Hierarchical Knowledge Retrieval using Transformer-based Ontology Embeddings in SNOMED CT

A dissertation submitted to The University of Manchester for the degree of

Master of Science in Advanced Computer Science

in the Faculty of Science and Engineering

Year of submission

2025

Student ID

8516047

School of Computer Science

Contents

Word count: 8862

Terms and abbreviations

Glossary

$\mathcal{O}$	OWL 2 (EL) ontology (SNOMED CT in experiments)
N_C, N_R, N_I	Named classes, roles, and individuals
$C \sqsubseteq D$	Subsumption; entailed by $\mathcal{O}$
$C \sqsubseteq_{\text{st}} D$	stated `subClassOf` relation $\in \mathcal{O}$
$Anc(C)$, $Anc_{\leq d}(C)$	(Bounded) ancestor (superclass) sets
$t(C)$	Textual representation of class C
q_s	Query string or entity mention
f_{emb}	Text encoder; maps text $\rightarrow$ embedding space
$Ret_X(q)$	Ranked retrieval set/rankedlist under method X
$Rel(q)$	Relevant gold reference set/ranked list for q_s
OOV (Mention)	Out-of-vocabulary entity mention ($\notin \mathcal{O}$)

Abstract

Many modern information systems rely on faithful knowledge retrieval to function effectively. For instance, large language models protect against hallucinations by leveraging techniques such as retrieval augmented generation (RAG). Meanwhile, the healthcare industry relies on structured knowledge bases, such as SNOMED CT, to aid decision support and to enable clinical reporting. Despite the central role that knowledge retrieval plays, LLMs continue to struggle with factual accuracy, and studies on effective retrieval for SNOMED CT remain limited. Motivated by these shortcomings, this work investigates the effectiveness of ontology-aware (hyperbolic) bi-encoders, focusing on the Hierarchy and Ontology Transformer frameworks (HiT and OnT, respectively). Through an investigation of ontology-grounded knowledge retrieval using SNOMED CT, we assess whether OnT-based retrieval improvements transfer to downstream tasks, including RAG-based BioMedical MCQA and web-based search. We construct an out-of-vocabulary (OOV) mention set using the MIRAGE benchmark, annotate gold target reference classes from SNOMED CT, and evaluate retrieval in single-target, multi-target and application-specific settings. We compare our results against strong lexical (TF–IDF, BM25) and contextual (Sentence-BERT) baselines, whilst evaluating potential for exploratory techniques such as mixed model spaces with heterogeneous curvature. For single-target retrieval, we find that the best-performing OnT models provide a 13-point gain in MRR over SBERT and more than double the relative performance when compared to lexical baselines. Similarly, in the multi-target setting, ontology encoders continue to outperform both lexical and contextual baselines, where a depth-biased subsumption score further improves mAP by 1–2 points compared to measures of pure geodesic distance (whilst eliciting minimal nDCG trade-off). Despite these performance improvements, single (top-1) concept retrieval applied to vanilla RAG for biomedical MCQA shows no significant accuracy gains on MIRAGE. Limitations likely stem from language model mismatch, short context length and insufficiencies tied to axiom verbalisation, suggesting that further work is required. We release a modular retrieval toolkit, annotated OOV queries, and a reproducible artefact to support future work, available at `https://github.com/jonathondilworth/uom-thesis`.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Acknowledgments

I would like to thank Dr. Jiaoyan Chen for providing me with continuous guidance throughout the project duration. I also extend my thanks to Dr. Hui Yang, who provided encouragement and lent time to assist in model training. I would also like to thank my family for continuous support throughout the MSc. I would like to thank my family for continuous support throughout the MSc.

1 Introduction

Large language models (LLMs), such as ChatGPT [1] and Gemini [2], have gained significant attention in recent years [3], mainly due to their ability to generate convincing natural language responses to user queries. Still, they face significant challenges, especially in domain-specific, knowledge-intensive tasks. For instance, they often hallucinate, providing incorrect or non-factual information [4]. These limitations severely impact their utility, particularly in high-stakes application domains such as healthcare [5], [6]. To address these challenges, researchers at Meta AI popularised the now widely adopted technique known as Retrieval-Augmented Generation (RAG) [7]. RAG aims to reduce hallucinations by retrieving additional context in an attempt to ground language model responses in verifiable information from external knowledge bases [8]. Its effectiveness, however, is based on the quality and structure of the underlying knowledge as well as the effectiveness of the employed knowledge retrieval techniques [9], [10].

Meanwhile, the healthcare industry requires high-quality, structured domain knowledge to aid in decision support, clinical reporting and biomedical research. For instance, SNOMED CT [11] is a clinical terminology widely used to organise electronic health records and provide semantic interoperability between clinical information systems [12], [13]. Given SNOMED CT's vital role in facilitating healthcare systems, a supplemental OWL 2 [14] distribution of the terminology is made available to knowledge engineers to maintain high levels of data quality and correctness [15]. However, despite SNOMED CT's importance, its navigation can often prove difficult for healthcare practitioners without formal training in specialised query languages [16]. Specifically, existing implementations that allow for user-based knowledge retrieval, e.g. the SNOMED CT browser, primarily support lexical search and, where available, expression constraint language (ECL), often associated with a steep learning curve [17]. Additionally, studies on improving retrieval performance are limited [12], [13], [16], with only a small number investigating the integration of semantic search for SNOMED CT.

These examples serve to highlight the importance of effective knowledge retrieval for downstream tasks, such as biomedical question answering and web-based search. Furthermore, in the case of SNOMED CT search, we note that knowledge representations often capture relationships that are difficult to model through lexical techniques alone. For instance, traditional retrieval methods, such as TF–IDF and BM25 [18], rely on surface-form overlap, are brittle to synonymy and polysemy [19], and ultimately fail to capture formally structured semantics. Similarly, contextual embedding-based approaches to retrieval do benefit from distributional semantics [20], though they often overlook notions of taxonomic depth and are likely insufficient to model complex ontological relationships without structural cues [21]. These shortcomings illustrate the relative difficulty of effective knowledge retrieval in such systems, including RAG pipelines, motivating further research of specialised embedding methods.

Ontology embeddings [22] are the product of recent research efforts in investigating neural representation learning as applied to ontologies, where applications include tasks such as logical inference, prediction and retrieval [23]. These efforts aim to deliver models that capture, encode and approximate structural properties of formal knowledge representations, such as transitivity, within the model's native embedding space. For instance, Nickel and Kiela [24] propose an embedding strategy based on Riemannian optimisation that maps hierarchical relationships to hyperbolic space within the Poincaré ball model. While their technique achieves strong performance on hierarchical link prediction, it assumes a graph-based input with a purely geometric objective, ignoring language modelling. On the other hand, many approaches that aim to jointly capture textual annotations and knowledge structure, such as OPA2Vec [25], are based on static word embeddings, which lack contextualisation and operate in Euclidean space, where the geometry does not naturally reflect ontological structure. To overcome these limitations, He et al. introduced the Hierarchy Transformer (HiT) [26] and Ontology Transformer (OnT) [27], which jointly capture structural and textual signals during pre-trained language model (PLM) re-training through optimisation of hyperbolic contrastive objectives. These approaches have proved effective in tasks such as axiom prediction and approximate inference; however, their use in applied information retrieval (e.g. user-based web search) and retrieval augmented generation pipelines remain relatively under-explored.

Motivated by the earlier examples, this work seeks to investigate the potential utility of ontology embeddings, as implemented through hyperbolic bi-encoders such as HiT and OnT, for retrieval and associated downstream applications. First, we assess whether HiT and OnT provide measurable benefits in ontology-grounded knowledge retrieval, then look to investigate potential applications, such as RAG. Additional contributions include a web-based search interface for knowledge retrieval, provided alongside a simple RAG prototype. Ablation studies that consider various ontology embedding models, including mixed models, are also conducted.

1.1 Aims and objectives

1.1.1 Aims

Primary Investigate the effectiveness of transformer-based ontology embeddings for single target and multi-target knowledge (named class/atomic concept) retrieval in $\mathcal{EL}$-ontologies (SNOMED CT), relative to strong lexical (TF–IDF, BM25) and contextual (SBERT) baselines.

Secondary Assess whether, and to what degree, observed performance in knowledge retrieval propagates to downstream biomedical question-answering for a standardised retrieval augmented generation (RAG) pipeline with concept-verbalisation and prompt-stuffing.

Tertiary Consider exploratory methods and ablation studies within the context of the primary and secondary research aims.

1.1.2 Research Questions

RQ1. How, and to what extent, does the use of transformer-based ontology-aware embedding methods (HiT, OnT) affect single and multiple target retrieval performance, and how does their effectiveness compare to lexical (TF-IDF, BM-25) and contextual (SBERT) baselines for simple $\mathcal{EL}$ ontologies?

RQ2. How does the quality of retrieved results vary when different scoring functions are applied to the ranking of embeddings; specifically, does the application of a depth biased *subsumption score* improve knowledge retrieval for OnT and HiT during multiple target retrieval?

RQ3. What is the relationship between ontology embedding-based knowledge retrieval with concept verbalisation and downstream MCQA accuracy for biomedical question answering?

RQ4. (Ablation) How do different ontology encoders trained across similar domains of interest (with distinct signatures) compare to domain-specific fine-tuned encoders?

RQ5. (Exploratory) Is there utility in adopting mixed model spaces (embeddings with distinct geometries) and ranking results via distance across their product manifold?

1.1.3 Objectives

1. Review related research and associated literature on language models, information retrieval, knowledge representation and reasoning, embedding models and ontology embeddings.

2. Define a methodological approach consisting of a set of methods for use in dataset construction and experimental design to evaluate the effectiveness of transformer-based ontology encoders compared to strong lexical and contextual baselines.

3. Obtain and prepare the necessary seed datasets (SNOMED CT, and a biomedical RAG benchmark–MIRAGE) for use in evaluation dataset construction and encoder fine-tuning.

4. Construct and annotate the necessary evaluation data that consists of relevant biomedical entity mentions (extracted from MIRAGE) and their associated gold-label reference targets (class IRIs in SNOMED CT).

5. Implement a modular retrieval framework supporting the use and configuration of BoW-based and transformer-based embedding methods and multiple scoring functions for experimental implementation and evaluation procedures.

6. Prepare a set of embedding models for evaluation using both public pre-trained model checkpoints and domain-specific (SNOMED CT) fine-tuned encoders.

7. Describe, implement and execute a standardised evaluation framework for single target knowledge retrieval measuring: MRR, H@k, median rank (Med), mean rank (MR), R@K; and multiple target subsumption retrieval with metrics mAP, nDCG@K and PR-AUC (interpolated, macro-averaged).

8. Design and implement a standardised RAG pipeline for biomedical MCQA with prompt enrichment capabilities supporting the existing knowledge retrieval framework.

9. Evaluate the extent to which retrieval performance does or does not correlate with downstream biomedical MCQA accuracy using the implemented RAG pipeline.

10. Prepare an exploratory set of mixed model-based encoders for observing potential utility associated with this approach.

1.2 Report structure

The remainder of this report is structured as follows. Section §2 provides an overview of the necessary background and preliminary information that informs the study. Section §3 introduces the methodology, outlining the problem definition, tasks, methods and system implementation. Section §4 outlines dataset construction processes and related experimental design. Section §5 presents the results with associated discussions. Finally, Section §6 concludes the report, summarises the contributions, notes limitations and highlights directions for future work.

2 Preliminaries

This section introduces the prerequisite concepts that the thesis builds upon. It is not meant to provide an exhaustive literature review; rather, this section supplements the introduction, helping to position the work and guide the reader. We introduce ontology and its role within computer science, shortly followed by encoder-only embedding models. Finally, the two ontology encoders used throughout are discussed: Hierarchy Transformers and Ontology Transformers.

2.1 Ontology & Semantic Web Technologies

Broadly speaking, ontology is the philosophical study of being–what entities exist and how they relate to one another. The study of ontology and category systems (taxonomies) dates back to the Ancient Greeks, most notably Aristotle's Categories and syllogistic logic. In computer science and Semantic Web technologies, an ontology is defined as a formal, explicit and shared specification of a conceptualisation for a given domain of discourse [28], often axiomatised in description logic-based (DL) languages [29]. Similarly, a DL knowledge base comprises a set of terminological (TBox; concepts/classes), assertional (ABox; instances/individuals) and role (RBox; properties/roles) axioms[1]. Examples of early ontology representation languages included OIL and DAML+OIL, whereas the most widely adopted standard today is the Web Ontology Language (OWL, 2004, 2009) [30], or OWL 2 (2012) [31], which provides formal, machine-interpretable representations for ontologies.

2.1.1 Web Ontology Language (OWL)

As a means to extend the W3C standard for Resource Description Framework (RDF) and its associated schemata (RDFS), OWL allows for additional expressiveness in the form of aforementioned axioms, and consists of a variety of syntaxes[2] [32]. To provide some intuition as to how OWL extends RDF(S), consider the following triples (presented graphically in Fig. 1):

[1] We note that the separation of TBox, ABox and RBox is a conceptual partitioning, rather than strictly logical **baader2017introduction.**

[2] Examples use generic triple notation for RDF/S and DL for OWL.

(Pancreas,	subClassOf,	BodyPart)
(FingerBone,	subClassOf,	BodyPart)
(Diabetes,	subClassOf,	Condition)
(FingerFracture,	subClassOf,	Condition)
(targets,	domain,	Condition)
(targets,	range,	BodyPart)

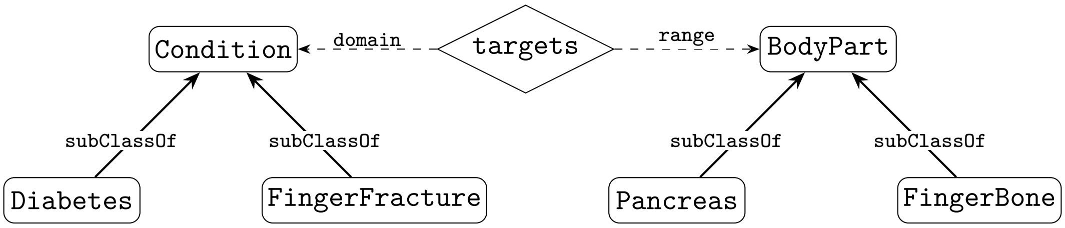


Fig. 1. An illustrative view of a toy medical fragment in RDF(S), where solid arrows model the subClassOf relation between classes and dashed arrows show the schema (object property targets, global domain = Condition and range = BodyPart). Note that in RDF(S) these domain/range statements induce global typing rather than local constraints, they cannot express class restrictions such as the OWL 2 EL existential axioms $\mathsf{Diabetes} \sqsubseteq \exists \mathsf{targets.Pancreas}$ or $\mathsf{FingerFracture} \sqsubseteq \exists \mathsf{targets.FingerBone}$ as is discussed in the text.

The problem we encounter here is that some conditions are defined by the specific body part they target. For instance, a finger fracture targets the finger bone (or some subclass thereof) and should not be interpreted as affecting any other body part (such as pancreas). However, in RDF(S), it is not possible to express existential (`some`), universal (`only`), or disjointness restrictions; for that, we require a more expressive semantics. As this work focuses on SNOMED CT and the EL family of description logics, we provide a brief example of how OWL 2 EL [14] may be used, along with intuition as to how simple description logics can be read. Given that $\sqsubseteq$ and $\sqcap$ are to be read 'is a' (i.e. subClassOf, 'subsumed by') and 'and' (conjunction), allow us to reconsider the previous example, and specify:

$$\mathsf{Diabetes} \sqsubseteq \mathsf{Condition} \sqcap \exists\, \mathsf{targets.Pancreas} \tag{T1}$$

$$\mathsf{FingerFracture} \sqsubseteq \mathsf{Condition} \sqcap \exists\, \mathsf{targets.FingerBone} \tag{T2}$$

The above TBox axioms exemplify existential restriction and state that 'every instance of Diabetes is a condition and targets some pancreas', whereas 'every instance of FingerFracture is a condition and targets some finger bone'. In contrast, RDF(S) can only state the generic fact that "conditions target body parts". In addition, a reasoner, such as ELK [33], can infer additional classifications (e.g. class membership) through entailment. For example, if we define

$$\mathsf{PancreaticCondition} \equiv \exists \mathsf{targets}.\mathsf{Pancreas} \tag{T3}$$

then it follows that $\mathsf{Diabetes} \sqsubseteq \mathsf{PancreaticCondition}$. This provides the necessary groundwork to clarify the difference between the stated view (axioms in themselves) versus the inferred view (the restructured ontology according to the logical consequence of its set of axioms) [34]. Following from the inferred view, the entailed subclass relation $\sqsubseteq$ forms a quasi-ordering [35] over the set of named classes (a transitively closed graph; or polyhierarchy–where classes can have multiple parents, e.g. $\mathsf{Diabetes} \sqsubseteq \mathsf{Condition}$ and $\mathsf{Diabetes} \sqsubseteq \mathsf{PancreaticCondition}$). Though the closure adds direct edges between all ancestor-descendant pairs (eliminating the notion of path length, since set inclusion is transitive). To restore a meaningful (hop-based) distance metric, the transitive reduction can be applied to the entailed $\sqsubseteq$ relation (after quotienting by equivalence), resulting in the Hasse diagram [36], as visualised in Fig. 2.

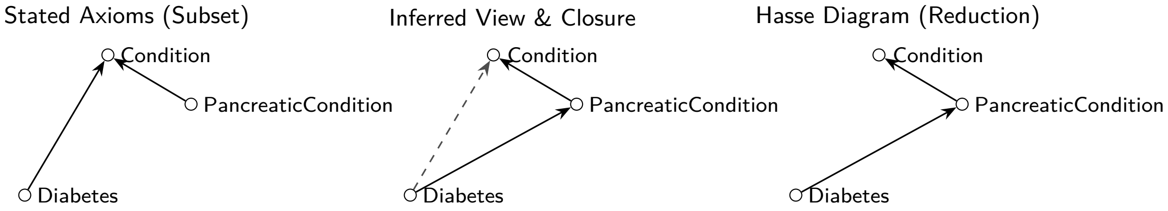


Fig. 2. Toy ontology fragment, simplified for illustrative purposes; i.e. the axiom $\mathsf{PancreaticCondition} \sqsubseteq \mathsf{Condition}$ is shown directly here, although in practice it would be entailed by a domain axiom. The stated axioms show a subset of the previously outlined example ontology (diabetes and conditions); each subdiagram shows how the subgraph changes from the stated view, to the inferred transitive closure, to the Hasse diagram (transitive reduction).

The stated view (or an $\mathcal{EL}$-normalised form thereof [27]) is used to train embedding models (discussed below), whereas the preliminaries surrounding graph construction are foundational to the task description (§3) and dataset construction (§4).

2.2 Encoder-only Language Models

Most contemporary embedding methods leverage either specialised graph-based algorithms, as in [24], [37], or neural representation learning through encoder-only language models [26], [27]. This subsection briefly reviews encoder-only transformer architectures used to directly support our work, with particular focus on BERT and its variants. Additional context on language models more broadly is provided under Appendix. A.

BERT & Cross-Encoders Bidirectional Encoder Representations from Transformers (BERT) [38] is one such example of an encoder-only model, which is trained using a masked-language modelling objective, i.e. predicting a hidden word in a sentence via a cloze statement based on its bidirectional context. Initially, BERT was introduced with a cross-encoder for question-answering and semantic textual similarity (STS) tasks. This means that both question and answer, sentence and document, or sentence and sentence are passed together to BERT (with special `[CLS]` and `[SEP]` tokens to guide the model to identify the relevant text) before reaching a classification head and outputting token-level predictions or a scalar relevance score (logit). While the approach yields high accuracy, it is often computationally infeasible for large-scale retrieval as each new pair of inputs requires a separate forward pass through BERT.

