Aspekty praktyczne wykorzystania technologii semantycznych

Piotr Sowiński

19 stycznia 2022

Politechnika Warszawska
Szkoła Doktorska nr 3

Agenda

1. (Very) brief introduction to semantics
2. Semantics, ML, and AI
3. Ontology quality
4. Large knowledge bases

1. Semantics

Why semantics?

- Computers are syntactic: they work with **symbols** and **data**
- …but humans are semantic creatures!
    - We work with **concepts** and **knowledge**
- The general idea of semantics:
    - Let computers reason with concepts
    - Process knowledge, not just data

- By the way: can we really say a neural network models knowledge? Or is it just a bunch of vectors and matrices?
    - We will come back to this later (sect. 2)

Ontologies in computer science

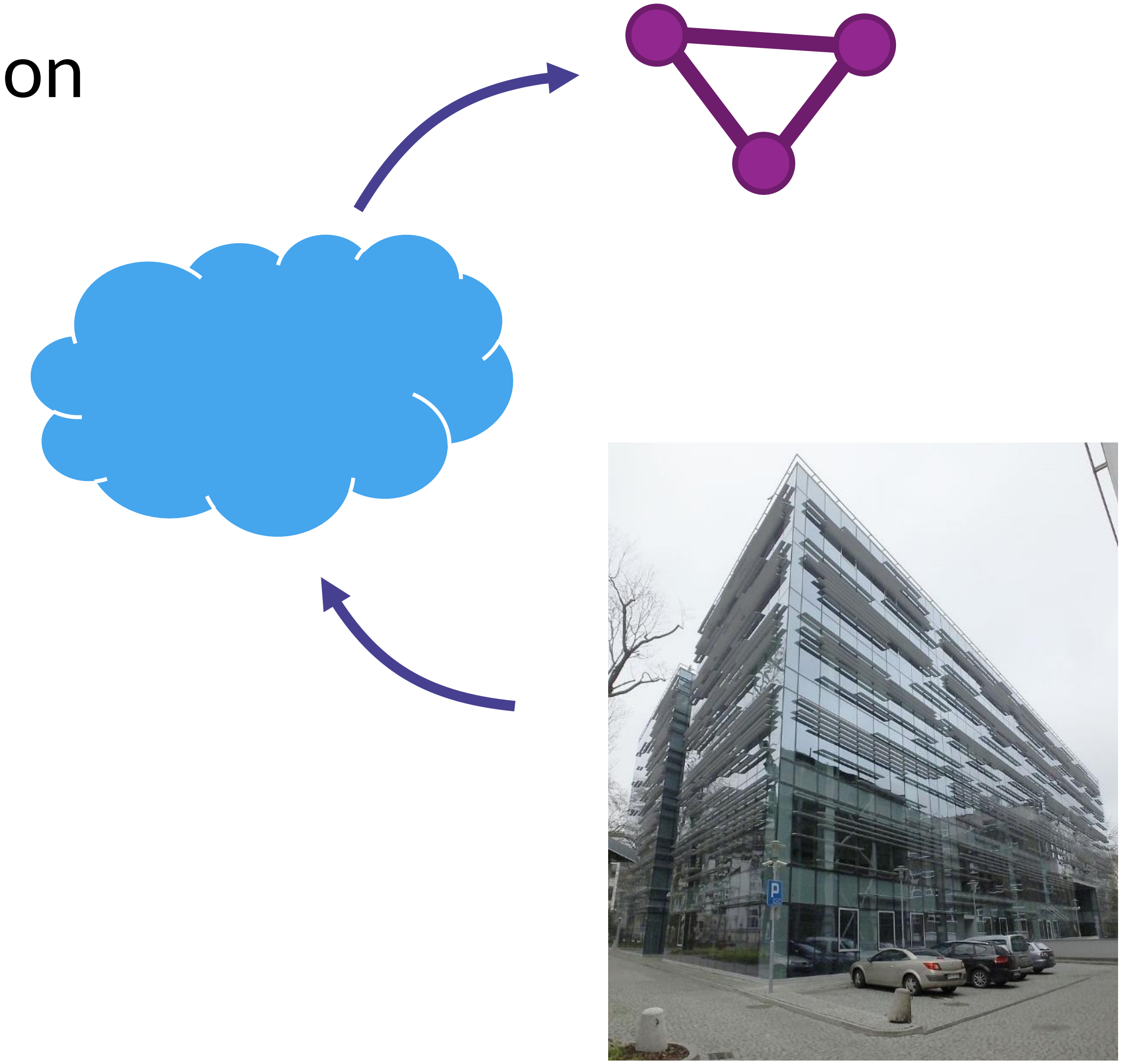
- **Ontology** – explicit specification of a conceptualization of a *world*. (Gruber 1995)

Ontologies in computer science

- **Ontology** – explicit specification

  of a conceptualization

  of a *world*.



