End-to-End Differentiable Proving

Tim Rocktäschel

Whiteson Research Lab, University of Oxford

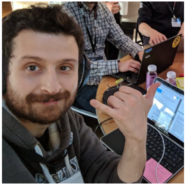
http://rockt.github.com   Twitter: @_rockt   tim.rocktaschel@cs.ox.ac.uk

Logic and Learning Workshop at The Alan Turing Institute

January 12, 2018

Joint Work With

Sebastian Riedel
University College London



Pasquale Minervini
University College London

Thomas Demeester
Ghent University

Sameer Singh
University of California, Irvine

What vegetable is on the plate?
Neural Net: broccoli
Ground Truth: broccoli

What color are the shoes on the person's feet ?
Neural Net: brown
Ground Truth: brown

How many school busses are there?
Neural Net: 2
Ground Truth: 2

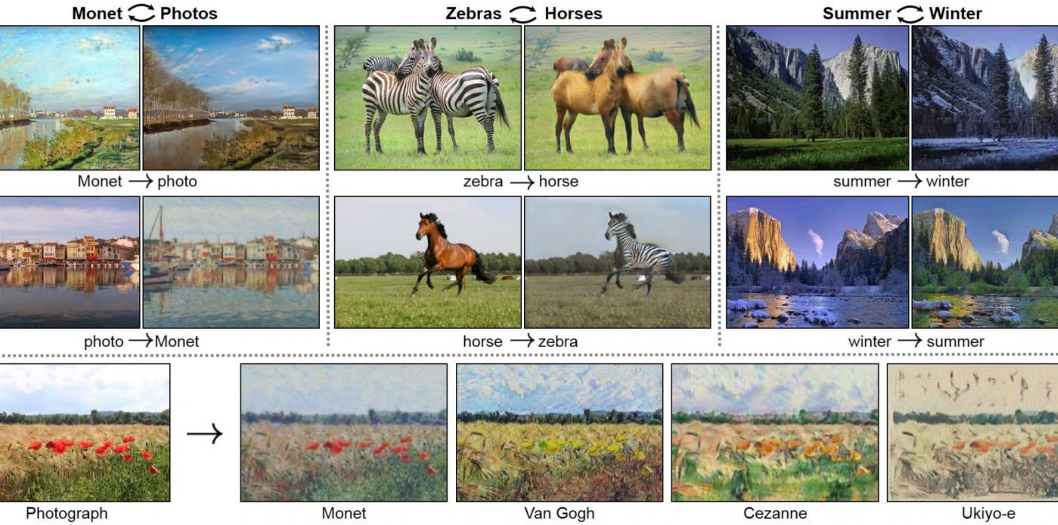
What sport is this?
Neural Net: baseball
Ground Truth: baseball

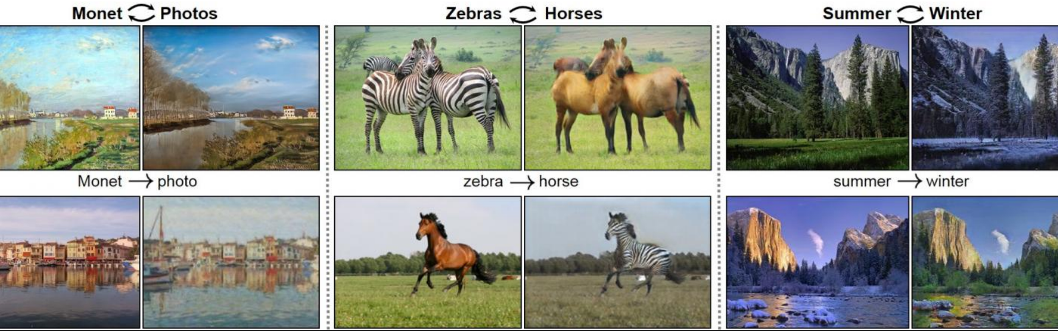
What is on top of the refrigerator?
Neural Net: magnets
Ground Truth: cereal

What uniform is she wearing?
Neural Net: shorts
Ground Truth: girl scout

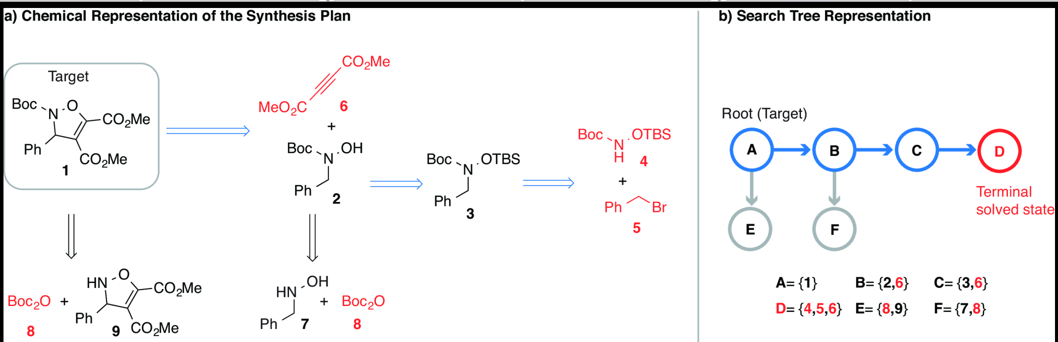
What is the table number?
Neural Net: 4
Ground Truth:40

What are people sitting under in the back?
Neural Net: bench
Ground Truth: tent


Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

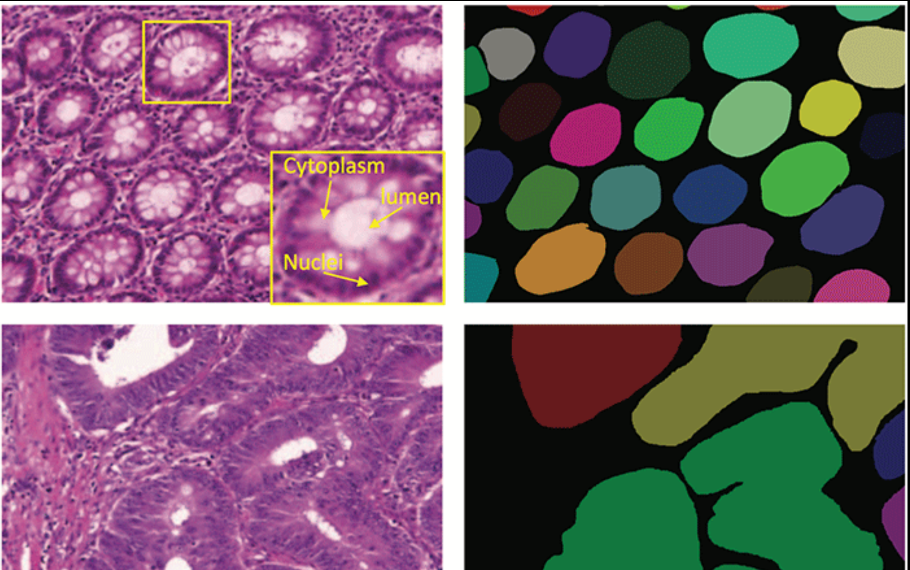


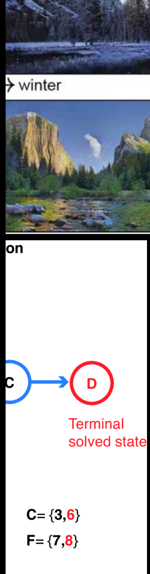
Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter

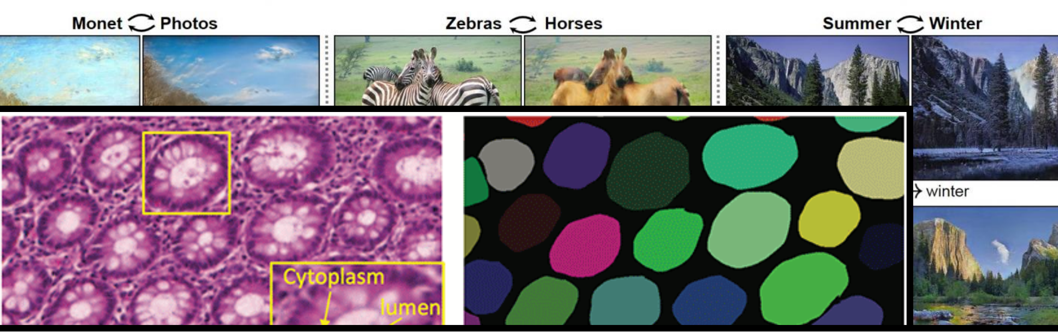


a) Chemical Representation of the Synthesis Plan
Target
Boc
N
O
CO_2Me
Ph
CO_2Me
1
CO_2Me
MeO_2C
6
+
Boc
N
OH
Ph
2
Boc
N
OTBS
Ph
3
Boc
N
H
OTBS
4
+
Ph
Br
5
Boc_2O
8
+
HN
O
CO_2Me
Ph
9
CO_2Me
HN
OH
Ph
7
+
Boc_2O
8
b) Search Tree Representation
Root (Target)
A
B
C
D
E
F
Terminal solved state
A= {1}
B= {2,6}
C= {3,6}
D= {4,5,6}
E= {8,9}
F= {7,8}


Monet Photos
Zebras Horses
Summer Winter


Cytoplasm
lumen
Nuclei


winter
on
C
D
Terminal
solved state
C= {3,6}
F= {7,8}



Translate from **GERMAN** (detected)

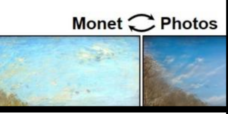
Translate into **ENGLISH**

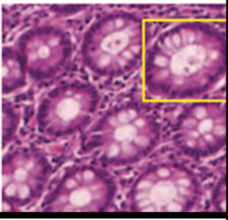
Die Polizei in den USA darf sich wieder schwere Ausrüstung und Waffen beim Militär besorgen. Das hat US-Präsident Donald Trump entschieden und so eine Anordnung seines Vorgängers Barack Obama aufgehoben, nach der es dem Verteidigungsministerium verboten war, die Polizei mit Granatwerfern, gepanzerten Fahrzeugen, Bajonetten, großkalibrigen Waffen und Munition auszurüsten.

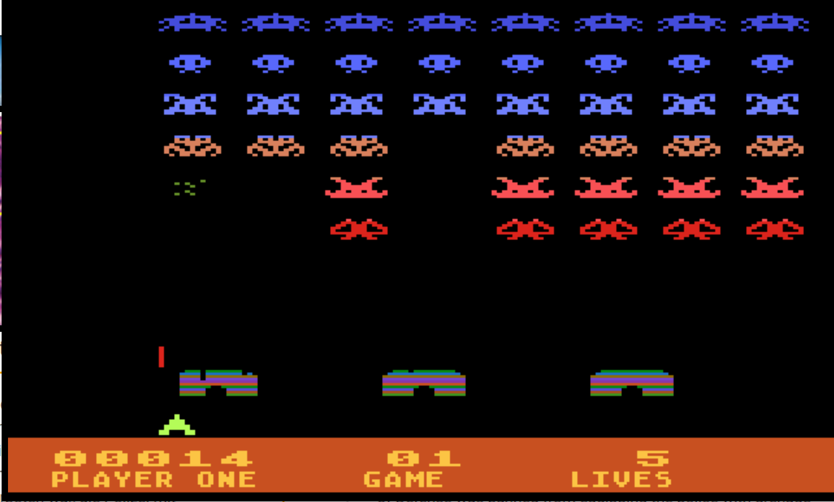
Mit der Maßnahme soll sichergestellt werden, dass die Polizei die lebensrettende Ausrüstung bekomme, die sie brauche, um ihren Job zu machen, sagte US-Justizminister Jeff Sessions.

