original panel, cropped

BERTSubs[43] step 1: concepts as sentences

Turn each side of a candidate subsumption into text

BERTSubs text rules

Named class $A \rightarrow$ its label $L(A)$

Restriction $\exists r.A \rightarrow$ "something $L(r)$ $L(A)$"

L is a label.
Add a preposition only when needed.

One candidate $C \sqsubseteq D$

$C =$ SoybeanMilk

$D = \exists derivesFrom.$SoybeanPlant

Text for C: "soybean milk"

Text for D: "something derives from soybean plant"

`[CLS]` soybean milk `[SEP]` something derives from soybean plant `[SEP]`

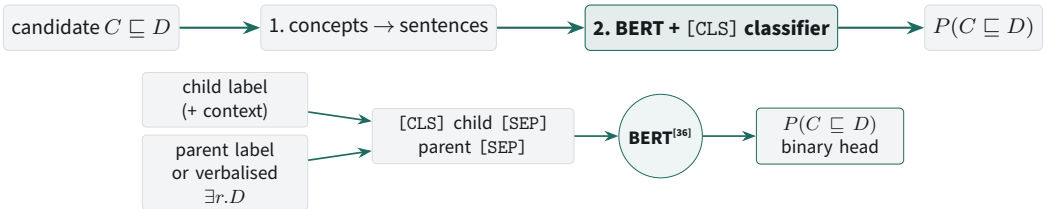
> One candidate subsumption $\rightarrow$ one sentence pair for BERT

Chen et al., World Wide Web 2023 [43], §3.2; adapted from the FoodOn example

BERTSubs step 2: a classifier on the [CLS] vector

in depth

Then read both sentences jointly with BERT (`bert-base-uncased`) and classify



- **Input**: the two sentences from step 1 (+ context), as one pair
- **Score**: [CLS] vector → linear + softmax → $P(C \sqsubseteq D)$
- **Training**: update BERT + head on positive / negative pairs
- **Output**: a directed pair score, no standalone concept vector

Cross-encoder: one joint BERT pass per candidate pair

Chen et al., World Wide Web 2023 [43], §3.1

BERTSubs[43] results: contextual vs. static vectors

FoodOn / GO: true parent vs. $\leq$50 hard negatives (not OWL2Vec*'s setting)

IC: class label only **PC**: child-down / parent-up path **BC**: breadth-first neighbours

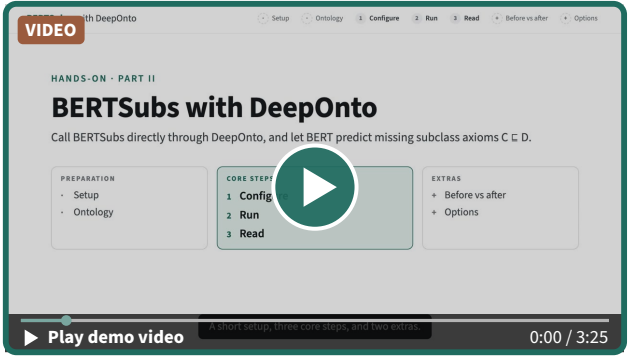
Method	FoodOn				GO			
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
TransE	0.479	0.332	0.654	0.816	0.320	0.192	0.444	0.605
Text-aware TransE	0.572	0.429	0.734	0.869	0.518	0.357	0.718	0.863
Word2Vec	0.562	0.426	0.717	0.866	0.416	0.284	0.549	0.721
Onto2Vec	0.591	0.451	0.762	0.875	0.428	0.291	0.570	0.751
OPA2Vec	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

OWL2Vec* $\rightarrow$ BERTSubs (PC), MRR: GO 0.46 $\rightarrow$ 0.61, FoodOn 0.63 $\rightarrow$ 0.64

[screenshot] Chen et al., World Wide Web 2023 [43], §3.2–3.4 and Table 3, excerpt

Demo: BERTSubs[43] with DeepOnto[45]

Call BERTSubs directly through DeepOnto: configure, run, read $P(C \sqsubseteq D)$



Click to play

`huiyang1997.github.io/OntoLM/demos/#bertsubs`

Run it yourself

Google Colab, GPU

Run all: ≈6 min

Open in Colab

`BERTSubs_Quickstart.ipynb`

Real run

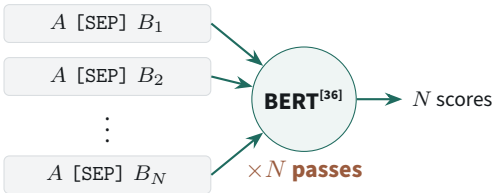
aunt, niece (no parents stated):
$P(\sqsubseteq \text{female}) = 1.00$, $P(\sqsubseteq \text{male}) = 0.00$

before fine-tuning: 0.55–0.71 for all four

DeepOnto 0.9.3 [45,46], `BERTSubsIntraPipeline`; family ontology

BERTSubs[43] at scale: one BERT[36] pass per pair

Task: find the parents of a new concept A among **all** N classes

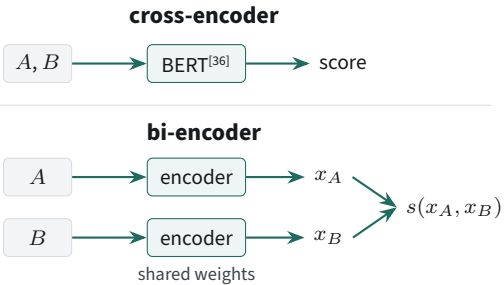


What this costs

- One new concept: N BERT passes
- Every pair: N^2 BERT passes
- SNOMED CT: $N \approx 375$k
  $\Rightarrow N^2 \approx 10^{11}$ for all pairs

Reported BERTSubs results: the true parent vs. $\leq$50 candidates, not all classes

Back to embeddings: encode each concept once



	Cross (BERTSubs[43])	Bi (HiT[47], OnT[48])
Offline	—	N passes, cache
Per query	N BERT[36] passes	N cheap scores

Reimers & Gurevych, EMNLP 2019 [37]; He et al., NeurIPS 2024 [47]; Yang et al., ISWC 2025 [48]

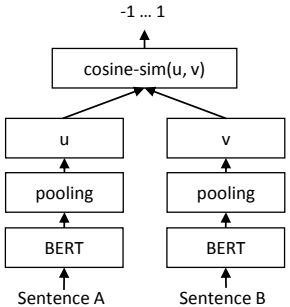
2.1

From Sentence-BERT to HiT

Encode each concept once
Then teach the vectors a hierarchy

Sentence-BERT[37]: train reusable sentence vectors

Goal: fast semantic similarity search with one vector per sentence



Supervised fine-tuning

- **Encoder**: pre-trained BERT, shared weights, pooling $\rightarrow u, v$
- **NLI data**: existing SNLI/MultiNLI pairs (entailment / neutral / contradiction)
- **STS data**: existing STS-B pairs with similarity scores (another variant)

Use: encode separately compare by cosine cache vectors for search.

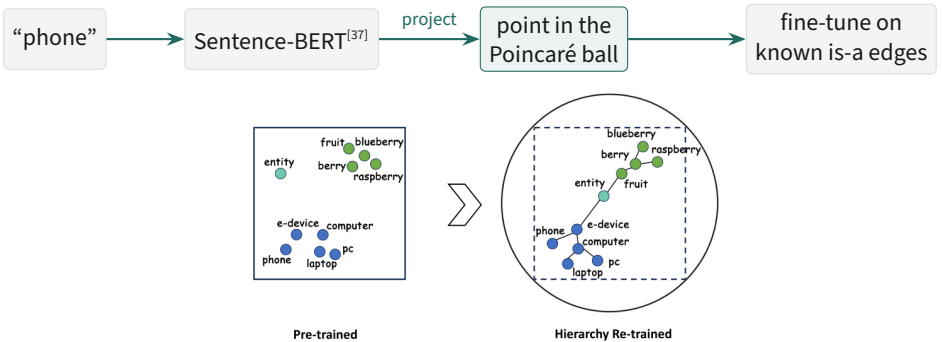
HiT / OnT: adapt these sentence vectors from similarity to hierarchy

Reimers & Gurevych, EMNLP 2019 [37], §3–4 and Fig. 2; original figure, cropped

HiT[47]: language models as hierarchy encoders

in depth

Idea: Sentence-BERT vectors → Poincaré ball → fine-tune on a known hierarchy



Label embeddings before (left) and after (right) fine-tuning

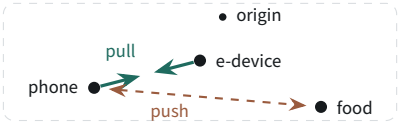
The encoder learns the hierarchy: general near the origin, specific near the edge

[screenshot] He et al., NeurIPS 2024 [47], Fig. 1

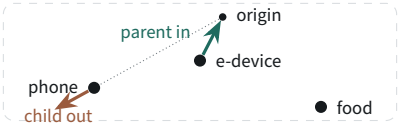
HiT[47] fine-tuning loss

For each known axiom $C \sqsubseteq D$ with a negative N (e.g. phone $\sqsubseteq$ e-device, $N =$ food)

$$\mathcal{L}_{\text{HiT}} = \underbrace{\left[\, d_\kappa(x_C, x_D) - d_\kappa(x_C, x_N) + \alpha \,\right]_+}_{\textbf{1}\text{ clustering}} + \underbrace{\left[\, \|x_D\|_\kappa - \|x_C\|_\kappa + \beta \,\right]_+}_{\textbf{2}\text{ centripetal}}$$



1 child near its parent, far from the negative



2 parent closer to the origin than its child

He et al., NeurIPS 2024 [47], Eqs. 3–6; Yang et al., ISWC 2025 [48], Fig. 2 (redrawn); $\|x\|_\kappa = d_\kappa(x, 0)$

HiT[47] score function

After fine-tuning: score any candidate $C \sqsubseteq D$ from its two vectors

HiT[47]: additive (λ tuned on validation)

$$s(C \sqsubseteq D) = -\big(\underbrace{d_\kappa(x_C, x_D)}_{\text{close?}} + \lambda \underbrace{\big(\|x_D\|_\kappa - \|x_C\|_\kappa\big)}_{\text{parent more general?}} \big)$$

Poincaré embeddings[7] (Nickel & Kiela 2017 [7]): multiplicative

$$s(C \sqsubseteq D) = -\big(1 + \alpha\,(\|x_D\| - \|x_C\|)\big) \cdot d(x_C, x_D)$$

Same ingredients (distance, norm): Poincaré emb. multiplies, HiT adds

HiT[47] results: multi-hop inference and transfer

Multi-hop inference on WordNet, F1

	random	hard
Poincaré emb.	0.86	0.83
Hyperbolic cones	0.90	0.38
MiniLM, pre-trained	0.21	0.19
MiniLM, fine-tuned	0.63	0.64
HiT[47] (MiniLM)	**0.90**	**0.91**

- Cones: hard siblings → F1 0.38
- Poincaré emb.: no encoder for unseen labels
- **Transfer** WordNet → DOID: HiT best (F1 0.70 / 0.57)

He et al., NeurIPS 2024 [47], Tables 2–3

HiT[47] vs. earlier hierarchy embeddings

	Earlier embeddings Poincaré embeddings, cones	**HiT**[47] Language-model encoder
Input	IDs only	Concept labels (text)
New concepts	Cannot handle unseen concepts	Encodes unseen concepts
Transfer learning	No transfer learning ability	Transfers across ontologies

Nickel & Kiela, NeurIPS 2017 [7]; Ganea et al., ICML 2018 [14]; He et al., NeurIPS 2024 [47], Table 3

2.2

OnT: from hierarchies to ontologies

From HiT[47] to OnT[48]: concept names are not enough

HiT targets **concept names**, but complex concepts bring logical operators and roles

What HiT[47] handles, and what it does not

- ✓ **Concept names**: is-a between names, e.g. Father $\sqsubseteq$ Parent
- × **Logical operators** $\sqcap$, $\exists$: verbalised as text, logic not modelled
- × **Roles** r: no role embedding

e.g. Father $\sqsubseteq$ Male $\sqcap \exists hasChild.$Person: read as plain text only

OnT[48]: Ontology Transformer encoder

- $\Rightarrow$ Embed any $\mathcal{EL}$ concept, named or complex
- $\Rightarrow$ Roles as functions

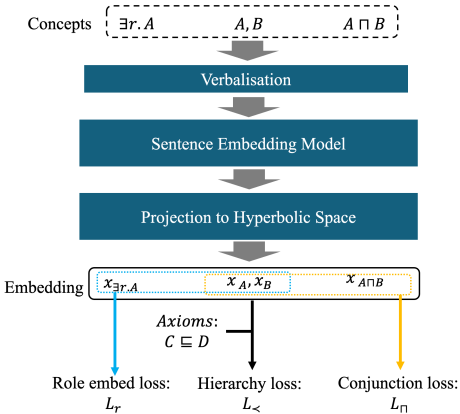
Yang et al., ISWC 2025 [48], §1; illustrative axioms

OnT[48] framework overview

Main steps

1. **Embed** concepts and roles in hyperbolic space
2. **Train** the embeddings with three losses

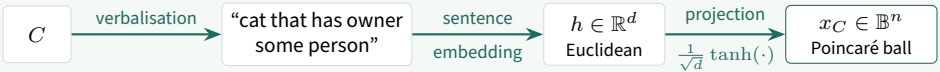
Hierarchy loss $\mathcal{L}_{\prec}$: from HiT
Role loss $\mathcal{L}_r$, conjunction loss $\mathcal{L}_{\sqcap}$: new



[screenshot] Figure from the OnT talk (ISWC 2025) [48]

Main component 1: concept embedding

Embed $C := \text{Cat} \sqcap \exists hasOwner.\text{Person}$



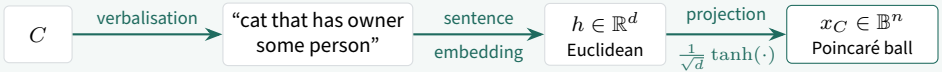
For any concept C and role $r \Rightarrow$ embeddings x_C and x_r

Verbalisation here: OntoLAMA/DeepOnto’s recursive concept verbaliser

Yang et al., ISWC 2025 [48], §4.1

Main component 1: concept embedding

Embed $C := \text{Cat} \sqcap \exists hasOwner.\text{Person}$



For any concept C and role $r \Rightarrow$ embeddings x_C and x_r

Verbalisation here: OntoLAMA/DeepOnto's recursive concept verbaliser

Challenge: text-only $x_{\exists r.C}$, $x_{\exists r.D}$ miss the monotonicity of $\exists r$:

$$\forall r. \quad C \sqsubseteq D \;\Rightarrow\; \exists r.C \sqsubseteq \exists r.D$$

Yang et al., ISWC 2025 [48], §4.1

Main component 1: role embedding

Role embedding

Model role r as a function f_r in hyperbolic space:

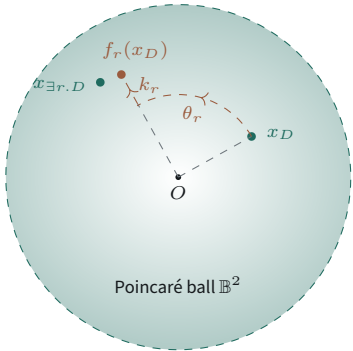
- f_r: 2-D rotations by θ_r, scaling factor k_r

$$f_r(x) = k_r \odot \big(R(\Theta_r)\, x\big)$$

- Key property (mimics monotonicity of $\exists r$):

$$d_\kappa(x_C, x_D) = d_\kappa\big(f_r(x_C), f_r(x_D)\big) \text{ when } k_r = 1$$

- Learn f_r by making $x_{\exists r.D} \approx f_r(x_D)$
  (θ_r, k_r): from x_r via a linear layer



Yang et al., ISWC 2025 [48], §4.2, Fig. 1 (recoloured)

Main component 2: training loss for $\exists r.D$

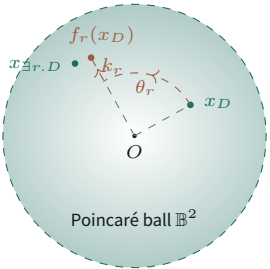
Building block: HiT's loss $\mathcal{L}_{\prec}(x_C \prec x_D)$ per axiom $C \sqsubseteq D$ (clustering + centripetal)

Modelling existential restrictions $\exists r.D$

Goal: $x_{\exists r.D} \sim f_r(x_D)$

$$\mathcal{L}_r(\exists r.D) = \tfrac{1}{2}\Big(\mathcal{L}_{\prec}\big(x_{\exists r.D} \prec f_r(x_D)\big) + \mathcal{L}_{\prec}\big(f_r(x_D) \prec x_{\exists r.D}\big)\Big)$$

Note: a direct distance loss $\|x_{\exists r.D} - f_r(x_D)\| \Rightarrow$ lower embedding quality



Main component 2: conjunctions and the final loss

Modelling conjunctions $C \sqcap D$

Goal: make sure $C \sqcap D \sqsubseteq C$ and $C \sqcap D \sqsubseteq D$

$$\mathcal{L}_{\sqcap}(C \sqcap D) = \tfrac{1}{2}\Big(\mathcal{L}_{\prec}(x_{C\sqcap D} \prec x_C) + \mathcal{L}_{\prec}(x_{C\sqcap D} \prec x_D)\Big)$$

Final loss: mean value of the three losses

$$\sum_{C \sqsubseteq D} \frac{\mathcal{L}_{\prec}(x_C \prec x_D)}{N} + \sum_{\exists r.D \text{ appears}} \frac{\mathcal{L}_r(\exists r.D)}{N_1} + \sum_{C \sqcap D \text{ appears}} \frac{\mathcal{L}_{\sqcap}(C \sqcap D)}{N_2}$$

Yang et al., ISWC 2025 [48], §4.3

Main component 2: conjunctions and the final loss

Modelling conjunctions $C \sqcap D$

Goal: make sure $C \sqcap D \sqsubseteq C$ and $C \sqcap D \sqsubseteq D$

$$\mathcal{L}_{\sqcap}(C \sqcap D) = \tfrac{1}{2}\Big(\mathcal{L}_{\prec}(x_{C\sqcap D} \prec x_C) + \mathcal{L}_{\prec}(x_{C\sqcap D} \prec x_D)\Big)$$

Final loss: mean value of the three losses

$$\sum_{C \sqsubseteq D} \frac{\mathcal{L}_{\prec}(x_C \prec x_D)}{N} + \sum_{\exists r.D \text{ appears}} \frac{\mathcal{L}_r(\exists r.D)}{N_1} + \sum_{C \sqcap D \text{ appears}} \frac{\mathcal{L}_{\sqcap}(C \sqcap D)}{N_2}$$

Compared with geometric embedding models

- Geometric models: normalise first (NF1–NF4)
- OnT: any axiom $C \sqsubseteq D$, no need for normalisation

Yang et al., ISWC 2025 [48], §4.3

OnT[48] score function

Inference with the score function (as in HiT):

$$s(C \sqsubseteq D) \equiv s(x_C \prec x_D) := -\big(d_\kappa(x_C, x_D) + \lambda\left(\|x_D\|_\kappa - \|x_C\|_\kappa\right)\big)$$

Yang et al., ISWC 2025 [48], §4.3, Prop. 1

OnT[48] score function

Inference with the score function (as in HiT):

$$s(C \sqsubseteq D) \equiv s(x_C \prec x_D) := -\big(d_\kappa(x_C, x_D) + \lambda\,(\|x_D\|_\kappa - \|x_C\|_\kappa)\big)$$

Proposition (rotation invariance)

For any $x_C, x_D \in \mathbb{B}^{2m}$ and rotation f_r with $k_r = 1$:

$$s\big(f_r(x_C) \prec f_r(x_D)\big) = s(x_C \prec x_D)$$

$\Rightarrow$ Role embeddings respect the monotonicity of $\exists r$:

$$\forall r. \quad C \sqsubseteq D \;\Rightarrow\; \exists r.C \sqsubseteq \exists r.D$$

Yang et al., ISWC 2025 [48], §4.3, Prop. 1

OnT[48] prediction results

Data: GALEN (clinical terms), GO (gene functions), Anatomy (Uberon, cross-species anatomy)

	GALEN			GO			Anatomy		
Method	**H@k**	**MRR**	**MR**	**H@k**	**MRR**	**MR**	**H@k**	**MRR**	**MR**
ELEm[18]	14/**50**/68	26	2,715	4/35/68	14	11,764	10/53/78	24	1,588
ELBE[20]	9/37/55	18	4,661	10/30/43	18	10,236	9/41/66	20	2,672
BoxEL[21]	0/0/2	0	13,824	0/0/2	0	65,846	1/2/4	2	12,257
Box^{2}EL[22]	12/38/58	21	4,593	8/43/64	19	7,975	11/39/65	20	2,828
TransBox[23]	11/41/62	22	2,972	8/43/67	19	7,092	9/49/73	22	1,299
OPA2Vec[41]	0/1/4	1	13,547	0/1/4	0	18,493	0/5/17	2	9,537
OWL2Vec*[42]	0/1/5	1	13,660	0/0/2	0	19,523	0/3/11	2	10,309
HiT[47]	**25**/47/62	33	2,349	36/60/73	44	15,080	19/54/78	31	722
OnT[48] (w/o r)	**26**/46/64	33	1,546	**38**/66/79	**48**	2,209	**22**/52/79	31	628
OnT[48]	**25**/**50**/**69**	**34**	**792**	37/**67**/**81**	46	**1,121**	**22**/**57**/**82**	**33**	**475**

