Neurosymbolic AI – some recent results related to knowledge graphs

Pascal Hitzler
Data Semantics Laboratory (DaSe Lab)
Kansas State University

http://www.daselab.org

Explainable AI using deductive reasoning over background knowledge

Problem setting:
why we need strong explainabilty
for deep learning systems

The black box problem

There have been enormous strides recently in methods and applications of Deep learning.

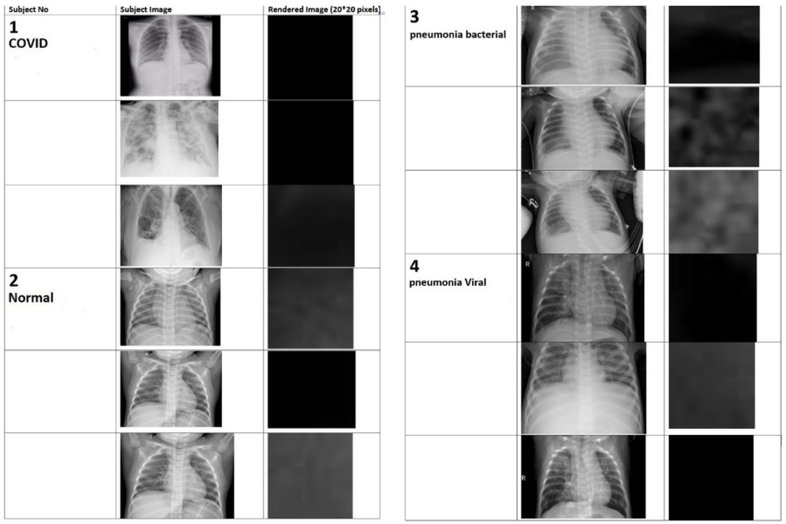
However

- Deep Learning system are black boxes
- Evaluation is only done statistically

This is insufficient for many application areas, and problematic for most.

The black box problem

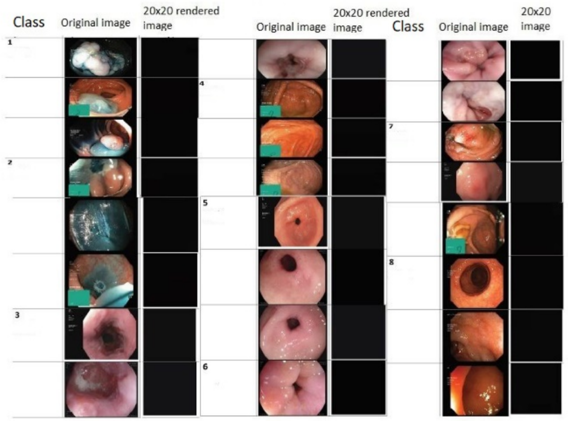
COVID-19 detection



<u>CNN classification accuracy</u>:

Original images – 67%
Blank background images – 62%
Mere chance accuracy – 25%

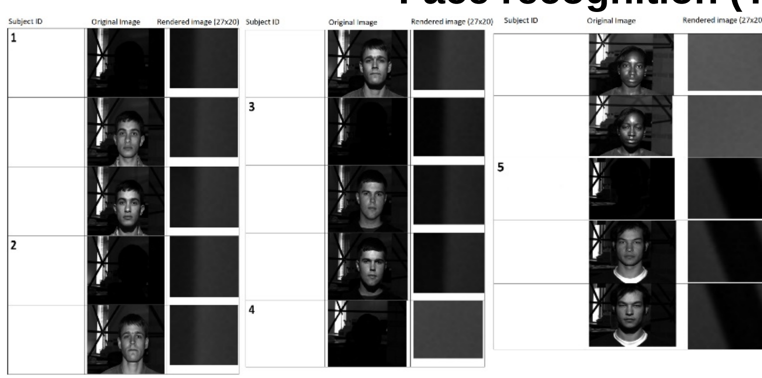
Gastrointestinal disease detection (Kvasir dataset)



<u>CNN classification accuracy</u>:

Original images – 77%
Blank background images – 41%
Mere chance accuracy – 12%

Face recognition (Yale B)



<u>CNN classification accuracy</u>:

Original images – 99%
Blank background images – 87%
Mere chance accuracy – 4%

Dhar, S., Shamir, L., 2021, *Visual Informatics*, 5(3), 92-101 – thanks to Lior Shamir for the slides input

Galaxy image annotation

Classification to spiral galaxies and elliptical galaxies

When the test set and training set are from the same part of the sky, the CNN shows a different Universe than when the training and test images come from different parts of the sky.

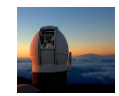
SDSS 

Training set and test set from the same part of the sky

	Elliptical	Spiral
Elliptical	2891	109
Spiral	85	2915

Training set and test set from different parts of the sky

	Elliptical	Spiral
Elliptical	2704	296
Spiral	31	2969

Pan-STARRS 

Training set and test set from the same part of the sky

	Elliptical	Spiral
Elliptical	7850	150
Spiral	756	7244

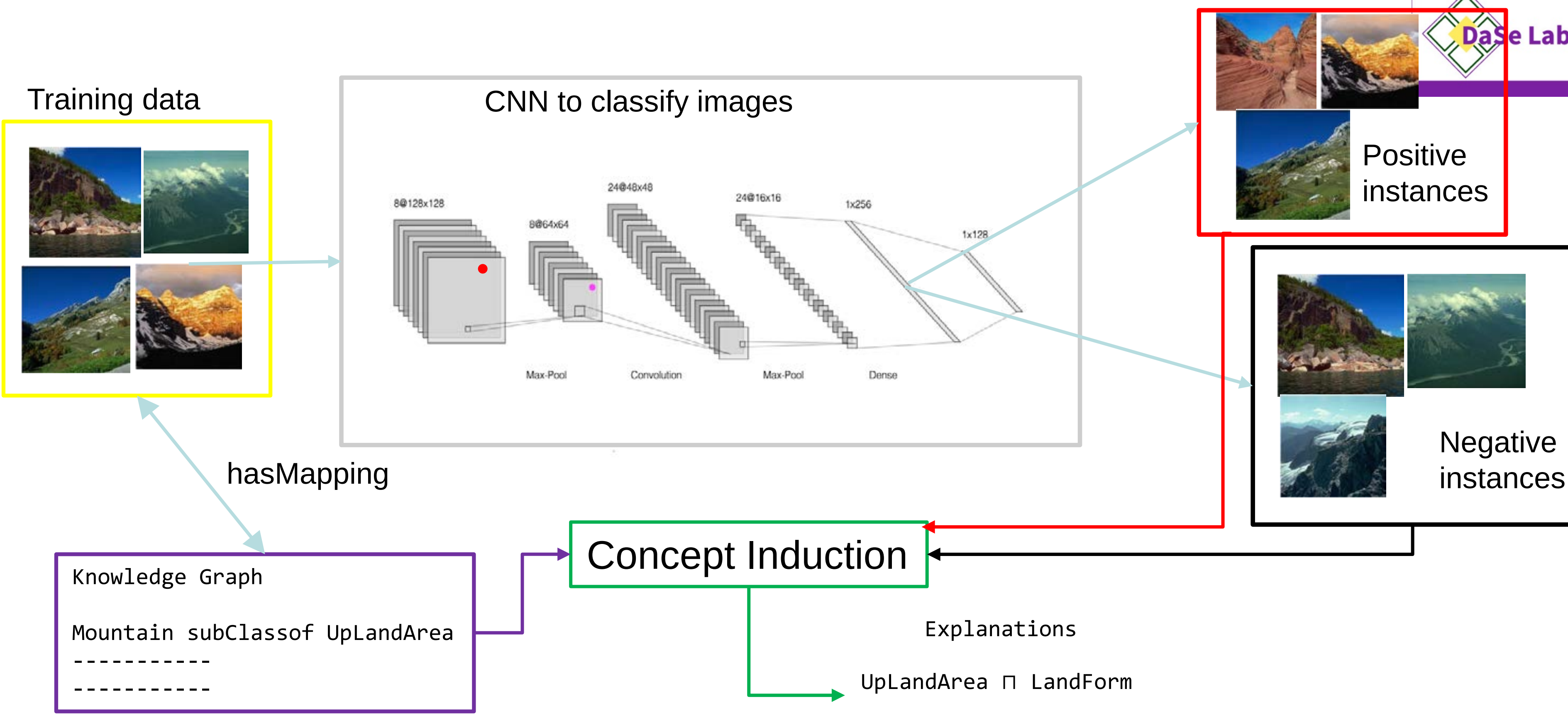
Training set and test set from the same part of the sky

	Elliptical	Spiral
Elliptical	7699	301
Spiral	450	7550

Dhar, S., Shamir, L., 2022, *Astronomy and Computing*, 38, 100545

Approach: Concept Induction for Hidden Layer Analysis

Idea



New results based on: Abhilekha Dalal, Md Kamruzzaman Sarker, Adrita Barua, Eugene Vasserman, Pascal Hitzler https://arxiv.org/abs/2308.03999.

Concept Induction

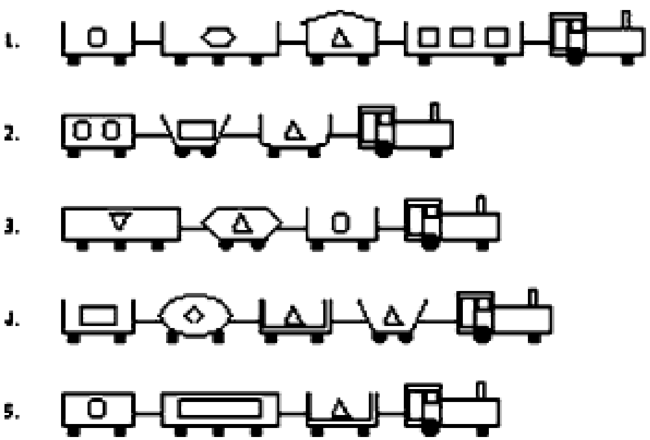
Some slides adapted from Joshua Schwartz, with permission.