SBERT, Bi-Encoders & Embeddings An alternative to implementing BERT as a cross-encoder is to adopt a bi-encoder pattern, introduced as Sentence-BERT (SBERT) [39]. The bi-encoder's architecture is split into a Siamese configuration (with shared weights), as is shown in Fig. 3. Training and fine-tuning is conducted on tasks such as STS and natural language inference (NLI) using contrastive and regression-based objectives. Notably, this approach separates the encoder from the scoring mechanism, allowing for pre-computation of embeddings (§3.2), enabling an efficient approach to scoring at query-time compared to a cross-encoder based implementation. Efficient computation against a pre-computed embedding store is often required for applications such as web-based search and RAG, and importantly, is harnessed in recent ontology embedding-based implementations such as [26], [27].

$$\mathrm{Sim}(\mathrm{A},\mathrm{B}) = \cos(\theta) = \frac{\mathrm{A} \cdot \mathrm{B}}{\|\mathrm{A}\| \times \|\mathrm{B}\|} \tag{1}$$

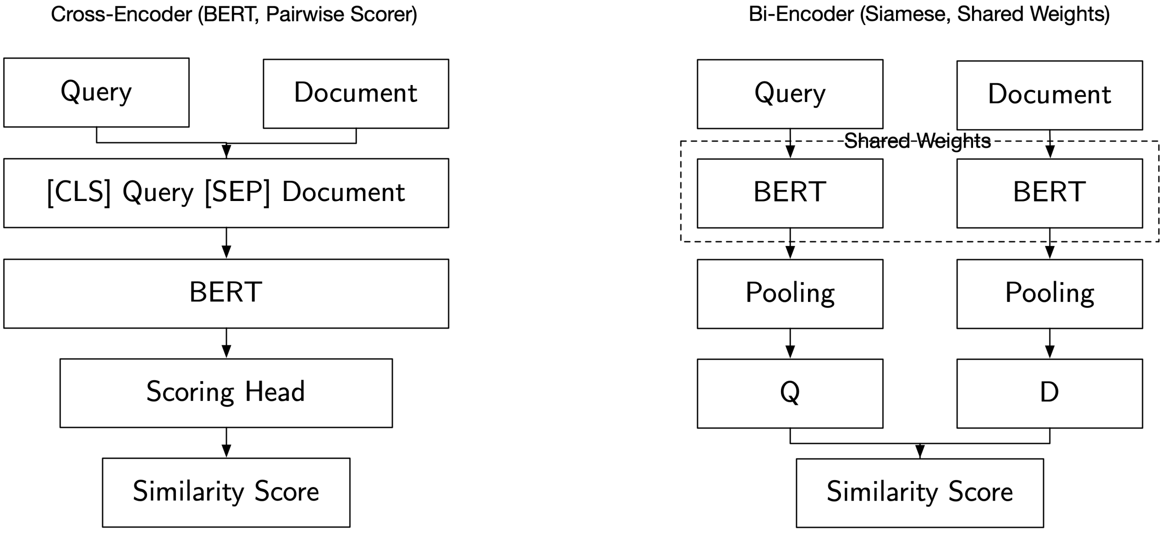


Fig. 3. The left hand-side shows a cross-encoder that concatenates *Query* and *Document* and encodes them jointly with BERT. It computes a pairwise score via a small MLP (Scoring Head). Whereas the right-hand side shows a bi-encoder that encodes *Query* and *Document* **separately** with shared BERT weights. It pools to fixed-size vectors **Q** and **D** and computes a similarity score (e.g. Cosine Similarity).

Sentence-BERT is foundational to both the hierarchy transformer and ontology transformer framework, since both approaches build upon SBERT, while modifying the associated loss function, thereby changing the training objective, as is subsequently discussed.

2.3 Hierarchy & Ontology Transformer

The ontology embedding methods upon which this work is based, HiT [26] and OnT [27], combine textual and structural properties of ontologies during PLM re-training to learn a mapping that can effectively model hierarchical relationships, and in the case of OnT, arbitrarily complex $\mathcal{EL}$ normalised concepts (Eq. (5)). The resulting embedding space should seem familiar, as it situates representations within the Poincaré ball model, similarly to [24], and is designed so that distances reflect some measure of both linguistic similarity and formal knowledge structure.

Hyperbolic Space A Riemannian manifold $\mathcal{M}$ [40], of dimension d is defined as a smooth (differentiable) manifold equipped with a Riemannian metric tensor g, such that the manifold is represented by $(\mathcal{M}, g)$ [24], [41]. For any point $x \in \mathcal{M}$, there exists a local neighbourhood whose geometry resembles Euclidean geometry[3]. Hyperbolic space $\mathbb{H}^n$ is a Riemannian manifold with a constant sectional curvature $-\kappa$, which can be represented in the Poincaré ball model whose points lie within the open ball [26], [27], [42], given by:

[3] Formally, we say that $\mathcal{M}$ is locally diffeomorphic to $\mathbb{R}^d$ [40, p.39]

$$B^n_\kappa = \{\, x \in \mathbb{R}^n : \|x\| < r \,\}, \qquad r = \frac{1}{\sqrt{\kappa}}, \tag{2}$$

where r is the radius of the ball[4]. The Poincaré metric g_κ induces the geodesic distance function d_κ between any two points $x, y \in B^n_\kappa$ (applied for scoring in §3):

$$d_\kappa(x, y) = \frac{1}{\sqrt{\kappa}} \cdot \text{arcosh}\left(1 + \frac{2\kappa\|x - y\|^2}{\big(1 - \kappa\|x\|^2\big) \cdot \big(1 - \kappa\|y\|^2\big)}\right), \qquad \|x\|, \|y\| < \frac{1}{\sqrt{\kappa}} \tag{3}$$

As $\|x\|$ and $\|y\|$ approach the boundary of the ball (norm $\to \frac{1}{\sqrt{\kappa}}$), distances diverge even if the Euclidean norm difference $\|x - y\|$ is not itself significant, meaning that points situated near the boundary can represent more specific nodes, elements or concepts (since their hyperbolic separation becomes large). This is in contrast to points situated toward the center, that represent more generic concepts.

Hierarchy Transformer (HiT) The Hierarchy Transformer (HiT) [26] adopts a bi-encoder model architecture, building on SBERT, allowing for PLM re-training where output embeddings are situated within B^n_κ. The re-training step employs two contrastive loss functions: hyperbolic clustering loss (clusters and distances related and unrelated entities, respectively) and hyperbolic centripetal loss (situates high-level concepts closer to the origin). The total loss is simply the linear combination of the pair. To estimate the likelihood that some subsumption relation holds, the framework adopts the following **subsumption score**, applied for multiple target retrieval (§3):

$$score_{sub}(u, v) = -\big(d_\kappa(\boldsymbol{x}_u, \boldsymbol{x}_v) + \lambda(\|\boldsymbol{x}_v\|_\kappa - \|\boldsymbol{x}_u\|_\kappa)\big), \tag{4}$$

where u is the prospective child, v represents candidate parents, λ provides a heuristic centripetal weighting (or, **depth bias**) and $\| \cdot \|_\kappa$ denotes the hyperbolic norm with curvature κ.

Ontology Transformer (OnT) The Ontology Transformer (OnT) [27] extends HiT to support the richer structures of $\mathcal{EL}$ ontologies. In addition to modelling taxonomic hierarchies, OnT incorporates role embeddings and verbalisations of $\mathcal{EL}$-normalised concepts (Eq. (5)), allowing it to capture existential restrictions and compound class expressions through general concept inclusion (GCI). This

[4] The HiT/OnT library-based instantiations of geoopt's `PoincareBall` set the curvature parameter $c = \frac{1}{d}$ (where d is the embedding dimension of the encoder module), thus $r = \sqrt{d}$. This ensures the encoder's tanh-based activation outputs, which fall within the hypercube $[-1, 1]^d$, fit within the ball; see `https://github.com/KRR-Oxford/HierarchyTransformers/blob/main/src/hierarchy_transformers/models/hierarchy_transformer/hyperbolic.py`

approach builds on the previously outlined hierarchical loss functions by accommodating additional loss functions for role embeddings and conjunctive axioms. Ultimately, the OnT model is capable of embedding both parent–child relations as well as more complex constructs present in OWL 2 EL ontologies, allowing for ontology-aware embeddings that unify lexical, hierarchical, and approximate logical signals.

$$A \sqsubseteq B, \quad A_1 \sqcap A_2 \sqsubseteq B, \quad A \sqsubseteq \exists r.B, \quad \exists r.B \sqsubseteq A \tag{5}$$

3 Methodology

3.1 Problem Definition

Notation Let $\mathcal{O}$ be an OWL 2 ontology (SNOMED CT) with a set of named classes N_C, identified by IRIs. For $C, D \in N_C$, the subsumption relation is written $C \sqsubseteq D$ iff $\mathcal{O} \models C \sqsubseteq D$. The reflexive ancestor set $Anc(C)$ includes both C and all entailed ancestors, written:

$$Anc(C) = \{D \in N_C \mid C \sqsubseteq D\} \tag{6}$$

Given a distance bound $d \in \mathbb{N} \cup \{\infty\}$, let

$$Anc_{\leq d}(C) = \{D \in Anc(C) \mid dist(C, D) \leq d\} \tag{7}$$

where $dist(C, D)$ is the length (in hops) of the shortest path from C to D in the transitive reduction of the entailed $\sqsubseteq$ relation over N_C after quotienting by equivalence, where $dist(C, D) = 0$ iff $C \equiv D$ and is undefined for $D \notin Anc(C)$.

Let Σ^* be the set of strings and Q be the set of queries. Each query $q \in Q$ consists of a natural-language mention $q_s \in \Sigma^*$ and its gold target class $C^\star(q) \in N_C$. If $\mathcal{L}$ is the normalised set of preferred terms (PTs) in $\mathcal{O}$, then q_s is out-of-vocabulary (OOV) when $norm(q_s) \notin \mathcal{L}$, where $norm(\cdot)$ removes parenthesised semantic tags (e.g. `(morphologic abnormality)`), converts to lowercase, collapses whitespace, and applies weak lemmatisation. The set of gold standard reference classes for any $q \in Q$ is defined:

$$Rel(q) = \begin{cases} \{\, C^\star(q) \,\} & \text{(single-target)} \\ Anc_{\leq d}\big(C^\star(q)\big) & \text{(multi-target)} \end{cases} \tag{8}$$

A method X returns a ranked list $Ret_X(q) = (c_1, c_2, \ldots, c_{|N_C|})$ over N_C, and each method defines a score $score_X(q, C)$, where $Ret_X(q)$ orders N_C by $score_X(q, C)$ (§3.2.2).

Knowledge Retrieval Given an OOV query string q_s, then an effective knowledge retrieval method ranks all $D \in Anc(C^\star)$ above all $D \notin Anc(C^\star)$. Knowledge retrieval is conducted under two regimes: *single-target*, where a single class is considered relevant, and *multi-target*, where any valid superclass under entailment is considered relevant.

Biomedical MCQA with Knowledge Retrieval-based RAG Given a biomedical question for a RAG system, identify relevant mentions contained within the question text; for each mention, retrieve a set of relevant classes to provide additional context during answer generation.

3.2 Embedding & Retrieval

Approach Overview For both single-target and multi-target retrieval, lexical methods (TF–IDF, BM25) [18], contextual embedding (SBERT) [20], [39] and ontology embedding methods (OnT, HiT) [26], [27] are employed. First, each method is used to produce an on-disk index or embedding store (§3.2.1). These consist of each named class's natural-language representation, given by $t(C)$. During retrieval, a query mention q_s is embedded within each model's native embedding space, yielding $\boldsymbol{x}_{q_s}$. Then each embedded class is scored against $\boldsymbol{x}_{q_s}$ using a geometrically appropriate similarity, distance, or scoring function $score_X$ (§3.2.2), yielding a ranking over N_C. The process is visualised schematically in Fig. 4.

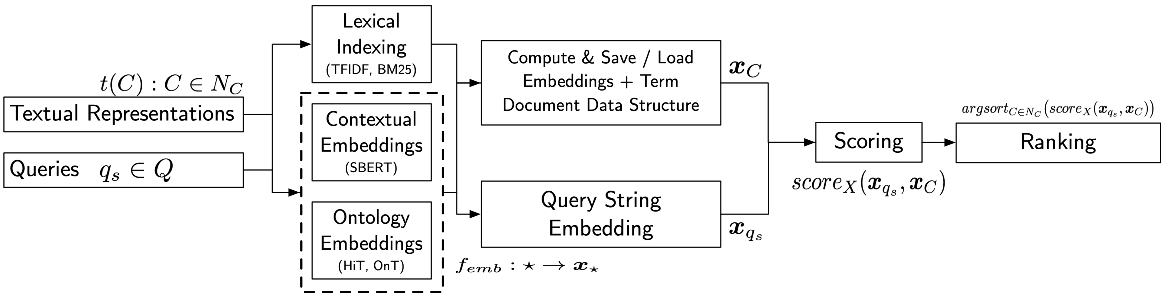

$t(C) : C \in N_C$
Textual Representations
Queries $q_s \in Q$
Lexical Indexing
(TFIDF, BM25)
Contextual Embeddings
(SBERT)
Ontology Embeddings
(HiT, OnT)
$f_{emb} : \star \rightarrow \boldsymbol{x}_\star$
Compute & Save / Load Embeddings + Term Document Data Structure
$\boldsymbol{x}_C$
Query String Embedding
$\boldsymbol{x}_{q_s}$
Scoring
$score_X(\boldsymbol{x}_{q_s}, \boldsymbol{x}_C)$
$argsort_{C \in N_C}(score_X(\boldsymbol{x}_{q_s}, \boldsymbol{x}_C))$
Ranking

Fig. 4. A high-level overview of the embedding and indexing process; class-based textual representations for an ontology are used as input to produce an on-disk index or embedding store. Embeddings are compared at query time to the newly embedded query string, resulting in a full ranking over N_C.

Textual Representation for Classes in $\mathcal{O}$ Each class's textual representation $t(C)$ is given by the normalised PT, which serves as the input during embedding store construction. We note that while natural-language verbalisations [27] are used for model training, they *can* differ substantially from surface forms observed in biomedical QA datasets, and are unlikely to resemble queries used during web-based search.

3.2.1 Embedding Methods

We adopt HiT and OnT as encoders since both are hyperbolic maps, producing embeddings that are situated within the Poincaré Ball where the geometry naturally accommodates hierarchically structured data. Furthermore, their training objectives allow for a depth-biased subsumption score, which is of note since *our multi-target task can be re-framed as a prediction problem*, where we compute the likelihood that some novel concept satisfies subsumption against all existing concepts, then rank classes accordingly (§3.2.2).

Moreover, the use of SBERT as a foundational architecture automatically incorporates transformer-based attention [43] during re-training and embedding. Thus, some degree of contextual disambiguation is likely retained from the base model and continues to be captured during embedding. Finally, OnT directly supports $\mathcal{EL}$ ontologies and is therefore well-aligned to our tasks, since SNOMED CT is authored in $\mathcal{EL}^{++}$.

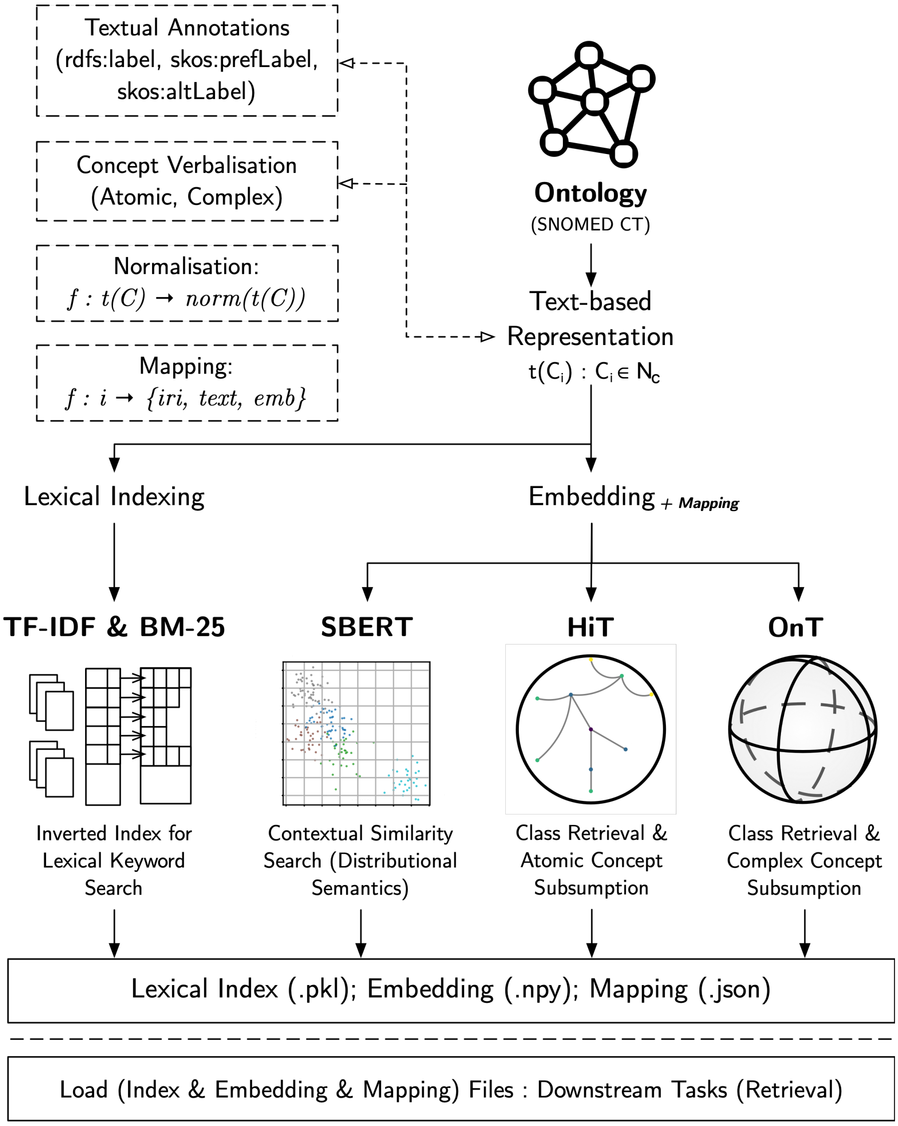

Textual Annotations (rdfs:label, skos:prefLabel, skos:altLabel)
Concept Verbalisation (Atomic, Complex)
Normalisation: $f : t(C) \rightarrow norm(t(C))$
Mapping: $f : i \rightarrow \{iri, text, emb\}$
Ontology
(SNOMED CT)
Text-based Representation
$t(C_i) : C_i \in N_C$
Lexical Indexing
Embedding + Mapping
TF-IDF & BM-25
SBERT
HiT
OnT
Inverted Index for Lexical Keyword Search
Contextual Similarity Search (Distributional Semantics)
Class Retrieval & Atomic Concept Subsumption
Class Retrieval & Complex Concept Subsumption
Lexical Index (.pkl); Embedding (.npy); Mapping (.json)
Load (Index & Embedding & Mapping) Files : Downstream Tasks (Retrieval)

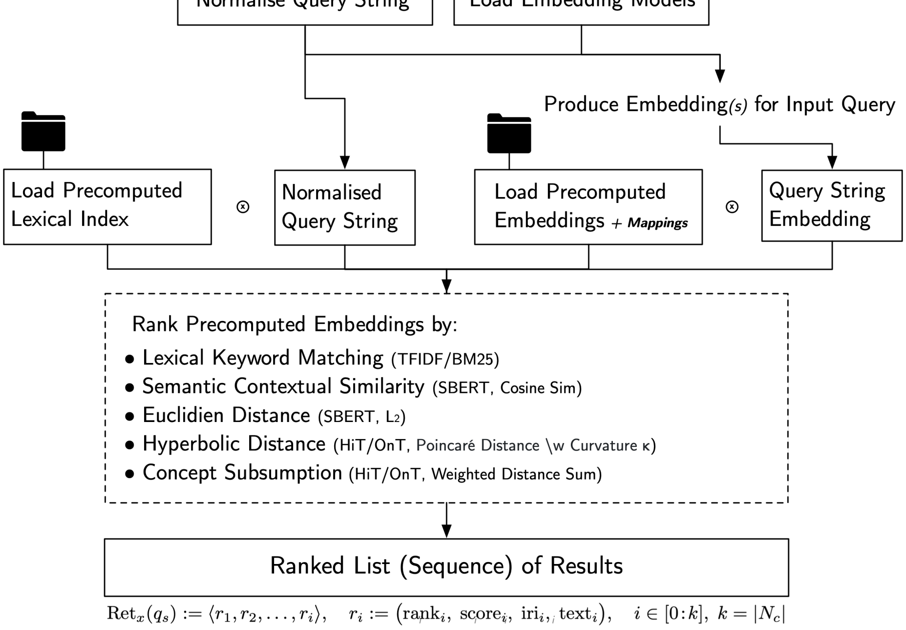
Fig. 5. Components of the embedding methodology.