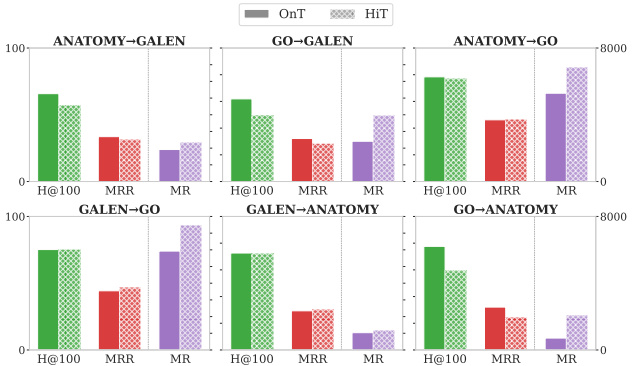
H@k: $k = 1/10/100$ (%)
MRR (%)
OnT (w/o r): without role embeddings as functions f_r

- Best overall: GO MR 7,092 (best geometric) $\rightarrow$ 1,121, and lower MR than OnT (w/o r) everywhere

Yang et al., ISWC 2025 [48], Table 2

OnT[48] transfer learning results

Train on one ontology, test on another: OnT vs. HiT (all-MiniLM-L12-v2)

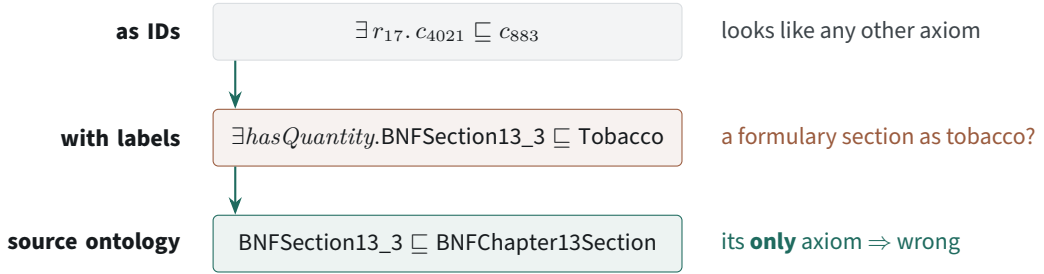


Better transfer: higher H@100 and lower MR than HiT in all six directions (MRR: mixed)

[screenshot] Yang et al., ISWC 2025 [48], Fig. 3; setup and inference results in backup

OnT[48] data note: an error that labels make visible

While NF3 results stayed weak, we read the training axioms with their labels

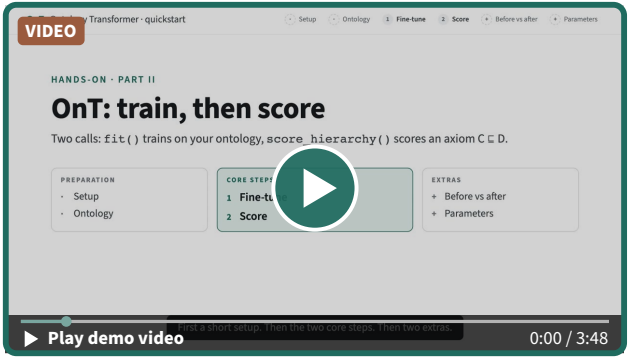


> Text helps spot errors that are invisible in IDs alone

Yang et al., ISWC 2025 [48], §5.1 footnote and App. B; IDs in the first row are illustrative

Demo: OnT[48] quickstart

Two core calls: `OntologyTransformer.fit(...)` trains, `score_hierarchy(C, D)` scores $C \sqsubseteq D$



Click to play

`huiyang1997.github.io/OntoLM/demos/#ont`

Run it yourself

Google Colab, free CPU

Run all: ≈8 min

Open in Colab

`OnT_Quickstart.ipynb`

Real run (higher = more plausible)

father ⊑ parent	stated	-8.89
father ⊑ male	stated	-9.86
father ⊑ mother	false	-10.06
father ⊑ child	false	-15.39

`ontology-transformer` 0.1.8 [50]; `family.owl` (7 axioms)

LM-based ontology embeddings side by side

Static token vectors $\rightarrow$ pre-trained pair classifiers and sentence encoders

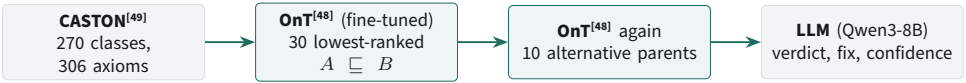
Method	Output	Complex concepts	Role embedding	New labels	One vector per concept
OWL2Vec*[42]	vector	×	×	partly[1]	✓
BERTSubs[43]	score per pair	✓[2]	×	✓	×
HiT[47]	point in $\mathbb{B}^d$	✓[2]	×	✓	✓
OnT[48]	point in $\mathbb{B}^d$	✓	✓ (f_r)	✓	✓

[1] only through known words. [2] any concept can be verbalised and encoded as text

> The step from HiT to OnT: role embeddings and losses for $\sqcap$, $\exists r.C$

Application: OnT[48] for ontology refinement

Case study: find and repair wrong is-a axioms in CASTON (metal casting)



Result (two domain experts)

- Fine-tuned OnT + Qwen3-8B: **6 / 6** high-confidence repairs accepted
- Fine-tuning matters: asserted parent in top 10 for 54.8% → 97.9%

Accepted repair

$\texttt{chromite_sand} \sqsubseteq \texttt{green_sand_molding}$
$\Rightarrow \texttt{chromite_sand} \sqsubseteq \texttt{foundry_sand}$

a material was placed under a process

OnT narrows the search → the LLM proposes → experts decide

Yang et al., ISWC 2026 [49] Posters & Demos (CASTON case study)

3

Discussion

Future work
Takeaways

Future work (1): richer input, embeddings + LLMs

Text → embedding models → embedding methods → enhancement



1. More information in the input

- Add definitions, synonyms, and parent paths
- Generate descriptions for weak labels
- Retrieve external evidence

2. Better embeddings + LLM reasoning

- Try larger embedders: EmbeddingGemma (308M), Qwen3-Embedding (0.6B)
- Combing with strong LLM (e.g., reranking)

Wei et al., NeurIPS 2022 [52]; Vera et al. [53]; Zhang et al. [54]

Future work (2): embedding methods, enhancement



3. Stronger embedding methods

- Adapt embeddings with region-based methods
- Explore full negation beyond OWL 2 EL

4. Reasoning-based enhancement

- **Data enhancement**: add deductive closure to the training data
- **Result enhancement**: apply logical consistency checks to predictions
- **Filtering enhancement**: filter out nonsensical combinations in advance

W3C OWL 2 EL profile; Mashkova et al., NeSy 2024 [51]

References (1/2): foundations and corpus methods

[34] Mikolov, Sutskever, Chen, Corrado & Dean. Distributed Representations of Words and Phrases and their Compositionality. NeurIPS 2013.

[35] Holter, Myklebust, Chen & Jiménez-Ruiz. Embedding OWL Ontologies with OWL2Vec. ISWC Posters & Demos 2019.

[36] Devlin, Chang, Lee & Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.

[37] Reimers & Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP 2019.

[39] Ristoski & Paulheim. RDF2Vec: RDF Graph Embeddings for Data Mining. ISWC 2016.

[40] Smaili, Gao & Hoehndorf. Onto2Vec: Joint Vector-Based Representation of Biological Entities and Their Ontology-Based Annotations. Bioinformatics 2018.

[41] Smaili, Gao & Hoehndorf. OPA2Vec: Combining Formal and Informal Content of Biomedical Ontologies to Improve Similarity-Based Prediction. Bioinformatics 2019.

[42] Chen, Hu, Jiménez-Ruiz et al. OWL2Vec*: Embedding of OWL Ontologies. Machine Learning 2021.

[43] Chen, He, Geng et al. Contextual Semantic Embeddings for Ontology Subsumption Prediction (BERTSubs). World Wide Web 2023.

References (2/2): encoders and applications

[45] He, Chen, Dong et al. DeepOnto: A Python Package for Ontology Engineering with Deep Learning. Semantic Web 15(5), 2024.

[47] He, Yuan, Chen & Horrocks. Language Models as Hierarchy Encoders (HiT). NeurIPS 2024.

[48] Yang, Chen, He, Gao & Horrocks. Language Models as Ontology Encoders (OnT). ISWC 2025.

[49] Yang, Li, Liu, Jarfors, Tan & Chen. Ontology Refinement with Embeddings and LLMs: A Case Study on CASTON. ISWC 2026 Posters & Demos.

[50] OnT code and models: `github.com/HuiYang1997/OnT`; notebook `OnT_Quickstart.ipynb`; PyPI `ontology-transformer` 0.1.8; Hugging Face `Hui97/OnT-*`; refinement: `github.com/HuiYang1997/OntRefine`.

[51] Mashkova, Zhapa-Camacho & Hoehndorf. Enhancing Geometric Ontology Embeddings for $\mathcal{EL}^{++}$ with Negative Sampling and Deductive Closure Filtering. NeSy 2024; extended as DELE, arXiv:2411.01574.

[52] Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022.

[53] Vera et al. EmbeddingGemma: Powerful and Lightweight Text Representations. arXiv:2509.20354, 2025.

[54] Zhang et al. Qwen3 Embedding: Advancing Text Embedding and Reranking. arXiv:2506.05176, 2025.

Geometric baselines: see Part I references.

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

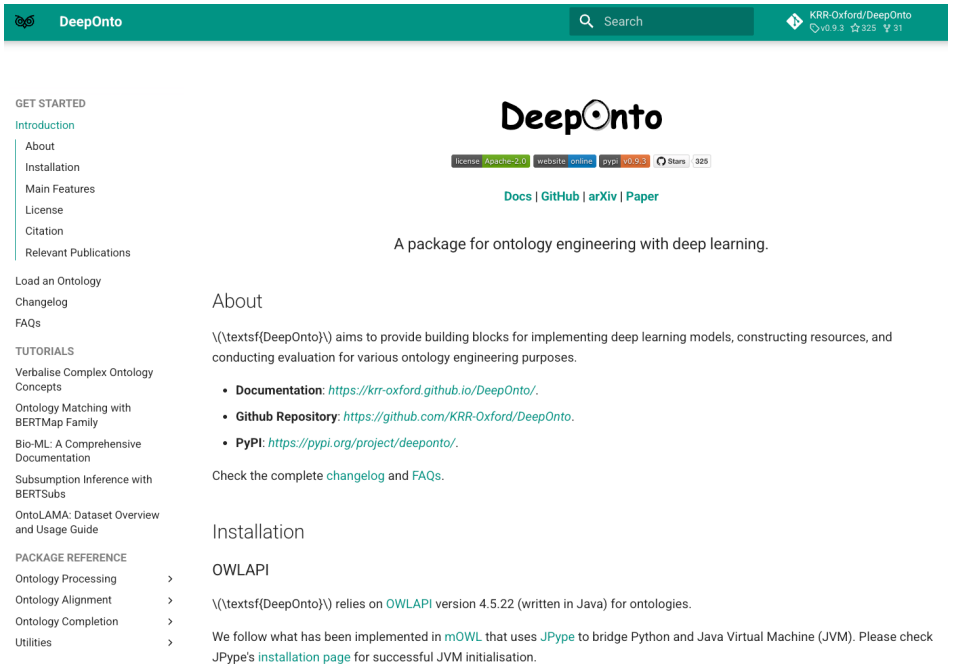
Yuan He [a], Jiaoyan Chen [b,a], Hang Dong [a], Ian Horrocks [a], Carlo Allocca [c], Taehun Kim [c] and Brahmananda Sapkota [c]

[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

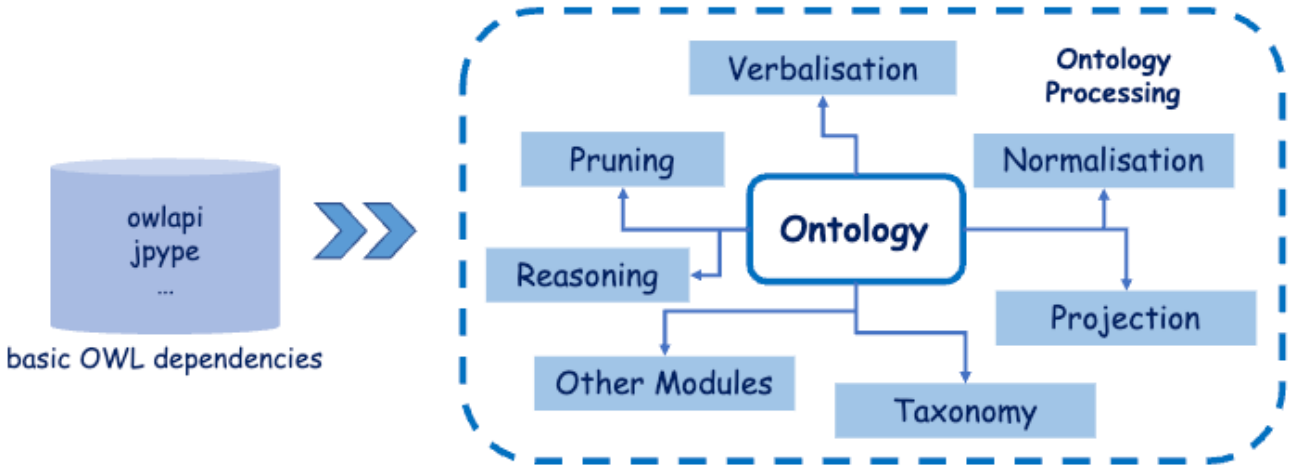
Abstract. Applying deep learning techniques, particularly language models (LMs), in ontology engineering has raised widespread attention. However, deep learning frameworks like PyTorch and Tensorflow are predominantly developed for Python programming, while widely-used ontology APIs, such as the OWL 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:

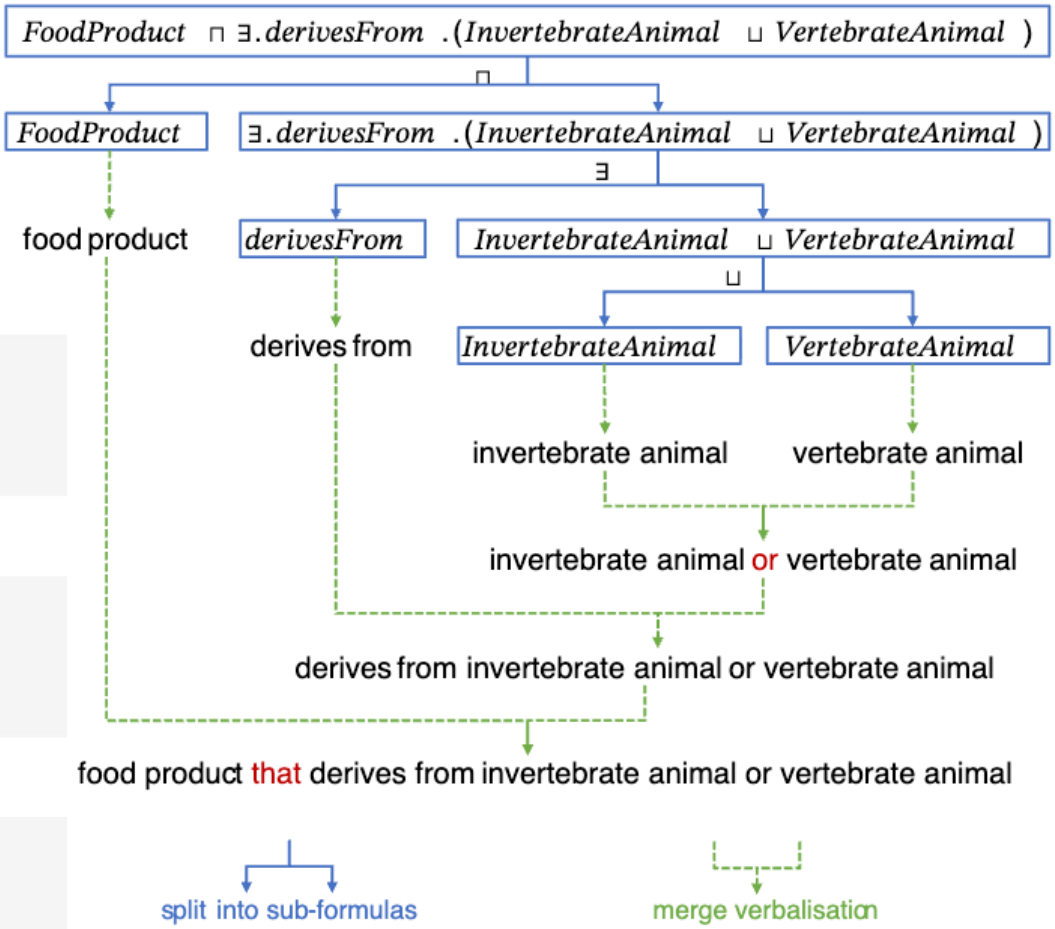
```
# get complex concepts asserted in the ontology
complex_concepts = list(onto.get_asserted_complex_classes())

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

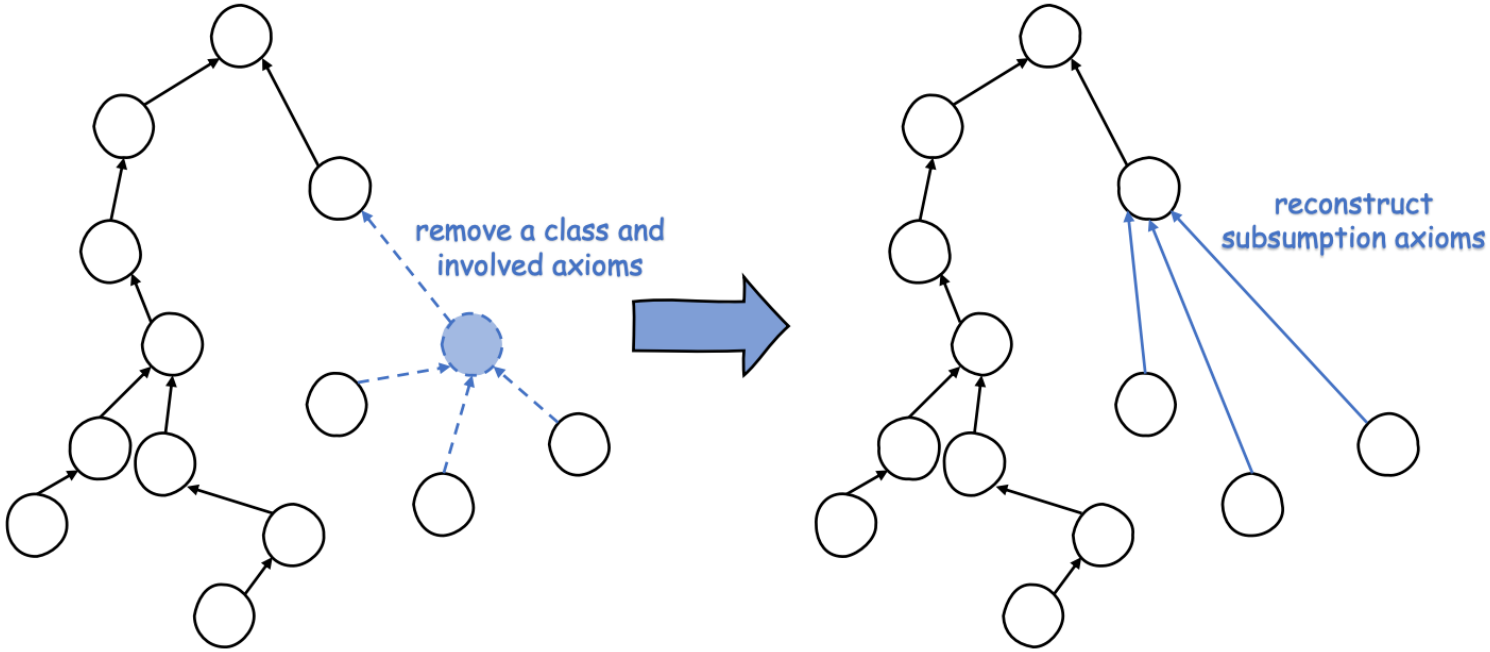
To verbaliser a class subsumption axiom:

```
# 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

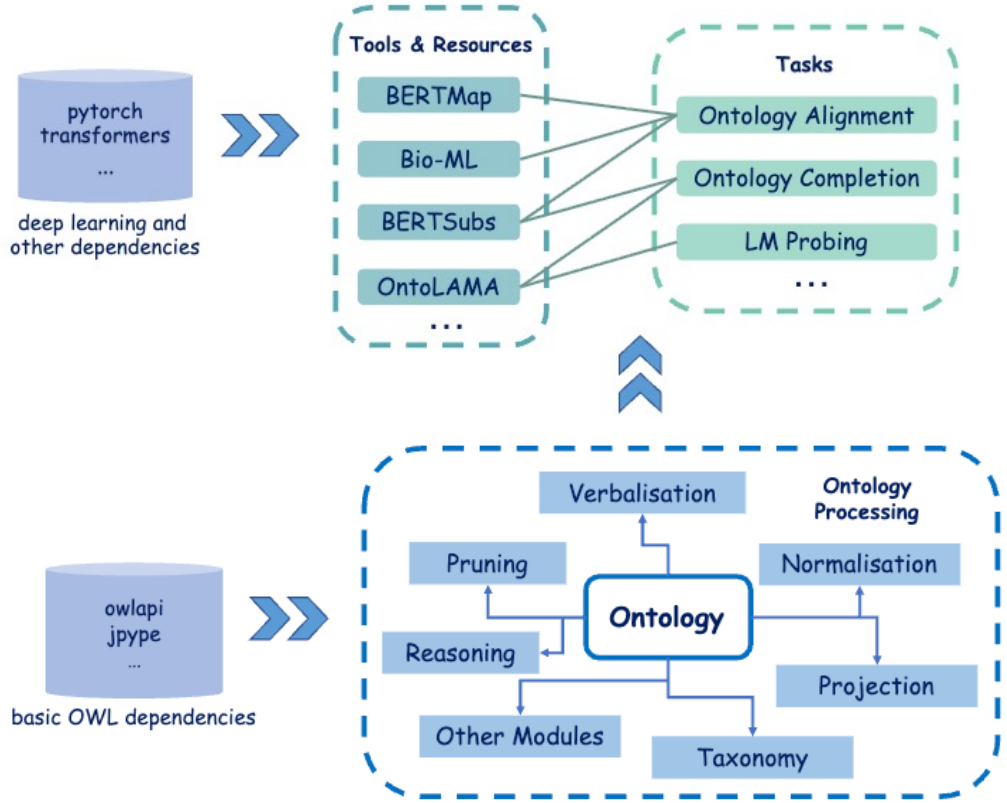


Ontology Processing

- 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:

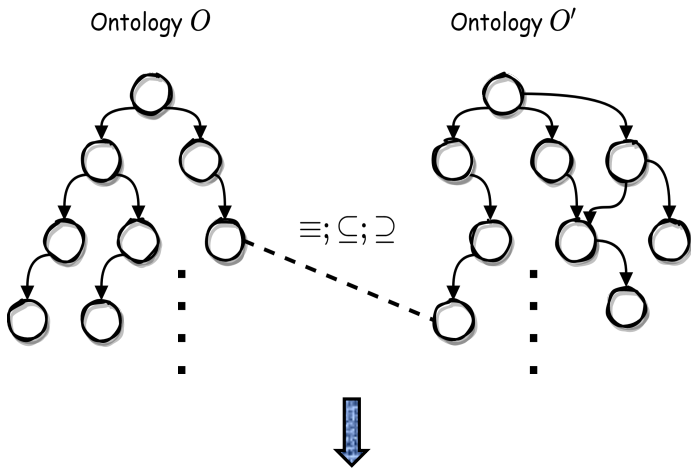
$$C \sqsubseteq D, C \sqcap C' \sqsubseteq D, C \sqsubseteq \exists r.D, \exists r.C \sqsubseteq D, r \sqsubseteq s, \text{and } r \circ r' \sqsubseteq s,$$

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***.

