Deep Deductive Reasoners

Pascal Hitzler

Data Semantics Laboratory (DaSe Lab)

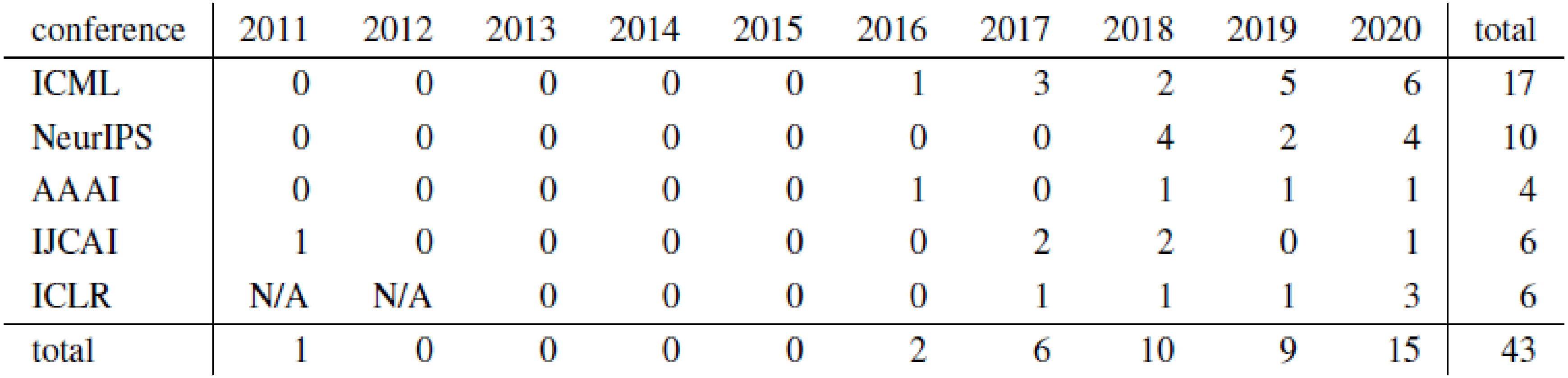
Kansas State University

http://www.daselab.org

Dutch National Graduate School SIKS, May 2021

Neuro-symbolic AI

Publications on neuro-symbolic AI in major conferences (research papers only):



conference	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	total
ICML	0	0	0	0	0	1	3	2	5	6	17
NeurIPS	0	0	0	0	0	0	0	4	2	4	10
AAAI	0	0	0	0	0	1	0	1	1	1	4
IJCAI	1	0	0	0	0	0	2	2	0	1	6
ICLR	N/A	N/A	0	0	0	0	1	1	1	3	6
total	1	0	0	0	0	2	6	10	9	15	43

See

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, Pascal Hitzler

Neuro-Symbolic Artificial Integration: Current Trends

https://arxiv.org/abs/2105.05330 (under review)

for more analysis.

Deep Deductive Reasoners

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, Pascal Hitzler,
Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners.
Applied Intelligence, 2021, to appear.

Pascal Hitzler, Frank van Harmelen
A reasonable Semantic Web.
Semantic Web 1 (1-2), 39-44, 2010.

Deep Deductive Reasoners

- We trained deep learning systems to do deductive reasoning.

- Why is this interesting?
  - For dealing with noisy data (where symbolic reasoners do very poorly).
  - For speed, as symbolic algorithms are of very high complexity.
  - Out of principle because we want to learn about the capabilities of deep learning for complicated cognitive tasks.

  - To perhaps begin to understand how our (neural) brains can learn to do highly symbolic tasks like formal logical reasoning, or in more generality, mathematics.
    A fundamental quest in Cognitive Science.

Reasoning as Classification

- Given a set of logical formulas (a theory).

- Any formula expressible over the same language is either
  - a logical consequence or
  - not a logical consequence.

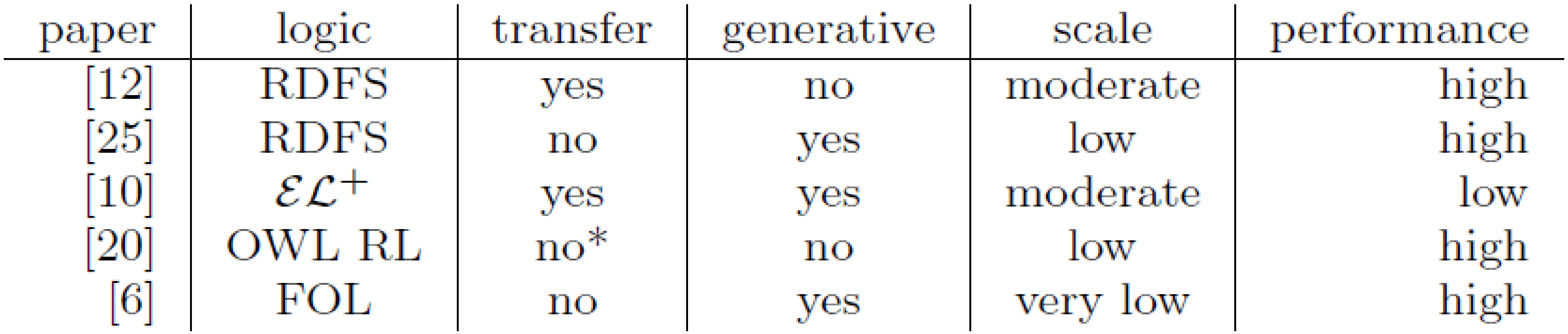
- This can be understood as a classification problem for machine learning.

- It turns out to be a really hard machine learning problem.

Knowledge Materialization

- Given a set of logical formulas (a theory).
- Produce all logical consequences under certain constraints.
- Without the qualifier this is in general not possible as the set of all logical consequences is infinite.
- So we have to constrain to consequences of, e.g., a certain syntactic form. For relatively simple logics, this is often reasonably possible.

Published deep deductive reasoning work



paper	logic	transfer	generative	scale	performance
[12]	RDFS	yes	no	moderate	high
[25]	RDFS	no	yes	low	high
[10]	$\mathcal{EL}^+$	yes	yes	moderate	low
[20]	OWL RL	no*	no	low	high
[6]	FOL	no	yes	very low	high

[12]: Ebrahimi, Sarker, Bianchi, Xie, Eberhart, Doran, Kim, Hitzler, AAAI-MAKE 2021

[25]: Makni, Hendler, SWJ 2019

[10]: Eberhart, Ebrahimi, Zhou, Shimizu, Hitzler, AAAI-MAKE 2020

[20]: Hohenecker, Lukasiewicz, JAIR 2020

[6]: Bianchi, Hitzler, AAAI-MAKE 2019

Deep Reasoners Overview

1. RDFS via Memory Networks (classification) [12].
2. RDFS via Pointer Networks (generative) [new].
3. OWL EL via LSTMs (generative) [10].
4. LTNs for first-order predicate logic [6].

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, Pascal Hitzler, Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners. Applied Intelligence, 2021, to appear. [covers 6,10,12]

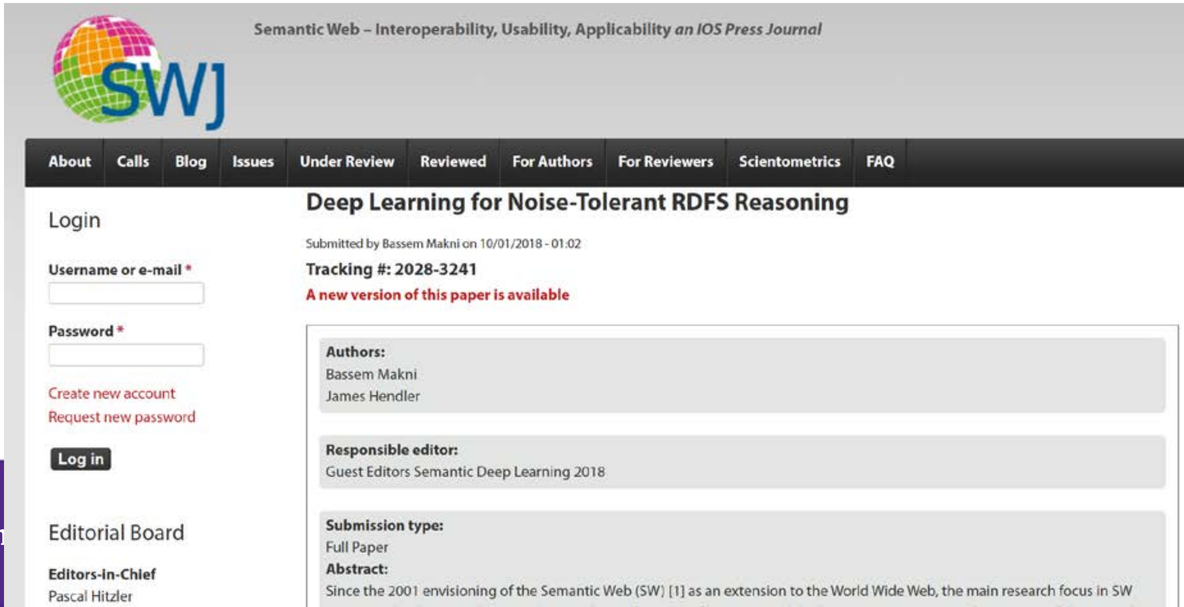
RDFS Reasoning using Memory Networks

Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie,
Aaron Eberhart, Derek Doran, Hyeongsik Kim, Pascal Hitzler,
Neuro-Symbolic Deductive Reasoning for Cross-Knowledge Graph Entailment.
In: Proc. AAAI-MAKE 2021.

additional analysis by Sulogna Chowdhury, Aaron Eberhart
and Brayden Pankaskie

RDF reasoning

- Essentially, RDF reasoning is Datalog reasoning restricted to:
  - Unary and binary predicates only.
  - A fixed set of rules that are not facts.
- You can try the following:
  - Use a vector embedding for one RDF graph.
  - Create all logical consequences.
  - Throw n% of them away.
  - Use the rest to train a DL system.
  - Check how many of those you threw away can be recovered this way.



Semantic Web – Interoperability, Usability, Applicability *an IOS Press Journal*

Deep Learning for Noise-Tolerant RDFS Reasoning

Submitted by Bassem Makni on 10/01/2018 - 01:02

Tracking #: 2028-3241

A new version of this paper is available

Authors:
Bassem Makni
James Hendler

Responsible editor:
Guest Editors Semantic Deep Learning 2018

Submission type:
Full Paper

Abstract:
Since the 2001 envisioning of the Semantic Web (SW) [1] as an extension to the World Wide Web, the main research focus in SW

Editorial Board

Editors-in-Chief
Pascal Hitzler

RDF reasoning

- The problem with the approach just described:
  - It works only with that graph.

- What you'd really like to do is:
  - Train a deep learning system so that you can present a new, unseen graph to it, and it can correctly derive the deductively inferred triples.

