A Configuration-Only, Sharable Pipeline for Stable Zero-Shot CDM Grounding of NLP Targets (That You Can Run on a Pretty Good Laptop)

Georgina Kennedy[1,2,3], Jared Houghtaling[4,5], Robert Miller[6], Fahim Alam[1,2,7], Lois Holloway[1,2,7,8], Tim Churches[1,2], Winston Liauw [2,3,8]

1. Ingham Institute for Applied Medical Research, 2. University of New South Wales, School of Clinical Medicine, Sydney, Australia 3. Maridulu Budyari Gumal (SPHERE) Cancer Clinical Academic Group 4. Tufts University School of Medicine – Institute for Clinical Research and Health Policy Studies (ICRHPS), Boston, MA, USA 5. Tufts Clinical and Translational Sciences Institute (CTSI), Boston, MA, USA 6. Miller Data Solutions 7. Liverpool and Macarthur Cancer Therapy Centre, Liverpool, Australia 8. Institute for Medical Physics, School of Physics, The University of Sydney, Australia 9. Cancer Care Centre, St George Hospital, Kogarah, Australia

Background

While LLMs can extract useful concepts from text with minimal prompting, converting those outputs into structured, semantically meaningful representations fit for integration with downstream analyses remains a challenging task. This is particularly evident in environments where the cost, governance, and reproducibility constraints of fine-tuning or large model deployment are unachievable.

This work was motivated by the need for a practical, stable, and lightweight approach to clinical text grounding (i.e. identifying underlying concepts within unstructured text) that satisfies several real-world constraints. Specifically, the processing pipeline must:

- Deliver accurate and stable zero-shot grounding of clinical text to OMOP-standardised terms, with no opportunity for hallucination or unpredictable model output.
- Respect domain-specific constraints and relationships, including task-dependent preferences for term sets and concept hierarchies.
- Operate reliably on typical professional-grade computers, ideally without requiring dedicated GPUs (i.e. models should be no larger the 3–7B parameter class).
- Maintain sufficiently high-level abstraction such that more powerful or bespoke models can be deployed where resourcing and throughput/reasoning demands allow.
- Support configuration-driven reuse and portability, allowing new targets and vocabularies to be specified declaratively without any need for retraining, and furthermore supporting the sharing of validated configurations without re-development.
- Run entirely in environments with heavily restricted inbound and outbound access, linking only to locally hosted models, vocabularies, and configuration resources.

Taken together, these requirements reflect the operational realities of many clinical and health system settings, where heavily restricted trusted research environments, limited computational resources, and demanding requirements for model accuracy, traceability and stability are common.

Methods

SPIRES (Structured Prompt Interrogation and Recursive Extraction of Semantics)[3] is a knowledge extraction approach that performs zero-shot schema-guided information retrieval from flexible prompts. The backbone of the SPIRES knowledge engine is the definition of LinkML extraction templates[4], which act as both prompts and output validators through generated Pydantic classes (Figure 1). This enables the language model to generate outputs that are directly aligned with downstream structured data requirements, improving stability, accuracy, and integration with controlled vocabularies. In this work, we describe an updated version of the OntoGPT SPIRES implementation. This update delivered two key enhancements (full description in Figure 2):

```python
# truncated snip from from RT_Region.py
class OMOPHierarchy(ConfiguredBaseModel):
    # abstract base class follows the structure of ontogpt NamedEntity base class,
    # with parent_id parameter for hierarchy management.

    linkml_meta: ClassVar[LinkMLMeta] = LinkMLMeta(
        {'abstract': True, 'from_schema': 'omop:convention'}
    )
    id: str = Field(description="A unique identifier for the named entity")
    parent_id: Optional[str] = Field(description="parent concept where grounded concept found")
    label: Optional[str] = Field(description="The label (name) of the named thing")

class BodySite(OMOPHierarchy):
    # non-abstract child class has values set for target parent concept

    linkml_meta: ClassVar[LinkMLMeta] = LinkMLMeta(
        {'annotations': { 'annotators': { 'tag': 'annotators', 'value': 'OMOP_OWL/ohdsi_test.db' },
                          'parent_id': {'tag': 'parent_id', 'value': 'omop:4190005'} },
         'from_schema': 'ohdsi:rt_region',
         'id_prefixes': ['omop']})
    id: str = ... # fields as above
```