Concept Induction

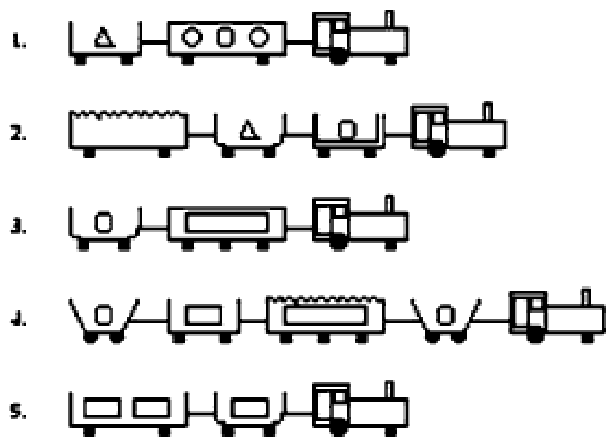
DaSe Lab

Approach similar to inductive logic programming, but using Description Logics (the logic underlying OWL).

Positive examples:



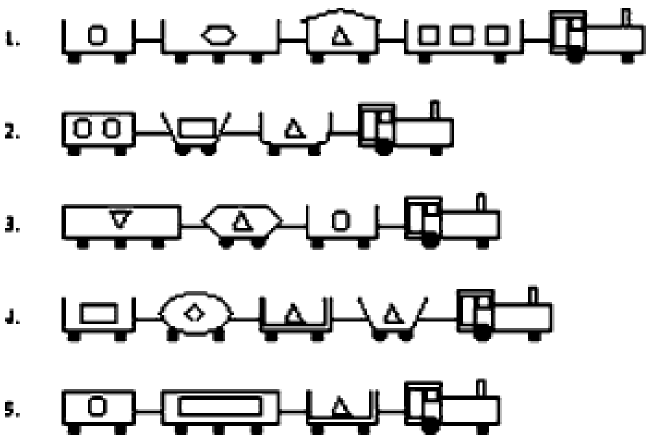
negative examples:



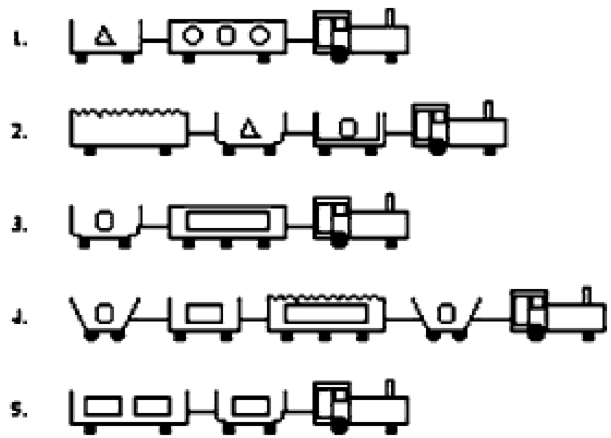
Task: find a class description (logical formula) which separates positive and negative examples.

DL-Learner

Positive examples:



negative examples:



DL-Learner result: $\exists \texttt{hasCar}.(\texttt{Closed} \sqcap \texttt{Short})$

In FOL: $\{x \mid \exists y(\mathrm{hasCar}(x,y) \wedge \mathrm{Closed}(y) \wedge \mathrm{Short}(y))\}$

Theory and system: [Lehmann & Hitzler 2010], DL-Learner

```
car(car_11).  car(car_12).  car(car_13).
car(car_14).
car(car_21).  car(car_22).  car(car_23).
car(car_31).  car(car_32).  car(car_33).
car(car_41).  car(car_42).  car(car_43).
car(car_44).
car(car_51).  car(car_52).  car(car_53).
car(car_61).  car(car_62).
car(car_71).  car(car_72).  car(car_73).
car(car_81).  car(car_82).
car(car_91).  car(car_92).  car(car_93).
car(car_94).
car(car_101).  car(car_102).


train(east1).  train(east2).  train(east3).
train(east4).  train(east5).
train(west6).  train(west7).  train(west8).
train(west9).  train(west10).
```

```
// eastbound train 1

has_car(east1,car_11).
has_car(east1,car_12).
has_car(east1,car_13).
has_car(east1,car_14).

short(car_12).
closed(car_12).
long(car_11).
long(car_13).
short(car_14).
open_car(car_11).
open_car(car_13).
open_car(car_14).
shape(car_11,rectangle).
shape(car_12,rectangle).
shape(car_13,rectangle).
shape(car_14,rectangle).
load(car_11,rectangle).
load_count(car_11,three).
load(car_12,triangle).
load_count(car_12,one).
load(car_13,hexagon).
load_count(car_13,one).
load(car_14,circle).
load(car_14,one).
wheels(car_11,two).
wheels(car_12,two).
wheels(car_13,three).
wheels(car_14,two).
```

Somewhat more formally…

generating complex description logic class expressions S from a given description logic knowledge base (or ontology) $\mathcal{O}$ and sets P and N of instances, understood as positive and negative examples, such that $\mathcal{O} \models S(a)$ for all $a \in P$, and $\mathcal{O} \not\models S(b)$ for all $b \in N$

Algorithmically – Refinement Operator

```
Start with simple formula E (e.g., ⊤ )
Loop: Expand E minimally in all possible ways to
            E1,...,En
      Check accuracy for E1 through En regarding P and N
      Replace E by highest-scoring Ei
Exit loop if perfect solution found or other stopping
criteria met

Return E
```

In reality, a list of formulas is returned, ranked by accuracy.

Accuracy can be f-measure, precision, recall, etc.

Checking accuracy needs deductive reasoning, i.e., is expensive.

[Lehmann & Hitzler, Machine Learning, 2010], DL-Learner system

Algorithmically – heuristic

- Restrict allowed syntax expansions (e.g., conjunctions only)
- Restrict complexity of logic in background knowledge (e.g., class hierarchy only)

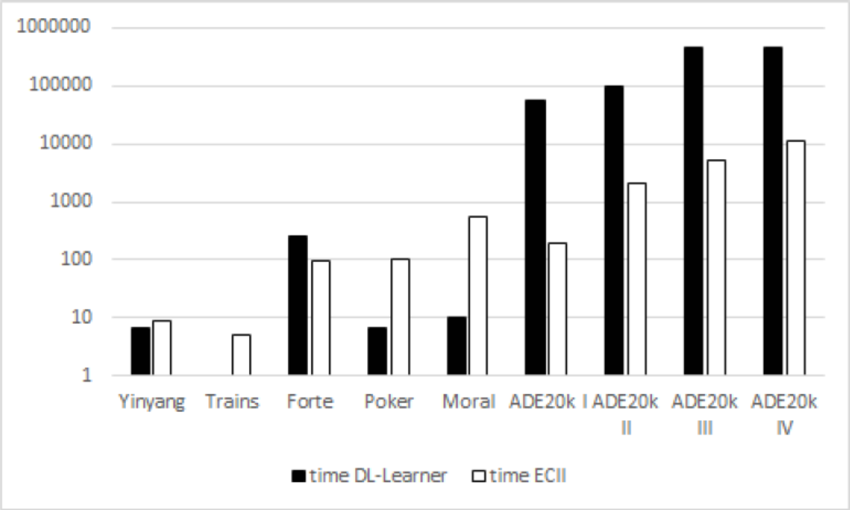


Figure 1: Runtime comparison between DL-Learner and ECII. The vertical scale is logarithmic in hundredths of seconds, and note that DL-Learner runtime has been capped at 4,500 seconds for ADE20k III and IV. For ADE20k I it was capped at each run at 600 seconds.

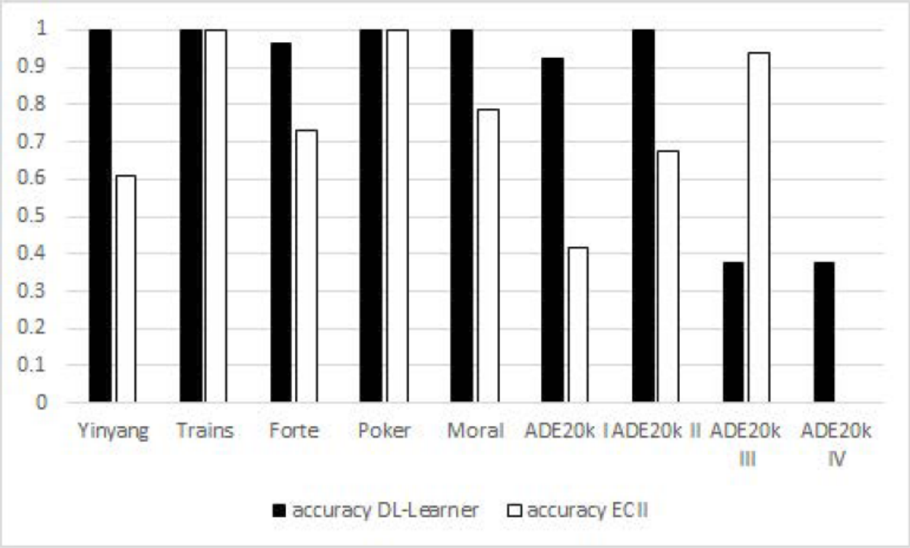


Figure 2: Accuracy (α_3) comparison between DL-Learner and ECII. For ADE20k IV it was not possible to compute an accuracy score within 3 hours for ECII as the input ontology was too large.

