

A Language Model based Framework for New Concept Placement in Ontologies

Hang Dong^{1,2}[0000-0001-6828-6891], Jiaoyan Chen^{3,1}[0000-0003-4643-6750],
Yuan He¹[0000-0002-4486-1262], Yongsheng Gao⁴[0000-0002-3468-2930], and
Ian Horrocks¹[0000-0002-2685-7462]

¹ University of Oxford, Oxford, UK
{yuan.he, ian.horrocks}@cs.ox.ac.uk

² University of Exeter, Exeter, UK
h.dong2@exeter.ac.uk

³ University of Manchester, Manchester, UK
jiaoyan.chen@manchester.ac.uk

⁴ SNOMED International, London, UK
yga@snomed.org

Abstract. We investigate the task of inserting new concepts extracted from texts into an ontology using language models. We explore an approach with three steps: *edge search* which is to find a set of candidate locations to insert (i.e., subsumptions between concepts), *edge formation and enrichment* which leverages the ontological structure to produce and enhance the edge candidates, and *edge selection* which eventually locates the edge to be placed into. In all steps, we propose to leverage neural methods, where we apply embedding-based methods and contrastive learning with Pre-trained Language Models (PLMs) such as BERT for edge search, and adapt a BERT fine-tuning-based multi-label Edge-Cross-encoder, and Large Language Models (LLMs) such as GPT series, FLAN-T5, and Llama 2, for edge selection. We evaluate the methods on recent datasets created using the SNOMED CT ontology and the MedMentions entity linking benchmark. The best settings in our framework use fine-tuned PLM for search and a multi-label Cross-encoder for selection. Zero-shot prompting of LLMs is still not adequate for the task, and we propose explainable instruction tuning of LLMs for improved performance. Our study shows the advantages of PLMs and highlights the encouraging performance of LLMs that motivates future studies.

Keywords: Ontology Enrichment · Concept Placement · Pre-trained Language Models · Large Language Models · SNOMED CT

1 Introduction

New concepts appear as they are discovered in the real world, for example, new diseases, species, events, etc. Ontologies are inherently incomplete and require evolution by enriching with new concepts. A main source for concepts is corpora, e.g., new publications that contain mentions of concepts not in an ontology.

In this work, we focus on the problem of placing a new concept into an ontology by inserting it into an edge which corresponds to a subsumption relationship between two atomic concepts, or between one atomic concept and one complex concept constructed with logical operators like existential restriction ($\exists r.C$). Distinct from previous work in taxonomy completion (e.g., [30, 33, 34]), the task allows natural language contexts together with the mention as an input and also considers logically complex concepts by Web Ontology Language (OWL). Distinct from previous work in new entity discovery (e.g., [8]), the task places the new entity into the ontology, a step further to their discovery from the texts. The task is more challenging than entity linking from a mention to a concept, considering that there are many more edges than already the large number of concepts and axioms in an ontology (of a form much more complex than a tree), even by limiting the edges to only those having one-hop or two-hop.

Recently, machine learning, neural network based methods, and especially pre-trained language models (PLM), have been applied to ontology engineering tasks. For new entity discovery tasks, typically, the entity linking or retrieval tasks comprise two steps, the first is to search relevant entities by narrowing down the candidates, and the second is to select the correct one. Previous studies on entity linking and new entity discovery mostly use BERT-based fine-tuning methods [31, 8]. We differ Large Language Models (LLMs) from PLMs by their vast difference in scale and language generation capabilities. There is a recent growth of studies using LLMs, e.g., for entity linking [29] and ontology matching [14], but the experimental results are yet to be confirmed and their advantages and drawbacks for concept placement are not clear. A more detailed investigation is needed to compare the methods for the representation, and a framework is needed for their comparison. In the texts below, we use LMs as a general term for both PLMs and LLMs and use more specific terms where necessary.

For concept placement, we propose a framework that extends the two-step process, with another edge enrichment step. After the edge search to narrow the edge candidates to a limited number, we enrich the edges by walking in the ontological graph by extending the parents and children to another layer. Then this enriched set of edges is re-ranked through the edge selection part (which can be modelled as a multi-label classification task). Using this framework, we are able to compare different data representation methods, including traditional inverted index, fixed embedding based similarity, contrastive learning based PLM fine-tuning, and instruction-tuning and prompting of LLMs.

The evaluation is based on the recent datasets in [7], created by using an ontology versioning strategy (i.e., comparing two versions of an evolving ontology) to synthesise new concepts and their gold edges to be placed w.r.t. the older version of the ontology. The ontology is SNOMED CT, under *Disease* and *CPP* (Clinical Finding, Procedure, and Pharmaceutical / biologic products) branches.

Results indicate that edge enrichment by leveraging the structure of ontology greatly improves the performance of new concept placement. Also, among the data representation methods, contrastive learning based PLM fine-tuning generally performed the best in all settings. The inadequate yet encouraging results

of LLMs under our experimental setting may be related to the input length restriction and the inherent knowledge deficiencies of LLMs for nuanced concept relations of domain specific ontologies. Instruction-tuning, especially with automated explainable prompts, improves over the zero-shot prompting (i.e., no further instruction-tuning) of LLMs. Our results suggest the potential of LLMs and motivate future studies to leverage them for ontology concept placement.⁵

2 Related Work

2.1 Ontology Concept Placement

Ontology concept placement is a key task in ontology engineering and evolution. It aims to automatically place or insert a new concept, in its natural language form and potentially with contexts in a corpora, to an existing ontology. This automated task helps to reduce the immense initial human effort to discover and insert new concepts, as humans may not be able to review all available new information at the rate when they are available, and the manual process while of high quality, is of high cost and low efficiency [11, 3].

The recent study in [7] summarised the related available datasets on ontology concept placement. Datasets for the relevant tasks include taxonomy completion, ontology extension, post-coordination, and new mention and entity discovery. The proposed new datasets in [7] supports a more comprehensive set of characteristics, including NIL entity discovery, contextual mentions, concept placement (under both atomic and complex concepts in ontologies). We extend the datasets in the work [7] and use them for benchmarking in this paper.

Another relevant task is entity linking, which links a textual mention to its concept in a Knowledge Base (KB) or an ontology [24]. Entity linking can be extended to the case for out-of-KB mentions [8]. Ontology Concept Placement is distinct from entity linking to a concept, which alternatively links an out-of-KB mention to an edge (of subsumption relations) in the structure of an ontology.

2.2 Pre-trained Language Models for Ontology Concept Placement

We consider pre-trained language model as a neural, Transformer model [27] that can be pre-trained using corpora using masked modelling or by predicting future tokens, processing very large amounts of text [18]. A Large Language Model is a scaled PLM to a vastly higher degree which can result in improved performance and emergent capabilities [35].

A relevant line of work to ontology concept placement is Knowledge Graph Construction, where BERT is evaluated and shows promise to enhance several relations in WikiData [28]. Other studies focus on formal KBs which are usually expressed as OWL Ontology, e.g., by predicting the subsumption relations [4].