Figure 1: Example portion of generated Pydantic class for enforcing OMOP inheritance

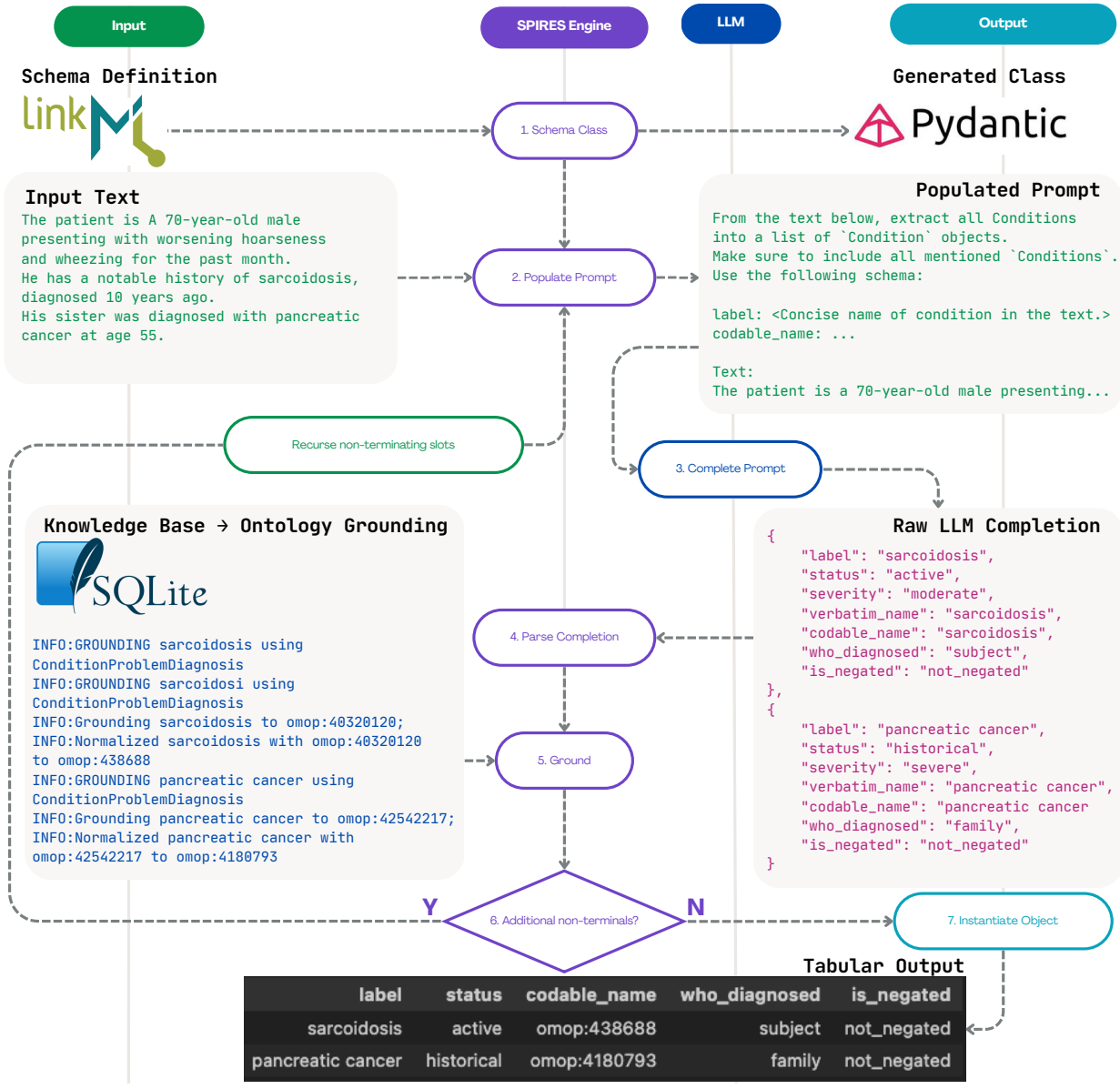


Figure 2: Pipeline - high level SPIRES functionality + modifications to OntoGPT implementation delivered by this work