[Sarker & Hitzler, AAAI, 2019]: ECII system

Background Knowledge

Background knowledge

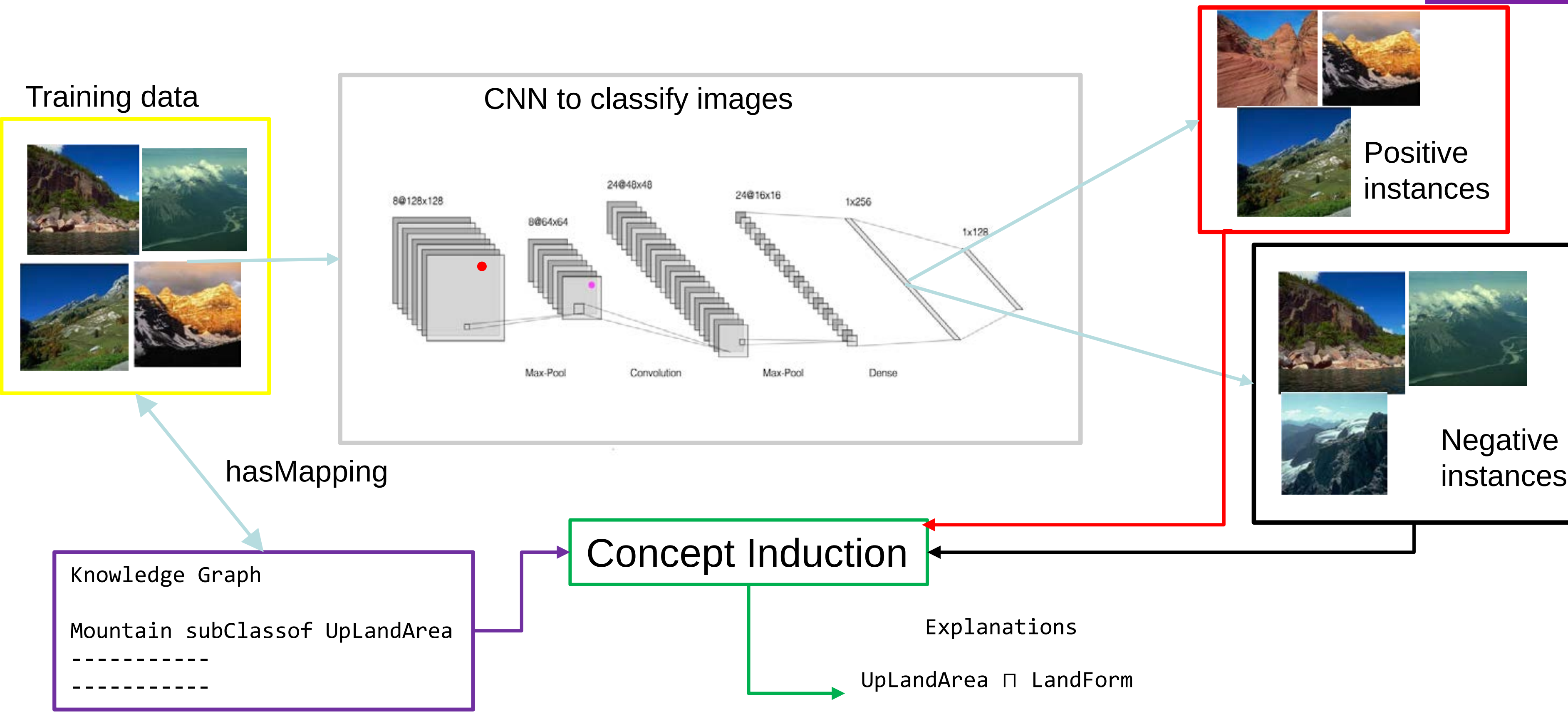
- Based on Wikipedia category hierarchy
  - which is not a hierarchy because it has loops, caused by crowd-sourcing

- Heuristically curated by removing loops
- Resulting class hierarchy has approx. 2M concepts
- Broad coverage (all things in Wikipedia)
- Can easily refer to it from instances by mapping to Wikipedia pages and looking up the page categories.

[Sarker et al., KGSWC2020]

Concrete Setting

Idea


Training data
CNN to classify images
8@128x128
8@64x64
24@48x48
24@16x16
1x256
1x128
Max-Pool
Convolution
Max-Pool
Dense
Positive instances
Negative instances
hasMapping
Knowledge Graph
Mountain subClassof UpLandArea

Concept Induction
Explanations
UpLandArea ⊓ LandForm

Scenario

- Scene recognition (from images)
- MIT ADE20k dataset
  http://groups.csail.mit.edu/vision/datasets/ADE20K/
- 10 overlapping scenes selected for our study
- Resnet50V2 trained (best of those we tried)
  - Training accuracy 87.6%
  - Validation accuracy 86.5%

Images annotations

The ADE20k images come with annotations of objects in the picture:

```
001 # 0 # 0 # sky # sky # ""
002 # 0 # 0 # road, route # road # ""
005 # 0 # 0 # sidewalk, pavement # sidewalk # ""
006 # 0 # 0 # building, edifice # building # ""
007 # 0 # 0 # truck, motortruck # truck # ""
008 # 0 # 0 # hovel, hut, hutch, shack, shanty # hut # ""
009 # 0 # 0 # pallet # pallet # ""
011 # 0 # 0 # box # boxes # ""
001 # 1 # 0 # door # door # ""
002 # 1 # 0 # window # window # ""
009 # 1 # 0 # wheel # wheel # ""
```

We ignore everything but the types of object on each image.

Mapping to Background Knowledge

- **String matching (Levenshtein with edit distance 0) from object types to Wikipedia categories**

contains(img1,road1)
contains(img1, window1)
contains(img1, door1)
contains(img1, wheel1)
contains(img1, sidewalk1)
contains(img1, truck1)
contains(img1, box1)
contains(img1, building1)

Label Hypothesis
Generation and Confirmation

Trained CNN

- Scene classification on ADE20k
- Resnet50V2; 64 hidden nodes in the dense layer

	precision	recall	f1-score	support
bathroom	0.90	0.78	0.84	134
bedroom	0.89	0.88	0.88	277
building_facade	0.68	0.60	0.64	45
conference_room	0.77	0.91	0.83	33
dining_room	0.75	0.84	0.79	82
highway	0.96	0.88	0.92	59
kitchen	0.84	0.87	0.86	130
living_room	0.76	0.74	0.75	139
skyscraper	0.90	0.88	0.89	64
street	0.92	0.96	0.94	407
accuracy			0.87	1370
macro avg	0.84	0.83	0.83	1370
weighted avg	0.87	0.87	0.87	1370

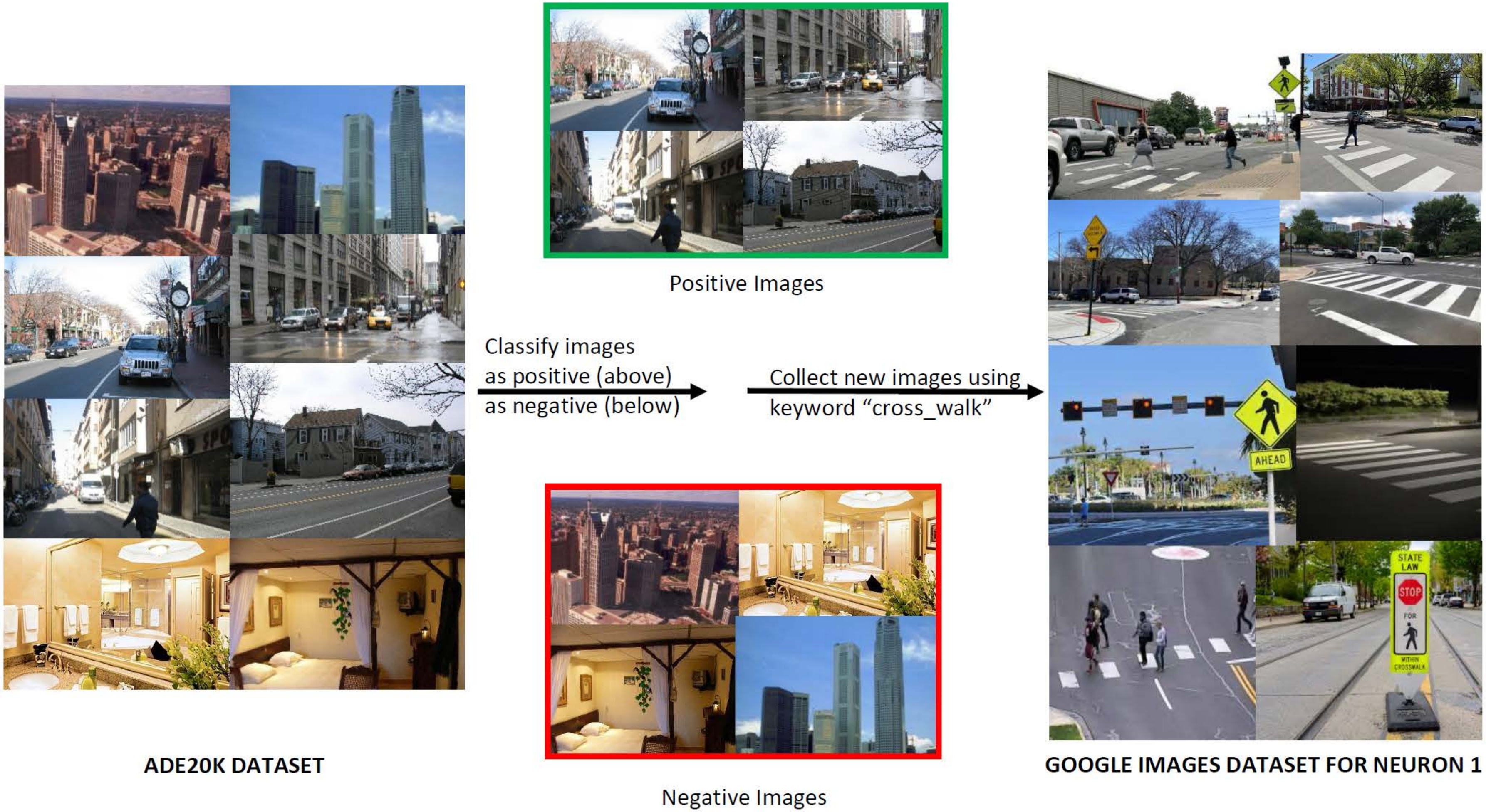


Figure 1: Example of images that were used for generating and confirming the label hypothesis for neuron 1

workflow: label hypothesis generation and
confirmation of label hypothesis with new images from Google images

Neuron #	Obtained Label(s)	Images	Coverage	Target %	Non-Target %
0	**building**	**164**	**0.997**	**89.024**	**72.328**
1	**cross_walk**	**186**	**0.994**	**88.710**	**28.923**
3	**night_table**	**157**	**0.987**	**90.446**	**56.714**
6	dishcloth, toaster	106	0.999	16.038	39.078
7	toothbrush, Pipage	112	0.991	75.893	59.436
8	**shower_stall, cistern**	**136**	**0.995**	**100.000**	**53.186**
11	river_water	157	0.995	31.847	22.309
12	baseboard, dish_rag	108	0.993	75.926	48.248
14	rocking_horse, rocker	86	0.985	54.651	47.816
16	**mountain, bushes**	**108**	**0.995**	**87.037**	**24.969**
17	stem	133	0.993	30.827	31.800
18	**slope**	**139**	**0.983**	**92.086**	**69.919**
19	**wardrobe, air_conditioning**	**110**	**0.999**	**89.091**	**65.034**
20	fire_hydrant	158	0.990	5.696	13.233
22	**skyscraper**	**156**	**0.992**	**99.359**	**54.893**
23	fire_escape	162	0.996	61.111	18.311
25	spatula, nuts	126	0.999	2.381	0.883
26	skyscraper, river	112	0.995	77.679	35.489
27	manhole, left_arm	85	0.996	35.294	26.640
28	flooring, fluorescent_tube	115	1.000	38.261	33.198
29	**lid, soap_dispenser**	**131**	**0.998**	**99.237**	**78.571**
30	**teapot, saucepan**	**108**	**0.998**	**81.481**	**47.984**
31	fire_escape	162	0.961	77.160	63.147
33	tanklid, slipper	81	0.987	41.975	30.214
34	left_foot, mouth	110	0.994	20.909	49.216