⁵ Our implementation of the methods and experiments are available at <https://github.com/KRR-Oxford/LM-ontology-concept-placement>.

The work [17] predicts a wider range of inter-ontology relations (e.g., equivalence, subsumption, meronymy, etc.) using PLMs (e.g., DistillBERT, RoBERTa, etc.).

For ontology concept placement, PLMs have also been applied. The study [21] aims to place concepts to SNOMED CT by pre-training and fine-tuning BERT for subsumption prediction. The study [23] uses a similar BERT-based Bi-encoder architecture and experiments with more medical ontologies. However, both works always place a concept as a leaf node, instead of higher levels.

Another approach utilising LLMs is through a prompting-based approach. The idea is to formulate an ontology-related task using natural language input that leverages the generative capability of a language model. While the recent study [29] explored prompting-based approaches of LLMs for concept equivalence linking, few studies have explored them for ontology concept placement.

In this paper, we propose an LM-based framework that leverages embedding, fine-tuning, prompting, and instruction-tuning of PLMs and LLMs for ontology concept placement. The task also considers contexts in a mention and the logically complex concepts in ontologies that are not considered in previous work.

3 Problem Statement

We use the definition of an OWL ontology, a Description Logic KB that contains a set of axioms [12, 2]. We focus on the TBox (terminology) part of an ontology, containing General Concept Inclusion axioms, each as $A \sqsubseteq B$, where A (and B) are atomic or complex concepts [1]. *Complex concepts* mean concepts that involve at least one logical operator, e.g., negation (\neg), conjunction (\sqcap), disjunction (\sqcup), existential restriction ($\exists r.C$), universal restriction ($\forall r.C$), etc. [1].

An ontology \mathcal{O} can be more simply defined as a set of concepts D (possibly complex) and directed edges E . A directed edge contains a direct parent and a direct child, where the parent or child can be complex concepts.⁶

Formally, the task is to place a new concept mention m (with surrounding contexts in a corpus) into edges in an ontology \mathcal{O} so that $C \sqsubseteq m \sqsubseteq P$ for an edge $\langle P, C \rangle$ (or as $P \rightarrow C$) that contains a parent concept P and a child concept C . The child concept C can be NULL when the mention is to be placed as a leaf node. Using SNOMED CT (version 1703) as an example, a mention “Psoriatic arthritis” (in a scientific paper) is to be placed as `Psoriatic arthritis with distal interphalangeal joint involvement` \sqsubseteq `Psoriatic arthritis` \sqsubseteq `Psoriasis with arthropathy`; and a mention “Neurocognitive Impairment” is to be placed as a leaf concept (so C is NULL), and the axioms include `Neurocognitive Impairment` \sqsubseteq `Cognitive disorder` and `Neurocognitive Impairment` \sqsubseteq `\exists RoleGroup.(\exists DueTo.Disease)`⁷.

Ontology concept placement can thus be considered matching from a textual mention (possibly surrounded by a context window) to edges in the structure of

⁶ We focus on the common case that only the parent can be a complex concept, as in the explicit axioms in the SNOMED CT ontology.

⁷ This means that Neurocognitive Impairment belongs to the role group [25] or a grouping of the characteristic that is caused by (“due to”) a disease.

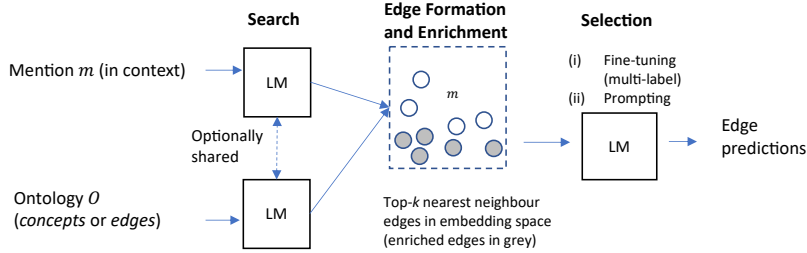


Fig. 1. An overall three-step framework for ontology concept placement with LMs.

an ontology. Given that a concept may have more than one parent and more than one child, it can be placed into many edges. Thus ontology concept placement can be formulated as a *multi-label learning* problem [10]. The task is to learn a mapping function f that can map the input (a textual mention possibly with contexts) to a set of labels (here as edges in E). Typically, a multi-label learning process can create a label *ranking* based on a metric score that orders the whole set of labels [10] or an ordered set without an explicit metric (e.g., by the order of text generation). This is distinct from the entity linking task which usually maps the input to only a single label (as an entity or a NIL entity) [8].

4 Methodology

Extending the general ideas in information retrieval and entity linking, we propose a three-step framework for ontology concept placement, as shown in Figure 1 below. Usually, retrieving a set of correct items (e.g., edges) needs two steps, *search* (or candidate generation) and *selection* (or candidate ranking). The search step aims to find a set of seed concepts (to form edges) or a set of seed edges directly. The selection step finds (and also ranks, as in multi-label classification) the correct edges among the candidates. Considering the structural nature of the edge generation process, we add another step in between, *edge formation and enrichment*, which forms seed edges from a seed concept (optionally) and enriches seed edges to derive the full candidate edges. We employ LMs in both the search and the selection steps, and further leverage the ontological structure for the edge formation and enrichment step.

4.1 Edge Search: Searching Seed Concepts or Edges

The search step inputs a textual mention m (with a context window) and an ontology O , both represented using LMs. For concept search, we encode a mention and the label of a concept using an LM with fixed parameters (or as two same LMs sharing parameters). For edge search, we encode a mention and an edge using two LMs with fine-tuning to align them into the same embedding space, given the distinct types of texts (in corpora and in ontologies) between them.

Concept Search with Fixed Embeddings We search concepts by using the nearest neighbours of LM-based embeddings, i.e., ranking using the cosine similarity of the mention embedding and every concept embedding in the ontology. A domain-specific ontology-pre-trained BERT, SapBERT [20], is used to represent both a mention and a concept. Complex concepts, with logical operators, can be verbalised using a rule-based verbaliser (e.g., in [16]), before their embedding.

Edge Search with Fine-tuning Edge-Bi-encoder We use two LMs to encode the mention and the edge separately, using the representation of the [CLS] token in the last layer, adapting the Bi-encoder architecture [31, 8]. A mention is represented as [CLS] `ctxtl` [M_s] `mention` [M_e] `ctxtr` [SEP], where `ctxtl` and `ctxtr` are the left and right contexts of the mention in the document, resp., and [M_s], [M_e] are the special tokens placed before and after the mention. In the setting without contexts, we set both `ctxtl` and `ctxtr` as empty strings. A directed edge (having a direct parent and a direct child) is represented as “[CLS] `parent tokens` [P-TAG] `child tokens` [C-TAG] [SEP]”. We use a special token [NULL] to represent the `child tokens` of a leaf concept in the ontology.

