Retrieve, Then Classify: Corpus-Grounded Automation of Clinical Value Set Authoring

Sumit Mukherjee SUMITMUKHERJEE2@GMAIL.COM
Oracle, USA

Nairwita Mazumder NAIRWITA.MAZUMDER@ORACLE.COM
Oracle, USA

Juan Shu JUANSHU30@GMAIL.COM
Oracle, USA

O. Tate Kernell TATE.KERNELL@ORACLE.COM
Oracle, USA

Celena A. Wheeler CELENA.WHEELER@ORACLE.COM
Oracle, USA

Chris Sidey-Gibbons DRCGIBBONS@GMAIL.COM@ORACLE.COM
Oracle, USA

Abstract

Clinical value set authoring — the task of identifying all codes in a standardized vocabulary that define a clinical concept — is a recurring bottleneck in clinical quality measurement and phenotyping. A natural approach is to prompt a large language model (LLM) to generate the required codes directly, but structured clinical vocabularies are large, version-controlled, and not reliably memorized during pretraining. We propose Retrieval-Augmented Set Completion (RASC): retrieve the K most similar existing value sets from a curated corpus to form a candidate pool, then apply a classifier to each candidate code. Theoretically, retrieve-and-select can reduce statistical complexity by shrinking the effective output space from the full vocabulary to a much smaller retrieved candidate pool. We demonstrate the utility of RASC on 11,803 publicly available VSAC value sets, constructing the first large-scale benchmark for this task. A cross-encoder fine-tuned on SAPBert achieves AUROC 0.852 and value-set-level F1 0.298, outperforming a simpler three-layer Multilayer Perceptron (AUROC 0.799, F1 0.250) and both reduce the number of irrelevant candidates per true positive from 12.3 (retrieval-only) to approximately 3.2 and 4.4 respectively. Zero-shot GPT-4o achieves value-set-level F1 0.105, with 48.6% of returned codes absent from VSAC entirely. This performance gap widens with increasing value set size, consistent with RASC's theoretical advantage. We observe similar performance gains across two other classifier model types, namely a cross-encoder initialized from pre-trained SAPBert and a LightGBM model, demonstrating that RASC's benefits extend beyond a single model class. The training code code and split manifest are publicly available at https://github.com/mukhes3/RASC.

1. Introduction

Clinical value set authoring is the task of identifying, from a controlled medical vocabulary, all codes that define a given clinical concept — for example, every SNOMED-CT and ICD-10-CM code that constitutes "Type 2 Diabetes Mellitus" for use in a quality measure. Value sets are a

foundational infrastructure of clinical quality measurement and phenotyping National Library of Medicine (2013), yet authoring them remains largely manual: a clinical informaticist must search a vocabulary of 10^5–10^6 codes, draw on domain expertise to judge relevance, and produce a list that is both complete and precise. The Value Set Authority Center (VSAC) now hosts over $10,000$ such sets, with creation rates accelerating, making the scalability of manual authoring an increasingly practical concern.

The same structure — construct a target subset $S^* \subseteq \mathcal{U}$ from a large discrete universe $\mathcal{U}$ given a concept query — appears in gene panel construction Genomics England (2019) and systematic review inclusion O'Mara-Eves et al. (2015), where the universe is a gene catalogue or a paper database respectively. What these tasks share is that $\mathcal{U}$ is large, structured, and version-controlled; the target S^* is small relative to $\mathcal{U}$; a corpus of related expert-curated sets already exists; and errors carry real downstream consequences.

A natural approach is to prompt a large language model (LLM) to generate S^* directly from the concept name. This is appealing in its simplicity, but clinical code identifiers are not natural language — they are structured, version-controlled strings that are not reliably memorized from pretraining data. We find empirically that GPT-4o (chosen due to it's widespread use) returns codes absent from VSAC entirely at a 48.6% rate, consistent with prior work on LLM reliability in clinical NLP Sivarajkumar et al. (2024). There is also a more domain specific limitation: prompting an LLM from scratch ignores the fact that most domains already possess a corpus of prior expert curation that encodes structured knowledge about $\mathcal{U}$. For instance, when a clinical informaticist authors a new value set for "Type 2 Diabetes Without Complications," the codes they need are largely a near-subset of codes already assembled in existing value sets for related diabetic concepts.

To overcome these, we propose *retrieval-augmented set completion* (RASC), a two-stage framework that exploits this signal directly. In Stage 1, semantic similarity search retrieves the K most similar existing sets from the corpus, forming a candidate pool $\mathcal{C} \subseteq \mathcal{U}$ that is small relative to $\mathcal{U}$ but enriched for members of S^*. In Stage 2, a lightweight binary classifier decides, for each item in $\mathcal{C}$, whether it belongs to S^*. This reduces an intractable generation problem over $|\mathcal{U}|$ items to a tractable classification problem over a pool of a few hundred candidates.

This paper makes three contributions. First, we formalize the set completion problem and the RASC framework, and provide a learning-theoretic analysis showing that RASC's sample complexity scales with $\log K$ rather than $\log N$, and derive an explicit sample complexity gap. Second, we construct the first large-scale benchmark for clinical value set authoring and demonstrate that a cross-encoder substantially outperforms both the retrieval-only baseline and zero-shot LLM generation across all value set types and sizes. Finally, we isolate the contribution of retrieval grounding from learned classification via a pool-grounded LLM generator.

Generalizable Insights about Machine Learning in the Context of Healthcare

The core generalizable insight is that large expert-curated corpora can be exploited as structured retrieval indices to convert intractable generation problems into tractable classification problems. This principle applies wherever a domain has: (1) a large, version-controlled vocabulary; (2) small target subsets; and (3) an existing corpus of prior expert curation.

2. Related Work

2.1. Retrieval-augmented generation.

Retrieval-augmented generation (RAG) Lewis et al. (2020) conditions a generative model on retrieved documents to ground outputs in evidence. RASC differs in a crucial way: retrieval in RAG provides *context* for generation, while retrieval in RASC provides the *feasible set* itself. The classifier is not conditioned on retrieved text; it operates over retrieved *items*, and the retrieved pool hard-constrains the output space. This distinction has non-trivial theoretical consequences: in RAG, the generative model still samples from $\mathcal{U}$; in RASC, the output is guaranteed to be a subset of $\mathcal{C}$.

2.2. Set prediction and subset selection.

Set prediction Zhang et al. (2019) and multi-label classification Tsoumakas and Katakis (2007) are closely related problems. Our formulation is distinct in that the label space is not fixed across examples: each query q induces a different candidate pool $\mathcal{C}(q)$, and the classifier must generalize across pools. This makes our setting closer to extreme multi-label classification Bhatia et al. (2015) over dynamic label sets, with the additional structure that negatives are structured (items drawn from proximal sets) rather than randomly sampled.