The police in the USA are allowed to get heavy equipment and weapons from the military again. This was decided by US President Donald Trump, who overturned an order from his predecessor Barack Obama, according to which the Department of Defense was banned from equipping the police with grenade launchers, armoured vehicles, bayonets, large-calibre weapons and ammunition.

The measure is designed to ensure that the police get the life-saving equipment they need to do their job, US Attorney General Jeff Sessions said.







Translat

Die Polizei in den USA darf si
Waffen beim Militär besorgen
Trump entschieden und so ei
Barack Obama aufgehoben, r
Verteidigungsministerium ve
Granatwerfern, gepanzerten Fahrzeugen, Bajonetten, großkalibrigen Waffen und Munition auszurüsten.

launchers, armoured vehicles, bayonets, large-calibre weapons and ammunition.

Mit der Maßnahme soll sichergestellt werden, dass die Polizei die lebensrettende Ausrüstung bekomme, die sie brauche, um ihren Job zu machen, sagte US-Justizminister Jeff Sessions.

The measure is designed to ensure that the police get the life-saving equipment they need to do their job, US Attorney General Jeff Sessions said.

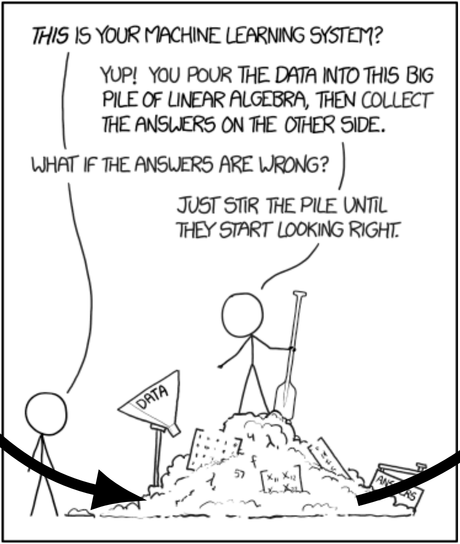
XKCD, 17th May 2017

Data &
Explanations

- Rules
- (Partial) Programs
- Natural Language

XKCD, 17th May 2017

Data &
Explanations
- Rules
- (Partial) Programs
- Natural Language

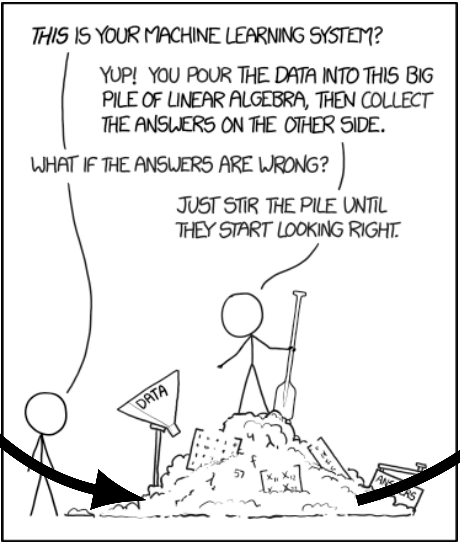


XKCD, 17th May 2017

Answers &
Explanations
- Rules
- Programs
- Natural Language
- Plans

Data &
Explanations

- Rules
- (Partial) Programs
- Natural Language



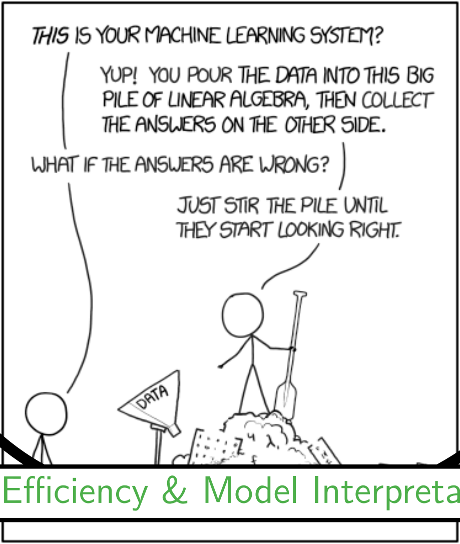
XKCD, 17th May 2017

Answers &
Explanations

- Rules
- Programs
- Natural Language
- Plans

Data &
Explanations

- Rules
- (Partial) Programs
- Natural Language



XKCD, 17th May 2017

Answers &
Explanations

- Rules
- Programs
- Natural Language
- Plans

Data Efficiency & Model Interpretability

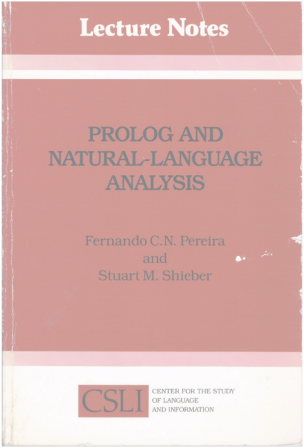
Lecture Notes

PROLOG AND NATURAL-LANGUAGE ANALYSIS

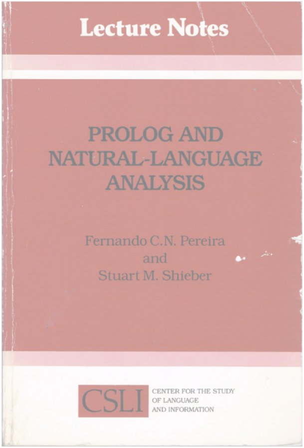
Fernando C.N. Pereira
and
Stuart M. Shieber

CSLI CENTER FOR THE STUDY OF LANGUAGE AND INFORMATION

```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```



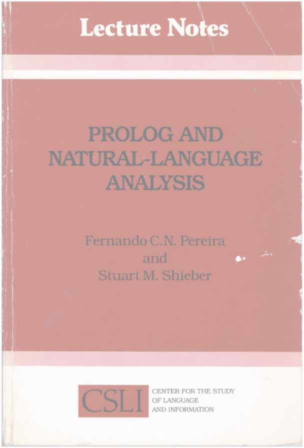
```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```



Expert Systems

- No/little training data
- Interpretable

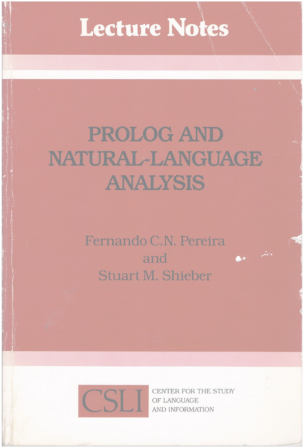
```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```

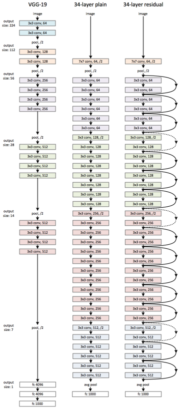


Expert Systems

- No/little training data
- Interpretable
- Rules manually defined
- No generalization

```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```

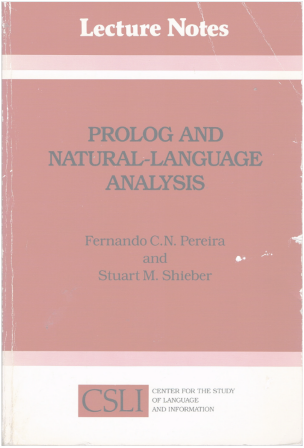


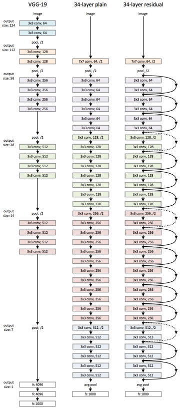


Expert Systems

- No/little training data
- Interpretable
- Rules manually defined
- No generalization

```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```





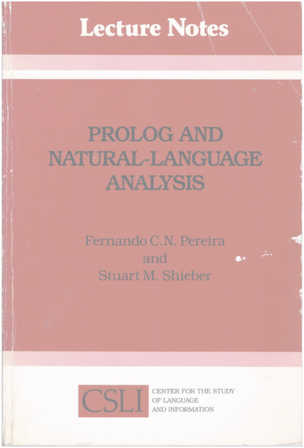
Expert Systems

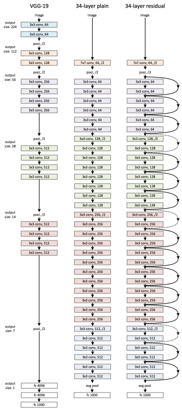
- No/little training data
- Interpretable
- Rules manually defined
- No generalization

Neural Networks

- Trained end-to-end
- Strong generalization

```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```





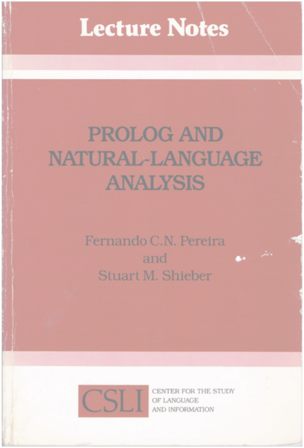
Expert Systems

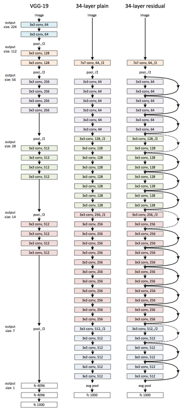
- No/little training data
- Interpretable
- Rules manually defined
- No generalization

Neural Networks

- Need a lot of training data
- Not interpretable
- Trained end-to-end
- Strong generalization

```
goal problem.

rule 1
  if not turn_over and
    battery_bad
  then problem is battery cf 100.

rule 2
  if lights_weak
  then battery_bad cf 50.

rule 3
  if radio_weak
  then battery_bad cf 50.

rule 4
 if turn_over and
    smell_gas
  then problem is flooded cf 80.

rule 5
  if turn_over and
    gas_gauge is empty
  then problem is out_of_gas cf 90.

rule 6
  if turn_over and
    gas_gauge is low
  then problem is out_of_gas cf 30.
```





Expert Systems

- No/little training data
- Interpretable

Neural Networks

- Trained end-to-end
- Strong generalization

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...
  - Statistical Predicate Invention (Kok and Domingos, 2007)

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...
  - Statistical Predicate Invention (Kok and Domingos, 2007)
- Neural-symbolic Connectionism

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...
  - Statistical Predicate Invention (Kok and Domingos, 2007)
- Neural-symbolic Connectionism
  - Propositional rules: EBL-ANN (Shavlik and Towell, 1989), KBANN (Towell and Shavlik, 1994), C-LIP (d'Avila Garcez and Zaverucha, 1999)

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...
  - Statistical Predicate Invention (Kok and Domingos, 2007)
- Neural-symbolic Connectionism
  - Propositional rules: EBL-ANN (Shavlik and Towell, 1989), KBANN (Towell and Shavlik, 1994), C-LIP (d'Avila Garcez and Zaverucha, 1999)
  - First-order inference (no training of symbol representations): Unification Neural Networks (Hölldobler, 1990; Komendantskaya, 2011), SHRUTI (Shastri, 1992), Neural Prolog (Ding, 1995), CLIP++ (Franca et al., 2014), Lifted Relational Networks (Sourek et al., 2015)