Neuron #	Obtained Label(s)	Images	Coverage	Target %	Non-Target %
35	utensils_canister, body	111	0.999	7.207	11.223
36	**tap, crapper**	**92**	**0.997**	**89.130**	**70.606**
37	cistern, doorcase	101	0.999	21.782	24.147
38	letter_box, go_cart	125	0.999	28.000	31.314
39	side_rail	148	0.980	35.811	34.687
40	sculpture, side_rail	119	0.995	25.210	21.224
41	**open_fireplace, coffee_table**	**122**	**0.992**	**88.525**	**16.381**
42	pillar, stretcher	117	0.998	52.137	42.169
43	**central_reservation**	**157**	**0.986**	**95.541**	**84.973**
44	saucepan, dishrack	120	0.997	69.167	36.157
46	Casserole	157	0.999	45.223	36.394
48	**road**	**167**	**0.984**	**100.000**	**73.932**
49	**footboard, chain**	**126**	**0.982**	**88.889**	**66.702**
50	night_table	157	0.972	65.605	62.735
51	**road, car**	**84**	**0.999**	**98.810**	**48.571**
53	pylon, posters	104	0.985	11.538	17.332
54	**skyscraper**	**156**	**0.987**	**98.718**	**70.432**
56	**flusher, soap_dish**	**212**	**0.997**	**90.094**	**63.552**
57	**shower_stall, screen_door**	**133**	**0.999**	**98.496**	**31.747**
58	plank, casserole	80	0.998	3.750	3.925
59	manhole, left_arm	85	0.994	35.294	21.589
60	paper_towels, jar	87	0.999	0.000	1.246
61	ornament, saucepan	102	0.995	43.137	17.274
62	sideboard	100	0.991	21.000	29.734
63	**edifice, skyscraper**	**178**	**0.999**	**92.135**	**48.761**

Evaluation

Approach

- Each row of the table is a hypothesis, e.g. "neuron 1 activates more strongly on cross_walk images (retrieved from Google images using keyword "cross_walk") than on other images."

- Null hypothesis: There is no difference in activations.

- There is no reason to assume a normal distribution,
- hence using Mann-Whitney U test for assessment.

Evaluation results

Neuron #	Label(s)	Images	# Activations (%)		Mean		Median		z-score	p-value
			targ	non-t	targ	non-t	targ	non-t		
0	building	42	80.95	73.40	2.08	1.81	2.00	1.50	-1.28	0.0995
1	cross_walk	47	91.49	28.94	4.17	0.67	4.13	0.00	-8.92	<.00001
3	night_table	40	100.00	55.71	2.52	1.05	2.50	0.35	-6.84	<.00001
8	shower_stall, cistern	35	100.00	54.40	5.26	1.35	5.34	0.32	-8.30	<.00001
16	mountain, bushes	27	100.00	25.42	2.33	0.67	2.17	0.00	-6.72	<.00001
18	slope	35	91.43	68.85	1.59	1.37	1.44	1.00	-2.03	0.0209
19	wardrobe, air_conditioning	28	89.29	65.81	2.30	1.28	2.30	0.84	-4.00	<.00001
22	skyscraper	39	97.44	56.16	3.97	1.28	4.42	0.33	-7.74	<.00001
29	lid, soap_dispenser	33	100.00	80.47	4.38	2.14	4.15	1.74	-5.92	<.00001
30	teapot, saucepan	27	85.19	49.93	2.52	1.05	2.23	0.00	-4.28	<.00001
36	tap, crapper	23	91.30	70.78	3.24	1.75	2.82	1.29	-3.59	<.00001
41	open_fireplace, coffee_table	31	80.65	15.11	2.03	0.14	2.12	0.00	-7.15	<.00001
43	central_reservation	40	97.50	85.42	7.43	3.71	8.08	3.60	-5.94	<.00001
48	road	42	100.00	74.46	6.15	2.68	6.65	2.30	-7.78	<.00001
49	footboard, chain	32	84.38	66.41	2.63	1.67	2.30	1.17	-2.58	0.0049
51	road, car	21	100.00	47.65	5.32	1.52	5.62	0.00	-6.03	<.00001
54	skyscraper	39	100.00	71.78	4.14	1.61	4.08	1.12	-7.60	<.00001
56	flusher, soap_dish	53	92.45	64.29	3.47	1.48	3.08	0.86	-6.47	<.00001
57	shower_stall, screen_door	34	97.06	32.31	2.60	0.61	2.53	0.00	-7.55	<.00001
63	edifice, skyscraper	45	88.89	48.38	2.41	0.83	2.36	0.00	-6.73	<.00001

Table 3: Evaluation details as discussed in Section 4. Images: number of images used for evaluation. # Activations: (targ(et)): Percentage of target images activating the neuron (i.e., activation at least 80% of this neuron's activation maximum); (non-t): Same for all other images used in the evaluation. Mean/Median (targ(et)/non-t(arget)): mean/median activation value for target and non-target images.

Discussion

-target images not activating neuron 1

Non-target images activating neuron 1

Figure 2: Examples of some Google images used: target images ("cross_walk") that did not activate the neuron; non-target images from labels like "central_reservation," "road and car," and "fire_hydrant" that activated the neuron.

Note: "bushes, bush" is the third-highest concept induction output (coverage 0.993; 48.052% of target images activating the neuron)

Going forward

We would really want to have labels with high target activation and low non-target activation.

- **make use of more concept induction results**
- **better background knowledge**
- **optimize parameters (like thresholds)**
- **investigate neuron ensembles ( )**

Label(s)	Images	# Activations (%)	
		targ	non-t
building	42	80.95	73.40
cross_walk	47	91.49	28.94
night_table	40	100.00	55.71
shower_stall, cistern	35	100.00	54.40
mountain, bushes	27	100.00	25.42
slope	35	91.43	68.85
wardrobe, air_conditioning	28	89.29	65.81
skyscraper	39	97.44	56.16
lid, soap_dispenser	33	100.00	80.47
teapot, saucepan	27	85.19	49.93
tap, crapper	23	91.30	70.78
open_fireplace, coffee_table	31	80.65	15.11
central_reservation	40	97.50	85.42
road	42	100.00	74.46
footboard, chain	32	84.38	66.41
road, car	21	100.00	47.65
skyscraper	39	100.00	71.78
flusher, soap_dish	53	92.45	64.29
shower_stall, screen_door	34	97.06	32.31
edifice, skyscraper	45	88.89	48.38

Concluding

- It works!
- But it needs to be refined.

Are Concept Induction Explanations Meaningful to Humans?

Cara Widmer, Md Kamruzzaman Sarker, Srikanth Nadella, Joshua Fiechter, Ion Juvina, Brandon Minnery, Pascal Hitzler, Joshua Schwartz, Michael Raymer, Towards Human-Compatible XAI: Explaining Data Differentials with Concept Induction over Background Knowledge
https://arxiv.org/abs/2209.13710

Are the results human-compatible? Part I

- **Hypothesis:**
  - **ECII explanations are better than semi-random explanations, but worse than human-generated explanations.**
- **Experimental setting as before.**
- **300 Amazon Mechanical Turk participants**
- **Seven concepts taken from top ECII results.**
- **45 image set pairs, each set corresponding to a category.**

Which of these better represents what the images in group A have that the images in group B do not?

Bake, Bakery, Bread, Indoor, Product, Store, Woman

Basket, Bread, Cake, Ceiling, Floor, Person, Wall

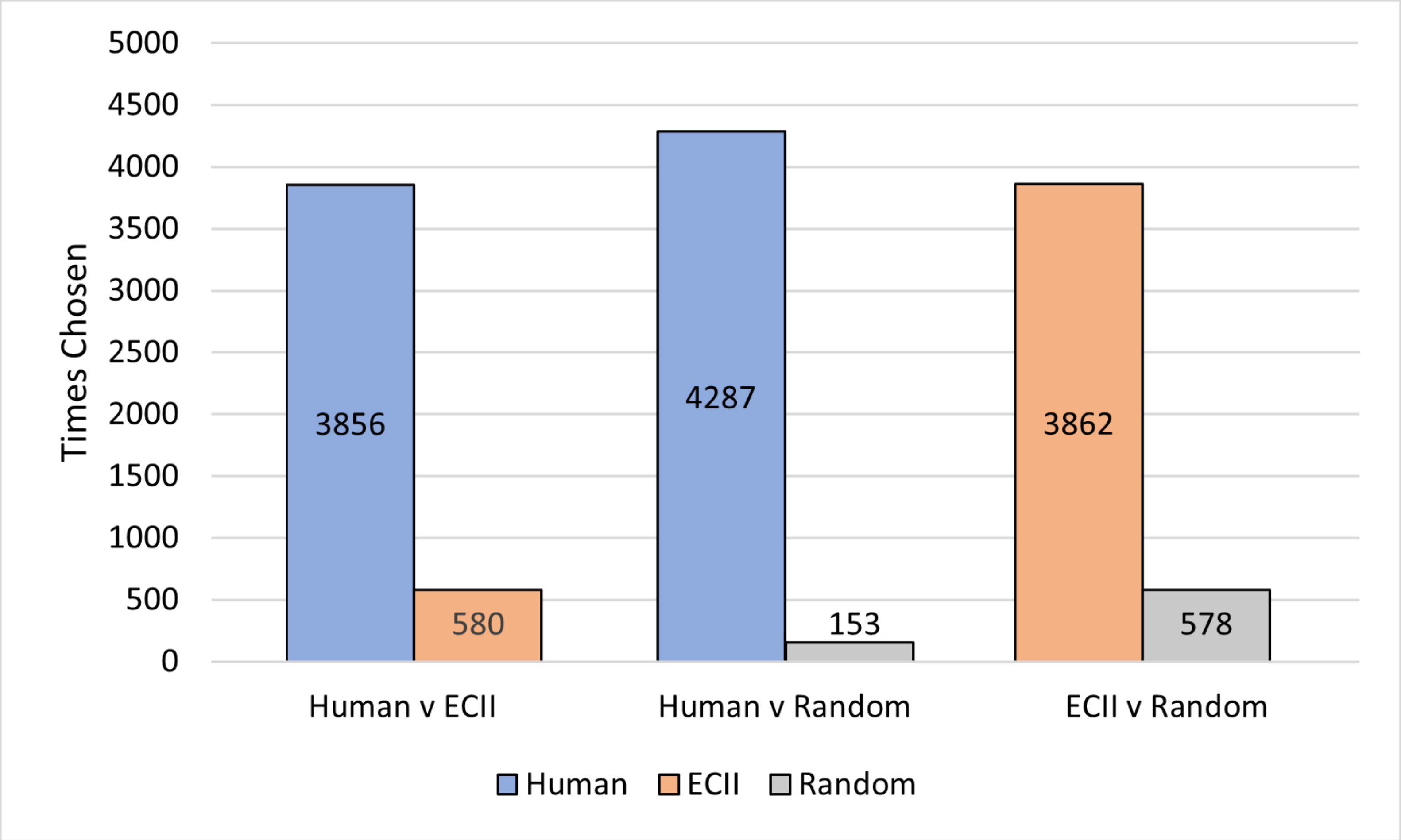
Are the results human-compatible? Part I