- Note:
  - You don't know the IRIs in the graph up front. The only overlap may or may not be the IRIs in the rdf/s namespace.
  - You don't know up front how "deep" the reasoning needs to be.
  - There is no lack of training data, it can be auto-generated.

RDF reasoning

- [Note: RDF is one of the simplest useful knowledge representation languages that is not propositional.]

- Think knowledge graph.
- Think node-edge-node triples such as

BarackObama	rdf:type	President
BarackObama	husbandOf	MichelleObama
President	rdfs:subClassOf	Human
husbandOf	rdfs:subPropertyOf	spouseOf

- Then there is a (fixed, small) set of inference rules, such as

  rdf:type(x,y) AND rdfs:subClassOf(y,z)THEN rdf:type(x,z)

Representation

- Goal is to be able to reason over unseen knowledge graphs. I.e. the out-of-vocabulary problem needs addressing.

- Normalization of vocabulary (i.e., it becomes shared vocabulary across all input knowledge graphs.
- One vocabulary item becomes a one-hot vector (dimension d, number of normalized vocabulary terms)
- One triple becomes a 3 x d matrix.
- The knowledge graph becomes an n x 3 x d tensor (n is the number of knowledge graph triples)

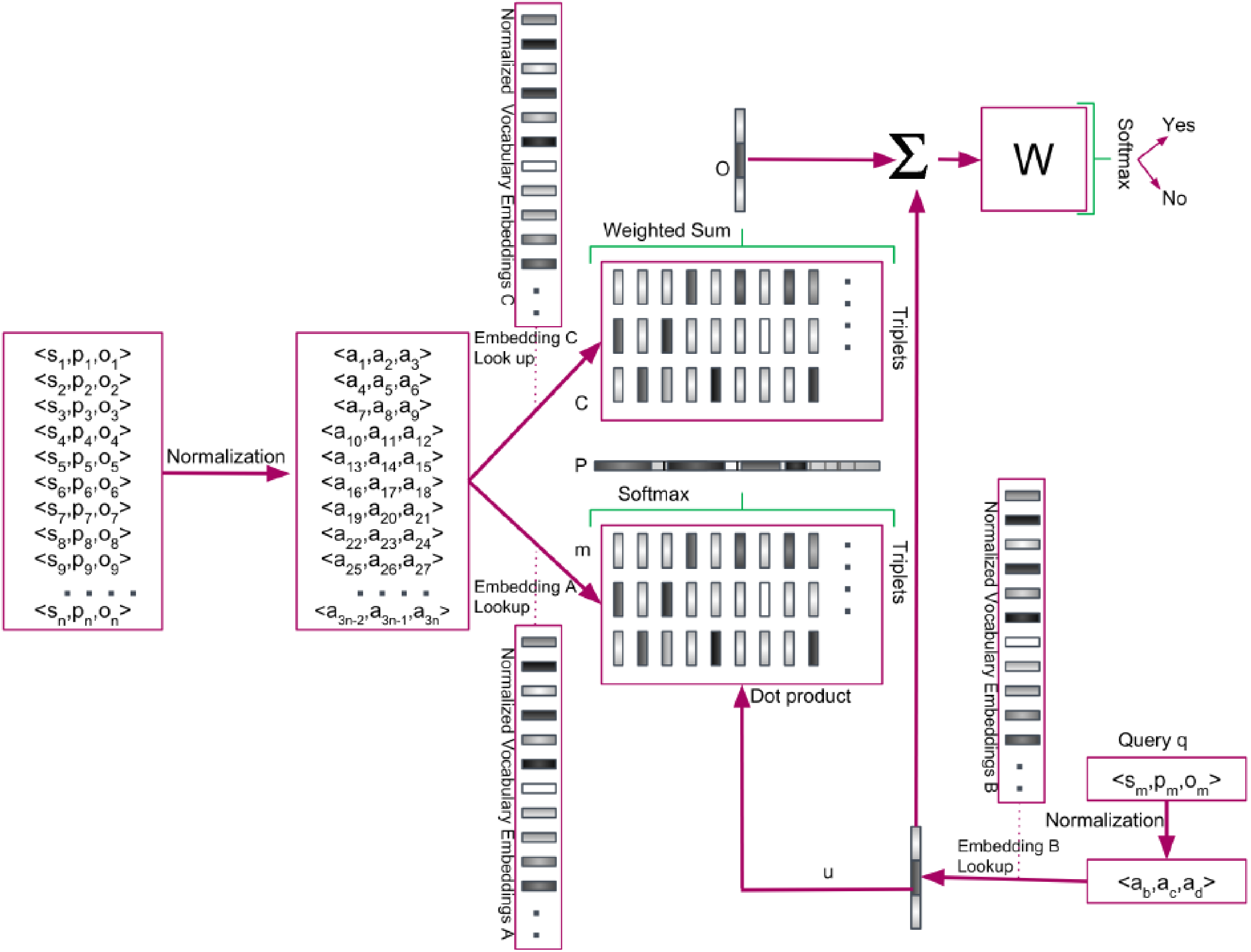
- Knowledge graph is stored in “memory”

Mechanics

- **An attention mechanism retrieves memory slots useful for finding the correct answer to a query.**
- **These are combined with the query and run through a (learned) matrix to retrieve a new (processed) query.**

- **This is repeated (in our experiment with 10 “hops”).**
- **The final out put is a yes/no answer to the query.**

Memory Network based on MemN2N



$<s_1,p_1,o_1>$
$<s_2,p_2,o_2>$
$<s_3,p_3,o_3>$
$<s_4,p_4,o_4>$
$<s_5,p_5,o_5>$
$<s_6,p_6,o_6>$
$<s_7,p_7,o_7>$
$<s_8,p_8,o_8>$
$<s_9,p_9,o_9>$
. . . .
$<s_n,p_n,o_n>$

Normalization

$<a_1,a_2,a_3>$
$<a_4,a_5,a_6>$
$<a_7,a_8,a_9>$
$<a_{10},a_{11},a_{12}>$
$<a_{13},a_{14},a_{15}>$
$<a_{16},a_{17},a_{18}>$
$<a_{19},a_{20},a_{21}>$
$<a_{22},a_{23},a_{24}>$
$<a_{25},a_{26},a_{27}>$
. . . .
$<a_{3n-2},a_{3n-1},a_{3n}>$

Normalized Vocabulary Embeddings C

Embedding C Look up

C

Weighted Sum

Triplets

O

Σ

W

Softmax

Yes

No

P

Softmax

m

Triplets

Embedding A Lookup

Normalized Vocabulary Embeddings A

Dot product

Normalized Vocabulary Embeddings B

Query q

$<s_m,p_m,o_m>$

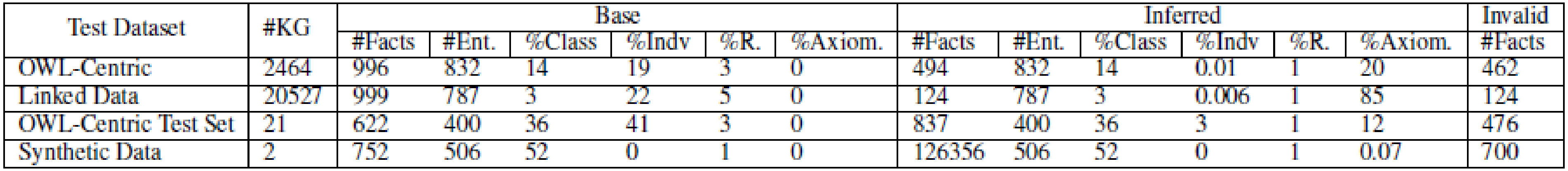
Normalization

Embedding B Lookup

$<a_b,a_c,a_d>$

u

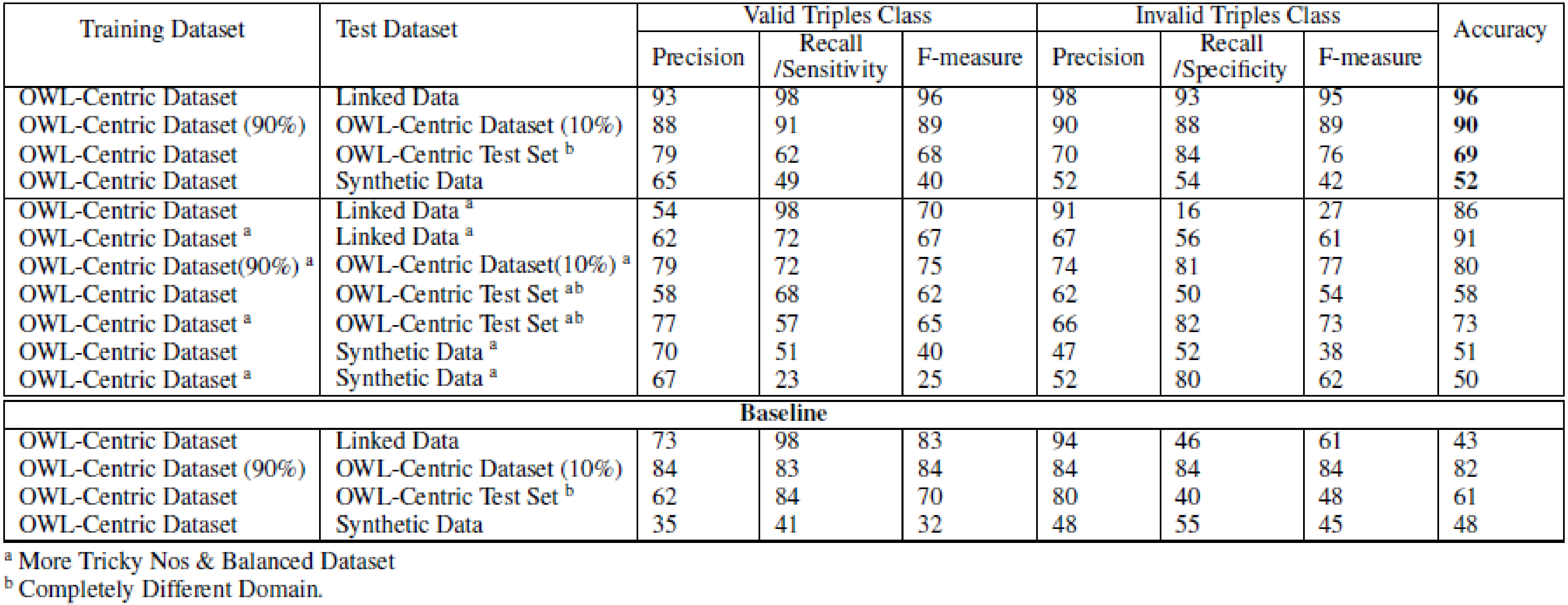
Experiments: Performance



Test Dataset	#KG	Base						Inferred						Invalid
		#Facts	#Ent.	%Class	%Indv	%R.	%Axiom.	#Facts	#Ent.	%Class	%Indv	%R.	%Axiom.	#Facts
OWL-Centric	2464	996	832	14	19	3	0	494	832	14	0.01	1	20	462
Linked Data	20527	999	787	3	22	5	0	124	787	3	0.006	1	85	124
OWL-Centric Test Set	21	622	400	36	41	3	0	837	400	36	3	1	12	476
Synthetic Data	2	752	506	52	0	1	0	126356	506	52	0	1	0.07	700