Ontologies in computer science

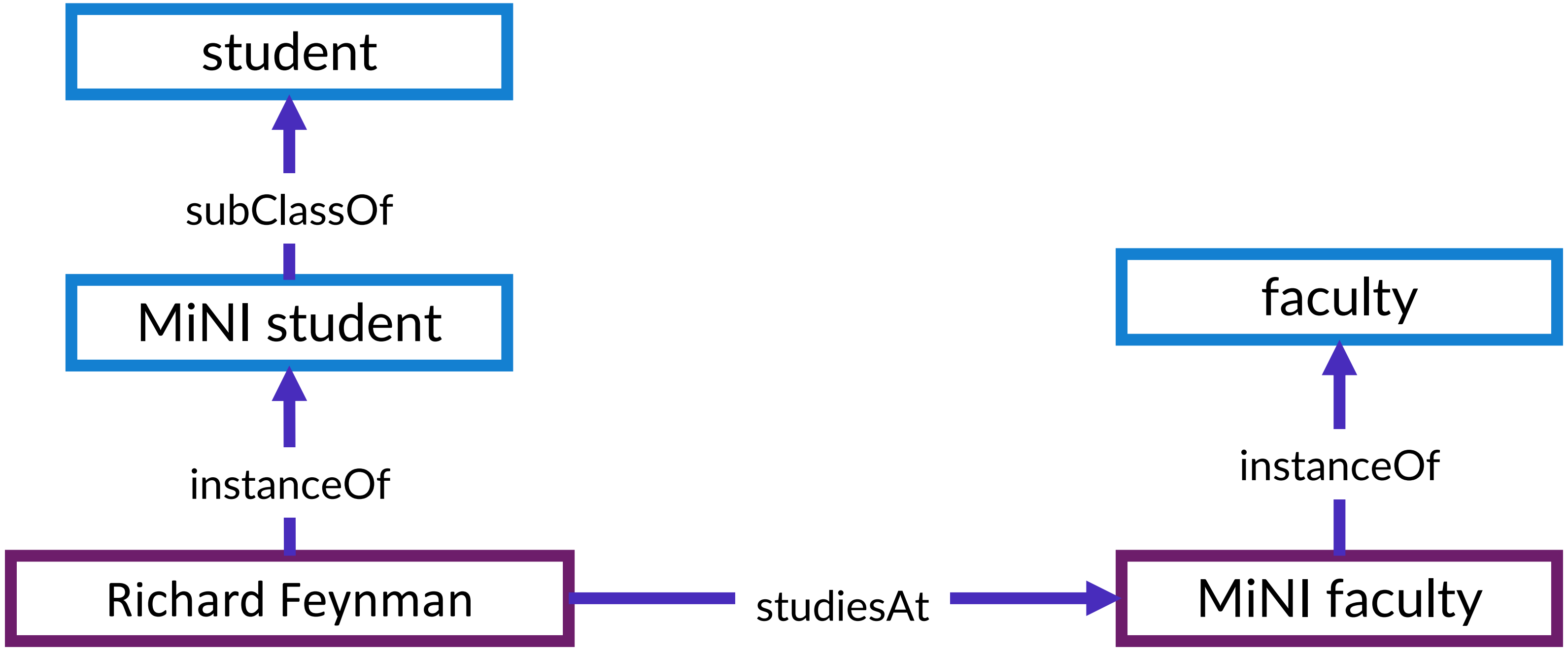
- Knowledge representations
  - Glossaries
  - Semantic networks
  - Formal taxonomies
  - Objects with properties (frames)
  - General logic constraints

Ontologies

Based in description logics!

-> we can use deduction

Ontologies: OWL and description logics



\`student\` studiesAt **min 1** \`faculty\`

\`MiNI student\` studiesAt **some** \`MiNI faculty\`

Ontologies: OWL Manchester syntax (Horridge et al. 2006)

OWL Constructor	DL Syntax	Manchester OWL S.	Example
intersectionOf	$C \sqcap D$	C **AND** D	Human **AND** Male
unionOf	$C \sqcup D$	C **OR** D	Man **OR** Woman
complementOf	$\neg$ C	**NOT** C	**NOT** Male
oneOf	$\{a\} \sqcup \{b\}$...	{a b ...}	{England Italy Spain}
someValuesFrom	$\exists$ R C	R **SOME** C	hasColleague **SOME** Professor
allValuesFrom	$\forall$ R C	R **ONLY** C	hasColleague **ONLY** Professor
minCardinality	$\geq$ N R	R **MIN** 3	hasColleague **MIN** 3
maxCardinality	$\leq$ N R	R **MAX** 3	hasColleague **MAX** 3
cardinality	$=$ N R	R **EXACTLY** 3	hasColleague **EXACTLY** 3
hasValue	$\exists$ R $\{a\}$	R **VALUE** a	hasColleague **VALUE** Matthew

Fig. 3. The Manchester OWL Syntax OWL 1.0 Class Constructors

Ontologies: OWL Manchester syntax (Horridge et al. 2006)

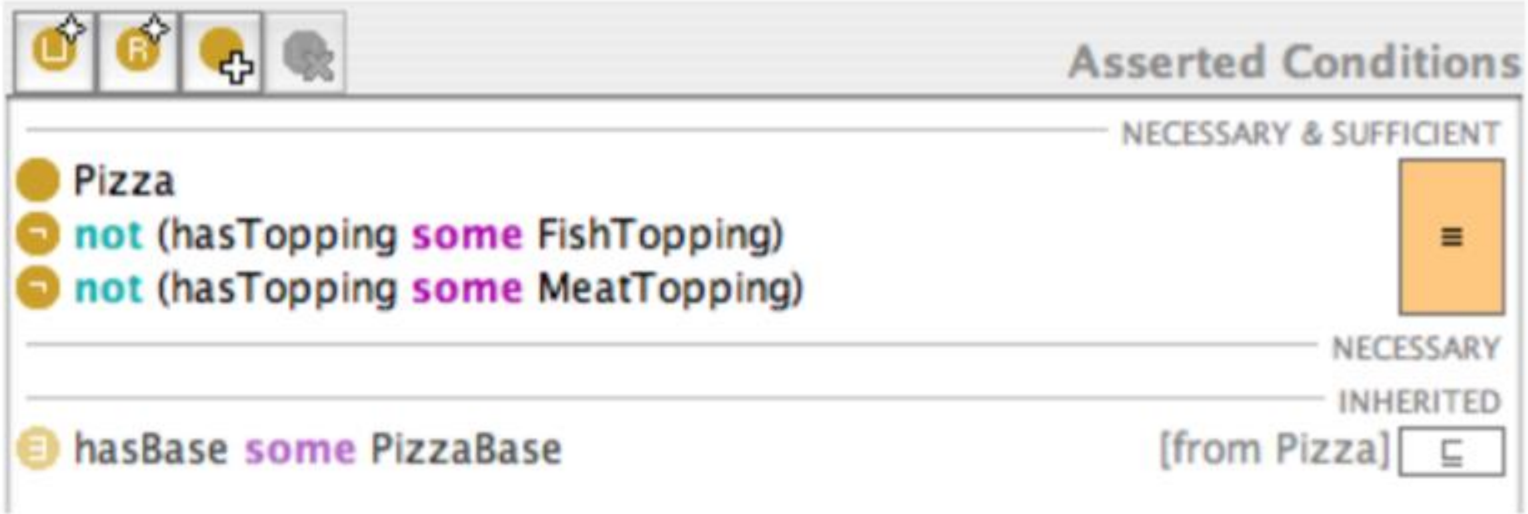


Fig. 4. An example of the Manchester OWL Syntax being used to represent the concept of a VegetarianPizza in Protégé-OWL

So, all ontologies are extremely expressive?

- No, not really.