2.3. Automated clinical coding and value sets.

Automated ICD coding from clinical documents Mullenbach et al. (2018); Huang et al. (2022) is superficially similar but fundamentally different: it assigns codes to documents, not constructs definitional sets of codes. Value set quality and maintenance Steele et al. (2017); Mo et al. (2015) and clinical phenotyping Kirby et al. (2016); Yu et al. (2018) are related downstream applications. To our knowledge no prior work has framed value set construction as a learning problem or constructed a benchmark dataset for it.

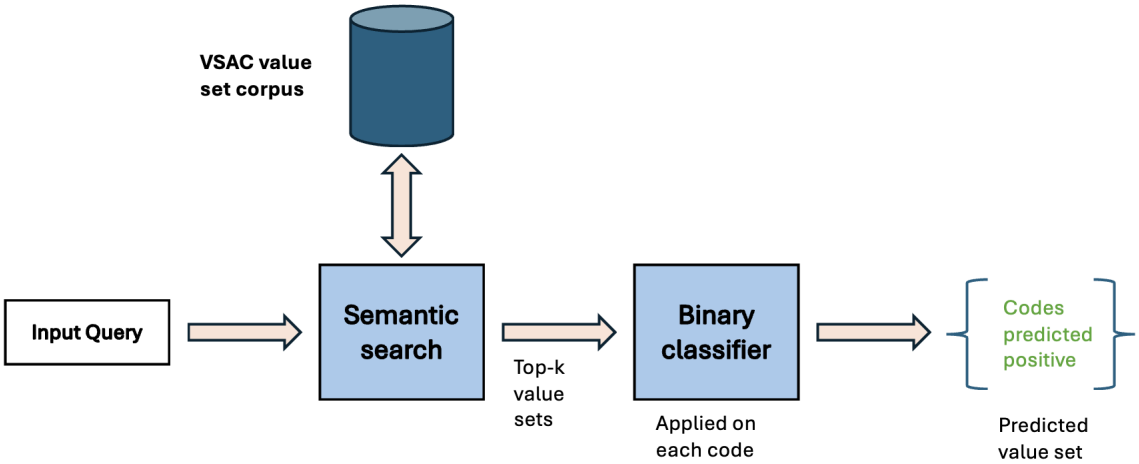

Figure 1: Overview of RASC workflow.

3. Statistical Motivation for RASC

Let $\mathcal{U}$ denote the universe of (code, system) pairs with $|\mathcal{U}| = N$. For a query q (a value set title), the task is to identify the target label set $Y(q) \subseteq \mathcal{U}$, where $|Y(q)| \ll N$ since each value set contains only a small fraction of the full ontology (see Figure 1 for visual overview). We assume a latent relevance score $\theta_q : \mathcal{U} \to \mathbb{R}$ and threshold τ such that $c \in Y(q)$ if and only if $\theta_q(c) \geq \tau$, with a margin $\gamma > 0$ separating members from non-members. A learned estimator $\hat{\theta}_n$ approximates θ_q from n training examples, with sub-Gaussian score errors that concentrate uniformly over any finite candidate set.

The primary comparator of RASC is direct generation of value sets via LLMs. To compare them mathematically, we specify two predictors. *Direct prediction* $\hat{Y}_{\text{dir}}(q)$ thresholds $\hat{\theta}_n$ over all of $\mathcal{U}$. *Retrieval-restricted prediction* $\hat{Y}_{\text{RASC}}(q)$ first retrieves a pool $\mathcal{C}(q)$ with $|\mathcal{C}(q)| \leq K \ll N$ and thresholds only within that pool. Exact recovery is impossible when $Y(q) \not\subseteq \mathcal{C}(q)$, which occurs with probability ε_{ret}.

Corollary 1 (Sample complexity gap; proof in Appendix D) *Fix $\delta \in (0, 1)$. Direct prediction achieves $\Pr(\hat{Y}_{\text{dir}}(q) \neq Y(q)) \leq \delta$ whenever*

$$n \;\geq\; \frac{2\sigma^2}{\gamma^2} \log \frac{2N}{\delta}.$$

Retrieval-restricted prediction achieves $\Pr(\hat{Y}_{\text{RASC}}(q) \neq Y(q)) \leq \delta$ whenever $\delta > \varepsilon_{\text{ret}}$ and

$$n \;\geq\; \frac{2\sigma^2}{\gamma^2} \log \frac{2K}{\delta - \varepsilon_{\text{ret}}}.$$

When ε_{ret} is small and $K \ll N$, RASC requires only $O((\sigma^2/\gamma^2) \log K)$ samples to reach a target recovery probability, versus $O((\sigma^2/\gamma^2) \log N)$ for direct prediction. In clinical vocabulary settings N/K is expected to be very large, so this reduction in log-complexity is substantial. The ε_{ret} term sets a hard floor on achievable error that no classifier can overcome regardless of n; when retrieval recall is high this floor is low, and the estimation advantage of RASC dominates. It is worth noting here that the comparison presented here is not specific to auto-regressive language models, but rather to a simplified direct generator model for the sake of tractability.

4. Methods

4.1. VSAC Corpus

We bulk-downloaded all publicly available value sets from the Value Set Authority Center (VSAC) National Library of Medicine (2013) via the HL7 FHIR `ValueSet/$expand` API under a free UMLS license, paginating through the `ValueSet` search endpoint to enumerate all OIDs. The resulting corpus, after filtering out value-sets containing less than three codes, contains **11,803 value sets** spanning 15 terminology systems and 847 publisher organizations. Value set sizes follow a heavy-tailed distribution (median 9 codes; 95th percentile 312 codes); 80.4% carry no human-authored description; over 80% draw codes from a single system.

4.2. Semantic Retrieval

We embed value set titles using SAPBert Liu et al. (2021), a BERT encoder pretrained on UMLS synonym pairs that maps biomedical entity mentions to a metric space robust to surface-form variation. We use CLS-token pooling throughout, instantiating the model explicitly with `pooling_mode_cls_token=True` to ensure consistent behavior across index construction and query-time inference. Robustness to typographic variation is validated empirically: case-perturbed title queries achieve 98.5% top-1 accuracy on a held-out sample of 200 value sets.

All 11,803 title embeddings are stored in a FAISS `IndexFlatIP` index Johnson et al. (2021); at this scale exact search completes in under 5 ms per query. We embed titles only, as 80.4% of value sets lack descriptions, making description-augmented embeddings inconsistent at inference time.

