Learning Continuous Semantic Representations of Symbolic Expressions

Symbols to Vectors to Semantics

Miltiadis Allamanis, Pankajan Chanthirasegaran
Pushmeet Kohli and **Charles Sutton**

8 August 2017

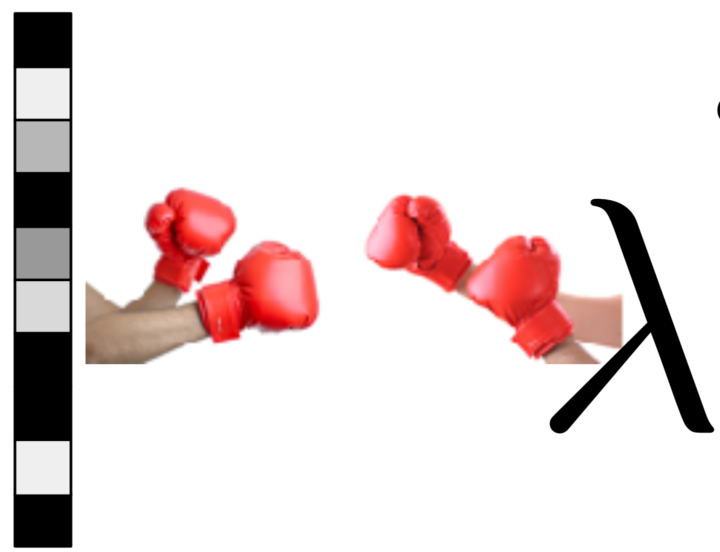
Companion Site: http://edin.ac/sutton-icml2017

THE UNIVERSITY of EDINBURGH informatics
The Alan Turing Institute
Microsoft Research
EPSRC Engineering and Physical Sciences Research Council

*Unify *this*!!!*

Subsymbolic son of a …

Mr Continuous Representation



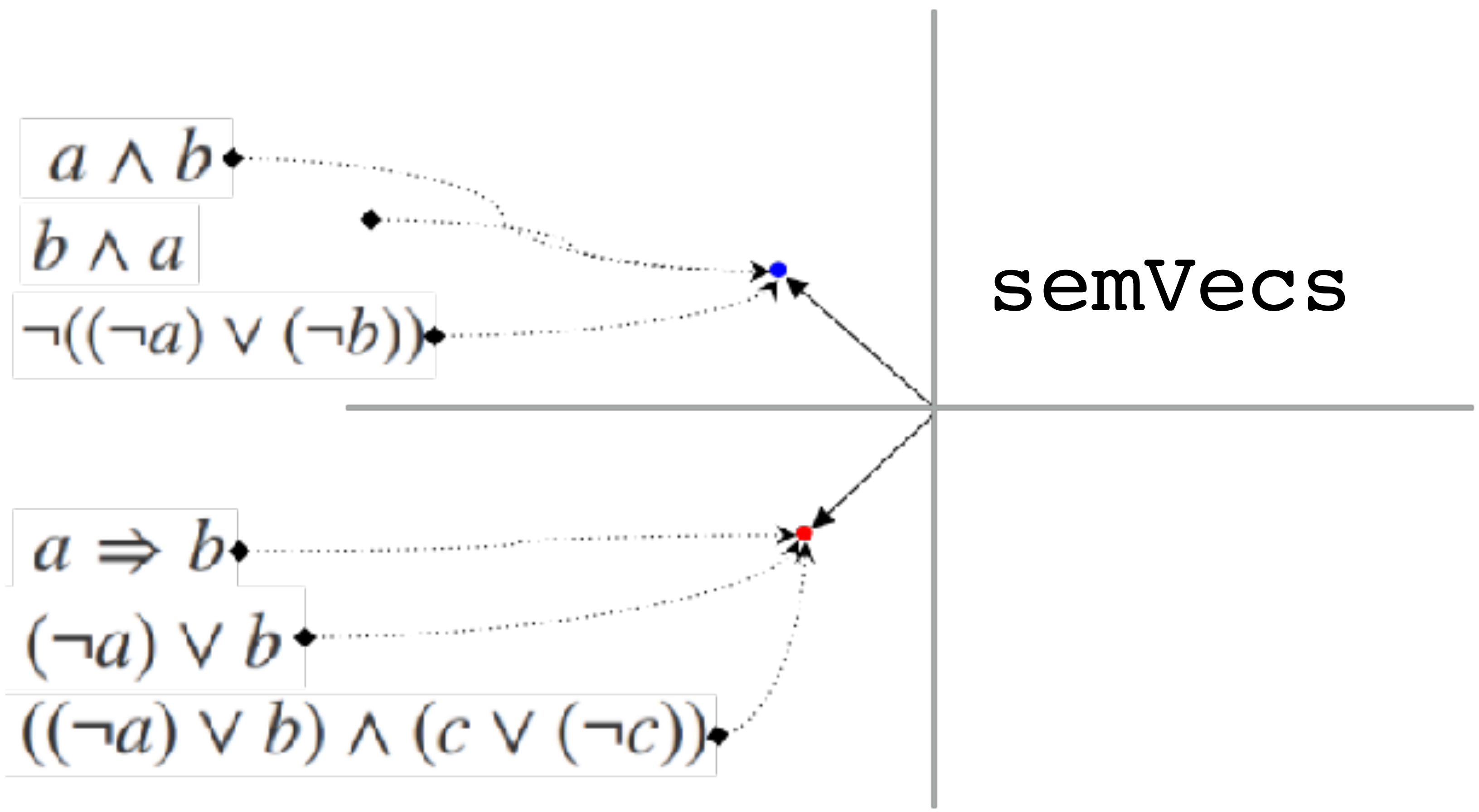
Professor Symbolic AI

Great for perceptual data

Efficient reasoning

Abstract reasoning — powerful transfer of knowledge

Can vectors help symbols?



How much symbolic semantics (**semantic equivalence**)
In this work: semantics = equivalence
can we compress into **continuous** vector?
Want similar continuous vectors —> logically equivalent

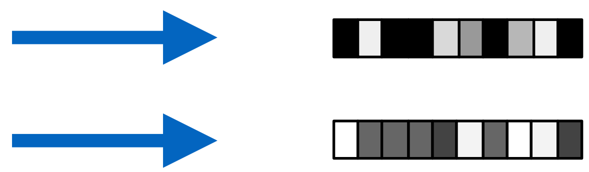
http://edin.ac/sutton-icml2017

Potential Uses

Logical expressions — Continuous vectors (`semVecs`)

$$a \lor (b \implies c)$$

$$a \lor \neg b \lor c$$



Symbolic reasoning: ~~search~~ **pattern recognition**

Theorem Proving

[DeepMath: Irving et al, 2016] *[Zaremba et al, 2014]*

Program Synthesis

[Gulwani et al, CACM 2015]