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) . . .
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) . . .
  - Statistical Predicate Invention (Kok and Domingos, 2007)
- Neural-symbolic Connectionism
  - Propositional rules: EBL-ANN (Shavlik and Towell, 1989), KBANN (Towell and Shavlik, 1994), C-LIP (d'Avila Garcez and Zaverucha, 1999)
  - First-order inference (no training of symbol representations): Unification Neural Networks (Hölldobler, 1990; Komendantskaya, 2011), SHRUTI (Shastri, 1992), Neural Prolog (Ding, 1995), CLIP++ (Franca et al., 2014), Lifted Relational Networks (Sourek et al., 2015)
- Recent: Logic Tensor Networks (Serafini and d'Avila Garcez, 2016), TensorLog (Cohen, 2016), Differentiable Inductive Logic (Evans and Grefenstette, 2017)

Machine Learning & Logic

- Fuzzy Logic (Zadeh, 1965)
- Probabilistic Logic Programming, e.g.,
  - IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov Logic Networks (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007) ...
- Inductive Logic Programming, e.g.,
  - Plotkin (1970), Shapiro (1991), Muggleton (1991), De Raedt (1999) ...
  - Statistical Predicate Invention (Kok and Domingos, 2007)
- Neural-symbolic Connectionism
  - Propositional rules: EBL-ANN (Shavlik and Towell, 1989), KBANN (Towell and Shavlik, 1994), C-LIP (d'Avila Garcez and Zaverucha, 1999)
  - First-order inference (no training of symbol representations): Unification Neural Networks (Hölldobler, 1990; Komendantskaya, 2011), SHRUTI (Shastri, 1992), Neural Prolog (Ding, 1995), CLIP++ (Franca et al., 2014), Lifted Relational Networks (Sourek et al., 2015)
- Recent: Logic Tensor Networks (Serafini and d'Avila Garcez, 2016), TensorLog (Cohen, 2016), Differentiable Inductive Logic (Evans and Grefenstette, 2017)

For overviews see Besold et al. (2017) and d'Avila Garcez et al. (2012)

Outline

1. Link prediction & symbolic vs. neural representations

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)
   - Model-agnostic and fast (Minervini et al., 2017)

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)
   - Model-agnostic and fast (Minervini et al., 2017)
3. End-to-end differentiable proving (Rocktäschel and Riedel, 2017)

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)
   - Model-agnostic and fast (Minervini et al., 2017)
3. End-to-end differentiable proving (Rocktäschel and Riedel, 2017)
   - Explicit multi-hop reasoning using neural networks

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)
   - Model-agnostic and fast (Minervini et al., 2017)
3. End-to-end differentiable proving (Rocktäschel and Riedel, 2017)
   - Explicit multi-hop reasoning using neural networks
   - Inducing rules using gradient descent

Outline

1. Link prediction & symbolic vs. neural representations
2. Regularize neural representations using logical rules
   - Model-agnostic but slow (Rocktäschel et al., 2015)
   - Fast but restricted (Demeester et al., 2016)
   - Model-agnostic and fast (Minervini et al., 2017)
3. End-to-end differentiable proving (Rocktäschel and Riedel, 2017)
   - Explicit multi-hop reasoning using neural networks
   - Inducing rules using gradient descent
4. Outlook & Summary

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)

Notation

- **Constant**: $\textsc{homer}$, $\textsc{bart}$, $\textsc{lisa}$ etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk
- **Predicate**: `fatherOf`, `parentOf` etc.
  function from terms to a Boolean

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk
- **Predicate**: `fatherOf`, `parentOf` etc.
  function from terms to a Boolean
- **Atom**: predicate and terms, e.g., `parentOf`(X, BART)

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk
- **Predicate**: `fatherOf`, `parentOf` etc.
  function from terms to a Boolean
- **Atom**: predicate and terms, e.g., `parentOf`(X, BART)
- **Literal**: atom or negated or atom, e.g.,
  not `parentOf`(BART, LISA)

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk
- **Predicate**: `fatherOf`, `parentOf` etc.
  function from terms to a Boolean
- **Atom**: predicate and terms, e.g., `parentOf`(X, BART)
- **Literal**: atom or negated or atom, e.g.,
  not `parentOf`(BART, LISA)
- **Rule**: `head` :– `body`.
  `head`: atom
  `body`: (possibly empty) list of literals representing conjunction
  Restricted to Horn clauses in this talk

Notation

- **Constant**: HOMER, BART, LISA etc. (lowercase)
- **Variable**: X, Y etc. (uppercase, universally quantified)
- **Term**: constant or variable
  Restricted to function-free terms in this talk
- **Predicate**: `fatherOf`, `parentOf` etc.
  function from terms to a Boolean
- **Atom**: predicate and terms, e.g., `parentOf`(X, BART)
- **Literal**: atom or negated or atom, e.g.,
  not `parentOf`(BART, LISA)
- **Rule**: `head` :– `body`.
  `head`: atom
  `body`: (possibly empty) list of literals representing conjunction
  Restricted to Horn clauses in this talk
- **Fact**: ground rule (no free variables) with empty body, e.g.,
  `parentOf`(HOMER, BART).

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

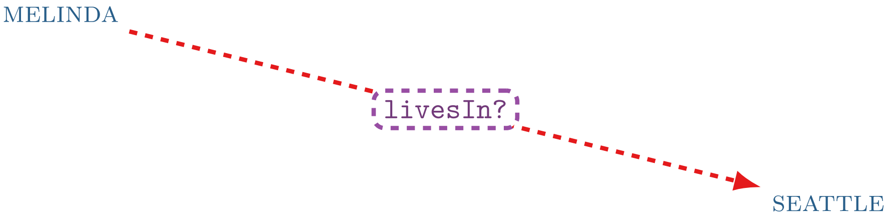
- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly
- Weak logical relationships that can be used for inferring facts

Das et al. (2017)

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly
- Weak logical relationships that can be used for inferring facts

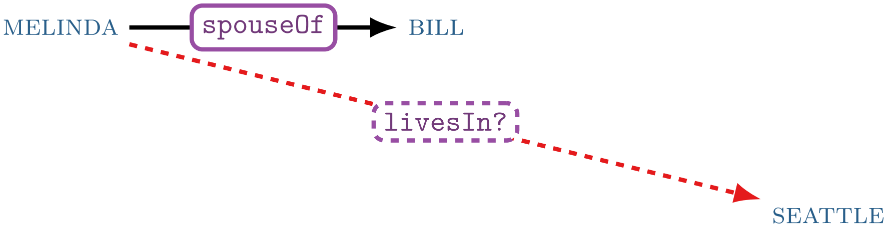


Das et al. (2017)

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly
- Weak logical relationships that can be used for inferring facts

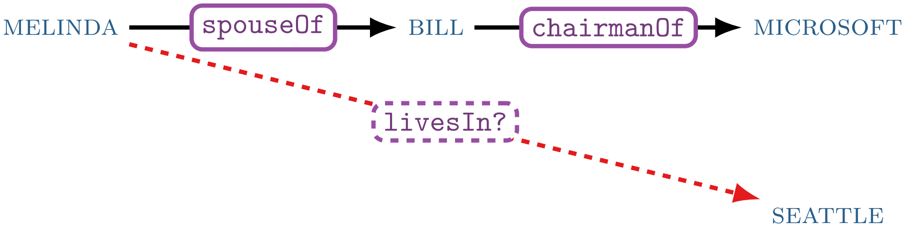


Das et al. (2017)

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly
- Weak logical relationships that can be used for inferring facts

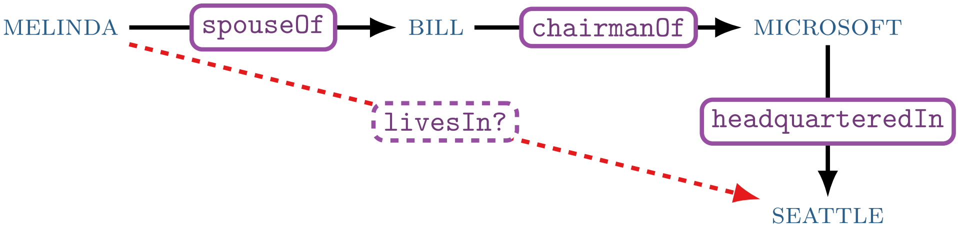


Das et al. (2017)

Link Prediction

Real world knowledge bases (like Freebase, DBPedia, YAGO, etc.) are incomplete!

- `placeOfBirth` attribute is missing for 71% of people!
- Commonsense knowledge often not stated explicitly
- Weak logical relationships that can be used for inferring facts



Symbolic Representations

- Symbols (constants and predicates) do not share any information:
  `grandpaOf` $\neq$ `grandfatherOf`

Symbolic Representations

- Symbols (constants and predicates) do not share any information: $\texttt{grandpaOf} \neq \texttt{grandfatherOf}$
- No notion of similarity: $\textsc{apple} \sim \textsc{orange}$, $\texttt{professorAt} \sim \texttt{lecturerAt}$

Symbolic Representations

- Symbols (constants and predicates) do not share any information: $\texttt{grandpaOf} \neq \texttt{grandfatherOf}$
- No notion of similarity: $\textsc{apple} \sim \textsc{orange}, \texttt{professorAt} \sim \texttt{lecturerAt}$
- No generalization beyond what can be symbolically inferred: $\texttt{isFruit}(\textsc{apple})$, $\textsc{apple} \sim \textsc{organge}$, $\texttt{isFruit}(\textsc{orange})$?

Symbolic Representations

- Symbols (constants and predicates) do not share any information:
  $\texttt{grandpaOf} \neq \texttt{grandfatherOf}$
- No notion of similarity: $\textsc{apple} \sim \textsc{orange}, \texttt{professorAt} \sim \texttt{lecturerAt}$
- No generalization beyond what can be symbolically inferred:
  $\texttt{isFruit}(\textsc{apple})$, $\textsc{apple} \sim \textsc{organge}$, $\texttt{isFruit}(\textsc{orange})$?
- Hard to work with language, vision and other modalities
  ``is a film based on the novel of the same name by''(X, Y)

Symbolic Representations

- Symbols (constants and predicates) do not share any information:
  $\texttt{grandpaOf} \neq \texttt{grandfatherOf}$
- No notion of similarity: $\textsc{apple} \sim \textsc{orange}$, $\texttt{professorAt} \sim \texttt{lecturerAt}$
- No generalization beyond what can be symbolically inferred:
  $\texttt{isFruit}(\textsc{apple})$, $\textsc{apple} \sim \textsc{organge}$, $\texttt{isFruit}(\textsc{orange})$?
- Hard to work with language, vision and other modalities
  $\texttt{``is a film based on the novel of the same name by''}(\mathrm{X}, \mathrm{Y})$
- But... leads to powerful inference mechanisms and proofs for predictions:
  $\texttt{fatherOf}(\textsc{abe}, \textsc{homer})$. $\texttt{parentOf}(\textsc{homer}, \textsc{lisa})$. $\texttt{parentOf}(\textsc{homer}, \textsc{bart})$.
  $\texttt{grandfatherOf}(\mathrm{X}, \mathrm{Y}) \mathrel{:\!-} \texttt{fatherOf}(\mathrm{X}, \mathrm{Z}), \texttt{parentOf}(\mathrm{Z}, \mathrm{Y})$.
  $\texttt{grandfatherOf}(\textsc{abe}, \mathrm{Q})$? $\quad \{\mathrm{Q}/\textsc{lisa}\}, \{\mathrm{Q}/\textsc{bart}\}$