The training follows a contrastive loss, more specifically, a max-margin triplet loss [22] described below, where α is a margin of small value (e.g., 0.2) and $[x]_+$ denotes $\max(x, 0)$, for each mention to its gold edge (the i -th) in a batch, $s(m, e)$ is the mention-edge similarity, calculated as the dot-product of the mention embedding and the concept embedding. The idea is to make each mention close to one of its edges in the embedding space, but far away from the other edges within the same batch. We use in-KB data for training and validation to form a model and then finally validate and test on out-of-KB data.

$$L_{m_i, e_i} = \sum_{j \neq i} [\alpha - s(m_i, e_i) + s(m_i, e_j)]_+; \quad s(m, e) = v_m \cdot v_e \quad (1)$$

4.2 Edge Formation and Enrichment

The idea of edge formation and enrichment is to leverage the ontological structure together with the LM-based embedding for candidate retrieval. The detailed process with examples is presented in Figure 2.

Edge Formation from Seed Concepts When concept candidates are selected from entities, for each concept A , we traverse the ontology by one hop to find the parents P_1, \dots, P_n and children C_1, \dots, C_n of the concept, and then using the set $S = \bigcup_i \{P_i \rightarrow A\} \cup \bigcup_j \{A \rightarrow C_j\} \cup \bigcup_i \bigcup_j \{P_i \rightarrow C_j\}$ as the candidate edge set, which includes all one-hop edges containing A and all two-hop edges which traverse through A (see an example in the left part of Figure 2). We further added leaf edges, $A \rightarrow \text{NULL}$, to S .

Edge Ranking after Edge Formation Then the edge set is ranked using the LM-based embedding w.r.t. the mention m , as the average cosine similarity of m to the parent and m to the child in the embedding space (see Equation 2

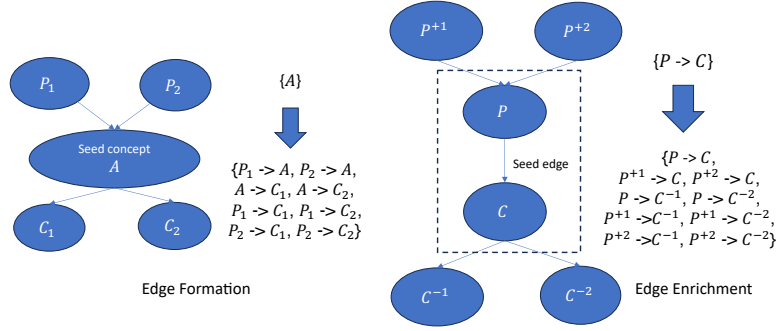


Fig. 2. An example of the edge formation and enrichment process using ontology structure. Edge formation transforms a seed concept into a set of edges, while edge enrichment augments the set of edges one by one. For methods that directly search edges (e.g., Edge-Bi-encoder), no edge formation is needed and only enrichment is applied.

below). For the edge score of leaf edges (where $C = \text{NULL}$), we first set a rule to deduce whether the mention is to be placed on a leaf edge by checking if the top ranked seed concept is a leaf concept, if so, we prioritise all enriched leaf edges of the mention with the highest edge score (i.e., better ranked than non-leaf edges).

$$\text{Edge_score}_{fixed}(m, \langle P, C \rangle) = \frac{\text{sim}(m, P) + \text{sim}(m, C)}{2}, \text{ where } C \neq \text{NULL} \quad (2)$$

Edge Enrichment from Seed Edges We further enrich the edges by traversing one-hop upper for parents and one-hop lower for children in the ontology. For each edge $P \rightarrow C$, we thus first find their one-hop upper parents P^{+1}, \dots, P^{+i} and one-hop lower children C^{-1}, \dots, C^{-j} , and enrich the set to $\{P \rightarrow C\} \cup \bigcup_i \{P^{+i} \rightarrow C\} \cup \bigcup_j \{P \rightarrow C^{-j}\} \cup \bigcup_i \bigcup_j \{P^{+i} \rightarrow C^{-j}\}$. We combine the enriched edges from all seed edges and then remove the duplicated edges (given that some of the enriched edges can be the same for different but similar seed edges). This covers more related edges based on the ontological structure and LM-based similarity and can greatly improve the recall of the edge retrieval. We also enrich a “leaf” edge, i.e., $P \rightarrow \text{NULL}$, when a parent P in a non-leaf edge is predicted.

Edge Ranking after Edge Enrichment The enriched edges are then ranked with scores from different edge search methods. For the fixed embedding approach, edges are ranked based on the edge score in Equation 2. For the fine-tuned embedding (Edge-Bi-encoder) approach, edges are ranked using the dot product scores, $s(m, e)$ (see Equation 3 below and the right part of Equation 1) for all edges (including both leaf and non-leaf edges) after the enrichment for the fine-tuned embedding (Edge-Bi-encoder) approach. The top- k candidate edges are then retrieved from the seed edges after this process.

$$\text{Edge_score}_{fine-tuned}(m, \langle P, C \rangle) = s(m, \langle P, C \rangle) \quad (3)$$

4.3 Edge Selection

The edge selection step aims to find the correct edges to place the concept mention from the k candidate edges. We utilise LMs based on their distinct architectures, i.e., we fine-tune BERT-like, encoder-only PLMs for multi-label classification, and prompts and instruction-tunes LLMs, which have a decoder, for result generation.

Fine-tuning PLMs: Multi-label Edge-Cross-encoder We adapt an LM-based cross-encoder in [8, 31], that encodes the interaction between sub-tokens in the contextual mention and an edge in the top- k edges, for multi-label classification. Specifically, for each of the k candidate edges the input is a concatenation of the contextual mention with the edge, i.e., [CLS] $ctxt_l$ [M_s] mention [M_e] $ctxt_r$ [SEP] parent tokens [P-TAG] child tokens [C-TAG] [SEP], and the output is a multi-label classification over all the inputs, i.e., the selection of the correct edges from the candidates. We use a special token [NULL] to represent the child tokens of a leaf edge. Each input is encoded with a BERT model into a vector v_{cross} (we use the representation of [CLS] in the last layer).

Therefore, the loss is a binary cross-entropy loss after a sigmoid activation of the score, linearly transformed from the representation vector, $S_{m,e}^{(cross)} = v_{cross}w$, of each input. All the inputs share the same BERT model for fine-tuning.