Table 2: Statistics of various datasets used in experiments

Baseline: non-normalized embeddings, same architecture



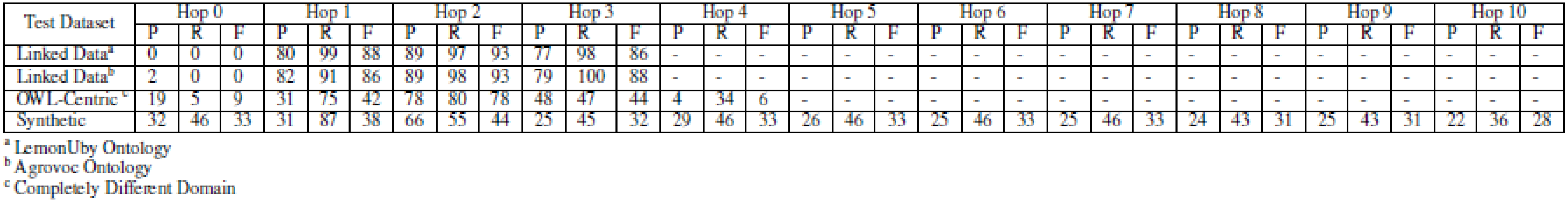
Training Dataset	Test Dataset	Valid Triples Class			Invalid Triples Class			Accuracy
		Precision	Recall /Sensitivity	F-measure	Precision	Recall /Specificity	F-measure	
OWL-Centric Dataset	Linked Data	93	98	96	98	93	95	**96**
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	88	91	89	90	88	89	**90**
OWL-Centric Dataset	OWL-Centric Test Set [b]	79	62	68	70	84	76	**69**
OWL-Centric Dataset	Synthetic Data	65	49	40	52	54	42	**52**
OWL-Centric Dataset	Linked Data [a]	54	98	70	91	16	27	86
OWL-Centric Dataset [a]	Linked Data [a]	62	72	67	67	56	61	91
OWL-Centric Dataset(90%) [a]	OWL-Centric Dataset(10%) [a]	79	72	75	74	81	77	80
OWL-Centric Dataset	OWL-Centric Test Set [ab]	58	68	62	62	50	54	58
OWL-Centric Dataset [a]	OWL-Centric Test Set [ab]	77	57	65	66	82	73	73
OWL-Centric Dataset	Synthetic Data [a]	70	51	40	47	52	38	51
OWL-Centric Dataset [a]	Synthetic Data [a]	67	23	25	52	80	62	50
Baseline								
OWL-Centric Dataset	Linked Data	73	98	83	94	46	61	43
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	84	83	84	84	84	84	82
OWL-Centric Dataset	OWL-Centric Test Set [b]	62	84	70	80	40	48	61
OWL-Centric Dataset	Synthetic Data	35	41	32	48	55	45	48

[a] More Tricky Nos & Balanced Dataset

[b] Completely Different Domain.

Table 3: Experimental results of proposed model

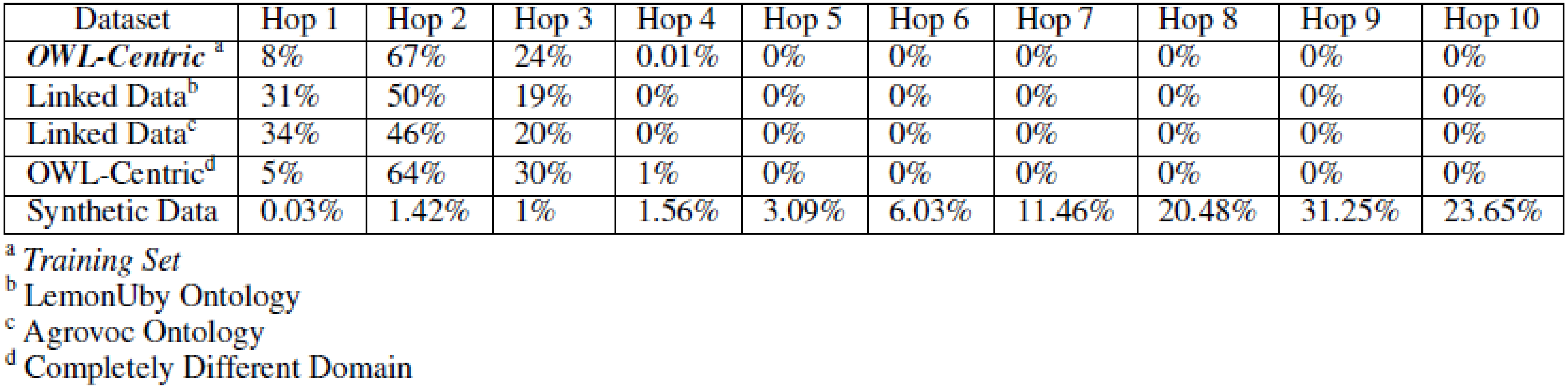
Experiments: Reasoning Depth



Test Dataset	Hop 0			Hop 1			Hop 2			Hop 3			Hop 4			Hop 5			Hop 6			Hop 7			Hop 8			Hop 9			Hop 10		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Linked Data[a]	0	0	0	80	99	88	89	97	93	77	98	86	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Linked Data[b]	2	0	0	82	91	86	89	98	93	79	100	88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OWL-Centric [c]	19	5	9	31	75	42	78	80	78	48	47	44	4	34	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Synthetic	32	46	33	31	87	38	66	55	44	25	45	32	29	46	33	26	46	33	25	46	33	25	46	33	24	43	31	25	43	31	22	36	28

[a] LemonUby Ontology
[b] Agrovoc Ontology
[c] Completely Different Domain

Table 4: Experimental results over each reasoning hop



Dataset	Hop 1	Hop 2	Hop 3	Hop 4	Hop 5	Hop 6	Hop 7	Hop 8	Hop 9	Hop 10
OWL-Centric [a]	8%	67%	24%	0.01%	0%	0%	0%	0%	0%	0%
Linked Data[b]	31%	50%	19%	0%	0%	0%	0%	0%	0%	0%
Linked Data[c]	34%	46%	20%	0%	0%	0%	0%	0%	0%	0%
OWL-Centric[d]	5%	64%	30%	1%	0%	0%	0%	0%	0%	0%
Synthetic Data	0.03%	1.42%	1%	1.56%	3.09%	6.03%	11.46%	20.48%	31.25%	23.65%

[a] *Training Set*
[b] LemonUby Ontology
[c] Agrovoc Ontology
[d] Completely Different Domain

Table 5: Data distribution per knowledge graph over each reasoning hop

Training time: just over a full day

Generative RDFS Reasoning using Pointer Networks

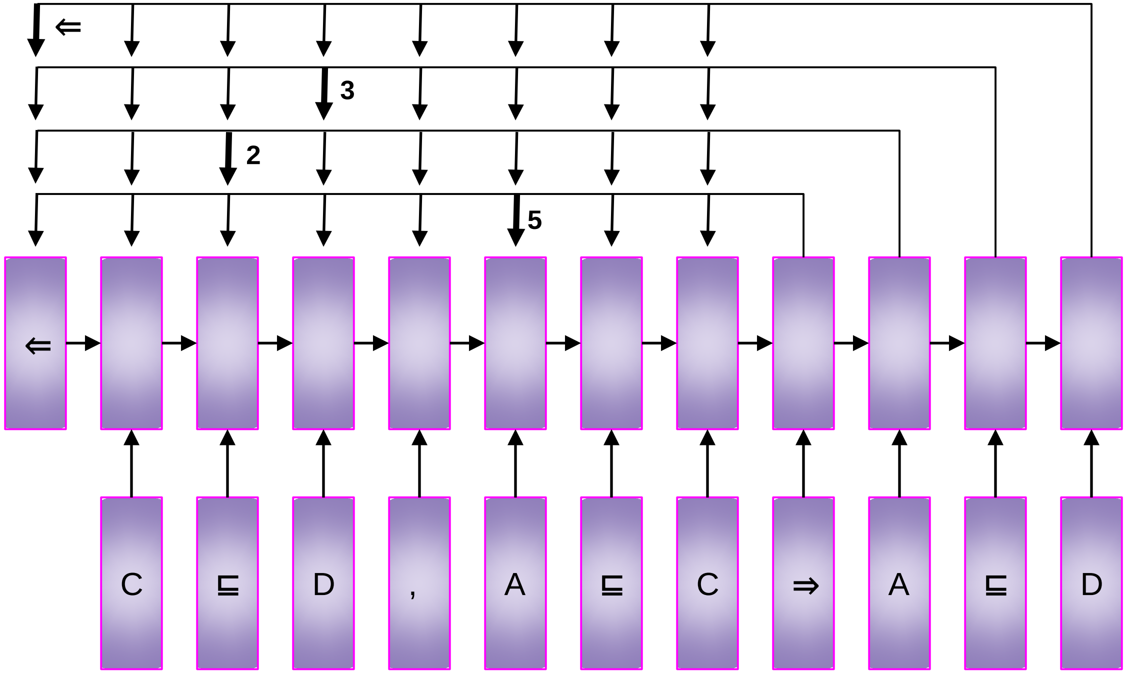
Monireh Ebrahimi, breaking results

Pointer Networks

- Pointer Networks ‘point’ to input elements!
- Ptr-Net approach specifically targets problems whose outputs are discrete and correspond to positions in the input.
- At each time step, the distribution of the attention is the answer!
- Application:
  - NP-hard Travelling Salesman Problem (TSP)
  - Delaunay Triangulation
  - Convex Hull
  - Text Summarization
  - Code completion
  - Dependency Parsing