For a target value set v^* with title t^*, we retrieve the $K = 10$ most similar value sets by cosine similarity (excluding v^* itself) and take their union of codes as the candidate pool $\mathcal{C}(v^*)$, collapsing duplicate $(\text{code}, \text{system})$ pairs by retaining the highest-scoring entry. Each candidate c inherits a similarity score $s_c \in [-1, 1]$ from its source value set. Retrieval coverage is quantified by

$$\text{RR@}K(v^*) = \frac{|\,\text{codes}(v^*) \cap \mathcal{C}(v^*)\,|}{|\,\text{codes}(v^*)\,|}, \tag{1}$$

which upper-bounds downstream classifier recall.

4.3. Train/Validation/Test Splits

The splitting unit is the value set: all candidate pairs from a given value set are assigned to the same split, preventing label leakage across concepts. We apply a two-level stratification scheme. First, the two largest publishers—Clinical Architecture ($n = 596$) and CSTE Steward ($n = 590$)—are held out entirely for test, evaluating cross-organization generalization to stewards unseen during training. The remaining value sets are divided 70/15/15 (train/val/test) by stratified sampling on (`type`, `publisher_bin`), preserving the joint type–publisher distribution across splits. Value sets with fewer than three member codes or missing expansions are excluded prior to split assignment. Table 1 reports pair-level counts after candidate-pool construction.

4.4. Feature Representation

Each example is a (value set, candidate code) pair. The feature vector $\mathbf{x} \in \mathbb{R}^{1545}$ concatenates: (1) the SAPBert CLS embedding of the value set title $\mathbf{e}_{t^*} \in \mathbb{R}^{768}$; (2) the SAPBert CLS embedding of the candidate display name $\mathbf{e}_{d_c} \in \mathbb{R}^{768}$; (3) a one-hot code system indicator over eight systems (SNOMED-CT, ICD-10-CM, RxNorm, LOINC, CPT, ICD-10-PCS, HCPCS, OTHER); and (4) the retrieval similarity score s_c. The title embedding is identical for all candidates evaluated against the same target; the code system indicator encodes structural constraints absent from the semantic embeddings (e.g. a SNOMED-CT value set should not contain RxNorm codes). Embeddings are computed once over all unique strings prior to feature assembly—7,052 unique titles and 202,573 unique display names—and cached, reducing encoder calls proportionally to the deduplication factor.

4.5. Classification Models

Code inclusion is posed as binary classification: given $\mathbf{x}$, estimate $\hat{y} = P(c \in \text{codes}(v^*) \mid \mathbf{x})$. Class imbalance is addressed via weighted binary cross-entropy, decision threshold tuned to maximize F1 on the validation set.

LightGBM. A gradient-boosted tree classifier trained on the 1545-dimensional feature vector, serving as a computationally efficient non-neural reference.

MLP. A three-hidden-layer network (1545→512→256→64→1) with batch normalization, ReLU activations, and dropout ($p = 0.3$) after each layer (941k parameters). Training uses AdamW (lr $= 3\times10^{-4}$, weight decay 10^{-4}) with `ReduceLROnPlateau` scheduling, batch size 2048, and early stopping on validation loss (patience 5).

Cross-encoder. We fine-tune SAPBert (110M parameters) as a cross-encoder by presenting the value set title and candidate display name as a single sentence pair:

$$\texttt{[CLS]}\ t_1 \cdots t_m\ \texttt{[SEP]}\ d_1 \cdots d_n\ \texttt{[SEP]}, \tag{2}$$

truncated to 128 tokens. The `[CLS]` representation is passed to a linear classification head $\hat{y} = \sigma(\mathbf{w}^\top \mathbf{z} + b)$. Unlike the MLP, which interacts title and code only through concatenation of independently computed embeddings, the cross-encoder attends jointly across all title and code tokens through 12 transformer layers, enabling fine-grained token-level interactions that the bi-encoder feature space cannot represent. Training uses AdamW (lr $= 2\times10^{-5}$, weight decay 10^{-2}) with linear warmup over 6% of steps, gradient accumulation over 4 steps (effective batch size 64), fp16 mixed precision, and gradient checkpointing. The cross-encoder was trained for 3 epochs.

4.6. Evaluation

Baselines. The *retrieval-only* baseline is uses the same first-stage retrieval as RASC i.e for a target value set, we retrieve the top-K most similar value sets by title embedding, take the union of their codes as the candidate pool, and predict every retrieved candidate as positive without applying a second-stage classifier.

The *LLM baseline* evaluates GPT-4o zero-shot on the full test set: each value set is presented with only its name and allowed code systems, with no access to the candidate pool or VSAC corpus. Outputs are matched via exact (code, system) pair matching after normalizing system URIs to canonical short forms.

Metrics. All methods are evaluated at the *value-set level*: precision, recall, and F1 are computed per value set and macro-averaged across the test set, providing a unified frame for classifiers and GPT-4o alike. For classifiers we additionally report code-pair-level AUROC and average precision (AP), which are threshold-free. For the LLM baseline we report a *hallucination rate*: the fraction of returned codes absent from the full test corpus, indicating fabricated identifiers rather than merely incorrect code assignments. Post-hoc stratified analyses are conducted across value set type, size, and publisher; precision is reported separately on the RR@$K = 1.0$ subset , where retrieval is perfect and any precision deficit is attributable to the model.

Table 1: Pair-level split statistics after candidate-pool construction. The lower positive rate in the test set reflects the two held-out publisher organizations (Clinical Architecture and CSTE Steward), whose value sets tend to be smaller and more specialized, yielding lower retrieval recall and a smaller fraction of true codes in the candidate pool.

Split	Examples	Positives	Pos. Rate	Publisher source
Train	3,520,599	432,851	12.3%	In-distribution
Validation	802,303	103,657	12.9%	In-distribution
Test	2,011,009	151,630	7.5%	In-dist. + held-out

4.7. Comparing LLM-as-Classifier with Zero-Shot Generation

To isolate the contribution of retrieval grounding from the contribution of learned classification, we compare two GPT-4o conditions on a cost-constrained subsample of the test set. The zero-shot generation condition is described in Section 4.6; here we introduce a complementary *LLM-as-classifier* condition in which the retrieved candidate pool is provided directly in context.

We restrict this comparison to value sets with $|S^*| \leq 50$ true codes, for which presenting the full candidate pool in-context is tractable, and draw a stratified random sample of 100 value sets across clinical types. For each value set, the model receives the value set title, the allowed code systems, and the full candidate pool as a list of `(code, system, display)` triples, and is asked to return the subset it judges relevant. The system prompt is otherwise identical to the zero-shot generation prompt (Appendix C), with the addition of an instruction to select only from the provided candidates.