Inductive Logic Programming

[Rocktaschel and Riedel, 2016] *[Rocktaschel and Riedel, arXiv 1705.11040 2017]*

Transfer Learning

Desiderata

Syntax directed: Semantics is compositional

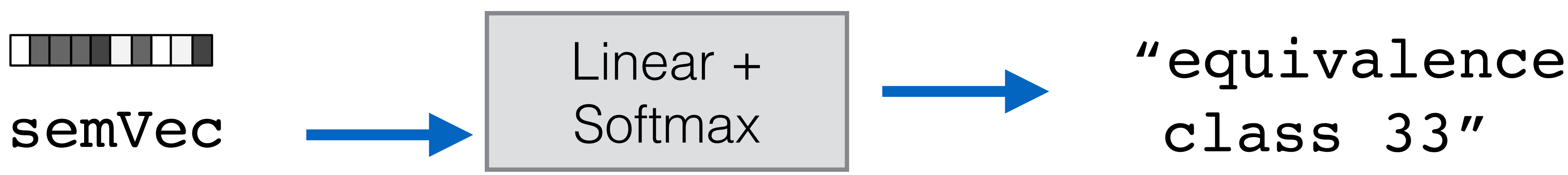
Not too much: Small syntax change —> big semantics

"man bites dog" problem

Computing semVecs

Training

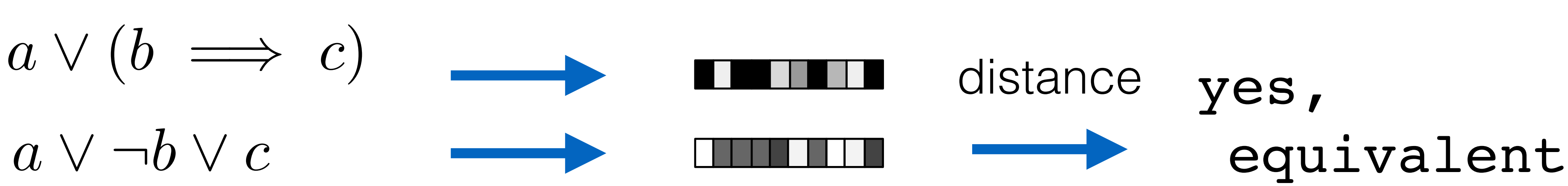
Partition training expressions into equivalence classes



Use a supervised max-margin loss

Testing

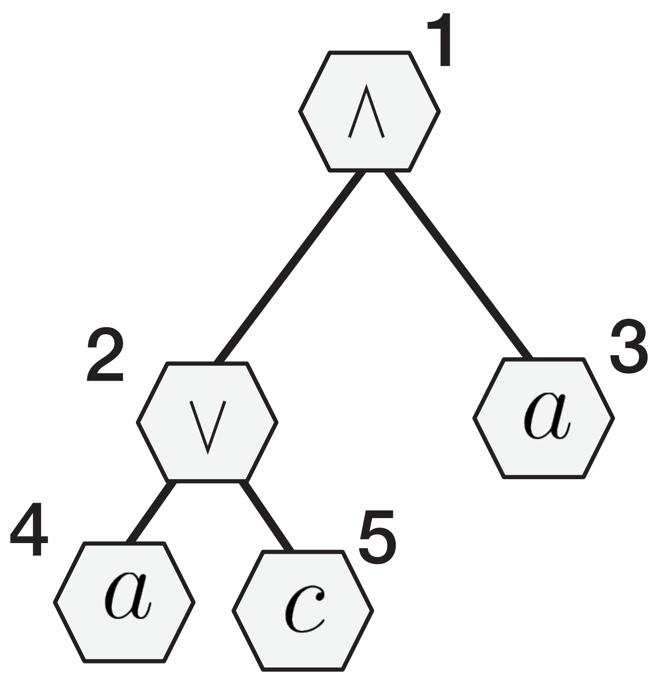
Use a semVec similarity only. Allows zero-shot learning on equiv classes.



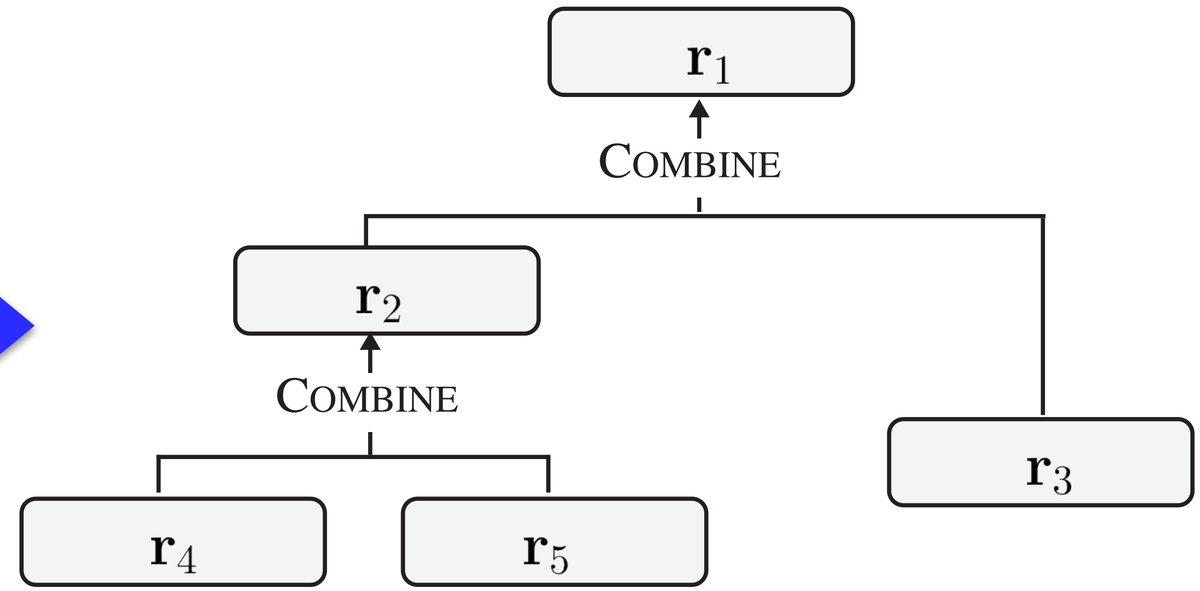
Allows zero-shot learning on equivalence classes.

Recursive NN (TreeNN)

$(a \vee c) \wedge a$



Syntax tree



Network architecture

Problem: Representations mostly syntactic. Too much syntax!

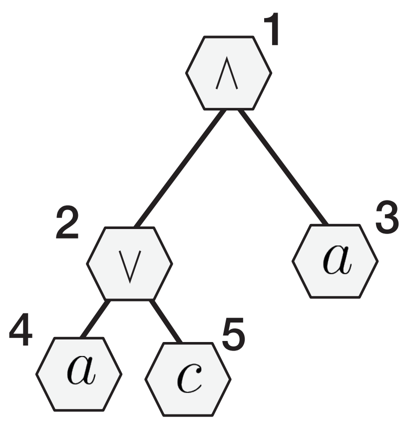
[Socher et al, 2011, 2013]