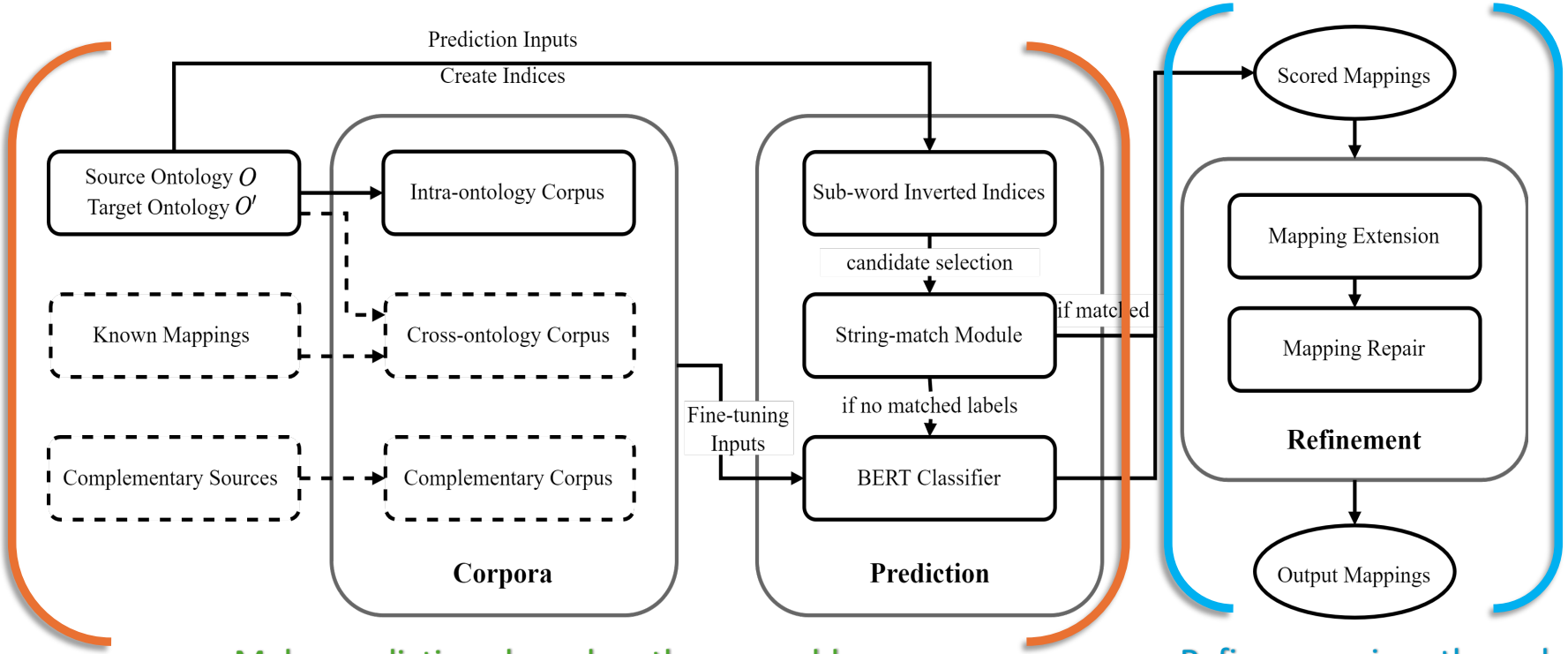
BERT (LM Encoder)

- **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.

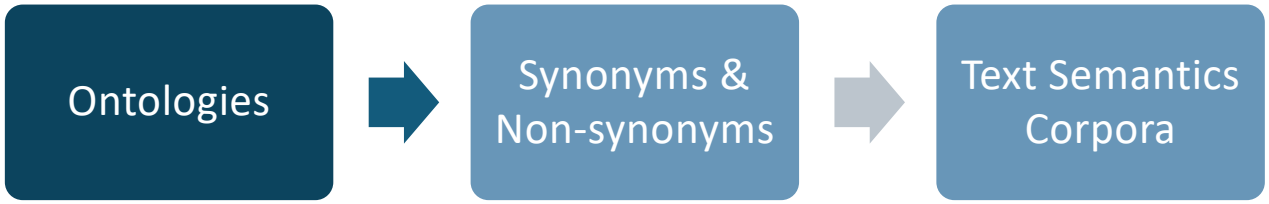
BERTMap Outline



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)

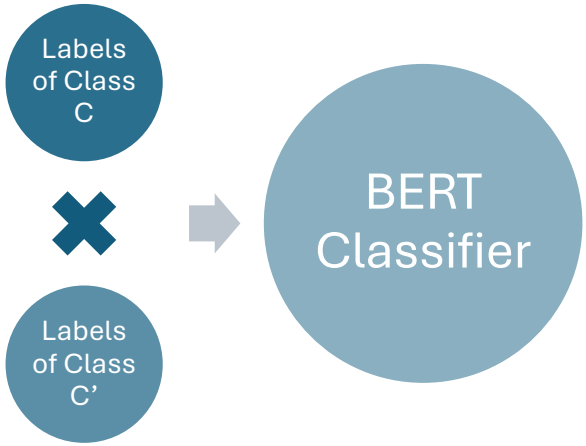
- *Synonyms* are extracted from label pairs of the matched classes.
- *Non-synonyms* are extracted from randomly aligned classes.

Complementary (from an auxiliary ontology)

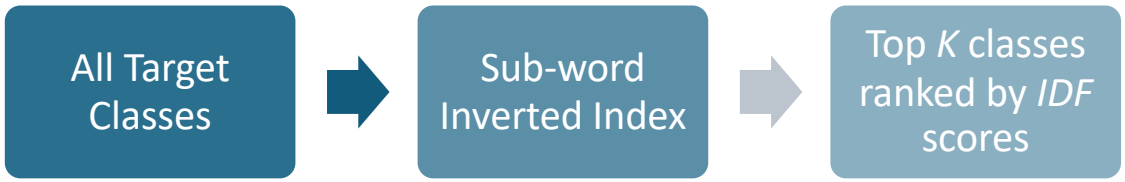
- Augmenting data for training the synonym classifier.

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

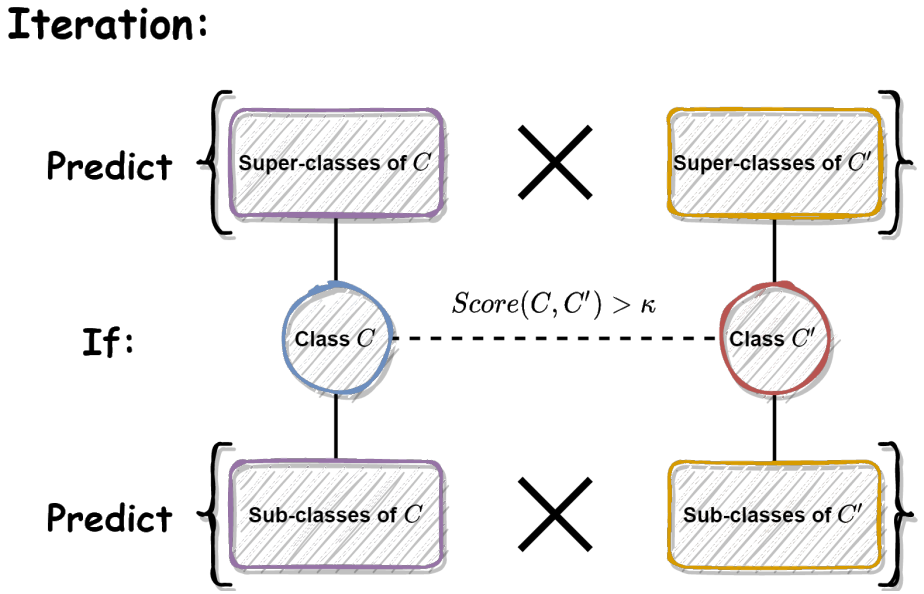


- For each source class *C*
  - Select target classes that share at least one sub-word
  - Get top *K* of them according to IDF score.
- **Outcome**: Reducing the quadratic search space to linear.
- **Limitation**: Miss classes that share no sub-word token with the given class.
- **Advantages**: Various 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"]

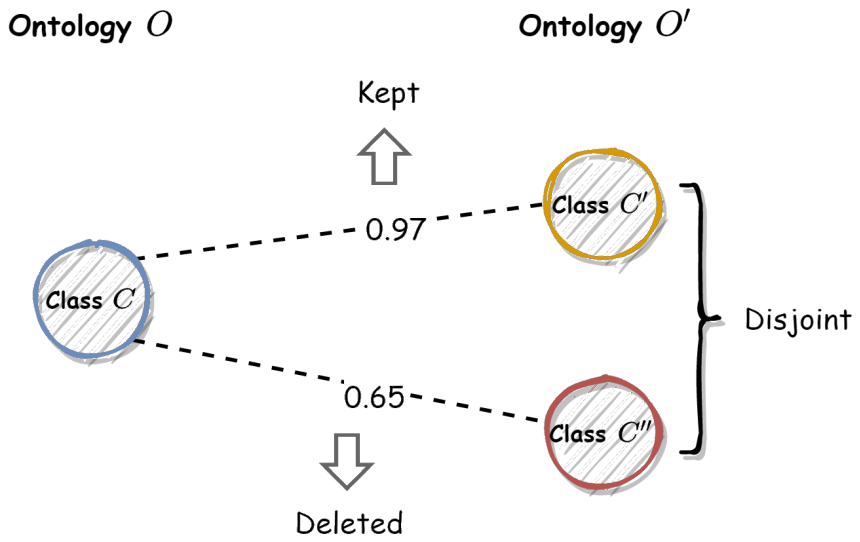
Mapping Refinement: Extension

- **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).



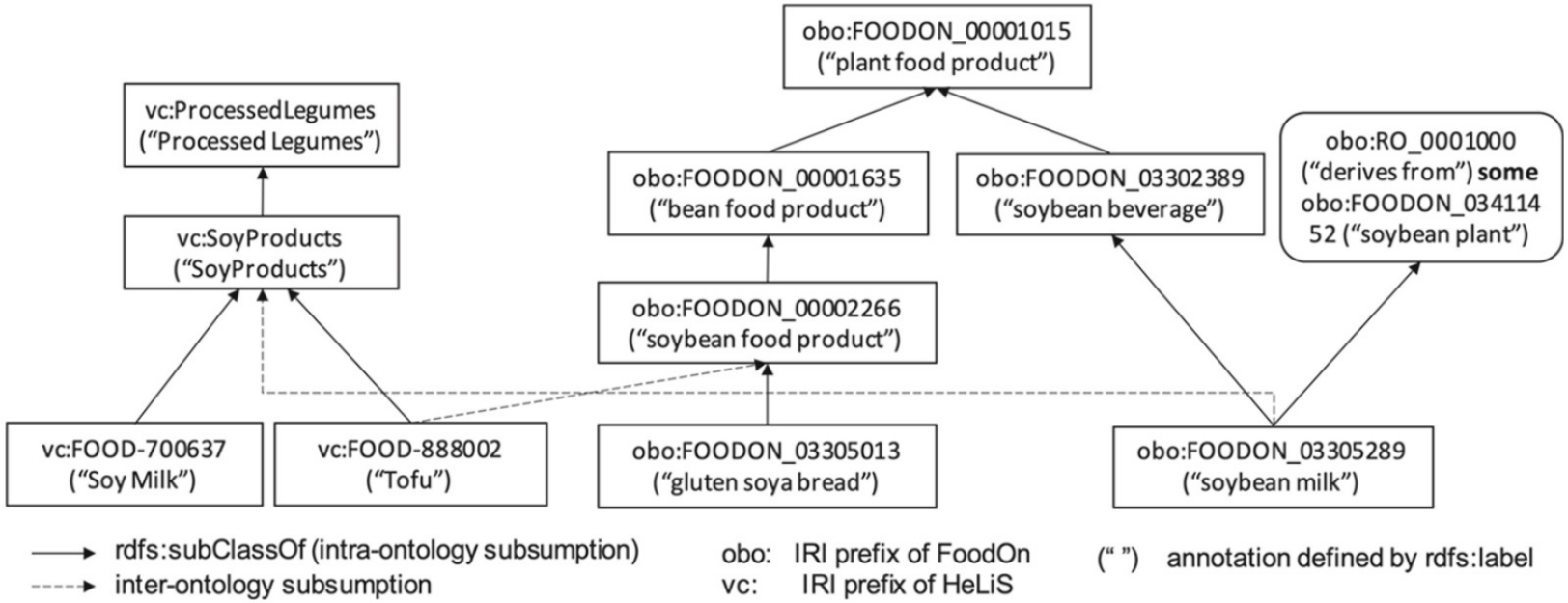
Mapping Refinement: Repair

- **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).

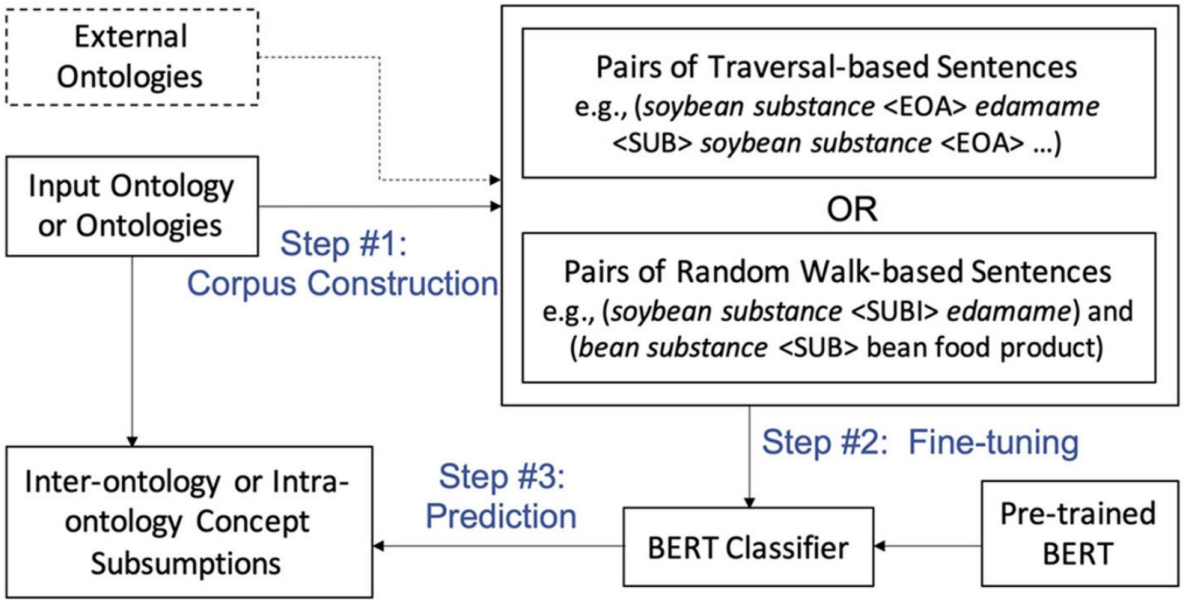
From Equivalence to Subsumption



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.

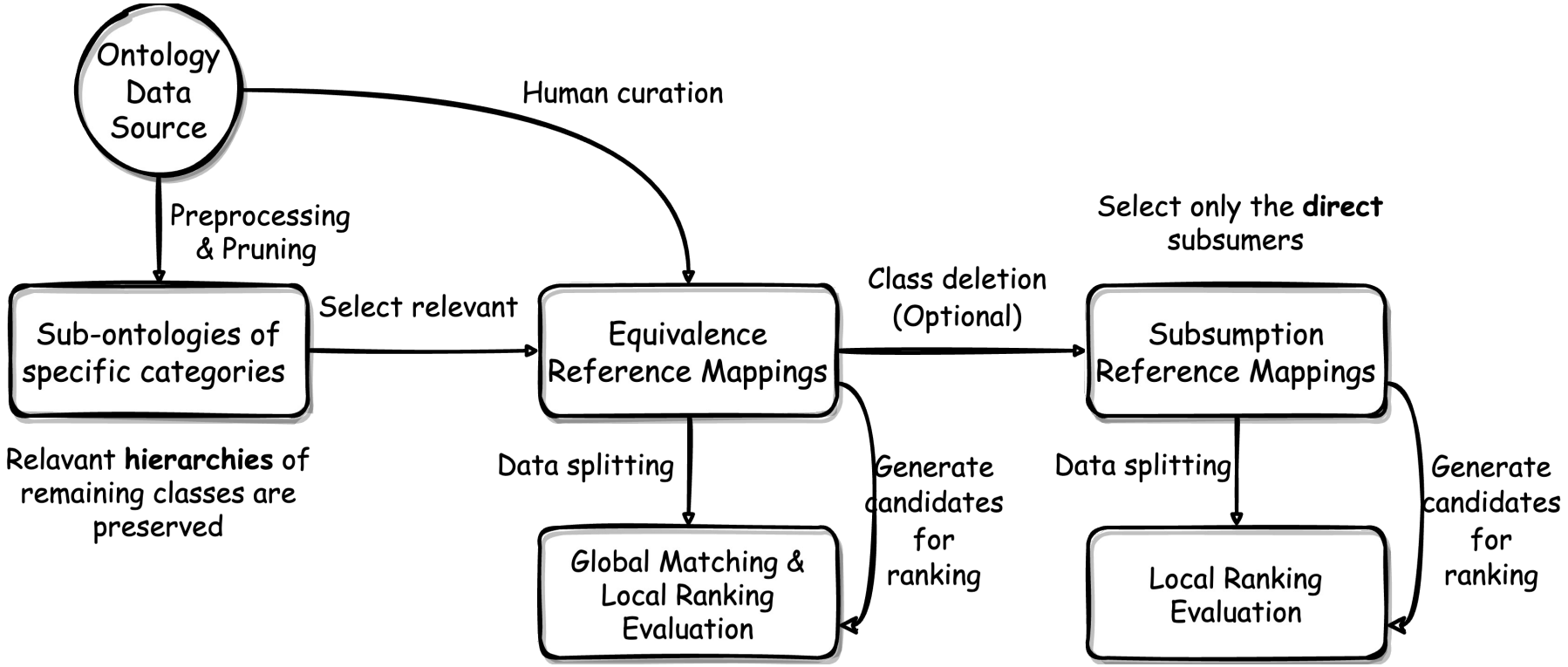
Bio-ML: New OM Benchmarks

- 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

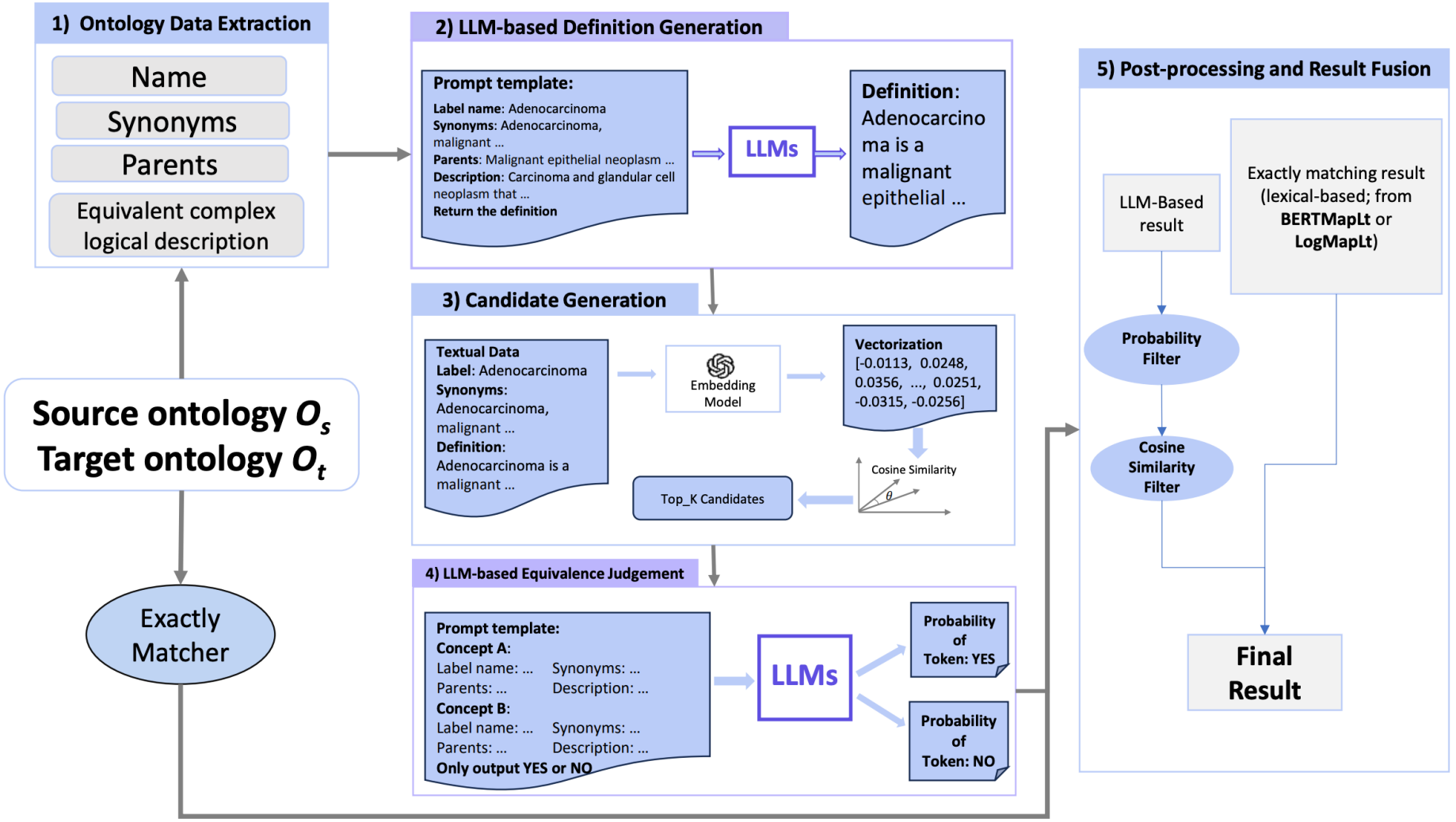
Subsumption Matching Results for SNOMED-FMA (Body)

⇕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?)