This condition is not practical at scale: candidate pools of 100–600 codes produce prompts of 1,000–6,000 tokens per value set, making inference roughly two orders of magnitude more expensive than the MLP and fundamentally limited to small value sets by context length constraints.

5. Results

5.1. Dataset Statistics

Table 1 reports pair-level example counts after candidate-pool construction across all value sets. The positive rate is 7.5% on the test set versus approximately 12% on train and validation, reflecting the held-out publishers: Clinical Architecture and CSTE Steward author smaller, more specialized value sets with lower retrieval overlap against the in-distribution corpus. The overall ratio of roughly 1 positive per 13 candidates motivates the classification stage: reducing this review burden is the central objective.

5.2. Classifier Results

Table 2 reports code-pair-level test performance. All three classifiers substantially outperform the retrieval-only baseline on precision and F1, and performance improves monotonically from LightGBM to MLP to Cross-Encoder across all threshold-free metrics.

Table 2: Code-pair-level test set performance. Decision thresholds are tuned on the validation set.

Model	AUROC	Avg. Prec.	Precision	Recall	F1
Retrieval-only	—	—	0.092	1.000	0.140
LightGBM	0.745	0.238	0.173	0.316	0.190
MLP	0.799	0.271	0.226	0.470	0.250
Cross-Encoder	0.852	0.374	0.272	0.483	0.298

Table 3: Value-set-level test performance (macro averages).

Model	Precision	Recall	F1	SE(F1)
Retrieval-only	0.092	0.553	0.136	0.003
LightGBM	0.173	0.316	0.190	0.005
MLP	0.226	0.470	0.250	0.005
Cross-Encoder	0.272	0.483	0.298	0.006
GPT-4o (zero-shot)	0.169	0.100	0.105	0.004

Precision–recall tradeoff. The retrieval-only baseline presents approximately 13 irrelevant candidates per true positive; LightGBM reduces this ratio while retaining 31.6% of true codes, and the MLP improves further to 47.0% recall at higher precision. The Cross-Encoder achieves the best performance on all metrics (AUROC 0.852, AP 0.374, F1 0.298), representing gains of +0.053 AUROC and +0.103 AP over the MLP. The precision gain (+0.046) is substantially larger than the recall difference (+0.013), indicating that the cross-encoder's joint attention over (title, code) token pairs primarily reduces false positives rather than recovering additional true codes missed by the MLP.

5.3. Value-Set Level Results

Table 3 reports macro-averaged value-set-level results, placing all methods on a common footing. The Cross-Encoder achieves the highest F1 (0.298) and precision (0.272), followed by the MLP (0.250), LightGBM (0.190), GPT-4o (0.105), and the retrieval-only baseline (0.136).

GPT-4o's recall collapses to 0.100 — well below the retrieval-only baseline (0.553) — with high variance (SE = 0.004, $\mu = 0.105$), indicating that correct returns are concentrated in a small fraction of value sets. Of all the codes returned by GPT-4o across all value sets, 48.6% are absent from the full test corpus, confirming hallucination as the dominant failure mode for zero-shot generation over structured clinical vocabularies.

5.3.1. Stratification by value set type.

Figure 2a shows F1 by value set type. The Cross-Encoder leads in every category, with the largest absolute gains over the MLP on Condition/Diagnosis (+0.078) and Lab/Observation (+0.060). GPT-4o remains competitive on Condition/Diagnosis (F1 0.277) and Procedure

(0.236) — categories where concepts are expressible in natural language and code sets are compact — but degrades to near zero on Medication (F1 0.003) and Condition/Clinical (0.029), which require enumerating large numbers of RxNorm and SNOMED-CT codes not reliably memorized from pretraining.

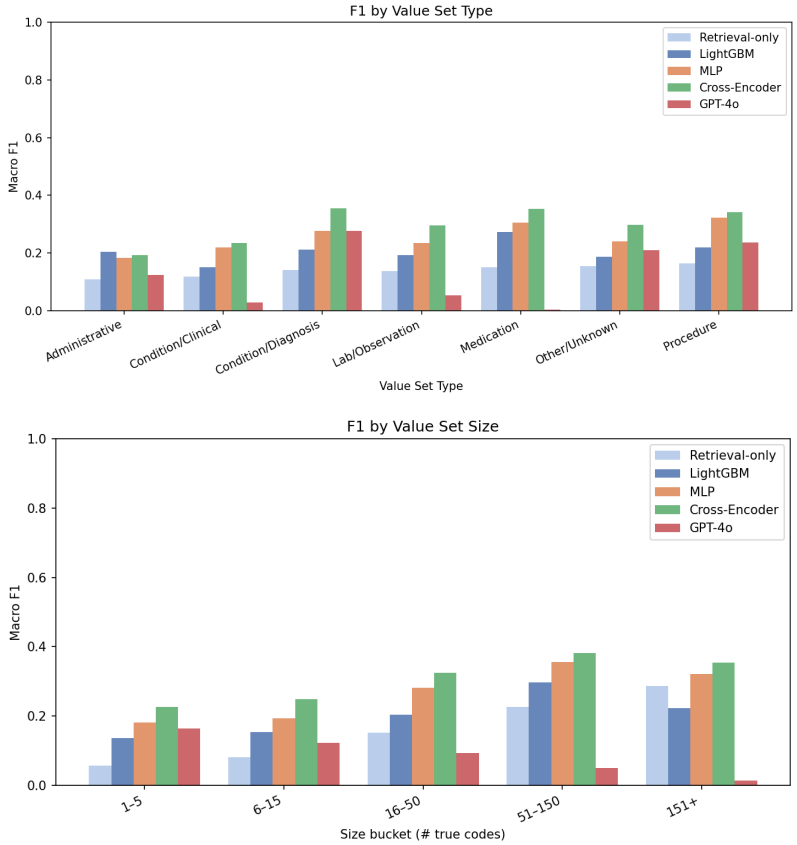

Figure 2: Stratified value-set-level F1 across clinical type (a) and value set size (b).

5.3.2. Stratification by value set size.

Figure 2b reveals a crossover between GPT-4o and the classifiers as a function of value set size. For very small value sets (1–5 true codes), GPT-4o (F1 0.164) is competitive with LightGBM (0.137) and approaches the MLP (0.182) and Cross-Encoder (0.226). Beyond 15 codes, GPT-4o degrades monotonically (F1 0.014 for sets with >150 codes), while all three classifiers improve with set size; the Cross-Encoder reaches F1 0.354 on the largest sets. This shows that RASC's estimation advantage over direct generation grows with target set size, as the pool-constrained classification problem becomes increasingly tractable relative to unconstrained generation over $|U| \approx 10^5$–10^6 identifiers.