Alternative Approaches When comparing the selected methods against alternative approaches, we note that pure lexical methods, such as TF–IDF and BM25 (§4), lack semantics and cannot model hierarchical relationships. Additionally, graph embeddings, such as Node2vec [37] and [24], are entirely geometric and ignore textual annotations, making them unsuitable for use. Moreover, approaches that do use annotations, such as OPA2Vec [25] and OWL2Vec* [44], either rely on static word-embeddings that lack contextualisation or operate in Euclidean space, where hierarchical data is not naturally represented.

Conversely, hyperbolic bi-encoders, i.e. HiT [26] and OnT [27], are uniquely positioned in that they capture contextual semantics via transformer-based attention, hierarchical relationships through hyperbolic geometry; and are computationally efficient compared to cross-encoder-based implementations; making them strong candidate methods for use in our case.

3.2.2 Retrieval & Ranking Methods

Each embedding method requires a specific scoring function $score_X(q_s, C)$ that measures the relevance between a query mention q_s and each named class. We note that the most appropriate score function is model-dependent, as the geometry varies across model spaces. As such, the rank ordering is also important, since not all scoring mechanisms are unbounded and monotonic (e.g. cosine similarity is a **similarity score**, whereas distance of any kind is, by definition, a **metric**). Thus, distance functions sort ascending and similarity functions sort descending.


Input Query String
(Textual Representation)
SBERT
HiT
OnT
Normalise Query String
Load Embedding Models
Produce Embedding(s) for Input Query
Load Precomputed Lexical Index
⊗
Normalised Query String
Load Precomputed Embeddings + Mappings
⊗
Query String Embedding
Rank Precomputed Embeddings by:
• Lexical Keyword Matching (TFIDF/BM25)
• Semantic Contextual Similarity (SBERT, Cosine Sim)
• Euclidien Distance (SBERT, L2)
• Hyperbolic Distance (HiT/OnT, Poincaré Distance \w Curvature κ)
• Concept Subsumption (HiT/OnT, Weighted Distance Sum)
Ranked List (Sequence) of Results

$\mathrm{Ret}_x(q_s) := \langle r_1, r_2, \ldots, r_i \rangle, \quad r_i := (\mathrm{rank}_i,\ \mathrm{score}_i,\ \mathrm{iri}_i,\ \mathrm{text}_i), \quad i \in [0\!:\!k],\ k = |N_c|$

Fig. 6. A visual description of retrieval and ranking for lexical (TF–IDF, BM25), contextual (SBERT) and ontology embedding-based (HiT, OnT) methods.

Cosine Similarity For SBERT embeddings in Euclidean space, we use cosine similarity measured between unit-normed vectors. Note that measuring the inner product between any two normalised SBERT embeddings equates to cosine similarity, whilst the L_2 distance corresponds to a monotonic transformation of it, yielding the same ranking as cosine similarity.

$$score_{\mathsf{cos}}(q_s, C) = \frac{\boldsymbol{x}_{q_s} \cdot \boldsymbol{x}_C}{\|\boldsymbol{x}_{q_s}\| \, \|\boldsymbol{x}_C\|} \tag{9}$$

Geodesic Distance We opt to apply nearest–neighbour retrieval with d_κ (§2.3) for single target retrieval, as it naturally preserves the embedding space's geometry. Additionally, since all candidate embeddings are not positioned in similar local neighbourhoods, a global measure of geodesic distance is preferable to other local measures, e.g. measuring inner product across tangent vectors. We avoid applying Euclidean metrics within this space, as they would be prone to distort a meaningful interpretation of distance. Note that d_κ is computed in a vectorised manner (pre-computing $\|x\|^2$ terms and retaining them in memory) to aid in efficient computation.

Subsumption Score Since the multi-target retrieval task aims to rank all valid ancestors $Anc(C^\star)$ ahead of non-ancestors, the depth-biased subsumption score (§2.3) is adopted for use and compared against geodesic distance in HiT and OnT for the multi-target task. Intuitively, the subsumption score's depth bias (centripetal weight) intentionally assigns a preferential weight to the most probable antecedents, prioritising a hierarchical pre-order over local neighbourhood retrieval. Since pure geodesic distance is symmetric and effectively depth-agnostic, we would anticipate an increase in relevant results for subsumption-aware scoring, so long as an appropriate weighting is used for λ; i.e. the value must be carefully balanced between collapsing the score to pure geodesic distance and over-extending up the hierarchy. The subsumption score sorts descending.

3.3 Exploratory Methods

3.3.1 Mixed Model Encoders

Consideration for the use of mixed models is primarily inspired by [41], where the authors postulate that, whilst hyperbolic and spherical representations have been gaining traction recently, most data is not necessarily best represented in such a uniform manner. As such, they suggest adopting a model space with heterogeneous curvature. To consider the structural properties of most real-world ontologies post-classification, the data might resemble a polyhierarchy (a complex graph) rather than a pure hierarchy, lending credence to the observation made by [41]. Thus, we introduce an exploratory framework for adopting such an approach, aligned with RQ5.

For a sequence of manifolds $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$, the Cartesian product of each component manifold defines the product manifold [41], noted $\mathcal{M}_\times = \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_n$. The shortest path between any pair of points $x, y \in \mathcal{M}_\times$ is given by the shortest path as measured across each component manifold.

A pair of embedding models is defined:

$$f_{ont} : \Sigma^* \to B^n_\kappa, \qquad f_{ctx} : \Sigma^* \to \mathbb{R}^d, \tag{10}$$

where f_{ont} is an OnT encoder and f_{ctx} is an SBERT encoder. Each operates over its own distinct model space, producing embeddings $\boldsymbol{x}$ for a given query string q_s:

$$f_{ont}(q_s) = \boldsymbol{x}^{ont}_{q_s} \in B^n_\kappa, \qquad f_{ctx}(q_s) = \boldsymbol{x}^{ctx}_{q_s} \in \mathbb{R}^d \tag{11}$$

As such, the pair:

$$\boldsymbol{x}^\times_{q_s} = (\boldsymbol{x}^{ont}_{q_s}, \boldsymbol{x}^{ctx}_{q_s}) \in \mathcal{M}_\times, \tag{12}$$

is a single point on the heterogeneous product manifold $\mathcal{M}_\times = B^n_\kappa \times \mathbb{R}^d$ [41]. Under the product Riemannian metric [45], [46], the squared distance is the sum of the squared component distances, thus:

$$d_\times\big((x_1, x_2),\, (y_1, y_2)\big) = \sqrt{d_\kappa(x_1, y_1)^2 + \|x_2 - y_2\|^2}, \qquad (x, y) \in \mathcal{M}_\times \tag{13}$$

Additionally, this can be applied over an extended product manifold consisting of any number of disjoint embedding spaces, for any number of encoders and distance metrics, so long as they satisfy the set of metric axioms[5]:

$$d_\times(x_i,\, y_i) = \left[\sum_{i=1}^{n} d_i(x_i, y_i)^2\right]^{\frac{1}{2}}, \qquad \text{with } x = (x_1, \ldots, x_n),\ y = (y_1, \ldots, y_n) \ :\ (x, y) \in \mathcal{M}_\times \tag{14}$$

[5] Non-negativity, identity, symmetry, triangle inequality [47, p.187–p188].

A point worth mentioning is that the previously discussed use of cosine similarity with SBERT does not apply here, as (i) it acts inversely to distance, and (ii) it not a valid metric (breaks triangle inequality[47]). However, it is convenient that L$_2$ distance applied over normalised SBERT embeddings yields a monotone transformation of cosine similarity, making the set of distance metrics outlined below the most immediate choice for adopted use and interpretation:

$$d_{hit} = d_\kappa \ \ (\kappa \leftarrow f_{HiT}), \qquad d_{ont} = d_\kappa \ \ (\kappa \leftarrow f_{OnT}), \qquad d_{ctx} = \|u - v\|_2 \tag{15}$$

While each of these functions increases monotonically with separation, there is a clear caveat, i.e. Euclidean distance is not directly comparable to geodesic distance, motivating the use of a set of free parameters that can be tuned to learn an appropriate distance weighting:

$$d_{\times,\gamma}(x_i, y_i) = \left[\sum_{i=1}^{n} \gamma_i \cdot d_i(x_i, y_i)^2\right]^{\frac{1}{2}}, \qquad \gamma_i > 0, \ \sum_i \gamma_i = 1 \tag{16}$$

Three initialisation schemes are suggested for the distance scaling parameters:

1. **Isotropic initialisation**: $\gamma_i = \frac{1}{n}, \ \forall i = 1, \dots, n.$
2. **Validation tuning**: apply a grid-search over γ to tune on a small held-out dataset.
3. **Joint learning**: treat the set of γ weights as free (trainable) parameters in a downstream task and back-propagate the error (given that distance is differentiable) as an adaptive layer, sitting above each frozen encoder.

While we aim to explore all suggested schemes, isotropic initialisation without any weight updates is the most appropriate baseline, though unlikely to be optimal as it collapses back to an unweighted approach, eliminating utility associated with introduced weightings. However, it does provide a reasonable starting position for tuning or training. Validation tuning is most appropriate for related knowledge retrieval tasks themselves, whereas joint learning would likely be of most interest for downstream tasks, as it would act as a reasonably appropriate means to indicate (through observing the final weights) how influential and effective each component encoder is for that given task (which directly relates this technique to RQ1, RQ3 and RQ5), though it would require additional engineering and data processing efforts.

3.4 System Implementation

3.4.1 High-Level Architecture

The system design and implementation is relatively simple, combining both functional and class-based design (Fig. 20), enabling a modular micro-framework for experimental implementation. Components include:

- **Core Utilities**: Modules for data models and vector math.
- **Retrieval**: The set of extendable retriever classes (Appendix. B.1); includes GPU-accelerated variants.
- **LLM**: A small wrapper around HuggingFace transformers library for prompt rendering and constrained decoding.
- **Orchestration**: Experimental harness that connects datasets, retrievers, selection criteria and LLMs into reproducible experiments with entry points in the `exp-*.py` files. Supplementary notebooks are also provided and used for retrieval experiments.

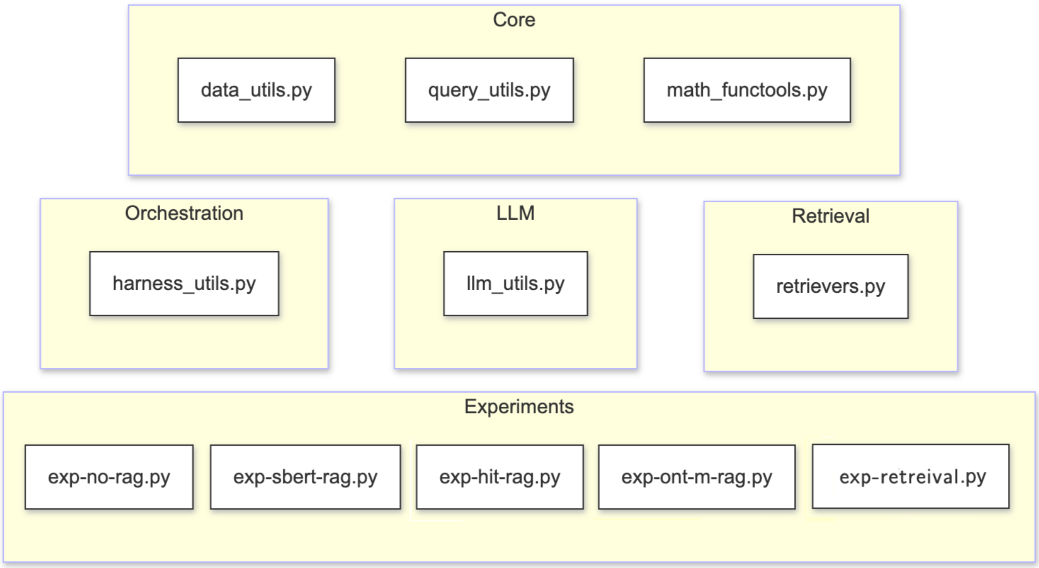


Fig. 7. High level overview of the packaged components.

Separation of Concerns The architectural design implements abstract base classes for retrievers that effectively specify contracts for extended classes, along with a BaseModelRetriever that implements common logic and exposes interfaces for registering models and score functions. This allows retrieval components to be easily extended and swapped within the context of downstream experimental runs (see Appendix. B.1).

3.4.2 Workflows

Offline Embedding Store Each ontology class's textual representation $t(C)$ is embedded once and persisted as a memory-mappable `.npy` matrix. SBERT uses unit-norm Euclidean vectors (i.e. `normalize_embeddings=True`), whereas HiT and OnT produce hyperbolic embeddings, with no normalisation (explicitly specified to avoid constraining the embeddings' norms to 1). Additionally, as embeddings produced with HiT and OnT are parameterised by a model-dependent sectional curvature κ, we ensure this is respected within the `math_functools.py`, and is calculated at run-time. The offline embedding store is produced using a simple procedural script.

Retrieval at Runtime A mention string q_s is encoded into the model's native embedding space and scored against the fixed embedding store matrix using a geometrically appropriate `score_fn` callback (registered by the class-specific implementation, e.g. `HiTRetriever`, `GPUOnTRetriever`, etc.). Candidates are then ranked by score through the application of $argsort_{C \in N_C}$ via a brute-force search over N_C. This ensures fairness across methods by retaining the entire list during each retrieval call. Fig. 8 shows the call sequence for retrieval experiments. Retrieval flows tied to the LLM experiment are presented under Appendix. B.2.

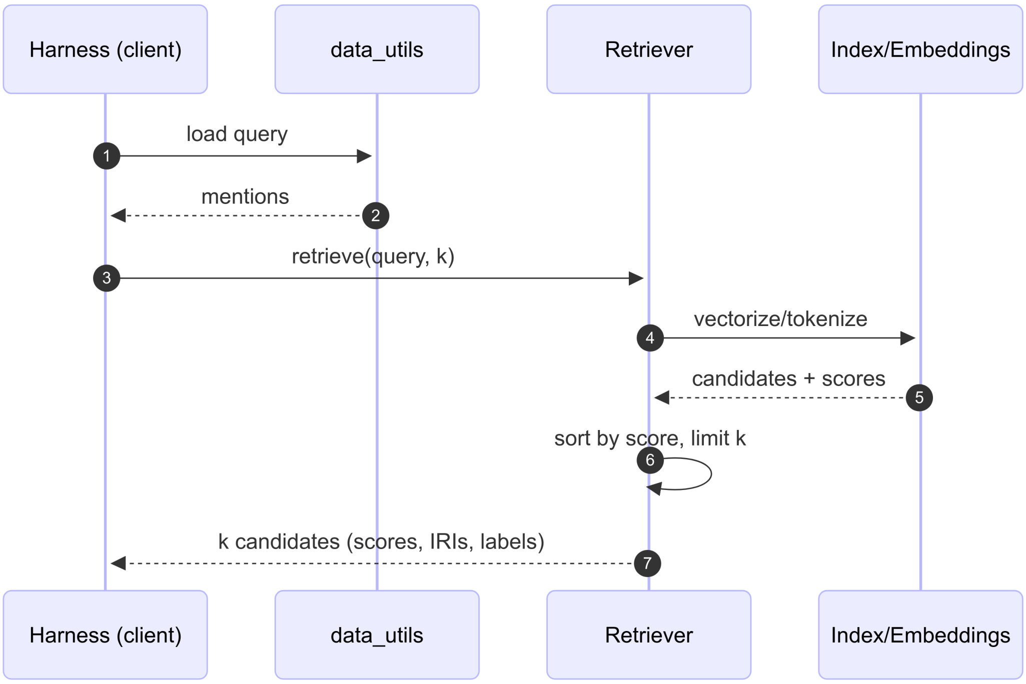


Fig. 8. The call sequence for retriever classes during experiments. The client (or experiment) loads a query and obtains the associated mentions, calls the retriever, then the retriever embeds the query, computes scores against the embedding store, sorts and returns candidates.

3.4.3 Reproducibility

Software

- OS: Ubuntu 24.04 LTS
- Python: 3.12
- CUDA: 12.8
- PyTorch: 2.7.1

Hardware

- CPU & Memory: Intel Sapphire Rapids (48c/96t), 128GB Samsung ECC-RDIMM
- GPUs: 2 × NVIDIA RTX A5000 24GB, 1 × NVIDIA RTX 4000 ADA SFF

Experiments are locally reproducible with Docker and suitable for bare metal or virtualised cloud instances via Makefile. Experiments were executed under a fixed seed value 42, configuration files, checkpoints and additional details are released alongside the codebase at `https://github.com/jonathondilworth/uom-thesis`.

4 Dataset Construction & Experimental Design

4.1 Datasets

We construct evaluation queries to test agaisnt (retrieval tasks, RQ1, RQ2, RQ4, RQ5) and select a benchmark to measure RAG performance (RQ3). SNOMED CT [11] is used in combination with the MIRAGE (Medical Information Retrieval-Augmented Generation Evaluation) benchmark[6] [48]. A high-level overview of this process is detailed in Fig. 9. Additionally, we provide a description of the necessary preprocessing, transformation and annotation steps in the subsections that follow.