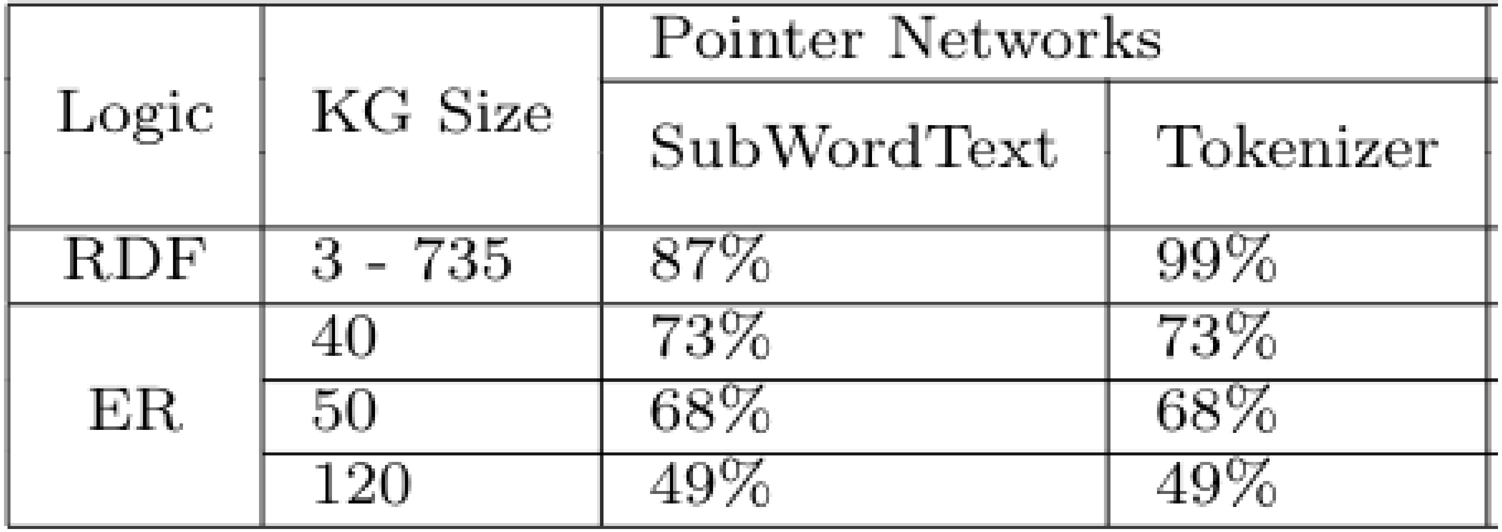
Pointer Networks for Reasoning

- To mimic human reasoning behaviour where one can learn to choose a set of symbols in different locations and copy these symbols to suitable locations to generate new logical consequences based on a set of predefined logical entailment rules



$$C \sqsubseteq D, A \sqsubseteq C \mapsto A \sqsubseteq D$$

Preliminary Results



Logic	KG Size	Pointer Networks	
		SubWordText	Tokenizer
RDF	3 - 735	87%	99%
ER	40	73%	73%
	50	68%	68%
	120	49%	49%

- On RDF, slightly outperforms [Hendler Makni SWJ 2019] approach.
- Our approach is a more straightforward application.
- Evaluation is on the same dataset.

Completion Reasoning Emulation for the Description Logic EL+

Aaron Eberhart, Monireh Ebrahimi, Lu Zhou, Cogan Shimizu, Pascal Hitzler, Completion Reasoning Emulation for the Description Logic EL+.
In: Andreas Martin, Knut Hinkelmann, Hans-Georg Fill, Aurona Gerber, Doug Lenat, Reinhard Stolle, Frank van Harmelen (eds.), Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE 2020, Palo Alto, CA, USA, March 23-25, 2020, Volume I.

EL+ is essentially OWL 2 EL

CX ⊑ CY

CX ⊓ CY ⊑ CZ

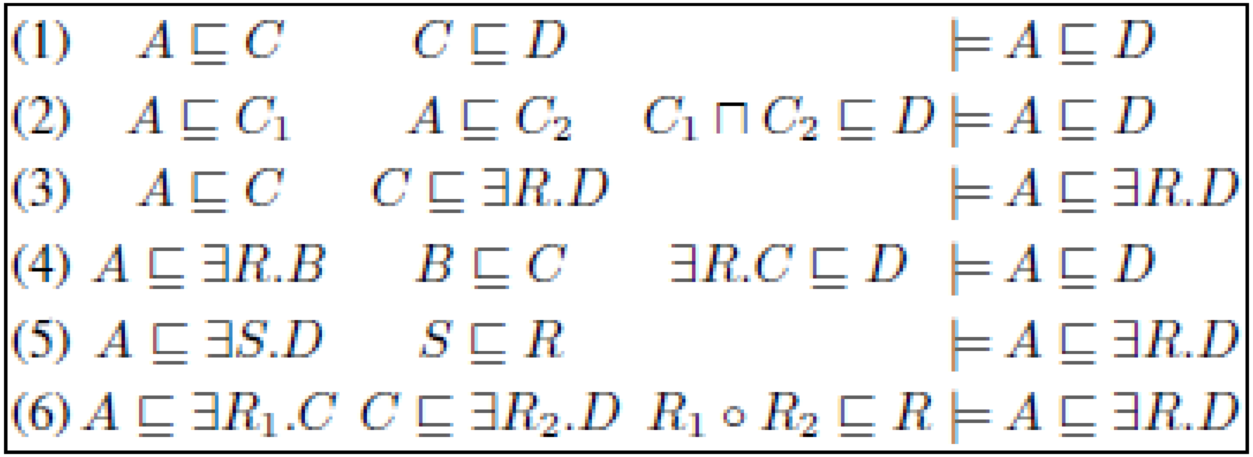
CX ⊑ ∃RY.CZ

∃RX.CY ⊑ CZ

RX ⊑ RY

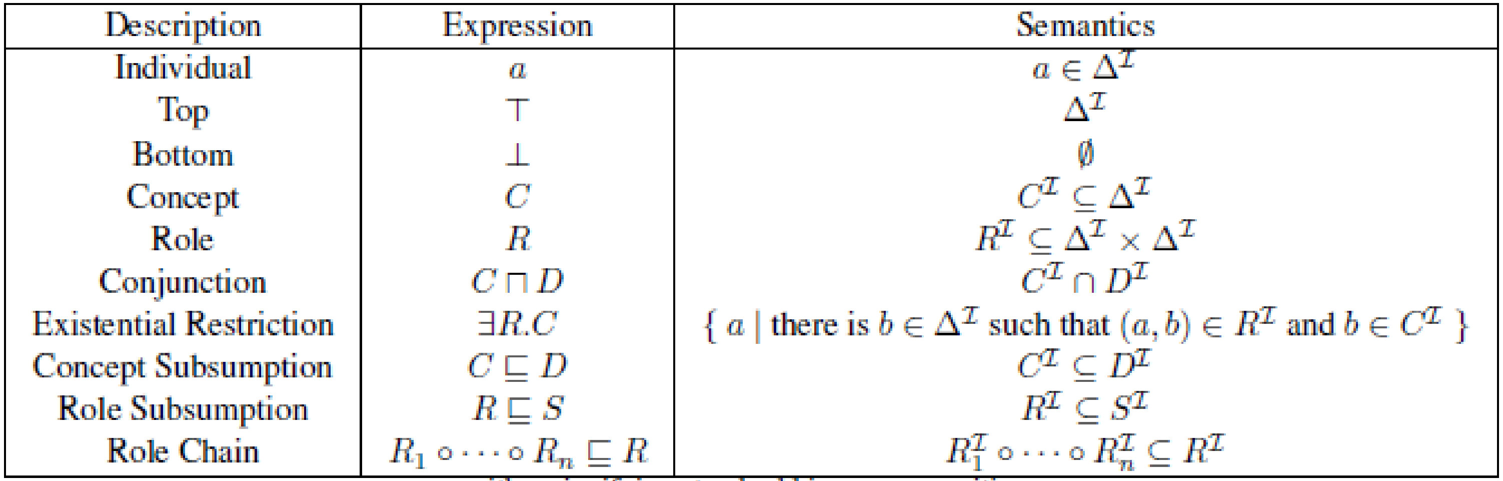
RX ∘ RY ⊑ RZ

Table 2: $\mathcal{EL}^+$ Completion Rules



(1)	$A \sqsubseteq C$	$C \sqsubseteq D$		$\models A \sqsubseteq D$
(2)	$A \sqsubseteq C_1$	$A \sqsubseteq C_2$	$C_1 \sqcap C_2 \sqsubseteq D$	$\models A \sqsubseteq D$
(3)	$A \sqsubseteq C$	$C \sqsubseteq \exists R.D$		$\models A \sqsubseteq \exists R.D$
(4)	$A \sqsubseteq \exists R.B$	$B \sqsubseteq C$	$\exists R.C \sqsubseteq D$	$\models A \sqsubseteq D$
(5)	$A \sqsubseteq \exists S.D$	$S \sqsubseteq R$		$\models A \sqsubseteq \exists R.D$
(6)	$A \sqsubseteq \exists R_1.C$	$C \sqsubseteq \exists R_2.D$	$R_1 \circ R_2 \sqsubseteq R$	$\models A \sqsubseteq \exists R.D$

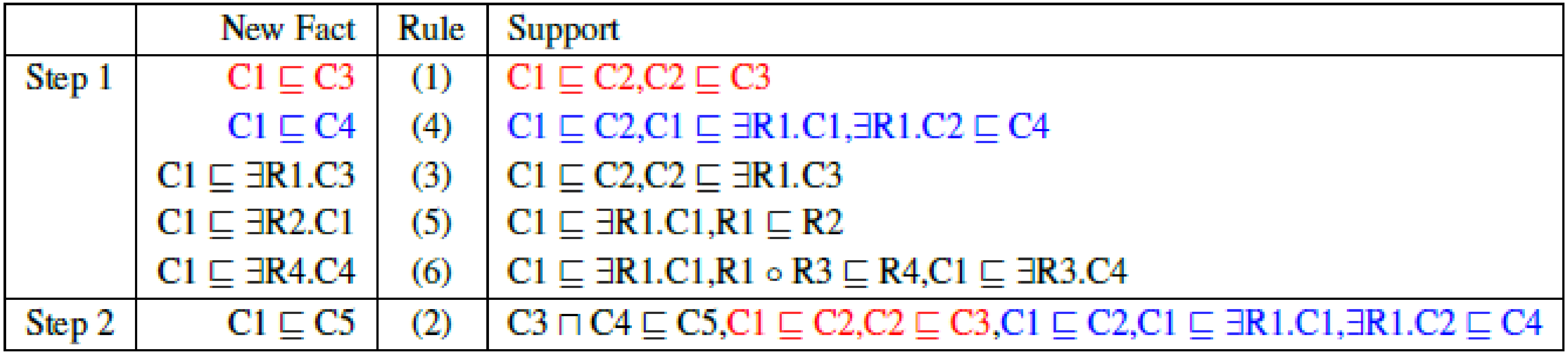
Table 1: $\mathcal{EL}^+$ Semantics



Description	Expression	Semantics
Individual	a	$a \in \Delta^\mathcal{I}$
Top	$\top$	$\Delta^\mathcal{I}$
Bottom	$\bot$	$\emptyset$
Concept	C	$C^\mathcal{I} \subseteq \Delta^\mathcal{I}$
Role	R	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
Conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
Existential Restriction	$\exists R.C$	$\{ a \mid$ there is $b \in \Delta^\mathcal{I}$ such that $(a, b) \in R^\mathcal{I}$ and $b \in C^\mathcal{I} \}$
Concept Subsumption	$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$
Role Subsumption	$R \sqsubseteq S$	$R^\mathcal{I} \subseteq S^\mathcal{I}$
Role Chain	$R_1 \circ \cdots \circ R_n \sqsubseteq R$	$R_1^\mathcal{I} \circ \cdots \circ R_n^\mathcal{I} \subseteq R^\mathcal{I}$

