babylon

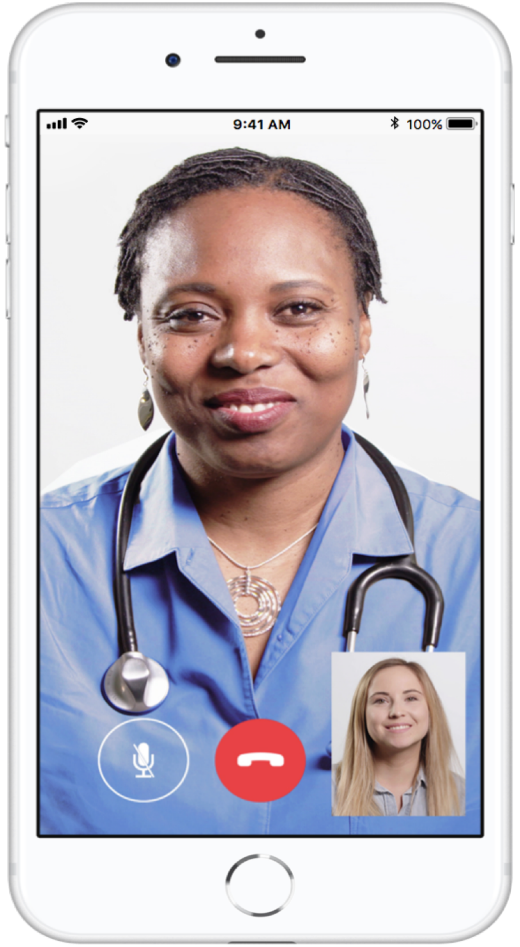
Methods and Metrics for Knowledge Base Engineering

Giorgos Stoilos, David Geleta, Szymon Wartak, Sheldon Hall, Mohammad Khodadadi
Babylon Health

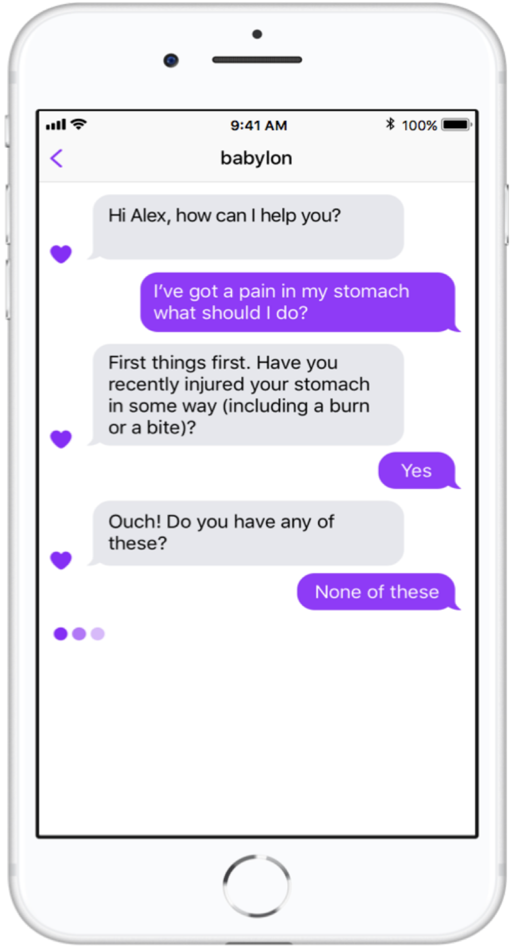
Yizheng Zhao, Ghadah Alghamdi, Renate Schmidt
University of Manchester

