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

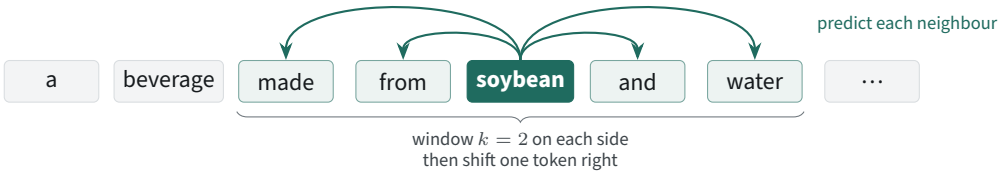
	1. Corpus-based methods	**2. Pre-trained Transformers**
Route	learn token vectors	adapt a text encoder
Core idea	ontology → context → Word2Vec[34]	text → contextual encoder → task training
Representation	fixed vector per word / entity token	context-dependent representation
Methods	RDF2Vec[39], Onto2Vec[40], 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.

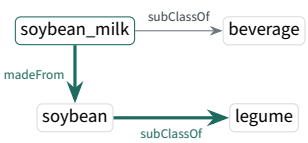
For ontologies: a corpus of class IDs, relations and words

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

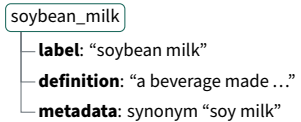
2. Axioms as sentences




```
soybean_milk SubClassOf madeFrom
some soybean
```

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

3. Annotations as text



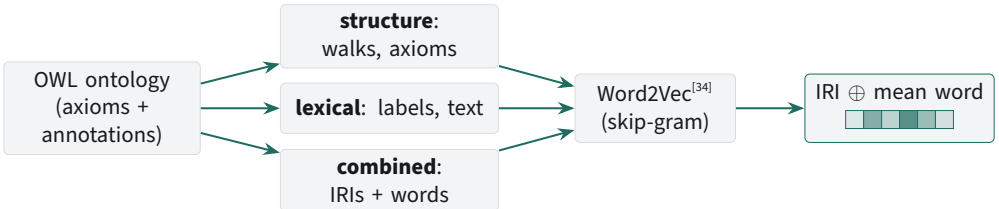

```
soybean_milk label soybean milk
```

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

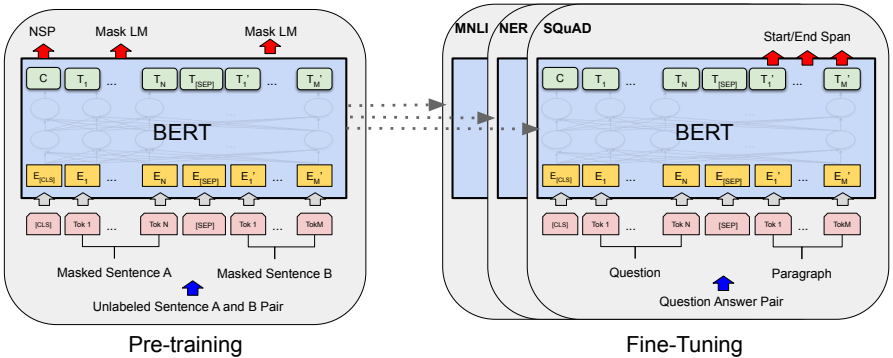
Chen et al., Machine Learning 2021 [42], §3; redrawn

2

Pre-trained Transformer methods

BERT + pair classification
Sentence-BERT + structured embeddings

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

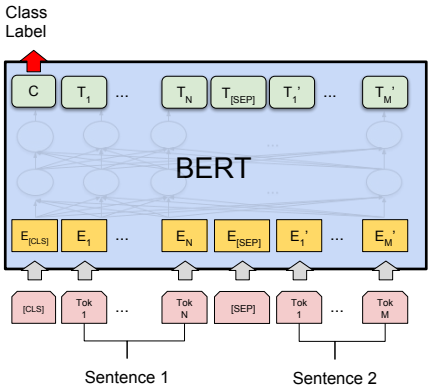


- **Pre-training**: masked tokens (MLM) + next-sentence prediction (NSP)
- **Fine-tuning**: update encoder + task head on labelled task data

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]`

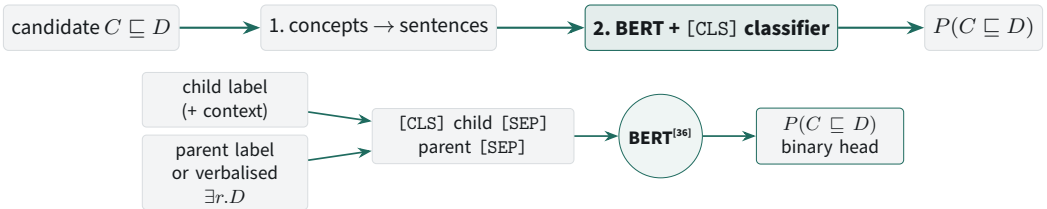
> One candidate subsumption $\rightarrow$ one sentence pair for BERT