OntoLM Tutorial | Chapter 5 of 7

ISWC BARI 2026

Complex Ontology Reasoning with LLMs

Hui Yang
Research Associate
University of Manchester

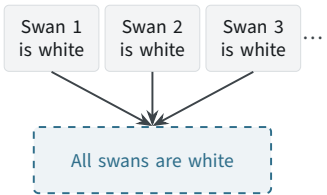
MANCHESTER 1824
The University of Manchester

University of Exeter

Three modes of reasoning

Induction

evidence ⇒ **plausible hypothesis**



probable

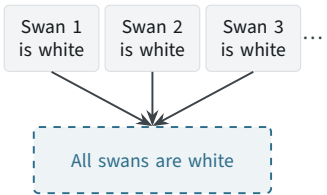
a black swan breaks it

Parts I–II: embedding methods

Three modes of reasoning

Induction

evidence ⇒ **plausible hypothesis**



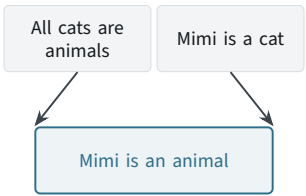
probable

a black swan breaks it

Parts I–II: embedding methods

Deduction

premises ⇒ **conclusion**



entailed

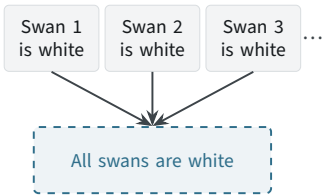
the conclusion follows if premises hold

§1: deductive benchmarks and proofs

Three modes of reasoning

Induction

evidence ⇒ **plausible hypothesis**



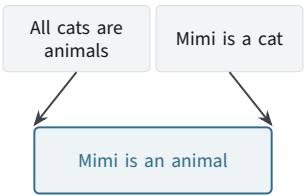
probable

a black swan breaks it

Parts I–II: embedding methods

Deduction

premises ⇒ **conclusion**



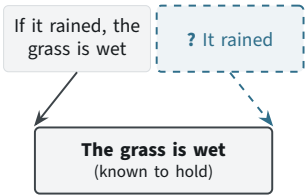
entailed

the conclusion follows if premises hold

§1: deductive benchmarks and proofs

Abduction

known fact ⇒ **missing premise**



plausible

or was it the sprinkler?

§2: NeurOWL

Deduction: entailed conclusion
Induction and abduction: plausible hypotheses

This part: what can language models do?

Deductive and abductive reasoning over ontologies

- **Deductive Reasoning** — 20 min
  Answer accuracy → proof-step grading → proof construction
- **Abductive Reasoning** — 4 min
  NeurOWL proposes a missing axiom
- **Summary** — 3 min
  Research gaps and three takeaways

1

Deductive Reasoning

Answer accuracy
Proof grading
Proof construction over ontologies

Deductive benchmarks: two distinctions

Two questions organize how we compare deductive benchmarks.

Question	One focus	Another focus
Research focus	Answer accuracy: how often is the conclusion correct?	Reasoning progress: where does the derivation succeed or fail?
What is graded?	The final answer: a label or choice.	The proof: its steps and whether they reach the goal.

Examples: OntoURL: final answer PrOntoQA: proof steps LLM4Proof: formal proof

> From answer correctness to a checkable proof

OntoURL[81]: grading the answer

OntoURL covers 15 tasks in understanding, reasoning, and learning.
Here we compare two class-membership tasks.

Toy ontology (illustrative)



U4: stated class
Which class is Nori explicitly asserted to belong to? **Penguin.**

R3: inferred class
Which further class follows from the ontology? **Bird.**

Scoring: choose a class from multiple-choice options.
Only the final answer is graded.

Qwen2.5-72B, zero-shot on OntoURL v1.1 (43 ontologies): **U4 85.6%** vs. **R3 77.2%**

R3: 8.4 points below U4 Score: final answer only

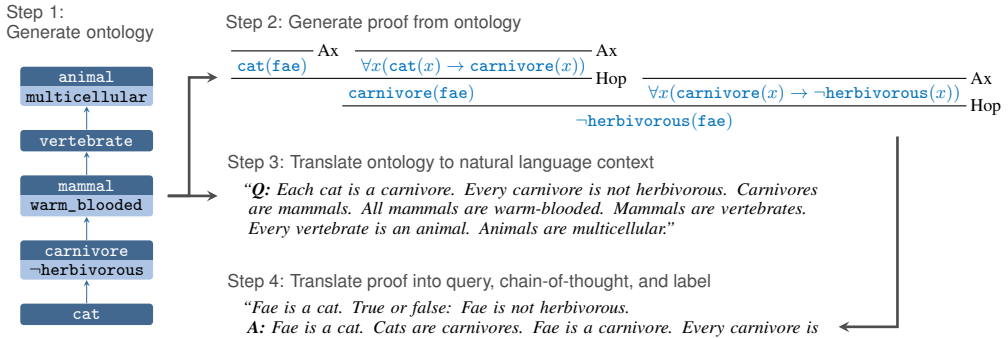
Zhang et al., J. Web Semantics 2026 [81]; OntoURL v1.1 task definitions and results

PrOntoQA[65] (1): testing LM reasoning

1. Goal: test whether LM chains of thought form valid, goal-directed proofs—beyond answer accuracy.

2. Dataset construction

Ontology → context; ontology → proof → query, CoT, label.



Saparov & He, ICLR 2023 [65], §3 and Fig. 2 (original)

PrOntoQA[65] (2): controlled factors

The dataset varies three conditions to test when the proof becomes harder.

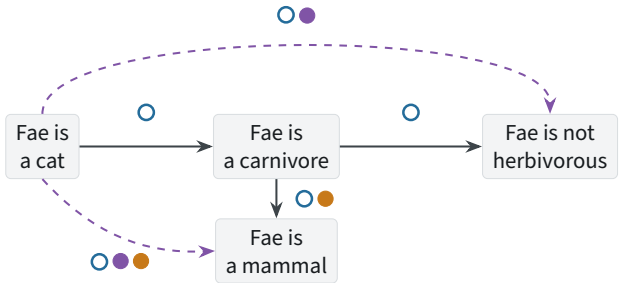
- **Proof length** 1, 3, or 5 reasoning hops
- **Context order** proof order (bottom-up) or reversed (top-down)

  Bottom-up: Cats are carnivores; carnivores are mammals; mammals are animals.

  Top-down: Mammals are animals; carnivores are mammals; cats are carnivores.
- **Prior knowledge**
  - *true*: familiar and consistent facts
  - *false*: familiar but counterfactual facts
  - *fictional*: invented names

Saparov & He, ICLR 2023 [65], §3 and §5.1; illustrative example

PrOntoQA[65] (3): grading each step of the proof



Legend

- Valid (modus ponens)
- Misleading: off path
- Skip: omitted steps
- Rule composition (broadly-valid)
- Invalid

Colors combine.

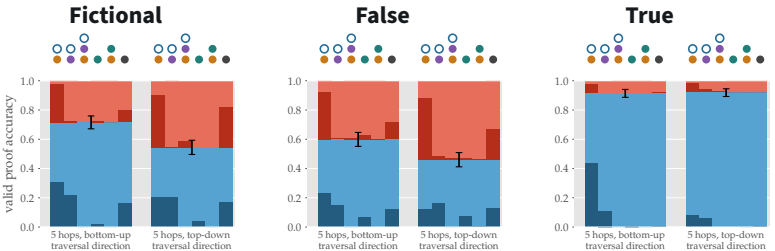
Rule composition: two rules ⇒ one new rule

Cat ⊑ Carnivore Carnivore ⊑ ¬Herbivorous
⇓ Cat ⊑ ¬Herbivorous (a rule about *all cats*)

Cases 1 2 3 4 5 6

PrOntoQA[65] (4): reading the proof analysis

InstructGPT, 8-shot CoT, 5 hops. Bar fill: blue = correct proofs; red = incorrect proofs.



Legend

- Valid (modus ponens)
- Misleading: off path
- Skip: omitted steps
- Rule composition (broadly-valid)
- Invalid

Colors combine.

Dark fill: proofs containing that step type / all examples.

Prior knowledge helps: true ontologies perform much better than false or fictional ones.
Planning remains difficult: valid local steps alone do not guarantee a successful proof.

Saparov & He, ICLR 2023 [65], Fig. 4 (original excerpts), §5.3–5.4; color markers added

Toward more complex proof

A correct answer alone does not show how the LM reached it.

What we have seen

- **Answer accuracy**: OntoURL checks the final choice
- **Short proofs**: PrOntoQA grades steps in generated linear chains

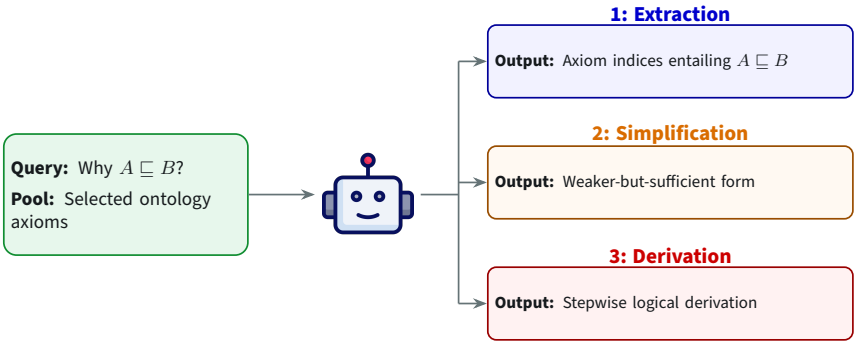
Next: LLM4Proof[60]

- Construct proofs over **real OWL ontologies**
- Study more complex, non-chain proof structure

Saparov & He, ICLR 2023 [65]; Yang et al., WWW 2026 [60]

Our focus: proof construction

Goal: Construct valid, step-by-step proofs from selected, noisy axioms in real OWL ontologies.



Yang, Chen, Sattler, WWW 2026 [60], Fig. 1

A concrete example

Premise Pool

(1) DomDog $\sqsubseteq$ Mammal
(2) Mammal $\sqsubseteq$ Animal
(3) CompAnimal $\equiv$ Animal $\sqcap$ $\exists$hasOwner.Human
(4) DomDog $\equiv$ $\exists$hasOwner.Human $\sqcap$ Dog
✗ (5) Fish $\sqsubseteq$ $\exists$livesIn.AquaticEnv
… [noise]

Prove: $\texttt{DomDog} \sqsubseteq \texttt{CompAnimal}$

Extraction

Select premises: **{1, 2, 3, 4}**

Simplification

(3′): Animal $\sqcap$ $\exists$hasOwner.Human $\sqsubseteq$ CompAnimal
(4′): DomDog $\sqsubseteq$ $\exists$hasOwner.Human

Derivation

S1: DomDog $\sqsubseteq$ Animal $\sqcap$ $\exists$hasOwner.Human
(from (1), (2), (4′))
S2: DomDog $\sqsubseteq$ CompAnimal
(from (3′), S1)

Yang, Chen, Sattler, WWW 2026 [60], Fig. 1; output format: Fig. 4

Dataset construction and evaluation

Extraction target: a justification

The minimal sufficient evidence for deriving the given conclusion.

$$J \in \mathrm{Just}(\alpha) \iff \begin{cases} J \subseteq \mathcal{O}, \quad J \models \alpha, \\ \forall J' \subsetneq J, \quad J' \not\models \alpha. \end{cases}$$

Reasoner-backed construction and evaluation

In the formal-axiom setting, we can **automatically**:

1. compute a gold justification for each selected conclusion
2. check simplified axioms and every derivation step.

- **Limited context window:** input a justification plus distractor axioms.

Yang, Chen, Sattler, WWW 2026 [60], Def. 1 and §4

Dataset construction: two difficulty measures

How hard is it to prove $\mathcal{O} \models A \sqsubseteq B$?

1. Atomic distance

$$d_{\text{sub}}(A, B) = \min\{k \mid A = A_0,\ A_k = B,\ A_0 \sqsubseteq_{\text{dir}} \cdots \sqsubseteq_{\text{dir}} A_k\}.$$

Minimum number of direct atomic subsumptions between A and B.

2. Justification length

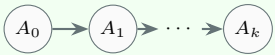
$$|J| = \#\{\text{gold axioms}\}.$$

Amount of evidence that must interact in the proof.

Difficulty: a chain versus interacting axioms

The two measures reveal different proof structures

Path-like proof



$$\mathcal{O}_1 = \{A_0 \sqsubseteq A_1, \ldots, A_{k-1} \sqsubseteq A_k\}$$

$$\mathcal{O}_1 \models A_0 \sqsubseteq A_k, \qquad d_{\mathrm{sub}}(A_0, A_k) = k.$$

$$J = \mathcal{O}_1, \qquad |J| = k.$$

Path and evidence align: close to transitive closure.

$|J| \approx d_{\mathrm{sub}} \Rightarrow$ often path-like

Non-chain proof

$$\mathcal{O}_2 = \{A \sqsubseteq \exists r.B,\ B \sqsubseteq B_1,\ \exists r.B_1 \sqsubseteq C\}$$

$$\mathcal{O}_2 \models A \sqsubseteq C, \qquad d_{\mathrm{sub}}(A, C) = 1.$$

$$J = \mathcal{O}_2, \qquad |J| = 3.$$

Concepts look close, but the proof needs extra structure.

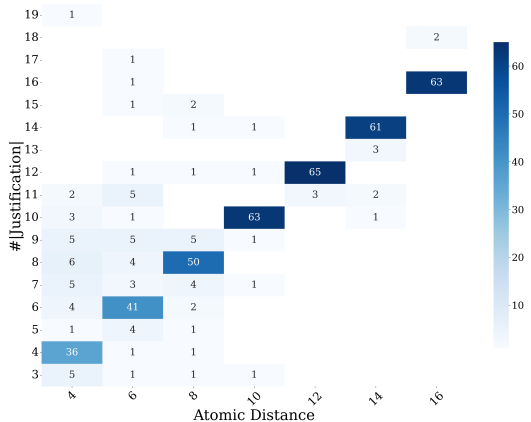
$|J| \gg d_{\mathrm{sub}} \Rightarrow$ often non-chain.

Yang, Chen, Sattler, WWW 2026 [60], §4.1 and Remark 1

Dataset construction: selected conclusions

Select conclusions across different atomic distances

- **Diagonal:** path-like proofs.
- **Upper-left:** short taxonomy path, longer proof evidence.



Selected conclusions by atomic distance and justification length.

Yang, Chen, Sattler, WWW 2026 [60], §4.3 and Fig. 2

Evaluation setup

Real ontology sources:

- Use the $\mathcal{EL}$ fragments of SNOMED CT, GO-Plus, and FoodOn.

8 LLMs Evaluated:

- **Commercial:** GPT-o4-mini.
- **32B Open:** Qwen3-32B, Qwen2.5-32B, DS-R1-Qwen-32B.
- **24B Open:** Magistral-Small-2506.
- **8B Open:** Qwen3-8B, DS-R1-Qwen-8B, DS-R1-Llama-8B.

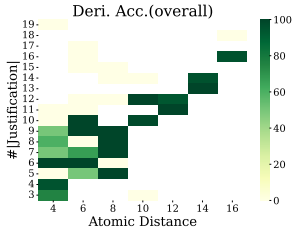
1. LLMs can prove, but struggle with complexity

GPT-o4-mini on SNOMED CT: standard vs. hard cases

Metric	Standard	Hard
Format-correct output	91.98%	51.72%
Extraction (Jaccard)	99.49%	94.39%
Simplification (whole task)	80.91%	40.14%
Valid derivation steps	92.33%	79.69%
Valid whole proofs	**80.30%**	**32.39%**

Hard: 261 further conclusions with $|J| \geq d_{\text{sub}} + 3$. Values are unweighted.

- Extraction scores the selected axioms against the gold justification.
- A whole proof succeeds only when every step is valid and the goal is reached.



Derivation accuracy by atomic distance and justification length

Yang, Chen, Sattler, WWW 2026 [60], Table 3 (App. B.1) and Fig. 7

2. Impact of Inference Rules

- **GPT-o4-mini:** Internalized logic; explicit rules can sometimes distract or hurt performance.
- **Qwen3-32B:** Benefits significantly; accuracy multiplies ($2\times$–$3\times$).

Inference Rules

```
## Inference Rules
```

1. Subsumption: If $A \sqsubseteq B$ and $B \sqsubseteq C$, then $A \sqsubseteq C$.

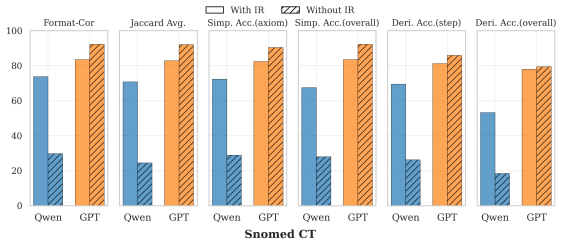
2. Conjunction: If $A \sqsubseteq B$, $A \sqsubseteq C$, then $A \sqsubseteq B \sqcap C$.

3. Existential: If $A \sqsubseteq \exists r.C$ and $C \sqsubseteq D$, then $A \sqsubseteq \exists r.D$.

4. Equivalence: If $A \equiv B$, then $A \sqsubseteq B$ and $B \sqsubseteq A$.

5. Conjunction Inference: If $A \sqsubseteq B \sqcap C$, then $A \sqsubseteq B$ and $A \sqsubseteq C$.

Here, $A, B, C, \ldots$ are concepts, r denotes relations. $\sqsubseteq$ means subclass, $\equiv$ means equivalence, $\exists r.A$ means the set of elements having a relationship r with some element in A.



Yang, Chen, Sattler, WWW 2026 [60], Fig. 5; original talk slide "Impact of Inference Rules"

3. Vulnerability to Noise

Increasing Noise Ratio (SNOMED CT):

- Evaluated ratios from 1:1 to **1:20**.

At 1:20 noise ratio (GPT-o4-mini):

- **Extraction:** drops ∼47%
- **Simplification:** drops ∼42%
- **Derivation:** drops ∼53%

Yang, Chen, Sattler, WWW 2026 [60], Table 2 and §6.2; original talk slide

4. Logical Form vs. Natural Language

🇦🇪 Does translating to English hurt?

- *Formal:* $\texttt{Teacher} \equiv \texttt{Person} \sqcap \exists \texttt{teach.Course}$
- *NL:* “Teacher is a person who teaches some courses”

Result:

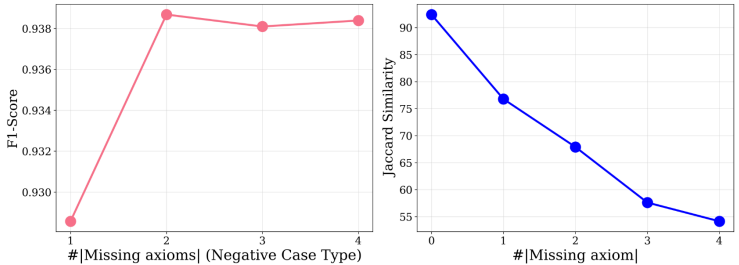
- Only $< 8\%$ performance drop.
- Logical complexity is the strict bottleneck, not format.