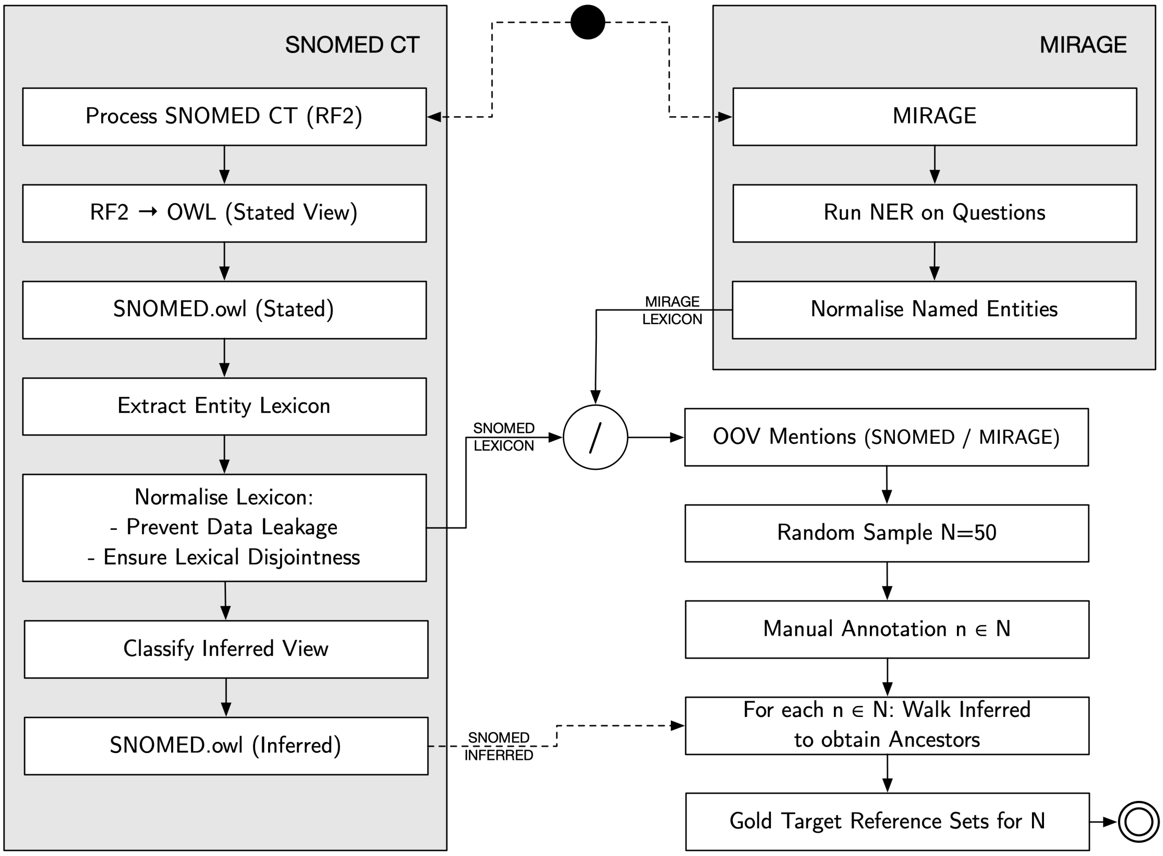


Fig. 9. High-level process overview. The set of OOV mentions is constructed by obtaining the set of all NER mentions across MIRAGE, $\mathcal{E}_{OOV} = \{e \in \mathcal{E} \mid \text{norm}(e) \notin \mathcal{L}\}$. After filtering, the OOV set is obtained with $|\mathcal{E}_{\text{OOV}}| = 3530$ unique strings. From the set of OOV mentions $N = 50$ mentions are sampled at random to form the basis upon which the evaluation query set Q is built. Each query $q \in Q$ is manually linked/annotated to a single target concept (named class) $C^{\star} \in N_C$ via its IRI; and the entailed ancestor set $Anc(C^{\star})$ is obtained.

[6] The SNOMED CT data version used is pinned to the international release (2025-07). The MIRAGE benchmark is publicly available at `https://github.com/Teddy-XiongGZ/MIRAGE/blob/main/benchmark.json`.

4.1.1 SNOMED CT Data Processing

Acquisition & File Conversion SNOMED CT is released under a license in the file format `Release Format 2` (`RF2`). To prepare SNOMED CT for further processing with knowledge representation and reasoning (KRR) software, it must first be interpretable and converted to the `OWL` file format (Fig. 10). This is accomplished using the `IHTSDO/snomed-owl-toolkit` [49].

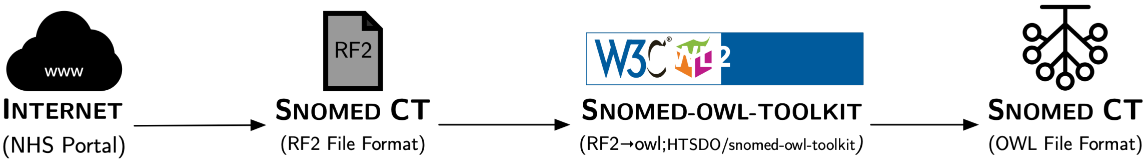


Fig. 10. SNOMED CT file Acquisition and conversion pipeline.

Stated & Inferred Views Both the stated ($C \sqsubseteq_{\text{st}} D$, iff `subClassOf` $(C, D) \in \mathcal{O}$) and the inferred view ($\mathcal{O} \models C \sqsubseteq D$) are obtained for dataset construction. The stated view is used to derive the SNOMED CT *entity lexicon* [26]; in this case, the entity lexicon describes the set of named classes via their IRI and `RDFS:label`. After processing the MIRAGE benchmark and performing manual annotation, the inferred view is used to derive $Anc(C^{\star})$ for multi-target retrieval (Fig. 11).

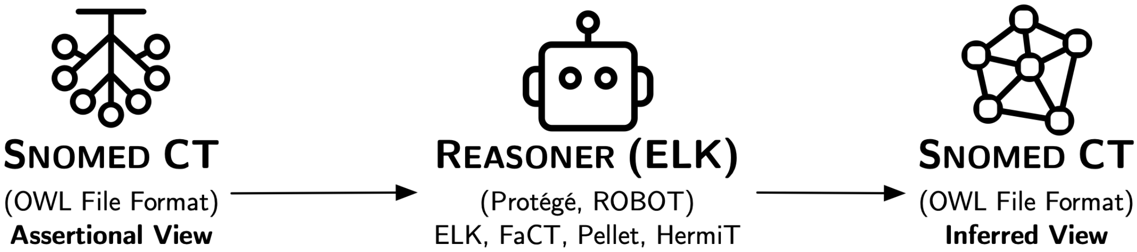
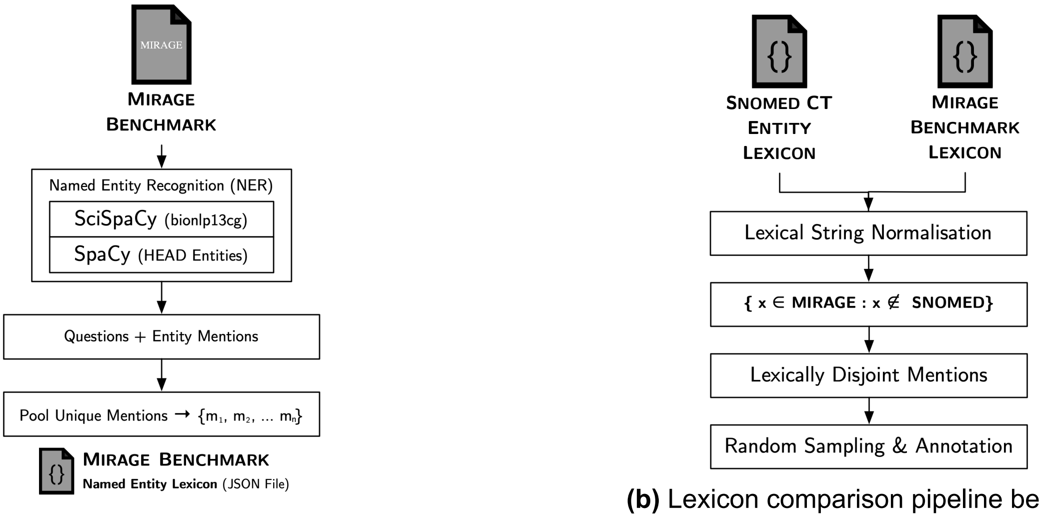


Fig. 11. SNOMED CT reasoning pipeline, converting from the stated view to the inferred view. We leverage ELK with the CLI tool ROBOT [50].

Normalisation of Lexical Surface Forms To prevent data leakage during downstream evaluation, each class `RDFS:label` is modified to remove any indication of high-level concept inclusion (semantic tags) [26]. For example, *'Acute hypertrophy (morphologic abnormality)'* is transformed to *'Acute hypertrophy'*. Additional normalisation steps include reducing each character to lower case, collapsing white space and applying a weak lemmatisation (morphological normalisation) step [18], which reduces plural terms to singular terms. Stemming is intentionally avoided as it risks reducing sibling terms to an equivalent surface form and has the potential to confuse annotation.

4.1.2 MIRAGE Data Processing

The MIRAGE [48] benchmark consists of 7663 biomedical question-answer pairs (answers being in the form of multiple choice options). It is built from five constituent datasets, including: (i) MedQA [51], (ii) MMLU [52], (iii) MedMCQA [53], (iv) PubMedQA [54], and (v) BioASQ [55]. Each question is processed using named entity recognition (NER) to extract a set of mentions (Fig. 12a). Mentions are tested against SNOMED CT for lexical disjointness and are subsequently sampled for annotation (Fig. 12b), i.e. providing its nearest valid parent concept for single target retrieval (RQ1). During NER only the question text is processed to guard against data leakage (introducing concepts from the answers). After annotation is complete, SNOMED CT is algorithmically parsed to obtain associated the entailed ancestor set for multi-target knowledge retrieval (RQ1, RQ2).


MIRAGE
MIRAGE BENCHMARK
Named Entity Recognition (NER)
SciSpaCy (bionlp13cg)
SpaCy (HEAD Entities)
Questions + Entity Mentions
Pool Unique Mentions → {m₁, m₂, ... mₙ}
{}
MIRAGE BENCHMARK
Named Entity Lexicon (JSON File)
{}
SNOMED CT ENTITY LEXICON
{}
MIRAGE BENCHMARK LEXICON
Lexical String Normalisation
{ x ∈ MIRAGE : x ∉ SNOMED}
Lexically Disjoint Mentions
Random Sampling & Annotation

(a) The MIRAGE NER pipeline.

(b) Lexicon comparison pipeline between SNOMED CT and MIRAGE.

Fig. 12. An overview of the MIRAGE data processing pipelines.

Named Entity Recognition & Extraction This step involves the use of `SciSpaCy` [56] for biomedical named entity recognition (NER) with the `en_ner_bionlp13cg_md` model [57], [58]. Additionally, the standard `SpaCy` library [59] is operationalised for identifying HEAD entities. Both sets of entity mentions are then merged (per question) to provide a mention set for each question in MIRAGE. The metadata (dataset identifier, question identifier), question text and mention sets are used to provide the basis for query set construction and are extracted for each question within the original benchmark, used to:

1. Obtain the lexical intersection between the set of MIRAGE entity mentions and the SNOMED CT entity lexicon, indicating the lexical overlap between the two.

2. Derive the set of lexically disjoint surface forms (also referred to as the set of *out-of-vocabulary*

(OOV) terms) by taking the set difference between MIRAGE and SNOMED CT.

Preprocessing MIRAGE for RAG MCQA During downstream biomedical question answering with retrieval augmented generation, prompts are required to include axiom verbalisations [60] for enrichment. To accomplish this, an additional verbalisation step is added to the previously outlined process, where the SNOMED CT graph is walked and a simple concept card (containing class IRI, parent nodes, child nodes and axiom verbalisation) is produced for each node, then saved to `snomed_axioms.js` This allows the biomedical RAG MCQA framework to quickly access any associated class information, eliminating the need for manual search of the ontology for every answer.

4.1.3 Dataset Construction & Annotation

The lexically disjoint set of entity mentions between SNOMED CT and MIRAGE provides the basis for a manual annotation process. This process involves reviewing each OOV mention, confirming its lexical disjointness and then searching SNOMED CT for potential insertion opportunities, i.e. where $\mathrm{M}_{\mathrm{oov}} \sqsubseteq D$ is valid. If no valid insertions are appropriate, the mention is disregarded and resampled. Insertions are not actually performed, rather the validity is only important insofar as the construction of the evaluation queries is concerned, since making the insertion and subsequently training the encoders would undermine any experimental results. Finally, datasets are constructed by noting the linkage between each mention and its corresponding SNOMED CT class IRI, and then algorithmically walking the transitive reduction of the entailed $\sqsubseteq$ graph to obtain the set of inferred ancestors.

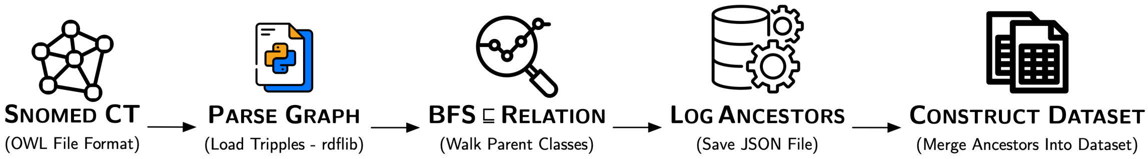


Fig. 13. An overview of the data pipeline for processing ancestor classes.

All $N = 50$ OOV queries are reserved for evaluation, with hyperparameters (BM25 k_1, b; centripetal weight λ) being tuned on a small set of 20 non-overlapping linked mentions.

4.1.4 Graded Relevance

For each annotated class, the multiple target gold set is constructed using the entailed $\sqsubseteq$ relation over named classes. To construct relevance for nDCG (§ 4.4), a grade $g_q : N_C \to \{1, 2, \dots\}$ is attached to each annotation according to the relative depth[7] from the most relevant target class during multi-target retrieval. Graded relevance is reported through nDCG [18], [61], [62], where exponential decay is applied; whereas mAP [18], [62] uses binary relevance only.

4.1.5 Licensing & Release

The list of query strings, their normalised forms, target reference IRIs, and the ancestor sets $Rel(q)\ \forall q \in Q$ are released, along with scripts used for their preparation. The initial SNOMED data (full release) is **not** redistributed[8], and data consumers must rebuild from their own licensed copies.

4.2 Experimental Baselines

To compare the performance of ontology-aware embedding-based methods (encoders), three baseline methods are selected that span both lexical and contextual domains. All baseline measures operate over the exact same candidate sets in N_C, with the usual normalisation function applied (§3.1, §4.1.1).

Lexical Baselines TF–IDF and BM25 [18] are used for lexical keyword matching, with the intended use of assessing performance deltas for embedding methods, ensuring any improvements cannot simply be attributed to trivial lexical overlap. Implementation details are provided under Appendix. E.1.

Contextual Baseline SBERT is included within the study as it provides a robust[9] contextual baseline against which experimental results can be compared. We anticipate ontology embedding-based approaches that build on SBERT ought to outperform their re-trained base model. Implementation details are provided under Appendix. E.2.

4.3 Model Variants

HiT and OnT are fine-tuned on the most recent release of SNOMED CT (2025-07). Model variants used during evaluation are outlined in the tables provided below (Table. 1, Table. 2).

[7] In practice, we measure the relative depth to each valid ancestor, record this as the ascent height, then backtrack through the graph, taking the reversed ascent height as the relevance; see the `retrieval`$_notebook.ipynbinthelinkedGitHubforadditionaldetail.$

[8] SNOMED CT is licensed through NHS TRUD, however, the build scripts provided under `https://github.com/jonathondilworth/uom-thesis` will fallback to a publicly available version hosted on Zenodo if no `NHS_API_KEY` is provided within `.env`; see the repositories `README.md` for details.

[9] SBERT is robust against OOV words due to subword tokenisation, a principal component of our tasks.

Table 1. Model Variants in use During Evaluation

Model	Ontology VAR	Public Checkpoint (HF)	batch size	Full/Mini	Model Label
SBERT	-	✓	-	-	SBERT
HiT	SNOMED	**X**	32	Full (F)	HiT SNOMED (F)
OnT	GALEN	✓	-	-	OnT GALEN
OnT	ANATOMY	✓	-	-	OnT ANATOMY
OnT	GO	✓	-	-	OnT GO
OnT	SNOMED	**X**	32	Full (F)	OnT SNOMED (F)
OnT	SNOMED	**X**	32	Mini (M)	OnT SNOMED (M-32)
OnT	SNOMED	**X**	64	Mini (M)	OnT SNOMED (M-64)
OnT	SNOMED	**X**	128	Mini (M)	OnT SNOMED (M-128)

The table shows the set of models used during evaluation; *Ontology VAR* includes the ontology on which the encoder was re-trained; public checkpoint (HF) indicates whether the model is loaded from HF, or locally trained. Mini (M) models are trained on a subset of SNOMED CT (with semantic branches: Body Structure, Finding, Event and Procedure), rather than on the full set, with varying batch sizes, see Appendix. F.

Table 2. Mixed Models Variants in use During Evaluation

Model	Mixture	Model_1	Model_2	Model_3	Manifold
Mix-1	✓	OnT SNO(F)	HiT SNO(F)	SBERT	$\mathbb{R} \times \mathbb{H} \times \mathbb{H}$
Mix-2	✓	OnT SNO(M-32)	OnT SNO(M-128)	SBERT	$\mathbb{R} \times \mathbb{H} \times \mathbb{H}$

4.4 Evaluation Metrics

Several methods are applied to each set of experimental results for evaluation and system testing. Employed methods measure performance (in terms of effectiveness) and provide some indication of transferability for ablations; which is in line with the stated research aims (§ 1.1).

Two sets of metrics are applied during the observation of single target and multiple target retrieval. For single target retrieval, mean reciprocal rank (MRR) [18], [63] and hit rate (H@K) [18] are used as the primary evaluation metrics; median rank (Med) and mean rank (MR) are reported secondary. For multiple target retrieval, mean average precision (mAP) [18], [62] and normalised discounted cumulative gain (nDCG) are employed as primary, whereas (trapezoidal) area under the precision-recall curve (PR-AUC) and recall (@100) are employed as secondary checks. Additionally, precision-recall curves are plotted for visual interpretation. A brief discussion of these metrics, including justification for their selection, is outlined in the subsequent subsections.

Mean reciprocal Rank MRR (Eq. (17)–(18)) preferentially weights highly ranked (matched) candidates, allowing for an emphasis on early precision (noted as particularly important for preparing retrieval systems for downstream biomedical MCQA with RAG). However, it ignores the tail of any retrieved list (past the first hit), making it less appropriate for use in multi-target scenarios.

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q} \tag{17}$$

$$\text{rank}_q = \text{min}\{k \geq 1 \mid C_k \in \text{Rel}(q)\} \tag{18}$$

Hit Rate, Hits-at-K Hits-at-K (H@K) (Eq. (19)) complements MRR with an operational success rate at ranks $k = \{1, 3, 5\}$. These thresholds are considered meaningful since they place a similar emphasis on obtaining the single correct match early, and they reflect the thresholds by which the downstream RAG application is tested, i.e. ($k_{bounds} = \{1, 5\}$).

$$\text{Hits@}k = \frac{1}{|Q|} \sum_{q \in Q} 1 \left[\exists c \in Ret_X(q)_k \cap Rel(q) \right], \qquad k = \{1, 3, 5\} \tag{19}$$

Secondary Metrics Median (first-hit) rank (Med) is employed as a robust descriptive statistic (as it is largely unaffected by outliers), providing some indication as to the retrieval effectiveness. Whereas, mean rank (MR) is reported in a supplemental fashion as **it is** sensitive to outliers and truncation. These metrics aid in keeping the single-target results interpretable and aligned with project aims (§1.1).

Mean Average Precision The multi-target setting provides an indication of the precision measured at each relevant occurrence in the retrieved set. Measuring mAP (Eq. (20)–(22)) over the entire set of retrieved classes (for each query), therefore, provides a more indicative effectiveness metric when compared to MRR, which is only useful to ascertain performance at the first relevant occurrence.

$$mAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q_s) \tag{20}$$

$$AP(q_s) = \frac{1}{|Rel(q)|} \sum_{k=1}^{|N_C|} P@k \cdot relevance(k), \qquad P@k = \frac{|Rel(q)_k \cap Ret_X(q)_k|}{|Ret_X(q)_k|} \tag{21}$$

$$relevance(k) = \begin{cases} 1, & \text{if k is relevant} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

Normalised Discounted Cumulative Gain Unlike mAP–which considers all relevant classes as positive hits–graded nDCG@K ($k = 10$) treats each relevant result within the retrieved set according to its assigned relevance value. That is, ancestors which are proximally further from the target entity are less informative (broader concepts that subsume the most immediately relevant concept). As such, nDCG should provide some indication as to how accurately the embedding space captures hierarchical relationships between sets of concepts and ought to reflect how well each retriever is ranking the results (with the ideal ordering scoring highest).