1. Replacing the pseudo-YAML representation and subsequent parsing steps with the newer *Instructor* library[5]. This library also centres on Pydantic models, thus preserving compatibility with existing configurations while improving robustness and clarity.
2. A new custom grounding layer, tuned specifically to OMOP (Figures 3, 4). Source OWL files encode the ancestry, synonyms, and standard mapping relationships from the target vocabularies available in Athena in SemSQL[6] format to be queried as local ontology resources. This approach eliminates reliance on external services such as BioPortal, allowing the system to function in fully air-gapped environments and supporting tighter domain-specific constraints.

```
<owl:Class rdf:about="omop:438688">
    <rdfs:subClassOf rdf:resource="omop:4027384"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="omop:in_class"/>
            <owl:someValuesFrom rdf:resource="omop:33099"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="omop:in_domain"/>
            <owl:someValuesFrom rdf:resource="omop:19"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    ...
    <rdfs:label xml:lang="en">Sarcoidosis</rdfs:label>
    <skos:altLabel>Besnier-Boeck-Schaumann syndrome</skos:altLabel>
    <skos:altLabel>Sarcoidosis (disorder)</skos:altLabel>
    <skos:exactMatch>omop:40320120</skos:exactMatch>
    <has_code>31541009</has_code>
    <standard_concept>S</standard_concept>
</owl:Class>
```

Figure 3: Example class from CDM Vocab OWL files

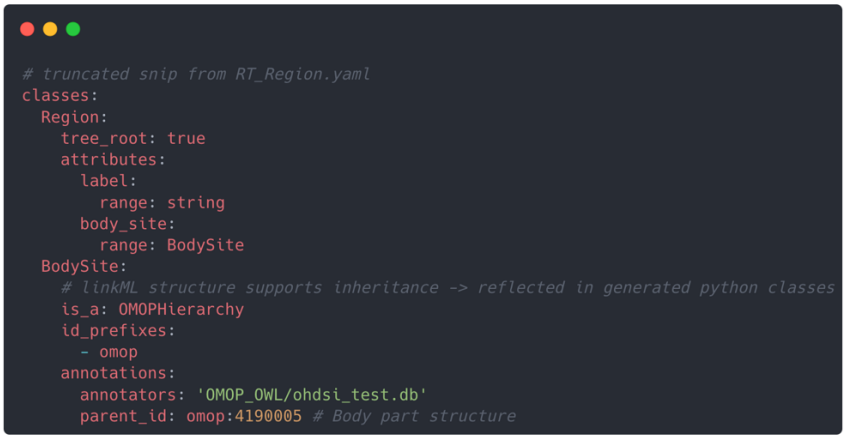
```
# during grounding, schemaview interface pulls linkML details inform targeted normalisation steps
ke.schemaview.induced_slot('body_site', 'Region') # field, class name

>> SlotDefinition({
  'name': 'body_site',
  'annotations': JsonObj(),
  'description': ('Specific body site or organ mentioned in the radiation therapy region. This '
     'should be a discrete anatomical site.  Do not use abbreviations or acronyms.'),
  'from_schema': 'ohdsi:rt_region',
  'alias': 'body_site',
  'owner': 'Region',
  'domain_of': ['Region'],
  'range': 'BodySite',
  'required': False})
```

Figure 4: Example usage of generated classes during grounding

Experiment 1: The first task was to extract and disambiguate conditions and their modifiers from synthetic clinical notes, before grounding to standard SNOMED codes. We generated fifty clinical notes using GPT-4.1, totalling 197 sentences. The LinkML template for this task was a customised version of the sample condition template within the OntoGPT library[7].