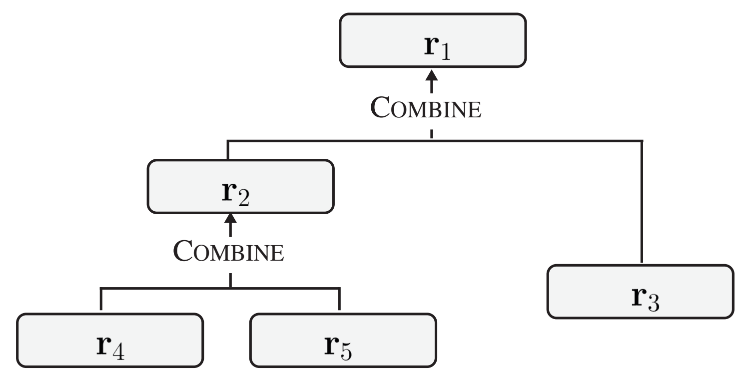
http://edin.ac/sutton-icml2017

EqNet

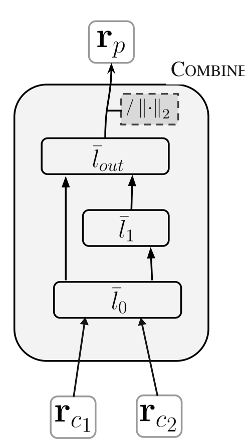
Start with TreeNNs

$(a \vee c) \wedge a$





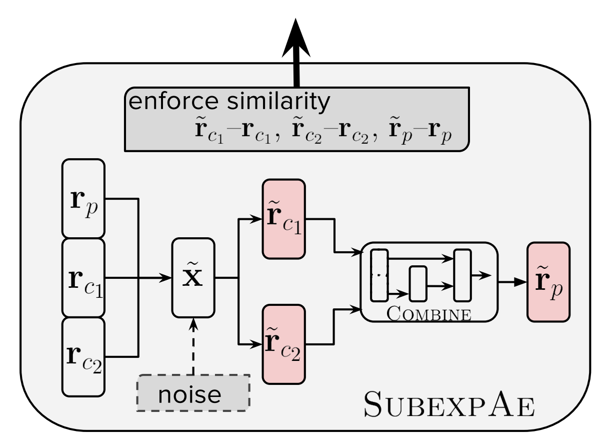
Add:



Moar! Layers!

$$\|\cdot\|_2$$

Normalization

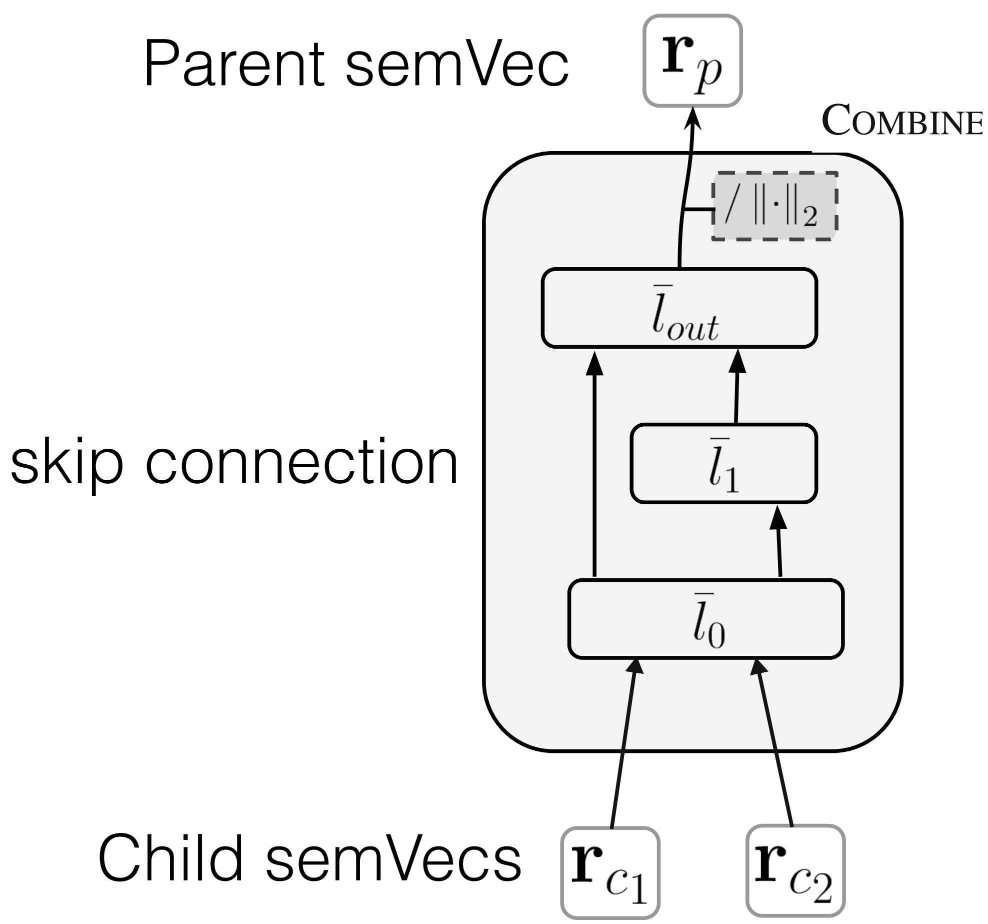


Subexpression AE

http://edin.ac/sutton-icml2017

Layers and Normalization

For one syntactic parent-child



$\textsc{Combine}\ (\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}, \tau_p)$

$\bar{l}_0 \leftarrow [\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}]$

$\bar{l}_1 \leftarrow \sigma\left(W_{i,\tau_p} \cdot \bar{l}_0\right)$

$\bar{l}_{out} \leftarrow W_{o0,\tau_p} \cdot \bar{l}_0 + W_{o1,\tau_p} \cdot \bar{l}_1$

return $\bar{l}_{out} / \left\|\bar{l}_{out}\right\|_2$

Big impact.

(Turns out you need both residual and normalisation together)

SubexprAE: Motivation

Semantic information is bidirectional

Not only do **children** provide info re **parents**

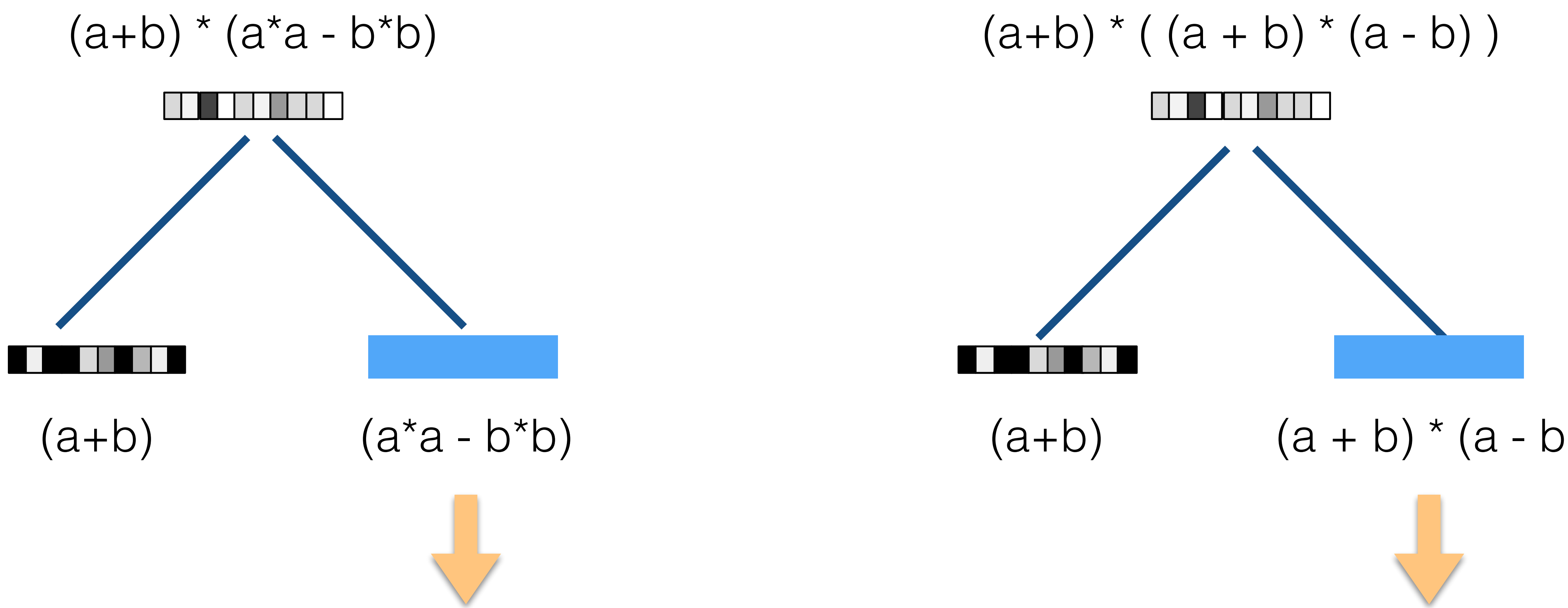
But **parents** provide info re **children**

```
uncle(?B,?A) :- parent(?Z,?A), brother(?Z,?B).
```