For multi-target evaluation $g_q : N_C \to \{0, 1, \dots\}$ is attached with:

$$g_q(D) \ = \begin{cases} 0 & D \equiv C^\star \\ 1 + \text{min dist}(C^\star, D) & D \in Anc(C^\star) \end{cases} \tag{23}$$

As such, nDCG gain is defined to *decrease* with distance, e.g. $gain(D) = 2^{-g_q(D)}$. Then:

$$DCG@k(q) = \sum_{i=1}^{k} \frac{gain(c_i)}{log_2(i+1)}, \qquad nDCG@k(q) = \frac{DCG@k(q)}{iDCG@k(q)} \tag{24}$$

Interpolated PR Curve & PR-AUC Retrieval effectiveness across multiple embedding models and target concepts may also be interpreted in terms of binary classification, i.e. items are either relevant or irrelevant. As such, plotting the associated precision–recall `curves` [62], [64] and noting the interpolated PR-AUC (trapezoidal rule [65]) provides at least one additional, interpretable view of system performance. The upper bound of precision can be represented alongside recall as a series of interpolated points, producing plots where intersections between curves highlight where one system outperforms another. Moreover, if one curve lies entirely within another, the visualisation provides an indication that one system consistently outperforms another (for precision and, or recall).

4.5 RAG Evaluation

Evaluation is conducted on MIRAGE, reporting per-subdataset accuracy (PubMedQA, BioASQ, MMLU, MedQA, MedMCQA) and a macro average across them. To avoid information leakage, only the question text is used for retrieval; the following RAG configurations are tested (Table. 3), implementation details and justification for model choices are provided under Appendix. C.

Table 3. RAG and No-RAG regimes with retriever settings as used during evaluation.

Regime	Retriever	k	d	Retrieval Method
No-RAG	–	–	–	–
RAG	SBERT	1	5	Cosine Similarity
RAG	HiT	1	5	d_κ
RAG	OnT	1	5	d_κ

Note: k: mention-level retrieval depth; d: ranking cutoff in subsumption retrieval; λ: subsumption-score weight. ”NN” denotes the hyperbolic distance-based score function. $s_{sub}(\cdot)$ denotes subsumption score, $s_{sub}(C \sqsubseteq D)$.

4.6 Ablation Studies

Planned ablations include:

1. Assessing performance across OnT encoders trained on ontologies with similar domains of interest with distinct signatures (ANATOMY, GO, GALEN) to assess encoder transferability.
2. Isolating particular top-level hierarchies in SNOMED CT prior to encoder fine-tuning to test performance relative to ontology size (measured in number of active axioms).
3. Modifying training parameters, specifically the batch size, during encoder fine-tuning to assess the trade off in batch size versus performance.
4. Measuring the performance of encoders that operationalise mixed model spaces.

5 Results, Discussion & Evaluation

This section reports the experimental results and discusses the research questions within the context of each task. First, the extent to which HiT, OnT-variant, and mixed models affect single-target retrieval performance (Task 1, RQ1, RQ5) is reviewed, followed by a similar assessment of multi-target retrieval (Task 1, RQ2). Considerations for ablations (cross-signature transfer, RQ4) are presented alongside each task, and the observed performance on downstream biomedical question answering with RAG (Task 2, RQ3) is subsequently discussed.

5.1 Single Target Knowledge Retrieval

Under the single-target regime, results are reported on the 50 OOV mentions measured across multiple models in Table. 13.

Table 4. Single target retrieval performance of OOV entity mentions measured across multiple models

Model	Variant	Scoring	MRR	H@1	H@3	H@5	Med	MR	R@100
BoW	-	TF–IDF	0.26	0.16	0.30	0.34	18.50	90021.30	0.62
BoW	-	BM25	0.29	0.16	0.38	0.42	15.50	44723.44	0.64
SBERT	MiniLM-L12-v2	COS-SIM	0.51	0.34	0.66	0.70	2.00	27.26	0.92
HiT	SNOMED (F)	d_κ (NN)	0.40	0.30	0.46	0.50	5.50	171.46	0.78
OnT	GALEN	d_κ (NN)	0.53	0.32	0.70	0.80	2.00	11.24	0.96
OnT	ANATOMY	d_κ (NN)	0.62	0.48	0.70	0.80	2.00	12.70	0.98
OnT	GO	d_κ (NN)	0.46	0.30	0.52	0.62	3.00	22.46	0.96
OnT	SNOMED (F)	d_κ (NN)	0.55	0.42	0.64	0.70	2.00	168.90	0.92
OnT	SNOMED (M-32)	d_κ (NN)	0.59	0.46	0.66	0.82	2.00	10.18	0.98
OnT	SNOMED (M-64)	d_κ (NN)	0.61	0.48	0.68	0.80	2.00	12.28	0.96
OnT	SNOMED (M-128)	d_κ (NN)	0.64	0.52	0.72	0.78	1.00	18.48	0.96
Mix$_1$	-	$d_\times$	0.45	0.34	0.50	0.54	3.50	78.08	0.82
Mix$_2$	-	$d_\times$	0.63	0.52	0.66	0.80	1.00	12.62	0.96

The results in Table.13 shows that both HiT and OnT encoders trained on the full SNOMED CT ontology outperform lexical baselines in every metric by a wide margin. However, their performance is somewhat less conclusive compared to the contextual baseline, with OnT and SBERT scoring

MRR$= 0.55$ and MRR$= 0.51$, respectively; HiT compares poorly, scoring MRR$= 0.40$[10]. While the hit rate supports MRR (e.g. H@1 scoring 0.42 for OnT-F versus 0.34 for SBERT and 0.3 for HiT-F), it also demonstrates that OnT-variant encoders (miniature models and cross-signature encoders; except GO) retrieve the correct class at rank one much more often than their contextual counterpart (SBERT), providing evidence that transformer-based ontology-aware embedding methods contribute meaningfully to single target knowledge retrieval performance.

The best performing OnT model (M-128) exhibits MRR$= 0.64$, followed closely by the mixed model Mix$_2$ (MRR$= 0.63$) and the ANATOMY variant (MRR$= 0.62$), demonstrating improvements over both lexical (MRR < 0.30) and contextual baselines (MRR$= 0.51$); thereby helping to establish a basis to argue for cross-signature encoder transferability and potential utility within the proposed mixture techniques. Additionally, in the case of mixed models, the scores essentially reflect the choice of model combinations; and whilst Mix$_2$ does leverage SBERT (in addition to OnT M-128 and M-32), it outperforms SBERT in every metric and remains competitive against all other model variants in at least one dimension. For instance, it demonstrates a single point loss in MRR compared to the best performing model, but improves MR considerably and ties at $H@1$.

The secondary reporting metrics support the narrative thus far, demonstrating solid median ranks and competitive recall (Med$= 1$-2, R@100$= 0.92$-0.98) across almost all OnT-variants, compared to SBERT (Med$= 2$, R@100=0.92), while mean rank scores (though tail-sensitive) help signal the previously footnoted training issue outlined in Appendix. F.

5.2 Multiple Target Knowledge Retrieval

For the multi-target regime, the same cross-section of models is evaluated against, using both nearest neighbour search with hyperbolic geodesic distance, as well as subsumption score. Additional figures for evaluation are provided under Appendix. D.

[10] It is worth noting the training process presented some difficulties in specific cases where the ontology used for training is relatively large, as is the case with SNOMED CT. This is accounted for by training on smaller ontology subsets, i.e. the miniture models, discussed in further detail under Appendix. F.

Table 5. Performance of fetching multiple relevant entities using OOV mentions with $\lambda = 0.35$, $d = 5$ (50 Queries)

Model	Variant	Scoring	mAP	nDCG@10	R@100
BoW	-	TFIDF	0.09	0.23	0.27
BoW	-	BM25	0.10	0.26	0.27
SBERT	MiniLM-L12-v2	cos-sim	0.18	0.44	0.41
HiT	SNOMED (F)	d_κ (NN)	0.18	0.34	0.56
HiT	SNOMED (F)	$score_{sub}$	0.25	0.35	0.59
OnT	GALEN	d_κ (NN)	0.21	0.46	0.48
OnT	GALEN	$score_{sub}$	0.21	0.45	0.48
OnT	ANATOMY	d_κ (NN)	0.23	0.52	0.47
OnT	ANATOMY	$score_{sub}$	0.24	0.51	0.48
OnT	GO	d_κ (NN)	0.19	0.41	0.47
OnT	GO	$score_{sub}$	0.20	0.41	0.46
OnT	SNOMED (F-32)	d_κ (NN)	0.18	0.45	0.38
OnT	SNOMED (F-32)	$score_{sub}$	0.16	0.42	0.36
OnT	SNOMED (M-32)	d_κ (NN)	0.21	0.49	0.51
OnT	SNOMED (M-32)	$score_{sub}$	0.23	0.45	0.57
OnT	SNOMED (M-64)	d_κ (NN)	0.21	0.50	0.51
OnT	SNOMED (M-64)	$score_{sub}$	0.22	0.44	0.57
OnT	SNOMED (M-128)	d_κ (NN)	0.22	0.51	0.49
OnT	SNOMED (M-128)	$score_{sub}$	0.22	0.46	0.56
Mix$_1$	-	$d_\times$	0.19	0.39	0.57
Mix$_2$	-	$d_\times$	0.22	0.51	0.50

The results, as presented in Table. 5 provides a complementary perspective to the findings outlined under the single-target regime. That is, ontology-aware encoders consistently outperform lexical baselines and remain competitive, often demonstrating improved performance when compared to their contextual counterparts (SBERT). For instance, both HiT and OnT variants more than double mAP and markedly improve early ranking quality relative to TF–IDF and BM25 (mAP ≤ 0.10; nDCG@10 ≤ 0.26); and while improvements against the contextual baseline (SBERT, mAP $= 0.18$, nDCG@10 $= 0.44$) are somewhat more modest for OnT (up to $+0.06$ mAP with OnT ANATOMY and $+0.05$ mAP with OnT SNOMED M-32), all model variants remain competitive.

Additionally, the application of subsumption scoring improves early retrieval precision on the multiple-target task, yielding a typical 1-2 point gain in mAP in almost all cases. The only exception is OnT

SNOMED F-32–likely due to the prior noted training issues, see Appendix. F–which shows an inferior score.

However, it is important to note that this approach appears to be trading some degree of relevance (with respect to hierarchical ordering, as captured through nDCG) for recall and binary precision. Specifically, there is a fairly consistent pattern showing mAP and R@100 increasing and nDCG decreasing (e.g. mAP$= 0.21 \rightarrow 0.23$, R@100$= 0.51 \rightarrow 0.57$, nDCG$= 0.49 \rightarrow 0.45$ for SNOMED M-32) between the application of NN scoring versus subsumption scoring; as we might expect.

Supprisingly, the most substantive difference in performance between NN distance (d_κ) and subsumption scoring is exhibited by HiT, where the mAP moves from $0.18 \rightarrow 0.25$. This may be, in part, due to the methods used to train HiT versus OnT; as HiT is trained purely on subclass hierarchies across entities in a given ontology, whereas OnT leverages $\mathcal{EL}$ normalised ontology verbalisations for both predictive and inference tasks, leading to an ancillary improvement for OnT and a much more direct, observable improvement with HiT.

Insofar as cross-signature performance is concerned, yet another surprising observation is the performance of OnT ANATOMY, which is actually the strongest OnT variant (mAP $= 0.24$, nDCG@10 $= 0.51$–0.52), likely reflecting substantial signature overlap with SNOMEDCT's `body structure` and `morphological abnormality` branches (Appendix. F). On the other hand, OnT GALEN (mAP $= 0.21$, nDCG@10 ≈ 0.45–0.46) and GO (mAP ≈ 0.19-0.20, nDCG@10 $= 0.41$) are comparatively weaker, suggesting partially misaligned structural or lexical overlap.

Preliminary experiments with mixed model spaces also demonstrate modest gains, with mAP$= 0.22$ and nDCG$= 0.51$. This is understandable, since the scoring method relies entirely on weighted distance measured across heterogeneous embedding space; and while this approach remains exploratory, learning an appropriate weighting is again, task-specific, and is not necessarily trivial. However, the naive weighting strategy used within this exploratory scope, coupled with the admirable performance (given the constraints) may suggest this technique warrants further exploration.

5.3 Ontology Grounded Biomedical MCQA

To evaluate whether improvements at retrieval translate into downstream question-answering performance with RAG, each question the LLM receives is provided with either no retrieved context (No-RAG) or a single axiom verbalisation (top-1) selected by a given retriever. Accuracies are averaged over five runs (with $\pm$ variance).

Table. 6 shows that the results are ultimately inconclusive and that any performance improvements

Table 6. Mirage benchmark results for No-RAG and RAG (5 experimental runs); enrichment with 1 axiom, results shown by retriever for BioMistral and Mistral-7B LLMs; all numbers are accuracy (% ± variance).

		MIRAGE Benchmark Dataset					
LLM	**Method**	**PubMedQA**	**BioASQ**	**MMLU**	**MedQA**	**MedMCQA**	**Avg**
BioMistral (7B)	No RAG	45.08 (±4.60)	59.03 (±1.30)	49.11 (±2.10)	39.81 (±3.46)	36.82 (±1.13)	45.97
	SBERT	44.00 (±6.78)	59.68 (±3.89)	50.49 (±2.02)	40.36 (±1.57)	36.54 (±1.48)	46.21
	HiT	45.48 (±7.60)	59.36 (±2.91)	49.20 (±2.67)	40.24 (±2.20)	36.16 (±1.48)	46.09
	OnT	44.92 (±3.60)	58.41 (±3.07)	50.40 (±2.21)	39.94 (±4.80)	35.93 (±0.77)	45.92
Mistral-7B (v0.3)	No RAG	30.60 (±2.60)	68.58 (±0.98)	62.22 (±1.93)	49.03 (±2.43)	44.04 (±0.93)	50.89
	SBERT	31.16 (±2.20)	68.98 (±2.02)	62.04 (±1.20)	48.85 (±1.73)	44.70 (±0.70)	51.15
	HiT	30.08 (±3.40)	68.77 (±1.62)	61.98 (±1.20)	48.97 (±1.18)	44.08 (±1.22)	50.78
	OnT	31.28 (±4.60)	68.45 (±1.58)	62.44 (±2.52)	49.36 (±0.78)	44.33 (±1.19)	51.17

are most likely attributable to random noise, as they fall well within the bounds of the measured variance. For instance, the relative gain observed between the No RAG control test, SBERT and HiT on BioMistral ($+0.24$ and $+0.12$ avg. accuracy, respectively) falls well below any of the reported variance across the entire set of experiments. Additionally, there is minimal correlation between gains measured between BioMistral and Mistral-7B (v0.03), except for perhaps SBERT, but again, the gains are attributable to variance. Thus, we report these results as inconclusive and further discuss potential issues within the conclusions, under limitations (§6.2).

5.4 Efficiency

By precomputing embeddings for all ontology classes offline, most of the work is shifted to a fast vector-based distance or similarity computation at query time. This computation is calculable in $O(|N_C| \times d)$, where d is a constant factor; with sorting $O(n \log n)$. While time complexity remains equal for hyperbolic embeddings (which use `arcosh`), the per-comparison computation is increasingly demanding. Thus, retaining embeddings within GPU memory across multiple experimental runs (for acceleration) is preferable.

6 Conclusions

The primary focus of this work was to evaluate whether transformer-based ontology-aware encoders, such as HiT and OnT, observably affect single-target and multiple-target knowledge retrieval performance compared to lexical and contextual baselines. Whereas the secondary and tertiary aims focused on performance translation to downstream RAG-based biomedical question answering, and considered exploratory methods such as mixed model spaces and related ablations (cross-signature encoder transferability).

Having used SNOMED CT as the primary ontology and the MIRAGE benchmark for dataset construction and downstream QA, several experiments were conducted to support these research aims and answer the associated research questions, as detailed below.

RQ1 & RQ2: Knowledge Retrieval Effectiveness of HiT & OnT In single target retrieval tasks, the OnT variant encoders regularly demonstrate performance benefits when compared to the contextual baseline; with the best miniature encoder reaching MRR$= 0.64$, versus SBERT MRR$= 0.51$, demonstrating the effective use of these methods for nearest neighbour, single target knowledge retrieval. Additionally, under multi-target knowledge retrieval, OnT and HiT variants more than double mAP over lexical baselines and provide moderate improvements compared to contextual baselines, suggesting that distributional semantics-based methods alone are insufficient for targeting structural depth in formal knowledge representations. This is further reinforced through the application and observation of the two selected scoring methods in multi-target retrieval experiments, which show that the depth bias introduced through λ meaningfully contributes to retrieving hierarchically structured entities, with a minor penalty to relevant ordering (under the assumption that our relevance assignment is task-aligned, which ought to be considered on a case-by-case basis). Additionally, it is noted that subsumption-aware scoring frequently helps to improve multi-target retrieval by $+1$–2 mAP points, further evidencing the effectiveness of these methods as applied to knowledge retrieval tasks.

RQ3: Retrieval Performance Translation to BioMedical MCQA RAG In short, the RAG results were inconclusive and certainly not statistically significant as macro-level accuracy shifts fell well within run-to-run-based variance across both models, BioMistral and Mistral-7B. Several reasons for these results are discussed under limitations (§6.2); and whilst no strong claims can be made, it would appear that naive axiom-stuffing-based prompt-enrichment does not reliably convert retrieval gains to downstream QA performance *under the tested conditions*.

RQ4: OnT-variant Cross-signature Transfer The OnT encoder checkpoints for ANATOMY, GALEN, and GO do not transfer equally well across retrieval tasks. For instance, ANATOMY, whose observed performance appears to indicate strong transfer capabilities, is often seen to perform as well as domain-tuned SNOMED CT variants, likely indicating signature overlap between `body structure` and `morphological abnormality`. Whereas GALEN and GO, whilst yielding acceptable performance, are relatively weaker. Still, the competitive performance of each cross-signature encoder (especially applied under multi-target tasks) helps to demonstrate relatively strong transfer capabilities of the OnT framework.

RQ5: Mixed Model Embeddings A simple mixed model leveraging OnT and SBERT and a naive weighting strategy yields competitive performance, especially in the single-target experiments (with single-target MRR$= 0.63$ versus the best performing encoder MRR$= 0.64$), since L_2 distance conveniently collapses to cosine similarity for SBERT and geodesic distance across OnT models already yielding strong results; this allows for an effective mixed encoder ensemble. Specifically, these results suggest that complementary signals tied to distributional semantics-based methods, such as SBERT, show potential for combination with hierarchical embedding-based representations to provide some degree of meaningful utility; contingent upon careful consideration of learning appropriate, task-specific weightings.

In conclusion, transformer-based ontology-aware embedding models, i.e. HiT and OnT, improve single-target retrieval effectiveness; subsumption-aware scoring also yields measurable benefits for multi-target retrieval. Additionally, performance translation to downstream biomedical MCQA with RAG proved inconclusive, and ablation studies demonstrated strong cross-signature encoder transferability. Furthermore, exploratory approaches, such as mixed models, show potential but require more principled strategies beyond exploratory experimentation to test their effectiveness.

6.1 Project Achievements

- An evaluation of transformer-based ontology embeddings with HiT and OnT, providing experimental results tracking performance against lexical and contextual baselines for subsequently defined knowledge retrieval tasks on out-of-vocabulary (OOV) entity mentions using a variety of scoring functions.
- Preliminary results evaluating the performance of ontology embedding-based retrieval augmented generation (RAG) on the MIRAGE benchmark, discussing the extent to which the approach may affect downstream MCQA performance.