Which of these better represents what the images in group A have that the images in group B do not?

Bake, Bakery, Bread, Indoor, Product, Store, Woman	Basket, Bread, Cake, Ceiling, Floor, Person, Wall

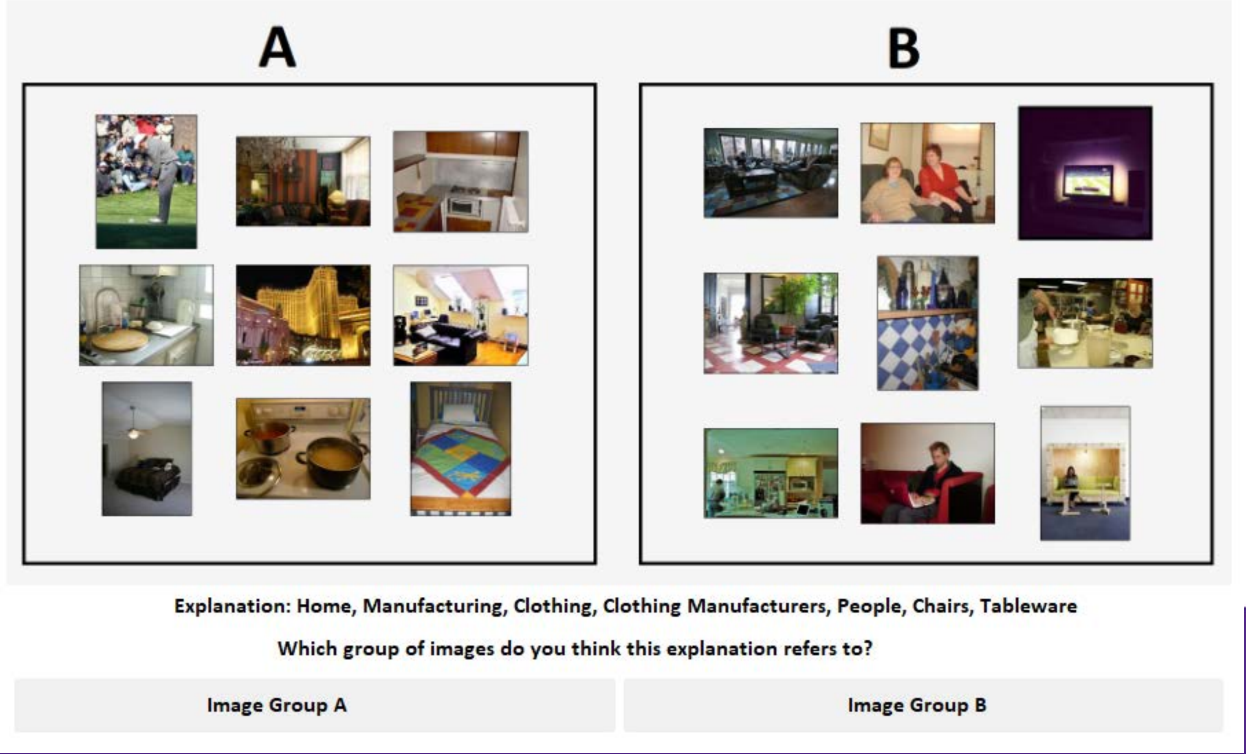
Are the results human-compatible? Part I

87-13 **97-3** **87-12**



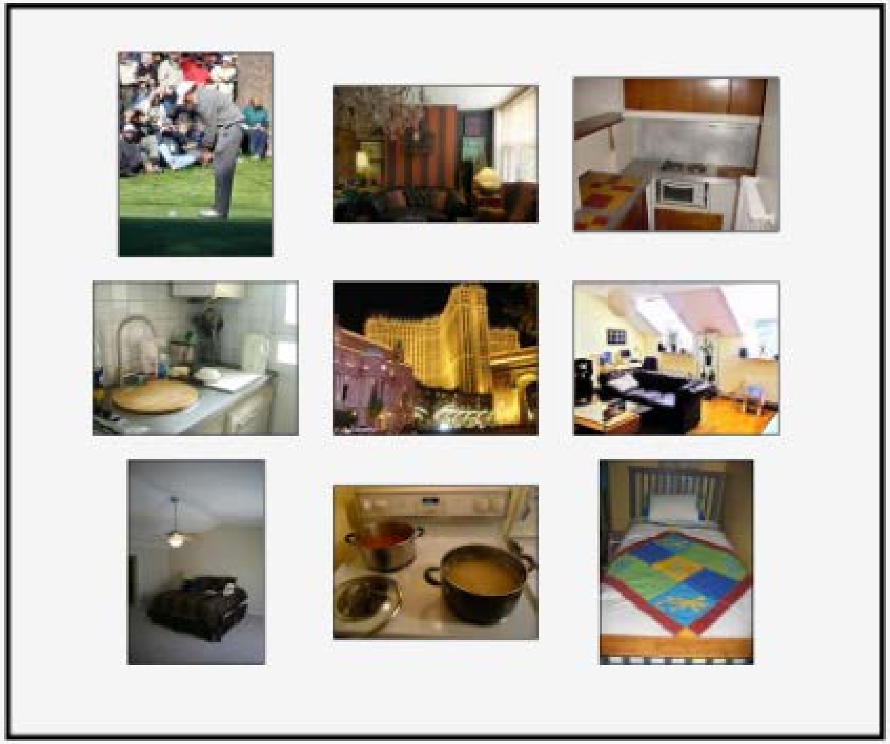
Are the results human-compatible? Part II

- **Hypothesis:**
  - **ECII explanations matched to correct images better than chance, but not as frequently as human generated explanations**
- **Experimental setting as before.**
- **100 Amazon Mechanical Turk participants**
- **16 image sets, from ML decision errors (logistic regression classifier)**



Are the results human-compatible? Part II

A



B

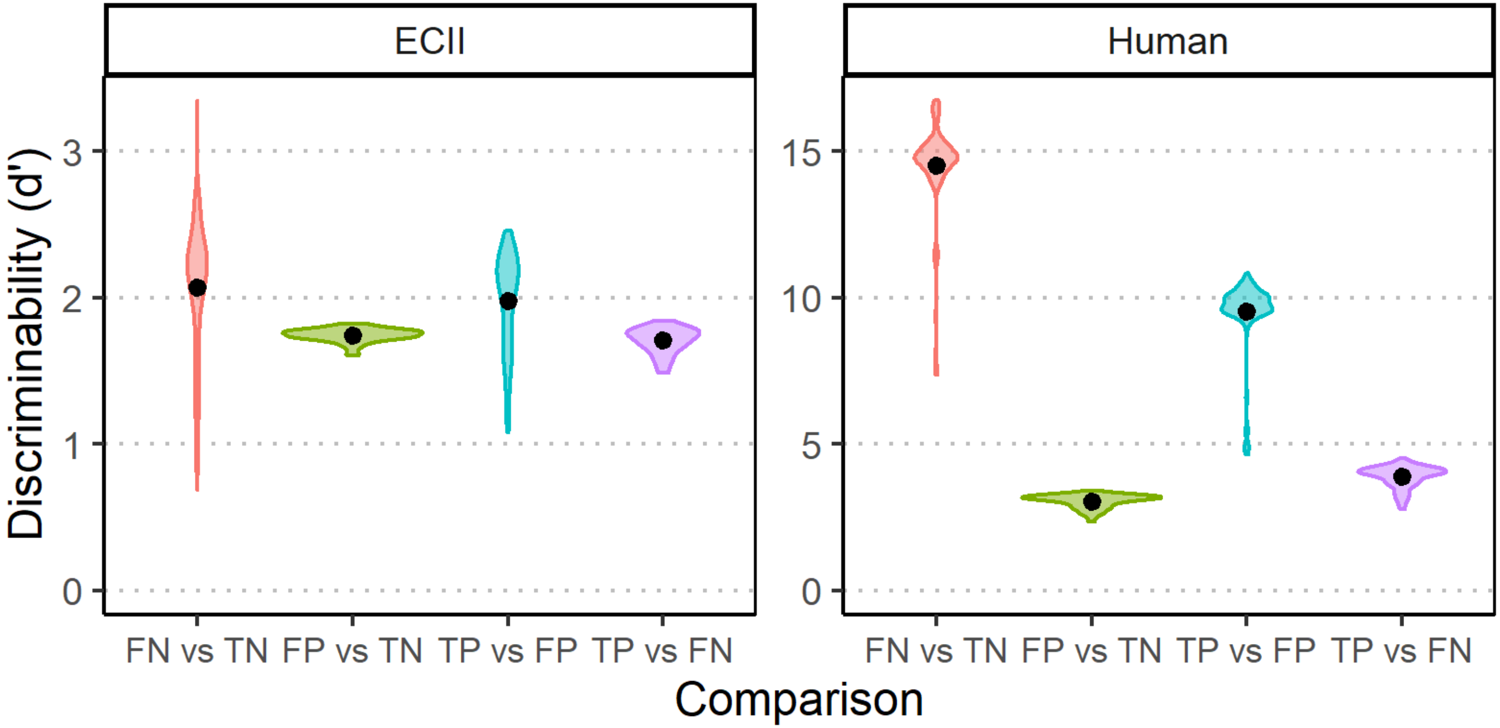
Explanation: Home, Manufacturing, Clothing, Clothing Manufacturers, People, Chairs, Tableware

Which group of images do you think this explanation refers to?

