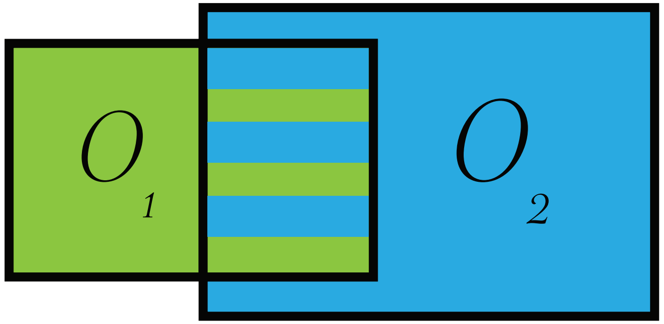
Tracking Logical Difference in Large-Scale Ontologies: A Forgetting-Based Approach

Yizheng Zhao Ghadah Alghamdi Renate A. Schmidt Hao Feng
Giorgos Stoilos Damir Juric Mohammad Khodadadi
University of Manchester, UK[1]
North China University of Science & Technology
Babylon Health

AAAI 2019, Honolulu, Jan 2019

[1] Thanks to EPSRC IAA 204, Babylon Health, SoCS

Our interest: Tracking differences in ontologies



Information belonging to the old version (first ontology)

Information belonging to the new version (second ontology)

Useful for:

- supporting ontology curation and maintenance
- ontology intergration, quality assurance

Problems: Syntactic differences of limited use
Logical difference only approximated
Inability to handle large-scale ontologies

Contribution:
New, practical method for logical difference analysis

UI-based logical difference approach

compact, finite (!) representation of logical difference

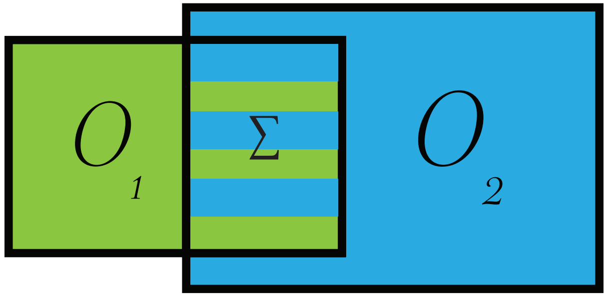
New UI/forgetting-based method

super-fast

Can track very large scale ontologies

SNOMED CT	$>$ 335K concept definitions
NCIt	$>$ 60K concept definitions

Logical Difference



$\Sigma = \mathsf{sig}(\mathcal{O}_1) \cap \mathsf{sig}(\mathcal{O}_2)$

Goal: Compute logical difference $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ s.t.

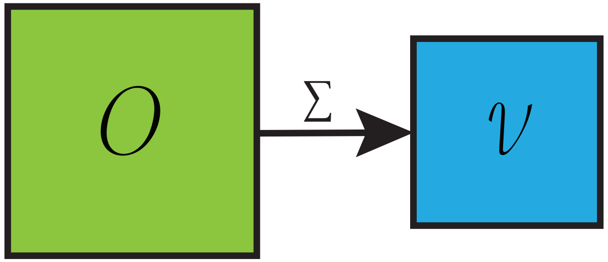
$$\alpha \in \mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2) \quad \text{iff} \quad \begin{array}{ll} \text{(i)} & \mathsf{sig}(\alpha) \subseteq \Sigma \\ \text{(ii)} & \mathcal{O}_2 \models \alpha, \text{ but } \mathcal{O}_1 \not\models \alpha \end{array}$$

Problem: Computationally infeasible because in general infinite
Therefore, existing methods approximate $\mathsf{Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$

- Syntactically generated and tested: ECCO
  α of the form $A \sqsubseteq B$ or $A \sqsubseteq \exists r.B$ $\quad \{A, B, r\} \subseteq \Sigma$
- depth-limited uniform interpolants: CEX

Our solution:
Finite UI-Diff representation

Uniform interpolation and forgetting



Goal: Compute uniform interpolant $\mathcal{V}$ s.t.