Symbolic Representations

- Symbols (constants and predicates) do not share any information:
  $\texttt{grandpaOf} \neq \texttt{grandfatherOf}$
- No notion of similarity: $\textsc{apple} \sim \textsc{orange}, \texttt{professorAt} \sim \texttt{lecturerAt}$
- No generalization beyond what can be symbolically inferred:
  $\texttt{isFruit}(\textsc{apple}),\ \textsc{apple} \sim \textsc{organge},\ \texttt{isFruit}(\textsc{orange})?$
- Hard to work with language, vision and other modalities
  $\texttt{``is a film based on the novel of the same name by''}(X, Y)$
- But... leads to powerful inference mechanisms and proofs for predictions:
  $\texttt{fatherOf}(\textsc{abe}, \textsc{homer}).\ \texttt{parentOf}(\textsc{homer}, \textsc{lisa}).\ \texttt{parentOf}(\textsc{homer}, \textsc{bart}).$
  $\texttt{grandfatherOf}(X, Y) :- \texttt{fatherOf}(X, Z), \texttt{parentOf}(Z, Y).$
  $\texttt{grandfatherOf}(\textsc{abe}, Q)? \quad \{Q/\textsc{lisa}\}, \{Q/\textsc{bart}\}$
- Fairly easy to debug and trivial to incorporate domain knowledge:
  Show to domain expert and let her change/add rules and facts

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\boldsymbol{v}_{\text{APPLE}}, \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\mathbf{v}_{\text{APPLE}}, \mathbf{v}_{\text{ORANGE}}, \mathbf{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$
- Can capture similarity and even semantic hierarchy of symbols:
  $\mathbf{v}_{\texttt{grandpaOf}} = \mathbf{v}_{\texttt{grandfatherOf}}$,
  $\mathbf{v}_{\text{APPLE}} \sim \mathbf{v}_{\text{ORANGE}}, \mathbf{v}_{\text{APPLE}} < \mathbf{v}_{\text{FRUIT}}$

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\boldsymbol{v}_{\text{APPLE}}, \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$
- Can capture similarity and even semantic hierarchy of symbols:
  $\boldsymbol{v}_{\texttt{grandpaOf}} = \boldsymbol{v}_{\texttt{grandfatherOf}}$,
  $\boldsymbol{v}_{\text{APPLE}} \sim \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\text{APPLE}} < \boldsymbol{v}_{\text{FRUIT}}$
- Can be trained from raw task data (e.g. facts in a knowledge base)

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\boldsymbol{v}_{\text{APPLE}}, \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$
- Can capture similarity and even semantic hierarchy of symbols:
  $\boldsymbol{v}_{\texttt{grandpaOf}} = \boldsymbol{v}_{\texttt{grandfatherOf}}$,
  $\boldsymbol{v}_{\text{APPLE}} \sim \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\text{APPLE}} < \boldsymbol{v}_{\text{FRUIT}}$
- Can be trained from raw task data (e.g. facts in a knowledge base)
- Can be compositional
  $\boldsymbol{v}_{\texttt{``is the father of''}} = \text{RNN}_\theta(\boldsymbol{v}_{\texttt{is}}, \boldsymbol{v}_{\texttt{the}}, \boldsymbol{v}_{\texttt{father}}, \boldsymbol{v}_{\texttt{of}})$

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\boldsymbol{v}_{\text{APPLE}}, \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$
- Can capture similarity and even semantic hierarchy of symbols:
  $\boldsymbol{v}_{\texttt{grandpaOf}} = \boldsymbol{v}_{\texttt{grandfatherOf}},$
  $\boldsymbol{v}_{\text{APPLE}} \sim \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\text{APPLE}} < \boldsymbol{v}_{\text{FRUIT}}$
- Can be trained from raw task data (e.g. facts in a knowledge base)
- Can be compositional
  $\boldsymbol{v}_{\texttt{``is the father of''}} = \text{RNN}_\theta(\boldsymbol{v}_{\texttt{is}}, \boldsymbol{v}_{\texttt{the}}, \boldsymbol{v}_{\texttt{father}}, \boldsymbol{v}_{\texttt{of}})$
- But... need large amount of training data

Neural Representations

- Lower-dimensional fixed-length vector representations of symbols (predicates and constants):
  $\boldsymbol{v}_{\text{APPLE}}, \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\texttt{fatherOf}}, \ldots \in \mathbb{R}^k$
- Can capture similarity and even semantic hierarchy of symbols:
  $\boldsymbol{v}_{\texttt{grandpaOf}} = \boldsymbol{v}_{\texttt{grandfatherOf}}$,
  $\boldsymbol{v}_{\text{APPLE}} \sim \boldsymbol{v}_{\text{ORANGE}}, \boldsymbol{v}_{\text{APPLE}} < \boldsymbol{v}_{\text{FRUIT}}$
- Can be trained from raw task data (e.g. facts in a knowledge base)
- Can be compositional
  $\boldsymbol{v}_{\texttt{``is the father of''}} = \text{RNN}_\theta(\boldsymbol{v}_{\texttt{is}}, \boldsymbol{v}_{\texttt{the}}, \boldsymbol{v}_{\texttt{father}}, \boldsymbol{v}_{\texttt{of}})$
- But... need large amount of training data
- No direct way of incorporating prior knowledge
  $\boldsymbol{v}_{\texttt{grandfatherOf}}(\text{X}, \text{Y}) :\!- \boldsymbol{v}_{\texttt{fatherOf}}(\text{X}, \text{Z}), \boldsymbol{v}_{\texttt{parentOf}}(\text{Z}, \text{Y}).$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_{\boldsymbol{\theta}}(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_{\boldsymbol{\theta}}(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top (\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk} \boldsymbol{v}_{ik} \boldsymbol{v}_{jk} \end{aligned}$$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top (\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk} \boldsymbol{v}_{ik} \boldsymbol{v}_{jk} \end{aligned}$$

ComplEx (Trouillon et al., 2016)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{C}^k$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top (\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk} \boldsymbol{v}_{ik} \boldsymbol{v}_{jk} \end{aligned}$$

ComplEx (Trouillon et al., 2016)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{C}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) = \; & \\ & \mathrm{real}(\boldsymbol{v}_s)^\top (\mathrm{real}(\boldsymbol{v}_i) \odot \mathrm{real}(\boldsymbol{v}_j)) \\ & + \mathrm{real}(\boldsymbol{v}_s)^\top (\mathrm{imag}(\boldsymbol{v}_i) \odot \mathrm{imag}(\boldsymbol{v}_j)) \\ & + \mathrm{imag}(\boldsymbol{v}_s)^\top (\mathrm{real}(\boldsymbol{v}_i) \odot \mathrm{imag}(\boldsymbol{v}_j)) \\ & - \mathrm{imag}(\boldsymbol{v}_s)^\top (\mathrm{imag}(\boldsymbol{v}_i) \odot \mathrm{real}(\boldsymbol{v}_j)) \end{aligned}$$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top(\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk}\boldsymbol{v}_{ik}\boldsymbol{v}_{jk} \end{aligned}$$

ComplEx (Trouillon et al., 2016)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{C}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) = \\ &\text{real}(\boldsymbol{v}_s)^\top(\text{real}(\boldsymbol{v}_i) \odot \text{real}(\boldsymbol{v}_j)) \\ &+ \text{real}(\boldsymbol{v}_s)^\top(\text{imag}(\boldsymbol{v}_i) \odot \text{imag}(\boldsymbol{v}_j)) \\ &+ \text{imag}(\boldsymbol{v}_s)^\top(\text{real}(\boldsymbol{v}_i) \odot \text{imag}(\boldsymbol{v}_j)) \\ &- \text{imag}(\boldsymbol{v}_s)^\top(\text{imag}(\boldsymbol{v}_i) \odot \text{real}(\boldsymbol{v}_j)) \end{aligned}$$

Training Loss

$$\mathfrak{L} = \sum_{r_s(e_i, e_j), y \in \mathcal{T}} -y \log\left(\sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right) - (1-y)\log\left(1 - \sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right)$$

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\textsc{Melinda}, \textsc{Seattle})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\textsc{Melinda}}, \boldsymbol{v}_{\textsc{Seattle}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top(\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk} \boldsymbol{v}_{ik} \boldsymbol{v}_{jk} \end{aligned}$$

ComplEx (Trouillon et al., 2016)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{C}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) = \\ &\mathrm{real}(\boldsymbol{v}_s)^\top(\mathrm{real}(\boldsymbol{v}_i) \odot \mathrm{real}(\boldsymbol{v}_j)) \\ &+ \mathrm{real}(\boldsymbol{v}_s)^\top(\mathrm{imag}(\boldsymbol{v}_i) \odot \mathrm{imag}(\boldsymbol{v}_j)) \\ &+ \mathrm{imag}(\boldsymbol{v}_s)^\top(\mathrm{real}(\boldsymbol{v}_i) \odot \mathrm{imag}(\boldsymbol{v}_j)) \\ &- \mathrm{imag}(\boldsymbol{v}_s)^\top(\mathrm{imag}(\boldsymbol{v}_i) \odot \mathrm{real}(\boldsymbol{v}_j)) \end{aligned}$$

Training Loss

$$\mathfrak{L} = \sum_{r_s(e_i, e_j), y \, \in \, \mathcal{T}} -y \log\left(\sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right) - (1-y) \log\left(1 - \sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right)$$

- Learn $\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j$ from data

State-of-the-art Neural Link Prediction

$$\texttt{livesIn}(\text{MELINDA}, \text{SEATTLE})? = f_\theta(\boldsymbol{v}_{\texttt{livesIn}}, \boldsymbol{v}_{\text{MELINDA}}, \boldsymbol{v}_{\text{SEATTLE}})$$

DistMult (Yang et al., 2015)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{R}^k$

$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) &= \boldsymbol{v}_s^\top(\boldsymbol{v}_i \odot \boldsymbol{v}_j) \\ &= \sum_k \boldsymbol{v}_{sk} \boldsymbol{v}_{ik} \boldsymbol{v}_{jk} \end{aligned}$$

ComplEx (Trouillon et al., 2016)

$\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j \in \mathbb{C}^k$

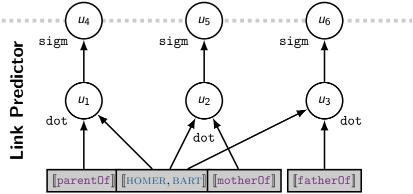
$$\begin{aligned} f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j) = \\ &\text{real}(\boldsymbol{v}_s)^\top(\text{real}(\boldsymbol{v}_i) \odot \text{real}(\boldsymbol{v}_j)) \\ &+ \text{real}(\boldsymbol{v}_s)^\top(\text{imag}(\boldsymbol{v}_i) \odot \text{imag}(\boldsymbol{v}_j)) \\ &+ \text{imag}(\boldsymbol{v}_s)^\top(\text{real}(\boldsymbol{v}_i) \odot \text{imag}(\boldsymbol{v}_j)) \\ &- \text{imag}(\boldsymbol{v}_s)^\top(\text{imag}(\boldsymbol{v}_i) \odot \text{real}(\boldsymbol{v}_j)) \end{aligned}$$

Training Loss

$$\mathfrak{L} = \sum_{r_s(e_i, e_j), y \,\in\, \mathcal{T}} -y \log\left(\sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right) - (1-y)\log\left(1 - \sigma(f_\theta(\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j))\right)$$

- Learn $\boldsymbol{v}_s, \boldsymbol{v}_i, \boldsymbol{v}_j$ from data
- Obtain gradients $\nabla_{\boldsymbol{v}_s}\mathfrak{L}$, $\nabla_{\boldsymbol{v}_i}\mathfrak{L}$, $\nabla_{\boldsymbol{v}_j}\mathfrak{L}$ by backprop