Unification propagates this info automatically

How to map to continuous space?

SubexprAE Motivation



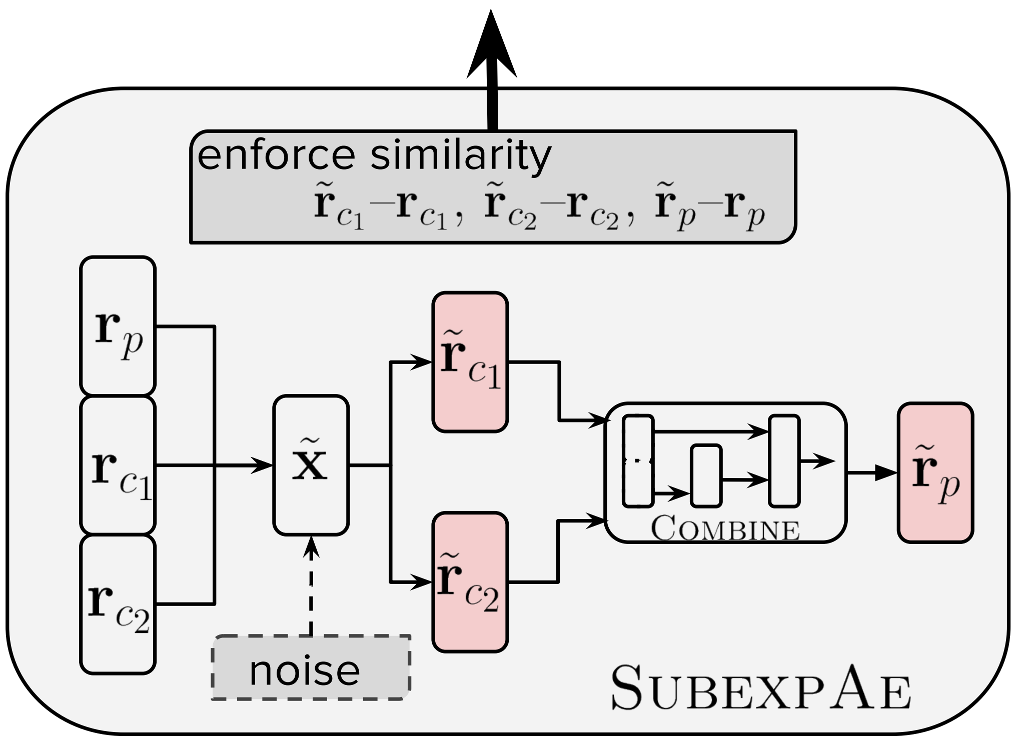
ensure this prediction problem is “easy”

semantic classes will be clustered together

http://edin.ac/sutton-icml2017

Subexpression Autoencoder

For every node in syntax tree, add regularisation



Denoising autoencoder plus bottleneck on (parent, child1, child2) semVecs

Intention is

Bottleneck → Abstraction

Denoising → Reversibility

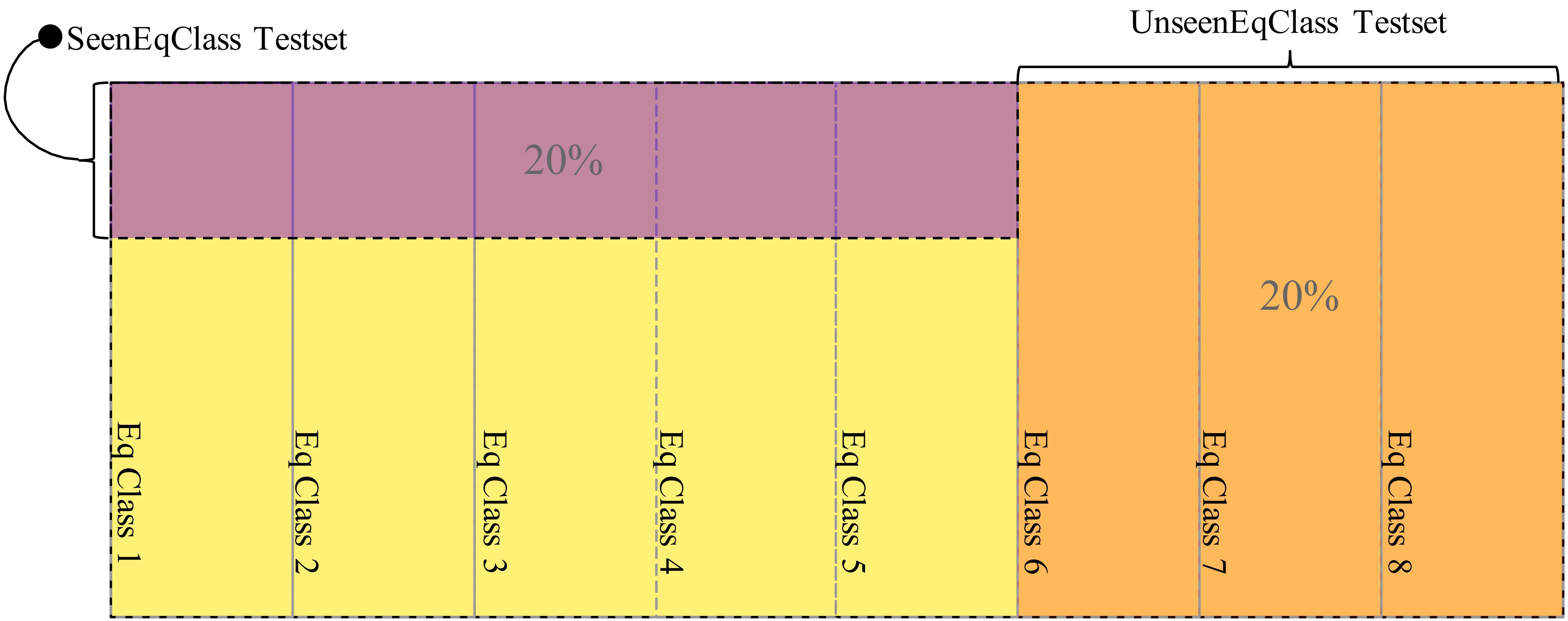
http://edin.ac/sutton-icml2017

Evaluation

Dataset	# Vars	# Equiv Classes	# Exprs	H
SIMPBOOL8	3	120	39,048	5.6
SIMPBOOL10^S	3	191	26,304	7.2
BOOL5	3	95	1,239	5.6
BOOL8	3	232	257,784	6.2
BOOL10^S	10	256	51,299	8.0
SIMPBOOLL5	10	1,342	10,050	9.9
BOOLL5	10	7,312	36,050	11.8
SIMPPOLY5	3	47	237	5.0
SIMPPOLY8	3	104	3,477	5.8
SIMPPOLY10	3	195	57,909	6.3
ONEV-POLY10	1	83	1,291	5.4
ONEV-POLY13	1	677	107,725	7.1
POLY5	3	150	516	6.7
POLY8	3	1,102	11,451	9.0