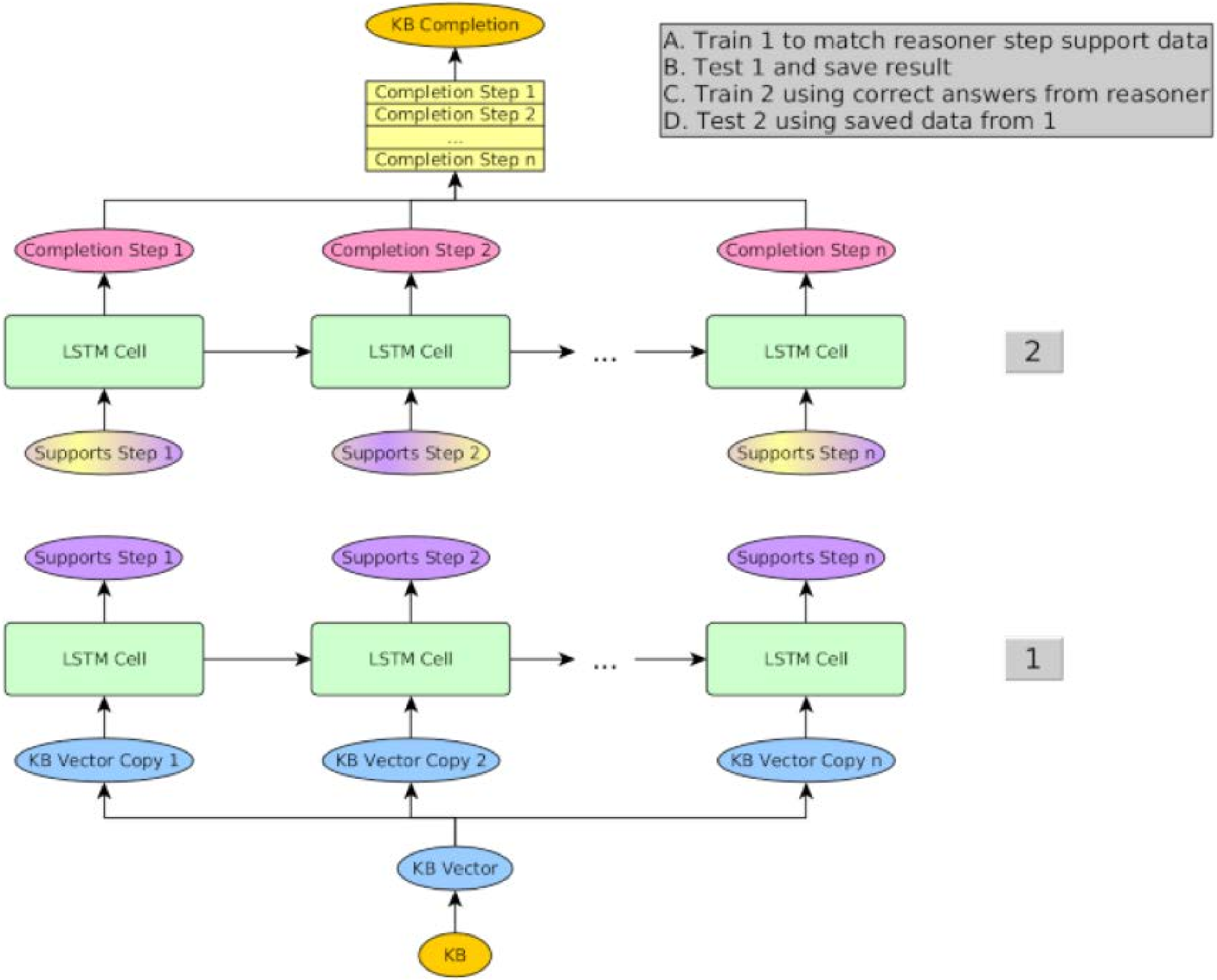
with ∘ signifying standard binary composition

Support



	New Fact	Rule	Support
Step 1	$C1 \sqsubseteq C3$	(1)	$C1 \sqsubseteq C2, C2 \sqsubseteq C3$
	$C1 \sqsubseteq C4$	(4)	$C1 \sqsubseteq C2, C1 \sqsubseteq \exists R1.C1, \exists R1.C2 \sqsubseteq C4$
	$C1 \sqsubseteq \exists R1.C3$	(3)	$C1 \sqsubseteq C2, C2 \sqsubseteq \exists R1.C3$
	$C1 \sqsubseteq \exists R2.C1$	(5)	$C1 \sqsubseteq \exists R1.C1, R1 \sqsubseteq R2$
	$C1 \sqsubseteq \exists R4.C4$	(6)	$C1 \sqsubseteq \exists R1.C1, R1 \circ R3 \sqsubseteq R4, C1 \sqsubseteq \exists R3.C4$
Step 2	$C1 \sqsubseteq C5$	(2)	$C3 \sqcap C4 \sqsubseteq C5, C1 \sqsubseteq C2, C2 \sqsubseteq C3, C1 \sqsubseteq C2, C1 \sqsubseteq \exists R1.C1, \exists R1.C2 \sqsubseteq C4$

Architecture



A. Train 1 to match reasoner step support data
B. Test 1 and save result
C. Train 2 using correct answers from reasoner
D. Test 2 using saved data from 1