Experiment 2: Radiation therapy (RT) regions and fields are historically unstructured and can also be highly patient-specific. These elements often require custom terminology to capture precise tumour locations and geometries and rarely map to structured data fields via established standards. The associated labels are typically captured through short free-text strings, a practice which poses challenges for standardized downstream analytics. We processed these labels to produce SNOMED code for target body parts and other modifiers according to a bespoke LinkML template that defined a target proposed structure for RT treatment episodes (Figure 5).



```yaml
# truncated snip from RT_Region.yaml
classes:
  Region:
    tree_root: true
    attributes:
      label:
        range: string
      body_site:
        range: BodySite
  BodySite:
    # linkML structure supports inheritance -> reflected in generated python classes
    is_a: OMOPHierarchy
    id_prefixes:
      - omop
    annotations:
      annotators: 'OMOP_OWL/ohdsi_test.db'
      parent_id: omop:4190005 # Body part structure
```

Figure 5: Example portion of definition file for radiotherapy region used in Experiment 2

Results

Results for each task are compared for models Llama3:8b[8] and Medllama3:7b, which has been fine-tuned for general medical reasoning tasks. Other fine-tuned clinical models based on older Llama versions struggled to produce structured target data outputs, indicating that the results here are a recent development. We also compare results against MedspaCy[9] grounding via QuickUMLS, and in experiment 2, Usagi[10].

Experiment 1:

The extracted results were highly accurate with strict (exact code) accuracy of 72%, valid (accurate either coded or uncoded) accuracy of 85% and hierarchical 'close match' accuracy (allowing parent and child codes if they did not add clinically spurious details) of 96%. Structured querying of LLMs missed very few (1%) present explicit conditions and returned invalid or improperly modified conditions in 4% of otherwise accurately extracted cases. The rules-based baseline was less accurate overall (20% strict, 64% close match). For detailed results and discussion, see supplement 1.

Experiment 2:

Using the zero-shot Llama3-8b-backed SPIRES pipeline, a 'correct' standard concept in the target hierarchy was found for 85% of input labels (n=576) – or 90% when including 'valid' (n=15) or 'near match' (n=19). 5% (n=35) were populated incorrectly and 5% (n=32) were not mapped although a valid label should exist. Comparable results (84% strictly correct; 89% either correct, valid, near match) for similarly sized models demonstrate stability of results. These results significantly outperform both baselines (56% and 62% for medSpaCy and Usagi respectively). Although shorter text snippets, these results are more representative of the true capabilities of the pipeline, given their heterogeneity and realistic nature. Refer to supplementary materials 2 for more detailed results and error analysis.

Conclusion

It is possible to extract standard OMOP-grounded condition and RT region/field codes from clinical text with a high degree of accuracy and completeness in a way that is fit for use in downstream analyses. The delivered solution takes advantage of the vast semantic knowledge encoded within the OMOP vocabularies through a pragmatic integration with low to mid-weight LLMs. With thoughtfully structured LinkML task definitions that chunk tasks into appropriate targets that have a modifier depth of at most 1, this pipeline can be quickly customised for future extraction targets in a straightforward and predictable manner, without fine-tuning.