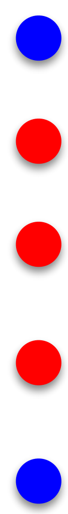
http://edin.ac/sutton-icml2017

Training / Test Split



http://edin.ac/sutton-icml2017

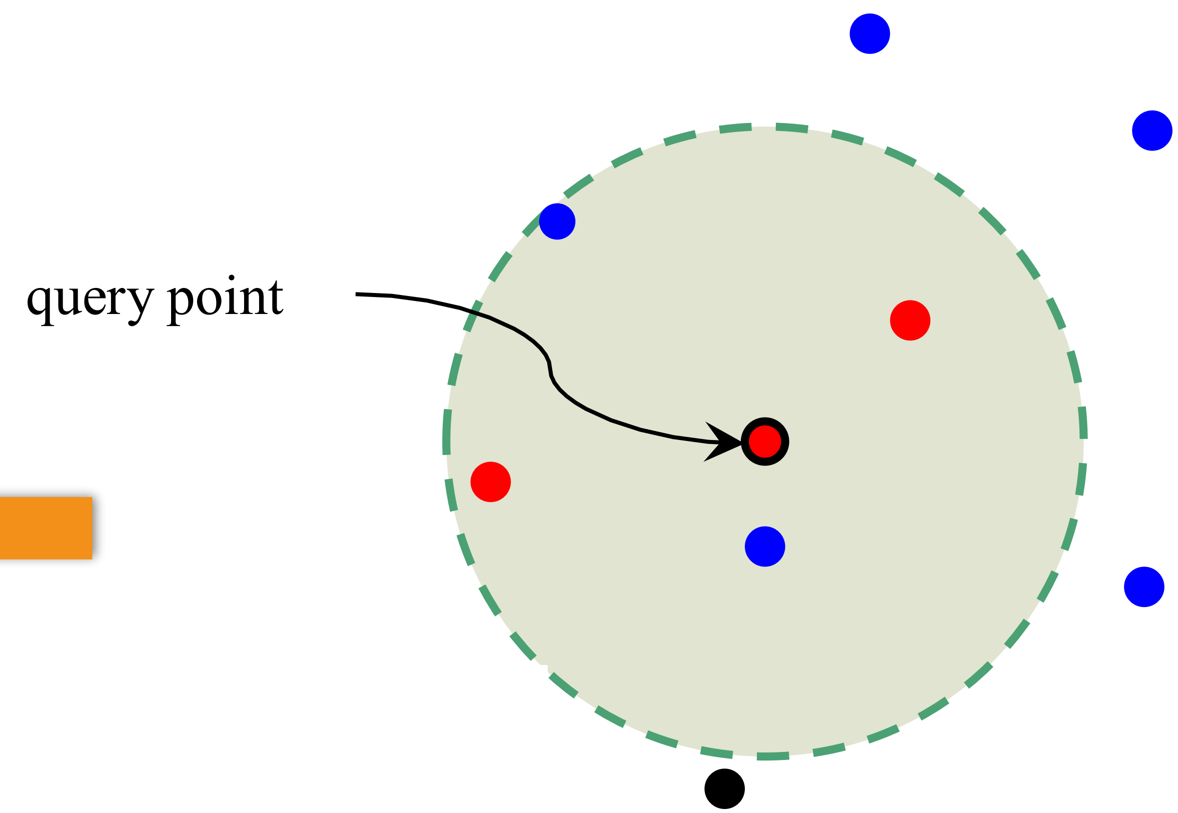
Evaluation Metric



Ranked list

(by distance to query)