Figure 2: Piecewise Architecture

Architecture

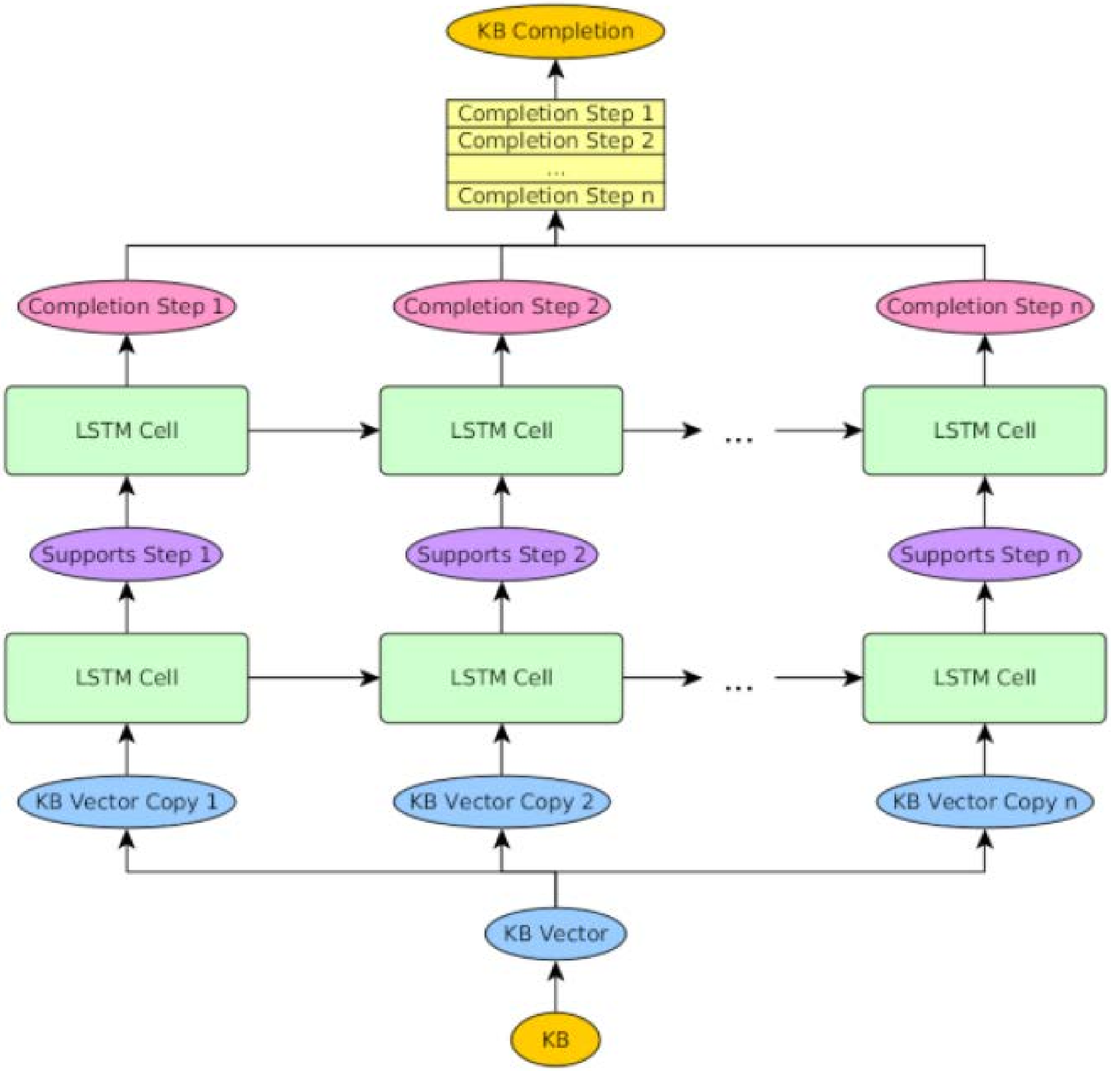


Figure 3: Deep Architecture

b

Architecture

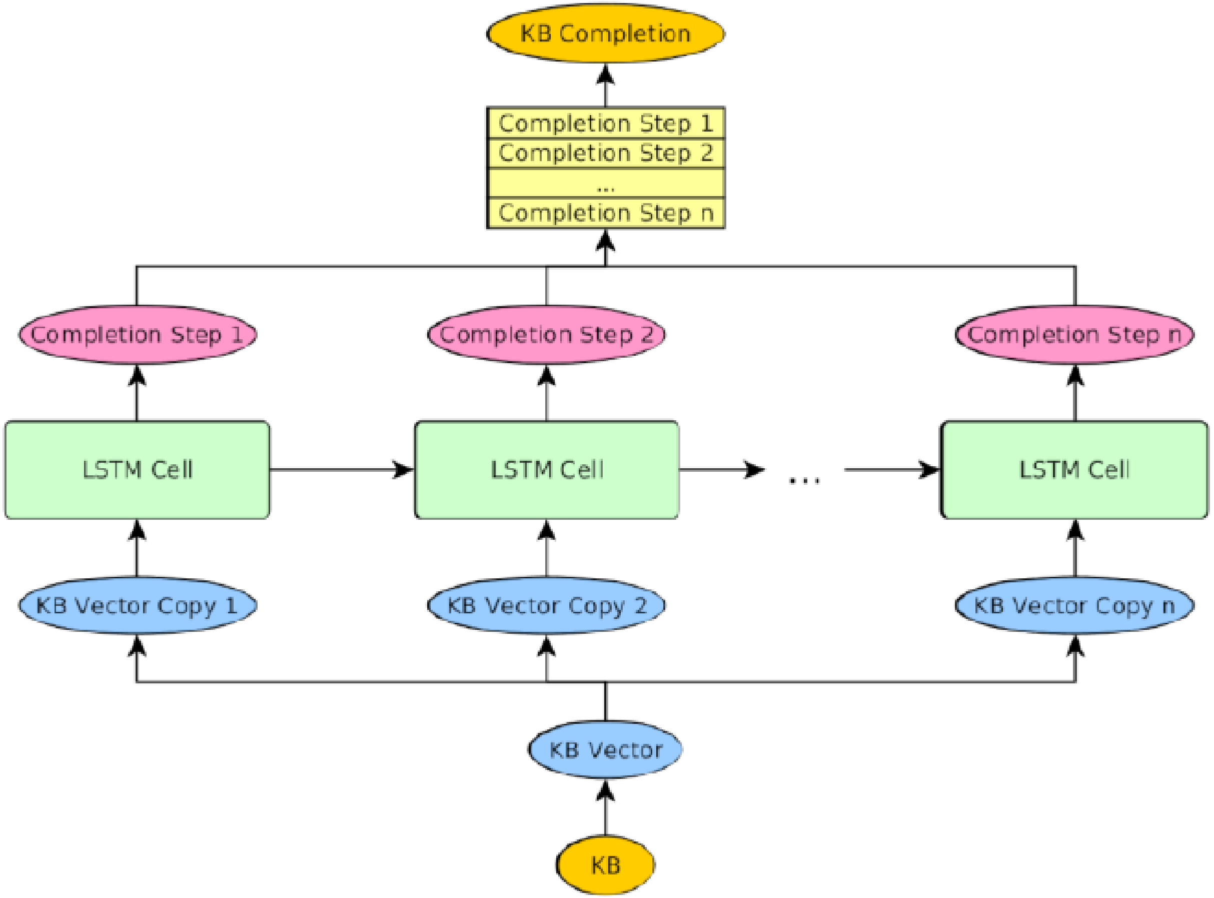
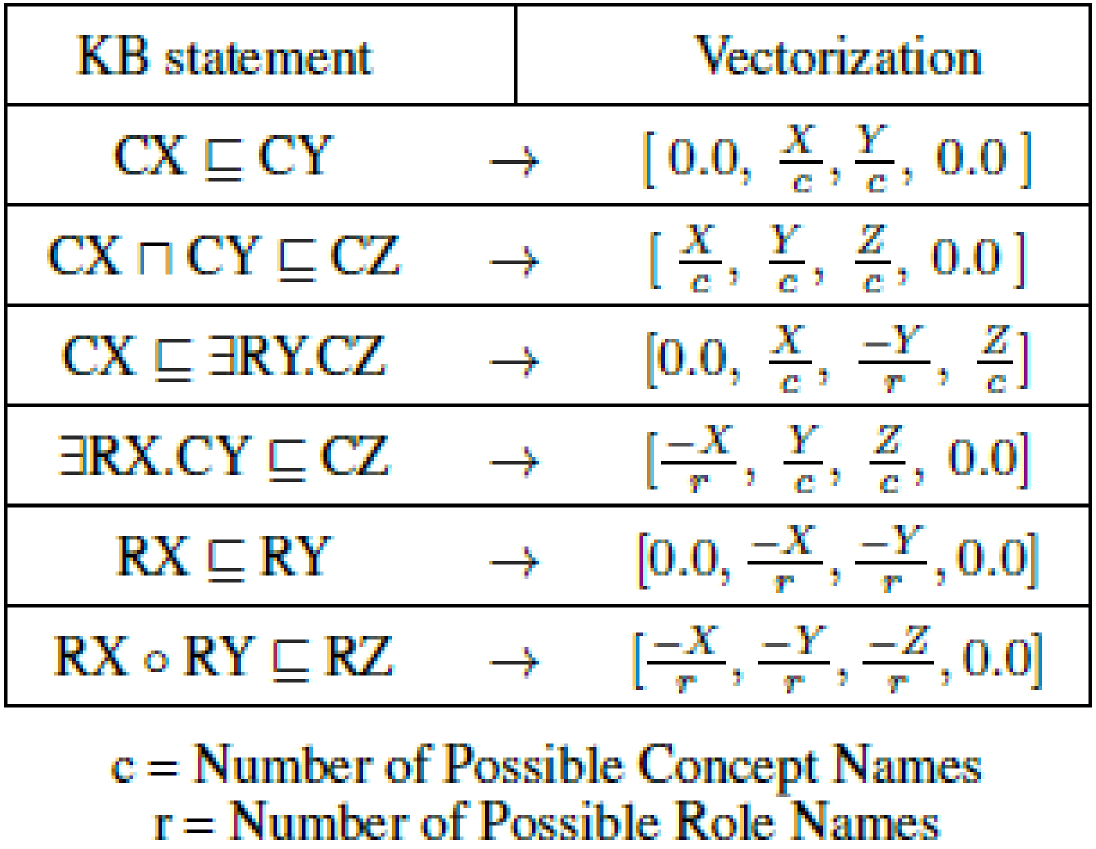


Figure 4: Flat Architecture

Encoding

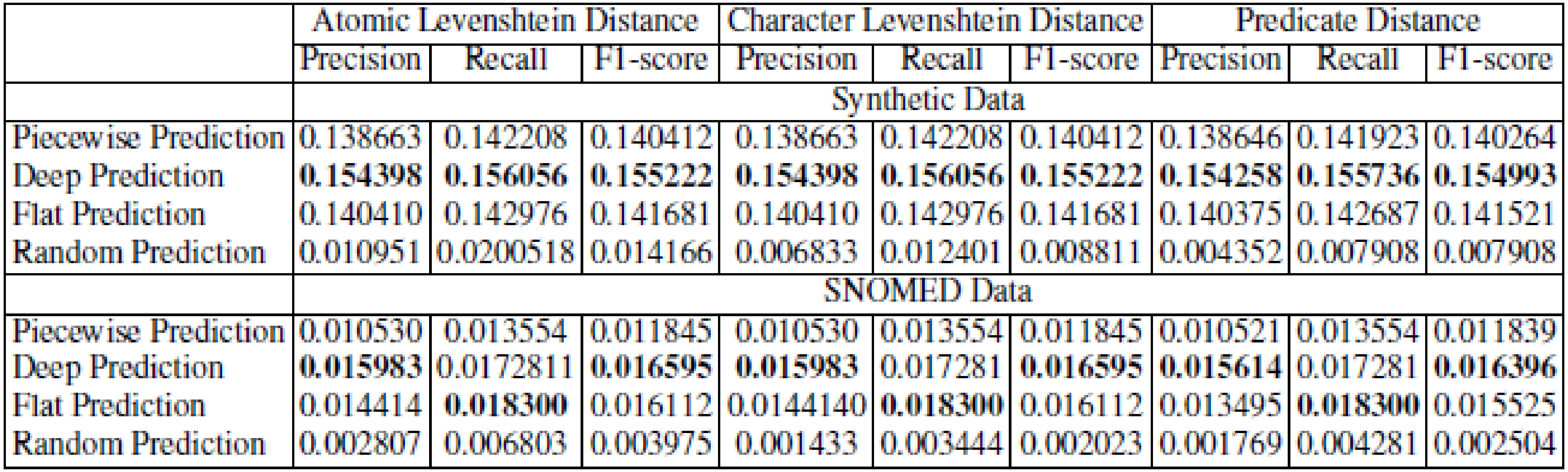


KB statement		Vectorization
$CX \sqsubseteq CY$	$\rightarrow$	$[0.0, \frac{X}{c}, \frac{Y}{c}, 0.0]$
$CX \sqcap CY \sqsubseteq CZ$	$\rightarrow$	$[\frac{X}{c}, \frac{Y}{c}, \frac{Z}{c}, 0.0]$
$CX \sqsubseteq \exists RY.CZ$	$\rightarrow$	$[0.0, \frac{X}{c}, \frac{-Y}{r}, \frac{Z}{c}]$
$\exists RX.CY \sqsubseteq CZ$	$\rightarrow$	$[\frac{-X}{r}, \frac{Y}{c}, \frac{Z}{c}, 0.0]$
$RX \sqsubseteq RY$	$\rightarrow$	$[0.0, \frac{-X}{r}, \frac{-Y}{r}, 0.0]$
$RX \circ RY \sqsubseteq RZ$	$\rightarrow$	$[\frac{-X}{r}, \frac{-Y}{r}, \frac{-Z}{r}, 0.0]$