- Nobody forces the amount of expressivity
- There are less and more formal ontologies and that is (usually) fine

Linked Data

- Use the Web as the underlying infrastructure
  - Every entity has a URI (Uniform Resource Identifier)
- Use common W3C standards (RDF, OWL, SPARQL)
- Reuse ontologies by linking and combining them
  - Knowledge reuse
  - Interoperability
  - Shared understanding
- Ideally – make them freely available (Linked Open Data)
- It this picture too rosy? (sect. 3)

Knowledge base (KB)

Ingredients:

- Ontology (ontologies?)
- Storage
- Query interface
- Update interface

In short: *database for knowledge*

2. Semantics, ML, and AI

Language Models As Knowledge Bases? (Petroni et al. 2019)

- Large LMs acquire a huge amount of knowledge during training
- On the other hand, KBs are insanely hard to produce and query (sect. 3, 4)
- So why not just query the LM?
- Only really works for 1:1 relations
- Can only query single-token objects
- Different question formulations give significantly different results
  - Does the LM really "know" anything?
  - No quantitative measurements! :(
- Of course, there were more similar papers…

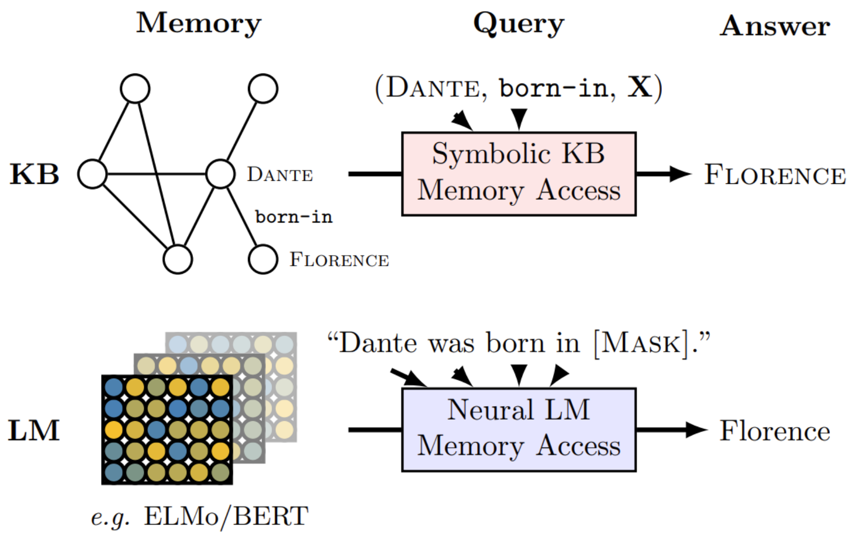


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

Language Models As or For Knowledge Bases? (Razniewski et al. 2021)

LMs' deficiencies:

- Impossible to "list" all the knowledge in the LM
- Correlations vs explicit knowledge

Example: When prompting GPT-3 for awards won by Alan Turing, its top-confidence prediction is the Turing Award, and lower-ranked outputs include "Nobel Prize" and "the war" (none of them correct).

- Know what you **don't** know

Example: Alan Turing was homosexual and never married. When prompting GPT-3 with the phrase "Alan Turing married", the top prediction is "Sara Lavington" with score 21%, and for the prompt "Alan Turing and his wife" it is "Sara Turing" (his mother's name). This is a case of LM hallucination [25, 26]. In contrast, Wikidata has an explicit statement ⟨ `Alan Turing, spouse, no value` ⟩ *denoting that he was unmarried.*

Language Models As or For Knowledge Bases? (Razniewski et al. 2021)

LMs' deficiencies, continued:

- No reasonable, systematic approach to curatability
- No provenance tracking
- Good entity disambiguation requires context
- Not all knowledge is text-based
- How to handle more complex relations? 1:n, n:m?

On the other hand:

- KBs' scope is limited by the set of defined predicates

Language Models? Knowledge Bases?

- Two very different animals.

- My view:
  - They can complement each other!

- How can we use LMs and other ML models in semantics?

3. Ontology quality

Ontology quality assurance

- Any errors in the ontology have a negative impact on its applications
- Errors include: wrong/missing relations, invalid hierarchies, invalid alignments, wrong/missing metadata, wrong/missing values
  - …and more

- Challenges for ontology QA:
  - Large knowledge bases
  - High velocity of changes (e.g., Wikidata)
  - Complex structures (high cognitive requirements)
  - Need for expert knowledge (**expensive!)**
  - Large number of heterogenous, dispersed ontologies (e.g., OBO Foundry)

OOPS! (Poveda-Villalón et al. 2014)

CRITICAL (1)

P01. Creating polysemous elements
P03. Creating the relationship "is" instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"
P05. Defining wrong inverse relationships
P06. Including cycles in the hierarchy
P14. Misusing "owl:allValuesFrom"
P15. Misusing "not some" and "some not"
P16. Misusing primitive and defined classes
P19. Swapping intersection and union
P27. Defining wrong equivalent relationships
P28. Defining wrong symmetric relationships
P29. Defining wrong transitive relationships
P31. Defining wrong equivalent classes
P37. Ontology not available
P39. Ambiguous namespace
P40. Namespace hijacking

OOPS! (Poveda-Villalón et al. 2014)

IMPORTANT (2)	MINOR (3)
P10. Missing disjointness	P02. Creating synonyms as classes
P11. Missing domain or range in properties	P04. Creating unconnected ontology elements
P12. Missing equivalent properties	P07. Merging different concepts in the same class
P17. Specializing a hierarchy exceedingly	P08. Missing annotations
P18. Specifying the domain or range exceedingly	P09. Missing basic information
P23. Using incorrectly ontology elements	P13. Missing inverse relationships
P24. Using recursive definition	P20. Misusing ontology annotations
P25. Defining a relationship inverse to itself	P21. Using a miscellaneous class
P26. Defining inverse relationships for a symmetric one	P22. Using different naming criteria in the ontology
P30. Missing equivalent classes	P32. Several classes with the same label
P34. Untyped class	P33. Creating a property chain with just one property
P35. Untyped property	P36. URI contains file extension
P38. No OWL ontology declaration	

FOOPS! (Garijo et al. 2021)