Yang, Chen, Sattler, WWW 2026 [60], Table 2 and §6.2; original talk slide

5. Detecting Incomplete Premises

What if some axioms are missing?

- LLMs detect the incomplete case well (F1 $\approx$ 0.93).
- Identifying the remaining useful pieces is hard.
  Jaccard similarity drops up to **38%**.



Yang, Chen, Sattler, WWW 2026 [60], Fig. 8 and §6.3; original talk slide

2

Abductive Reasoning

The missing-axiom problem
NeurOWL: proposing a plausible bridge

Abduction over ontologies: the problem

Problem setting

Input
An incomplete ontology $\mathcal{O}'$ and a query α, with $\mathcal{O}' \not\models \alpha$.

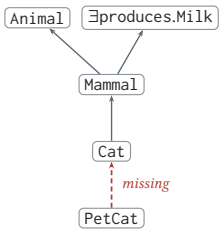
1. Judge the query
Should α hold? Predict **True** or **False**.

2. Identify missing axioms (if True)
Find H such that $\mathcal{O}' \cup H \models \alpha$.

Example: every pet cat is an animal

Query α
`PetCat` $\sqsubseteq$ `Animal` ?

incomplete ontology $\mathcal{O}'_1$:



✗ **OWL reasoner** — **No**
- α is not entailed by $\mathcal{O}'_1$

? **TBox abduction** — *predefined scope*
- finds `PetCat` $\sqsubseteq$ `Cat` only if in scope
- pure logic: no textual information
- assumes α holds: never **No**

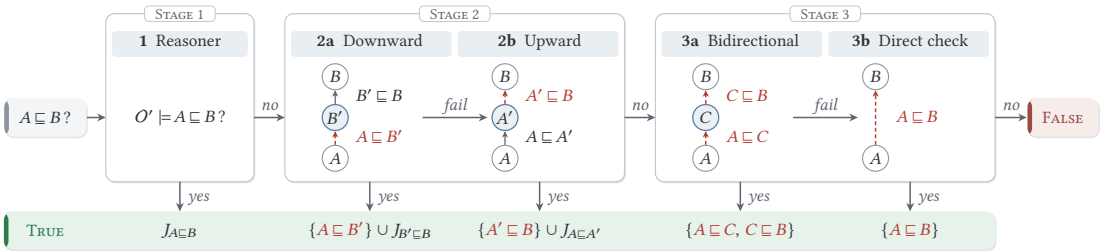
✓ **NeurOWL** — **Yes**
- completes $\mathcal{O}'_1$ with `PetCat` $\sqsubseteq$ `Cat`
- no predefined scope needed
- integrates textual information
- implausible queries: **No**

Yang et al., NeurOWL (preprint, revised version) [91], Fig. 1 and problem formulation

NeurOWL[91]: overall method

preprint

Three-stage reasoning over an incomplete ontology



1. Existing proof

The reasoner checks entailment and returns a justification.

2. One missing axiom

Test downward and upward bridges using known subsumptions.

3. Bridge or direct check

Test two bridge links, then the target directly; return False if all fail.

Yang et al., NeurOWL (preprint, revised version) [91], Fig. 2 and §4 (overall framework)

3

Summary

Three takeaways
Future Work

Summary: three takeaways

1. **LMs can prove, but struggle with complex cases**
   - GPT-o4-mini: 80% whole-proof accuracy on standard SNOMED CT cases.
   - On the hard cases, whole-proof accuracy falls to 32%.
2. **Noise limitations**
   - Distractors hurt extraction, simplification, and derivation.
   - Natural-language wording has a limited effect in these tests $\Rightarrow$ logical complexity is the main bottleneck.
3. **Textual information supports abductive reasoning**
   - NeurOWL uses textual information to identify and assess candidate axioms.

Yang et al., WWW 2026 [60], Tables 2–3 and §6.2; Yang et al., NeurOWL [91], §4

Future Work

1. **Scale LM proof generation to large, complex ontologies**
   - Design retrieval structures that supply proof-relevant ontology axioms.
   - Use retrieval and filtering to select reliable, relevant premises.
2. **Analyze proofs from informal input**
   - Analyze general reasoning processes expressed entirely in natural language.
3. **Extend abduction to more complex missing knowledge**
   - Improve performance on structurally complex missing axioms.
   - Integrate abduction with ontology repair and ontology construction.

Research directions building on LLM4Proof [60], §7, and NeurOWL [91]

References

[60] Yang, Chen, Sattler. Large Language Model for OWL Proofs (LLM4Proof). WWW 2026. github.com/HuiYang1997/LLMOwlR.

[65] Saparov & He. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought (PrOntoQA). ICLR 2023.

[81] Zhang, Lai, Meng, Bos. OntoURL: A Benchmark for Evaluating LLMs on Symbolic Ontological Understanding, Reasoning and Learning. J. Web Semantics 91, 2026.

[91] Preprint: Yang, Chen, Song, Schmidt, Zhang. NeurOWL: An LLM-Based Neural-Symbolic Framework for Incomplete OWL Ontology Reasoning. arXiv 2607.15776, 2026.

New Concept Placement in Ontologies

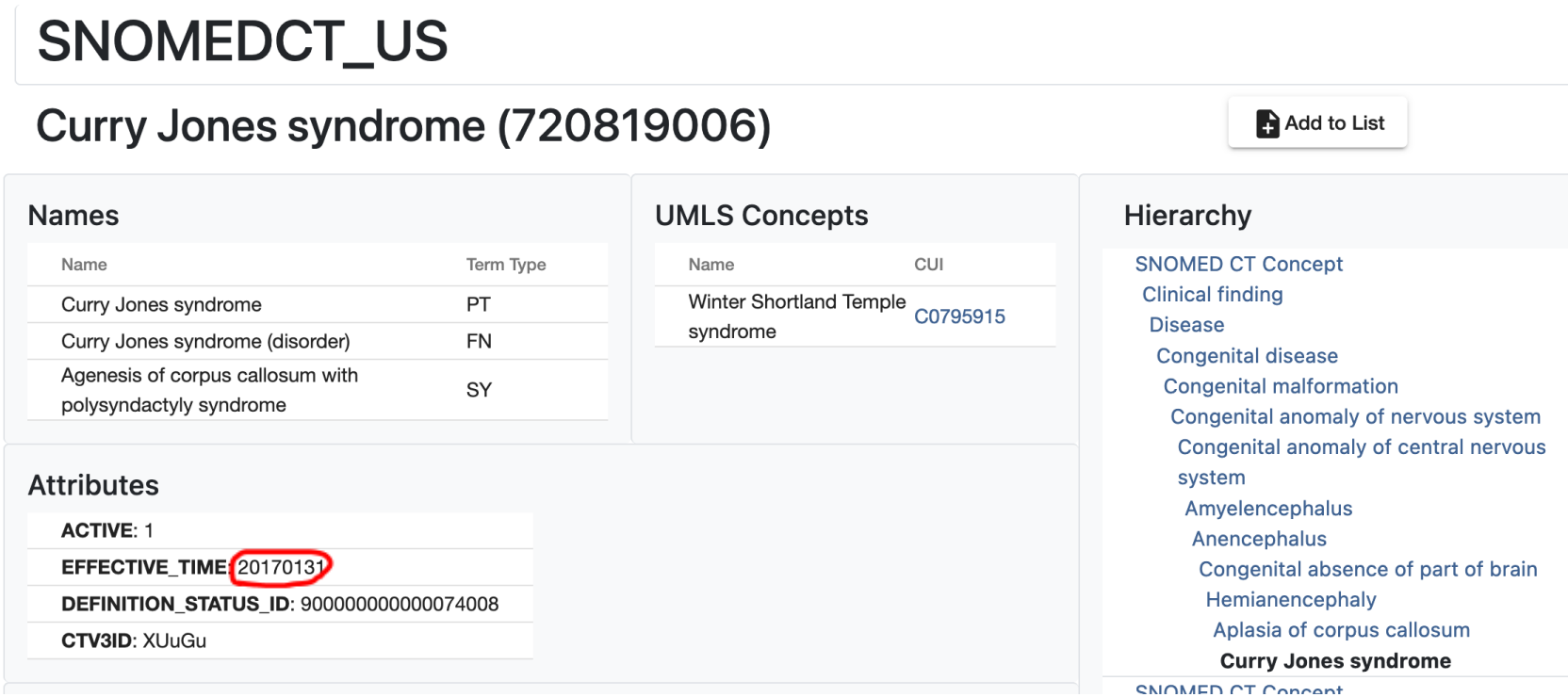
Hang Dong

Lecturer

University of Exeter

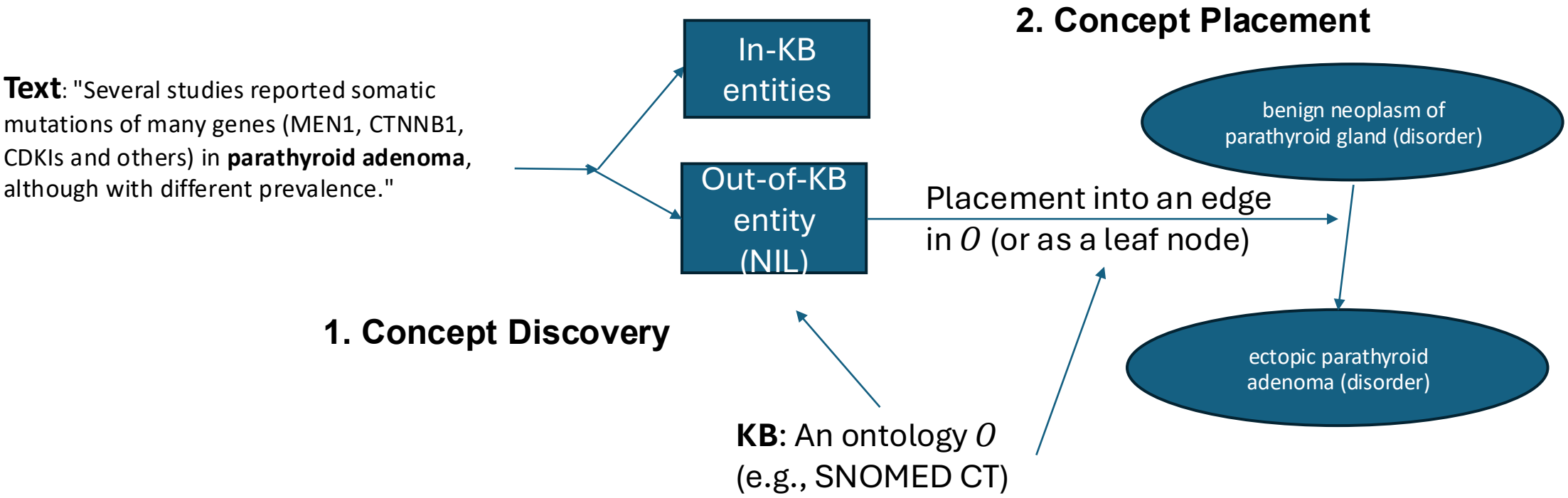
Emerging concepts to enrich Knowledge Bases

- Knowledge Bases (KB) are inherently incomplete.
- New entities emerge over time.
- **We focus on identifying new entities from texts in corpora.**
  - e.g., from scientific publications, and clinical notes.



Concept Discovery and Placement

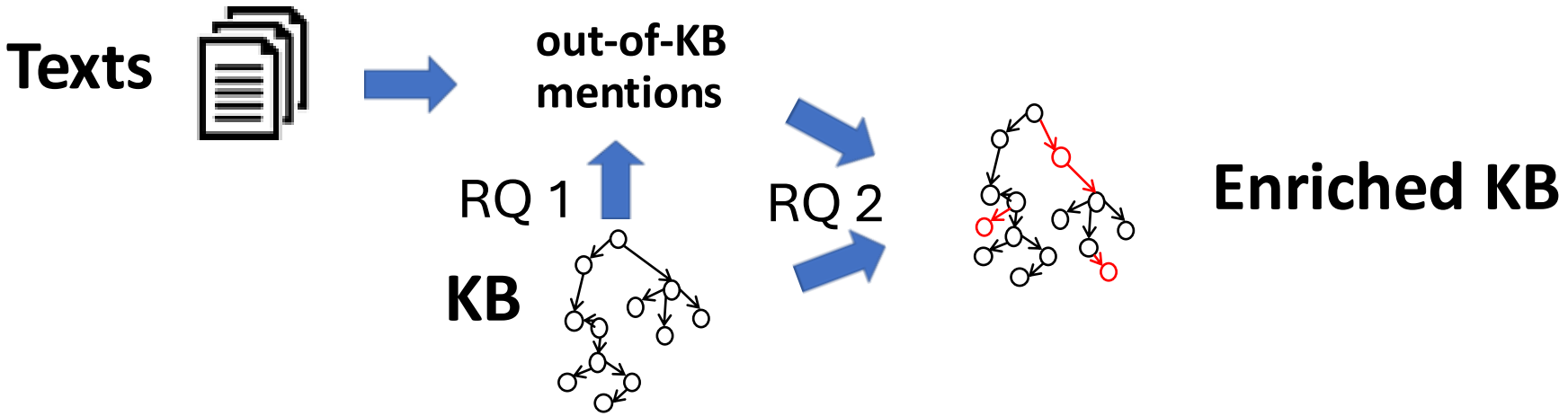
- Concept Discovery and Placement: inputs contextual in-KB and out-of-KB mentions in a corpus and a KB and outputs an enriched KB where each out-of-KB mention is inserted into a *directed edge*, i.e., < parent, child >, of the KB.
- Edge can be <parent, NULL> if it is a leaf node; parent can be a *complex concept*.



Research Questions (RQs)

1. How to identify out-of-KB mention (or NIL mention), i.e., uncaptured by a Knowledge Base, from texts?
   Dong et al., CIKM, 2023a

2. How to insert out-of-KB mentions as entities into a Knowledge Graph?
   Dong et al., CIKM, 2023b, ESWC, 2024



H. Dong, J. Chen, Y. He, Y. Liu, and I. Horrocks, 'Reveal the Unknown: Out-of-Knowledge-Base Mention Discovery with Entity Linking', in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, Birmingham, United Kingdom, Oct. 2023, pp. 452–462. doi: 10.1145/3583780.3615036.

Entity Linking for KBs

- Linking a mention (*m*) to a concept (*c*) in a KB (e.g., in UMLS)
  - "The patient has a small bowel obstruction." -> small bowel obstruction (C0235329)
  - "… male with ESRD on HD …" -> Kidney Failure, Chronic (C0022661), Hemodialysis (C0019004)

- Focus on KB, not just on documents
  - Extracting **new concepts** and merge them into an existing KB (Rao, McNamee, Dredze, 2013).
  - NIL entity problem, or *absence* problem.

Rao D, McNamee P, Dredze M. Entity Linking: Finding Extracted Entities in a Knowledge Base. In: Poibeau T, Saggion H, Piskorski J, Yangarber R, editors. Multi-source, Multilingual Information Extraction and Summarization. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 93–115. (Theory and Applications of Natural Language Processing).

Problem Formulation

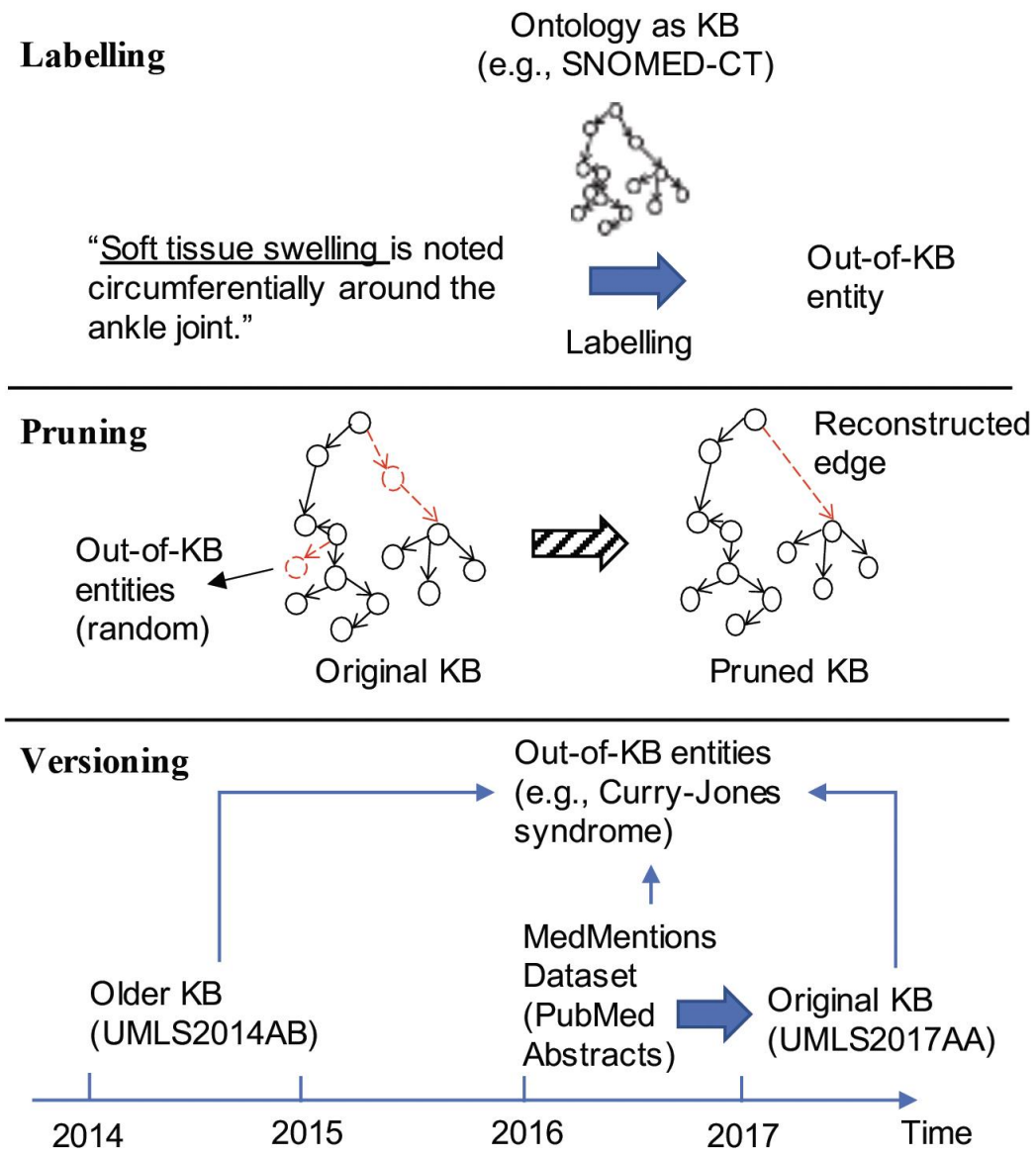
- **Out-of-KB mention discovery:** an extended entity linking task

Developing a function f such that each contextual mention (i.e., each in a context window in a document) d_m in D_M is mapped to an entity,

$$f(d_m) \in E \cup \{\text{NIL}\},$$

where NIL is a special entity denoting that there are no entities in E that can be matched with d_m.

Data Creation Strategies



Strategies for NIL-enhanced EL dataset construction: Manual Labelling, Ontology Pruning, and Ontology Versioning. (The specified UMLS ontologies are the subset linked to SNOMED CT under the semantic type of T047, "Disease or Syndrome".)

ShARe/CLEF 2013

Linked to

$\text{UMLS}_{disease\ group} \cap$ SNOMED-CT

MedMention

Linked to

$\text{UMLS 2017AA}_{T047} \cap$ SNOMED-CT

(T047: Disease or Syndrome)

NILK

Linked to

WikiData

Out-of-KB Labelled Datasets

	ShARe/CLEF 2013 train / test	MM-pruned-0.1 train / valid / test	MM-pruned-0.2 train / valid / test	MM-2014AB train / valid / test	NILK-sample train / valid / test
Out-of-KB mention creation	Manual Labelling	KB Pruning	KB Pruning	KB Versioning	KB Versioning
Domain or document type	Clinical notes	PubMed abstracts	PubMed abstracts	PubMed abstracts	Wikipedia articles
# of docs	199 / 99	2,635 / 878 / 879	2,635 / 878 / 879	2,635 / 878 / 879	81,334 / 10,587 / 10,528
# of mentions	5,816 / 5,351	6,201 / 2,121 / 2,000	6,201 / 2,121 / 2,000	6,181 / 2,112 / 1,988	86,379 / 10,688 / 10,613
# of out-of-KB mentions	1,639 / 1,723	456 / 201 / 155	1,031 / 405 / 415	307 / 82 / 103	1,312 / 154 / 158
% of out-of-KB mentions	28.2% / 32.3%	7.3% / 9.5% / 7.8%	16.6% / 19.1% / 20.8%	5.0% / 3.9% / 5.2%	1.5% / 1.4% / 1.5%
# of in-KB mentions	4,109 / 3,604	5,745 / 1,920 / 1,845	5,170 / 1,716 / 1,585	5,874 / 2,030 / 1,885	85,067 / 10,534 / 10,455
% mentions w/ zero-shot entities	- / 11.6%	- / 24.8% / 25.3%	- / 23.2% / 22.0%	- / 27.8% / 28.0%	- / 98.6% / 98.5%
# of entities in KB	88,150	35,392	31,460	35,398	79,412
# of entities & synonyms in KB	288,490	126,188	112,097	124,132	121,191
% of entities having synonyms	99.0%	99.4%	99.4%	99.3%	25.5%
Ave # syns per entity (having syns)	2.3 (2.3)	2.6 (2.6)	2.6 (2.6)	2.5 (2.5)	0.5 (2.1)

Table 1: Statistics of out-of-KB linking datasets. Slashes separate the statistics of training, validation, and testing sets. "MM" denotes MedMentions; pruned-0.1 and pruned-0.2 denote the percentage of pruned concepts in the ontology; "2014AB" refers to the older version of the UMLS applied; "NILK-sample" denotes the random sample of the NILK dataset that enriches WikiData.

