Medical Knowledge Graph Construction by Aligning Large Biomedical Datasets

Giorgos Stoilos, David Geleta,
Jetendr Shamdasani, and Mohammad Khodadadi

Babylon Health, London, SW3 3DD, UK
firstname.lastname@babylonhealth.com

1 Extended Abstract

Building large Knowledge Bases can be realised by aligning and integrating existing data sources. To support AI-based digital healthcare services within Babylon Health[1] significant effort to build a large medical KB was recently undertaken. To realise this goal a highly configurable and modular ontology integration pipeline has been created which works as follows: an initial ontology is used as a seed KB ($\mathcal{KB}_0$) and additional data sources are integrated into it creating new extended versions of $\mathcal{KB}_0$. The integration process is based on a *Matching* phase, an *Aggregation* phrase, and a final *PostProcessing* phase. In the *Matching* phase the following matchers can be used:

- An in-house LabelMatcher which is based along similar ideas as the label matcher in [1], i.e., label normalisation, inverted indexes, and more.
- The state-of-the-art systems AML [1] and LogMap [3] in both its versions LogMap$_o$[2] and LogMap$_c$[3].
- A UMLS-synonym and a UMLS-CUI based matcher, or mappings from 3rd parties like BioPortal, NHS, and more.

The mappings from the previous stage are *Aggregated* using a weighted average and a threshold is applied. Finally, post-processing performs the following:

- Mappings of higher-multiplicity (i.e., mapping multiple classes to the same one) are separated from the rest. The former are handled by *multiplicity-disambiguation* techniques which reduce them to 1-to-1 or 1-to-m mappings.
- All mappings go through existing [2] and novel [4] conservativity-based mapping repair methods in order to avoid altering the structure of the seed KB.

Significant efforts were spent to determine which matching algorithm to use in the *Matching* phase. The Large BioMedTrack datasets were considered for evaluating the methods, however, surprisingly enough these datasets are much older, smaller and with somewhat different content compared to the recent releases of

[1] https://www.babylonhealth.com/
[2] https://github.com/ernestojimenezruiz/logmap-matcher
[3] https://github.com/asolimando/logmap-conservativity/

Table 1. Evaluation results on aligning official releases of SNOMED and NCI

	precision	recall	f-Value	Time(sec)	♯mapppings
LabelMatcher	0.356	0.77	0.49	13	28457
LogMap	0.372	0.78	0.50	2 850	27342
AML	0.410	0.50	0.45	596	15861

Table 2. Statistics about the KB after each integration/enrichment iteration.

	SNOMED	+NCI	+CHV	+FMA
Classes	340 995	429 241	429 241	524 837
Properties	93	124	124	219
$\|A \sqsubseteq B\|$	511 656	617 542	617 542	713 313
$\|\langle A\ p\ \mathsf{iri} \cup \mathsf{Lit}\rangle\|$	1 069 562	1 611 543	1 708 616	2 173 649

SNOMED, NCI, and FMA that are considered in Babylon. For example, NCI in BioTrack is almost half the size of the NCI December 2017 release (the former contains 96K axioms whereas the latter 185K), FMA is almost 1/4 and SNOMED almost 1/3 of their recent releases. In addition, synonym labels of classes seem to be completely missing from all ontologies. For those reasons the reference set between SNOMED and NCI in the BioTrack was refactored to point to codes in the official releases and then a precision/recall evaluation of our LabelMatcher, AML, LogMap, and XMap was conduced using the official releases (see Table 1); XMap did not manage to terminate.

As can be seen, although in theory simple, LabelMatcher provides comparable precision/recall and is orders of magnitude faster; the very low precision is because of the extra mappings found in the larger ontology versions which are confused as false positives. Given the scalability results and adequate precision/recall, we used our LabelMatcher in the pipeline to integrate the latest versions of NCI, CHV, and FMA on top of SNOMED (indeed this process could not be completed using AML or LogMap$_o$). Statistics about the KBs that we created after each integration are depicted in Table 2; moreover, no conservativity violations could be detected due to our post-processing.

We have also compared our post-processing approach against mapping repairing implemented in AML, LogMap$_c$ and LogMap$_o$. In cases that these systems don't terminate we used smaller versions of our (test) ontologies. In all cases a large number of conservativity violations could be identified (in contrast to none detectable after running our approach); detailed results can be found in [4].

References

1. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: Proc. of OTM (2013)
2. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Ontology integration using mappings: Towards getting the right logical consequences. In: Proc. of ESWC (2009)
3. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y.: Logmap 2.0: towards logic-based, scalable and interactive ontology matching. In: Proc. of SWAT4(HC)LS. pp. 45–46 (2011)
4. Stoilos, G., Geleta, D., Shamdasani, J., Khodadadi, M.: A novel approach and practical algorithms for ontology integration. In: Proceedings of ISWC (2018)