Zero-shot Prompting LLMs Alternatively, a recent paradigm is to prompt LLMs to generate answers directly. We formulate a prompt to allow LLMs to generate the indices of the options. The prompt provides contexts and all necessary information including the top- k candidate edge options to allow the LLMs to be conditioned and generate the answer. The prompt is structured as below, which contains an input (including task description, the mention in context, and the options of k edges) and a response headline. The sequence which is underlined (after ### Response) is expected to be generated by the LLM.

```
### 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:
Our aim was to verify the occurrence of selected mutations of the EZH2 and ZFX genes
in an Italian cohort of 23 sporadic *parathyroid carcinomas*, 12 atypical and 45 typical
adenomas.
options:
0.primary malignant neoplasm → parathyroid carcinoma
1.malignant neoplastic disease → malignant tumor of parathyroid gland
2.malignant neoplastic disease → primary malignant neoplasm of parathyroid gland
...
8.primary malignant neoplasm of parathyroid gland → NULL
...
(till all the  $k$  candidates are listed)
### Response:
2,8
```


Explainable Instruction-tuning LLMs We can observe that it would not be straightforward for an LLM to directly figure out the edges and output the option numbers (e.g., 2,8 in the example above). Fine-tuning with in-KB training data would be needed. To bridge the reasoning gap between the input and the response, we propose to add an explanation section that describes the reasoning steps in a narrative form.

Thus, we automatically synthesise explanations by steps to solve the new concept placement problem: (i) List all possible parents in the candidates; (ii) Find correct parents; (iii) Narrow the list of children based on the correct parents; (iv) Find correct children; (v) List the final answer based on the correct children. The explanation `Expl_texts` is a function of the k candidate edges and the gold edges of a mention, i.e., $\text{Expl_texts} = \text{Template}(E_{cand}, E_{gold})$. The template is below, where elements in the lists (in square brackets) are separated by comma.

```
### Explanation:
From the parents in the options above, including [all candidate parents], the correct parents of the mention, [mention name], include [correct gold parents]. Thus the options are narrowed down to [option numbers having correct gold parents]. From the children in the narrowed options, including [children in the filtered options], the correct children of the mention, [mention name], include [correct gold children in the filtered options]. Thus, the final answers are [correct option numbers].
```

We place the explanation section (`### Explanation`) before the response section (`### Response`). During training, the whole explanation is fed into the LLM to allow it to be conditioned to generate the response. During inference, the instruction-tuned LLM is expected to generate an explanation of the same template structure, after the explanation section mark (`### Explanation`), with a response (as a part of the explanation and also in the response section).

An issue with current LLMs is the limited text window it can support. This long context issue however will be addressed with future LLMs. At this stage, we test the framework with an openly available LLM, Llama 2, which supports 4,096 tokens as input, sufficient for a low or medium top- k setting as 10 or 50.

5 Experiments

5.1 Data Construction

We adapt datasets MM-S14-Disease and MM-S14-CPP from the work in [7] for new concept placement in ontologies⁸. The datasets are constructed by using two versions of SNOMED CT (2014.09 and 2017.03) with a text corpus where mentions are linked to UMLS. Then mapping between UMLS and SNOMED CT is also available in the UMLS. New mentions are therefore synthesised by considering the gap between the two versions of SNOMED CT. The edges to be inserted into the ontology for each new mention are also created, by finding the nearest parents and children for the new mention in the old version of SNOMED CT. The statistics of the dataset are displayed in Table 1.

⁸ <https://zenodo.org/records/10432003>

Table 1. Statistics for datasets for Concept Placement, for SNOMED CT (ver 20140901, “S14”) under different categories: “Disease” and “CPP”, i.e., *Clinical finding*, *Procedure*, and *Pharmaceutical / biologic product*. A mention-edge pair or link (in L) denotes a mention (in M) and one of its directed edges in the KB. The mention-edge pair is complex (i.e. L_{comp}) when the edge involves a complex concept. Mentions are from the MedMentions dataset (“MM”). The numbers of edges are those having one hop (including leaf nodes to NULL) and two hops from any paths in the ontology. (Table adapted from the study [7].)

		MM-S14-Disease	MM-S14-CPP
Ontology:	concepts	64,900 (824)	175,895 (2,718)
# all (# complex)	edges	237,826 (4,997)	625,994 (19,401)
Corpus:	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
# M / # L / # L_{comp}			

The number of edges (one-hop including leaf nodes and two-hop) is numerous, over 3.5 times of the number of concepts. This makes the task of placement into edges less tractable than entity linking into a concept for a mention.

We consider the unsupervised setting of concept placement common to the real-world scenario, which means that no mention-edge pairs for out-of-KB concepts are available for the training. This can, however, be approached using in-KB self-supervised data creation: we can see from Table 1 that it is possible to generate edges for in-KB concepts; this is simply by looking at the directed parents and children of a concept in the current ontology (i.e., the older version of SNOMED CT). Thus, we use in-KB data for training and validation, and then use out-of-KB data solely for external validation and testing.

5.2 Metrics

We present new metrics for new concept placement, as insertion rate for any edges (InR_{any}) and for all edges (InR_{all}) predicted for mentions. Here “any” means that one of the gold edges is predicted for a mention, whereas “all” means that all of the gold edges are predicted. The metrics can be defined as below in Equation 4, where the value of $\mathbb{1}(x)$ is 1 where the statement x is true, otherwise 0, and Z_i and Y_i are the set of predicted edges and gold labels (or edges), resp. Also, we use the insertion rates at k (i.e., $InR_{any}@k$ and $InR_{all}@k$) to denote the performance after predicting the top- k edges, this measures whether the “positive” edges are ranked before the “negative” ones. We select k as 1, 5, and 10, considering that terminologists can select from a few edges (as few as 10 or less) suggested by a system for updating an ontology.

$$InR_{any} = \frac{1}{|M|} \sum_{m_i \in M} \mathbb{1}(Z_i \cap Y_i \neq \emptyset); \quad InR_{all} = \frac{1}{|M|} \sum_{m_i \in M} \mathbb{1}(Z_i \supseteq Y_i) \quad (4)$$

The proposed metrics can be considered a loose version (“any”) and a strict version (“all”) of the example-based metrics [10] for multi-label learning. The standard multi-label learning requires a complete set of gold labels, while ontologies that follow the open-world assumption are inherently incomplete (i.e., edges

which are not in the gold standard may also be correct), thus the ranking-based metrics, $InR_{any}@k$ and $InR_{all}@k$, are more appropriate.

The insertion rate metrics can be used to evaluate both edge candidates and final edge selection. We also separately evaluate the insertion rate metrics for leaf edges (where the child edge is NULL) and non-leaf edges.

5.3 Experimental Settings and Baseline Methods

We select two representative top- k values after the edge enrichment step, $k = 10$ and $k = 50$, enriched from $\frac{k}{2}$ edges, or 5 and 25 seed edges resp., after an initial investigation of a range of k values⁹. Then for each of the top- k settings, the models select the final set of top 1, 5, and 10 edges after the edge selection step.