Existing methods for out-of-KB mention discovery

- Threshold-based (Ji et al., 2020; Chen et al., 2021):
  - $S(d_m, E) \Longrightarrow m \leftarrow \mathrm{NIL}$, S is entity linking system; d_m, mention m in context; E, all in-KB entities; NIL, out-of-KB entity.
  - $\forall_{e \in E} s(d_m, e) < Th \Longrightarrow m \leftarrow \mathrm{NIL}$. Th is a threshold.

- Classification:
  - With features or embeddings related to out-of-KB entities

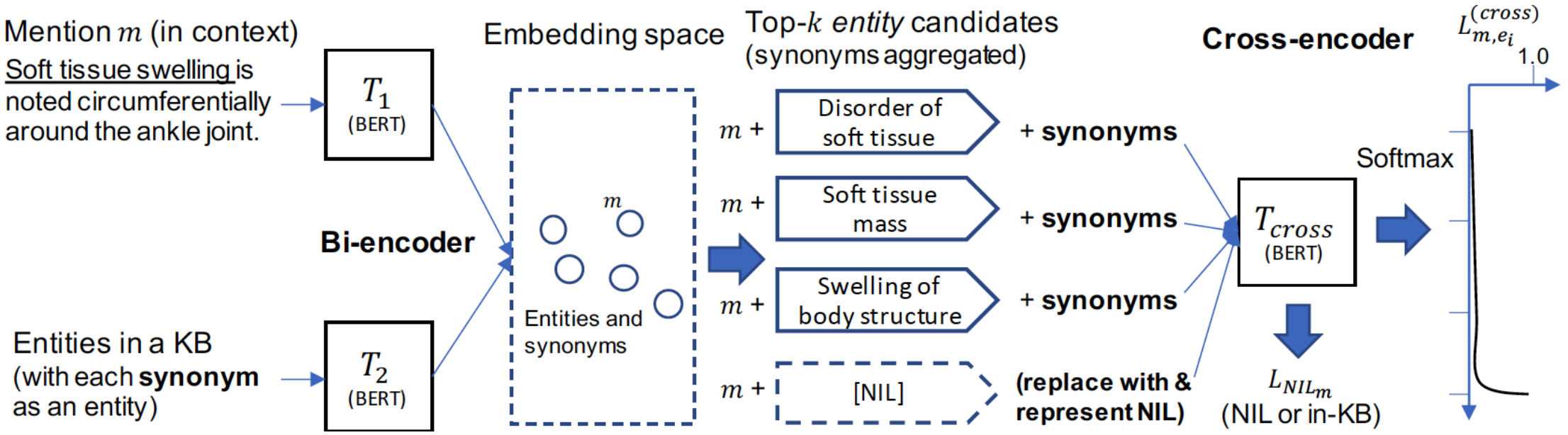
  (i) Adding *NIL* as a candidate with in-KB entities (Rao et al., 2013)

  (ii) Binary (or three-class) classification: Mention as in-KB, out-of-KB, (or not relevant) (Gu et al., 2021)

NIL entity representation

- How to represent an out-of-KB entity (to embeddings)?
  - Threshold-based – no need to represent it (Ji et al., 2020; Chen et al., 2021).
  - Classification-based – mostly do not represent NILs, but classify into NILs based on features or embeddings, using in-KB entities (Rao et al., 2013)
  - Using extra texts/corpora that contain the out-of-KB entity (Hoffart et al., 2014; Ran et al., 2022)
  - Clustering-based: need to detect out-of-KB mentions first,
    - Form clusters of mentions into entity (Kassner et al., 2022; Agarwal et al., 2021) and get embeddings of the cluster (Kassner et al., 2022).
    - Usually based on the above methods, threshold or classification.
- Fewer studies have explored NIL entity representation with neural LMs (Kassner et al., 2022; Agarwal et al., 2021).

Methods: NIL Representation & Classification, with Synonym Enhancement

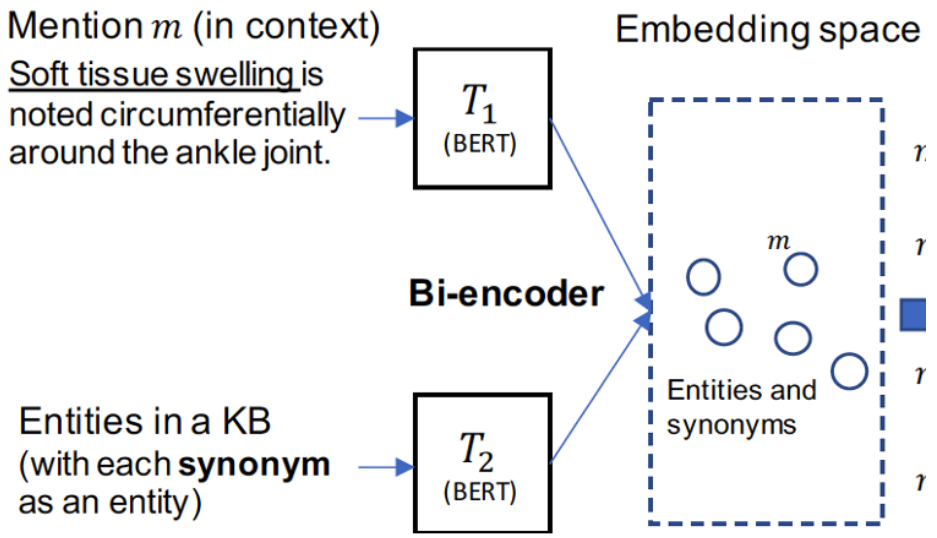


Bi-encoder (candidate generation)

Input mention d_m: **[CLS] ctxt_left [M_s] mention [M_e] ctxt_right [SEP]**
Input entity OR synonym d_e: [CLS] title [ENT] description [SEP]

- Train Bi-encoder

$$v_m = \text{red}(BERT_m(d_m))$$
$$v_e = \text{red}(BERT_e(d_e))$$
$$s(m, e) = v_m \cdot v_e$$
$$L_{m_i,e_i} = \sum_{j \neq i} [\alpha - s(m_i, e_i) + s(m_i, e_j)]_+$$



- Generate top-*k* entities for each mention
  - Aggregating over the synonyms into top-*k* entities (using the score of the most similar synonym of each entity)

Cross-encoder (candidate ranking)

```
Input mention (m):
        [CLS] ctxt_left [M_s] mention [M_e] ctxt_right

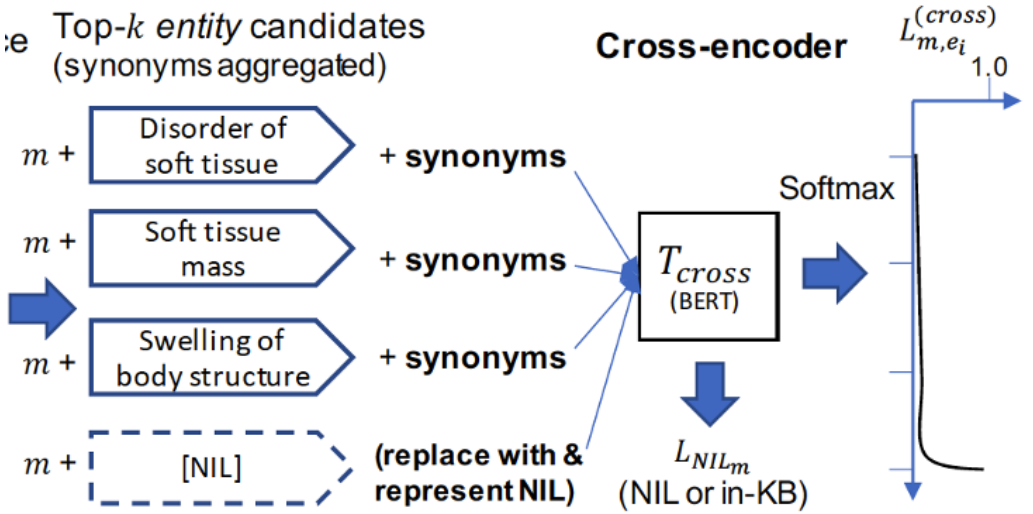
Input entity+synonym (es):
        title [ENT] syn_1 [SYN] syn_2 ... syn_n [SYN] description [SEP]

Input d_{m,es} <- Concat(Input mention, Input entity+synonym)
```

- Train Cross-encoder

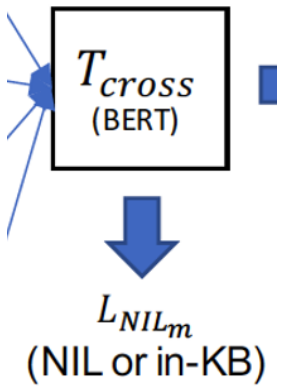
$$s_{m,e}^{(cross)} = v_{m,e}^{(cross)} w; \quad v_{m,e}^{(cross)} = \mathrm{red}(BERT_{cross}(d_{m,e}))$$

$$L_{m,e_i}^{(cross)} = -\log\left(\frac{\exp(s_{m,e_i}^{(cross)})}{\sum_{j \in E_{cand}} \exp(s_{m,e_j}^{(cross)})}\right)$$



Joint training for NIL verification

A binary classification to verify NIL, conditioned on mention representation



$$s_{isNIL_m} = \sigma(v_m^{(cross)} w_{NIL});\ v_m^{(cross)} = \mathrm{red}(BERT_{cross}(d_m))$$

$$L_{NIL_m} = -y_i \log(s_{isNIL_m}) + (1 - y_i)\log(1 - s_{isNIL_m})$$

$$L_m = L_{m,e_i}^{(cross)} + \lambda_{NIL} * L_{NIL_m}$$

Inspired by Gu, Y., Qu, X., Wang, Z., Huai, B., Yuan, N. J., & Gui, X. (2021, May). Read, retrospect, select: An MRC framework to short text entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 14, pp. 12920-12928).

Overall and out-of-KB results

Main contributors of BLINKout's performance:
(i) Fine-tuned NIL entity representation.
(ii) Synonym enhancement.
(iii) Domain-specific BERT, e.g., SapBERT.

	ShARe/CLEF 2013				MM-pruned-0.1				MM-pruned-0.2				MM-2014AB				NILK-sample			
	A	P_o	R_o	$F1_o$	A	P_o	R_o	$F1_o$	A	P_o	R_o	$F1_o$	A	P_o	R_o	$F1_o$	A	P_o	R_o	$F1_o$
Rule-based (Sieve-based)	90.0	84.7	92.6	88.5	**83.8**	36.3	**94.8**	52.5	**85.3**	63.4	**97.1**	76.7	**82.9**	26.6	**98.1**	41.9	72.1	6.6	**86.7**	12.2
Ft-based Classifier	36.0	49.1	71.6	58.2	19.1	11.2	69.7	19.2	25.8	27.5	63.9	38.4	20.1	7.2	37.9	12.1	24.2	1.6	70.3	3.0
Ft-based BLINKout $L_{NIL}^{(Ft)}$	60.9	61.4	**95.0**	74.6	42.3	14.0	92.3	24.2	48.0	39.3	95.9	55.7	43.7	8.9	85.4	16.1	45.2	1.8	61.4	3.5
Th-based BM25+BERT+syn	69.5	64.9	55.8	60.0	72.3	22.8	64.5	33.7	70.2	46.2	60.7	52.5	70.6	9.7	26.2	14.1	77.4	7.0	39.2	11.9
Th-based BM25+SapBERT+syn	76.2	71.3	64.7	67.8	75.6	28.6	84.7	42.2	74.7	51.3	72.5	60.1	71.4	16.9	81.6	28.0	75.8	7.6	70.3	13.8
BLINK (BERT)	53.0	-	-	-	67.0	-	-	-	57.2	-	-	-	62.4	-	-	-	88.0	-	-	-
+ syn	58.1	-	-	-	70.4	-	-	-	60.2	-	-	-	68.8	-	-	-	87.8	-	-	-
Th-based BLINK	72.4	65.4	67.7	66.5	68.1	21.9	42.6	28.9	71.1	51.6	77.6	62.0	65.7	21.7	74.8	33.6	88.1	17.8	49.4	26.2
+ syn	80.1	74.4	76.2	75.3	72.7	25.5	54.8	34.8	69.5	47.7	60.5	53.4	71.1	22.2	48.5	30.5	87.8	16.3	44.3	23.8
NIL-rep-based BLINK	74.8	67.6	89.2	76.9	69.7	24.8	47.1	32.5	71.9	56.9	74.9	64.7	65.7	19.1	80.6	30.9	87.9	16.3	24.7	19.6
+ syn	80.3	79.8	73.5	76.5	72.6	27.8	40.7	33.0	73.8	54.4	77.1	61.7	71.4	18.6	74.8	29.8	87.9	16.2	55.7	25.0
BLINKout (- syn, BERT)	77.7	74.1	91.2	81.8	62.3	18.4	62.6	28.5	68.8	51.1	80.7	62.6	68.2	25.2	78.6	38.1	88.1	24.3	49.4	32.6
BLINKout (BERT)	86.0	85.3	87.9	86.6	70.5	33.1	58.7	42.3	75.2	55.3	77.8	64.7	69.9	33.7	84.5	48.2	87.9	21.4	72.2	33.0
BLINKout (SapBERT)	89.9	90.0	89.8	89.9	81.8	53.5	54.8	54.1	81.4	**72.6**	81.0	76.5	81.5	57.3	79.6	66.7	86.6	22.3	69.0	33.7
+ k and NIL rep tuned*	90.8	**91.4**	90.3	**90.9**	83.3	**58.4**	65.2	**61.6**	84.8	**72.6**	88.7	**79.8**	**82.9**	**86.2**	72.8	**79.0**	**90.7**	33.0	70.3	44.9
+ L_{NIL}	**91.2**	88.5	92.5	90.5	83.1	47.0	75.5	57.9	83.0	65.2	91.6	76.2	82.5	59.3	83.5	69.4	90.4	**37.1**	67.1	**47.8**

Table 2: Comparison results for Entity Linking with *out-of-KB* mentions (Precision, Recall, F_1 scores w.r.t. out-of-KB mentions and *overall* Accuracy w.r.t. both in- and out-of-KB mentions). *The parameters k and NIL rep are tuned using BLINKout (SapBERT) for ShARe/CLEF 2013 and MM datasets and BLINKout (BERT) for NILK-sample.

Qualitative analysis

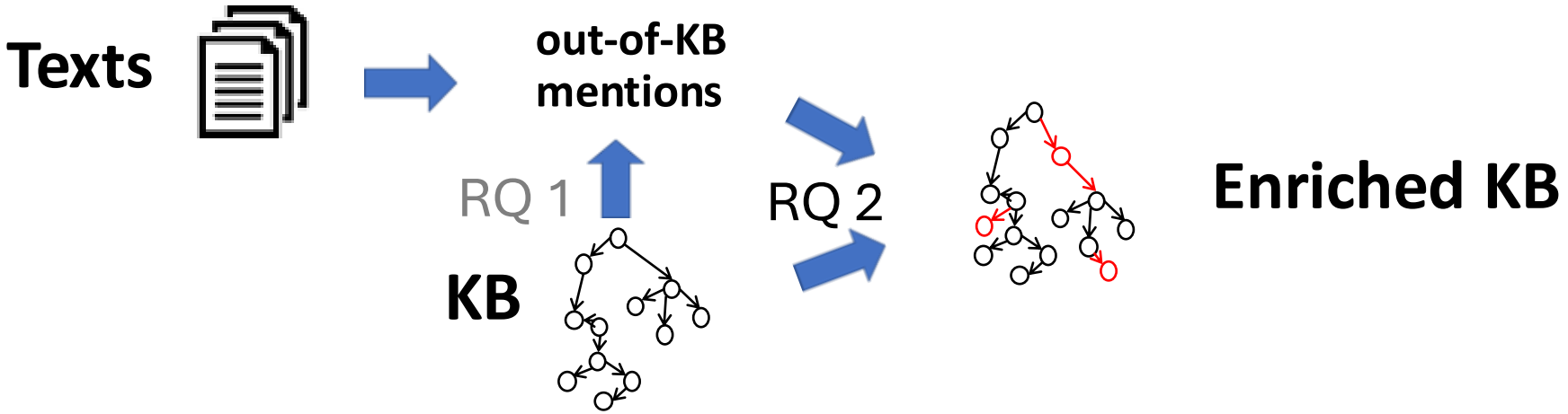
Data w/ row ID (starts from 0)	**Mention** in context	Gold	Th-based BLINK+syn	NIL-rep-based BLINK+syn	BLINKout (tuned)	BLINKout (tuned)+L_{NIL}
ShARe/ CLEF 2013 test-2353	Discharge Diagnosis: **PEA arrest** of unclear etiology	NIL	Electromechanical dissociation (C0340861) 1.000>Th_{cross} (0.95)	Electromechanical dissociation (C0340861) 1.000 (where NIL 0.000, 2nd)	Electromechanical dissociation (C0340861) 0.888 (where NIL 0.001, 2nd)	NIL 0.718
ShARe/ CLEF 2013 test-5083	Low dose beta blockade **decreased HR** to 95 in sinus and synthroid had been restarted.	Bradycardia (C0428977)	NIL (Cardiac murmur, intensity grade I/VI (C0232249) 0.191<Th_{cross} (0.95)) (Gold not in top-k)	NIL 0.973 (Gold not in top-k)	Bradycardia (C0428977) 0.989	NIL 0.947 (where Bradycardia (C0428977) 0.053, 2nd)
NILK-sample test-68	The ancient cities of Friesland are shown below: Dutch West Frisian Charter granted ... Workum Warkum 1399 **Bolsward** Boalsert 1455 Harlingen Harns 1234 Franeker Frjentsjer 1374	NIL	NIL (Ferdinand Bol (Q374039) 0.403<Th_{cross} (0.45))	Ferdinand Bol (Q374039) 0.480 (where NIL 0.276, 2nd)	NIL 0.377	NIL 0.191
NILK-sample test-2054	... **70720 Davidskillman** Named in honor of David R. Skillman (b. 1945) for his decades-long contributions to asteroid searching, stellar binary star systems and as lead systems engineer for the Hubble Space Telescope at Goddard Space Flight Center.	NIL	DAVID (Q5204342) 0.980>Th_{cross} (0.45)	DAVID (Q5204342) 0.987 (where NIL 0.019, 2nd)	NIL 0.798	NIL 0.965
NILK-sample test-755	TMS Entertainment (known for "Lupin III" and "Detective Conan") would animate the popular animated series "Tiny Toon Adventures" and "**Rainbow Brite**". ...	NIL	Rainbow Brite and the Star Stealer (Q2625526) 1.000>Th_{cross} (0.45)	Rainbow Brite and the Star Stealer (Q2625526) 1.000 (where NIL 0.000, 2nd)	Rainbow Brite and the Star Stealer (Q2625526) 1.000 (where NIL 0.000, 2nd)	Rainbow Brite and the Star Stealer (Q2625526) 0.999 (where NIL 0.001, 2nd)

Wrong predictions are marked with red. Normalised prediction score (after softmax) and the rank (if not 1st) are displayed after the predicted entity.

Research Questions

1. How to identify out-of-KB mention (or NIL mention), i.e., uncaptured by a Knowledge Base, from texts?

2. How to insert out-of-KB mentions as entities into a Knowledge Base?



H. Dong, J. Chen, Y. He, and I. Horrocks, 'Ontology Enrichment from Texts: A Biomedical Dataset for Concept Discovery and Placement', in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, Birmingham, United Kingdom, Oct. 2023, pp. 5316–5320. doi: 10.1145/3583780.3615126.

H. Dong, J. Chen, Y. He, Y. Gao, and I. Horrocks, 'A Language Model Based Framework for New Concept Placement in Ontologies', in *The Semantic Web*, ESWC 2024, Cham: Springer Nature Switzerland, 2024, pp. 79–99. doi: 10.1007/978-3-031-60626-7_5.

Datasets for ontology enrichment from texts

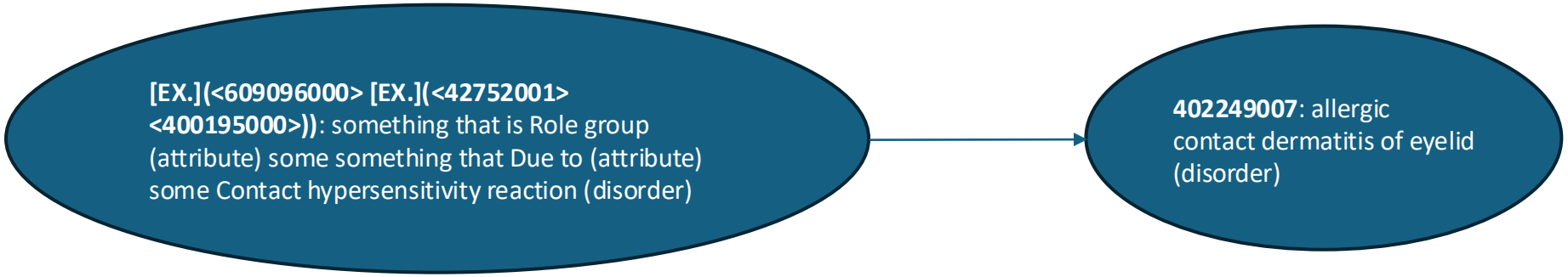
Datasets (with public access link in citations)	Task	NIL Discovery	Contextual Term	Concept Placement	Complex Concepts
MAG, WordNet [21, 30, 35]; OSConcepts, DroneTaxo, MeSH, SemEval [32, 34]	Taxonomy Completion	No	No	Yes	No
ChEBI_{500}, ChEBI^{+}_{500} [12]	Ontology Extension	No	No	Yes	No
SNOMED CT (English, manual, small-scale) [23]; SNOMED CT (Spanish, automated) [5]*	Concept Post-coordination	No	No	Yes	Yes
NILK [18]; ShARe/CLEF 2013 [31]; CLEF HIPE 2020 [11]; EDIN [22]; NEEL 2015-2016 [27, 29]	Out-of-KB Mention and Entity Discovery	Yes	Yes	No	No
MedMentions-SNOMED-CT-14 (-CPP, -Disease) [**this work**]	Concept Discovery and Placement	Yes	Yes	Yes	Yes

- **NIL Discovery**: discovering out-of-KB mentions (*cf.* in-KB mentions).
- **Contextual Term**: the input term has a context window in a text corpus.
- **Concept Placement**: the task finally places (or can be used to place) the term in the KB.
- **Complex Concepts**: the placement position in the KB includes *complex concepts*.