(i) $\mathrm{sig}(\mathcal{V}) \subseteq \Sigma$

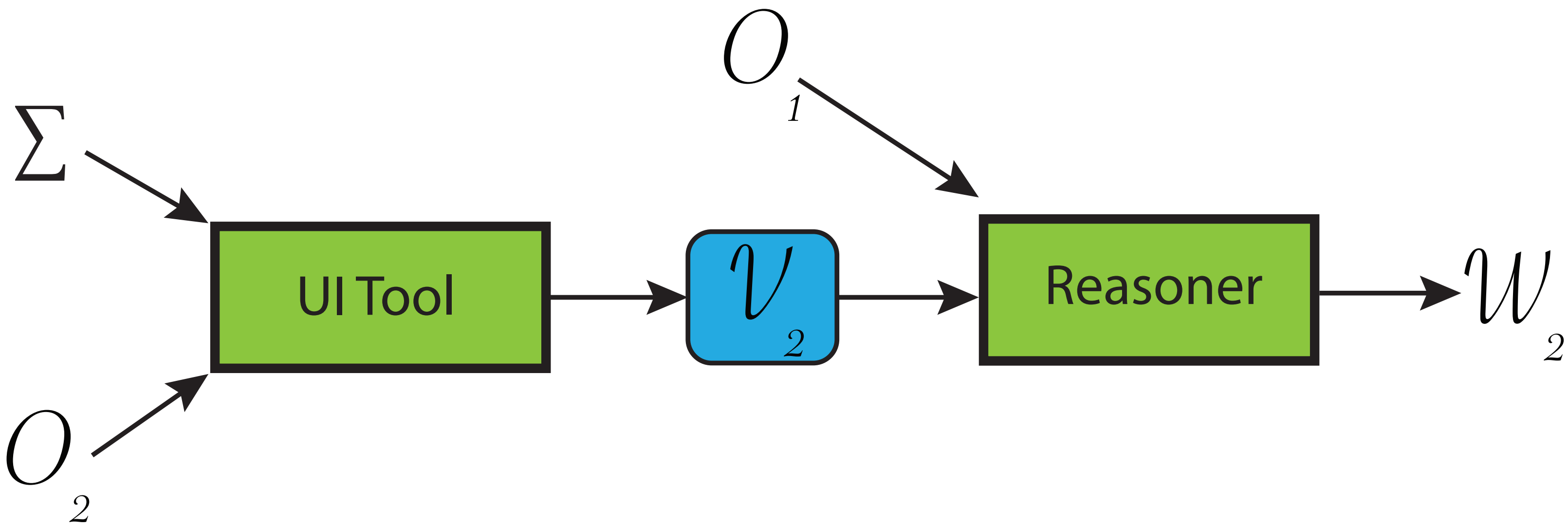
(ii) $\mathcal{O} \models \mathcal{V}$ and $\mathcal{V}$ is strongest such entailment

Useful in many domains:

ontology	$\longrightarrow$	abstraction capturing all info involving Σ
ontology	$\longrightarrow$	restricted view obtained by forgetting $\overline{\Sigma}$
ontology, observation	$\longrightarrow$	abduced hypothesis

For $\mathcal{ALC}$ ontologies and others UI is solvable in an extended language

UI-based logical differences



Algorithm:

(1) Compute uniform interpolant $\mathcal{V}_2$ of $\mathcal{O}_2$ for $\Sigma = \text{sig}(\mathcal{O}_1) \cap \text{sig}(\mathcal{O}_2)$

(2) Compute set of **UI-witnesses**

$$\alpha \in \text{UI-Diff}_\Sigma(\mathcal{O}_1, \mathcal{O}_2) \quad \text{iff} \quad \text{(i) } \text{sig}(\alpha) \subseteq \Sigma$$

$$\text{(ii) } \alpha \in \mathcal{V}_2, \text{ but } \mathcal{O}_1 \not\models \alpha$$

UI-Diff$_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$ is a finite representation of Diff$_\Sigma(\mathcal{O}_1, \mathcal{O}_2)$