For edge search, the baselines include an inverted index based approach, fixed BERT embeddings, and fine-tuned BERT embeddings with contrastive learning (as Edge-Bi-encoder). For all methods, the sub-token length of contexts and concepts are 32 and 128 resp. We choose SapBERT [20] as the BERT model in edge search (fixed and fine-tuned embeddings). For the inverted index based approach, we create an inverted index from all SNOMED CT concepts, where a key is a sub-token from a concept and a corresponding value is all the concepts, and we use the index of sub-tokens created using the SentencePiece tokenizer (also used by FLAN-T5) [19]. The similarity score based on the inverted index between a mention and a concept is then calculated as the sum of inverse document frequency scores ($\text{sim}_{idf}(m, C, \mathcal{T}, I, |D|) = \sum_{t \in \mathcal{T}(C) \cap \mathcal{T}(m)} \log \frac{|D|}{|I[t]|}$) of all the common sub-tokens t that appear in both the mention m and a concept C in the set of all concepts D , and \mathcal{T} is the tokenizer and I is the index from a sub-token t to the list of concepts. Then the edge score w.r.t. a mention is calculated similarly to Equation 2, as the average of mention-parent similarity and mention-child similarity score using the inverted index. For inverted index and fixed embedding, we use the mention only without contexts, considering that methods do not learn the relation between the concept and the natural language context; for Edge-Bi-encoder we explored mentions with or without contexts.

We apply all baseline methods with the steps of edge formation and enrichment. Then, for edge selection, we choose PubMedBERT [13] as the model for fine-tuning cross-encoder-based method; we also choose GPT-3.5 (“gpt-3.5-turbo-0613”)¹⁰, and Llama 2 [26] for the zero-shot prompting of LLMs, both models allowing 4,096 sub-tokens as input. FLAN-T5 [5] has a limited input token length of 512, below the token usage of our prompts with the top-50 settings (between 1,556 and 3,014 sub-tokens for the datasets for top-50), thus we only use it for the top-10 setting. The model GPT-4 has a much higher cost, 30 folds of the price compared to GPT-3.5, and is slower and less stable in querying, and GPT-4 is also under updating, thus we only report results for GPT-3.5.

⁹ We also investigated k up to 300, while the insertion rate at k improves, the overall results after edge selection are worse than smaller k values as 10 and 50. A larger k also leads to a substantially longer running time for edge enrichment and selection.

¹⁰ <https://platform.openai.com/docs/models/gpt-3-5>

Table 2. Results on edge search, formation and enrichment for MM-S14-Disease and MM-S14-CPP datasets. Each setting has validation and testing results, separated by a slash (/) sign. “*lf*” and “*nlf*” mean *leaf* and *non-leaf*, resp.

MM-S14-Disease	k	InR_{any}	InR_{all}	InR_{any}, lf	InR_{all}, lf	InR_{any}, nlf	InR_{all}, nlf
Inverted Index	10	10.0 / 12.0	9.1 / 10.1	9.5 / 14.2	9.2 / 11.9	14.3 / 3.4	8.6 / 3.4
	50	41.3 / 40.6	37.7 / 38.8	44.6 / 50.0	41.2 / 48.2	14.3 / 5.2	8.6 / 3.4
Fixed embs	10	16.1 / 13.0	7.0 / 12.3	18.0 / 16.1	7.8 / 16.0	0.0 / 1.7	0.0 / 0.0
	50	35.3 / 31.9	28.3 / 30.8	38.4 / 38.1	30.6 / 37.6	8.6 / 8.6	8.6 / 5.2
Fine-tuned embs (Edge-Bi-enc)	10	31.9 / 25.7	14.6 / 8.0	28.9 / 12.4	14.6 / 8.7	57.1 / 75.9	14.3 / 5.2
	50	57.8 / 50.0	40.1 / 38.0	55.4 / 38.1	38.4 / 33.5	77.1 / 94.8	54.3 / 55.2
MM-S14-CPP	k	InR_{any}	InR_{all}	InR_{any}, lf	InR_{all}, lf	InR_{any}, nlf	InR_{all}, nlf
Inverted Index	10	5.5 / 5.8	5.1 / 5.3	5.3 / 5.7	5.1 / 5.7	6.9 / 6.3	5.2 / 3.1
	50	23.1 / 23.4	21.0 / 22.5	24.9 / 26.9	22.8 / 25.8	6.9 / 3.1	5.2 / 3.1
Fixed embs	10	11.3 / 8.3	8.3 / 7.4	12.4 / 9.2	9.2 / 8.7	1.7 / 3.1	0.0 / 0.0
	50	28.4 / 26.9	25.9 / 25.2	30.4 / 30.2	28.8 / 29.6	10.3 / 7.8	0.0 / 0.0
Fine-tuned embs (Edge-Bi-enc)	10	32.0 / 27.8	19.7 / 14.4	31.2 / 19.3	21.4 / 16.0	39.7 / 76.6	5.2 / 4.7
	50	50.9 / 48.4	36.8 / 34.5	50.4 / 42.4	38.8 / 38.0	55.2 / 82.8	19.0 / 14.1

For LLM instruction-tuning, we use the Supervised Fine-tuning (SFT)¹¹ with 4-bit quantisation to fine-tune the Llama-2 model; the efficient instruction-tuning uses QLoRA, quantisation with Low Rank Adapters (LoRA) [6].

For all supervised models (fine-tuning and instruction-tuning), we use in-KB data for training. The best models were selected by using the validation set of the in-KB data. We then report results on the validation and the test sets for the out-of-KB data. Note that the out-of-KB validation set is not used for parameter tuning and is independent of model development.¹²

5.4 Results

We report results on the first two steps to determine the best edge search methods, followed by the overall results of the full framework, with edge selection. The metric results in all Tables are presented as percentage scores.

Results on Edge Search, Formation and Enrichment Results are presented in Table 2. The “all” metrics are generally lower than the “any” metrics (also for results in the other tables) as the full completion for concept placement is more challenging than the placement into any correct edges.

It can be observed that the fine-tuned Edge-Bi-encoder achieves the best overall results under the settings. The inverted index approach has a higher coverage of leaf edges for Diseases (but not for the broader categories of CPP) - this may be because the parent disease names are likely to be lexically similar to the new mention, while for non-leaf edges, fixed and fine-tuned embedding-based methods achieve higher performance; also, the inverted index and fixed embeddings tend to prioritise leaf edges, based on the rule by checking whether the top seed concept is a leaf concept.

¹¹ https://huggingface.co/docs/trl/sft_trainer

¹² More details on experimental settings and time usage are in Appendix 1.

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
+ 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
+ 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
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
+ 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
+ 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

Overall Results after Edge Selection We then add the edge selection steps mainly on the candidates from the fine-tuned embedding (Edge-Bi-encoder) approach, given its best overall performance in generating edge candidates.

As shown in Table 3, the multi-label Edge-Cross-encoder achieves the best performance in most experimental settings. Edge-Cross-encoder further reranks the edge candidates and helps substantially improve the performance over Edge-Bi-encoder (edge search only), e.g., by around 8-9% absolute scores for $InR_{any}@5$ for the datasets and around 12-16% for $InR_{any}@10$ (except for the same @10 results for top-10 setting, where re-ranking does not make a difference).