References

1. Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., & Farajtabar, M. (2025). The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. arXiv preprint arXiv:2506.06941.
2. https://github.com/AustralianCancerDataNetwork/omop-links
3. Caufield JH, Hegde H, Emonet V, Harris NL, Joachimiak MP, Matentzoglu N, et al. Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. Bioinformatics, Volume 40, Issue 3, March 2024, btae104, https://doi.org/10.1093/bioinformatics/btae104
4. Moxon SA, Solbrig H, Unni DR, Jiao D, Bruskiewich RM, Balhoff JP, et al. The Linked Data Modeling Language (LinkML): A General-Purpose Data Modeling Framework Grounded in Machine-Readable Semantics. ICBO. 2021 Sep 15;3073:148-51.
5. https://python.useinstructor.com/
6. https://github.com/INCATools/semantic-sql
7. https://github.com/monarch-initiative/ontogpt/
8. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783. 2024 Jul 31.
9. Eyre, H., Chapman, A. B., Peterson, K. S., Shi, J., Alba, P. R., Jones, M. M., ... & Patterson, O. V. (2022, February). Launching into clinical space with medspaCy: a new clinical text processing toolkit in Python. In AMIA Annual Symposium Proceedings (Vol. 2021, p. 438).
10. https://ohdsi.github.io/Usagi/

Supplementary Materials 1: Task Details for Experiment 1

Source Data

A high-specification model (GPT 4.1) was selected to generate input samples to ensure sufficient heterogeneity and realistic sample generation. These samples were generated in batches for specific target specialty domains (oncology, cardiology, gastroenterology and endocrinology). The prompt included a request to produce varied samples in terms of abbreviations, formality of structure, style and tone to avoid overly rigid and non-representative structures. The model also produced weak labels for intended conditions noted in the sample.

Example Generated Input:

Samples below show generated text and weak labels for domains oncology, endocrinology and gastroenterology respectively. Weak labels were spot-checked and found to be accurately representative of what was produced in the text. This produced 390 target conditions (189 active, 55 inactive, 5 suspected, 130 family history) across 117 notes and 332 sentences. Direct grounding of the weak labels in OMOP vocabularies under target concepts directly from OWL files was successful for 92% of the generated conditions.

note	sentence	condition	who	when
Pt w/ history of CLL now relapsed, has not yet started salvage therapy. DM type 2 on metformin, otherwise well. Family hx positive for bladder ca (father).	Pt w/ history of CLL now relapsed, has not yet started salvage therapy.	Chronic Lymphocytic Leukemia	subject	current
	DM type 2 on metformin, otherwise well.	Type 2 Diabetes	subject	current
	Family hx positive for bladder ca (father).	Bladder Cancer	family	
Pt dx hypothyroidism, on levothyroxine, stable dosing. PMHx significant for recurrent DVT (on apixaban), and appendectomy in youth. Mom dx Hashimoto's, paternal GF TIA x2.	Pt dx hypothyroidism, on levothyroxine, stable dosing.	Hypothyroidism	subject	current
	PMHx significant for recurrent DVT (on apixaban), and appendectomy in youth.	Deep vein thrombosis	subject	unknown
		Appendectomy	subject	historical
	Mom dx Hashimoto's, paternal GF TIA x2.	Hashimoto's thyroiditis	family	
		Transient ischemic attack	family	
Referred for further work-up of Crohn's disease, under care of her rheumatologist for psoriatic arthritis. She has had a cholecystectomy in 2017 for symptomatic gallstones. Brother with ulcerative colitis.	Referred for further work-up of Crohn's disease, under care of her rheumatologist for psoriatic arthritis.	Crohn's disease	subject	current
		Psoriatic arthritis	subject	current
	She has had a cholecystectomy in 2017 for symptomatic gallstones.	Cholelithiasis	subject	historical
	Brother with ulcerative colitis.	Ulcerative colitis	family	

Task Verification Guide

Domain	**Definition**
Factuality	Did the extracted condition exist in the input? Exact: identical code was extracted as per grounded condition in input data Valid: exact, or a condition was validly extracted that was not grounded in input data, or the right condition label was extracted without successful grounding. Close match: coded value was extracted that sits above (more general) the target value in the input data hierarchy or below (more specific, but not contradictory)
Completeness	Was a condition missed that should have been extracted?
Context	Were fields (specifically in this case, family history or patient) extracted with all appropriate modifiers and therefore usable in downstream analyses? For correctly extracted values only.