Babylon

- **Digital Healthcare services via a Phone App**



GP consultation
1 every minute, 24/7

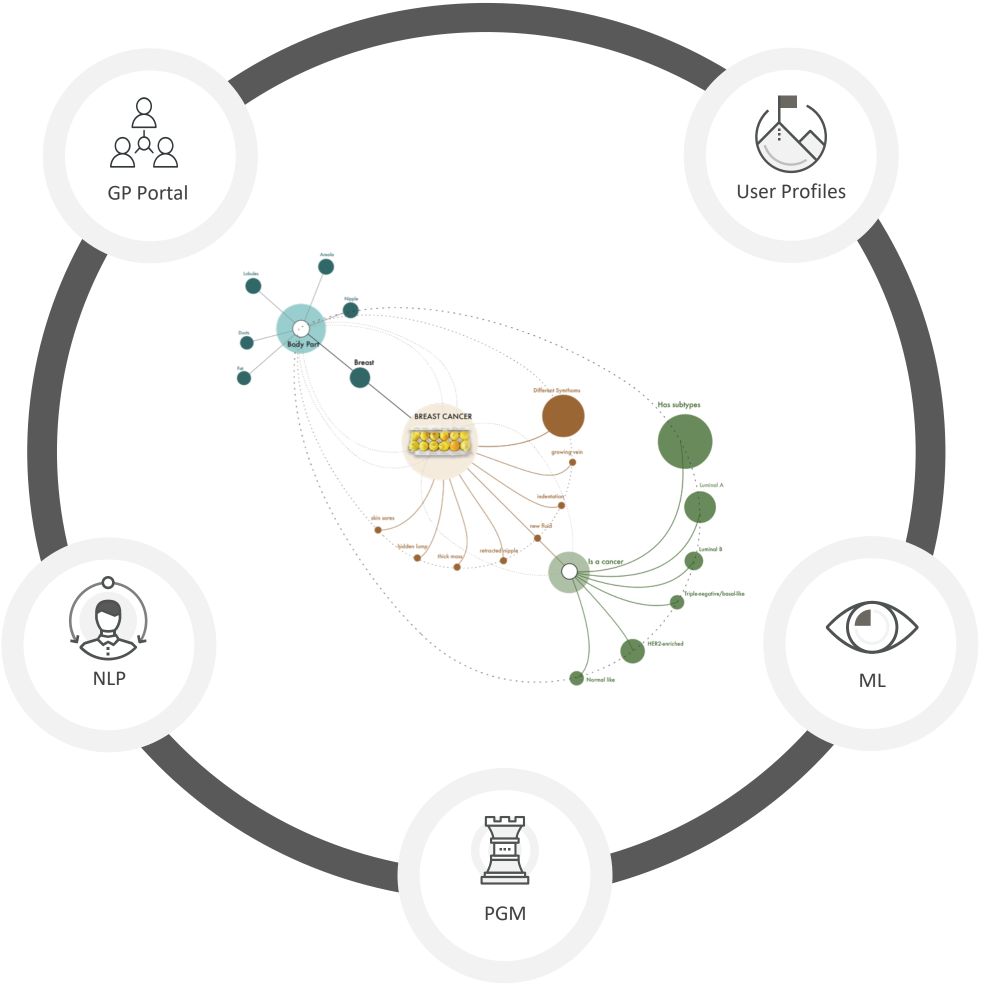


AI-based chatbot
3 interaction every minute

How is it done?

- **Various background AI-based services**
  - User text processing (NLP, NLU)
  - Intention detection, data analytics (ML)
  - Symptom Checking Engine (PGM)
  - GP-portal
  - User Profiles
- **At the core: Medical Knowledge Base**
  - Provides common vocabulary
  - Formal rich semantics
  - Standardisation (coding systems, SNOMED, …)
  - Reasoning Services [Thursday, 11th, Posters, Merrill Hall]

    [Thursday, 11th , in-use track, 14:40-15:00]