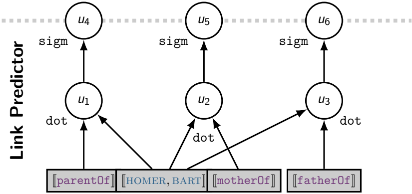
Regularization by Propositional Logic



Rocktäschel et al. (2015), NAACL

Regularization by Propositional Logic

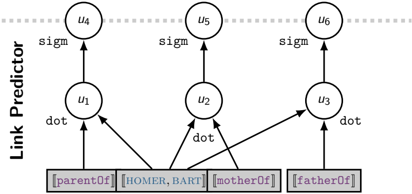
$\texttt{fatherOf}(X, Y) :- \texttt{parentOf}(X, Y), \neg\, \texttt{motherOf}(X, Y)$



Regularization by Propositional Logic

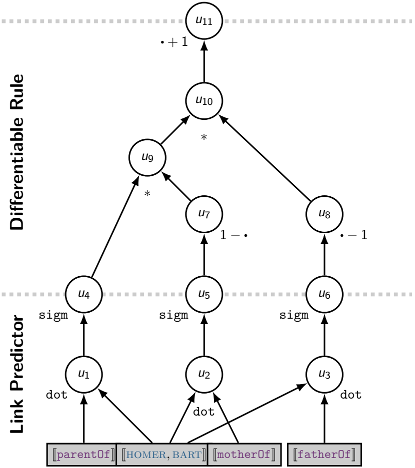
$$\texttt{fatherOf}(X, Y) \;{:}\!-\; \texttt{parentOf}(X, Y), \neg\, \texttt{motherOf}(X, Y)$$

$$p(F) = [\![F]\!] = \begin{cases} f_\theta(s, i, j) & \text{if } F = s(i, j) \\ 1 - [\![A]\!] & \text{if } F = \neg\, A \\ [\![A]\!]\,[\![B]\!] & \text{if } F = A \wedge B \\ [\![A]\!] + [\![B]\!] - [\![A]\!]\,[\![B]\!] & \text{if } F = A \vee B \\ [\![B]\!]\,([\![A]\!] - 1) + 1 & \text{if } F = A \;{:}\!-\; B \end{cases}$$



Rocktäschel et al. (2015), NAACL

Regularization by Propositional Logic

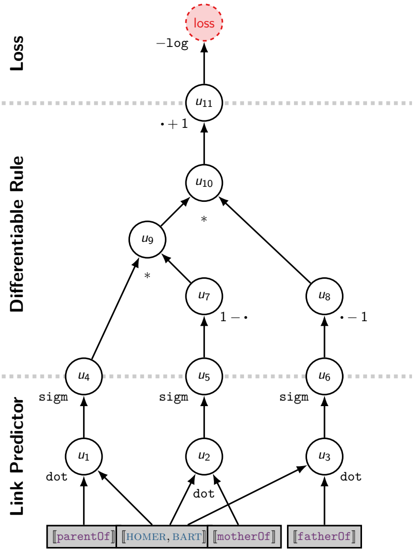


$$\texttt{fatherOf}(X, Y) \text{ :– } \texttt{parentOf}(X, Y), \neg\, \texttt{motherOf}(X, Y)$$

$$p(F) = [\![F]\!] = \begin{cases} f_\theta(s, i, j) & \text{if } F = s(i, j) \\ 1 - [\![A]\!] & \text{if } F = \neg\, A \\ [\![A]\!]\,[\![B]\!] & \text{if } F = A \wedge B \\ [\![A]\!] + [\![B]\!] - [\![A]\!]\,[\![B]\!] & \text{if } F = A \vee B \\ [\![B]\!]\,([\![A]\!] - 1) + 1 & \text{if } F = A \text{ :– } B \end{cases}$$

Rocktäschel et al. (2015), NAACL

Regularization by Propositional Logic



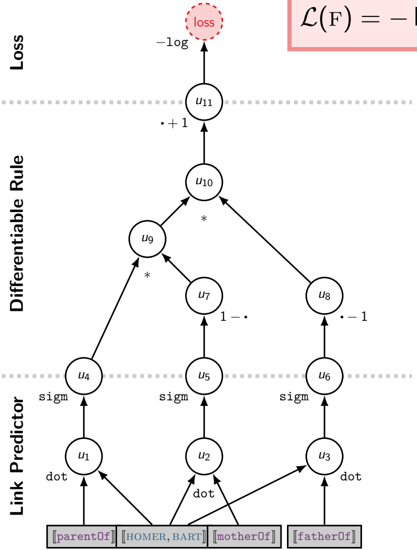
$$\mathtt{fatherOf}(\mathrm{X},\mathrm{Y}) \text{ :– } \mathtt{parentOf}(\mathrm{X},\mathrm{Y}), \neg\,\mathtt{motherOf}(\mathrm{X},\mathrm{Y})$$

$$p(F) = [\![F]\!] = \begin{cases} f_\theta(s,i,j) & \text{if } F = s(i,j) \\ 1 - [\![A]\!] & \text{if } F = \neg A \\ [\![A]\!]\,[\![B]\!] & \text{if } F = A \wedge B \\ [\![A]\!] + [\![B]\!] - [\![A]\!]\,[\![B]\!] & \text{if } F = A \vee B \\ [\![B]\!]\,([\![A]\!] - 1) + 1 & \text{if } F = A \text{ :– } B \end{cases}$$

$$\mathcal{L}\Big([\![\mathtt{fatherOf}(\mathrm{HOMER},\mathrm{BART}) \text{ :– } \mathtt{parentOf}(\mathrm{HOMER},\mathrm{BART}) \wedge \neg\,\mathtt{motherOf}(\mathrm{HOMER},\mathrm{BART})]\!]\Big)$$

Rocktäschel et al. (2015), NAACL

Regularization by Propositional Logic



$$\boxed{\mathcal{L}(\mathrm{F}) = -\log\left(\llbracket \forall \mathrm{X}, \mathrm{Y} : \mathrm{F}(\mathrm{X}, \mathrm{Y}) \rrbracket\right) = -\sum_{(e_i, e_j) \in \mathcal{C}^2} \log \llbracket \mathrm{F}(e_i, e_j) \rrbracket}$$

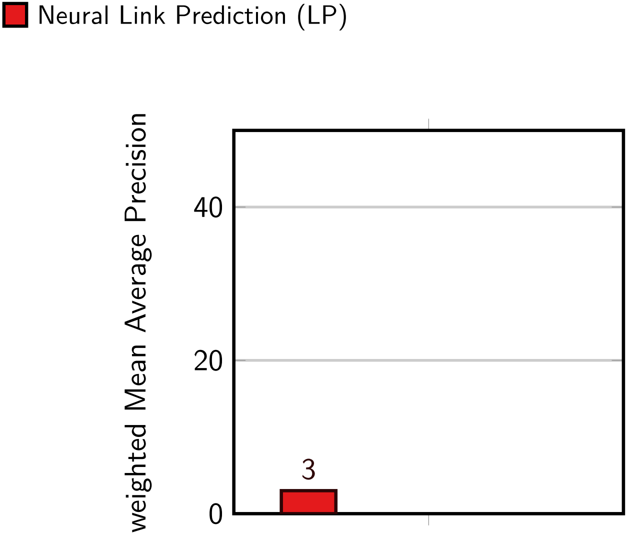
$$\texttt{fatherOf}(\mathrm{X}, \mathrm{Y}) :\!- \texttt{parentOf}(\mathrm{X}, \mathrm{Y}), \neg\, \texttt{motherOf}(\mathrm{X}, \mathrm{Y})$$

$$p(F) = \llbracket F \rrbracket = \begin{cases} f_\theta(s, i, j) & \text{if } F = s(i, j) \\ 1 - \llbracket A \rrbracket & \text{if } F = \neg A \\ \llbracket A \rrbracket \llbracket B \rrbracket & \text{if } F = A \wedge B \\ \llbracket A \rrbracket + \llbracket B \rrbracket - \llbracket A \rrbracket \llbracket B \rrbracket & \text{if } F = A \vee B \\ \llbracket B \rrbracket (\llbracket A \rrbracket - 1) + 1 & \text{if } F = A :\!- B \end{cases}$$

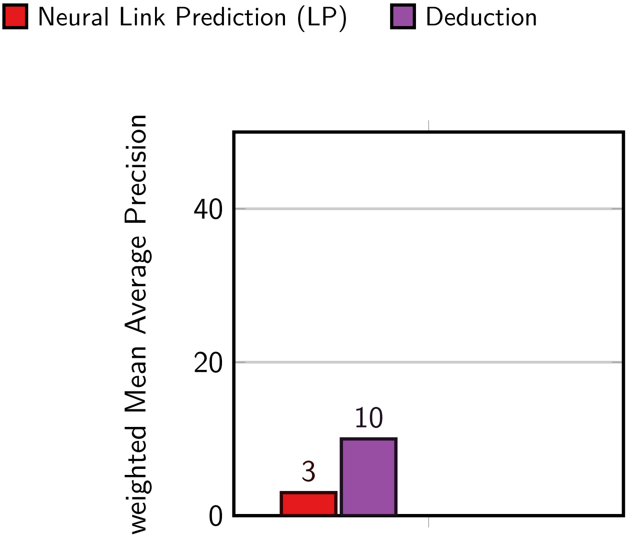
$$\mathcal{L}\Big(\llbracket \texttt{fatherOf}(\mathrm{HOMER}, \mathrm{BART}) :\!- \texttt{parentOf}(\mathrm{HOMER}, \mathrm{BART}) \wedge \neg\, \texttt{motherOf}(\mathrm{HOMER}, \mathrm{BART}) \rrbracket\Big)$$