Complex Concept and Complex Edges

- Concepts that involve at least one logical operators, e.g., negation ($\neg$), conjunction ($\sqcap$), disjunction ($\sqcup$), existential restriction ($\exists r.C$), universal restriction ($\forall r.C$), etc.
- We use a rule-based verbaliser (in DeepOnto) to transform complex concepts into words.
- Complex *edges*: having one complex *concept* (usually as parent).

An example of the form $\exists r_1.(\exists r_2.A) \sqsupseteq B$

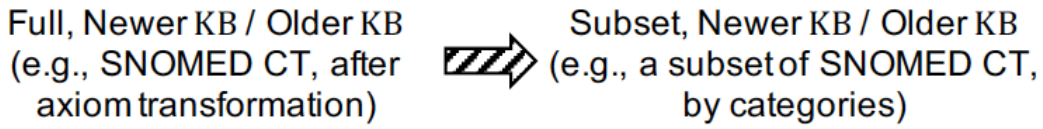


DeepOnto's Complex Concept Verbaliser: https://krr-oxford.github.io/DeepOnto/verbaliser/

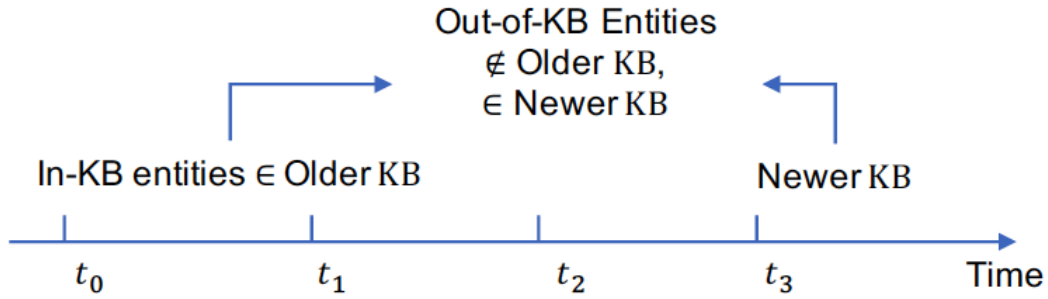
He Y, Chen J, Dong H, Horrocks I, Allocca C, Kim T, Sapkota B. DeepOnto: A Python package for ontology engineering with deep learning. Semantic Web Journal. 2023

Data Construction Pipeline

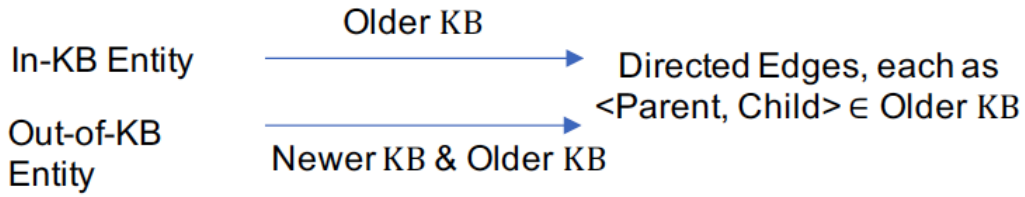
Step 0: KB and Subset Selection



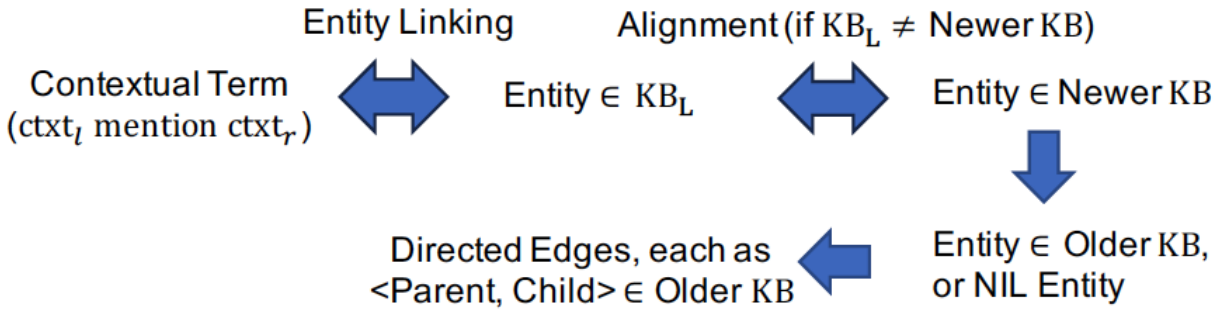
Step 1: KB Versioning to Synthesize Out-of-KB Entities



Step 2: Edge Extraction for In-KB and Out-of-KB Entities



Step 3: Mention-to-Edge Data Creation with Sources of Entity Linking and Alignment



SNOMED CT under different categories:

"**Disease**", and

"**CPP**", i.e., Clinical finding, Procedure, and Pharmaceutical / biologic product.

Older KB: ver 20140901, "S14" (included in UMLS 2014AB).

Newer KB: ver 20170301 (included in UMLS 2017AA) .

Entity Linking source: MedMentions, having around 5000 publication abstracts where mentions are linked to UMLS 2017AA.

Data Statistics

	Ontology: # all (# complex)		Corpus: # Mentions / # Mention-edge pairs / # Mention-edge pairs with *complex* edges			
	concepts	edges*	train, in-KB	valid, in-KB	test, in-KB	out-of-KB
MM-S14-Disease	64,900 (824)	237,826 (4,997)	11,812 / 887,840 / 917	4,248 / 383,457 / 203	3,970 / 316,319 / 393	605 / 1,637 / 13
MM-S14-CPP	175,895 (2,718)	625,994 (19,401)	34,704 / 1,398,111 / 9,475	11,707 / 548,295 / 4,305	11,564 / 478,424 / 4,129	1,000 / 2,131 / 22

A mention-edge pair denotes a mention (in a corpus) and one of its directed edges in the KB.

* The numbers of edges are those having one hop (including leaf nodes to NULL) and two hops from any paths in the ontology.

Data Format

- MM-S14-Disease
  - .DS_Store
  - mention-edge-pair-level
    - test-NIL-complex.jsonl
    - test-NIL.jsonl
    - test-in-KB-complex.jsonl
    - test-in-KB.jsonl
    - train-complex.jsonl
    - train.jsonl
    - valid-complex.jsonl
    - valid.jsonl
  - mention-level
    - .DS_Store
    - syn_attr
      - test.jsonl
      - train.jsonl
      - valid.jsonl
  - ontology

See more details at https://github.com/KRR-Oxford/OET/tree/main/dataset and https://zenodo.org/record/8043690

JSON keys for mention-level data:

```
"context_left"          # left context
"mention"               # mention
"context_right"         # right context
"label_concept_UMLS"    # label concept in UMLS
"label_concept"         # label concept in SNOMED CT older version
"label_concept_ori"     # label concept original, in SNOMED CT newer version if out of the older version
"label"                 # entity description
"parents_concept"       # parent concept SNOMED CT ID (or complex concept IDs)
"parents"               # parent concept title
"children_concept"      # child concept SNOMED CT ID (or complex concept IDs)
"children"              # child concept title
"label_id"              # row id in the entity catalogue ontology/xxx_syn_attr_hyp-all.jsonl file
"label_title"           # entity title
```

JSON keys for mention-edge-pair-level data:

```
"context_left"          # left context
"mention"               # mention
"context_right"         # right context
"label_concept_UMLS"    # label concept in UMLS
"label_concept"         # label concept in SNOMED CT older version
"label_concept_ori"     # label concept original, in SNOMED CT newer version if out of the older version
"entity_label_id"       # row id in the entity catalogue ontology/xxx_syn_attr_hyp-all.jsonl file
"entity_label"          # entity description
"entity_label_title"    # entity title
"parent_concept"        # parent concept SNOMED CT ID (or complex concept IDs)
"child_concept"         # child concept SNOMED CT ID (or complex concept IDs)
"parent"                # parent concept title
"child"                 # child concept title
"edge_label_id"         # row id in the edge catalogue ontology/xxx_edges.jsonl file
```

Concept placement

- Measured by micro-level (or mention-edge-pair level) precision@k , recall@k, and F1@k.
  - Other metrics: instance-level (or mention-level) metrics, e.g., percentage of mentions being corrected inserted in any of the edges (and in all the edges); ranking-based metrics, e.g., MRR.
- Methods:
  - Edge-Bi-encoder: for edge ranking
  - Edge-Bi-encoder + LLM (e.g., GPT-3.5-turbo):
    - (i) select top-k candidate edges (k as 50 in experiments) from Edge-Bi-encoder and
    - (ii) re-rank or select top-1 from top-k using LLM.
- GPT-3.5 may have limited knowledge in biomedical ontologies w.r.t. the prompt under zero-shot setting

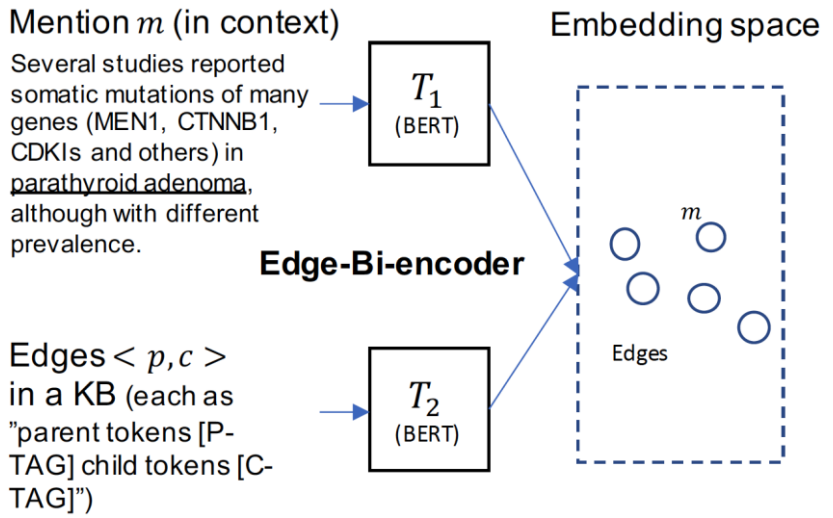
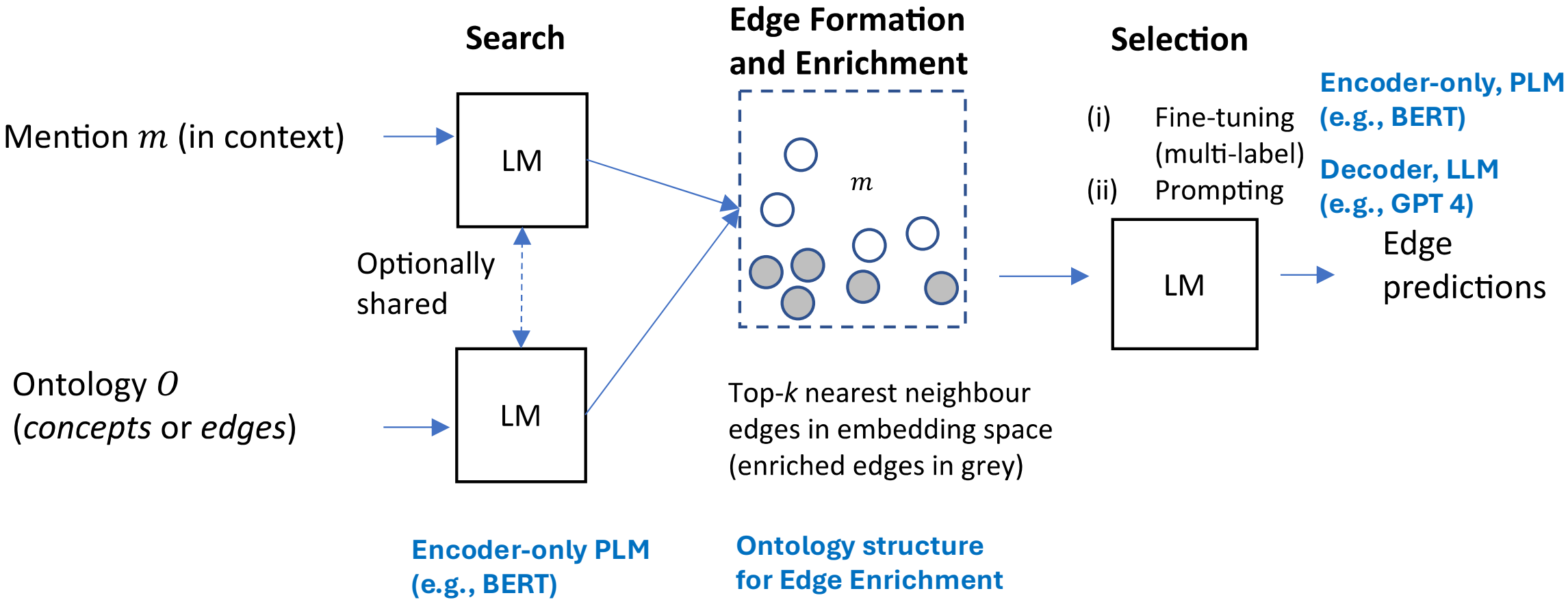


Table 4: Results on out-of-KB concept placement

MM-S14-Disease	P / R @1	P / R @5	P / R @10	P / R @50
Edge-Bi-encoder	**4.5 / 1.6**	6.0 / 11.0	5.4 / 19.9	2.1 / 38.4
+GPT-3.5	4.3 / 1.6	-	-	-
MM-S14-CPP	**P / R @1**	**P / R @5**	**P / R @10**	**P / R @50**
Edge-Bi-encoder	2.2 / 1.0	2.2 / 5.2	2.0 / 9.4	1.4 / 32.4
+GPT-3.5	**2.5 / 1.2**	-	-	-

The details of the prompts are available at https://github.com/KRR-Oxford/OET/tree/main/baseline-methods/concept-placement#prompting-gpt-35

A Language Model framework for new concept placement in ontologies



H. Dong, J. Chen, Y. He, Y. Gao, and I. Horrocks, 'A Language Model Based Framework for New Concept Placement in Ontologies',in *The Semantic Web*, A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, and P. Lisena, Eds, Cham: Springer Nature Switzerland, 2024, pp. 79–99. doi: 10.1007/978-3-031-60626-7_5.

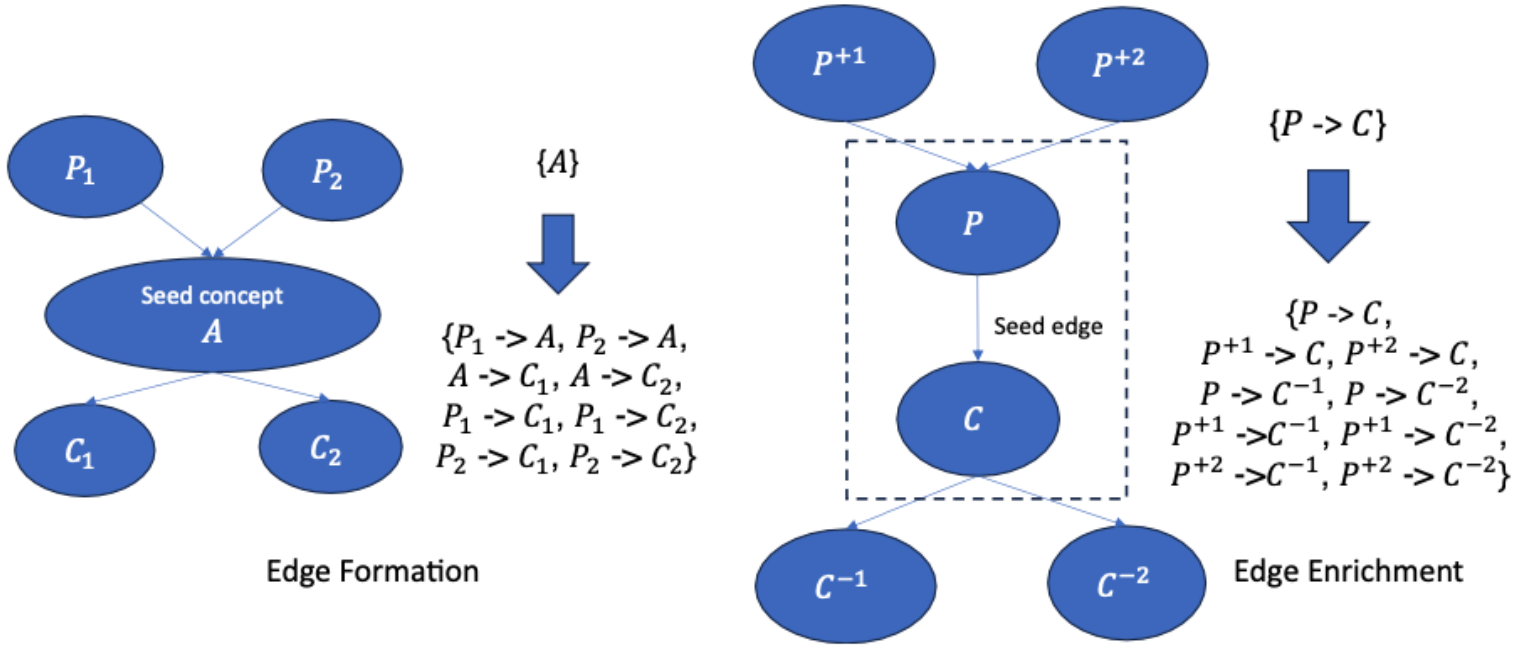
Edge Search

- Fixed BERT and its variations (e.g., SapBERT):
  - Concept search: Cosine similarity between embeddings of mention and concept label.

- Fine-tuning BERT: Edge Bi-encoder
  - Contextual Mention: [CLS] $ctxt_l$ [Ms] mention [Me] $ctxt_r$ [SEP]
  - Edge: [CLS] parent tokens [P-TAG] child tokens [C-TAG] [SEP]
    - Special token [NULL] to represent the child tokens of a leaf edge.

$$L_{m_i,e_i} = \sum_{j \neq i} [\alpha - s(m_i, e_i) + s(m_i, e_j)]_+; \quad s(m,e) = v_m \cdot v_e$$

Edge Formation and Enrichment



Edge Ranking after Edge Formation (with fixed BERT):

$$\text{Edge_score}_{fixed}(m, < P, C >) = \frac{\text{sim}(m, P) + \text{sim}(m, C)}{2}, \text{where } C \neq \texttt{NULL}$$

Edge Ranking after Edge Enrichment (with fine-tuned BERT):

$$\text{Edge_score}_{fine-tuned}(m, < P, C >) = s(m, < P, C >)$$

In Edge Formation: Prioritise leaf edge (when C is NULL) with the highest edge score if the most similar concept (A) of *m* is leaf.

Edge Selection

(i) Fine-tuning PLMs: Multi-label Edge-Cross-encoder (with PubMedBERT)

- Input as mention and edge concatenation:
  - [CLS] $ctxt_l$ [Ms] mention [Me] $ctxt_r$ [SEP] parent tokens [P-TAG] child tokens [C-TAG] [SEP]
  - Special token [NULL] to represent the child tokens of a leaf edge.

- Binary cross-entropy loss after a sigmoid activation

$$L_{m,e}^{cross} = -y_i \log(s_{m,e}^{(cross)}) + (1 - y_i)\log(1 - s_{m,e}^{(cross)})$$

$$s_{m,e}^{(cross)} = v_{cross} w$$

Edge Selection (ii) with LLMs: Prompt format and training instructions

Input:
Can you identify the correct ontological edges for the given mention (marked with *) based on the context? The ontological edge consists of a pair where the left concept represents the parent of the mention, and the right concept represents the child of the mention. If the mention is a leaf node, the right side of the edges will be NULL. If the context is not relevant to the options, make your decision solely based on the mention itself. There may be multiple correct options. Please answer briefly using option numbers, separated by commas. If none of the options is correct, please answer None.

mention in context:
There were no significant differences in age or sex distribution. Atrial fibrillation (*AF*) and previous unsuccessful ablation were more common in group I. A negative delta wave in lead II was the ECG finding with best sensitivity and specificity for the presence of a diverticulum. ...

options:
0.atrial arrhythmia -> atrial fibrillation and flutter
1.atrial arrhythmia -> lone atrial fibrillation
2.fibrillation -> lone atrial fibrillation
3.atrial arrhythmia -> NULL
4.cardiac arrhythmia -> NULL

Prompting: ### Input part + "### Response" -> ?
With explanations: ### Input part + "### Explanation" -> ?