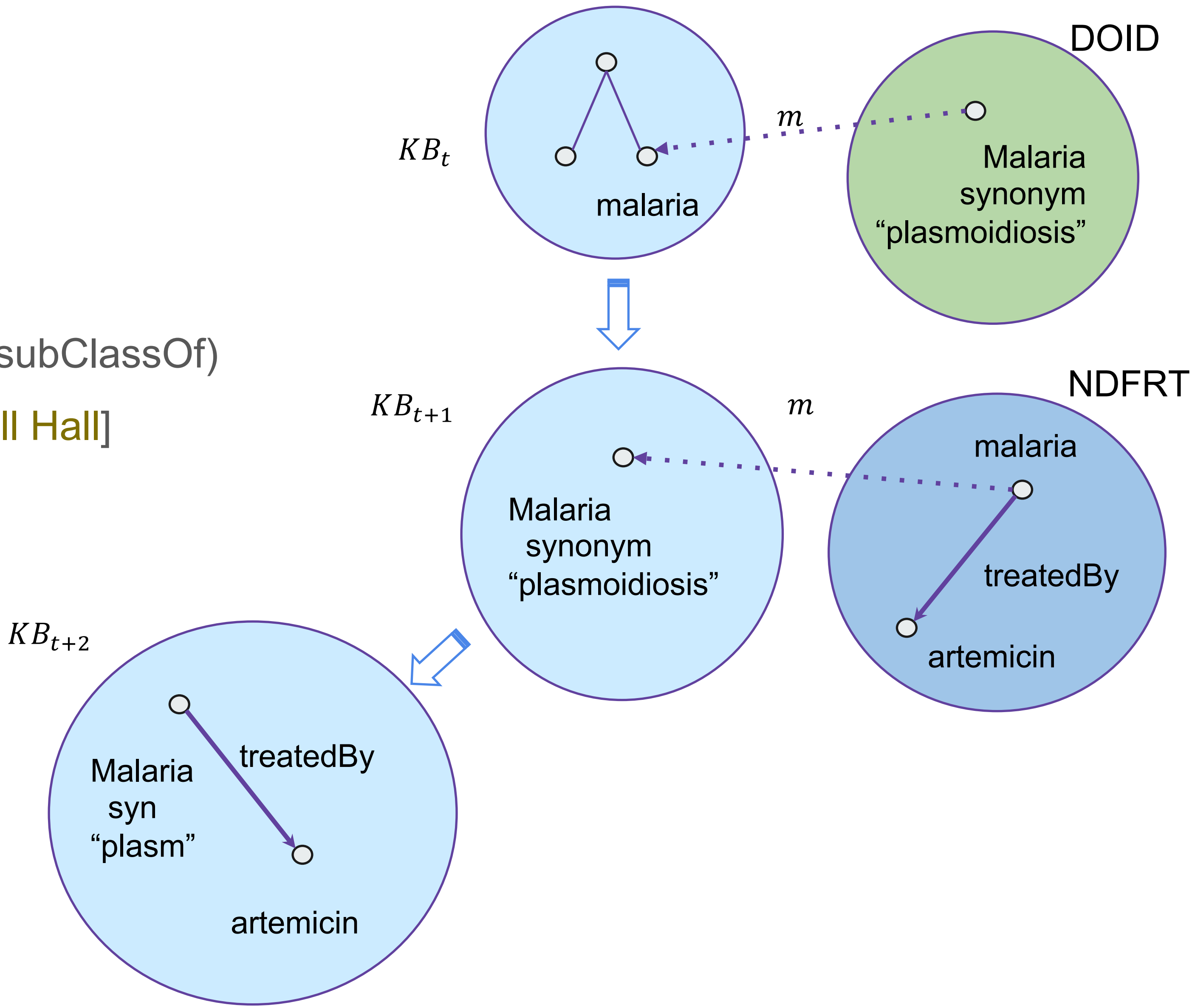


Constructing Babylon KB

- **Ontology Integration**
  - Start from a seed ontology KB_0
  - Enrich it iteratively with new sources
  - Matching (m)
  - "Copying" Axioms (labels, relations, subClassOf)

[Friday, 12th 11:40, Merrill Hall]

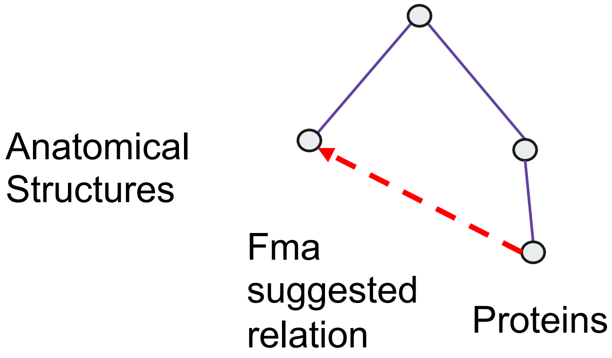
- **Information Extraction**
  - From web resources
  - Bibliography
  - Unstructured text



Problem Statement

- **Enrichment is good but can introduce**
  - Logical or structural changes
    - inconsistencies, change in service behaviour
  - Relation misuses
    - data from IE
  - Lexical changes
    - Synonym overlaps → ambiguity

which may negatively affect services



- **Goal: Monitor/analyse how KB evolves**
  - Logical, structural, lexical changes
  - Information gain after integration (did KB improve?)
  - Visualise differences, pinpoint areas of great change

All these at a great scale!!