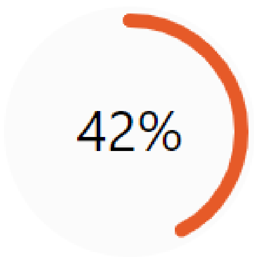
URI

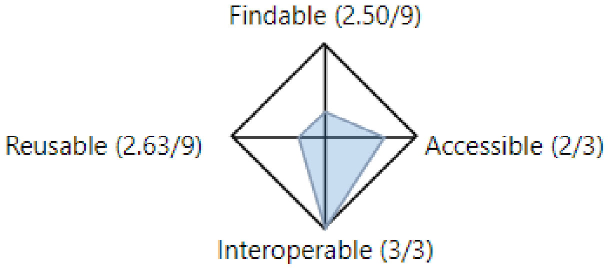
http://docs.inter-iot.eu/ontology/owl/GOIoTP.rdf

Example: https://w3id.org/example (click here to enter this ontology)

RUN

Title:	Generic Ontology for IoT Platforms
URI:	http://inter-iot.eu/GOIoTP#
License:	unknown





FOOPS! (Garijo et al. 2021)

R1.1: (meta)data are released with a clear and accessible data usage license

OM4.1: License availability 0%

Description: This check verifies if a license associated with the ontology

Explanation: License or rights not found

R1.2: (meta)data are associated with detailed provenance

OM5_2: Detailed provenance metadata 50%

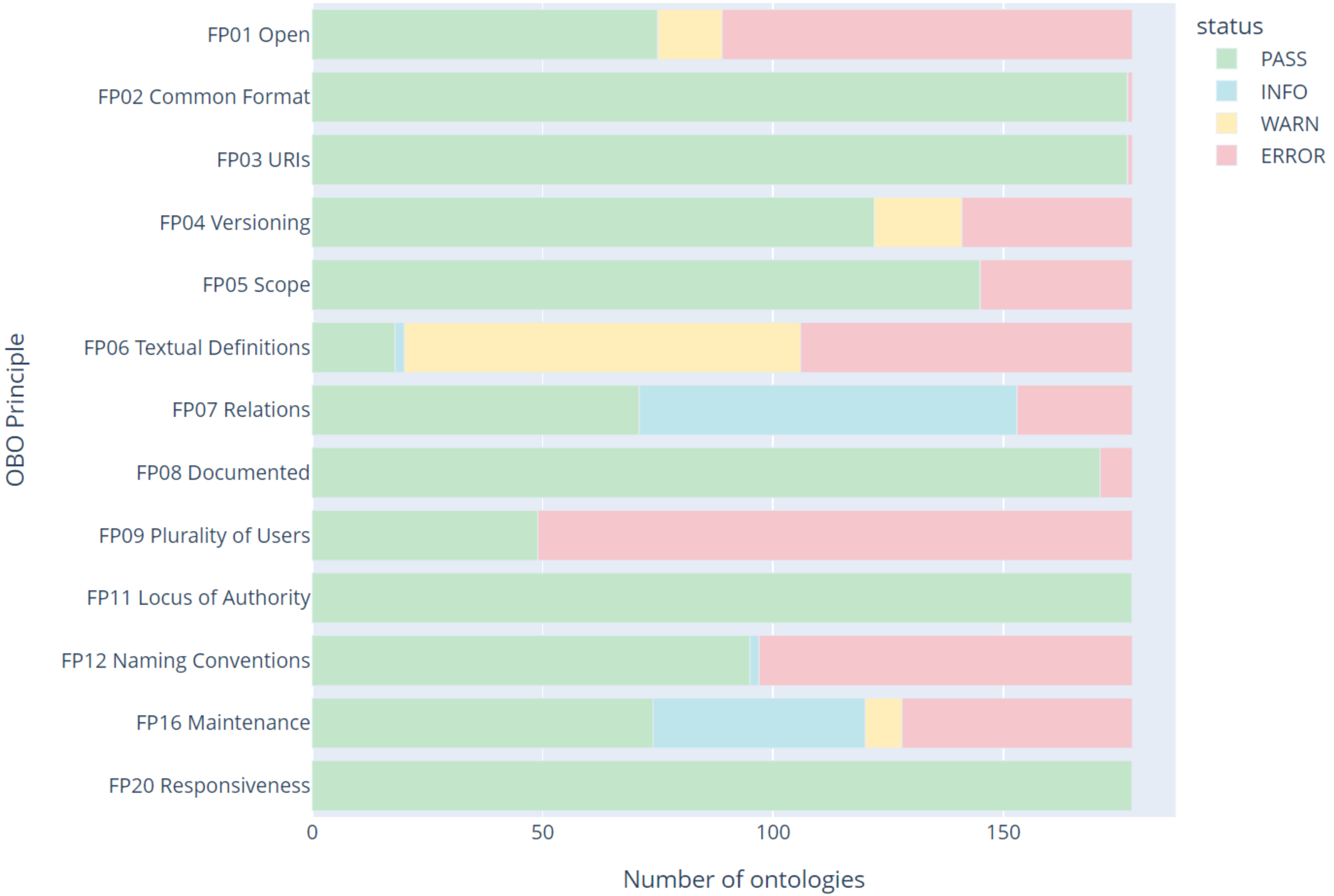
Description: This check verifies if detailed provenance information is available for the ontology: [issued date, publisher]

Explanation: The following provenance information was not found: publisher

OBO Dashboard (Jackson et al. 2021)

Ontology (click for details)	Open	Format	URIs	Versioning	Scope	Definitions	Relations	Documented	Users	Authority	Naming	Maintained	Responsiveness	ROBOT Report	Summary
aeo	✖	✔	✔	✔	✔	⚠	ℹ	✔	✖	✔	✔	✖	✔	✖	✖
agro	✔	✔	✔	✔	✔	✖	ℹ	✔	✔	✔	✖	✔	✔	✖	✖
aism	✔	✔	✔	✔	✔	⚠	ℹ	✔	✖	✔	✖	✔	✔	✖	✖
amphx	✔	✔	✔	✔	✔	✖	✔	✔	✖	✔	✔	✔	✔	✖	✖
apo	✔	✔	✔	✔	✔	✖	✔	✔	✖	✔	✖	✔	✔	✖	✖
apollo_sv	✔	✔	✔	⚠	✔	✖	ℹ	✔	✖	✔	✖	ℹ	✔	✖	✖
aro	✖	✔	✔	✖	✖	✖	✔	✔	✖	✔	✔	✖	✔	✖	✖
bco	✔	✔	✔	✔	✔	⚠	ℹ	✔	✖	✔	✖	✔	✔	✖	✖
bfo	✔	✔	✔	✔	✔	⚠	✔	✔	✔	✔	✔	⚠	✔	✖	✖
bspo	✔	✔	✔	✔	✔	✖	ℹ	✔	✖	✔	✖	✔	✔	✖	✖
bto	✔	✔	✔	✔	✔	✖	ℹ	✖	✖	✔	✖	✔	✔	✖	✖
caro	✖	✔	✔	⚠	✔	⚠	✔	✔	✖	✔	✖	ℹ	✔	✖	✖
cdao	⚠	✔	✔	✔	✖	⚠	✖	✔	✖	✔	✔	⚠	✔	✖	✖
cdno	✖	✔	✔	✔	✔	⚠	✔	✔	✖	✔	✔	✔	✔	⚠	✖
chebi	✖	✔	✔	⚠	✔	✖	ℹ	✔	✔	✔	✖	ℹ	✔	✖	✖