Rocktäschel et al. (2015), NAACL

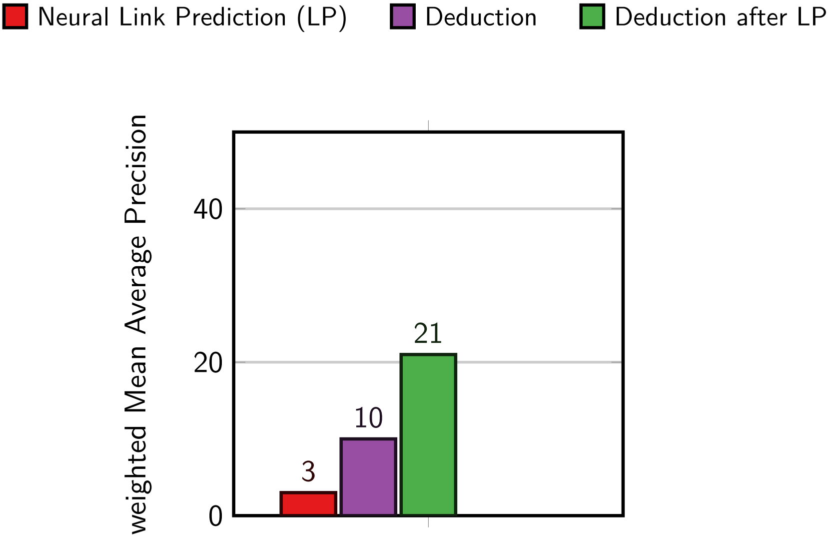
Zero-shot Learning Results



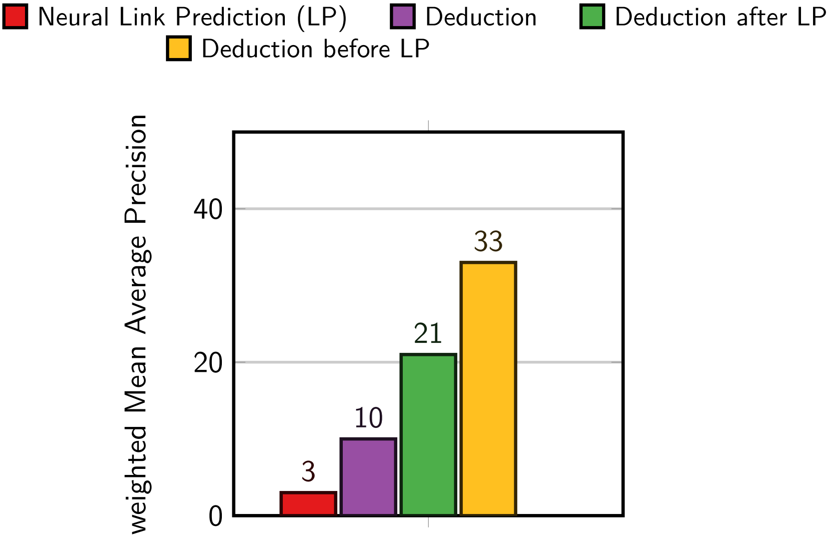
Zero-shot Learning Results



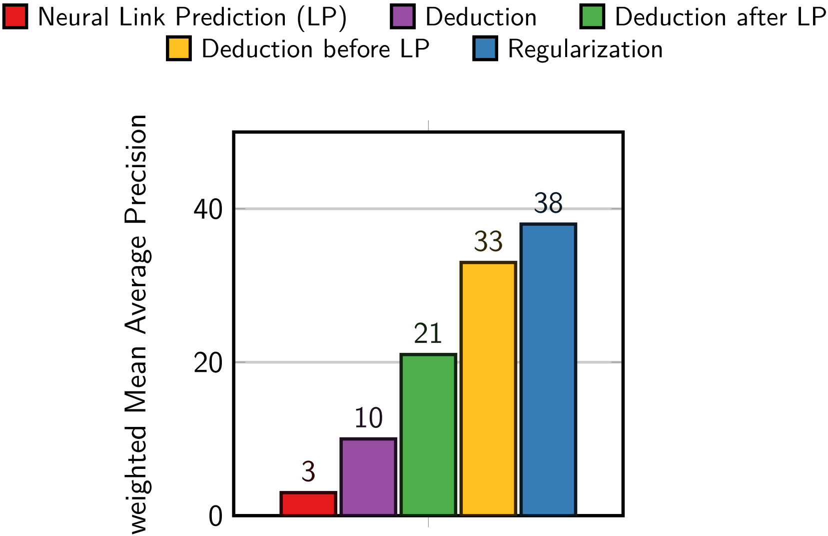
Zero-shot Learning Results



Zero-shot Learning Results

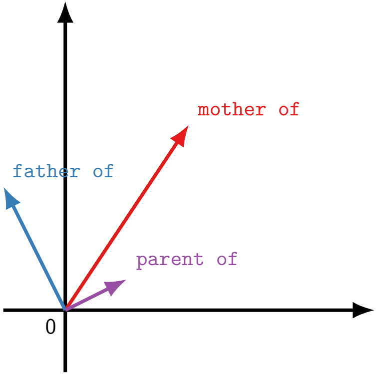


Zero-shot Learning Results



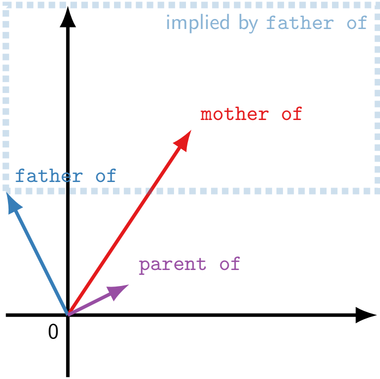
Lifted Regularization by Implications

Every father is a parent
Every mother is a parent



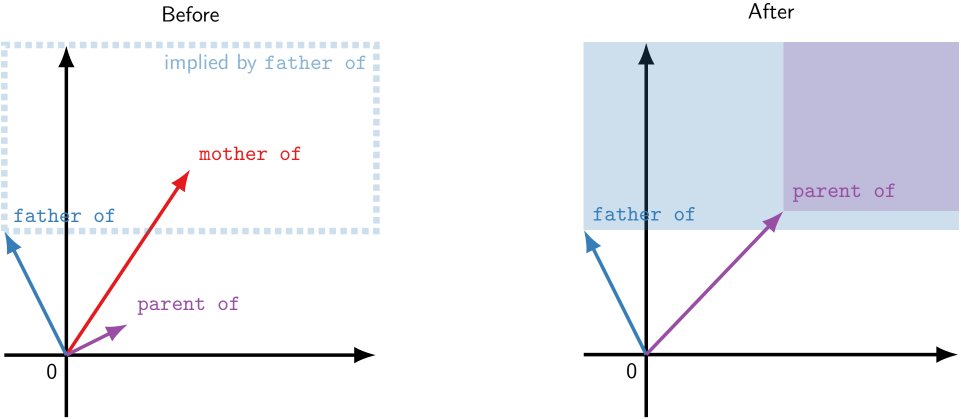
Lifted Regularization by Implications

Every father is a parent
Every mother is a parent



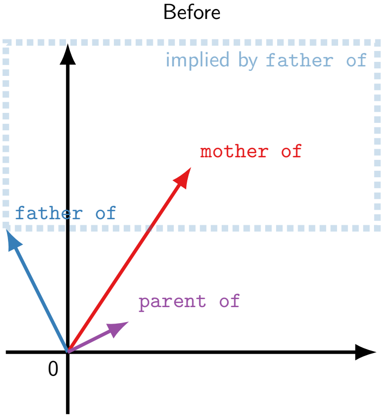
Lifted Regularization by Implications

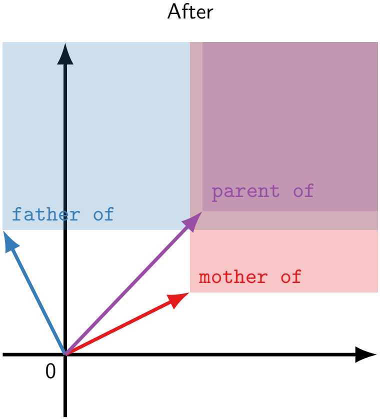
Every father is a parent
Every mother is a parent



Lifted Regularization by Implications

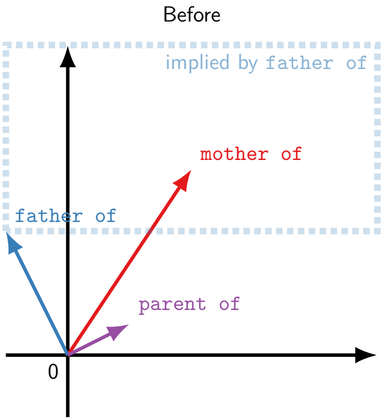
Every father is a parent
Every mother is a parent

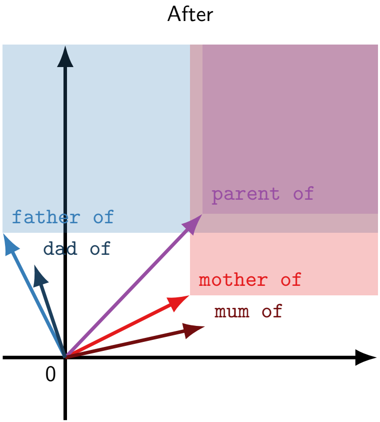




Lifted Regularization by Implications

Every father is a parent Generalises to similar relations (*e.g.* dad)
Every mother is a parent Generalises to similar relations (*e.g.* mum)



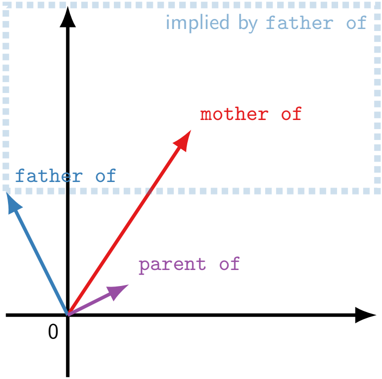


Demeester et al. (2016), EMNLP

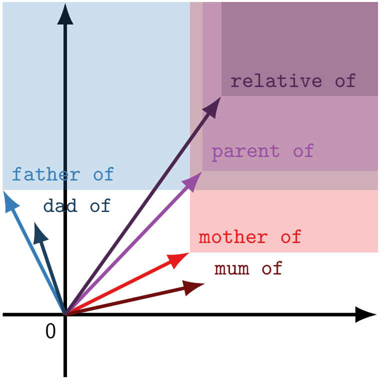
Lifted Regularization by Implications

Every father is a parent	Generalises to similar relations (*e.g.* dad)
Every mother is a parent	Generalises to similar relations (*e.g.* mum)
Every parent is a relative	**No training facts needed!**

Before



After

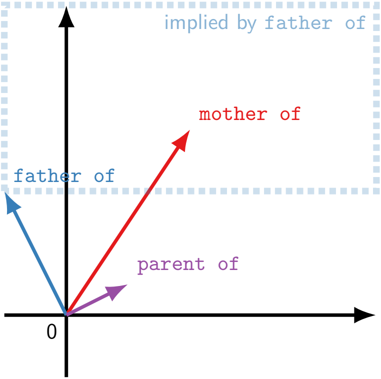


Lifted Regularization by Implications

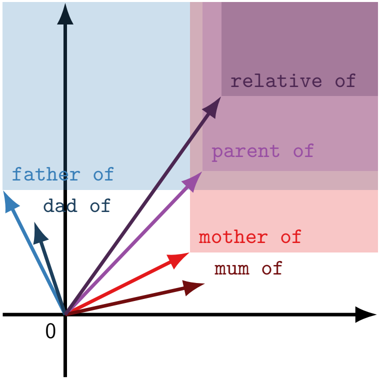
Every father is a parent
Every mother is a parent
Every parent is a relative

$$\forall \mathrm{X}, \mathrm{Y} \;:\; \mathtt{h}(\mathrm{X}, \mathrm{Y}) \coloneq \mathtt{b}(\mathrm{X}, \mathrm{Y})$$
$$\forall (e_i, e_j) \in \mathcal{C}^2 : [\![\mathtt{h}]\!]^\top [\![e_i, e_j]\!] \geq [\![\mathtt{b}]\!]^\top [\![e_i, e_j]\!]$$
$$[\![\mathtt{h}]\!] \geq [\![\mathtt{b}]\!], \quad \forall (e_i, e_j) \in \mathcal{C}^2 : [\![e_i, e_j]\!] \in \mathbb{R}_+^k$$

Before

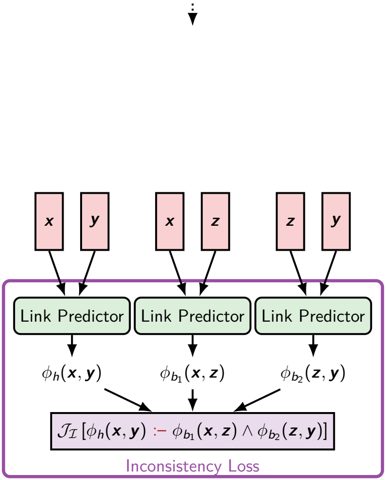


After



Adversarial Regularization

Clause $\mathcal{A}$: $h(\mathrm{X}, \mathrm{Y}) \coloneq b_1(\mathrm{X}, \mathrm{Z}) \wedge b_2(\mathrm{Z}, \mathrm{Y})$

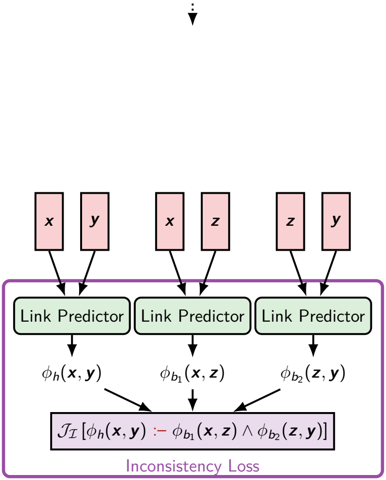


- Regularization by propositional rules needs grounding – does not scale to large domains!

