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
• 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., Zaverij]: 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

\[ \text{for every } A \in KB, KB \not\models A \sqsubseteq \bot \]

• practical implementation using SPARQL over GraphDB:

\[ \text{no } A \text{ s.t. } KB \models_{rdfls} A \sqsubseteq C \cap D, \quad C \text{ disjointWith } D \]

• Entailment Invariability/Conservativity [Konev, Jiménez-Ruiz]

• Measures how much \( \sqsubseteq \)-entailments changed

\[ LDif f(KB_t, KB_{t+1}) := \{ A \sqsubseteq B \mid KB_{t+1} \models A \sqsubseteq B \text{ and } KB_t \not\models A \sqsubseteq B \} \]

• Implementations

• Scalable but approximate based on SPARQL (\( LDif f_{rdfls} \))

• Optimised expressive uniform-interpolation \( \mathcal{ALC} \) [Zhao; submitted] (\( LDif f_{alc} \))
KB Integrity II

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

  \[
  tang(O) = \frac{|Concepts|}{|A \mid A \sqsubseteq C_1, A \sqsubseteq C_2|}
  \]

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

  \[
  tang(A) = \{E \mid A \sqsubseteq C_1, A \sqsubseteq C_2, E \in lcs(C_1, C_2)\}
  \]

  • how many fork/re-joins below a class

  \[
  tang_1(A) = \Sigma \{tang(C), KB \models C \sqsubseteq A\}
  \]

• **Label Integrity / Ambiguity**

  • Set of labels that appears in different classes

  \[
  ambig(T) = \{\ell \mid \langle A_1 \text{ skos: label } \ell \rangle, \langle A_2 \text{ skos: label } \ell \rangle\}
  \]

  Heuristics to eliminate ambiguity
Information Change (Completeness Assessment)

• Population of relations and classes
  • Relations
    
    \[
    \text{usage}(R) := \{(A, R, B) \mid A \text{ in the domain or } R \text{ and } B \text{ in its range}\}
    \]
  • Classes
    
    \[
    \text{undef}(A) := \{R \mid A \text{ a descendant of a domain or } R\}
    \]

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

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

- 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

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

- $\text{LDiff}$ kept to $\emptyset$, Ambiguity reduced via heuristics
Advanced $\text{LDiff}$ 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 $\text{LDiff}_{alc}$**
  - $\text{LDiff}_{alc}(\text{Snomed}, \text{Snomed}_{cn}) = \emptyset$
  - Safely enriches snomed with additional labels and classes (no hierarchy changes)
  - $|\text{LDiff}_{alc}(\text{Snomed}, \text{Snomed}_{aust})| = 67$
  - Even the case that $A \subseteq B \in \text{SNOMED}$ is $B \subseteq A \in \text{SNOMED}_{aust}$
Thanks!

Questions?