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

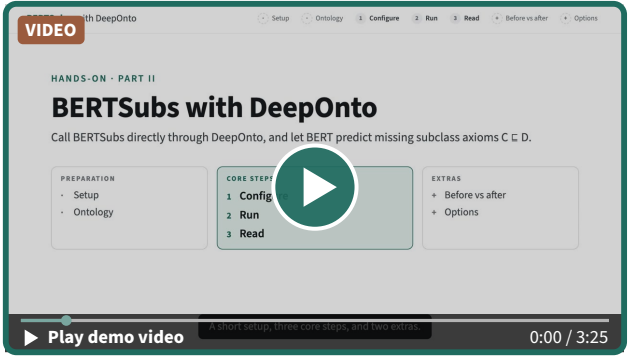
Method	FoodOn				GO			
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10
TransE	0.479	0.332	0.654	0.816	0.320	0.192	0.444	0.605
Text-aware TransE	0.572	0.429	0.734	0.869	0.518	0.357	0.718	0.863
Word2Vec	0.562	0.426	0.717	0.866	0.416	0.284	0.549	0.721
Onto2Vec	0.591	0.451	0.762	0.875	0.428	0.291	0.570	0.751
OPA2Vec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

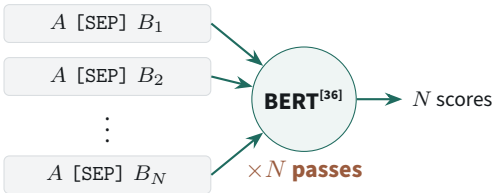
aunt, niece (no parents stated):
$P(\sqsubseteq \text{female}) = 1.00$, $P(\sqsubseteq \text{male}) = 0.00$

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

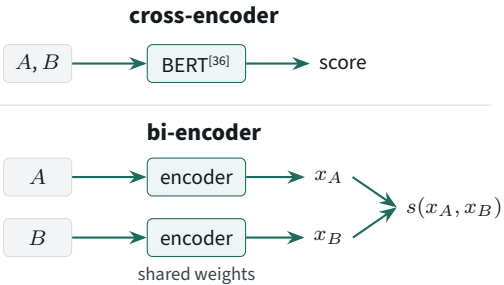


What this costs

- 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]

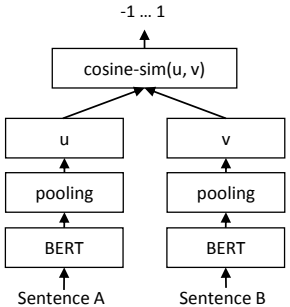
2.1

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.

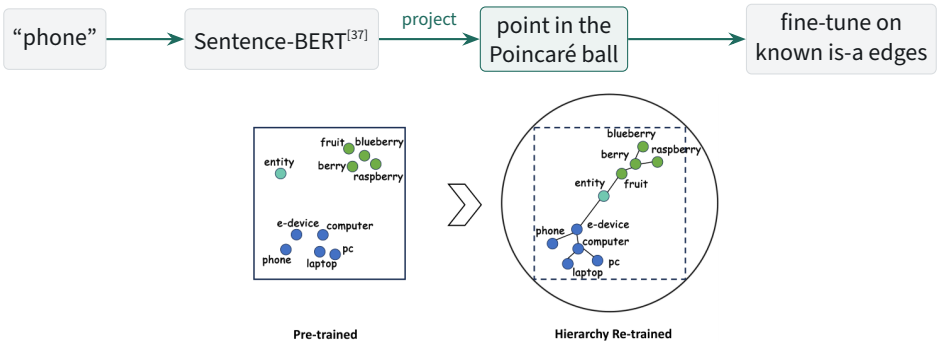
HiT / OnT: adapt these sentence vectors from similarity to hierarchy

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

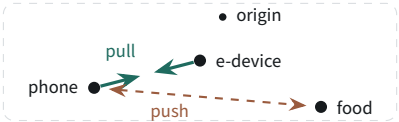
The encoder learns the hierarchy: general near the origin, specific near the edge

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

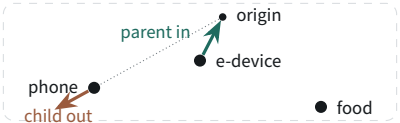
HiT[47] fine-tuning loss

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

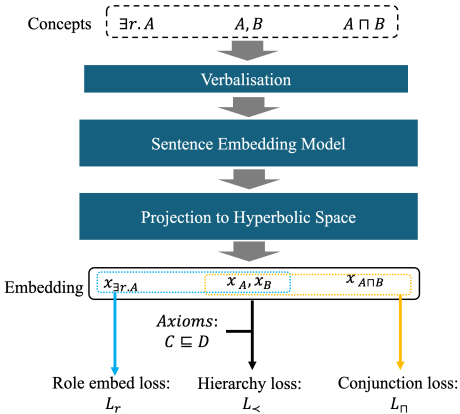
Yang et al., ISWC 2025 [48], §1; illustrative axioms