5.3.3. Precision at full retrieval coverage.

On the 905 value sets where RR@$K = 1.0$ (candidate pool contains every true code), any precision deficit is attributable entirely to the model. Macro-averaged precision on this subset is: Cross-Encoder 0.438, MLP 0.410, LightGBM 0.349, GPT-4o 0.199, and retrieval-only 0.160. The gap between the Cross-Encoder and the retrieval-only baseline quantifies the classification value added under ideal retrieval, and suggests that further gains are achievable through improved retrieval coverage of the remaining value sets.

5.4. Zero-Shot LLM vs. LLM-as-Classifier

The zero-shot GPT-4o results in Table 4 leave open whether the classifier–LLM gap reflects a fundamental limitation of LLM-based generation or simply the absence of retrieval grounding. To isolate this, we provide GPT-4o with the full candidate pool in-context and re-evaluate on a stratified subsample of 100 value sets with $|S^*| \leq 50$ (to keep pool-sized prompts tractable in number of tokens).

Table 4: GPT-4o zero-shot vs. pool-grounded, macro-averaged over 95 value sets with $|S^*| \leq 50$.

Condition	Precision	Recall	F1
GPT-4o zero-shot	0.131	0.086	0.089
GPT-4o w/ pool	0.284	0.456	0.294

Pool grounding triples F1 (Table 4), with the gain driven almost entirely by recall. GPT-4o recognises relevant codes when shown them but cannot generate them reliably from memory. This suggests that LLM based generation can be tractable if the LLM is treated as a classifier grounded in a pool of retrieved codes. However, such an approach will likely not be as scalable or cost-effective as using a classifier model.

6. Benchmark Release

We provide code to reproduce the VSAC data used for model training and evaluation, to support reproducible research on corpus-grounded set completion (currently in anonymized form). Since VSAC data is distributed under a UMLS license that precludes direct re-distribution, we provide a download script that accepts a UMLS API key, bulk-fetches all value set expansions via the FHIR `ValueSet/$expand` endpoint, and reconstructs the corpus used in this paper. A split manifest—containing OIDs, split assignments, RR@K scores, value set type, and publisher for all 11,803 value sets—is released as a lightweight index (no VSAC content) that users can match against their local download to recover identical train/validation/test splits. The candidate pool construction pipeline then produces the labeled `(value set, candidate code)` pairs from the downloaded corpus, with deterministic retrieval and de-duplication ensuring reproducible datasets across users. The embedding-model-agnostic design means any representation—dense embeddings, sparse retrievers, or hand-crafted features—can be evaluated against the same labels. The training code code and split manifest are publicly available at https://github.com/mukhes3/RASC.

7. Conclusion

We introduced RASC, a two-stage framework for set completion over large discrete vocabularies that replaces de novo generation with retrieval followed by binary classification. The key observation is that expert-curated set corpora, present in most knowledge-intensive domains, can be leveraged to reduce an intractable generation problem over $|\mathcal{U}|$ items to a tractable classification problem over a much smaller candidate pool. Under standard margin and concentration assumptions, this reduction yields a sample complexity gap of $O(\log K)$ versus $O(\log N)$, a condition verified empirically: the classifier advantage over zero-shot generation grows monotonically with value set size, directly consistent with the theoretical prediction.

On 11,803 VSAC value sets, RASC classifiers substantially outperform zero-shot GPT-4o generation, which returns codes absent from VSAC entirely at a 48.6% rate and achieves value-set-level F1 of 0.105. Providing GPT-4o with the retrieved candidate pool in-context closes most of this gap on small value sets, suggesting that retrieval grounding might enable better quality LLM based generation. However, such an approach would suffer from higher cost and poor scalability. Cross-organization generalization to two entirely held-out publisher stewards further suggests that the learned representations reflect clinically meaningful structure rather than publisher-specific authoring patterns.

Retrieval quality remains the binding constraint: on value sets with perfect retrieval coverage (RR@$K = 1.0$), the cross-encoder achieves precision 0.438, suggesting that classification is effective when the pool is adequate and that retrieval improvement is the most promising direction for further gains. The framework relies on related expert-curated value sets being present in the retrieval corpus. If no closely related value sets exist, including for a new concept in a rapidly evolving domain, the candidate pool can have low recall and the classifier cannot recover codes that were never retrieved. Future work will look at extending RASC to other application domains such as gene panel construction and systematic review inclusion, where the same structural conditions hold: large structured vocabularies, small target sets, and existing curated collections that encode prior expert judgment. We release code to generate the identical VSAC training and evaluation data to enable the use of a consistent benchmark for future efforts in this direction.

References

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 730–738, 2015.

Genomics England. PanelApp: A publicly available gene panel repository. https://panelapp.genomicsengland.co.uk, 2019. Accessed 2025.

Chao-Wei Huang, Shang-Chi Tsai, and Hen-Hsen Chen. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.clinicalnlp-1.2.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

Jacqueline C Kirby, Peter Speltz, Luke V Rasmussen, Melissa Basford, Omri Gottesman, Peggy L Peissig, et al. PheKB: a catalog and workflow for creating electronic phenotype algorithms for transportability. *Journal of the American Medical Informatics Association*, 23(6):1046–1052, 2016. doi: 10.1093/jamia/ocv202.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474, 2020.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4228–4238. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.334.

Huan Mo, William K Thompson, Luke V Rasmussen, Jennifer A Pacheco, Guoqian Jiang, Richard Kiefer, Qian Zhu, Jie Xu, Elizabeth Montague, Lemuel R Waitman, et al. Desiderata for computable representations of electronic health records-driven phenotype algorithms. *Journal of the American Medical Informatics Association*, 22(6):1220–1230, 2015. doi: 10.1093/jamia/ocv112.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1101–1111. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1100.

National Library of Medicine. Value set authority center (VSAC). https://vsac.nlm.nih.gov, 2013. Accessed 2025.

Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic Reviews*, 4(1):5, 2015. doi: 10.1186/2046-4053-4-5.

Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing. *Journal of the American Medical Informatics Association*, 31(9):1935–1945, 2024. doi: 10.1093/jamia/ocae122.

Nathan A Steele, Sanjukta Bhattacharyya, Manasi Bhagwat, Allyn Morales, Jay R Desai, and Abel N Kho. Quality and consistency of VSAC value sets for clinical quality measurement. *Journal of the American Medical Informatics Association*, 24(4):716–722, 2017. doi: 10.1093/jamia/ocw183.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007. doi: 10.4018/jdwm.2007070101.