Previous Approaches

- **Linked Data Analysis**
  - [Ngomo et al., Zaveri]: focus on data quality (labels, trust, accessibility)
  - Rashid et al.: focus on property assertion evolution.
- **Ontology Evaluation**
  - Gangemi: focus on graph-structure (paths, fan-outness, depth, etc.)
  - Vrandecic: focus on ontology domain modelling.
- **Some metrics are suitable but need custom ones**

KB Integrity

- **Coherence**

$$for\ every\ A \in KB, KB \nvDash A \sqsubseteq \bot$$

  - practical implementation using SPARQL over GraphDB:

$$no\ A\ s.t.\ KB \vDash_{rdfs} A \sqsubseteq C \sqcap D, \qquad C\ \mathbf{disjointWith}\ D$$

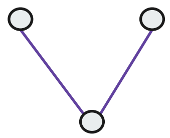
- **Entailment Invariability/Conservativity [Konev, Jiménez-Ruiz]**
  - Measures how much $\sqsubseteq$-entailments changed

$$LDiff(KB_t, KB_{t+1}) \coloneqq \{A \sqsubseteq B \mid KB_{t+1} \vDash A \sqsubseteq B\ and\ KB_t\ \nvDash A \sqsubseteq B\}$$

  - Implementations
    - Scalable but approximate based on SPARQL ($LDiff_{rdfs}$)
    - Optimised expressive uniform-interpolation $\mathcal{ALC}$ [Zhao; submitted] ($LDiff_{alc}$)

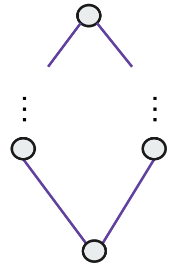
KB Integrity II

- **Graph-based Invariability**
  - Tangledness [Cangemi06]: characterises multi-hierarchical nature of KB



$$tang(O) := \frac{|\boldsymbol{Concepts}|}{|A \mid A \sqsubseteq C_1, A \sqsubseteq C_2|}$$

  - single number; too coarse, not very informative
  - Where do forks re-join



$$tang(A) \coloneqq \{\mathrm{E} \mid A \sqsubseteq C_1, A \sqsubseteq C_2, \mathrm{E} \in lcs(\mathrm{C}_1, \mathrm{C}_2)\}$$

  - how many fork/re-joins below a class

$$tang_{\downarrow}(A) \coloneqq \Sigma\,\{\mathrm{tang(C)}, \mathrm{KB} \vDash C \sqsubseteq \mathrm{A}\}$$

- **Label Integrity / Ambiguity**
  - Set of labels that appears in different classes

$$\mathrm{ambig(T)} \coloneqq \{\ell \mid \langle \mathrm{A}_1\ \mathrm{skos{:}\,label}\ \ell\rangle, \quad \langle A_2\ skos{:}\,label\ \ell\rangle\}$$

Heuristics to eliminate ambiguity

Information Change (Completeness Assessment)

- **Population of relations and classes**
  - Relations

$$usage(R) \coloneqq \{\langle A\ R\ B\rangle \mid A\ in\ the\ domain\ or\ R\ and\ B\ in\ its\ range\}$$