Additional considerations for evaluation

Factuality	If the condition could have been validly excluded (e.g. referring only to presenting symptoms), but was included in the extract, it was required to be factual and complete in context, as the processing pipeline has no way of excluding once extracted by the models.
Negation	A condition that was negated in the text could be correct either through omission or explicit negation in the extracted code set.
Exclusion	Presenting symptoms that are listed but included without a diagnosis are not counted as "missed", although it is permissible to include them as well.

Detailed Results: Llama3:8b

Speciality	**Factuality**			**Complete**	**Context**
	Exact	**Valid**	**Close**		
Oncology	0.73	0.86	0.98	0.98	0.99
Cardiology	0.70	0.80	0.98	0.98	0.98
Endocrinology	0.73	0.86	0.95	0.97	0.99
Gastroenterology	0.72	0.85	0.92	0.93	0.99

Detailed Results: MedSpaCy with QuickUMLS

Speciality	**Factuality**			**Complete**	**Context**
	Exact	**Valid**	**Close**		
Oncology	0.18	0.40	0.52	0.84	1.0
Cardiology	0.28	0.59	0.73	0.94	0.99
Endocrinology	0.17	0.56	0.67	0.87	1.0
Gastroenterology	0.19	0.65	0.80	0.94	1.0

Result Notes

- It is noteworthy that for those conditions that were successfully extracted by the rules-based system, there was near 100% accuracy of disambiguation of family vs. personal history context. An ideal solution may comprise a combination of LLM sensitivity to detect and rules-based specificity to contextualise.

Supplementary Materials 2: Task Details for Experiment 2

Source Data

The radiation therapy region name field in the Elekta MOSAIQ system is free text, although merely 20 characters long. This leads to highly abbreviated terminology pertaining to the region details only, and despite a decent 'long tail' of unique/near unique records, sufficient overlaps in convention to query distinct labels belonging to big enough groups of patients that can be used as input without any risk to patient privacy, according to local governance requirements. Site A provided 466 unique input labels, with 341 additional provided from site B, giving a total of 676 unique labels after deduplication across sites.

Pre-processing & Prompt Tuning

During prototyping, it was found that spinal regions were more successfully coded as a standalone slot in the yaml file. This was therefore split out into a standalone enum (parent: 4227378). We also performed trivial pre-processing to remove decimal numbers and perform lower case normalisation but otherwise left the labels untouched.

Task Verification Guide

Labels were reviewed by human annotators according to the following schema:

Category	**Definition**
Correct	A *preferred* target code was selected (includes terms that were correctly skipped due to being unmappable by humans)
Valid (less specific)	A valid umbrella term selected
Near match (more specific)	A close-match term – may introduce spurious specificity or slight variance from ideal target, but still clinically useful
Invalid	A code was provided however it is misleading or incorrect (includes terms that should not be mappable or that could not be mapped by a human, because these should have returned no code)
No match	No mapping where there should have been one

Detailed Results

Of the correct/valid matches for both SPIRES pipelines, all were standard codes under the correct target concept (4190005 *Body Part Structure,* 4240671 *Anatomical structure* or 4227378 *Structure of vertebral column*) in the OMOP hierarchy and therefore a complete end to end mapping. It is likely that some of the unmatched concepts would have been found with a loosened requirement for these target hierarchies.

<u>Per label results</u>

Tool	**Correct**	**Valid**	**Near match**	**Invalid**	**No match**
OMOP-SPIRES + Llama3:8b	0.85	0.02	0.03	0.05	0.05
OMOP-SPIRES + Medllama3:7b	0.84	0.01	0.04	0.06	0.04
MedSpaCy + QuickUMLS	0.56	<0.01	0.04	0.01	0.39
Usagi	0.62	0.01	0.17	0.21	0.0

Per RT record results

When we add back in frequency counts, all annotation pipelines improve in their scores. This reflects the complementary facts that (1) more common regions are noted with more regularity and (2) both models and knowledgebases tend to have higher accuracy for more recognisable regions through presence in training data and richer synonym sets respectively.