Instruction tuning:
Input part + "### Response" -> 0,1,2
Input part + ### Explanation part + "###Response" -> 0,1,2 (with explanation)

("###" are Llama-2 model headers)

Explanation:
From the parents in the options above, including atrial arrhythmia (disorder), fibrillation (disorder), cardiac arrhythmia (disorder), the correct parents of the mention, AF, include atrial arrhythmia (disorder), fibrillation (disorder). Thus the options are narrowed down to 0, 1, 3, 2. From the children in the narrowed options, including atrial fibrillation and flutter (disorder), lone atrial fibrillation (disorder), NULL, the correct children of the mention, AF, include atrial fibrillation and flutter (disorder), lone atrial fibrillation (disorder). Thus, the final answers are 0, 1, 2.

Response:
0,1,2

Template for explanation generation

The highlighted parts are automatically generated given the candidate edges and the gold edges

$$\text{Expl_texts} = \text{Template}\,(\text{Edge}_{cand}, \text{Edge}_{gold})$$

Explanation by steps:
1. List all possible parents in the candidates
2. Find correct parents
3. Narrow the list of children
4. Find correct children
5. List the final answer

- From the parents in the options above, including xxx, xxx, xxx, the correct parents of the mention, xxx, include xxx, xxx, xxx. Thus the options are narrowed down to num1,num2,num3,...,numN. From the children in the narrowed options, including xxx, xxx, xxx, the correct children of the mention, xxx, include xxx, xxx, xxx. Thus, the final answers are num1,num2,num3,...,numN.

Data Statistics

		MM-S14-Disease	MM-S14-CPP
Ontology: # all (# complex)	concepts	64,900 (824)	175,895 (2,718)
	edges	237,826 (4,997)	625,994 (19,401)
Corpus: # M / # L / # L_{comp}	train, in-KB	11,812 / 887,840 / 917	34,704 / 1,398,111 / 9,475
	valid, in-KB	4,248 / 383,457 / 203	11,707 / 548,295 / 4,305
	valid, out-of-KB	329 / 672 / 10	568 / 979 / 13
	test, out-of-KB	276 / 965 / 3	432 / 1,152 / 9

A mention-edge pair or link (in L) denotes a mention in M (from a corpus) and one of its directed edges in the KB.

L_{comp} denotes concept edges, i.e., the edges involving a complex concept.

* The numbers of edges are those having one hop (including leaf nodes to NULL) and two hops from any paths in the ontology.

Self-supervised learning process:

Using in-KB to fine-tune or instruction-tune LMs

Using out-of-KB for validation and testing

Results

- Most **decoder-only**, generative LLMs performed worse than **encoder-only** LMs (i.e., edge-bi-encoder + edge-cross-encoder).
- **GPT-4** performs best in some top-1 & top-5 settings, but <u>not</u> in top-10.

Table 3. Overall results after edge selection for MM-S14-Disease and MM-S14-CPP datasets. Each setting has validation and testing results, separated by a slash (/) sign.

MM-S14-Disease	k	InR_{any}@1	InR_{all}@1	InR_{any}@5	InR_{all}@5	InR_{any}@10	InR_{all}@10
Inverted Index	10	0.6 / 0.0	0.0 / 0.0	1.8 / 2.5	0.9 / 0.7	10.0 / 12.0	9.1 / 10.1
	50	0.6 / 0.0	0.0 / 0.0	0.9 / 1.8	0.0 / 0.0	3.3 / 4.0	0.9 / 1.5
Fixed embs	10	4.0 / 1.4	0.9 / 0.0	6.7 / 2.2	1.5 / 0.7	16.1 / 13.0	7.0 / **12.3**
	50	4.0 / 1.4	0.9 / 0.0	6.7 / 2.2	1.5 / 0.7	13.4 / 4.4	3.0 / 2.5
Edge-Bi-enc	10	4.0 / 11.6	0.3 / 0.0	9.7 / **17.4**	2.7 / 1.4	**31.9** / 25.7	**14.6** / 8.0
	50	4.0 / 11.6	0.3 / 0.0	9.7 / **17.4**	2.7 / 1.4	13.7 / 20.3	4.3 / 2.5
+ Edge-Cross-enc	10	0.6 / 2.2	0.0 / 0.4	12.2 / 14.1	1.5 / 3.6	**31.9** / 25.7	**14.6** / 8.0
	50	7.3 / 7.6	**1.8** / **1.5**	17.9 / 15.6	**7.3** / **4.7**	25.8 / **26.5**	10.6 / 8.7
+ GPT-3.5	10	4.0 / 4.0	0.0 / 0.0	5.5 / 4.3	2.4 / 1.4	5.5 / 4.3	2.4 / 1.4
	50	3.3 / 1.5	0.0 / 0.0	4.6 / 3.6	1.5 / 0.4	4.6 / 3.6	1.5 / 0.4
+ GPT-4	10	**7.6** / 2.5	0.0 / 0.0	**21.0** / 9.1	8.8 / 4.0	21.0 / 9.1	8.8 / 4.0
	50	5.8 / 1.8	0.0 / 0.0	20.4 / 9.4	8.5 / 2.9	21.0 / 11.2	9.1 / 4.0
+ FLAN-T5-XL	10	2.7 / 1.8	0.6 / 0.0	2.7 / 1.8	0.6 / 0.0	2.7 / 1.8	0.6 / 0.0
+ FLAN-T5-XXL	10	4.6 / 5.1	1.2 / 0.7	4.6 / 5.1	1.2 / 0.7	4.6 / 5.1	1.2 / 0.7
+ Llama-2-7B	10	2.7 / 4.3	0.3 / 0.0	5.8 / 6.2	2.1 / 0.0	8.8 / 7.2	3.3 / 1.1
	50	1.8 / 3.3	0.0 / 0.0	3.7 / 5.8	1.2 / 0.7	4.0 / 6.9	1.2 / 0.7
+ Llama-2-7B-tuned	10	5.2 / **13.8**	0.0 / 0.0	7.6 / 16.3	1.5 / 1.8	7.6 / 16.3	1.5 / 1.8
	50	6.1 / 13.0	0.0 / 0.0	8.5 / 15.2	1.5 / 1.1	8.5 / 15.6	1.5 / 1.5
+ Llama-2-13B	10	3.3 / 6.9	0.0 / 0.0	7.0 / 10.1	0.3 / 1.8	8.5 / 11.2	0.6 / 2.2
	50	2.4 / 5.8	0.0 / 0.0	5.5 / 9.4	1.8 / 0.4	7.0 / 9.8	2.4 / 0.4
+ Llama-2-13B-tuned	10	7.3 / 10.1	0.0 / 0.0	7.6 / 10.5	0.6 / 0.7	7.6 / 10.5	0.6 / 0.7
	50	4.6 / 7.2	0.0 / 0.0	6.1 / 8.0	1.2 / 1.4	6.1 / 8.3	1.2 / 1.4

See resource and time usage in Appendix 1:

- All results with an NVIDIA Quadro RTX 8000 GPU card (48GB GPU).
- Instruction-tuning inference slow, e.g., Llama-2-7B to generate explanation and answer taking around 1.5 minutes (78s) per mention.

Results (cont.)

Applicability:

- Potential to suggest edge ranking for terminologists.
- A balance to achieve: Larger k vs. more manual effort

MM-S14-CPP	k	InR_{any}@1	InR_{all}@1	InR_{any}@5	InR_{all}@5	InR_{any}@10	InR_{all}@10
Inverted Index	10	0.4 / 0.0	0.0 / 0.0	0.9 / 0.5	0.0 / 0.0	5.5 / 5.8	5.1 / 5.3
	50	0.4 / 0.0	0.0 / 0.0	0.4 / 0.0	0.0 / 0.0	0.9 / 1.4	0.0 / 0.5
Fixed embs	10	2.8 / 1.2	0.7 / 0.2	6.3 / 3.7	2.3 / 2.1	11.3 / 8.3	8.3 / 7.4
	50	2.8 / 1.2	0.7 / 0.2	6.3 / 3.7	2.3 / 2.1	7.9 / 6.0	3.9 / 4.6
Edge-Bi-enc	10	2.5 / 6.3	0.0 / 0.2	6.2 / 11.8	1.2 / 1.9	**32.0** / **27.8**	**19.7** / **14.4**
	50	2.5 / 6.3	0.0 / 0.2	6.2 / 11.8	1.2 / 1.9	8.6 / 14.4	3.0 / 3.5
+ Edge-Cross-enc	10	3.4 / **9.3**	0.2 / 0.0	7.8 / 13.7	2.1 / 2.3	**32.0** / **27.8**	**19.7** / **14.4**
	50	4.9 / 3.9	**2.1** / 0.2	15.3 / **17.6**	6.3 / **6.9**	24.8 / 26.6	13.2 / 14.4
+ GPT-3.5	10	5.1 / 3.9	0.0 / 0.0	7.9 / 6.0	3.3 / 3.5	7.9 / 6.0	3.3 / 3.4
	50	1.8 / 1.9	0.0 / 0.0	3.9 / 2.8	0.9 / 0.7	4.0 / 2.8	0.9 / 0.7
+ GPT-4	10	**6.2** / 2.6	0.0 / 0.0	**16.5** / 8.4	**9.3** / 5.8	16.7 / 8.4	9.7 / 5.8
	50	3.3 / 2.1	0.0 / 0.0	10.9 / 8.6	6.0 / 5.6	12.7 / 9.0	7.2 / 6.0
+ FLAN-T5-XL	10	2.6 / 1.9	0.5 / 0.7	2.6 / 1.9	0.5 / 0.7	2.6 / 1.9	0.5 / 0.7
+ FLAN-T5-XXL	10	3.5 / 4.2	1.2 / **0.9**	3.5 / 4.2	1.2 / 0.9	3.5 / 4.2	1.2 / 0.9
+ Llama-2-7B	10	1.8 / 4.6	0.0 / 0.2	4.8 / 7.2	0.7 / 1.9	8.8 / 10.4	3.9 / 3.9
	50	1.2 / 3.5	0.0 / 0.0	2.5 / 5.1	0.7 / 0.9	3.0 / 6.3	1.1 / 1.2
+ Llama-2-7B-tuned	10	2.6 / 7.2	0.0 / 0.0	6.5 / 10.6	1.9 / 1.2	7.6 / 12.7	2.5 / 3.2
	50	2.5 / 4.6	0.0 / 0.0	3.3 / 6.7	0.5 / 0.6	4.0 / 8.1	0.9 / 1.4
+ Llama-2-13B	10	1.8 / 3.9	0.0 / 0.0	5.5 / 8.3	2.6 / 1.4	7.2 / 9.7	3.9 / 2.8
	50	2.1 / 3.5	0.0 / 0.0	2.0 / 5.6	0.7 / 0.5	4.2 / 6.0	0.7 / 0.9
+ Llama-2-13B-tuned	10	4.2 / 2.5	0.0 / 0.0	5.3 / 3.5	2.3 / 1.2	5.5 / 3.7	2.5 / 1.2
	50	5.3 / 3.2	0.0 / 0.0	6.5 / 4.2	2.3 / 0.9	6.7 / 4.2	2.3 / 0.9

A case study

Mention in context

...Since no one had **CKD** in partial nephrectomized patients, we determined risk factors for CKD in radical nephrectomized patients...

http://snomed.info/id/709044004
Chronic kidney disease

Not available in SNOMED CT 2014.09
(although was modified from "Chronic renal impairment" ID 236425005)

Edge-bi-encoder candidates (top-10) – with correct ones in bold (incomplete edges)

options:
0.renal impairment → end stage renal disease
1.renal impairment → renal failure following molar and/or ectopic pregnancy
2.renal impairment → renal failure syndrome
3.renal impairment → chronic kidney disease due to hypertension
4.kidney disease → impaired renal function disorder
5.kidney disease → renal function impairment with growth failure
6.kidney disease → renal failure syndrome
7.renal impairment → chronic renal failure syndrome
8.disorder of the genitourinary system → impaired renal function disorder
9.renal impairment → hypertensive heart and renal disease with renal failure

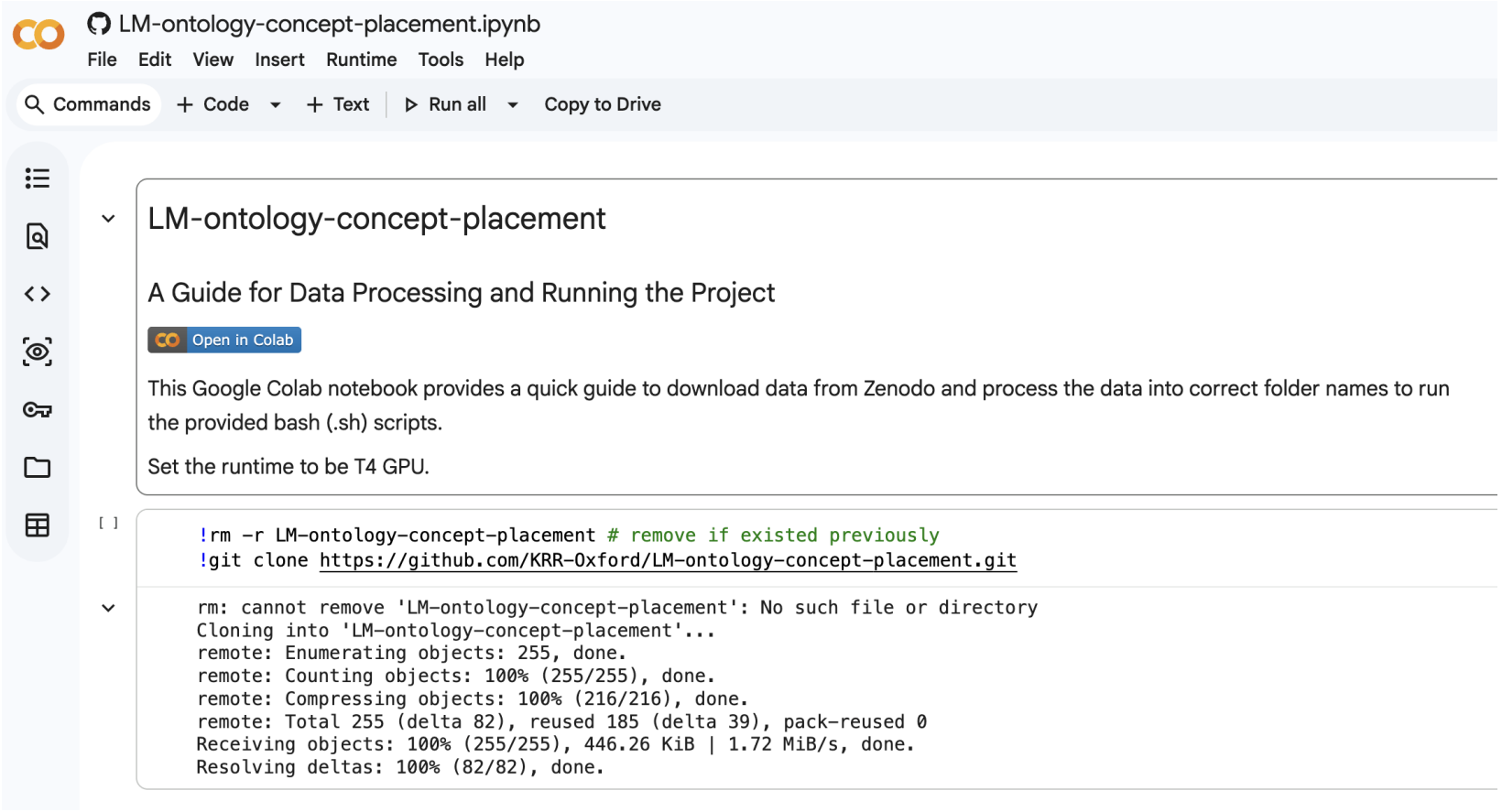
Explainable instruction-tuned Llama-2-7B

Explanation:
From the parents in the options above, including renal impairment , kidney disease , disorder of the genitourinary system , the correct parents of the mention, CKD, include **renal impairment** , kidney disease , disorder of the genitourinary system . Thus the options are narrowed down to 0, 1, 2, 3, 7, 9, 4, 5, 6, 8. From the children in the narrowed options, including end stage renal disease , renal failure following molar and/or ectopic pregnancy , renal failure syndrome , chronic kidney disease due to hypertension , chronic renal failure syndrome , impaired renal function disorder , renal function impairment with growth failure , the correct children of the mention, CKD, include **chronic kidney disease due to hypertension** , **chronic renal failure syndrome** , impaired renal function disorder , renal function impairment with growth failure . Thus, the final answers are **3, 7**, 5, 6, 8.

Response: **3,7**,5,6,8

OET data Jupyter Notebook

- https://github.com/KRR-Oxford/LM-ontology-concept-placement/blob/main/LM-ontology-concept-placement.ipynb -> “Open in Colab”



Taxonomy completion

J. Shi, H. Dong, J. Chen, Z. Wu, and I. Horrocks, 'Taxonomy Completion via Implicit Concept Insertion', in *Proceedings of the ACM Web Conference 2024*, in WWW '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 2159–2169. doi: 10.1145/3589334.3645584.



Figure 1: Implicit concept examples ("Clothing" and "Shoes")

Code and data available at https://github.com/jingcshi/ICON

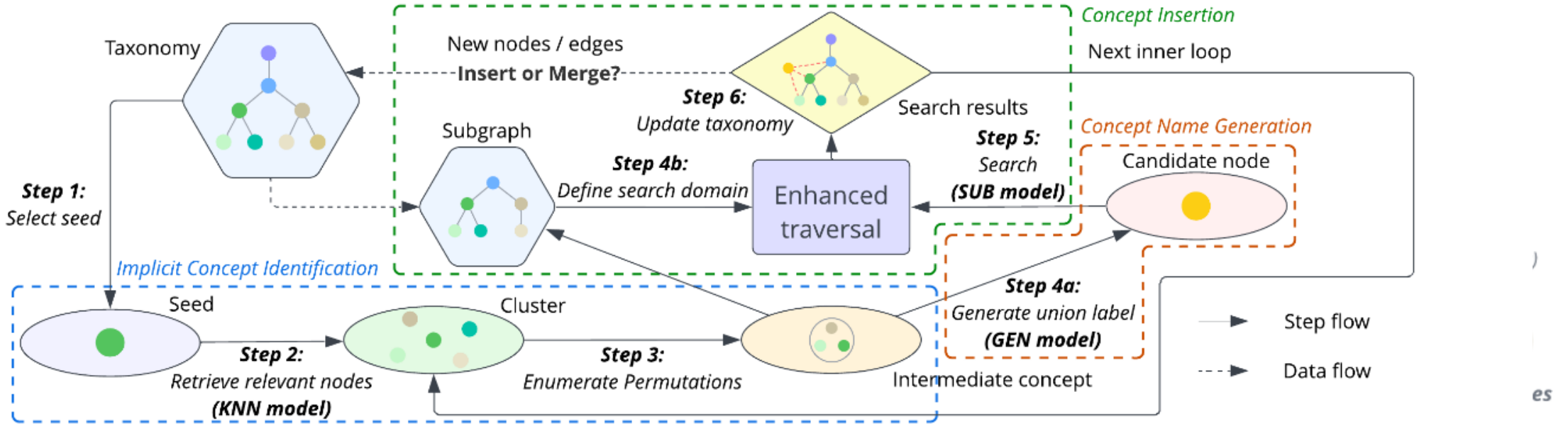


Figure 3: Workflow of ICON. In steps 1 - 2, it retrieves a cluster of nodes for a given seed. In steps 3 - 6, it traverses some subsets of the cluster, generates a labelled concept for each subset and inserts it into the taxonomy. After looping steps 3 - 6 over all the intended subsets, it goes back to step 1 and selects a new seed. Steps 1 - 2 form the outer loop, and steps 3 - 6 form the inner loop.

Open access links

RQ1 – identify out-of-KB mentions from texts

- BLINKout, GitHub: https://github.com/KRR-Oxford/BLINKout, Dataset: https://zenodo.org/record/8228371

RQ2 – insert out-of-KB mentions as entities into an ontology

- OET, GitHub: https://github.com/KRR-Oxford/OET, Dataset: https://zenodo.org/record/8043690
- LM-ontology-concept-placement, GitHub: https://github.com/KRR-Oxford/LM-ontology-concept-placement, Dataset: https://zenodo.org/records/10432003

Taxonomy completion, implicit - ICON, GitHub: https://github.com/jingcshi/ICON

This work was supported by EPSRC projects, including ConCur (EP/V050869/1), OASIS (EP/S032347/1), UK FIRES (EP/S019111/1); Samsung Research UK (SRUK); eBay Inc..

Conclusion

- We formulated the problem of new concept placement as two sub-questions (identification & insertion) and provided datasets with our solutions.

- Future studies:
  - **Multimodality**:
    - Can cross-modal information, e.g. images, genes, tables, help enrich new concepts into a Knowledge Graph?

  - **Mechanisms of explanations in LLM**:
    - Can KG evaluate & enhance LLMs' generated explanation?
    - Mechanistic explanation for ontology enrichment

  - **Human+AI collaboration**: working with terminologists and clinical coders (assigning concepts to clinical texts).

- Acknowledgement
  - Other key contributors:
    - Yuan He (Oxford, now in Amazon; DeepOnto, BERTMap, Bio-ML, HiT)
    - Mathias Jackermeier (Oxford; Box^{2}EL),
    - Jingchuan Shi (Oxford, now in eBay; ICON)
    - Yiping Song (Manchester; GenOM)
    - Ian Horrocks (Oxford)
  - Collaborators
    - SNOMED International, Sansumg Research UK, eBay
  - Main funders (projects):
    - EPSRC (OntoEm & ConCur), Samsung Research UK (Personal KG), eBay
- **Feel free to contact us**
  - hui.yang-2@manchester.ac.uk
  - jiaoyan.chen@manchester.ac.uk
  - h.dong2@exeter.ac.uk