Sheng Yu, Abhishek Chakrabortty, Yu-Han Ho, Bertrand Hidalgo, Joshua C Denny, Judith Blessing, et al. PheNorm: A gene-regularized topic model for unsupervised EHR phenotyping. *Journal of the American Medical Informatics Association*, 25(1):54–60, 2018. doi: 10.1093/jamia/ocx111.

Xingyou Zhang, Yaron Lipman, and Honglak Lee. Deep set prediction networks. 32, 2019.

Appendix A. VSAC Corpus Analysis

We present an exploratory analysis of the entire **unfiltered** VSAC corpus to characterise the data distribution and motivate several design decisions in the main paper.

A.1. Temporal Growth

Figure 3 shows value set creation and update activity by year. Growth has been substantial and accelerating: fewer than 100 value sets were added in each of 2012 and 2013, rising to approximately 1,750 per year in 2023–2024. More value sets were created in those two years alone than in the entire period 2012–2020 combined, reflecting both the expansion of eCQM reporting requirements and the growing adoption of FHIR-based quality measurement.

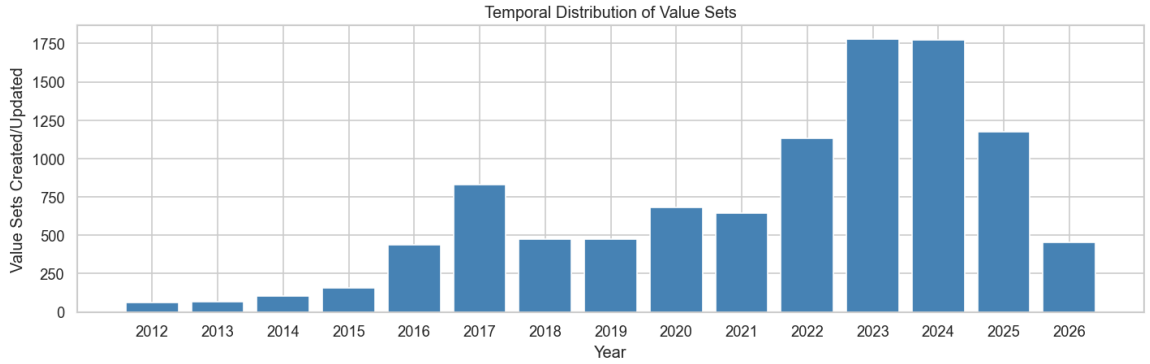

Figure 3: Temporal distribution of VSAC value sets by year of creation or last update. Growth accelerates sharply after 2021, reaching approximately 1,750 value sets per year in 2023–2024.

A.2. Terminology System and Value Set Type

Figure 4 shows the distribution of value sets by primary code system. SNOMED-CT dominates with approximately 4,250 value sets, followed by ICD-10-CM (∼2,500), LOINC (∼1,500), and RxNorm (∼1,300). Critically, the right panel shows that approximately 8,800 of the ∼10,700 analysed value sets draw codes from a single terminology system, with roughly 1,000 spanning two systems and approximately 350 spanning three.

Figure 5 shows the inferred semantic type distribution alongside value set size stratified by type. Condition/Clinical and Condition/Diagnosis together account for over 60% of the corpus. Medication value sets exhibit the largest size variance, with a heavy tail exceeding 200 codes. Lab/Observation value sets are comparatively compact. This size heterogeneity motivates stratified evaluation across types.

A.3. Description Coverage

Figure 6 shows the fraction of value sets carrying a human-authored textual description, and the distribution of description lengths among those that do. Only 19.6% of value sets include a description field; the remaining 80.4% provide a title alone. Among value sets with

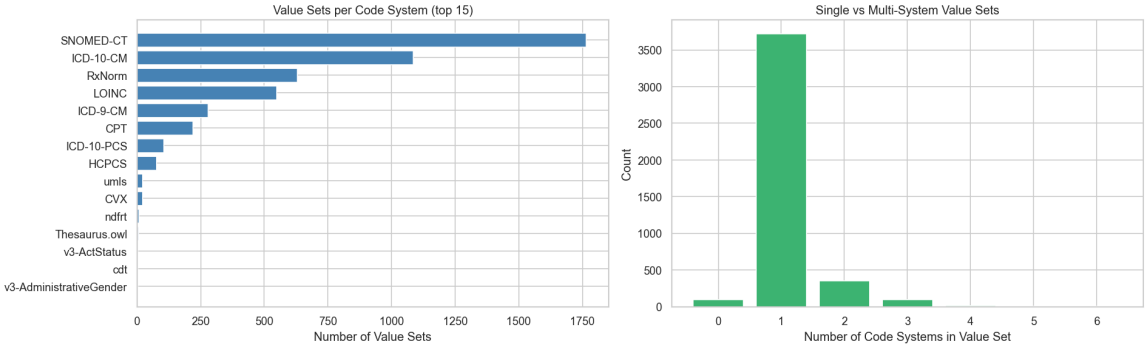

Figure 4: Left: number of value sets per code system (top 15). SNOMED-CT and ICD-10-CM together account for over 60% of the corpus. Right: distribution of the number of distinct code systems per value set; over 80% of value sets are single-system.

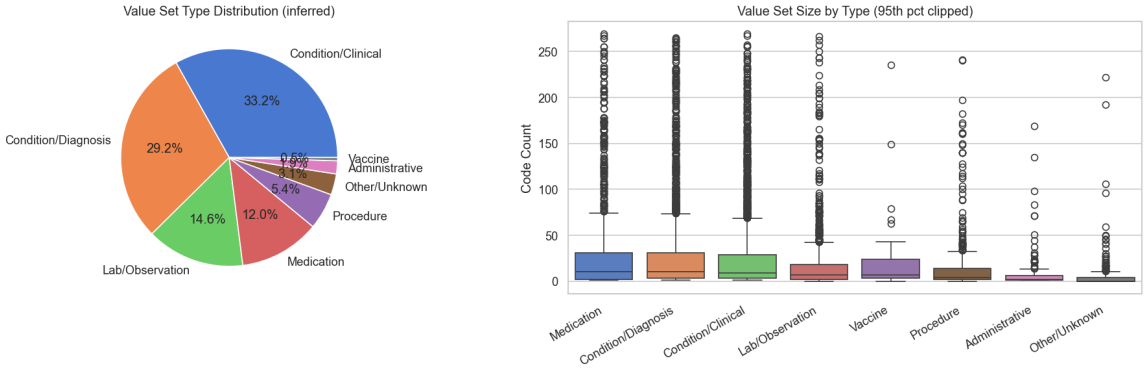

Figure 5: Left: inferred value set type distribution. Condition types dominate the corpus. Right: value set size (code count) by type, clipped at the 95th percentile. Medication value sets exhibit substantially higher size variance than other types.

descriptions, lengths are right-skewed, peaking between 5 and 15 words, with most consisting of brief definitional phrases rather than detailed clinical narrative. This near-absence of descriptions has a direct methodological consequence: augmenting title embeddings with description text would produce an inconsistent representation across the corpus, since no description signal exists for four out of five value sets at inference time. We therefore embed titles only.