Tool	Correct	Valid	Near match	Invalid	No match
OMOP-SPIRES + Llama3:8b	0.90 (+5%)	<0.01	0.02	0.03	0.04
OMOP-SPIRES + Medllama3:7b	0.91 (+7%)	<0.01	0.01	0.03	0.05
MedSpaCy + QuickUMLS	0.65 (+9%)	<0.01	0.01	<0.01	0.32
Usagi	0.67 (+5%)	0.01	0.08	0.24	0.0

Result Notes

- These results slightly overstate the accuracy of Usagi, as there was no option for the mapper to return a list of mappings (as opposed to multiple candidate maps), and therefore these were marked as correct if at least one term mapped correctly e.g. SPIRES map of 'rt hip + femur' to '4044034 Pelvic region; 4323581 Bone structure of femur' was marked correct, whereas Usagi provided '4323581 Bone structure of femur' only, which was also accepted.
- SPIRES pipelines were held to strict valid parentage & standard constraints as described above; Usagi was restricted to Concept Class = *Spec Anatomic Site* and standard concept; MedSpaCy + QuickUMLS was mapped to one of the accepted semantic types [T029, T023, T030] (body location, body part or body space) within valid SNOMED codes.

Example error analysis showing distinct behaviours between LLM, string-level and concept-level-matching strategies

INPUT	llama3 label	IDs	evaluation	usagi label	IDs	evaluation	medspacy label	IDs	evaluation
rectum + pelvis	Pelvic region;Rectum structure	4044034;4144541	correct	PEL	4123163	invalid			no match
(l) lung	Lung structure	4213162	correct	Lung structure	4213162	correct	Entire lung	4111459	correct
ph i prostate+sv	Prostatic structure	4165732	valid	Prostatic structure	4165732	correct			no match
glottis	Glottis structure	4047227	correct	Glottis structure	4047227	correct	Entire glottis	4131315	correct
l chest wall	Chest wall structure	4193513	correct	Chest wall structure	4193513	correct	Entire chest wall	4109932	correct
lt femur	Bone structure of femur	4323581	correct	Bone structure of femur	4323581	correct	Entire bone of femur	37115374	correct
ph1 prostate	Prostatic structure	4165732	correct	Prostatic structure	4165732	correct	Entire prostate	4110208	correct
l/s spine	lumbar	4045660	correct	Structure of vertebral column	4227378	valid	Entire vertebral column	4185891	valid
ph2 prostate bed	Prostatic structure	4165732	correct	Prostatic structure	4165732	correct	Entire prostate	4110208	correct
distal oesophagus	Esophageal structure	4140098	correct	Esophageal structure	4140098	correct	Esophageal structure	4140098	correct
rtbreast+scf+imc+sib	Breast structure	4298444	valid	Breast structure	4298444	correct			no match
prostate + pelvis	Prostatic structure;Pelvic region	4165732;4044034	correct	PEL	4123163	invalid	Entire prostate, Entire pelvis	4110208, 4041832	correct
rt nasal ala			no match	Lateral nasal artery	37157433	invalid			no match
t11-l3	thoracic;lumbar	4047490;4045660	correct	ST11	4159026	invalid	Level of the eleventh thoracic vertebra	4134469	near match
t9-l3	thoracic	4047490	correct	T9-T10 rotator thoracis	4077547	invalid			no match
upper pelvis	Pelvic region	4044034	correct	PEL	4123163	invalid	Entire pelvis	4041832	correct
rt parietal	Brain structure	4133034	correct	Structure of left parietal bone	37158682	near match			no match
right pelvis	Pelvic region	4044034	correct	Structure of right renal pelvis	4184440	invalid	Entire pelvis	4041832	correct
(r) breast/low axilla	Breast structure;Axillary region structure	4298444;4238919	correct	Axillary region structure	4238919	correct	Entire breast	4108283	correct
thyroid	Thyroid structure	4321375	correct	Thyroid structure	4321375	correct	Thyroid structure	4321375	correct