OBO Dashboard (Jackson et al. 2021)



So, OBO Foundry is a good example, right?

(to be published)

Table 2. Summary of issues found in OBO Foundry ontologies

Ontology	Rare prop.[1]	Prop. obj.[2]	Xref: blank[3]	Xref: URI[4]	Xref: unk.[5]
AEO	4	0	0	10	136
AGRO	18	51	0	1 266	6 710
APOLLO-SV	4	308	214	2	21
BFO	0	0	0	0	0
BTO	3	0	0	0	3 479
CARO	1	6	0	380	1 800
CHEBI	12	0	0	0	313 736
CL	38	236	0	2 297	34 296
DOID	2	2	0	1	12 824
DRON	9	6	0	0	35 148
EHDAA2	3	0	0	5	67
ENVO	3	1 612	0	3 299	1 649
FOBI	5	0	0	0	0
FoodOn	0	5 702	0	8 416	6 329
GAZ	0	6	0	0	25 505
GO	1	2 536	0	354	118 473
HP	45	313	0	3 520	28 386
IAO	0	22	0	0	0
MP	47	388	0	15 253	37 229
NCBITaxon	0	0	0	0	0
OBI	0	1 295	0	0	0
PATO	13	96	0	3 485	17 144
PCO	3	19	0	9	41
PECO	2	0	0	0	685
PO	3	24	0	3	6 547
RO	2	35	0	0	15
SYMP	2	0	0	1	449
Uberon	87	375	0	23 845	14 627
UO	9	0	0	0	0
XCO	3	0	0	0	494
All	**278**	**12 296**	**214**	**52 122**	**655 934**

1 Invalid occurrences of rarely-used properties.
2 Property object type mismatch (URI instead of literal or vice versa).
3 Cross-references pointing to blank nodes.
4 Cross-references pointing to URIs instead of identifiers.
5 Non-resolvable cross-reference identifiers.

Synonym or different concept? (to be published)

- Case study: Computer Science Ontology (CSO)
  - Essentially a taxonomy of CS research topics
  - Semi-automatically constructed
- CSO groups topics into synonym sets
  - Like WordNet, but it's often quite bad.
  - Can we find such mistakes with NLP might?

Subject	Predicate	Object
sensor data	alternative label of	sensor device
"	alternative label of	sensor readings
"	alternative label of	sensor systems

Synonym or different concept? (to be published)

- Setup:
  - Group entities into synonym sets
  - Encode their labels using sentence BERT (all-mpnet-base-v2)
  - Compute all-to-all similarity matrices within clusters
  - Find least consistent clusters by looking at mean and stdev
  - Have a few experts review this
- Generated **115** suspicious clusters
  - At least 3 entities each

Synonym or different concept? (to be published)

2203	computational efficiency	TRUE	definitely good	
2203	computation efficiency	FALSE	definitely good	
2203	computational time	FALSE	definitely wrong	
2203	computational costs	FALSE	probably good	
2203	computation time	FALSE	definitely wrong	
		Overall	definitely wrong	
645	neural networks	TRUE	definitely good	
645	artificial neural networks	FALSE	definitely good	
645	artificial neural network	FALSE	definitely good	
645	neural network model	FALSE	definitely good	
645	back-propagation neural networks	FALSE	definitely wrong	There are other ways of establishing NN weights, like genetic algorithms.
645	neural network	FALSE	definitely good	
645	back-propagation neural network	FALSE	definitely wrong	
645	back propagation neural networks	FALSE	definitely wrong	
		Overall	definitely wrong	
1760	multi-core processor	TRUE	definitely good	
1760	multi-core processors	FALSE	definitely good	
1760	multicore processors	FALSE	definitely good	
1760	multicore processor	FALSE	definitely good	

Synonym or different concept? (to be published)

- Results
  - **Majority vote**: at least 2 reviewers agreed that **84/115** clusters are wrong
  - At least 1 reviewer marked **95/115** clusters as wrong
  - All 3 reviewers agreed that **58/115** clusters are wrong
- Other observations
  - A lot of the valid synonyms are useless
  - Often found out-of-scope clusters (genetics, didactics)
  - Conflation of *problem*, *method*, *accuracy*, *algorithm*, etc.

Correcting Knowledge Base Assertions (Chen et al. 2020)

- Often, the issue is an invalid value of a property
  - E.g., *Manchester City* instead of *Manchester United*
- Easiest approach to "fix it": remove the assertion
- This work's contribution: actually fixing the assertion

- It illustrates several approaches for using ML with KBs :)

Correcting Knowledge Base Assertions (Chen et al. 2020)