We also test the LLMs, it can be seen that the tested medium scale LLMs (GPT-3.5, FLAN-XL, and Llama-2-7B), especially not instruction-tuned for the task, can still not be directly used for concept placement, although GPT-3.5 has notably better results on top edge suggestion ($InR_{any}@1$) from the top-10 setting. The explainable instruction tuning approach greatly improves the performance of Llama-2-7B. This shows that training on in-KB data by generating an automated explanation before generating the results is practically useful to enhance the capability of LLMs on ontology reasoning tasks. Most results from LLMs, except for the top edge suggestion ($InR_{any}@1$ and $InR_{all}@1$), are still below the original candidates from Edge-Bi-encoder. Nevertheless, the results are encouraging and can motivate future studies using LLMs for concept placement.

We also notice a performance gap between the validation set and the test set on the two datasets in Tables 2-3, which may be due to the high variance caused by the small number of mentions in the sets (between 200 and 600, see Table 1) and the distinct data distribution based on concept drift (e.g., different lexical mentions between the sets), showing the challenge to generalise to new concepts.

Discussion on Model Applicability The overall performance of the models is not high, especially for LLMs, as shown in Table 3. The best InAny@10 is around 30% with Edge-Bi-encoder and Edge-Cross-encoder. This shows that the models cannot support an automated application, but still, they may potentially be applicable to suggest a ranking of the edges for human terminologists to add a new concept to an ontology. In practice, having a larger k can help improve the metrics of $InR_{any}@k$ and $InR_{all}@k$, but can also increase the effort of manual selection, thus a balance needs to be achieved and warrants future studies.

Case Study In Appendix 3, we select a few test mentions and display the 5 top edge suggestions, under the top-50 setting from Edge-Bi-encoder, Edge-Cross-encoder, Llama-2-7B, and Llama-2-7B fine-tuned models. For Llama-2 models, we display the generated answers. Without instruction tuning, Llama-2-7B sometimes generates answers in an incorrect format or generates irrelevant outputs. With explainable instruction-tuning, Llama-2-7B generates explanations that follow a natural language reasoning path to lead to the correct edge option. We also note that merging with existing concepts is needed as a further step after the placement of the mention, e.g., for Chronic kidney disorder.

We also note that many edge predictions are not completely wrong, for example for the first case, the predicted parent (e.g., kidney disease) in the methods is more general than the gold direct parent (e.g., renal impairment), and the predicted child (e.g., hypertensive heart and renal disease with renal failure) is not far from the gold children in the ontology structure. However, calculating a lenient, soft score (e.g., with Wu & Palmer similarity [32]) between every prediction and the set of gold edges instead of a binary evaluation is not time efficient in our experiments. We leave an efficient, lenient evaluation for future studies.

5.5 Ablation Studies

Our ablation studies aim to investigate how edge enrichment and automated explainable instruction tuning can enhance the performance of concept placement. We use top-50 setting and MM-S14-Disease dataset as an example, and other top- k settings and MM-S14-CPP dataset follow a similar pattern of results.

Edge Enrichment Edge enrichment has greatly improved the overall results for Edge-Bi-encoder, about above 30% absolute improvement of InR_{any} and InR_{all} as shown in Figure 3 part (a). Results with inverted index and fixed embeddings are in Appendix 2, showing a general improvement with edge enrichment.

Contextual Information of Mention The left and right contexts of a mention are useful in edge search with Edge-Bi-encoder to learn the similarity between a contextual mention and a concept, even though the type of texts is distinct from ontology concept labels, as shown in Figure 3 part (b).

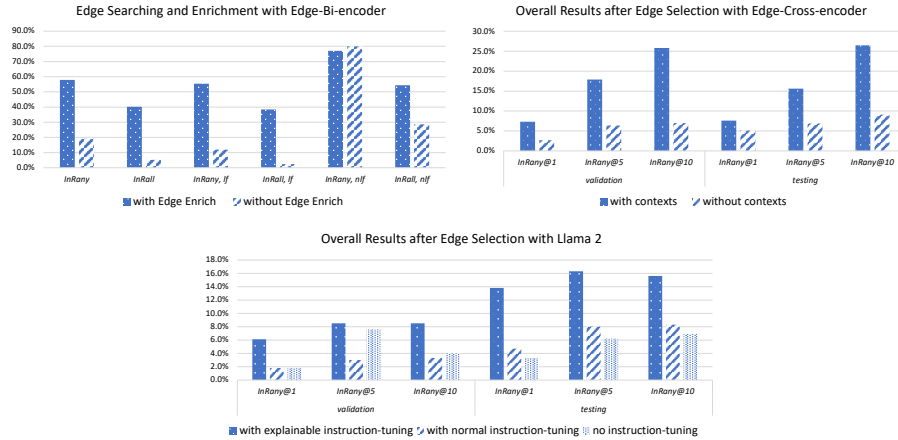


Fig. 3. Ablation results on top-50 edge candidates with MM-S14-Disease dataset: (a) Top-left: results with Edge-Bi-encoder, with or without the edge enrichment step, on validation set; (b) Top-right: overall results with Edge-Cross-encoder, with or without contexts; (c) Bottom: overall results after Edge Selection with Llama-2-7B, with explainable instruction-tuning, normal instruction-tuning, or without instruction-tuning.

Explainable Instruction Tuning Explainable instruction tuning helps improve the performance of the LLM, Llama-2-7B, under the top-50 setting, especially on the test set of MM-S14-Disease, as displayed in Figure 3 part (c). In contrast, normal instruction tuning that directly generates the edge option number does not always improve over the case without instruction tuning.

5.6 Conclusion and Future Studies

We propose an LM-based framework for new concept placement in ontologies. The framework uses a three-step approach, that enhances the two-step information retrieval with edge formation and enrichment leveraging the ontological structure. The results overall show that methods that fine-tune PLMs perform the best, while there is an encouraging performance for the recent LLMs, especially with explainable instruction tuning. Our case study shows that explanations can be generated to detail the steps for concept placement. We focused on placing directly the mentions into edges in this work, a following step is to group or merge mentions of the same new concept, and also with existing concepts if they have the same meaning, when they are placed into the same edges. Future studies will further explore LLM-generated explanations and leverage advanced Retrieval Augmented Generation [9] and prompting strategies. Future studies also need to investigate how to use the methods to assist human terminologists.