Image Group A	Image Group B

Are the results human-compatible? Part II

- **Bayesian hierarchical signal-detection model (SDT)**
  - **yields discriminability measure**



Summary

- We have clear indications that concept induction can help decipher hidden layer activations.

- Concept induction explanations appear to be meaningful to humans.

- There is lots of work to do
  - sharpening the explanation results
  - in particular, understanding metaparameters
  - in particular, what does *not* activate each neuron?
  - does the activated neuron contribute to the output?
  - how can we cast this into a practical explanations interace?

Deep Deductive Reasoning

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, Pascal Hitzler, Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners. Applied Intelligence 51 (9), 6326-6348, 2021.

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

Hitzler, Rayan, Zalewski, Saki Norouzi, Eberhart, Vasserman, Deep Deductive Reasoning is a Hard Deep Learning Problem, 2023, under review for Neurosymbolic Artificial Intelligence.

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 DDR work

[Hitzler, Rayan, Zalewski, Saki Norouzi, Eberhart, Vasserman, NAI 2023]

Method	Logic	Generative	Transferable	Scalability	Training time	Testing time	Accuracy
[7]	RDF	No	No	Moderate	$\approx$ 60 min	$<$ 1ms	Accuracy of 87-99%
[8]	RDF	**Yes**	No	Moderate	N/A	N/A	F1-score of 0.03-1
[9]	RDFS	**Yes**	No	Low	$\approx$ 12 min	N/A	Accuracy of $\approx$ 100%
[10]	RDFS	**Yes**	No	Low	N/A	N/A	Accuracy of 95%
[11]	RDFS	No	**Yes**	Moderate	N/A	N/A	Accuracy of 52-96%
[12]	$\mathcal{EL}^+$	**Yes**	No	Moderate	N/A	N/A	Somewhat better than guessing
[13]	$\mathcal{EL}^{++}$	No	No	High	N/A	N/A	Hits at rank 1 $\approx$ 0.06
[14]	OWL 2 RL	No	No	Low	N/A	N/A	Accuracy of 99%
[15]	ASP	**Yes**	No	Very low	N/A	N/A	N/A
[16]	OWL DL	**Yes**	No	High	N/A	$<$ 355 s	F1-score of $\approx$ 0.95
[17]	$\mathcal{ALC}$	No	No	Moderate	$<$ 27 min	N/A	Accuracy of $\approx$ 97%
[18]	FOL	**Yes**	No	Very low	$\approx$ 20 sec	N/A	Precision of 0.7

DDR theoretical limitations

With reasonable assumptions on complexity analysis:

- Logics of ExpTime or harder (such as OWL DL) are beyond the scope of deep learning – more precisely it is not possible to learn *precise* reasoning over such logics under reasonable assumptions on the size of the network.

- This means that even NP-complete reasoning (such as SAT) may be out of scope.

Details/discussion in
Reasoning is a Hard Deep Learning Problem, 2023, under review for Neurosymbolic Artificial Intelligence.

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

- [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

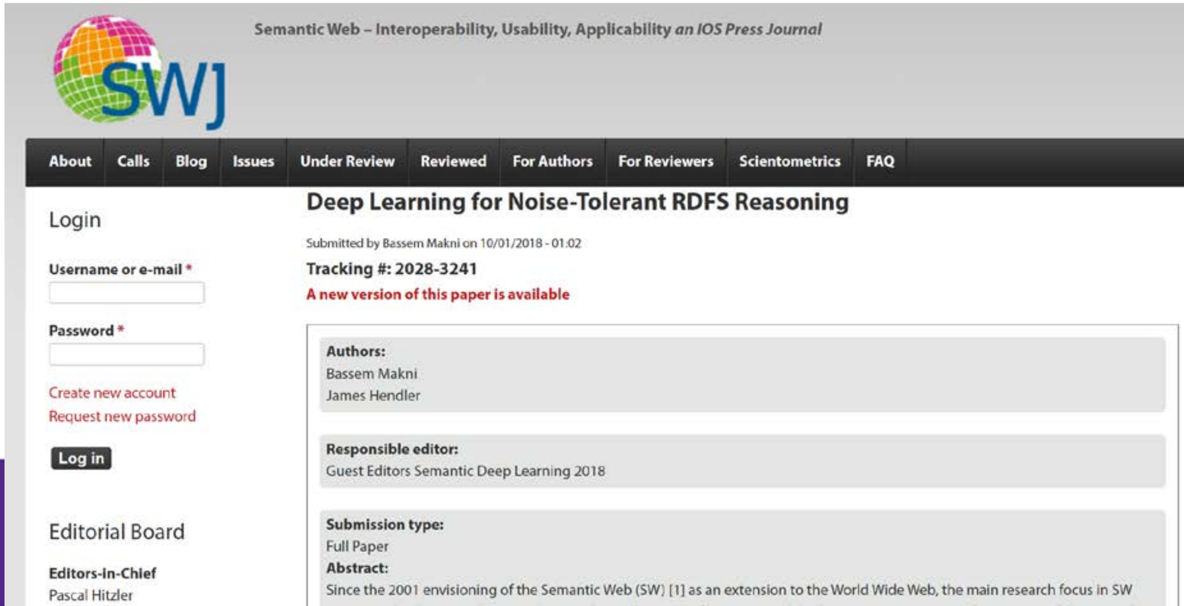
BarackObama	rdf:type	President
BarackObama	husbandOf	MichelleObama
President	rdfs:subClassOf	Human
husbandOf	rdfs:subPropertyOf	spouseOf

- 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)

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.



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.

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)

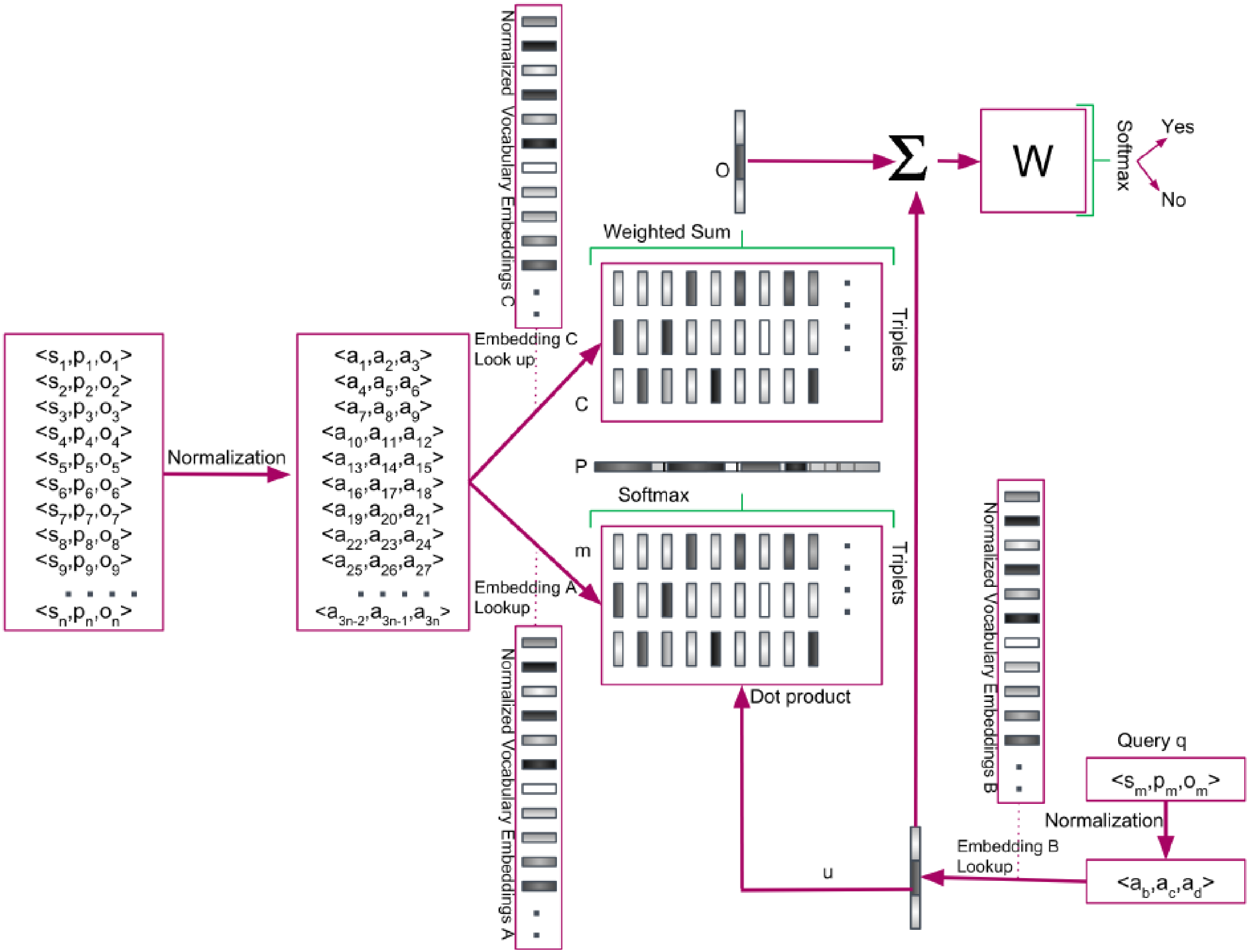
- Knowledge graph is stored in “memory”

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



Experiments: Performance

Training	Test	Valid Triples Class			Invalid Triples Class			Accuracy
		Prec (%)	Rec	F-Measure	Prec	Rec	F-Measure	
A	LD 1	93	98	96	98	93	95	**96**
A (90%)	A (10%)	88	91	89	90	88	89	**90**
A	B	79	62	68	70	84	76	**69**
A	Synth 1	65	49	40	52	54	42	**52**
A	LD 2	54	98	70	91	16	27	86
C	LD 2	62	72	67	67	56	61	91
C (90%)	C (10%)	79	72	75	74	81	77	80
A	D	58	68	62	62	50	54	58
C	D	77	57	65	66	82	73	73
A	Synth 2	70	51	40	47	52	38	51
C	Synth 2	67	23	25	52	80	62	50

Baseline: non-normalized embeddings, same architecture

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

Thanks!