Precision
and recall



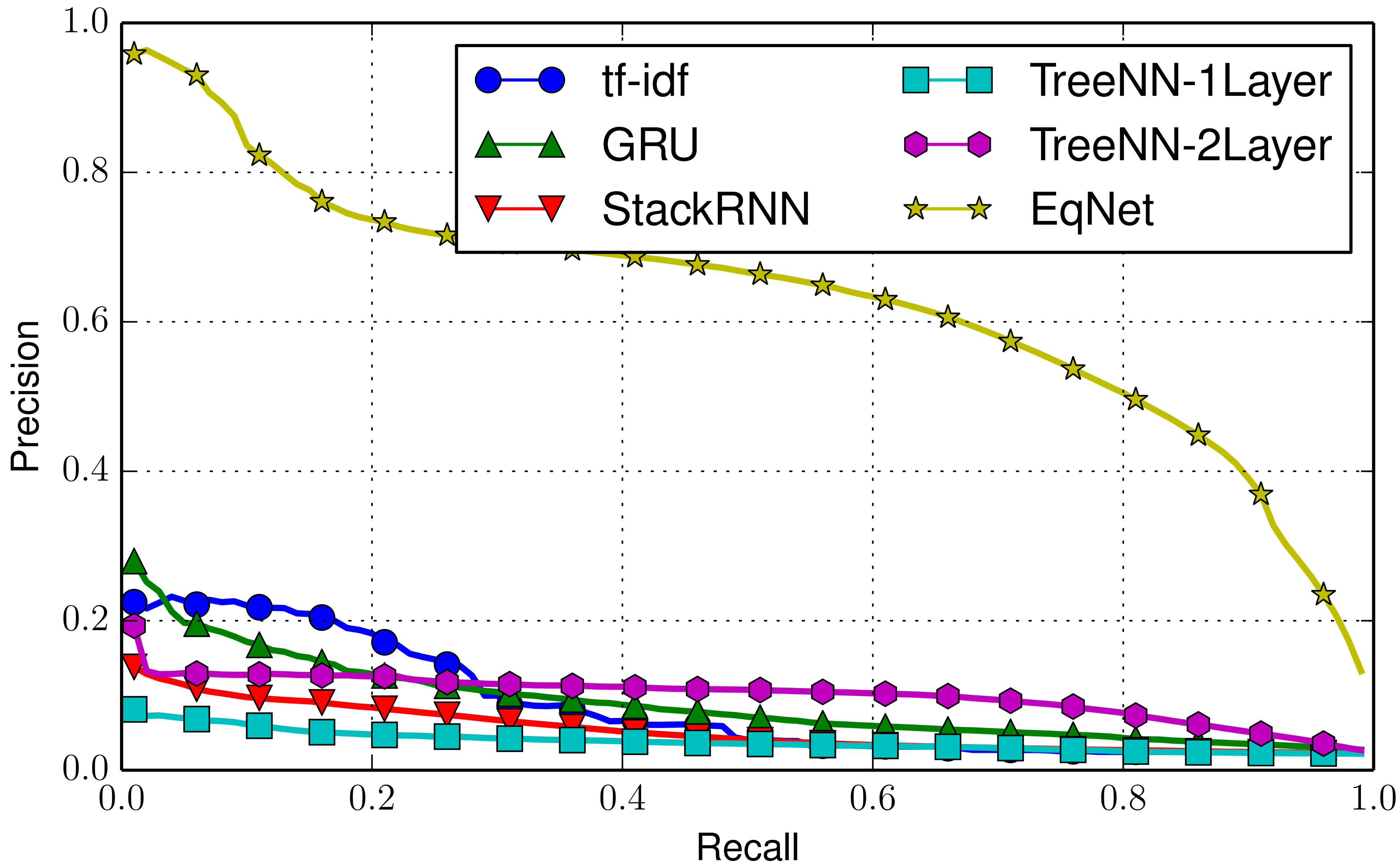
semantic space

Same equivalence class to query
Different equivalence class to query

http://edin.ac/sutton-icml2017

Seen equivalence classes

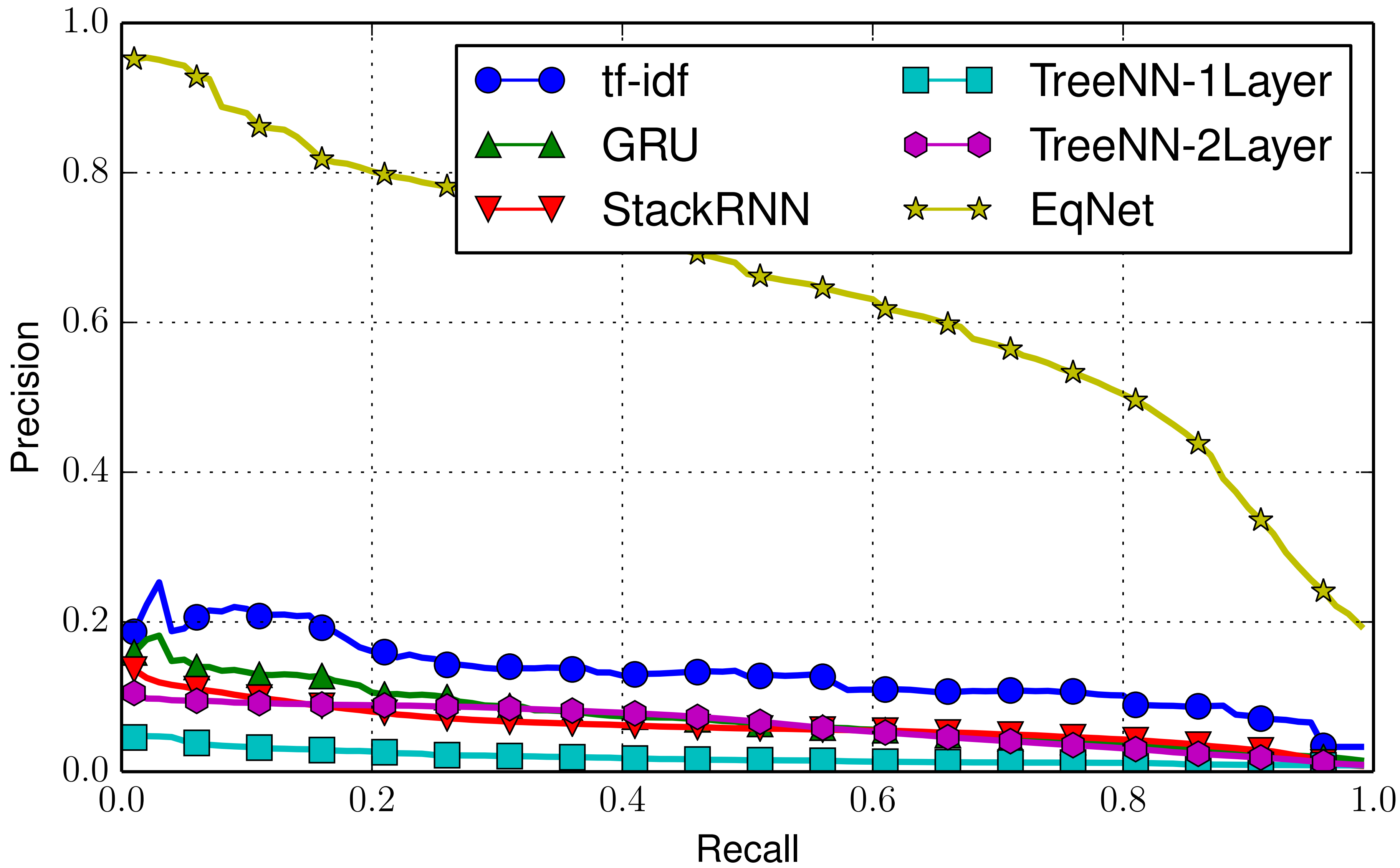
Equivalent expressions to the queries were in training set



http://edin.ac/sutton-icml2017

Unseen equivalence classes

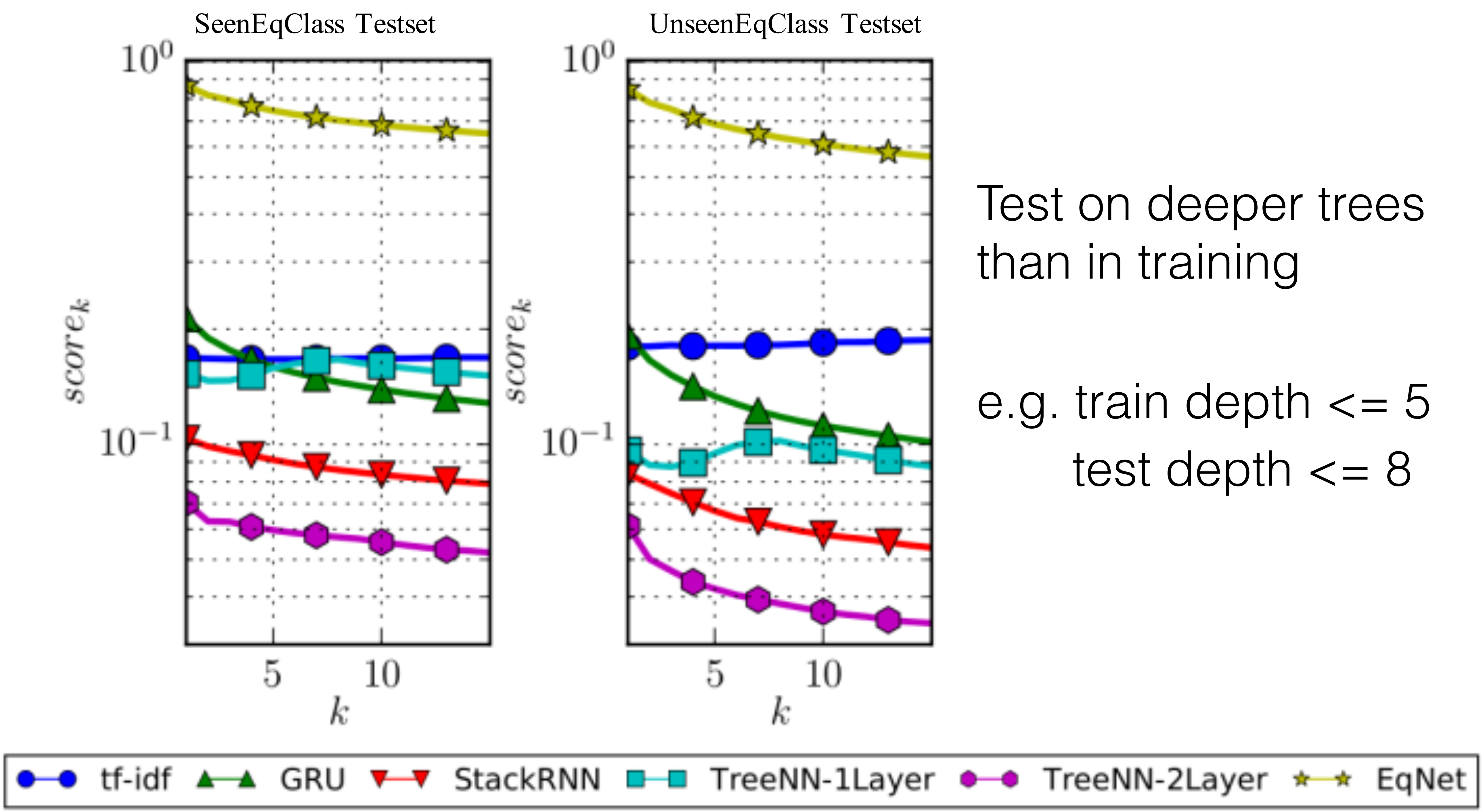
Zero shot learning. No training examples of equivalent expressions.