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

References

1. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: A Basic Description Logic, p. 10–49. Cambridge University Press, Cambridge (2017). <https://doi.org/10.1017/9781139025355.002>
2. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: Ontology Languages and Applications, p. 205–227. Cambridge University Press, Cambridge (2017). <https://doi.org/10.1017/9781139025355.008>
3. Chen, J., Dong, H., Hastings, J., Jiménez-Ruiz, E., Lopez, V., Monnin, P., Pesquita, C., Škoda, P., Tamma, V.: Knowledge graphs for the life sciences: Recent developments, challenges and opportunities. arXiv preprint arXiv:2309.17255 (2023)
4. Chen, J., He, Y., Geng, Y., Jiménez-Ruiz, E., Dong, H., Horrocks, I.: Contextual semantic embeddings for ontology subsumption prediction. World Wide Web pp. 1–23 (2023)
5. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416 (2022)
6. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314 (2023)
7. Dong, H., Chen, J., He, Y., Horrocks, I.: Ontology enrichment from texts: A biomedical dataset for concept discovery and placement. In: Proceedings of the 32nd ACM International Conference on Information & Knowledge Management. Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3583780.3615126>
8. Dong, H., Chen, J., He, Y., Liu, Y., Horrocks, I.: 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. p. 452–462. CIKM '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3583780.3615036>, <https://doi.org/10.1145/3583780.3615036>
9. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)
10. Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Comput. Surv. **47**(3) (apr 2015). <https://doi.org/10.1145/2716262>, <https://doi.org/10.1145/2716262>
11. Glauer, M., Memariani, A., Neuhaus, F., Mossakowski, T., Hastings, J.: Interpretable ontology extension in chemistry. Semantic Web **Pre-press**(Pre-press), 1–22 (2023)
12. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for owl. Journal of Web Semantics **6**(4), 309–322 (2008), semantic Web Challenge 2006/2007
13. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H.: Domain-specific language model pretraining for biomedical natural language processing. ACM Trans. Comput. Healthcare **3**(1) (oct 2021). <https://doi.org/10.1145/3458754>
14. He, Y., Chen, J., Dong, H., Horrocks, I.: Exploring large language models for ontology alignment. arXiv preprint arXiv:2309.07172 (2023)
15. He, Y., Chen, J., Dong, H., Horrocks, I., Allocca, C., Kim, T., Sapkota, B.: Deep-onto: A python package for ontology engineering with deep learning. arXiv preprint arXiv:2307.03067 (2023)

16. He, Y., Chen, J., Jimenez-Ruiz, E., Dong, H., Horrocks, I.: Language model analysis for ontology subsumption inference. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics: ACL 2023. pp. 3439–3453. Association for Computational Linguistics, Toronto, Canada (Jul 2023). <https://doi.org/10.18653/v1/2023.findings-acl.213>, <https://aclanthology.org/2023.findings-acl.213>
17. Hertling, S., Paulheim, H.: Transformer based semantic relation typing for knowledge graph integration. In: European Semantic Web Conference. pp. 105–121. Springer (2023)
18. Jurafsky, D., Martin, J.H.: Speech and Language Processing (3rd Edition). Online (2023)
19. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Blanco, E., Lu, W. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). <https://doi.org/10.18653/v1/D18-2012>, <https://aclanthology.org/D18-2012>
20. Liu, F., Shareghi, E., Meng, Z., Basaldella, M., Collier, N.: Self-alignment pretraining for biomedical entity representations. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4228–4238. Association for Computational Linguistics, Online (Jun 2021). <https://doi.org/10.18653/v1/2021.naacl-main.334>
21. Liu, H., Perl, Y., Geller, J.: Concept Placement Using BERT Trained by Transforming and Summarizing Biomedical Ontology Structure. *J. of Biomedical Informatics* **112**(C) (2020)
22. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1410>
23. Ruas, P., Couto, F.M.: Nilinker: Attention-based approach to nil entity linking. *Journal of Biomedical Informatics* p. 104137 (2022). <https://doi.org/https://doi.org/10.1016/j.jbi.2022.104137>, <https://www.sciencedirect.com/science/article/pii/S1532046422001526>
24. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* **27**(2), 443–460 (2014)
25. Spackman, K.A., Dionne, R., Mays, E., Weis, J.: Role grouping as an extension to the description logic of ontolog, motivated by concept modeling in snomed. In: Proceedings of the AMIA Symposium. p. 712. American Medical Informatics Association (2002)
26. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)

28. Veseli, B., Singhanian, S., Razniewski, S., Weikum, G.: Evaluating language models for knowledge base completion. In: Pesquita, C., Jimenez-Ruiz, E., McCusker, J., Faria, D., Dragoni, M., Dimou, A., Troncy, R., Hertling, S. (eds.) *The Semantic Web*. pp. 227–243. Springer Nature Switzerland, Cham (2023)
29. Wang, Q., Gao, Z., Xu, R.: Exploring the in-context learning ability of large language model for biomedical concept linking. arXiv preprint arXiv:2307.01137 (2023)
30. Wang, S., Zhao, R., Zheng, Y., Liu, B.: Qen: Applicable taxonomy completion via evaluating full taxonomic relations. In: *Proceedings of the ACM Web Conference 2022*. p. 1008–1017. WWW '22, Association for Computing Machinery, New York, NY, USA (2022), data available at <https://github.com/sheryc/QEN>
31. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 6397–6407. Association for Computational Linguistics, Online (Nov 2020). <https://doi.org/10.18653/v1/2020.emnlp-main.519>
32. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: *32nd Annual Meeting of the Association for Computational Linguistics*. pp. 133–138. Association for Computational Linguistics, Las Cruces, New Mexico, USA (Jun 1994). <https://doi.org/10.3115/981732.981751>, <https://aclanthology.org/P94-1019>
33. Zeng, Q., Lin, J., Yu, W., Cleland-Huang, J., Jiang, M.: Enhancing taxonomy completion with concept generation via fusing relational representations. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. p. 2104–2113. Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3447548.3467308>, data available at <https://github.com/DM2-ND/GenTaxo>
34. Zhang, J., Song, X., Zeng, Y., Chen, J., Shen, J., Mao, Y., Li, L.: Taxonomy completion via triplet matching network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. pp. 4662–4670. AAAI Press, Palo Alto, California, USA (2021), data available at <https://github.com/JieyuZ2/TMN>
35. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)

Appendix 1: Detailed model settings and time usage

The approaches are implemented using PyTorch and Huggingface Transformers. Edge-Bi-encoder and Edge-Cross-encoder are originally based on the architectures of BLINKout [8] (based on BLINK [31]). Inverted index with ontology concepts is based on DeepOnto Library [15]. The batch sizes for Edge-Bi-encoder and Edge-Cross-encoder are 16 and 1, resp. The fine-tuning of Edge-Bi-encoder and Edge-Cross-encoder takes 1 and 4 epochs, resp. We limit the rows to 200,000 for training the Edge-Cross-encoder models given the sufficient amount the data for model convergence and the long time of training. The instruction tuning of Llama-2-7B uses a 4-bit quantisation and takes 3 epochs with a batch size of 4.