A.4. Publisher Distribution

Figure 7 shows the top 20 publisher organizations by value set count. The distribution is highly concentrated: CSTE Steward alone contributes approximately over 1500 value sets, followed by NCQA PHEMUR , Clinical Architecture , and HL7 Patient Care WG , with a long tail of hundreds of smaller organizations across more than 800 distinct publishers in total.

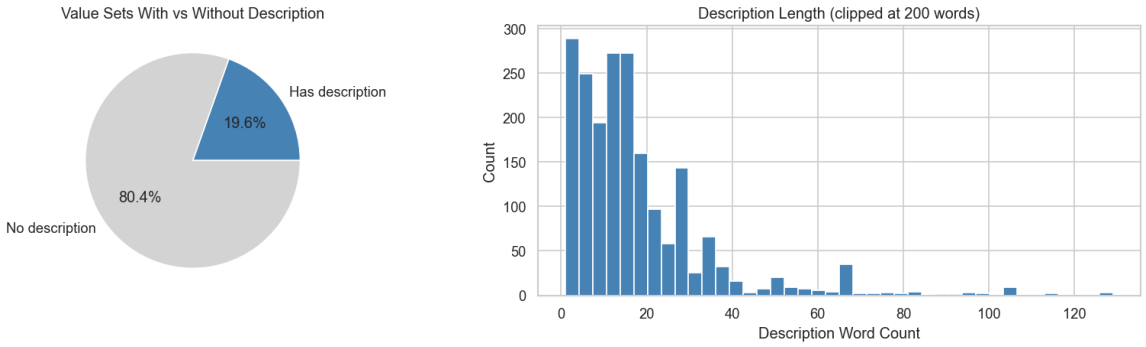

Figure 6: Left: fraction of value sets with a human-authored textual description. Only 19.6% carry one. Right: distribution of description word count among value sets that have a description; most are brief (under 20 words).

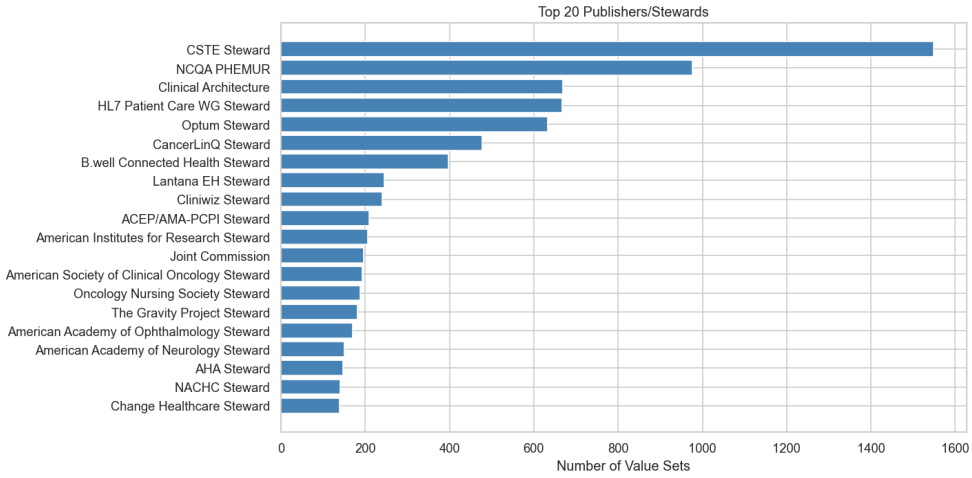

Figure 7: Top 20 VSAC publisher organizations by number of value sets. The distribution is highly concentrated; the top 5 publishers account for a disproportionate share of the corpus.

Appendix B. Additional Retrieval Analyses

B.1. Modern embedding retrieval baselines

We repeated the retrieval-only evaluation at $K = 10$ with three additional pretrained embedding models. Table 5 reports macro value-set F1. Qwen3-Embedding-0.6B performs best among these retrieval-only models, but remains below classifier-based RASC, indicating that the benefit of retrieval followed by classification is not specific to SapBERT retrieval.

B.2. Sensitivity to the number of retrieved value sets

We evaluated $K \in \{5, 10, 15\}$ on the held-out test split using the fixed cross-encoder setup. As shown in Table 6, retrieval recall increases with K, but the candidate pool grows substantially and retrieval-only macro F1 decreases. Thus, larger K improves coverage at the cost of presenting many more irrelevant candidates to the second stage.

Table 5: Retrieval-only performance with alternative embedding models at $K = 10$.

Embedding model	Macro value-set F1
Qwen3-Embedding-0.6B	0.1406
MedCPT	0.1300
SapBERT	0.1258
bge-m3	0.1139

Table 6: Sensitivity to the number of retrieved value sets on the held-out test split.

K	Retrieval recall@K	Mean candidates	Retrieval-only F1
5	0.4895	292.5	0.1942
10	0.5535	536.8	0.1357
15	0.5840	775.3	0.1034

Appendix C. GPT-4o Prompt

Each value set is evaluated with a single zero-shot call. The system prompt and user message are shown below.

System prompt.

```
You are a clinical informaticist expert in medical terminologies including
SNOMED CT, ICD-10-CM, RxNorm, LOINC, ICD-10-PCS, CPT, HCPCS, and NCI Thesaurus.

Your task is to generate the expansion of a clinical value set given its name
and the allowed code systems.  A value set expansion is a list of standardised
clinical codes that define a specific clinical concept.

Return ONLY a JSON array.  Each element must be an object with exactly these
fields:  ‘‘code’’ (the exact code identifier), ‘‘system’’ (one of the allowed
systems listed for this value set), and ‘‘display’’ (the human-readable name
for the code).

Only use codes from the allowed systems listed for this value set.  Use real,
valid codes --- do not invent codes.  Be comprehensive but precise.  Return
ONLY the JSON array, no explanation or markdown fences.
```

User message.

```
Value set name:  {title}
Allowed code systems:
  - {system_1}
  - {system_2}
⋮

Return the value set expansion as a JSON array of {"code", "system", "display"}
objects.
```

Temperature was set to 0 for deterministic outputs and `max_tokens` to 4,096. Allowed code systems were inferred from the candidate pool rather than specified a priori, ensuring the

prompt reflects only information available at inference time. Responses were parsed as JSON arrays; bare single-object responses were wrapped into a list, and system URI strings were normalized to canonical short forms prior to matching (e.g. `http://www.nlm.nih.gov/research/umls/rxnorm` → `RxNorm`).