c = Number of Possible Concept Names
r = Number of Possible Role Names

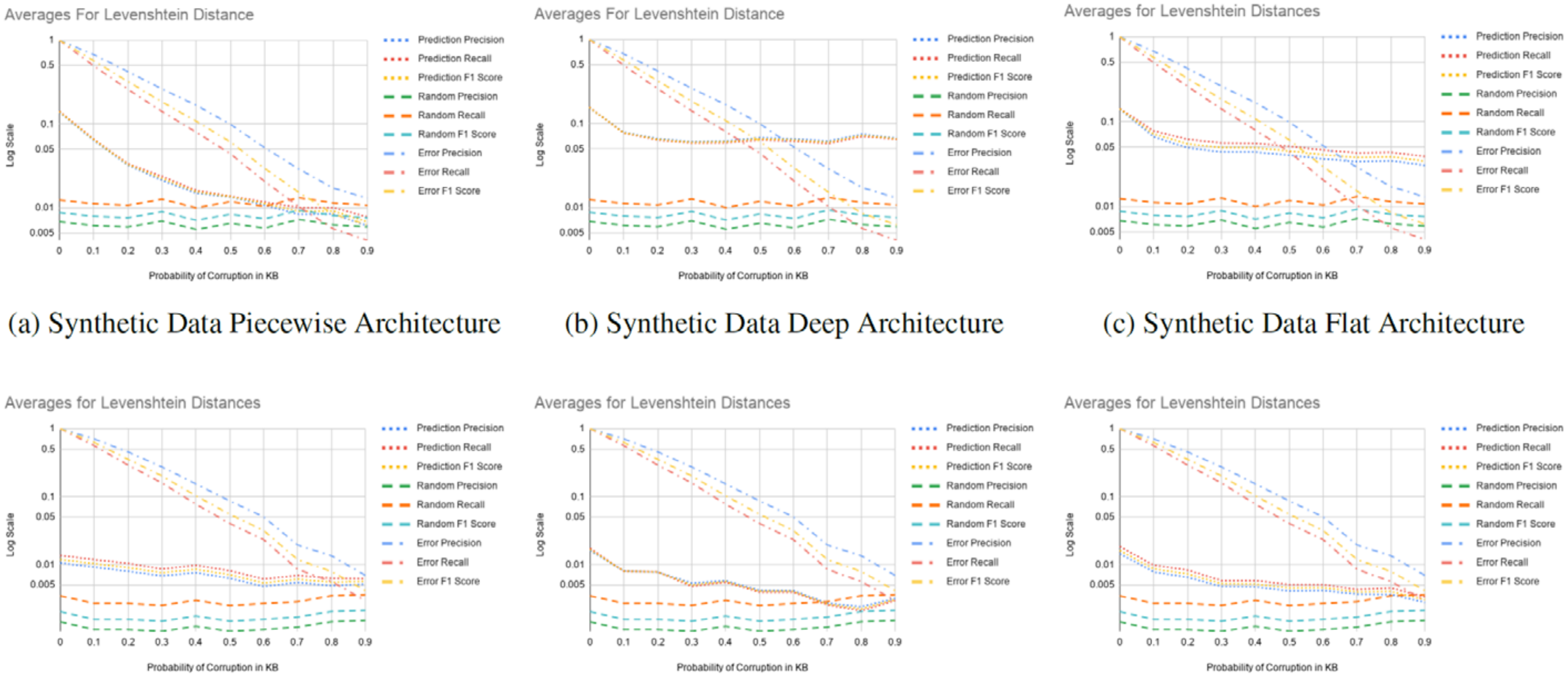
Results

Table 7: Average Precision Recall and F1-score For each Distance Evaluation



	Atomic Levenshtein Distance			Character Levenshtein Distance			Predicate Distance		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
	Synthetic Data								
Piecewise Prediction	0.138663	0.142208	0.140412	0.138663	0.142208	0.140412	0.138646	0.141923	0.140264
Deep Prediction	**0.154398**	**0.156056**	**0.155222**	**0.154398**	**0.156056**	**0.155222**	**0.154258**	**0.155736**	**0.154993**
Flat Prediction	0.140410	0.142976	0.141681	0.140410	0.142976	0.141681	0.140375	0.142687	0.141521
Random Prediction	0.010951	0.0200518	0.014166	0.006833	0.012401	0.008811	0.004352	0.007908	0.007908
	SNOMED Data								
Piecewise Prediction	0.010530	0.013554	0.011845	0.010530	0.013554	0.011845	0.010521	0.013554	0.011839
Deep Prediction	**0.015983**	0.0172811	**0.016595**	**0.015983**	0.017281	**0.016595**	**0.015614**	0.017281	**0.016396**
Flat Prediction	0.014414	**0.018300**	0.016112	0.0144140	**0.018300**	0.016112	0.013495	**0.018300**	0.015525
Random Prediction	0.002807	0.006803	0.003975	0.001433	0.003444	0.002023	0.001769	0.004281	0.002504

Noisy data



(a) Synthetic Data Piecewise Architecture

(b) Synthetic Data Deep Architecture

(c) Synthetic Data Flat Architecture

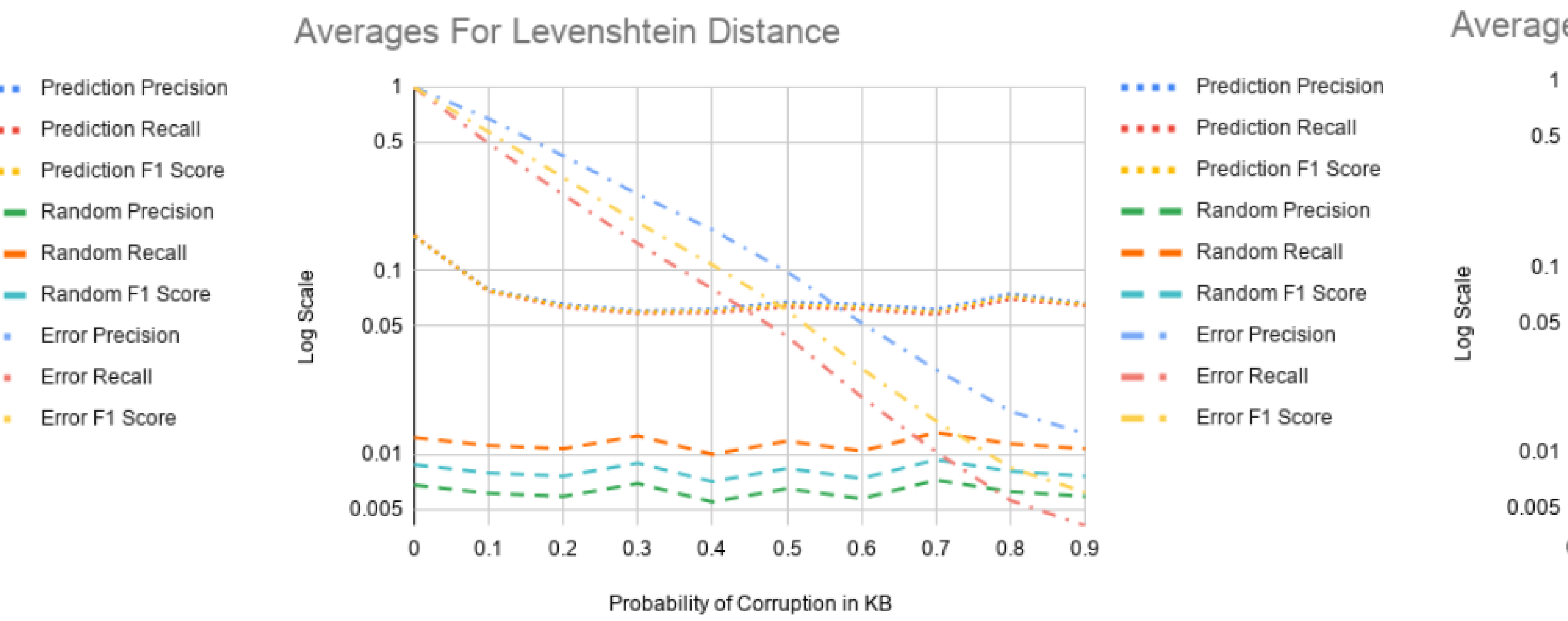
(d) SNOMED Data Piecewise Architecture

(e) SNOMED Data Deep Architecture

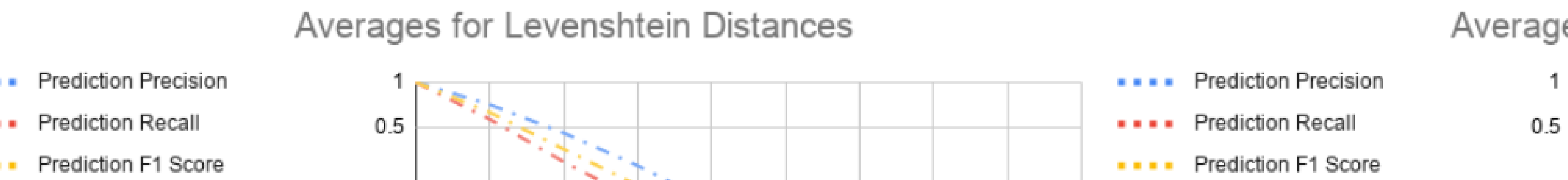
(f) SNOMED Data Flat Architecture

Figure 8: Character Levenshtein Distance Precision, Recall, and F1-score

Noisy data



hitecture (b) Synthetic Data Deep Architecture (c



The Deductive Capability of Logic Tensor Networks

Federico Bianchi, Pascal Hitzler, On the Capabilities of Logic Tensor Networks for Deductive Reasoning. In: Andreas Martin et al. (eds.), Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019) Stanford University, Palo Alto, California, USA, March 25-27, 2019, Stanford University, Palo Alto, California, USA, March 25-27, 2019. CEUR Workshop Proceedings 2350, CEUR-WS.org 2019.

Logic Tensor Networks

Based on Neural Tensor Networks.

Logic Tensor Networks are due to Serafini and Garcez (2016). They have been used for image analysis under background knowledge.

Their capabilities for deductive reasoning have not been sufficiently explored.

Underlying logic: First-order predicate, fuzzyfied.

Every language primitive becomes a vector/matrix/tensor.

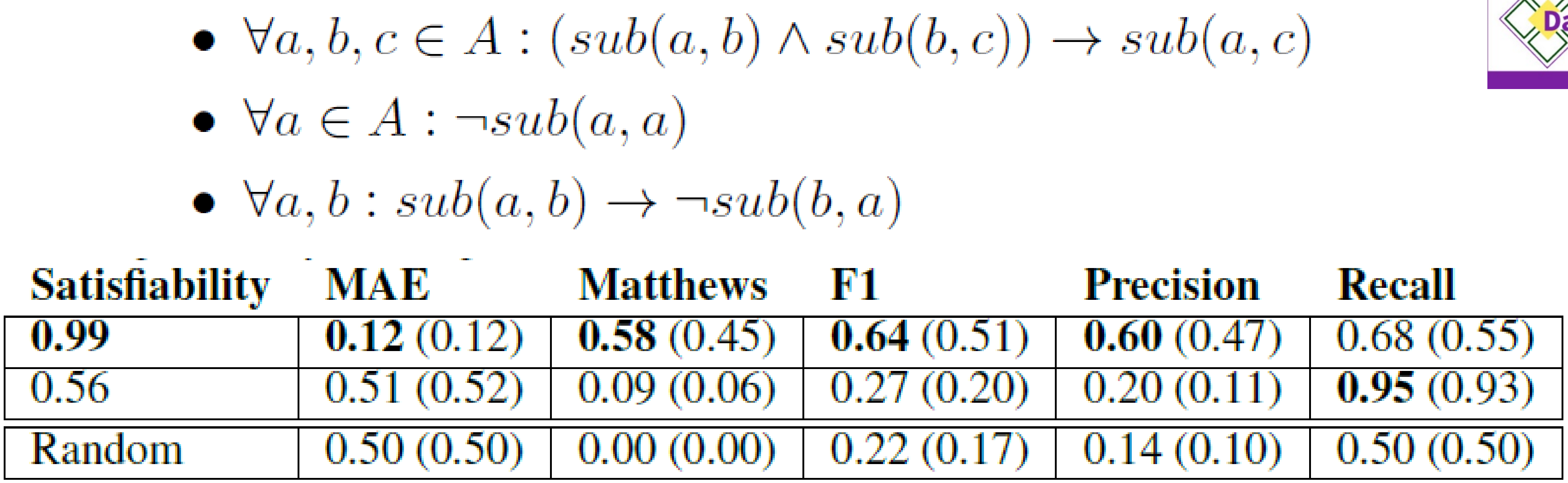
Terms/Atoms/Formulas are embedded as corresponding tensor/matrix/vector multiplications over the primitives.