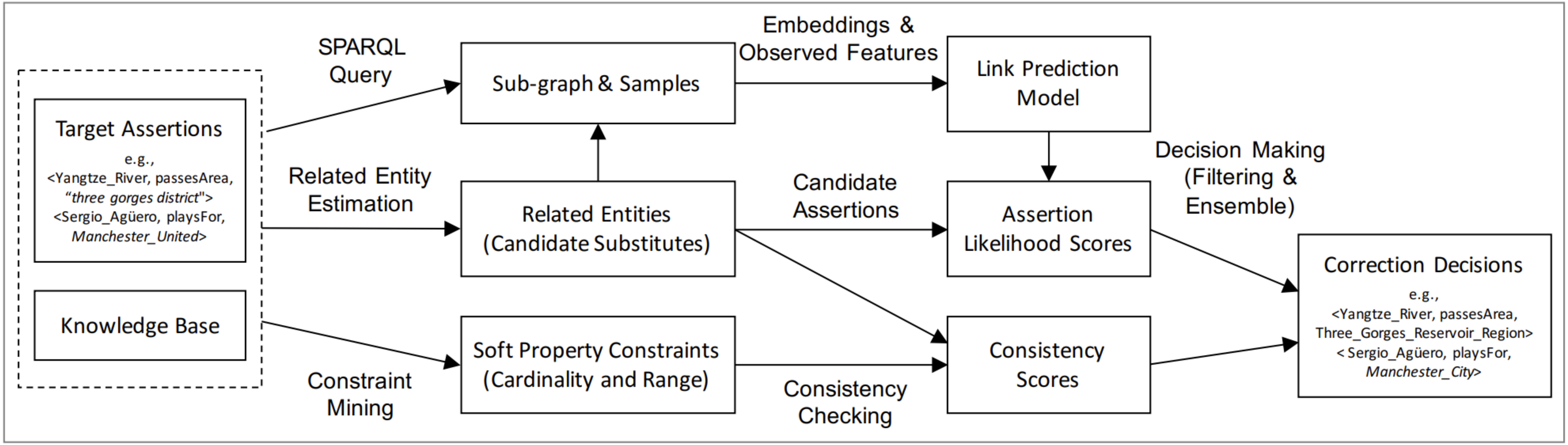


Figure 1: The Overall Framework for Assertion Correction

Correcting Knowledge Base Assertions (Chen et al. 2020)

Dataset

- DBpedia: generated straight from the KB
- Unnamed enterprise medical KB: real issues found & corrected by experts

	Assertions (with Entity GT) #	Properties #	Subjects #
DBP-Lit	725 (499)	127	668
MED-Ent	272 (225)	7	200

Table 1: Some statistics of DBP-Lit and MED-Ent.

Correcting Knowledge Base Assertions (Chen et al. 2020)

Methods	DBP-Lit		MED-Ent	
	C-Rate	Acc	C-Rate	Acc
Lexical Matching	0.597	0.611	0.149	0.123
Lookup*	0.635	0.516	—	—
Word2Vec	0.553	0.410	0.089	0.076
REE + LP ($\mathcal{M}_{np}$)	0.677	0.677	0.360	0.327
REE + LP ($\mathcal{M}_{dm}$)	0.635	0.628	0.600	0.588
REE + CV ($\mathcal{M}_{ran}$)	0.671	0.668	0.271	0.239
REE + CV ($\mathcal{M}_{car}$)	0.639	0.622	0.164	0.147
REE + CV ($\mathcal{M}_{ran+car}$)	0.677	0.684	0.271	0.246
REE + LP + CV	**0.701**	**0.690**	**0.609**	**0.599**

Table 4: Optimum correction rate (C-Rate) and accuracy (Acc). REE denotes Related Entity Estimation: DBP-Lit uses Lookup*, MED-Ent uses Edit Distance.

4. Large knowledge bases

Really large knowledge bases in practice

- DBpedia: ~10 billion triples, 6 million entities
- Wikidata: ~13.6 bilion triples, growing fast*
  - ...and **hundreds** of edits per minute from all over the world
  - Single primary MariaDB node tracks all changes **(!!!)** and propagates them
  - Queries handled by batch-updated servers, duct-tape replication
  - **22** query servers: 2x6 cores, 128 GB RAM
- Wikidata is starting to hit the software limits of Blazegraph**
  - No "good" alternatives, sadly
- Doing any research with Wikidata? You need expensive hardware and a lot of patience.

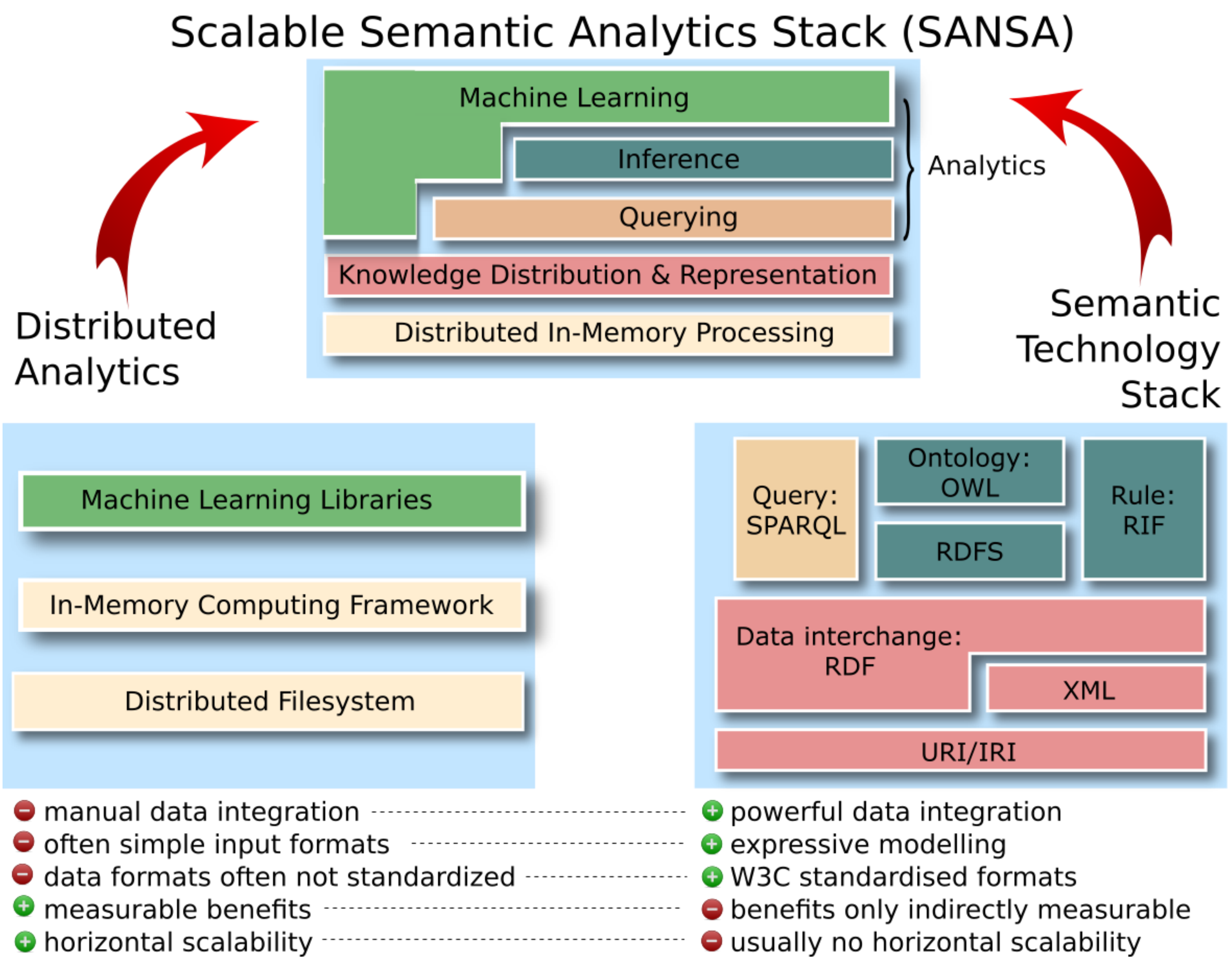
* It's all public, see for example:
https://grafana.wikimedia.org/d/000000154/wikidata?orgId=1
https://wikitech.wikimedia.org/wiki/Wikidata_Query_Service
https://wikitech.wikimedia.org/wiki/MariaDB