Appendix D. Proof of the Sample Complexity Gap

We provide the formal setup and proof for Corollary 1. The analysis compares direct open-world prediction over the full universe $\mathcal{U}$ against retrieval-restricted prediction over a candidate pool $\mathcal{C}(q) \subseteq \mathcal{U}$.

Assumption 2 (Sparse labels) *For every query q, $|Y(q)| \le s$ with $s \ll N$.*

Assumption 3 (Margin) *There exists $\gamma > 0$ such that for every query q, $\theta_q(c) \ge \tau + \gamma$ for all $c \in Y(q)$, and $\theta_q(c) \le \tau - \gamma$ for all $c \notin Y(q)$.*

Assumption 4 (Retrieval coverage and candidate size) *For each query q, the retriever outputs $\mathcal{C}(q)$ with $|\mathcal{C}(q)| \le K \ll N$ and $\Pr(Y(q) \subseteq \mathcal{C}(q)) \ge 1 - \varepsilon_{\text{ret}}$.*

Assumption 5 (Uniform score estimation) *There exists $\sigma > 0$ such that for every query q, every finite $A \subseteq \mathcal{U}$, and every $t > 0$,*

$$\Pr\left(\sup_{c \in A} |\hat{\theta}_n(q,c) - \theta_q(c)| > t \;\middle|\; q\right) \le 2|A| \exp\left(-\frac{nt^2}{2\sigma^2}\right).$$

This is a standard finite-class union-bound concentration: if estimation errors are sub-Gaussian with variance σ^2/n for each code c individually, a union bound over A yields the above, with the cost of uniformity growing only as $\log|A|$.

For any $A \subseteq \mathcal{U}$, define the thresholded predictor $\hat{Y}_A(q) := \{c \in A : \hat{\theta}_n(q,c) \ge \tau\}$. The direct and retrieval-restricted predictors are then $\hat{Y}_{\text{dir}}(q) := \hat{Y}_{\mathcal{U}}(q)$ and $\hat{Y}_{\text{RASC}}(q) := \hat{Y}_{\mathcal{C}(q)}(q)$ respectively.

Theorem 6 (Exact recovery probability) *Under Assumptions 2–5, for every query q,*

$$\Pr\big(\hat{Y}_{\text{dir}}(q) \ne Y(q) \mid q\big) \;\le\; 2N \exp\left(-\frac{n\gamma^2}{2\sigma^2}\right),$$

and

$$\Pr\big(\hat{Y}_{\text{RASC}}(q) \ne Y(q) \mid q\big) \;\le\; \varepsilon_{\text{ret}}(q) \;+\; 2K \exp\left(-\frac{n\gamma^2}{2\sigma^2}\right),$$

where $\varepsilon_{\text{ret}}(q) := \Pr(Y(q) \not\subseteq \mathcal{C}(q) \mid q)$. Averaging over q,

$$\Pr\big(\hat{Y}_{\text{RASC}}(q) \ne Y(q)\big) \;\le\; \varepsilon_{\text{ret}} \;+\; 2K \exp\left(-\frac{n\gamma^2}{2\sigma^2}\right).$$

Proof We first analyze $\hat{Y}_A(q)$ for a generic finite set $A \subseteq \mathcal{U}$. If $\sup_{c\in A} |\hat{\theta}_n(q,c) - \theta_q(c)| < \gamma$, then for every $c \in Y(q) \cap A$ we have $\hat{\theta}_n(q,c) \geq \theta_q(c) - \gamma \geq \tau$, and for every $c \in A \setminus Y(q)$ we have $\hat{\theta}_n(q,c) \leq \theta_q(c) + \gamma \leq \tau$. Hence $\hat{Y}_A(q) = Y(q) \cap A$ on this event, so

$$\Pr\big(\hat{Y}_A(q) \neq Y(q) \cap A \mid q\big) \;\leq\; \Pr\left(\sup_{c\in A} |\hat{\theta}_n(q,c) - \theta_q(c)| \geq \gamma \mid q\right) \;\leq\; 2|A| \exp\left(-\frac{n\gamma^2}{2\sigma^2}\right),$$

where the last inequality applies Assumption 5 with $t = \gamma$.

Taking $A = \mathcal{U}$ and using $Y(q) \cap \mathcal{U} = Y(q)$ gives the bound for $\hat{Y}_{\mathrm{dir}}(q)$.

For $\hat{Y}_{\mathrm{RASC}}(q)$, exact recovery additionally requires the event $E_q := \{Y(q) \subseteq \mathcal{C}(q)\}$. By the law of total probability,

$$\Pr\big(\hat{Y}_{\mathrm{RASC}}(q) \neq Y(q) \mid q\big) \;\leq\; \Pr(E_q^c \mid q) \;+\; \Pr\big(\hat{Y}_{\mathcal{C}(q)}(q) \neq Y(q) \cap \mathcal{C}(q) \mid E_q, q\big).$$

By Assumption 4, $\Pr(E_q^c \mid q) = \varepsilon_{\mathrm{ret}}(q)$. On E_q we have $Y(q) \cap \mathcal{C}(q) = Y(q)$, and since $|\mathcal{C}(q)| \leq K$, the generic bound above gives

$$\Pr\big(\hat{Y}_{\mathcal{C}(q)}(q) \neq Y(q) \cap \mathcal{C}(q) \mid E_q, q\big) \;\leq\; 2K \exp\left(-\frac{n\gamma^2}{2\sigma^2}\right).$$

Combining and averaging over q completes the proof. ■

Theorem 6 separates the two error sources: an approximation term $\varepsilon_{\mathrm{ret}}$ from retrieval misses, and an estimation term controlled by K rather than N. Corollary 1 follows immediately by solving each bound for n.

Proof [Proof of Corollary 1] Setting $2N \exp(-n\gamma^2/2\sigma^2) \leq \delta$ and solving for n gives the direct prediction condition. For RASC, on the event E_q the estimation term satisfies $2K \exp(-n\gamma^2/2\sigma^2) \leq \delta - \varepsilon_{\mathrm{ret}}$; solving for n gives the stated condition. ■