Embeddings of primitives are learned s.t. the truth values of all formulas in the given theory are maximized.

A-priori Limitations

- Not clear how to adapt this such that you can transfer to unseen input theories.
- Scalability is an issue.
- While apparently designed for deductive reasoning, the inventors hardly report on this issue.

Transitive closure



- $\forall a, b, c \in A : (sub(a, b) \land sub(b, c)) \rightarrow sub(a, c)$
- $\forall a \in A : \neg sub(a, a)$
- $\forall a, b : sub(a, b) \rightarrow \neg sub(b, a)$

Satisfiability	MAE	Matthews	F1	Precision	Recall
0.99	**0.12** (0.12)	**0.58** (0.45)	**0.64** (0.51)	**0.60** (0.47)	0.68 (0.55)
0.56	0.51 (0.52)	0.09 (0.06)	0.27 (0.20)	0.20 (0.11)	**0.95** (0.93)
Random	0.50 (0.50)	0.00 (0.00)	0.22 (0.17)	0.14 (0.10)	0.50 (0.50)

parentheses: only newly entailed part of KB
MAE: mean absolute error;
Matthews: Matthews coefficient (for unbalanced classes)
top: top performing model, layer size and embeddings: 20, mean aggregator

Bottom: one of the worst performing models.

Multi-hop inferences difficult.

More take-aways from experiments

- **Error decreases with increasing satisfiability.**

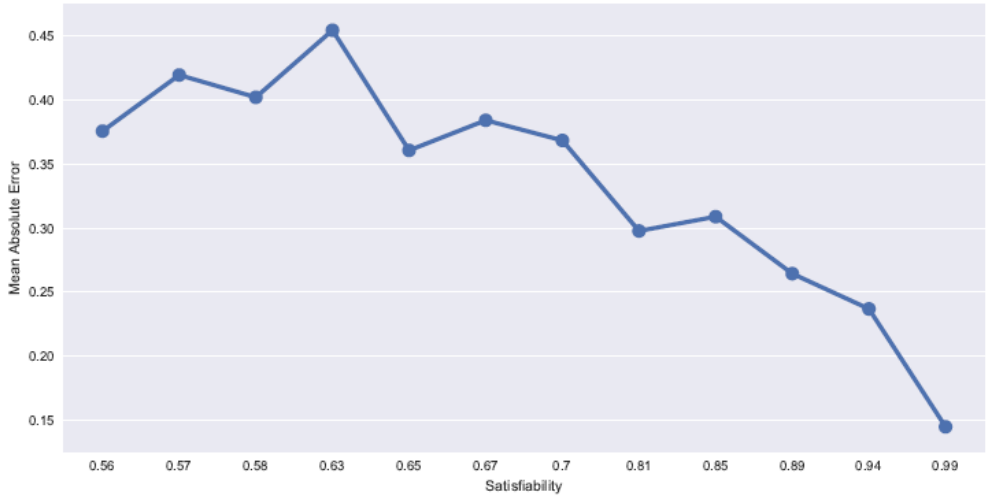
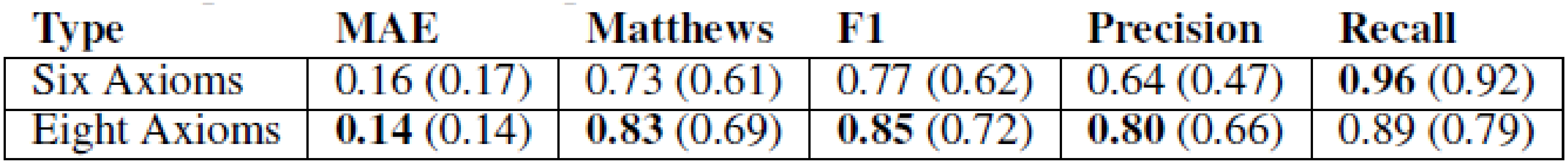


Figure 3: Average MAE for the ancestors tasks on rounded level of satisfiability. MAE decreases with the increase of satisfiability.

- **Adding redundant formulas to the input KB decreases error.**



Type	MAE	Matthews	F1	Precision	Recall
Six Axioms	0.16 (0.17)	0.73 (0.61)	0.77 (0.62)	0.64 (0.47)	**0.96** (0.92)
Eight Axioms	**0.14** (0.14)	**0.83** (0.69)	**0.85** (0.72)	**0.80** (0.66)	0.89 (0.79)

More take-aways from experiments

- **Higher arity of predicates significantly increases learning time.**

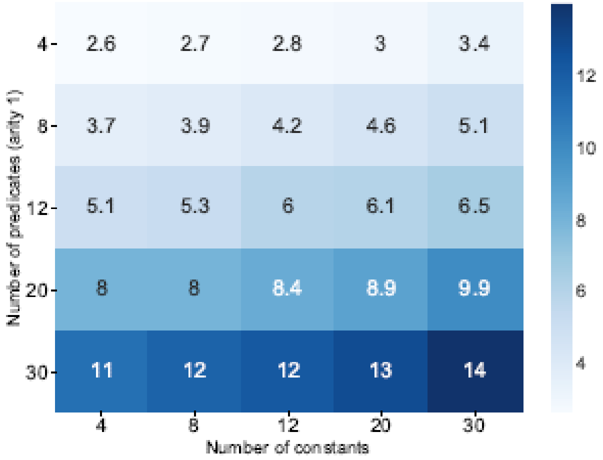


Figure 5: Computational times in seconds for predicates of arity one and constants

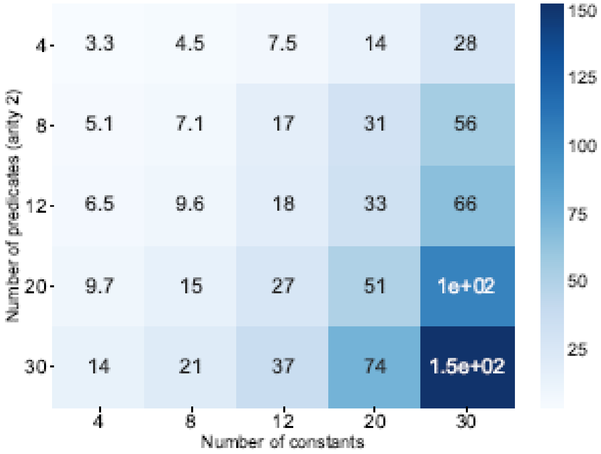


Figure 6: Computational times in seconds for predicates of arity two and constants

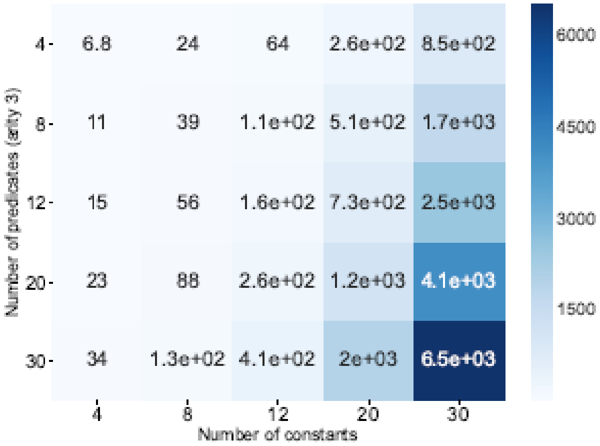


Figure 7: Computational times in seconds for predicates of arity three and constants

More take-aways from experiments

- Model seems to often end up in local minima. This may be addressable using known approaches.
- LTNs seem to predict many false positives, while they are better regarding true negatives. This may be just because of the test knowledge bases we used, but needs to be looked at.
- Overfitting is a problem, but it doesn't seem straightforward to address this for LTNs. [e.g. cross-validation may need completeness information, which may bias the network]
- Increasing layers and embedding size makes optimizing parameters much more difficult.

- Hence, there's a path for more investigations, we're only starting to understand this.

Conclusions

Conclusions

- Bridging the neural-symbolic gap is still a major quest.

- But there are tons of opportunities.

Thanks!

References

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, Pascal Hitzler, Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners. Applied Intelligence, 2021, to appear.

Barbara Hammer and Pascal Hitzler (eds), Perspectives on Neural-Symbolic Integration. Springer, 2007

Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, Gerson Zaverucha, Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. https://arxiv.org/abs/1711.03902 (2017)

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, Pascal Hitzler

Neuro-Symbolic Artificial Integration: Current Trends

https://arxiv.org/abs/2105.05330

References

Federico Bianchi, Pascal Hitzler, On the Capabilities of Logic Tensor Networks for Deductive Reasoning. In: Andreas Martin, Knut Hinkelmann, Aurona Gerber, Doug Lenat, Frank van Harmelen, Peter Clark (eds.), Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019) Stanford University, Palo Alto, California, USA, March 25-27, 2019, Stanford University, Palo Alto, California, USA, March 25-27, 2019. CEUR Workshop Proceedings 2350, CEUR-WS.org 2019.

Aaron Eberhart, Monireh Ebrahimi, Lu Zhou, Cogan Shimizu, Pascal Hitzler, Completion Reasoning Emulation for the Description Logic EL+. In: Andreas Martin, Knut Hinkelmann, Hans-Georg Fill, Aurona Gerber, Doug Lenat, Reinhard Stolle, Frank van Harmelen (eds.), Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE 2020, Palo Alto, CA, USA, March 23-25, 2020, Volume I.

References

Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie, Aaron Eberhart, Derek Doran, Hyeongsik Kim, Pascal Hitzler, Neuro-Symbolic Deductive Reasoning for Cross-Knowledge Graph Entailment. In: Proc. AAAI-MAKE 2021.

Bassem Makni, James Hendler, Deep learning for noise-tolerant RDFS reasoning. Semantic Web 10(5): 823-862 (2019)

Pascal Hitzler, Frank van Harmelen, A reasonable Semantic Web. Semantic Web 1 (1-2), 39-44, 2010.

Pascal Hitzler, Federico Bianchi, Monireh Ebrahimi, Md Kamruzzaman Sarker, Neural-Symbolic Integration and the Semantic Web. Semantic Web 11 (1), 2020, 3-11.

References

Federico Bianchi, Matteo Palmonari, Pascal Hitzler, Luciano Serafini, Complementing Logical Reasoning with Sub-symbolic Commonsense. In: Paul Fodor, Marco Montali, Diego Calvanese, Dumitru Roman, Rules and Reasoning - Third International Joint Conference, RuleML+RR 2019, Bolzano, Italy, September 16-19, 2019, Proceedings. Lecture Notes in Computer Science 11784, Springer 2019, pp. 161-170.

Sebastian Bader, Pascal Hitzler, Dimensions of neural-symbolic integration – a structured survey. In: S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb and J. Woods (eds). We Will Show Them: Essays in Honour of Dov Gabbay, Volume 1. International Federation for Computational Logic, College Publications, 2005, pp. 167-194.

Thanks!