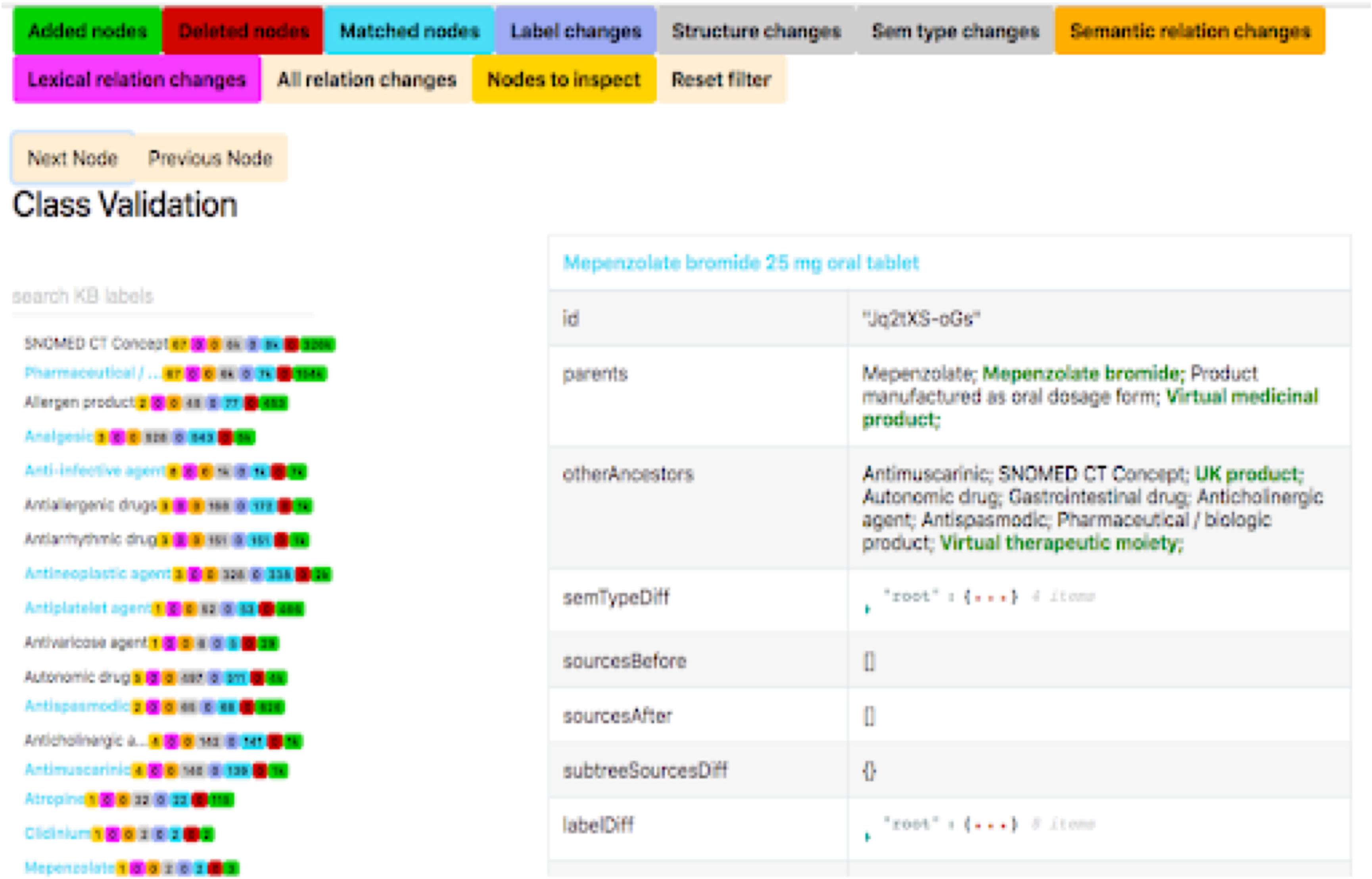
  - Classes

$$undef(A) \coloneqq \{R \mid A\ a\ descendant\ of\ a\ domain\ or\ R\}$$

Diseases are domains of hasSymptom, treatedBy, causedBy, …

Inspecting Metrics output

- **OntoDiff**



Building the Babylon KB

- **Which ontology to use as a "seed"**
- **Which sources to integrate (their quality, label ambiguity)?**
- **Used metrics to understand data sources**

	SNOMED	NCI	MeSH	MedDRA	CTV3	ICD-10	Read2	FMA
Classes	340 995	133 239	28 474	24 603	322 300	44 539	89 618	104 438
Count(tang>0)	118 120	12 529	7 950	8 248	10 092	0	0	0
ambig	1 072	4 873	0	5	24 960	708	1 139	261

- Snomed is the most multi-hierarchical; MeSH/MedDRA almost all re-join points (lcs) owl:Thing
- ICD-10, Read2 have 0 (they are coding/classification systems); NCI low (was initially a thesaurus)
- NCI, CTV3 Highly ambiguous ; synonyms used in a loose way; cannot use them safely in matching

The Babylon KB

	SNOMED	+NCI	+CHV	+FMA
Classes	340 995	429 241	429 241	524 837
Properties	93	124	124	219
subClassOf axioms	511 656	617 542	617 542	713 313
objProp assertions	526 146	664 742	664 742	962 190
dataProp assertions	543 416	946 801	1 043 874	1 211 459
Ambiguity	1072	5768	9207	9811
Ambiguity-repair	180	1266	1892	2078

- $LDiff$ **kept to $\emptyset$, Ambiguity reduced via heuristics**

Advanced $LDiff$ for SNOMED extensions

- **Several country extensions: Australian-snmd, Canadian-snmd**
  - Can we seamlessly integrate them in the KB?
  - Are they conservative extensions of SNOMED?

- **Used $LDiff_{alc}$**
  - $LDiff_{alc}(Snomed, Snomed_{cnd}) = \emptyset$ ☺
    - Safely enriches snomed with additional labels and classes (no hierarchy changes)
  - $|LDiff_{alc}(Snomed, Snomed_{austr})| = 67$ ☹
    - Even the case that $A \sqsubseteq B \in SNOMED$ is $B \sqsubseteq A \in SNOMED_{austr}$

Thanks!

Questions?