UI and forgetting tools

Tool	Complete	Method
NUI	no	depth-bounded uniform interpolation
LETHE	yes	uniform interpolation
Fame 1.0	no	semantic forgetting

All too slow for SNOMED CT

Contribution: New forgetting method

- The first method capable of computing the logical differences between very large ALC-ontologies
- Compute finite set of the differences set Diff($\mathcal{O}_1$,$\mathcal{O}_2$) using UI method
- Can handle cyclic ALC-ontologies by introducing definer names in a conservative manner
- Uses Ackermann rules with purification and combination rules to eliminate symbols in a fast and efficient way

Evaluation of the forgetting method

Tool	$\#\mathcal{F}$ ($\%\mathcal{F}$)	Time	T.O.	S. Rate	Extra
Prototype	199 (10%)	0.7s	1.8%	**96.4%**	**1.8%**
	597 (30%)	1.5s	2.5%	**93.7%**	**3.8%**
	995 (50%)	2.4s	4.8%	**90.1%**	**5.1%**
LETHE	199 (10%)	25.6s	8.8%	81.4%	9.8%
	597 (30%)	75.3s	19.0%	65.0%	16.0%
	995 (50%)	127.1s	29.3%	50.5%	20.2%
Fame 1.0	199 (10%)	0.6s	1.8%	87.2%	11.0%
	597 (30%)	1.3s	2.5%	73.5%	24.0%
	995 (50%)	2.0s	4.8%	66.7%	28.5%

At least 25 times faster than Lethe. Up to 24 % points better success rates than Fame 1.0

Corpus: 396 slightly-adjusted NCBO BioPortal $\mathcal{ALC}$-ontologies
Intel Core i7-4790 4 core CPU, 3.60 GHz, 8GB, DDR3-1600 MHz RAM
1000 seconds timeout

SNOMED CT logical difference evaluation

SNOMED CT ontologies: Core Jan., Core July, Australia extension, Canada extension (2017)

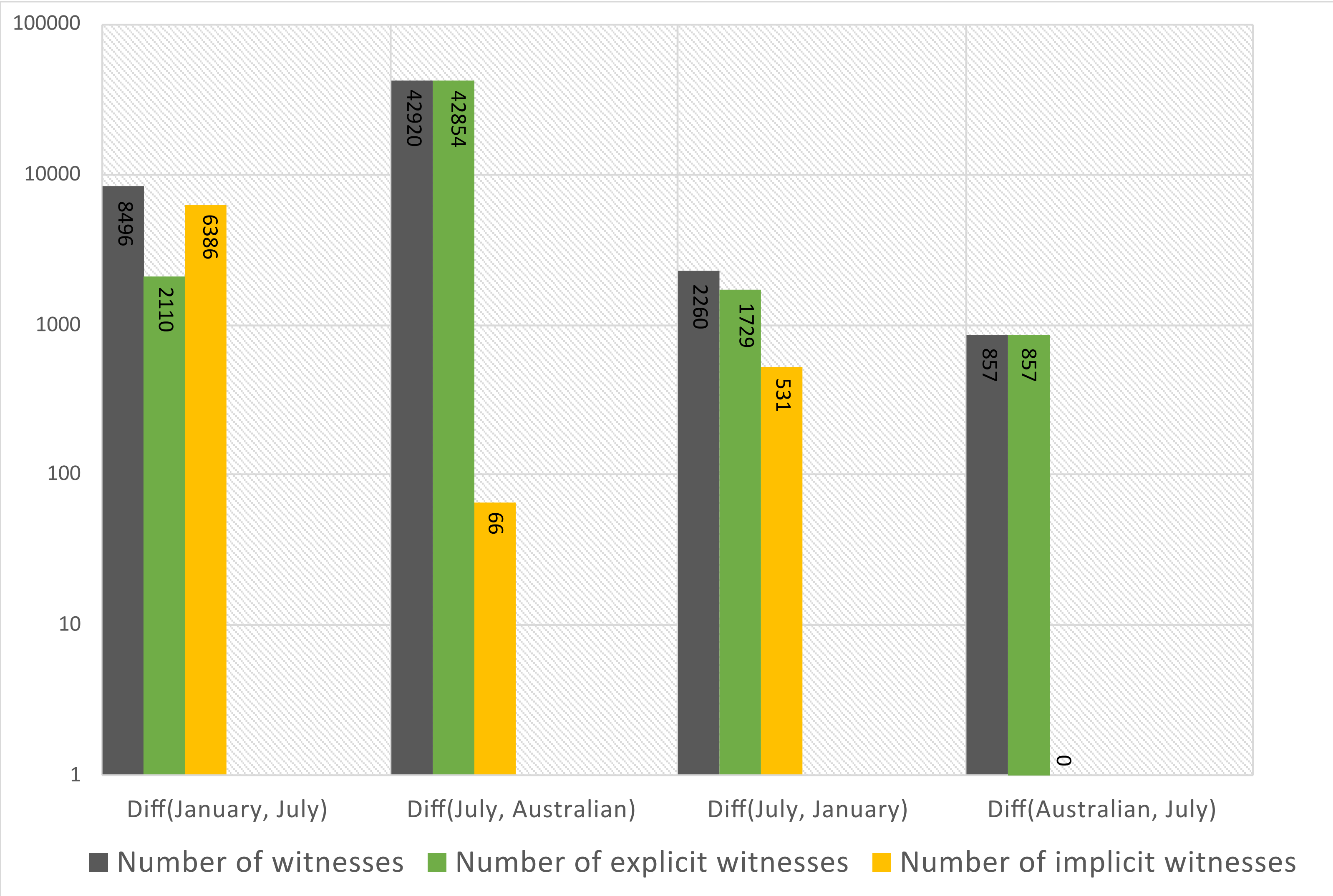
Statistics for forgetting tasks within logical difference implementation:

Case	**UI-Diff($\mathcal{O}_1$, $\mathcal{O}_2$)**	$\#\mathcal{F}_{\mathbf{C}}$	$\#\mathcal{F}_{\mathbf{R}}$	**#cls_set**
1	**UI-Diff**(January, July)	10696	17	648080
2	**UI-Diff**(July, January)	614	0	630483
3	**UI-Diff**(July, Australian)	102880	15	1435778
4	**UI-Diff**(Australian, July)	6	0	648080
5	**UI-Diff**(July, Canadian)	1700	0	650341
6	**UI-Diff**(Canadian, July)	0	0	648080

$\#\mathcal{F}_{\mathbf{C}}$: Number of concepts   #**cls_set**: Number of clauses

$\#\mathcal{F}_{\mathbf{R}}$: Number of roles

SNOMED CT logical difference evaluation



Canadian extension is a conservative extension of the core July 2017 edition

NCIt logical difference evaluation

NCIt ontologies: 8 versions, January to August (2018)

Only information gained was computed

Statistics for forgetting tasks within logical difference implementation:

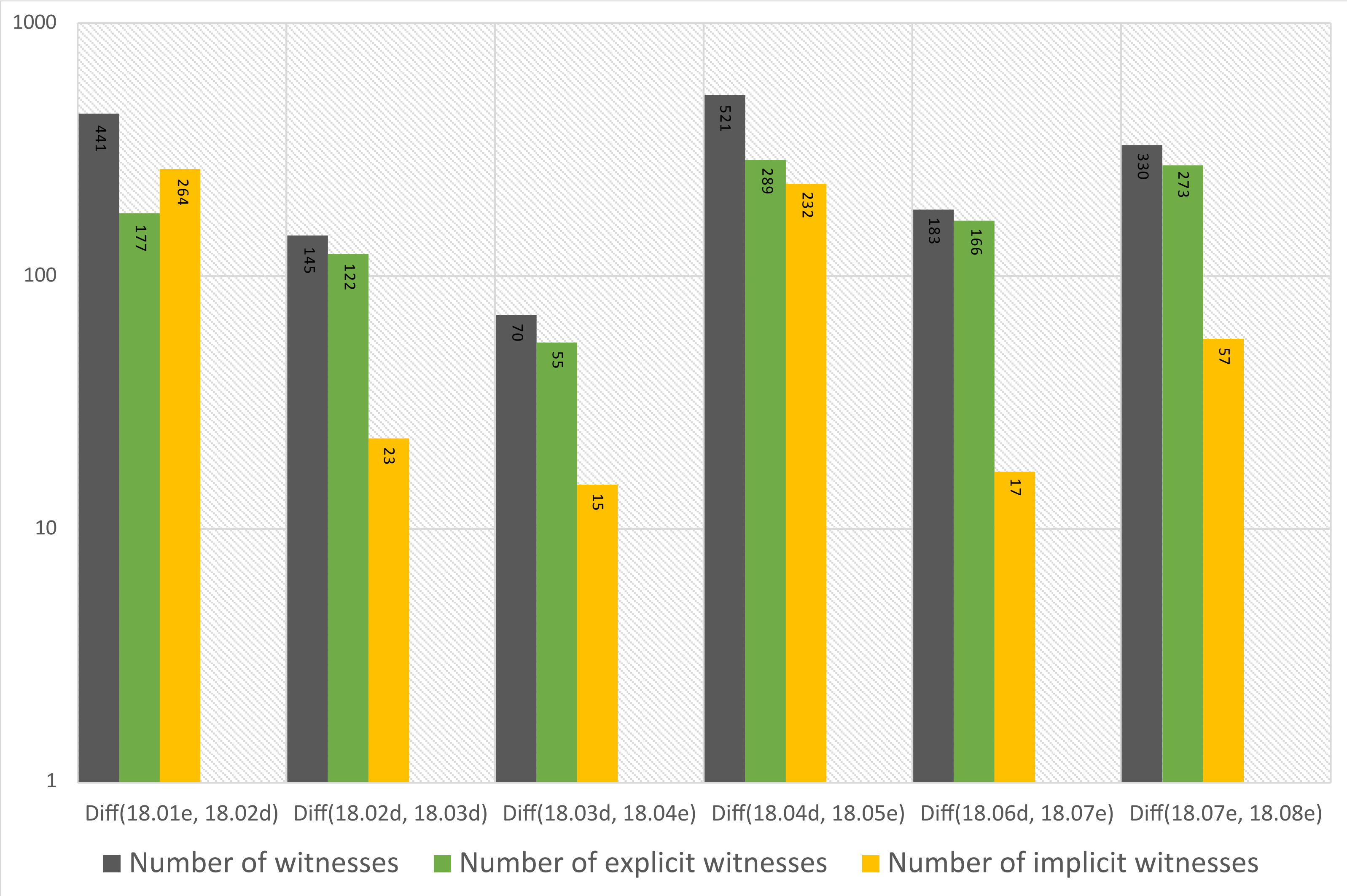
Case	**UI-Diff($\mathcal{O}_1$, $\mathcal{O}_2$)**	$\#\mathcal{F}_{\mathbf{C}}$	$\#\mathcal{F}_{\mathbf{R}}$	#**cls_set**
1	**UI-Diff**(18.01e, 18.02d)	3719	0	283326
2	**UI-Diff**(18.02d, 18.03d)	963	0	284806
3	**UI-Diff**(18.03d, 18.04e)	1294	0	286861
4	**UI-Diff**(18.04e, 18.05d)	2404	0	291462
5	**UI-Diff**(18.06d, 18.07e)	299	0	292091
6	**UI-Diff**(18.07e, 18.08e)	778	0	293426

$\#\mathcal{F}_{\mathbf{C}}$: Number of concepts

#**cls_set**: Number of clauses

$\#\mathcal{F}_{\mathbf{R}}$: Number of roles

NCIt logical difference evaluation



Concluding remarks

Contributions

- Practical logical difference analysis tool for very large ontologies
- Based on new high-performance deductive forgetting method and tool
- Gives ontology engineers a powerful tool for tracking changes in real-world ontologies

Future work

- Eliminate the use of external reasoner