EqNet performance on seen and unseen is similar!

http://edin.ac/sutton-icml2017

Learned compositionality?



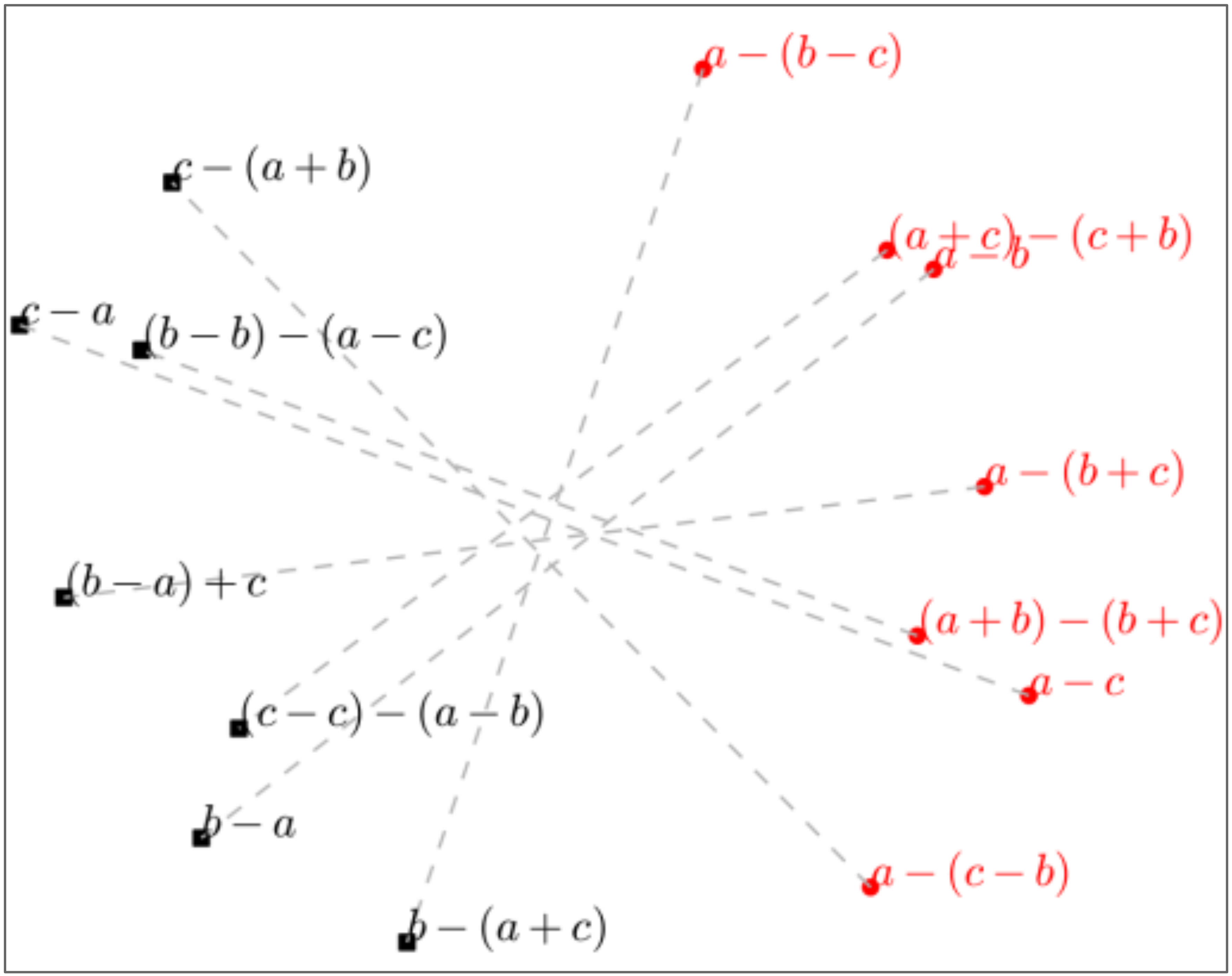
Test on deeper trees than in training

e.g. train depth <= 5
test depth <= 8

http://edin.ac/sutton-icml2017

Visualizing polynomials

`multivariatePolynomial2vec?`

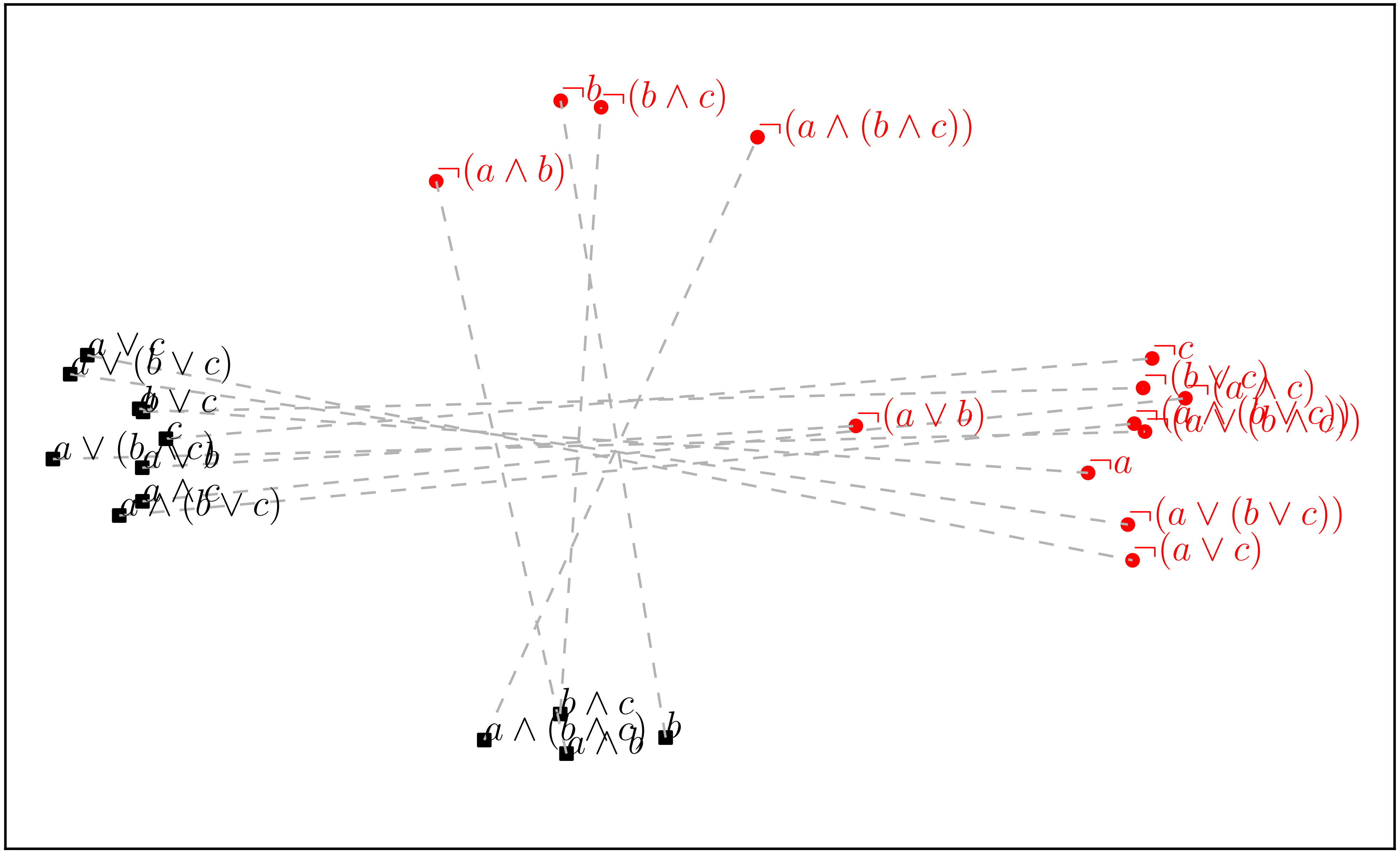


PCA visualization of semVecs

http://edin.ac/sutton-icml2017

Visualizing boolean expression

`booleanExpression2vec?`