** See:
https://www.wikidata.org/wiki/Wikidata:Query_Service_scaling_update_Aug_2021
https://phabricator.wikimedia.org/T206560

Large KBs vs large DBs

- Huge databases, both relational and noSQL are a pretty much a **solved** issue
- We also saw incredible advancements in big data, with e.g., Apache Spark becoming virtually a **standard**
- So why can't we even have a properly replicated, open-source triple store?
- Why should we (researchers) care?
  - Technology enables research

SANSA stack (Lehmann et al. 2017)



SANSA stack – Sparklify (Stadler et al. 2019)

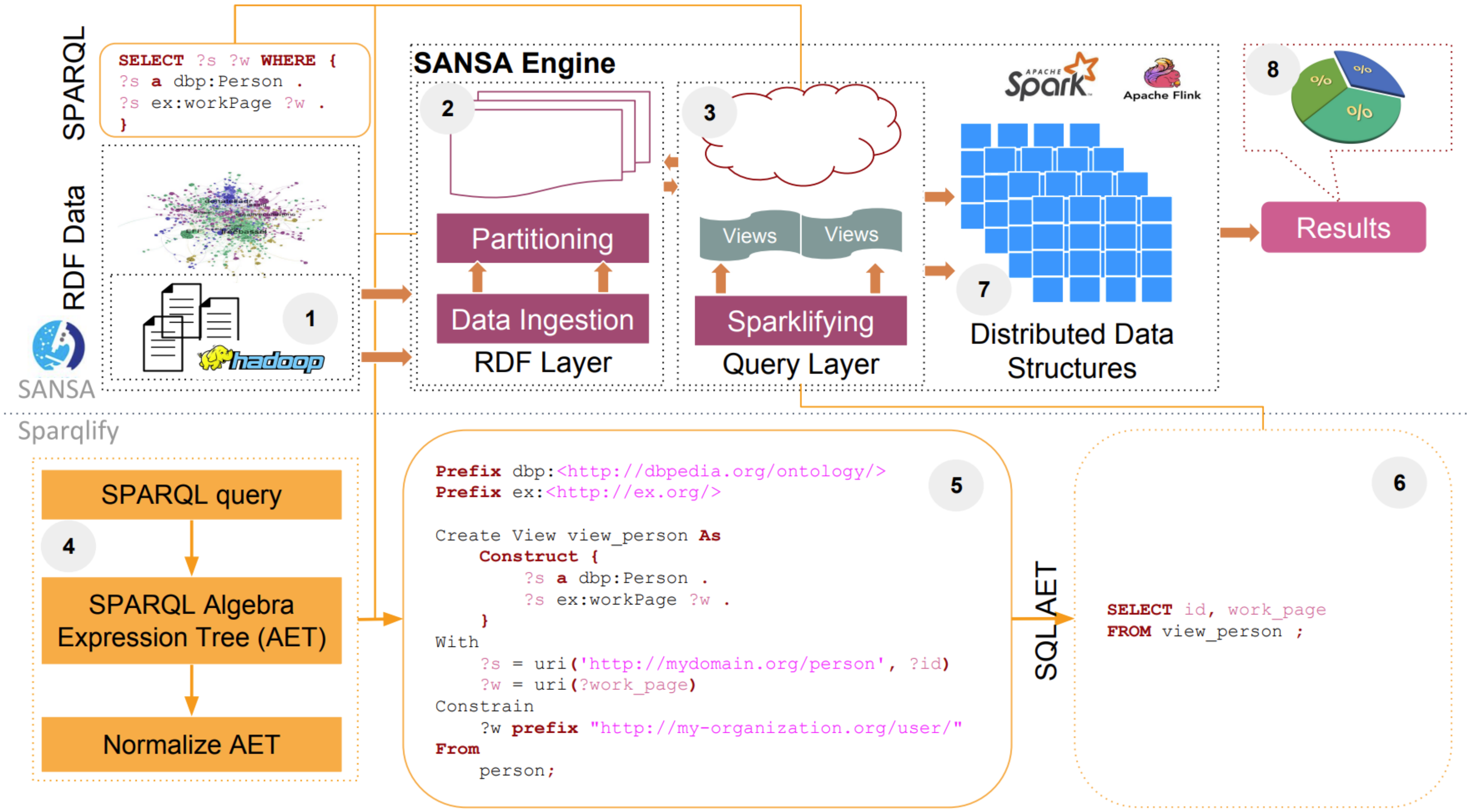


Fig. 1. Sparklify Architecture Overview.

SANSA stack – Sparklify (Stadler et al. 2019)

Result presented here shows that Sparklify can achieve linear scalability in the performance, which addresses Q3.