- A curated evaluation dataset for evaluating ontology-embedding based knowledge retrieval on single target and multiple target tasks.
- An open-sourced modular retrieval toolkit supporting lexical, contextual and ontology embedding-based methods (with optional mixed-model configurations) for knowledge retrieval, including reproducible configurations, model checkpoints and experimental logs.
- Demonstration and open-sourced distribution of a functional knowledge retrieval search-based system, and end-to-end biomedical retrieval augmented generation system, with accompanying user interface and instructions for set-up and reuse[11].

6.2 Limitations and Future Work

Perhaps one of the most limiting factors of this study is the size of the annotated OOV set ($N = 50$); though, the time expense required for manual annotation makes the construction of a large dataset infeasible given the timeframe of this project. Additionally, the use of single concept verbalisation ($k = 1$) for RAG enrichment is a potentially limiting factor, leading to no substantial gains for the RAG portion of this study. On the modelling side, we did not explore minimal-module extraction for verbalisation, nor did we apply learnt context selection for RAG (we relied on simple prompt-stuffing).

Future work might benefit from first broadening the evaluation set, time constraints notwithstanding. In addition, the adoption of minimal modules, as in [66], or entailment-aware filtering could prove beneficial, since our approach did not consider any notion of cross-domain or interpretation-based correctness. For mixed models, we suggest learning mixture weights (rather than simply applying isotropic initialisation) and utilising additional model space curvatures, as this may yield further insights. Finally, exploring additional LLM backbones and investigating more appropriate RAG regimes, e.g. $k>1$–with length budgeting, adaptive selection, and so forth–is suggested.

6.3 Closing Remarks

Our results show clear value in ontology-aware encoders for knowledge retrieval, especially for OnT in both single and multi-target regimes; and whilst naive RAG integration does not yet exploit these gains, future work could certainly expand on our methods. The findings support continued development of $\mathcal{EL}$-aware encoders for prediction-framed retrieval through geometry-matched scoring. These efforts could be coupled with more selective, logic-aware context integration during RAG

[11] See the supplementary materials that are provided through the GitHub link and video demonstration.

(with more appropriate LLM backbones), as we believe this would help to close the gap between retrieval effectiveness and downstream biomedical QA performance.

References

[1] J. Achiam et al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023 (cited on p. 10).

[2] G. Team et al., "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023 (cited on p. 10).

[3] M. U. Hadi et al., "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," *Authorea preprints*, vol. 1, no. 3, pp. 1–26, 2023, Publisher: Authorea (cited on p. 10).

[4] Y. Gao et al., *Retrieval-Augmented Generation for Large Language Models: A Survey*, en, arXiv:2312.10997 [cs], Mar. 2024. doi: 10.48550/arXiv.2312.10997. Accessed: Jul. 2, 2025. [Online]. Available: http://arxiv.org/abs/2312.10997 (cited on p. 10).

[5] J. Gravel, M. D'Amours-Gravel, and E. Osmanlliu, "Learning to Fake It: Limited Responses and Fabricated References Provided by ChatGPT for Medical Questions," en, *Mayo Clinic Proceedings: Digital Health*, vol. 1, no. 3, pp. 226–234, Sep. 2023, issn: 29497612. doi: 10.1016/j.mcpdig.2023.05.004. Accessed: Jul. 2, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2949761223000366 (cited on p. 10).

[6] C. Agarwal, S. H. Tanneru, and H. Lakkaraju, *Faithfulness vs. Plausibility: On the (Un)Reliability of Explanations from Large Language Models*, arXiv:2402.04614 [cs], Mar. 2024. doi: 10.48550/arXiv.2402.04614. Accessed: Jul. 2, 2025. [Online]. Available: http://arxiv.org/abs/2402.04614 (cited on p. 10).

[7] P. Lewis et al., *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, arXiv:2005.11401 [cs], Apr. 2021. doi: 10.48550/arXiv.2005.11401. Accessed: Sep. 11, 2025. [Online]. Available: http://arxiv.org/abs/2005.11401 (cited on p. 10).

[8] B. Ni et al., "Towards trustworthy retrieval augmented generation for large language models: A survey," *arXiv preprint arXiv:2502.06872*, 2025 (cited on p. 10).

[9] A. J. Oche et al., "A systematic review of key retrieval-augmented generation (RAG) systems: Progress, gaps, and future directions," *arXiv preprint arXiv:2507.18910*, 2025 (cited on p. 10).

[10] X. Li and J. Ouyang, "How does knowledge selection help retrieval augmented generation?" *arXiv preprint arXiv:2410.13258*, 2024 (cited on p. 10).

[11] SNOMED International, “SNOMED CT international edition, september 2025 release,” SNOMED International, London, UK, manual, Sep. 2025. [Online]. Available: `https://www.snomed.org/snomed-ct` (cited on pp. 10, 28).

[12] D. Lee et al., “A survey of SNOMED CT implementations,” *Journal of biomedical informatics*, vol. 46, no. 1, pp. 87–96, 2013, Publisher: Elsevier (cited on p. 10).

[13] E. Chang and J. Mostafa, “The use of SNOMED CT, 2013-2020: A literature review,” *Journal of the American Medical Informatics Association*, vol. 28, no. 9, pp. 2017–2026, 2021, Publisher: Oxford University Press (cited on p. 10).

[14] B. Motik, P. F. Patel-Schneider, and B. Parsia, *OWL 2 web ontology language profiles (second edition)*, tex.howpublished: W3C Recommendation, Dec. 2012. [Online]. Available: `https://www.w3.org/TR/owl2-profiles/` (cited on pp. 10, 14).

[15] D. Wei et al., “Structural measures to track the evolution of SNOMED CT hierarchies,” *Journal of biomedical informatics*, vol. 57, pp. 278–287, 2015, Publisher: Elsevier (cited on p. 10).

[16] D. Lee et al., “Literature review of SNOMED CT use,” *Journal of the American Medical Informatics Association*, vol. 21, no. e1, e11–e19, 2014, Publisher: BMJ Publishing Group BMA House, Tavistock Square, London, WC1H 9JR (cited on p. 10).

[17] T. Ohlsen, A. Sander, and J. Ingenerf, *ECLed – A Tool Supporting the Effective Use of the SNOMED CT Expression Constraint Language*. May 2025. doi: `10.21203/rs.3.rs-6644476/v1` (cited on p. 10).

[18] C. D. Manning, *Introduction to information retrieval*. Syngress Publishing, 2008 (cited on pp. 10, 20, 29, 32, 33).

[19] E. M. Voorhees, “Natural language processing and information retrieval,” in *International summer school on information extraction*, Springer, 1999, pp. 32–48 (cited on p. 10).

[20] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering.,” in *Emnlp (1)*, 2020, pp. 6769–6781 (cited on pp. 10, 20).

[21] J. Chen et al., “Contextual semantic embeddings for ontology subsumption prediction,” *World Wide Web-internet and Web Information Systems*, vol. 26, no. 5, pp. 2569–2591, 2023, Publisher: Springer (cited on p. 10).

[22] J. Chen et al., “Ontology embedding: A survey of methods, applications and resources,” *IEEE Transactions on Knowledge and Data Engineering*, 2025, Publisher: IEEE (cited on p. 11).

[23] M. Kulmanov et al., “El embeddings: Geometric construction of models for the description logic el++,” *arXiv preprint arXiv:1902.10499*, 2019 (cited on p. 11).

[24] M. Nickel and D. Kiela, *Poincaré Embeddings for Learning Hierarchical Representations*, arXiv:1705.0803 [cs], May 2017. doi: 10.48550/arXiv.1705.08039. Accessed: Jul. 3, 2025. [Online]. Available: http://arxiv.org/abs/1705.08039 (cited on pp. 11, 15, 17, 21).

[25] F. Z. Smaili, X. Gao, and R. Hoehndorf, "OPA2Vec: Combining formal and informal content of biomedical ontologies to improve similarity-based prediction," *Bioinformatics (Oxford, England)*, vol. 35, no. 12, pp. 2133–2140, 2019, Publisher: Oxford University Press (cited on pp. 11, 22).

[26] Y. He et al., *Language Models as Hierarchy Encoders*, arXiv:2401.11374 [cs], Nov. 2024. doi: 10.48550/arXiv.2401.11374. Accessed: Jul. 2, 2025. [Online]. Available: http://arxiv.org/abs/2401.11374 (cited on pp. 11, 15–18, 20, 22, 29).

[27] H. Yang et al., "Language models as ontology encoders," *arXiv preprint arXiv:2507.14334*, 2025 (cited on pp. 11, 15–18, 20, 22).

[28] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993, Publisher: Elsevier (cited on p. 13).

[29] F. Baader et al., *An introduction to description logic*. Cambridge University Press, 2017 (cited on p. 13).

[30] P. Hitzler et al., "OWL 2 web ontology language primer," *W3C recommendation*, vol. 27, no. 1, p. 123, 2009 (cited on p. 13).

[31] W3C OWL Working Group, *OWL 2 web ontology language document overview (second edition)*, tex.howpublished: W3C Recommendation, Dec. 2012. [Online]. Available: https://www.w3.org/TR/owl2-overview/ (cited on p. 13).

[32] B. Motik et al., *OWL 2 web ontology language: Structural specification and functional-style syntax (second edition)*, tex.howpublished: W3C Recommendation, Dec. 2012. [Online]. Available: https://www.w3.org/TR/owl2-syntax/ (cited on p. 13).

[33] Y. Kazakov, M. Krötzsch, and F. Simančík, "The incredible ELK: From polynomial procedures to efficient reasoning with □ □ ontologies," *Journal of automated reasoning*, vol. 53, no. 1, pp. 1–61, 2014, Publisher: Springer (cited on p. 15).

[34] F. Baader, *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003 (cited on p. 15).

[35] R. Shearer and I. Horrocks, "Exploiting Partial Information in Taxonomy Construction," en, in *The Semantic Web - ISWC 2009*, ISSN: 1611-3349, Springer, Berlin, Heidelberg, 2009, pp. 569–584, isbn: 978-3-642-04930-9. doi: 10.1007/978-3-642-04930-9_36. Accessed: Sep. 12, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-04930-9_36 (cited on p. 15).

[36] B. A. Davey and H. A. Priestley, *Introduction to lattices and order*. Cambridge university press, 2002 (cited on p. 15).

[37] A. Grover and J. Leskovec, “Node2vec: Scalable Feature Learning for Networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16, New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 855–864, isbn: 978-1-4503-4232-2. doi: 10.1145/2939672.2939754. Accessed: Sep. 12, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939754 (cited on pp. 15, 21).

[38] J. Devlin et al., “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186 (cited on p. 16).

[39] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, arXiv:1908.10084 [cs], Aug. 2019. doi: 10.48550/arXiv.1908.10084. Accessed: Jul. 3, 2025. [Online]. Available: http://arxiv.org/abs/1908.10084 (cited on pp. 16, 20).

[40] M. P. Do Carmo and J. Flaherty Francis, *Riemannian geometry*. Springer, 1992, vol. 2 (cited on p. 17).

[41] A. Gu et al., “Learning mixed-curvature representations in product spaces,” in *International conference on learning representations*, 2018 (cited on pp. 17, 23, 24).

[42] J. G. Ratcliffe, *Foundations of hyperbolic manifolds*. Springer, 2019 (cited on p. 17).

[43] A. Vaswani et al., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017 (cited on pp. 21, 49).

[44] J. Chen et al., *OWL2Vec*: Embedding of OWL Ontologies*, arXiv:2009.14654 [cs], Jan. 2021. doi: 10.48550/arXiv.2009.14654. Accessed: Sep. 12, 2025. [Online]. Available: http://arxiv.org/abs/2009.14654 (cited on p. 22).

[45] J. M. Lee, *Riemannian manifolds: an introduction to curvature*. Springer Science & Business Media, 2006, vol. 176 (cited on p. 24).

[46] J. M. Lee, *Introduction to riemannian manifolds*. Springer, 2018, vol. 2 (cited on p. 24).

[47] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006 (cited on pp. 24, 25).

[48] G. Xiong et al., “Benchmarking retrieval-augmented generation for medicine,” in *Findings of the association for computational linguistics ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, Aug. 2024, pp. 6233–6251. doi: 10.18653/v1/2024.findings-acl.372. [Online]. Available: https://aclanthology.org/2024.findings-acl.372 (cited on pp. 28, 30).

[49] SNOMED International, *SNOMED OWL toolkit*, 2018. [Online]. Available: https://github.com/IHTSDO/snomed-owl-toolkit (cited on p. 29).

[50] R. C. Jackson et al., “ROBOT: A tool for automating ontology workflows,” *BMC bioinformatics*, vol. 20, no. 1, p. 407, 2019, Publisher: Springer (cited on p. 29).

[51] D. Jin et al., “What disease does this patient have? A large-scale open domain question answering dataset from medical exams,” *arXiv preprint arXiv:2009.13081*, 2020 (cited on p. 30).

[52] D. Hendrycks et al., “Measuring massive multitask language understanding,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021 (cited on p. 30).

[53] A. Pal, L. K. Umapathi, and M. Sankarasubbu, “MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering,” en, in *Proceedings of the Conference on Health, Inference, and Learning*, ISSN: 2640-3498, PMLR, Apr. 2022, pp. 248–260. Accessed: May 28, 2025. [Online]. Available: https://proceedings.mlr.press/v174/pal22a.html (cited on p. 30).

[54] Q. Jin et al., “PubMedQA: A dataset for biomedical research question answering,” 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)*, 2019, pp. 2567–2577 (cited on p. 30).

[55] A. Krithara et al., “BioASQ-QA: A manually curated corpus for biomedical question answering,” *Scientific Data*, vol. 10, p. 170, 2023. [Online]. Available: https://doi.org/10.1038/s41597-023-02068-4 (cited on p. 30).

[56] M. Neumann et al., “ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing,” in *Proceedings of the 18th BioNLP workshop and shared task*, tex.eprint: arXiv:1902.07669, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327. doi: 10.18653/v1/W19-5034. [Online]. Available: https://www.aclweb.org/anthology/W19-5034 (cited on p. 30).

[57] S. Pyysalo et al., “Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013,” *BMC bioinformatics*, vol. 16, no. Suppl 10, S2, 2015, Publisher: Springer (cited on p. 30).

[58] Allen Institute for AI, *scispaCy models*, 2019. [Online]. Available: https://allenai.github.io/scispacy/ (cited on p. 30).

[59] M. Honnibal et al., “spaCy: Industrial-strength natural language processing in python,” 2020. doi: 10.5281/zenodo.1212303 (cited on p. 30).

[60] Y. He et al., “DeepOnto: A Python package for ontology engineering with deep learning,” *Semantic Web*, vol. 15, no. 5, pp. 1991–2004, 2024 (cited on p. 31).

[61] H. Su et al., “Bright: A realistic and challenging benchmark for reasoning-intensive retrieval,” *arXiv preprint arXiv:2407.12883*, 2024 (cited on p. 32).

[62] V. Lavrenko, *IR13: Evaluating search engines*, tex.howpublished: YouTube playlist, 2013. [Online]. Available: `https://www.youtube.com/playlist?list=PLBv09BD7ez_6nqE9YU9bQXpjJ5jJ1Kgr9` (cited on pp. 32, 33, 35).

[63] E. M. Voorhees et al., “The trec-8 question answering track report.,” in *Trec*, vol. 99, 1999, pp. 77–82 (cited on p. 33).

[64] E. M. Voorhees, D. K. Harman, et al., *TREC: Experiment and evaluation in information retrieval*. MIT press Cambridge, 2005, vol. 63 (cited on p. 35).

[65] R. Burden, J. Faires, and A. Burden, *Numerical analysis*. Pacific Grove, CA: Brooks/Cole Pub. Co, 2015 (cited on p. 35).

[66] B. C. Grau et al., “Just the right amount: Extracting modules from ontologies,” en, in *Proceedings of the 16th international conference on World Wide Web*, Banff Alberta Canada: ACM, May 2007, pp. 717–726, isbn: 978-1-59593-654-7. doi: `10.1145/1242572.1242669`. Accessed: Sep. 14, 2025. [Online]. Available: `https://dl.acm.org/doi/10.1145/1242572.1242669` (cited on p. 42).

[67] R. M. Schmidt, *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*, arXiv:1912.05911 [cs], Nov. 2019. doi: `10.48550/arXiv.1912.05911`. Accessed: Feb. 11, 2025. [Online]. Available: `http://arxiv.org/abs/1912.05911` (cited on p. 49).

[68] Y. A. Malkov and D. A. Yashunin, “Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, Apr. 2020, issn: 1939-3539. doi: `10.1109/TPAMI.2018.2889473`. Accessed: Sep. 14, 2025. [Online]. Available: `https://ieeexplore.ieee.org/document/8594636` (cited on p. 51).

[69] L. Prokhorenkova et al., “Graph-based Nearest Neighbor Search in Hyperbolic Spaces,” en, Oct. 2021. Accessed: Sep. 14, 2025. [Online]. Available: `https://openreview.net/forum?id=USIgIY6TNDe` (cited on p. 51).

[70] C. Zheng et al., “Large language models are not robust multiple choice selectors,” *arXiv preprint arXiv:2309.03882*, 2023 (cited on p. 52).

Appendices

A Language Models

Contemporary language models used for text generation usually take the form of neural networks; examples include recurrent neural networks (RNNs)[12], sequence-to-sequence (Seq2Seq) models, and, more recently, Transformer architectures based on self-attention, in encoder-only, decoder-only, and encoder-decoder variants.

In the encoder-decoder model, the encoder network produces contextual representations for each token in the source sequence using self-attention and a feed-forward neural architecture. On the other hand, the decoder network applies masked self-attention over previously generated tokens, and cross-attention[13] to the encoder's output, resulting in a probability distribution over the token vocabulary via a softmax function. This *generation* produces a target sequence, one token at a time[14].

B System Implementation

B.1 Class Design & Diagram

- **BaseRetriever** An abstract base class (ABC) acts as a contract specification for BaseModelRetriever and other extended classes.
- **BaseModelRetriever** Implements common logic, such as the `retrieve` method and enables the registration of the score callback functions.
- **BaseEntitySelector** An LLM-specific interface that encapsulates functionality for cross-mention candidate merging (in user prompts) and enables naive re-ranking.
- **MistralLLM** The LLM interface for the standardisation of generation, providing prompt templates, an injection API, and a logits processor for MCQA (see `nvidia/logits-processor-zoo`).
- **QATestHarness** The single point of orchestration and logging for LLM-specific experiments.

[12] Additional variants are omitted for brevity. For additional literature review on long short-term memory networks (LSTMs), bidirectional LSTMs, gated recurrent units (GRUs), and so forth, see [67].

[13] Cross-attention refers to combining attentional keys (K) and values (V) from the encoder, and queries (Q) from the decoder's sequence state [43].

[14] Decoder-only models omit cross-attention, relying solely on masked self-attention within stacked decoder layers.

- **Experiment Files & Notebooks** Allow for and document experimental runs.