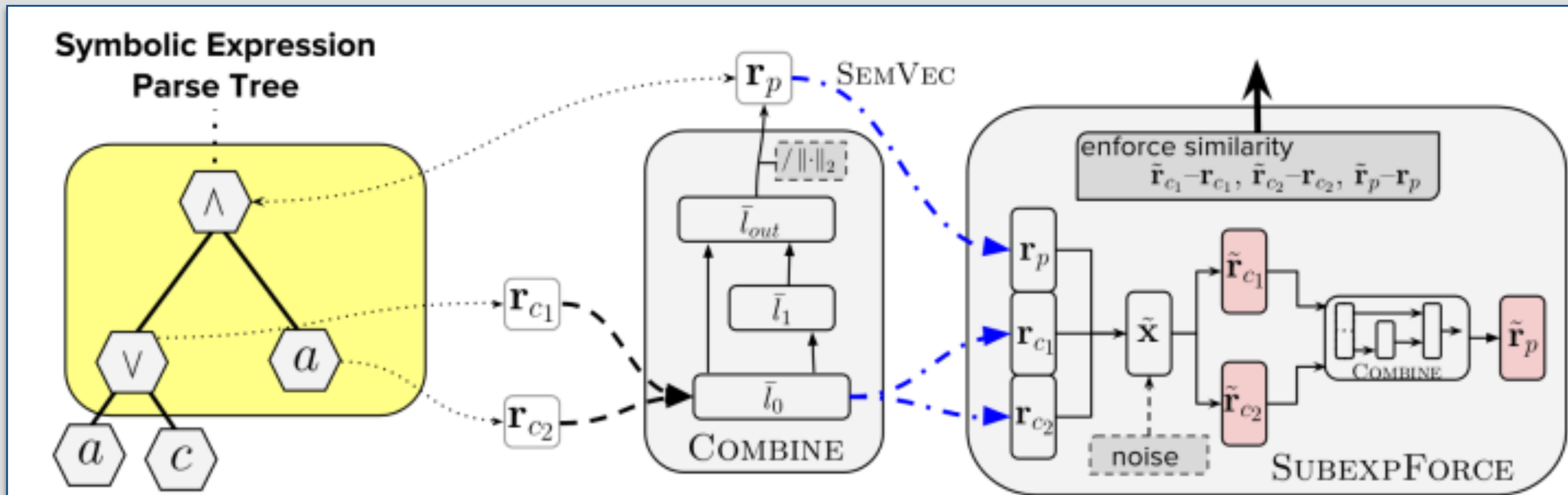
PCA visualization of semVecs

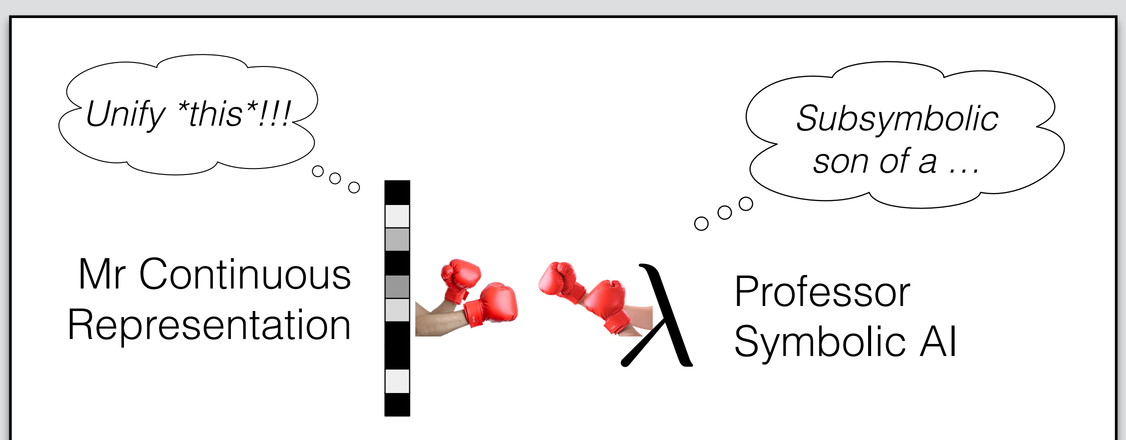
http://edin.ac/sutton-icml2017

Learning Continuous Semantic Representations of Symbolic Expressions

Miltiadis Allamanis, Pankajan Chanthirasegaran,
Pushmeet Kohli and Charles Sutton

http://edin.ac/sutton-icml2017





THE UNIVERSITY of EDINBURGH informatics
The Alan Turing Institute
Microsoft Research
EPSRC Engineering and Physical Sciences Research Council