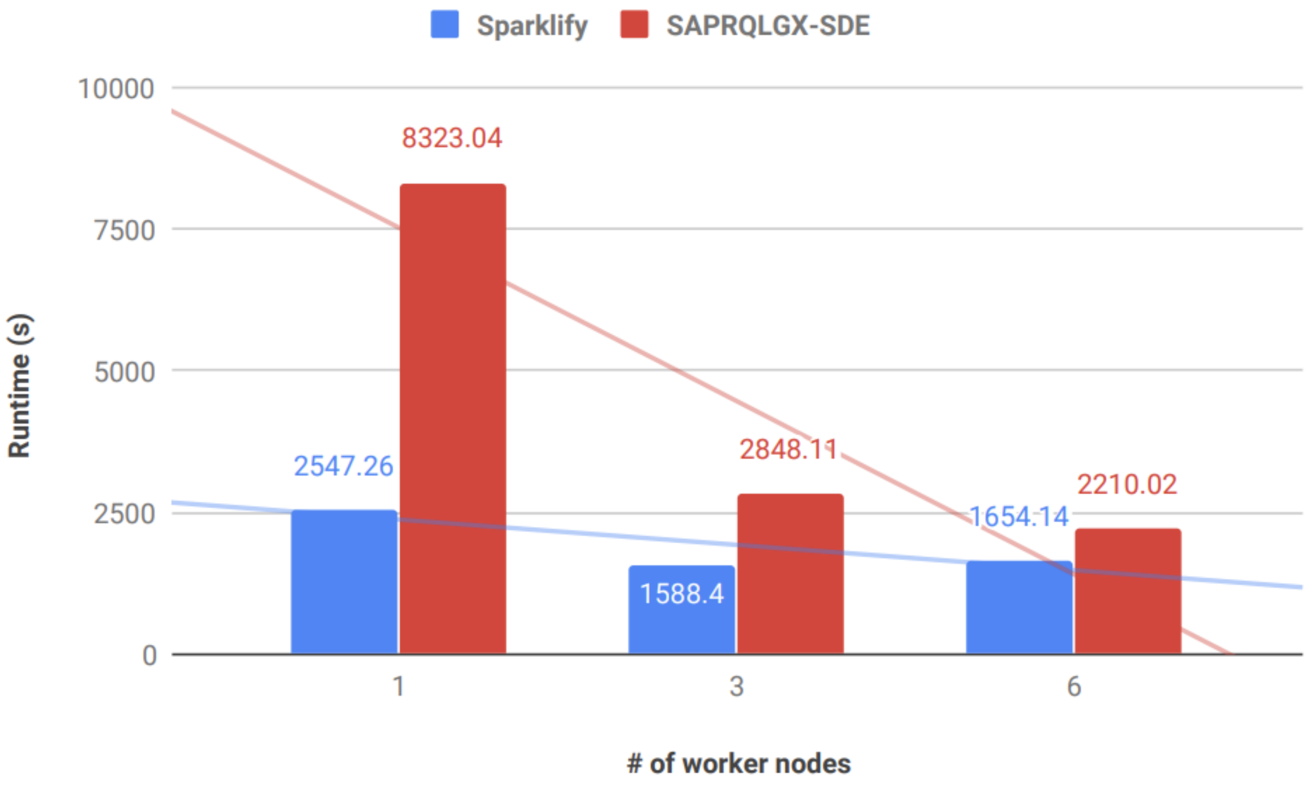


Fig. 3. Node scalability (on Watdiv-100M).

SANSA stack (Lehmann et al. 2017)

- After 4 years, much of it is still very experimental
- No reliable performance evaluations/comparisons
  - (at least to my knowledge)
- Does not solve the "expensive hardware" part
- Querying works, but the language is limited
- Very programmer-oriented, hard to get started
- Missing documentation

- Loooooong way ahead to "productionalizing" it :)

Summary

I wish I had the time for…

- Knowledge graph embeddings
- Large KB reasoning
- Cross-ontology references
- Ontology reuse in practice – including social aspects
- KBs and network analysis

- Maybe next time…?

Bibliography

- Chen, J., Chen, X., Horrocks, I., B. Myklebust, E., & Jimenez-Ruiz, E. (2020, April). Correcting knowledge base assertions. In *Proceedings of The Web Conference 2020* (pp. 1537-1547).
- Garijo, D., Corcho, O., & Poveda-Villalón, M. (2021). FOOPS!: An Ontology Pitfall Scanner for the FAIR principles. In Proceedings of the ISWC.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. *International journal of human-computer studies*, *43*(5-6), 907-928.
- Horridge, M., Drummond, N., Goodwin, J., Rector, A. L., Stevens, R., & Wang, H. (2006, November). The Manchester OWL syntax. In *OWLed* (Vol. 216).
- Jackson, R., Matentzoglu, N., Overton, J. A., Vita, R., Balhoff, J. P., Buttigieg, P. L., ... Peters, B. (10 2021). OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database*, 2021. doi:10.1093/database/baab069
- Lehmann, J., Sejdiu, G., Bühmann, L., Westphal, P., Stadler, C., Ermilov, I., ... & Jabeen, H. (2017, October). Distributed semantic analytics using the SANSA stack. In *International Semantic Web Conference* (pp. 147-155). Springer, Cham.
- Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019, November). Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 2463-2473).
- Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *10*(2), 7-34.

Bibliography, continued

- Razniewski, S., Yates, A., Kassner, N., & Weikum, G. (2021). Language Models As or For Knowledge Bases. *arXiv preprint arXiv:2110.04888*.
- Stadler, C., Sejdiu, G., Graux, D., & Lehmann, J. (2019, October). Sparklify: A Scalable Software Component for Efficient evaluation of SPARQL queries over distributed RDF datasets. In *International Semantic Web Conference* (pp. 293-308). Springer, Cham.

Other further reading

- Formal representations of knowledge: https://www.obitko.com/tutorials/ontologies-semantic-web/formal-representation.html
- Simple explanation of OWL class expressions: http://protegeproject.github.io/protege/class-expression-syntax/
- What is Wikidata: https://www.wikidata.org/wiki/Wikidata:Introduction

Image sources

- Slide 6: https://commons.wikimedia.org/wiki/File:Gmach_matematyki_PW.JPG – Panek, CC-BY-SA-4.0 International
- Slides 23–24: https://foops.linkeddata.es/FAIR_validator.html – Daniel Garijo & María Poveda-Villalón
- Slides 27–28: http://dashboard.obofoundry.org/dashboard/index.html (Jackson et al. 2021)

- Other images were either created by me or were taken from the article referenced on the slide.

Thank you for your attention!