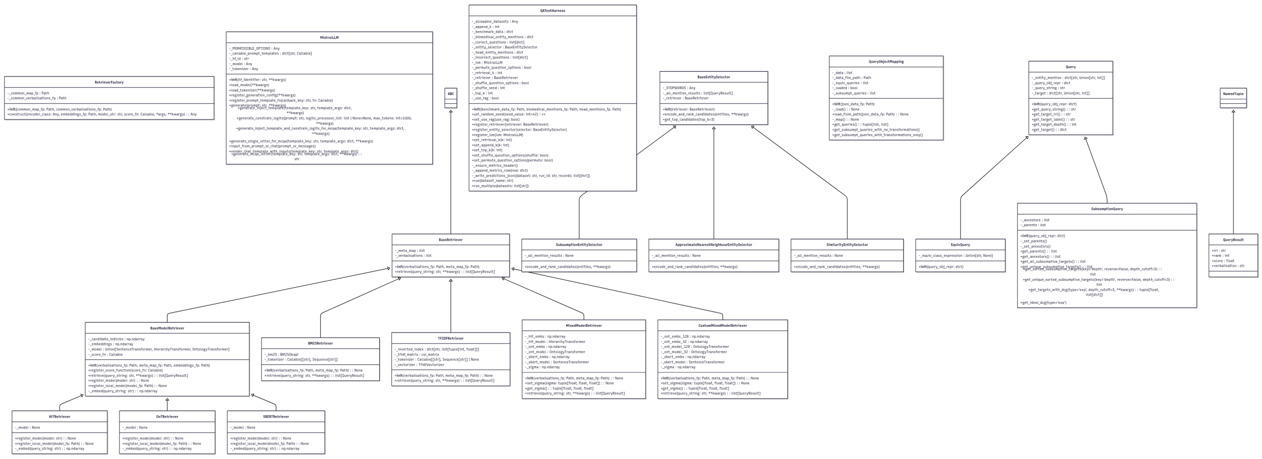

RetrieverFactory
-_common_map_fp : Path
-_common_verbalisations_fp : Path
+init(common_map_fp: Path, common_verbalisations_fp: Path)
+construct(encoder_class: Any, embeddings_fp: Path, model_str: str, score_fn: Callable, *args, **kwargs) : : Any
MistralLLM
-_PERMISSIBLE_OPTIONS : Any
-_callable_prompt_templates : dict[str, Callable]
-_hf_id : str
-_model : Any
-_tokenizer : Any
+init(hf_identifier: str, **kwargs)
+load_model(**kwargs)
+load_tokenizer(**kwargs)
+register_generation_config(**kwargs)
+register_prompt_template_fn(callback_key: str, fn: Callable)
+generate(prompt: str, **kwargs)
+generate_inject_template(template_key: str, template_args: dict, **kwargs)
+generate_constrain_logits(prompt: str, logits_processor_list: list | None=None, max_tokens: int=1000, **kwargs)
+generate_inject_template_and_constrain_logits_for_mcqa(template_key: str, template_args: dict, **kwargs)
+generate_single_letter_for_mcqa(template_key: str, template_args: dict, **kwargs)
+input_from_prompt_or_chat(prompt_or_messages)
+render_chat_template_with_inputs(template_key: str, template_args: dict)
+generate_mcqa_letter(template_key: str, template_args: dict, **kwargs) : : str
ABC
QATestHarness
-_allowable_datasets : Any
-_append_k : int
-_benchmark_data : dict
-_biomedical_entity_mentions : dict
-_correct_questions : list[dict]
-_entity_selector : BaseEntitySelector
-_head_entity_mentions : dict
-_incorrect_questions : list[dict]
-_llm : MistralLLM
-_permute_question_options : bool
-_retrieval_k : int
-_retriever : BaseRetriever
-_shuffle_question_options : bool
-_shuffle_seed : int
-_top_k : int
-_use_rag : bool
+init(benchmark_data_fp: Path, biomedical_mentions_fp: Path, head_mentions_fp: Path)
+set_random_seed(seed_value: int=42) : <>
+set_use_rag(use_rag: bool)
+register_retriever(retriever: BaseRetriever)
+register_entity_selector(selector: BaseEntitySelector)
+register_llm(llm: MistralLLM)
+set_retrieval_k(k: int)
+set_append_k(k: int)
+set_top_k(k: int)
+set_shuffle_question_options(shuffle: bool)
+set_permute_question_options(permute: bool)
-_ensure_metrics_header()
-_append_metrics_row(row: dict)
-_write_predictions_jsonl(dataset: str, run_id: str, records: list[dict])
+run(dataset_name: str)
+run_multiple(datasets: list[str])
BaseEntitySelector
-_STOPWORDS : Any
-_all_mention_results : list[QueryResult]
-_retriever : BaseRetriever
+init(retriever: BaseRetriever)
+encode_and_rank_candidates(entities, **kwargs)
+get_top_candidates(top_k=3)
QueryObjectMapping
-_data : list
-_data_file_path : Path
-_equiv_queries : list
-_loaded : bool
-_subsumpt_queries : list
+init(json_data_fp: Path)
-_load() : : None
+load_from_path(json_data_fp: Path) : : None
-_map() : : None
+get_queries() : : tuple[list, list]
+get_subsumpt_queries_with_no_transformations()
+get_subsumpt_queries_with_transformations_only()
Query
-_entity_mention : dict[str, Union[str, int]]
-_query_obj_repr : dict
-_query_string : str
-_target : dict[str, Union[str, int]]
+init(query_obj_repr: dict)
+get_query_string() : : str
+get_target_iri() : : str
+get_target_label() : : str
+get_target_depth() : : int
+get_target() : : dict
NamedTuple
BaseRetriever
-_meta_map : list
-_verbalisations : list
+init(verbalisations_fp: Path, meta_map_fp: Path)
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
SubsumptionEntitySelector
-_all_mention_results : None
+encode_and_rank_candidates(entities, **kwargs)
ApproximateNearestNeighbourEntitySelector
-_all_mention_results : None
+encode_and_rank_candidates(entities, **kwargs)
SimilarityEntitySelector
-_all_mention_results : None
+encode_and_rank_candidates(entities, **kwargs)
EquivQuery
-_equiv_class_expression : Union[str, None]
+init(query_obj_repr: dict)
SubsumptionQuery
-_ancestors : list
-_parents : list
+init(query_obj_repr: dict)
-_set_parents()
-_set_ancestors()
+get_parents() : : list
+get_ancestors() : : list
+get_all_subsumptive_targets() : : list
+get_unique_subsumptive_targets() : : list
+get_sorted_subsumptive_targets(key='depth', reverse=False, depth_cutoff=3) : : list
+get_unique_sorted_subsumptive_targets(key='depth', reverse=False, depth_cutoff=3) : : list
+get_targets_with_dcg(type='exp', depth_cutoff=3, **kwargs) : : tuple[float, list[dict]]
+get_ideal_dcg(type='exp')
QueryResult
+iri : str
+rank : int
+score : float
+verbalisation : str
BaseModelRetriever
-_candidate_indicies : np.ndarray
-_embeddings : np.ndarray
-_model : Union[SentenceTransformer, HierarchyTransformer, OntologyTransformer]
-_score_fn : Callable
+init(verbalisations_fp: Path, meta_map_fp: Path, embeddings_fp: Path)
+register_score_function(score_fn: Callable)
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
+register_model(model: str) : : None
+register_local_model(model_fp: Path) : : None
-_embed(query_string: str) : : np.ndarray
BM25Retriever
-_bm25 : BM25Okapi
-_tokenizer : Callable[[str], Sequence[str]]
+init(verbalisations_fp: Path, meta_map_fp: Path) : : None
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
TFIDFRetriever
-_inverted_index : dict[str, list[tuple[int, float]]]
-_tfidf_matrix : csr_matrix
-_tokenizer : Callable[[str], Sequence[str]] | None
-_vectorizer : TfidfVectorizer
+init(verbalisations_fp: Path, meta_map_fp: Path) : : None
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
MixedModelRetriever
-_hit_embs : np.ndarray
-_hit_model : HierarchyTransformer
-_ont_embs : np.ndarray
-_ont_model : OntologyTransformer
-_sbert_embs : np.ndarray
-_sbert_model : SentenceTransformer
-_sigma : np.ndarray
+init(verbalisations_fp: Path, meta_map_fp: Path) : : None
+set_sigma(sigma: tuple[float, float, float]) : : None
+get_sigma() : : tuple[float, float, float]
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
CustomMixedModelRetriever
-_ont_embs_128 : np.ndarray
-_ont_embs_32 : np.ndarray
-_ont_model_128 : OntologyTransformer
-_ont_model_32 : OntologyTransformer
-_sbert_embs : np.ndarray
-_sbert_model : SentenceTransformer
-_sigma : np.ndarray
+init(verbalisations_fp: Path, meta_map_fp: Path) : : None
+set_sigma(sigma: tuple[float, float, float]) : : None
+get_sigma() : : tuple[float, float, float]
+retrieve(query_string: str, **kwargs) : : list[QueryResult]
HiTRetriever
-_model : None
+register_model(model: str) : : None
+register_local_model(model_fp: Path) : : None
-_embed(query_string: str) : : np.ndarray
OnTRetriever
-_model : None
+register_model(model: str) : : None
+register_local_model(model_fp: Path) : : None
-_embed(query_string: str) : : np.ndarray
SBERTRetriever
-_model : None
+register_model(model: str) : : None
+register_local_model(model_fp: Path) : : None
-_embed(query_string: str) : : np.ndarray

Fig. 14. Retriever class hierarchy and key methods (rendered at high DPI, allowing for effective zooming, a detailed breakdown of class design is shipped in the associated notebooks, see the GitHub repo).

B.2 LLM Experiment Retrieval Flows

All retrieval flows tied to the LLM experiment are presented in Fig. 15.

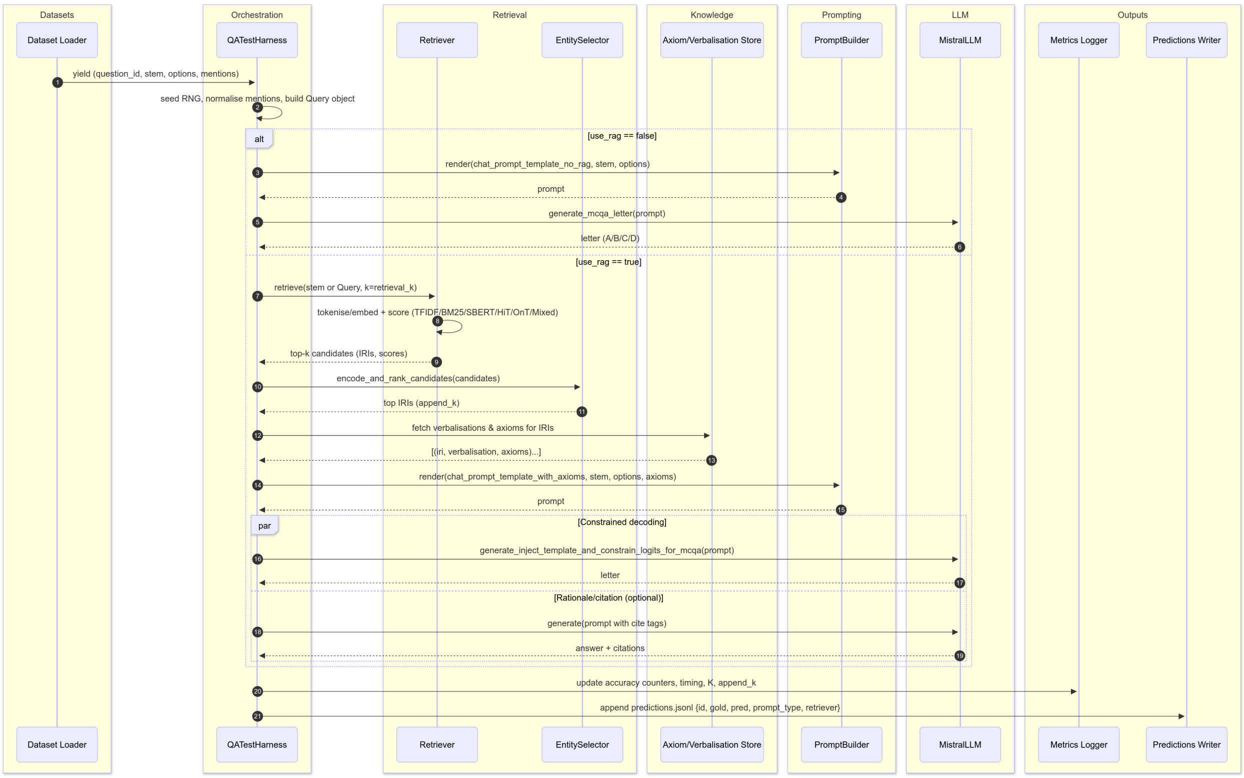

Datasets
Orchestration
Retrieval
Knowledge
Prompting
LLM
Outputs
Dataset Loader
QATestHarness
Retriever
EntitySelector
Axiom/Verbalisation Store
PromptBuilder
MistralLLM
Metrics Logger
Predictions Writer
1 yield (question_id, stem, options, mentions)
2 seed RNG, normalise mentions, build Query object
alt
[use_rag == false]
3 render(chat_prompt_template_no_rag, stem, options)
4 prompt
5 generate_mcqa_letter(prompt)
6 letter (A/B/C/D)
[use_rag == true]
7 retrieve(stem or Query, k=retrieval_k)
8 tokenise/embed + score (TFIDF/BM25/SBERT/HiT/OnT/Mixed)
9 top-k candidates (IRIs, scores)
10 encode_and_rank_candidates(candidates)
11 top IRIs (append_k)
12 fetch verbalisations & axioms for IRIs
13 [(iri, verbalisation, axioms)...]
14 render(chat_prompt_template_with_axioms, stem, options, axioms)
15 prompt
par
[Constrained decoding]
16 generate_inject_template_and_constrain_logits_for_mcqa(prompt)
17 letter
[Rationale/citation (optional)]
18 generate(prompt with cite tags)
19 answer + citations
20 update accuracy counters, timing, K, append_k
21 append predictions.jsonl {id, gold, pred, prompt_type, retriever}
Dataset Loader
QATestHarness
Retriever
EntitySelector
Axiom/Verbalisation Store
PromptBuilder
MistralLLM
Metrics Logger
Predictions Writer

Fig. 15. High level overview of LLM-centric sequence flows (detailed view).

B.3 Indexing & Candidate Selection

Algorithms for approximating nearest neighbour search, such as HNSW [68], are often used for the purposes of indexing embedding stores in place of true nearest neighbour algorithms, as they provide an efficient means to obtain a 'good enough' result set. However, their adoption typically introduces some degree of approximation bias; they are usually applied at a much larger scale and are more commonly used for Euclidean representations (e.g. SBERT). Whilst approaches for indexing hyperbolic embeddings have been proposed [69], they are, in some cases, immature. Thus, our approach ensures score functions do not leverage such techniques and instead, we rely entirely on computationally efficient general-matrix vector multiplication (GEMV) and GPU optimisation as methods to support exhaustive linear search ranking in brute force retrieval, allowing for reliable and fair comparison.

C RAG Evaluation Procedure

Context Retrieval During pre-retrieval, named entity recognition (NER) is applied to each question to extract a set of mentions $\{q_{s1}, q_{s2}, \ldots, q_{sn}\}$. Heuristic filtering (stop word removal) is run to remove trivial terms, then each mention is passed to a retriever. The top-k retrieved results are then verbalised according to their subclass or equivalence axiom and appended to the original question prompt as contextual knowledge. Experiments vary k between 1 and 5 to assess the effect of retrieval depth on downstream QA performance. Lexical retrieval methods and combinatorial strategies such as Reciprocal Rank Fusion are not applied here, as RQ3 specifically focuses on evaluating the contribution of ontology-aware embeddings to retrieval-augmented generation. SBERT and 'no-RAG' based approaches are ran as controls.

Model Selection & Justification For the application of retrieval augmented generation (Task 2), we employ several open, decoder-only instruct model variants of Mistral-7B, including BioMistral. These models are selected due to their relative ease to be run locally in a reproducible manner, and they are not prohibitively memory-intensive. Additionally, BioMistral benefits from pretraining on PubMed documents, it is therefore *reasonable to expect* that the language model would assign a higher likelihood to injected biomedical strings from SNOMED CT's entity lexicon with associated concept verbalisations (due to domain overlap), as compared to its general-purpose counterpart, Mistral-7B. Thus, the choice of models reflects the *assumption* that a lower perplexity score (or, a more fluent model) might be more likely to utilise SNOMED CT concept verbalisations more effectively for MCQA. Alternative models such as PMC-LLaMa and MEDITRON were also considered, but the additional engineering effort and associated time constraints required to support both LLaMa and Mistral-based architectures within a unified test harness ultimately coincided with project time constraints.

Bias Reduction Given the inherent selection bias observed in LLMs during evaluation on MCQs [70], two selected strategies to mitigate against this phenomenon include option shuffling, where the order in which answers are presented is modified upon every prompt, and option permutation, where MCQA option-answer mappings are randomly permuted according to a preset seed.

D Additional Evaluation Figures

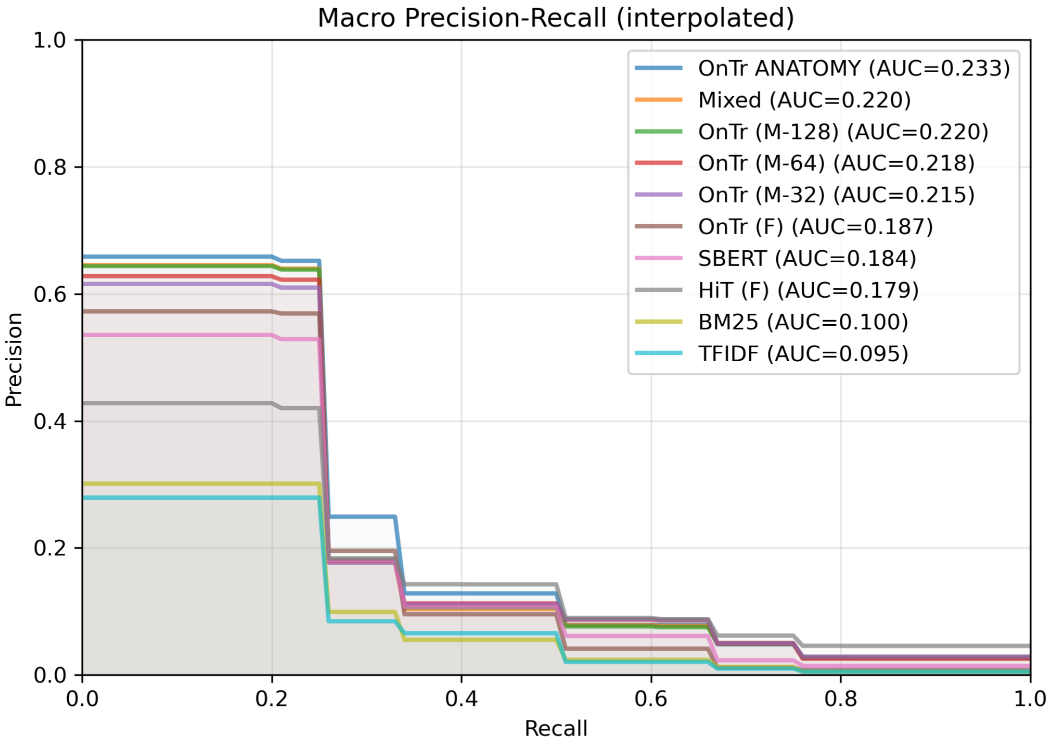


Fig. 16. An interpolated Precision-Recall plot, showing performance across multiple systems for multi-target retrieval.

E Experimental Implementation

E.1 Lexical Baselines

TF–IDF Class labels are used as *documents*, and the `sklearn` libraries `TfidfVectorizer` class is instanciated at run-time to construct and persist a sparse document-term matrix to disk (or may be retained in memory during the course of an experimental run). Configuration settings are set to `lowercase=True`, `stop_words="english"`, `smooth_idf=True` and `norm=None`; all other settings are left as stock.

BM25 The `rank_bm25` Python library is leveraged for ease of implementation and for producing a reliable baseline score. Unlike TF–IDF, which uses a stock implementation, the hyperparameters k_1 and b are tuned on a small non-overlapping query set (REF) and set to $k_1 = 1.5;\ \ b = 0.7$.