References

Dhar, S., Shamir, L., 2021, Visual Informatics, 5(3), 92-101

Dhar, S., Shamir, L., 2022, Astronomy and Computing, 38, 100545

Abhilekha Dalal, Md Kamruzzaman Sarker, Adrita Barua, Eugene Vasserman, Pascal Hitzler, https://arxiv.org/abs/2308.03999.

Jens Lehmann, Pascal Hitzler, Concept learning in description logics using refinement operators. Mach. Learn. 78(1-2): 203-250 (2010)

Md Kamruzzaman Sarker, Pascal Hitzler, Efficient Concept Induction for Description Logics. In:The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press 2019 , pp. 3036-3043.

References

Md Kamruzzaman Sarker, Joshua Schwartz, Pascal Hitzler, Lu Zhou, Srikanth Nadella, Brandon Minnery, Ion Juvina, Michael L. Raymer, William R. Aue, Wikipedia Knowledge Graph for Explainable AI. In: Boris Villazón-Terrazas, Fernando Ortiz-Rodríguezm Sanju M. Tiwari, Shishir K. Shandilya (eds.), Knowledge Graphs and Semantic Web. Second Iberoamerican Conference and First Indo-American Conference, KGSWC 2020, Mérida, Mexico, November 26-27, 2020, Proceedings. Communications in Computer and Information Science, vol. 1232, Springer, Heidelberg, 2020, pp. 72-87.

Cara Widmer, Md Kamruzzaman Sarker, Srikanth Nadella, Joshua Fiechter, Ion Juvina, Brandon Minnery, Pascal Hitzler, Joshua Schwartz, Michael Raymer, Towards Human-Compatible XAI: Explaining Data Differentials with Concept Induction over Background Knowledge https://arxiv.org/abs/2209.13710

References

Monireh Ebrahimi, Aaron Eberhart, Federico Bianchi, Pascal Hitzler, Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners. Applied Intelligence 51 (9), 6326-6348, 2021.

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

Hitzler, Rayan, Zalewski, Saki Norouzi, Eberhart, Vasserman, Deep Deductive Reasoning is a Hard Deep Learning Problem, 2023, under review for Neurosymbolic Artificial Intelligence.

R. Ferreira, C. Lopes, R. Gonçalves, M. Knorr, L. Krippahl and J. Leite, Deep neural networks for approximating stream reasoning with C-SPARQL, in: Progress in Artificial Intelligence: 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7–9, 2021, Proceedings 20, Springer, 2021, pp. 338–350.

References

X. Zhu, B. Liu, Z. Ding, C. Zhu and L. Yao, Approximate Ontology Reasoning for Domain-Specific Knowledge Graph based on Deep Learning, in: 2021 7th International Conference on Big Data and Information Analytics (BigDIA), 2021, pp. 172–179. doi:10.1109/BigDIA53151.2021.9619694.

B. Makni and J. Hendler, Deep learning for noise-tolerant RDFS reasoning, Semantic Web 10(5) (2019), 823–862.

B. Makni, I. Abdelaziz and J. Hendler, Explainable deep RDFS reasoner, arXiv preprint abs/2002.03514 (2020).

M. Ebrahimi, M.K. Sarker, F. Bianchi, N. Xie, A. Eberhart, D. Doran, H. Kim and P. Hitzler, Neuro-Symbolic Deductive Reasoning for Cross-Knowledge Graph Entailment, in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2021.

A. Eberhart, M. Ebrahimi, L. Zhou, C. Shimizu and P. Hitzler, Completion reasoning emulation for the description logic EL+, arXiv preprint abs/1912.05063 (2019).

References

B. Mohapatra, A. Bhattacharya, S. Bhatia, R. Mutharaju and G. Srinivasaraghavan, EmEL-V: EL Ontology Embeddings for Many-to-Many Relationships

P. Hohenecker and T. Lukasiewicz, Ontology reasoning with deep neural networks, Journal of Artificial Intelligence Research 68 (2020), 503–540.

T. Sato, A. Takemura and T. Inoue, Towards end-to-end ASP computation, arXiv preprint abs/2306.06821 (2023).

X. Zhu, B. Liu, C. Zhu, Z. Ding and L. Yao, Approximate Reasoning for Large-Scale ABox in OWL DL Based on Neural-Symbolic
Learning, Mathematics 11(3) (2023), 495.

D.M. Adamski and J. Potoniec, Reason-able embeddings: Learning concept embeddings with a transferable neural reasoner, Semantic Web
(2023). doi:10.3233/SW-233355.

F. Bianchi and P. Hitzler, On the Capabilities of Logic Tensor Networks for Deductive Reasoning., in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2019.

Overflow slides

DDR via Logic Tensor Networks – doesn't scale

Federico Bianchi, Pascal Hitzler

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

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

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.**

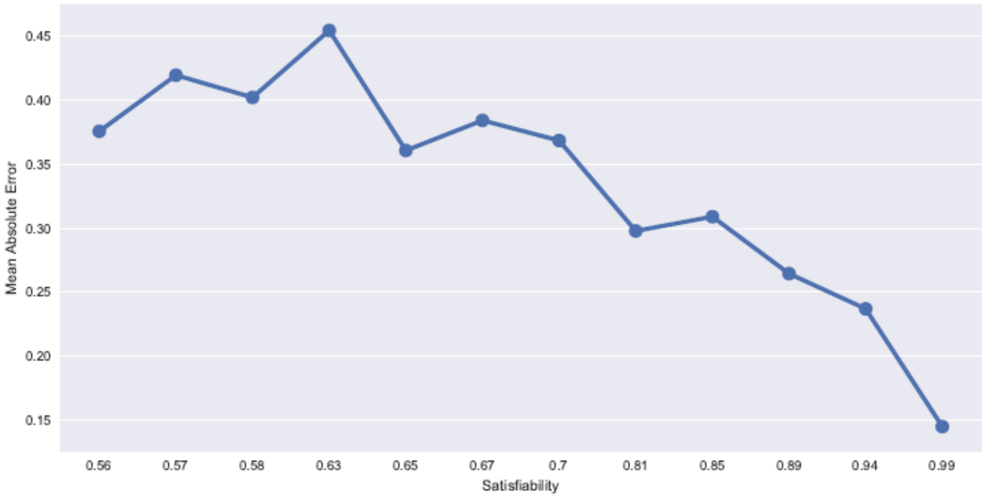


Figure 3: Average MAE for the ancestors tasks on rounded level of satisfiability. MAE decreases with the increase of satisfiability.

- **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

DaSe Lab

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

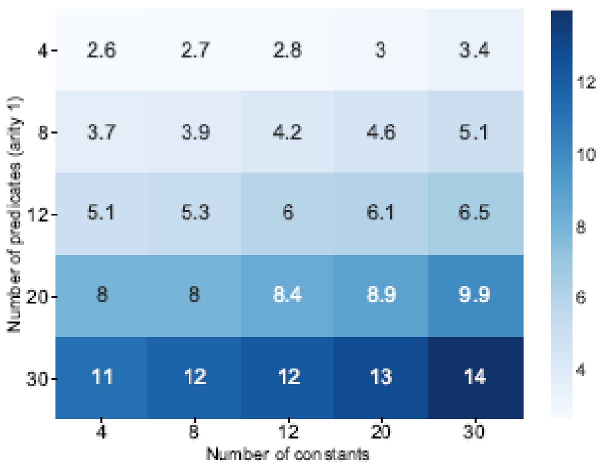


Figure 5: Computational times in seconds for predicates of arity one and constants

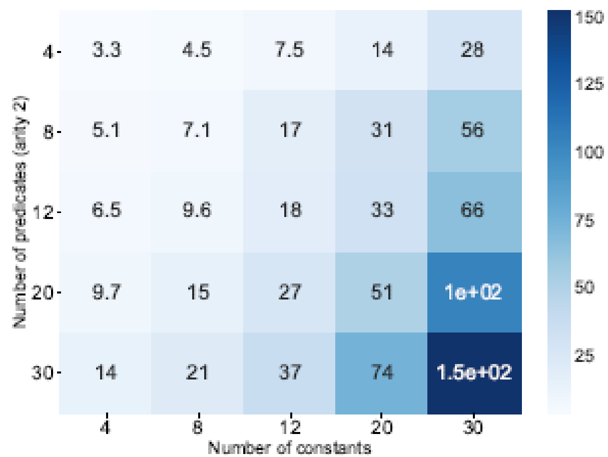


Figure 6: Computational times in seconds for predicates of arity two and constants

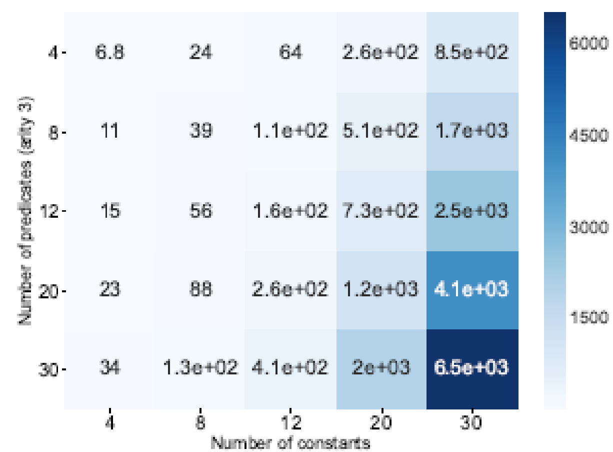


Figure 7: Computational times in seconds for predicates of arity three and constants

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.