OnT[48] framework overview

Main steps

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

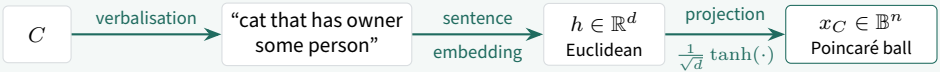
Hierarchy loss $\mathcal{L}_{\prec}$: from HiT
Role loss $\mathcal{L}_r$, conjunction loss $\mathcal{L}_{\sqcap}$: new



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

Main component 1: concept embedding

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



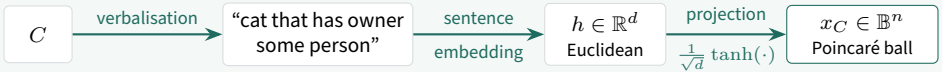
For any concept C and role $r \Rightarrow$ embeddings x_C and x_r

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:

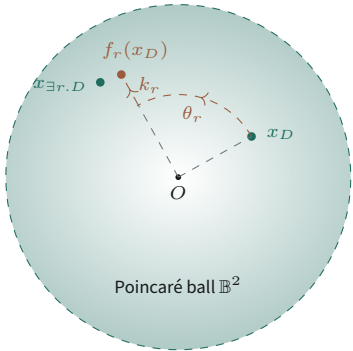
- f_r: 2-D rotations by θ_r, scaling factor k_r

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

- 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 $\exists r.D$

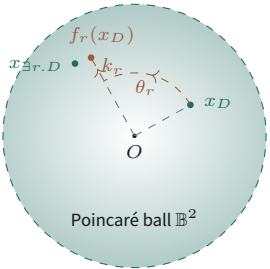
Building block: HiT's loss $\mathcal{L}_{\prec}(x_C \prec x_D)$ per axiom $C \sqsubseteq D$ (clustering + centripetal)

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**

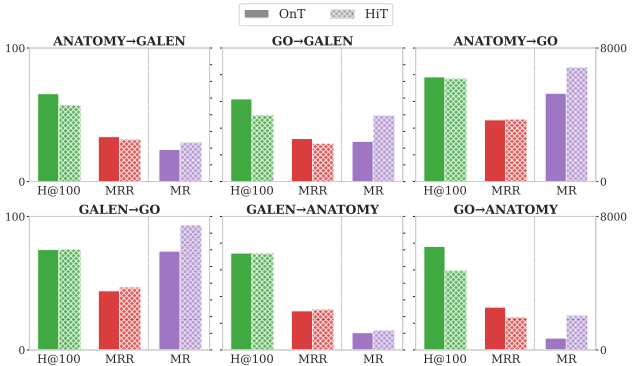
H@k: $k = 1/10/100$ (%)
MRR (%)
OnT (w/o r): without role embeddings as functions f_r

- 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)

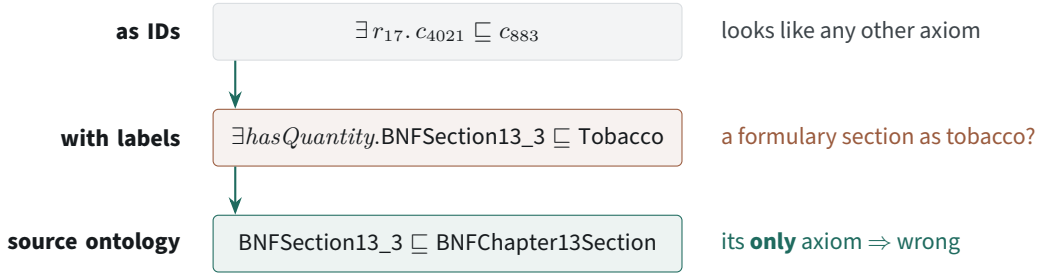


Better transfer: higher H@100 and lower MR than HiT in all six directions (MRR: mixed)

[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

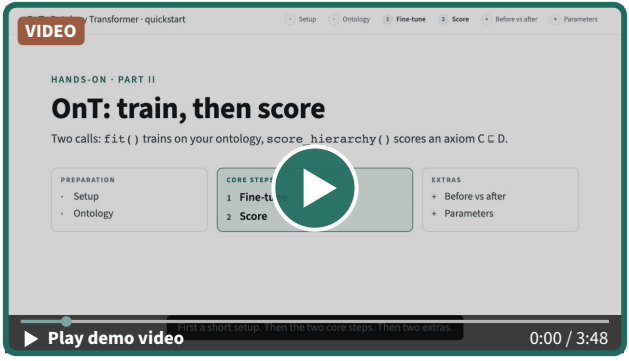


> Text helps spot errors that are invisible in IDs alone

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

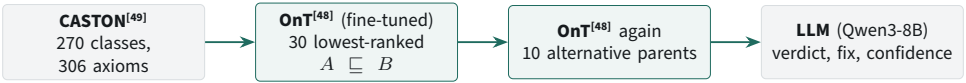
Method	Output	Complex concepts	Role embedding	New labels	One vector per concept
OWL2Vec*[42]	vector	×	×	partly[1]	✓
BERTSubs[43]	score per pair	✓[2]	×	✓	×
HiT[47]	point in $\mathbb{B}^d$	✓[2]	×	✓	✓
OnT[48]	point in $\mathbb{B}^d$	✓	✓ (f_r)	✓	✓

[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.