Time usage We run all models using an NVIDIA Quadro RTX 8000 GPU card (48GB GPU). We report the time usage estimate for MM-S14-Disease under the top-50 setting. Training bi-encoder took around 29 hours. Training cross-encoder took around 4 hours. Instruction tuning of Llama-2-7B took around 16 hours. Inferencing with fixed embeddings and inverted index with edge enrichment is within around 0.5 and 1 second per mention, resp. Inferencing with Edge-Bi-encoder only takes around 0.2 second per mention. The whole inferencing with both Edge-Bi-encoder and Edge-Cross-encoder takes around 2.3 seconds per mention. The prompting of an explainable instruction-tuned Llama-2-7B model takes around 78 seconds per mention to output natural language explanations.

Appendix 2: Detailed results on edge enrichment

We applied edge formation enrichment over inverted index and fixed embedding approach. Results in Table 4 show a substantial improvement for InR_{any} and InR_{all} . We see that the mentions to be placed to non-leaf edges are not improved with inverted index and fixed embeddings, but are improved with the fine-tuned, Edge-Bi-encoder, this is because the latter places a more lenient score for the leaf edges that do not always rank them before the non-leaf edges.

Table 4. Results on edge search and enrichment (vs. not using edge enrichment) for MM-S14-Disease, under the top-50 setting. Each setting has validation and testing results, separated by a slash (/) sign. “*lf*” and “*nlf*” mean *leaf* and *non-leaf*, resp.

MM-S14-Disease	k	InR_{any}	InR_{all}	$InR_{any, lf}$	$InR_{all, lf}$	$InR_{any, nlf}$	$InR_{all, nlf}$
Inverted Index	50	41.3 / 40.6	37.7 / 38.8	44.6 / 50.0	41.2 / 48.2	14.3 / 5.2	8.6 / 3.5
w/o Edge Enrich	50	7.3 / 6.5	4.6 / 3.6	2.7 / 3.7	0.7 / 0.5	45.7 / 17.2	37.1 / 15.5
Fixed embs	50	35.3 / 31.9	28.3 / 30.8	38.4 / 38.1	30.6 / 37.6	8.6 / 8.6	8.6 / 5.2
w/o Edge Enrich	50	19.1 / 17.3	4.3 / 4.3	16.0 / 6.4	3.7 / 4.1	45.7 / 58.6	8.6 / 5.2
Edge-Bi-Enc	50	57.8 / 50.0	40.1 / 38.0	55.4 / 38.1	38.4 / 33.5	77.1 / 94.8	54.3 / 55.2
w/o Edge Enrich	50	19.1 / 23.9	5.2 / 5.8	11.9 / 5.0	2.4 / 2.8	80.0 / 94.8	28.6 / 17.2

Appendix 3: Qualitative examples

Examples of a non-leaf and a leaf concept placement, with prompt options, model predictions, and instruction-tuned Llama-2-7B’s explanations, are in Table 5.

Table 5. Examples of two mentions in the out-of-KB test set of MM-S14-Disease to enrich SNOMED CT 2014.09. The correct predictions are in **bold**. (Note: while the concept Chronic kidney disease in SNOMED CT ver 2017.03 is not in ver 2014.09, it is modified from Chronic renal impairment, ID 236425005, in the older ontology.)

	Test, out-of-KB, 13	Test, out-of-KB, 138
Mention in contexts	...Since no one had *CKD* in partial nephrectomized patients, we determined risk factors for CKD in radical nephrectomized patients...	Development of a novel near-infrared fluorescent theranostic combretastain A-4 analogue, YK-5-252, to target triple negative breast cancer. The treatment of triple negative breast cancer (*TNBC*) is a significant challenge to cancer research...
Gold Concept	http://snomed.info/id/709044004 Chronic kidney disease (not available in SNOMED CT 2014.09)	http://snomed.info/id/706970001 Triple negative malignant neoplasm of breast (not available in SNOMED CT 2014.09)
Gold Edges	Parents: (i) Renal impairment → Children: (i) Chronic renal impairment associated with type II diabetes mellitus; (ii) Hypertensive heart and chronic kidney disease; (iii) Chronic kidney disease stage 1; (iv) Chronic kidney disease stage 2; (v) Chronic kidney disease stage 3; (vi) Chronic kidney disease stage 4; (vii) Chronic kidney disease stage 5; (viii) Chronic renal failure syndrome; (viiii) Hypertensive heart AND chronic kidney disease on dialysis; (x) Chronic kidney disease due to hypertension; Malignant hypertensive chronic kidney disease	Parents: (i) Human epidermal growth factor 2 negative carcinoma of breast; (ii) Malignant tumor of breast; (iii) Hormone receptor negative neoplasm → Children: (i) NULL
Edge-Bi-enc	(i) renal impairment → end stage renal disease (ii) renal impairment → renal failure following molar and/or ectopic pregnancy (iii) renal impairment → renal failure syndrome (vi) renal impairment → chronic kidney disease due to hypertension (v) kidney disease → impaired renal function disorder	(i) malignant tumor of breast → lobular carcinoma of breast (ii) carcinoma of breast → lobular carcinoma of breast (iii) malignant tumor of breast → mucinous carcinoma of breast (iv) carcinoma of breast → cancer en cuirasse (v) malignant tumor of breast → malignant phylloides tumor of breast
Edge-Cross-enc	(i) kidney disease → renal function impairment with growth failure (ii) kidney disease → impaired renal function disorder (iii) disorder of the genitourinary system → impaired renal function disorder (iv) renal impairment → chronic kidney disease due to hypertension (v) kidney disease → renal failure syndrome	(i) neoplasm of breast → malignant phylloides tumor of breast (ii) carcinoma of breast → cancer en cuirasse (iii) malignant tumor of breast → NULL (iv) carcinoma of breast → NULL (v) neoplasm of breast → NULL
Prompt, only options are displayed (correct options in bold), the full prompt template is in Sect 4.3	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	options: 0.malignant tumor of breast → lobular carcinoma of breast 1.carcinoma of breast → lobular carcinoma of breast 2.malignant tumor of breast → mucinous carcinoma of breast 3.carcinoma of breast → cancer en cuirasse 4.malignant tumor of breast → malignant phylloides tumor of breast 5.malignant tumor of breast → cancer en cuirasse 6.neoplasm of breast → malignant phylloides tumor of breast 7. malignant tumor of breast → NULL 8.carcinoma of breast → NULL 9.neoplasm of breast → NULL
Gold answer	3,7 (option incomplete)	7 (option incomplete)
Llama-2-7B	1	Fail to generate answer with the relevant content
Llama-2-7B instruction-tuned	### 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	### Explanation: From the parents in the options above, including malignant tumor of breast , carcinoma of breast , neoplasm of breast , the correct parents of the mention, TNBC, include malignant tumor of breast . Thus the options are narrowed down to 0, 2, 4, 5, 7. From the children in the narrowed options, including lobular carcinoma of breast , mucinous carcinoma of breast , malignant phylloides tumor of breast , cancer en cuirasse , NULL , the correct children of the mention, TNBC, include NULL. Thus, the final answers are 7 . ### Response: 7