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

- [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

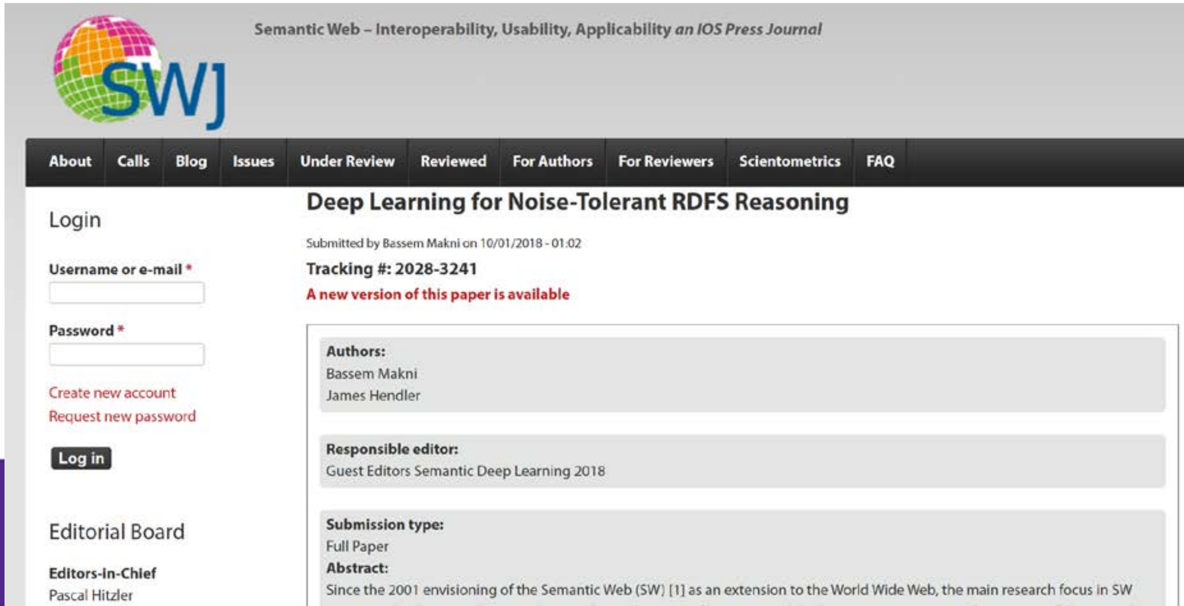
BarackObama	rdf:type	President
BarackObama	husbandOf	MichelleObama
President	rdfs:subClassOf	Human
husbandOf	rdfs:subPropertyOf	spouseOf

- 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)

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.



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.**

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)**

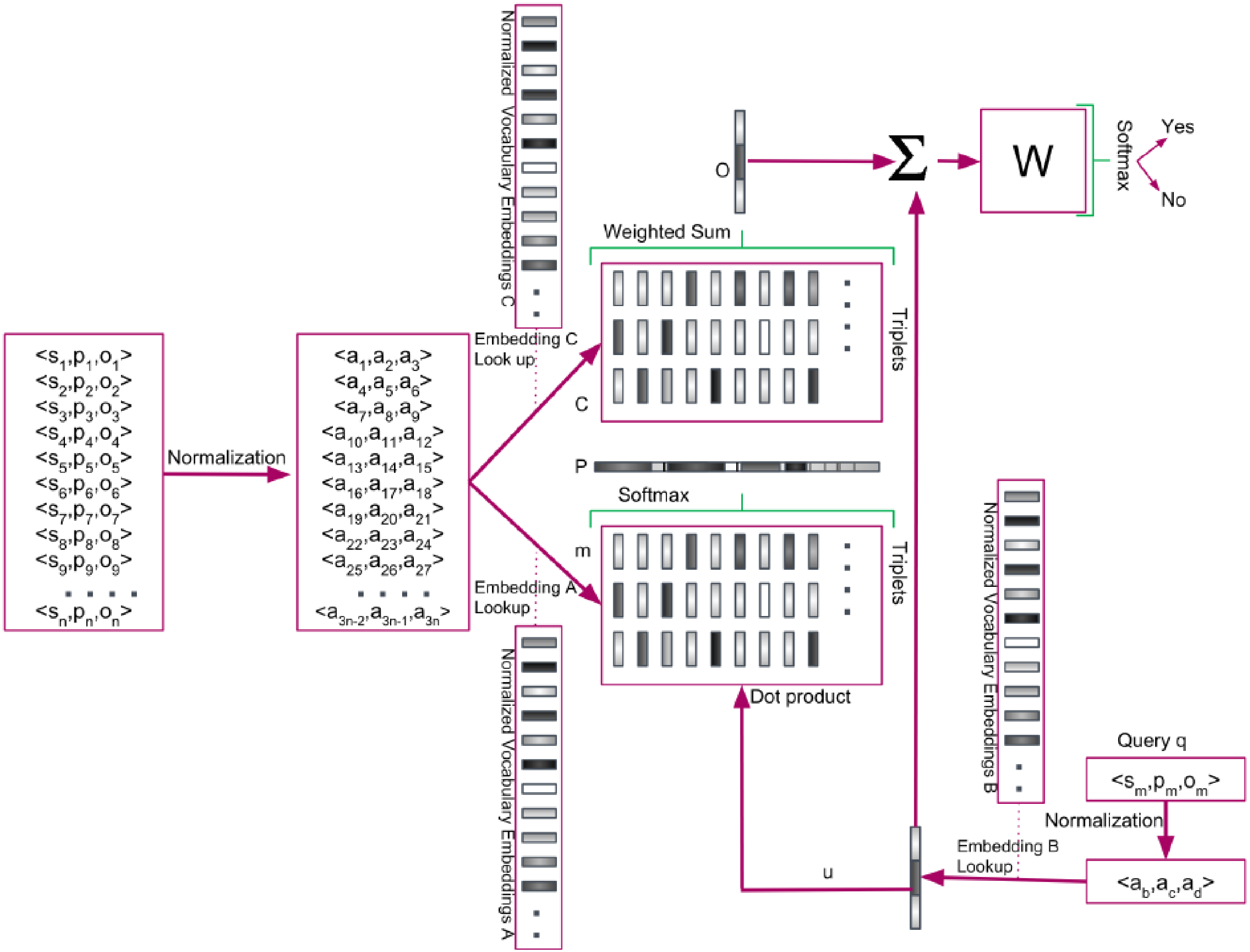
- **Knowledge graph is stored in "memory"**

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



Experiments: Performance

Training	Test	Valid Triples Class			Invalid Triples Class			Accuracy
		Prec (%)	Rec	F-Measure	Prec	Rec	F-Measure	
A	LD 1	93	98	96	98	93	95	**96**
A (90%)	A (10%)	88	91	89	90	88	89	**90**
A	B	79	62	68	70	84	76	**69**
A	Synth 1	65	49	40	52	54	42	**52**
A	LD 2	54	98	70	91	16	27	86
C	LD 2	62	72	67	67	56	61	91
C (90%)	C (10%)	79	72	75	74	81	77	80
A	D	58	68	62	62	50	54	58
C	D	77	57	65	66	82	73	73
A	Synth 2	70	51	40	47	52	38	51
C	Synth 2	67	23	25	52	80	62	50

Baseline: non-normalized embeddings, same architecture

Experiments: Reasoning Depth

Test Dataset	Hop 0			Hop 1			Hop 2			Hop 3			Hop 4			Hop 5		
	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

Test Dataset	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
Linked Data[a]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Linked Data[b]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OWL-Centric[c]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Synthetic	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

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

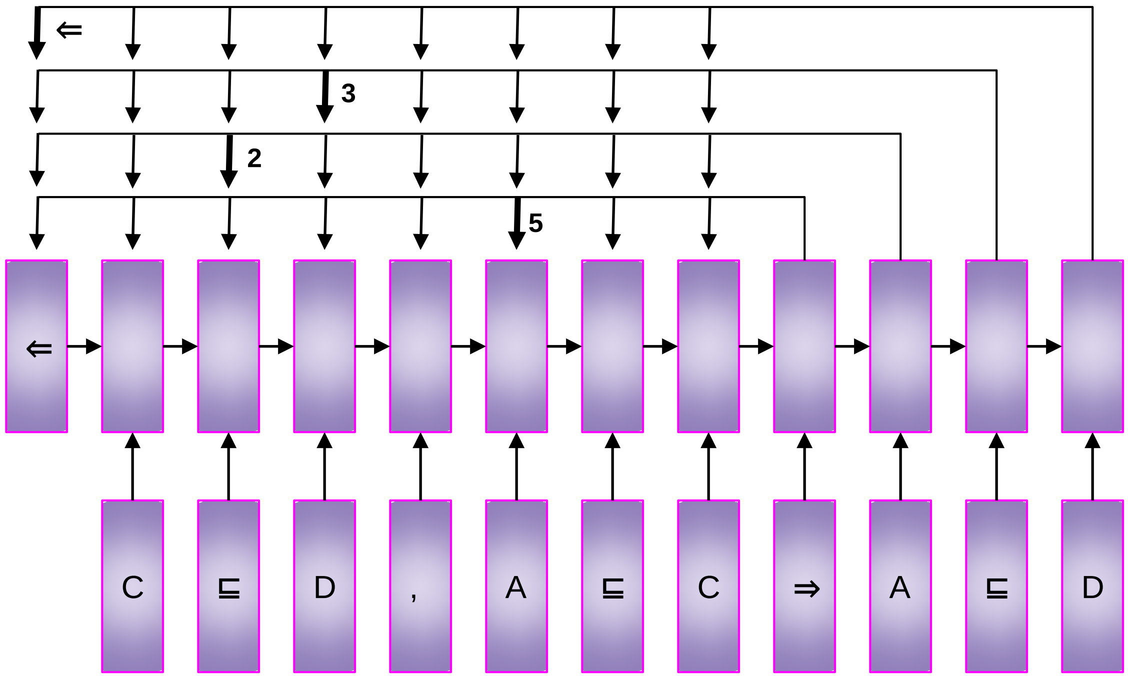
Generative RDFS Reasoning using Pointer Networks – doesn't work

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**



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

Completion Reasoning Emulation for the Description Logic EL+ - hardly works

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

$CX \sqsubseteq CY$

$CX \sqcap CY \sqsubseteq CZ$

$CX \sqsubseteq \exists RY.CZ$

$\exists RX.CY \sqsubseteq CZ$

$RX \sqsubseteq RY$

$RX \circ RY \sqsubseteq RZ$

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 $\circ$ signifying standard binary composition

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