E.2 Contextual Baseline

The pre-trained Sentence-BERT encoder (`all-MiniLM-L12-v2`) is employed without domain fine-tuning, such that each $t(C)$ text-based class representation and query (mention) q_s are embedded to $\mathbb{R}^d$. Vectors are L2-normalised, and scored by cosine similarity (Eq. (9)). This baseline tests whether a distributional semantics-based approach would suffice for text-based knowledge retrieval in place of ontology embeddings for the proposed experiments.

F Assumption Testing

While OnT models are learning/re-training on the full SNOMED CT ontology, their performance at mapping NF2-NF4 is still much to be desired. There is considerable room for improvement in prediction tasks compared to the results presented in the original OnT paper (for training logs, see the GitHub repo). The H1 score for NF1 is aligned with the expected performance, though this does not seem to be the case for the remaining scores. Several factors could be influencing the model, including (but not limited to) the number of epochs (set to one), the batch size/learning rate, representative class balance, and the sheer size of the ontology.

SNOMED CT is significantly larger and more complex than ANATOMY, Gene Ontology (GO) and GALEN, as SNOMED CT has $\geq\ 300,000$ stated axioms and $\geq\ 700,000$ verbalisations when EL-normalised. Our working theory, understanding, or assumption is that adopting a smaller ontology with a potentially more balanced set of representative normal forms (NF1 $\rightarrow$ NF4) may help to improve the training process. Additionally, we assume that the conceptual domain overlap between SNOMED CT and ANATOMY is significant, which we test under Appendix. F.

Following from our assumptions, we perform multiple tests, which include training miniature models (with varying batch sizes) and comparing performance between cross-domain queries; i.e., anatomical queries, taken to be the set of axioms that relate to body structure and morphological abnormality, and non-anatomical queries, aiming to:

1. Test whether we can effectively leverage a subset of the original data (a smaller ontology) during model training, such that any experimental results obtained during downstream retrieval can aid in supporting our assumption about ontology size.
2. Test the performance of ANATOMY (along with domain-tuned models) on subsets of the original evaluation data to test whether ANATOMY does, in fact, perform better on anatomical queries, and to what degree this difference is apparent.
3. Capture enough of the SNOMED CT ontology during miniature model training such that the performance can generalise well enough to the retrieval tasks on our evaluation data.

We accomplish this by training miniature models (with batch sizes: 32, 64, 128) on a subset of SNOMED CT (with only $36,420$ subclass axioms and $13,680$ equivalent classes) containing the following concept branches:

- Body Structure
- Clinical Finding
- Event
- Procedure

In addition, we divide our evaluation queries into ANATOMY-related and ANATOMY-unrelated subsets, then re-run the original experiments against each constituent subset, allowing us to identify whether (and to what degree) our assumption about domain-overlap holds.

We find that the ontology size appears to affect model performance. Specifically, much larger and more complex ontologies are more challenging for the model to learn effective mappings, and there does appear to be domain overlap between SNOMED CT and ANATOMY, helping to support both of our assumptions in this case. We present the findings in support of our conclusions below.

Table 7. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on the procedure subset (4 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.65	0.50	0.75	0.75	1.50	4.00	1.00
OnT	ANATOMY	0.65	0.50	0.75	0.75	1.50	3.50	1.00
OnT	GO	0.44	0.25	0.50	0.75	3.00	101.75	0.75
OnT	SNOMED (F-64)	0.79	0.75	0.75	0.75	1.00	2.25	1.00
OnT	SNOMED (M-32)	0.65	0.50	0.75	1.00	2.00	2.25	1.00
OnT	SNOMED (M-64)	0.63	0.50	0.75	1.00	2.00	2.50	1.00
OnT	SNOMED (M-128)	0.65	0.50	0.75	0.75	1.50	3.25	1.00

Table 8. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on the substance subset (12 Queries). *Note that SNOMED Mini has not been exposed (trained on) substance data either.*

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.53	0.33	0.67	0.83	2.50	3.00	1.00
OnT	ANATOMY	0.61	0.50	0.67	0.67	2.00	3.17	1.00
OnT	GO	0.57	0.50	0.50	0.67	2.50	5.83	1.00
OnT	SNOMED (F-64)	0.65	0.50	0.67	0.83	1.50	2.83	1.00
OnT	SNOMED (M-32)	0.57	0.33	0.83	1.00	2.50	2.33	1.00
OnT	SNOMED (M-64)	0.60	0.33	0.83	1.00	2.00	2.17	1.00
OnT	SNOMED (M-128)	0.60	0.33	0.83	1.00	2.00	2.17	1.00

Table 9. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on the disorder subset (6 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.42	0.25	0.50	0.67	3.50	18.58	0.92
OnT	ANATOMY	0.62	0.50	0.58	0.83	1.50	4.58	1.00
OnT	GO	0.32	0.17	0.33	0.50	5.50	11.92	1.00
OnT	SNOMED (F-64)	0.53	0.42	0.67	0.67	2.50	32.92	0.92
OnT	SNOMED (M-32)	0.79	0.67	0.92	0.92	1.00	9.50	1.00
OnT	SNOMED (M-64)	0.83	0.75	0.92	0.92	1.00	12.00	0.92
OnT	SNOMED (M-128)	0.88	0.83	0.92	0.92	1.00	16.58	0.92

Table 10. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on body structure (20 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.57	0.35	0.80	0.80	2.00	14.25	0.95
OnT	ANATOMY	0.58	0.45	0.65	0.75	2.00	26.65	0.95
OnT	GO	0.48	0.30	0.60	0.65	2.00	24.90	0.95
OnT	SNOMED (F-64)	0.51	0.35	0.65	0.70	2.00	67.30	0.90
OnT	SNOMED (M-32)	0.52	0.40	0.50	0.80	3.50	14.65	0.95
OnT	SNOMED (M-64)	0.51	0.40	0.50	0.70	3.00	18.70	0.95
OnT	SNOMED (M-128)	0.54	0.45	0.55	0.65	2.50	32.05	0.95

Table 11. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on morphicological abnormalities examples (6 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.59	0.33	0.83	1.00	2.00	2.33	1.00
OnT	ANATOMY	0.78	0.67	1.00	1.00	1.00	1.67	1.00
OnT	GO	0.54	0.33	0.67	0.67	2.00	5.17	1.00
OnT	SNOMED (F-64)	0.57	0.50	0.67	0.67	2.00	1111.33	0.83
OnT	SNOMED (M-32)	0.54	0.50	0.50	0.50	5.00	11.50	1.00
OnT	SNOMED (M-64)	0.57	0.50	0.67	0.67	2.00	9.33	1.00
OnT	SNOMED (M-128)	0.57	0.50	0.67	0.67	2.00	8.83	1.00

Table 12. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on combined negative examples (24 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.48	0.29	0.58	0.75	2.50	10.96	0.96
OnT	ANATOMY	0.61	0.46	0.67	0.79	2.00	3.83	1.00
OnT	GO	0.42	0.29	0.42	0.58	4.50	24.75	0.96
OnT	SNOMED (F-64)	0.58	0.46	0.62	0.71	2.00	17.96	0.96
OnT	SNOMED (M-32)	0.67	0.50	0.83	0.92	1.50	6.12	1.00
OnT	SNOMED (M-64)	0.70	0.54	0.83	0.92	1.00	7.67	0.96
OnT	SNOMED (M-128)	0.73	0.58	0.88	0.92	1.00	9.58	0.96

Table 13. Testing the ANATOMY assumption by assessing retrieval performance of OOV entity mentions measured on combined positive (anatomical) examples (26 Queries)

Model	Variant	MRR	H@1	H@3	H@5	Med	MR	R@100
OnT	GALEN	0.57	0.35	0.81	0.85	2.00	11.50	0.96
OnT	ANATOMY	0.62	0.50	0.73	0.81	1.50	20.88	0.96
OnT	GO	0.49	0.31	0.62	0.65	2.00	20.35	0.96
OnT	SNOMED (F-64)	0.52	0.38	0.65	0.69	2.00	308.23	0.88
OnT	SNOMED (M-32)	0.52	0.42	0.50	0.73	3.50	13.92	0.96
OnT	SNOMED (M-64)	0.53	0.42	0.54	0.69	2.50	16.54	0.96
OnT	SNOMED (M-128)	0.55	0.46	0.58	0.65	2.50	26.69	0.96

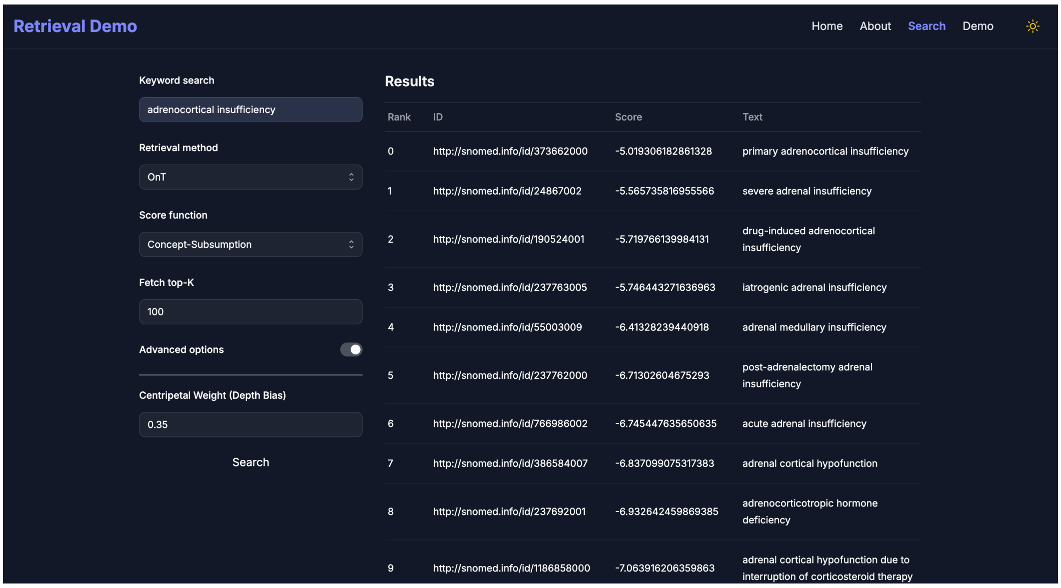
F.0.1 Discussion

The ablation experiments seem to confirm that performance is sensitive to ontology signature overlap. For instance, ANATOMY scores higher than SNOMED CT-tuned models for anatomy-specific queries, especially on morphological abnormalities. However, this effect does not generalise across unrelated hierarchies such as substances or procedures (where SNOMED CT tuned models score higher), suggesting that improvements are driven more by signature overlap, rather than by robust cross-ontology generalisation.

G System Demonstration

To gain further insight into potential applications of hyperbolic bi-encoder-based ontology embeddings (HiT and OnT), such as in web-based knowledge retrieval and retrieval augmented generation, a system demonstration is constructed. The demonstration implements both web-based knowledge retrieval via a simple user interface and includes a prototype RAG pipeline for biomedical question-answering. We provide screenshots of the retrieval system, showcased within the supporting video, as linked to within the provided GitHub.

G.1 Web-based Search & knowledge retrieval



Keyword search: adrenocortical insufficiency

Retrieval method: OnT

Score function: Concept-Subsumption

Fetch top-K: 100

Advanced options

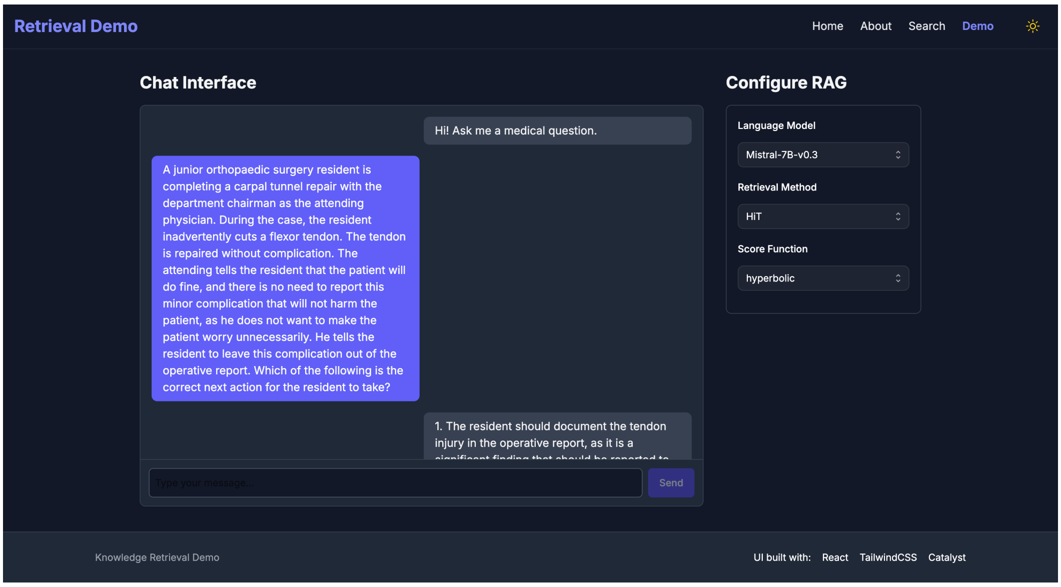
Centripetal Weight (Depth Bias): 0.35

Results

Rank	ID	Score	Text
0	http://snomed.info/id/373662000	-5.019306182861328	primary adrenocortical insufficiency
1	http://snomed.info/id/24867002	-5.565735816955566	severe adrenal insufficiency
2	http://snomed.info/id/190524001	-5.719766139984131	drug-induced adrenocortical insufficiency
3	http://snomed.info/id/237763005	-5.746443271636963	iatrogenic adrenal insufficiency
4	http://snomed.info/id/55003009	-6.41328239440918	adrenal medullary insufficiency
5	http://snomed.info/id/237762000	-6.71302604675293	post-adrenalectomy adrenal insufficiency
6	http://snomed.info/id/766986002	-6.745447635650635	acute adrenal insufficiency
7	http://snomed.info/id/386584007	-6.837099075317383	adrenal cortical hypofunction
8	http://snomed.info/id/237692001	-6.932642459869385	adrenocorticotropic hormone deficiency
9	http://snomed.info/id/1186858000	-7.063916206359863	adrenal cortical hypofunction due to interruption of corticosteroid therapy

Fig. 17. A screenshot of OnT-based knowledge retrieval conducted via a React-based front-end, and a FastAPI wrapped back-end.

G.2 Biomedical RAG MCQA



Chat Interface

Hi! Ask me a medical question.

A junior orthopaedic surgery resident is completing a carpal tunnel repair with the department chairman as the attending physician. During the case, the resident inadvertently cuts a flexor tendon. The tendon is repaired without complication. The attending tells the resident that the patient will do fine, and there is no need to report this minor complication that will not harm the patient, as he does not want to make the patient worry unnecessarily. He tells the resident to leave this complication out of the operative report. Which of the following is the correct next action for the resident to take?

1. The resident should document the tendon injury in the operative report, as it is a significant finding that should be reported to

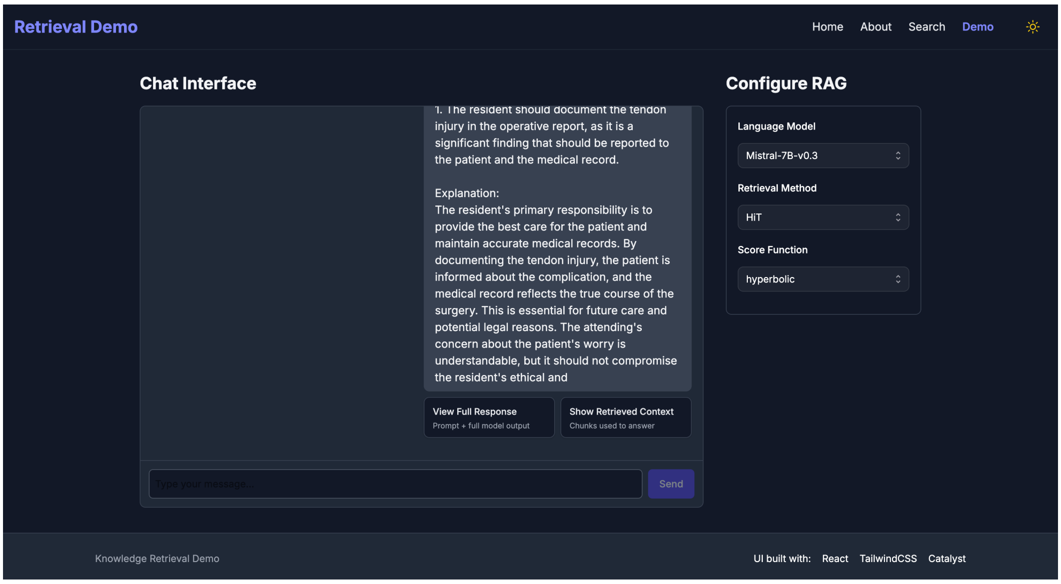
Configure RAG

Language Model: Mistral-7B-v0.3

Retrieval Method: HiT

Score Function: hyperbolic

Fig. 18. A screenshot of ontology embedding-based retrieval augmented generation conducted via a React-based front-end, and a FastAPI wrapped back-end (1).



1. The resident should document the tendon injury in the operative report, as it is a significant finding that should be reported to the patient and the medical record.

Explanation:
The resident's primary responsibility is to provide the best care for the patient and maintain accurate medical records. By documenting the tendon injury, the patient is informed about the complication, and the medical record reflects the true course of the surgery. This is essential for future care and potential legal reasons. The attending's concern about the patient's worry is understandable, but it should not compromise the resident's ethical and

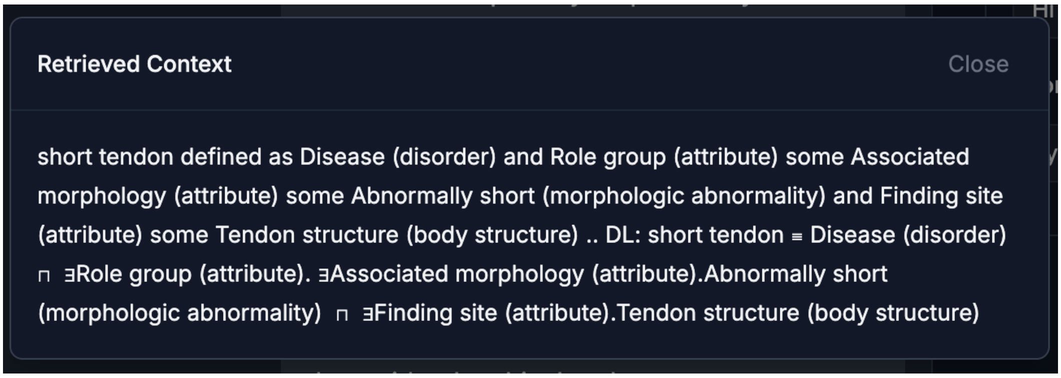
Configure RAG

Language Model: Mistral-7B-v0.3

Retrieval Method: HiT

Score Function: hyperbolic

Fig. 19. A screenshot of ontology embedding-based retrieval augmented generation conducted via a React-based front-end, and a FastAPI wrapped back-end (2).



short tendon defined as Disease (disorder) and Role group (attribute) some Associated morphology (attribute) some Abnormally short (morphologic abnormality) and Finding site (attribute) some Tendon structure (body structure) .. DL: short tendon ≡ Disease (disorder) ⊓ ∃Role group (attribute). ∃Associated morphology (attribute).Abnormally short (morphologic abnormality) ⊓ ∃Finding site (attribute).Tendon structure (body structure)

Fig. 20. Associated fetched context, provided to the user for fast verification during ontology-embedding based RAG.