Minervini et al. (2017), UAI

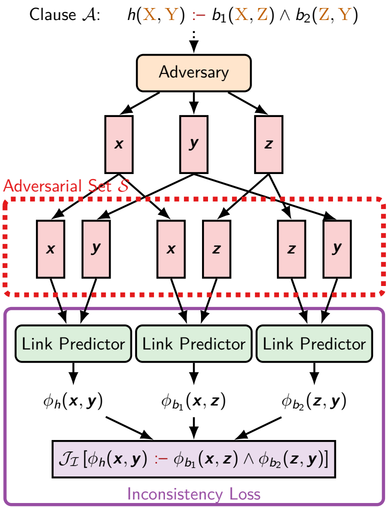
Adversarial Regularization

Clause $\mathcal{A}$: $h(\mathrm{X}, \mathrm{Y}) :- b_1(\mathrm{X}, \mathrm{Z}) \wedge b_2(\mathrm{Z}, \mathrm{Y})$



- Regularization by propositional rules needs grounding – does not scale to large domains!
- Lifted regularization only supports direct implications

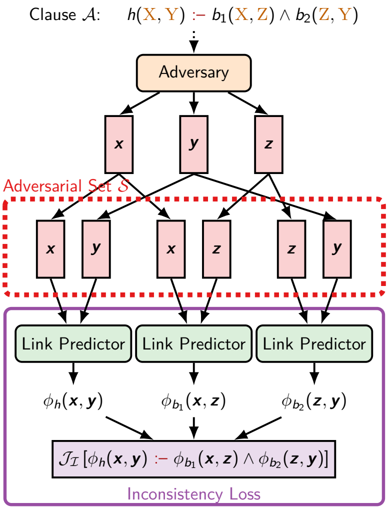
Adversarial Regularization



- Regularization by propositional rules needs grounding – does not scale to large domains!
- Lifted regularization only supports direct implications
- Idea: let grounding be generated by an adversary and optimize minimax game...

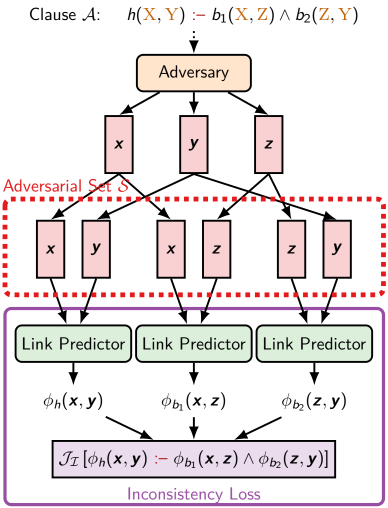
Minervini et al. (2017), UAI

Adversarial Regularization



- Regularization by propositional rules needs grounding – does not scale to large domains!
- Lifted regularization only supports direct implications
- Idea: let grounding be generated by an adversary and optimize minimax game...
- Adversary finds maximally violating grounding for a given rule

Adversarial Regularization



- Regularization by propositional rules needs grounding – does not scale to large domains!
- Lifted regularization only supports direct implications
- Idea: let grounding be generated by an adversary and optimize minimax game...
- Adversary finds maximally violating grounding for a given rule
- Neural link predictor attempts to minimize rule violation for given generated groundings

Minervini et al. (2017), UAI

End-to-End Differentiable Prover

- Neural network for proving queries to a knowledge base

End-to-End Differentiable Prover

- Neural network for proving queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols

End-to-End Differentiable Prover

- Neural network for proving queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Learn vector representations of symbols** end-to-end from proof success

Rocktäschel and Riedel (2017), NIPS

End-to-End Differentiable Prover

- Neural network for proving queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Learn vector representations of symbols** end-to-end from proof success
- **Make use of provided rules** in soft proofs

End-to-End Differentiable Prover

- Neural network for proving queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Learn vector representations of symbols** end-to-end from proof success
- **Make use of provided rules** in soft proofs
- **Induce interpretable rules** end-to-end from proof success

Rocktäschel and Riedel (2017), NIPS

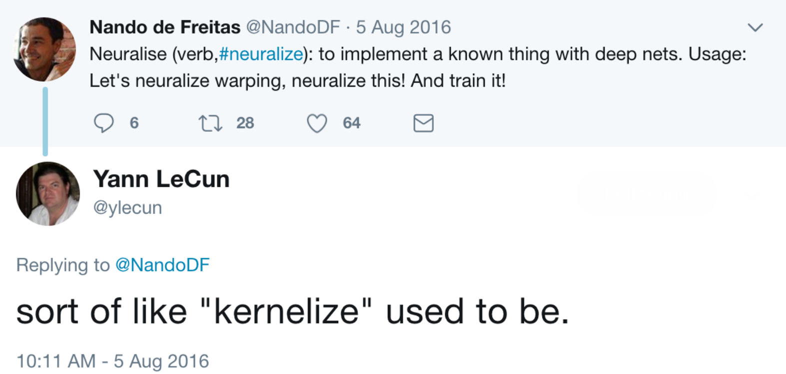
Approach


Nando de Freitas @NandoDF · 5 Aug 2016

Neuralise (verb,#neuralize): to implement a known thing with deep nets. Usage: Let's neuralize warping, neuralize this! And train it!

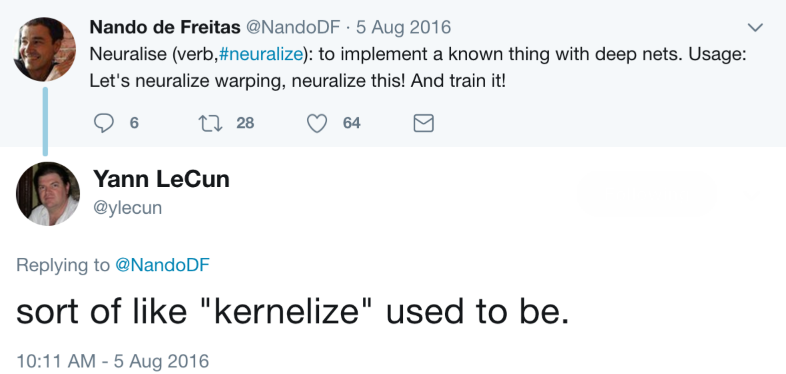
6 28 64

Approach



Nando de Freitas @NandoDF · 5 Aug 2016
Neuralise (verb,#neuralize): to implement a known thing with deep nets. Usage: Let's neuralize warping, neuralize this! And train it!
6
28
64
Yann LeCun
@ylecun
Replying to @NandoDF
sort of like "kernelize" used to be.
10:11 AM - 5 Aug 2016

Approach



Let's **neuralize** Prolog's Backward Chaining using a Radial Basis Function **kernel** for unifying vector representations of symbols!

Prolog's Backward Chaining

Example Knowledge Base:

1. `fatherOf`(ABE, HOMER).
2. `parentOf`(HOMER, BART).
3. `grandfatherOf`(X, Y) :–
   `fatherOf`(X, Z),
   `parentOf`(Z, Y).

Prolog's Backward Chaining

Example Knowledge Base:

1. $\texttt{fatherOf}(\textsc{abe}, \textsc{homer})$.
2. $\texttt{parentOf}(\textsc{homer}, \textsc{bart})$.
3. $\texttt{grandfatherOf}(X, Y)$:–
   $\texttt{fatherOf}(X, Z)$,
   $\texttt{parentOf}(Z, Y)$.

Intuition:

- Backward chaining translates a query into subqueries via rules, *e.g.*,
  $\texttt{grandfatherOf}(\textsc{abe}, \textsc{bart}) \xrightarrow{3.} \texttt{fatherOf}(\textsc{abe}, Z), \texttt{parentOf}(Z, \textsc{bart})$

Prolog's Backward Chaining

Example Knowledge Base:

1. $\texttt{fatherOf}(\text{ABE}, \text{HOMER})$.
2. $\texttt{parentOf}(\text{HOMER}, \text{BART})$.
3. $\texttt{grandfatherOf}(X, Y)$:–
   $\texttt{fatherOf}(X, Z)$,
   $\texttt{parentOf}(Z, Y)$.

Intuition:

- Backward chaining translates a query into subqueries via rules, *e.g.*,
  $\texttt{grandfatherOf}(\text{ABE}, \text{BART}) \xrightarrow{\boxed{3.}} \texttt{fatherOf}(\text{ABE}, Z), \texttt{parentOf}(Z, \text{BART})$
- It attempts this for all rules in the knowledge base and thus specifies a depth-first search

Unification

Example Knowledge Base:

1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).

Query

grandfatherOf ABE BART

Unification

Example Knowledge Base:

1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).

Query

grandfatherOf	ABE	BART

1.

fatherOf	ABE	HOMER

Unification

Example Knowledge Base:

1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).

Query

grandfatherOf	ABE	BART
$\stackrel{?}{=}$	$\stackrel{?}{=}$	$\stackrel{?}{=}$
fatherOf	ABE	HOMER

1.

Unification

Example Knowledge Base:

1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).

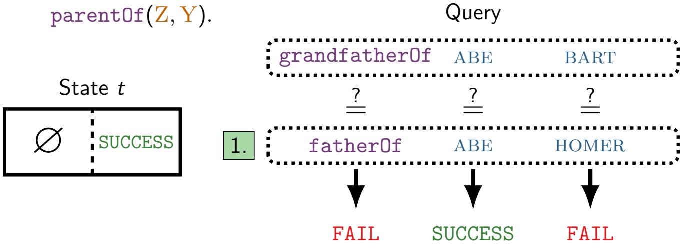
Query

	grandfatherOf	ABE	BART
	$\stackrel{?}{=}$	$\stackrel{?}{=}$	$\stackrel{?}{=}$
1.	fatherOf	ABE	HOMER
	FAIL	SUCCESS	FAIL

Unification

Example Knowledge Base:

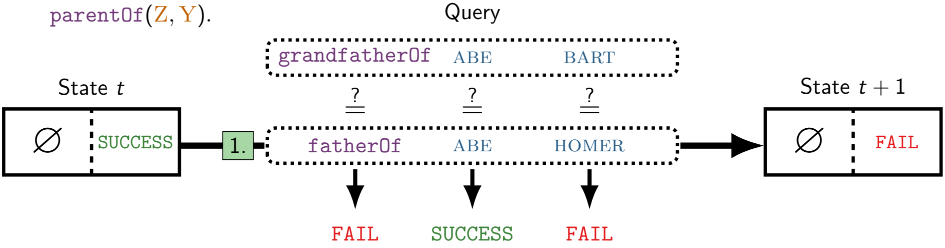
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Unification

Example Knowledge Base:

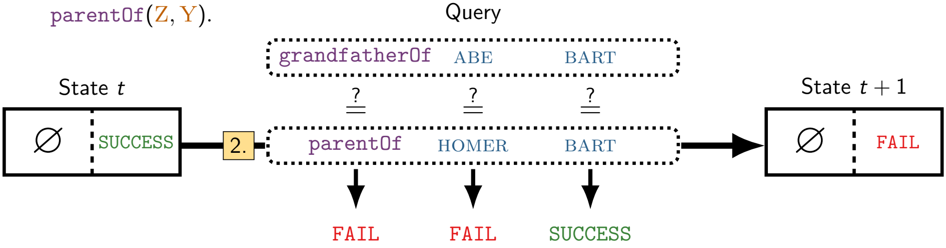
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Unification

Example Knowledge Base:

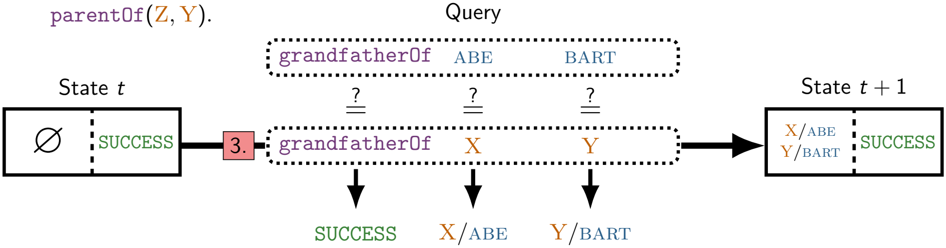
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Unification

Example Knowledge Base:

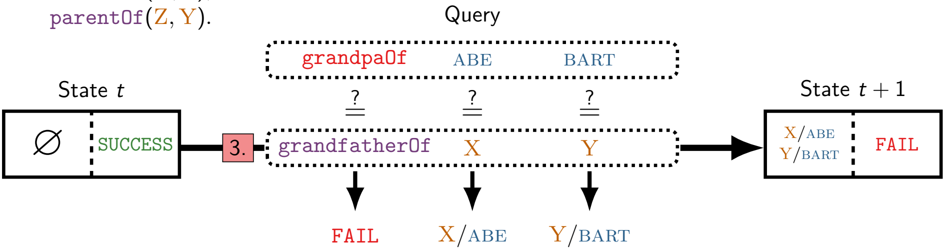
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
    fatherOf(X, Z),
    parentOf(Z, Y).



Unification Failure

Example Knowledge Base:

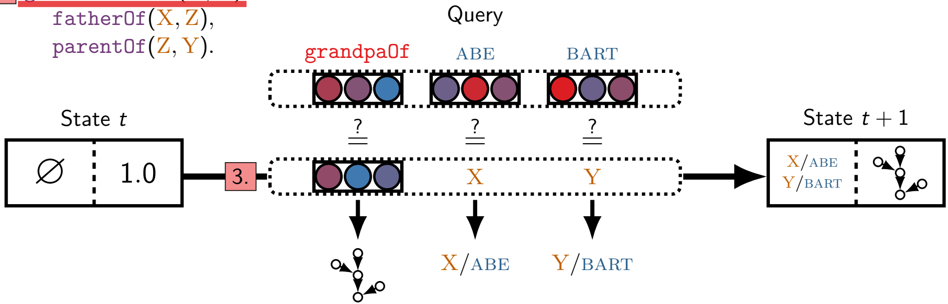
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Neural Unification

Example Knowledge Base:

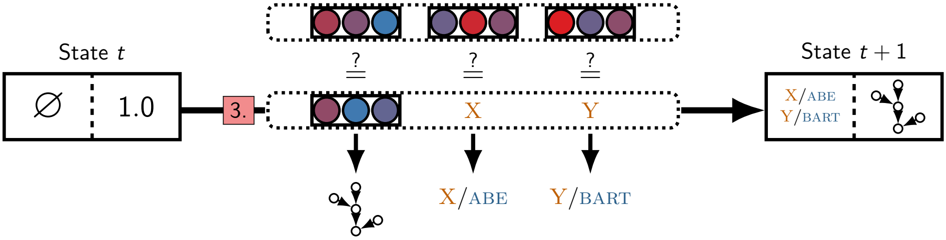
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Neural Unification

Example Knowledge Base:

1. `fatherOf`(ABE, HOMER).
2. `parentOf`(HOMER, BART).
3. `grandfatherOf`(X, Y) :–
   `fatherOf`(X, Z),
   `parentOf`(Z, Y).

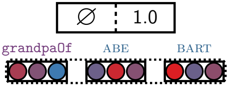


$$\min\left(1.0, \exp\left(\frac{-\|\mathbf{v}_{\texttt{grandpaOf}} - \mathbf{v}_{\texttt{grandfatherOf}}\|_2}{2\mu^2}\right)\right)$$

Differentiable Prover

Example Knowledge Base:

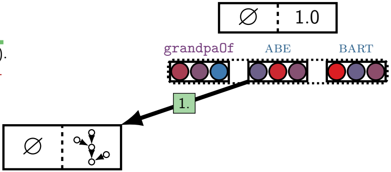
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
    fatherOf(X, Z),
    parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

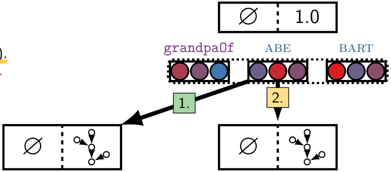
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

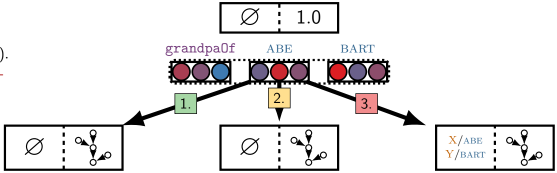
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).

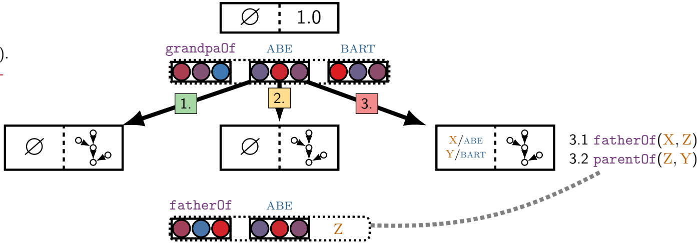


3.1 fatherOf(X, Z)
3.2 parentOf(Z, Y)

Differentiable Prover

Example Knowledge Base:

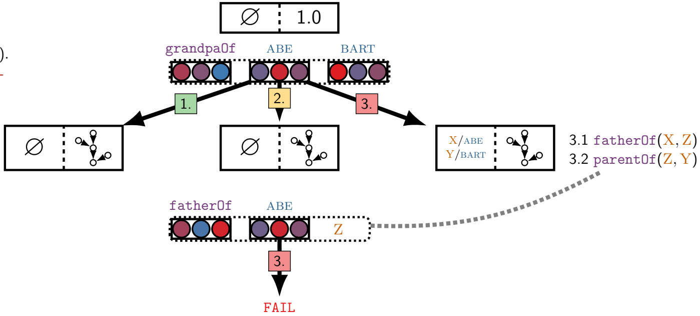
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
    fatherOf(X, Z),
    parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

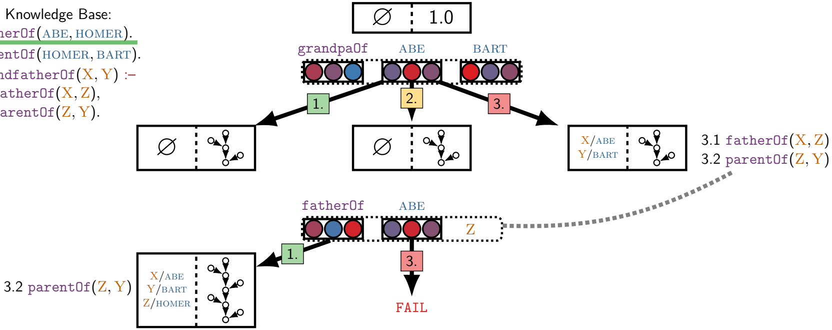
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
    fatherOf(X, Z),
    parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

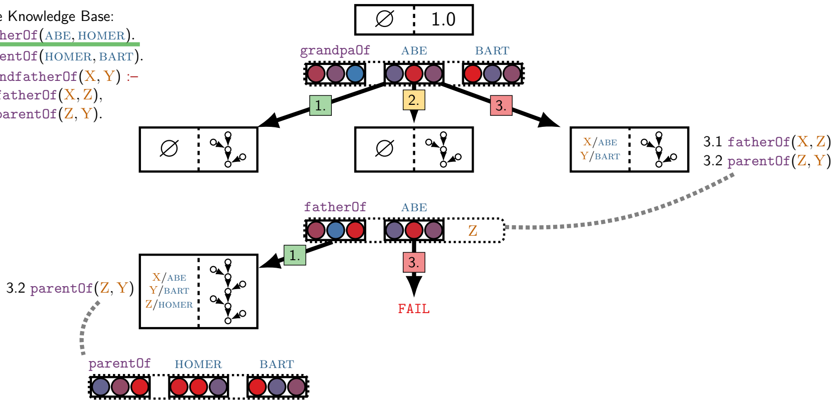
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

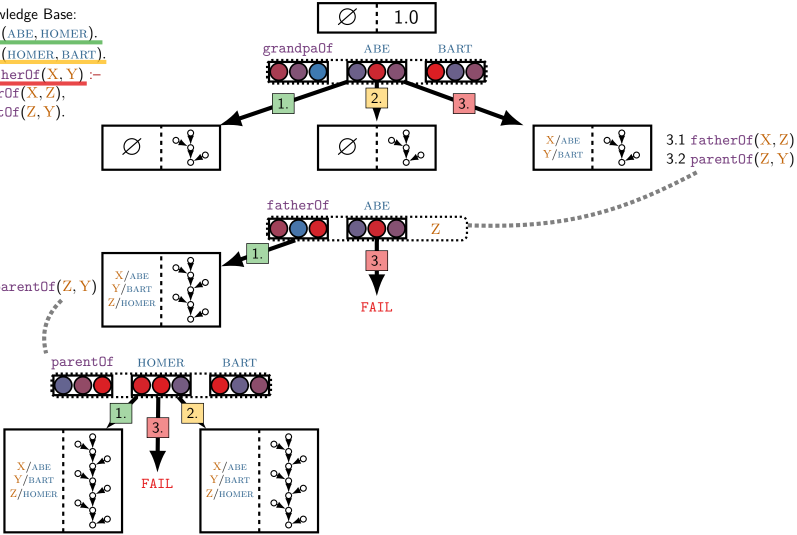
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

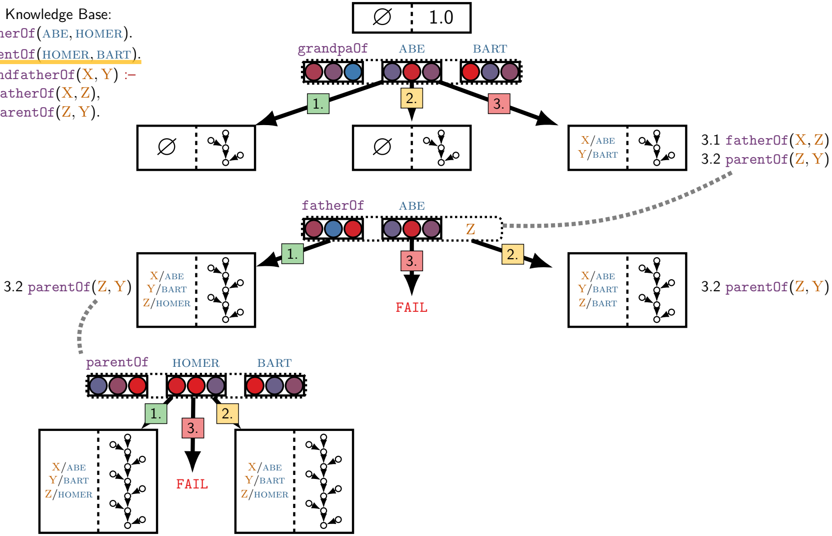
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

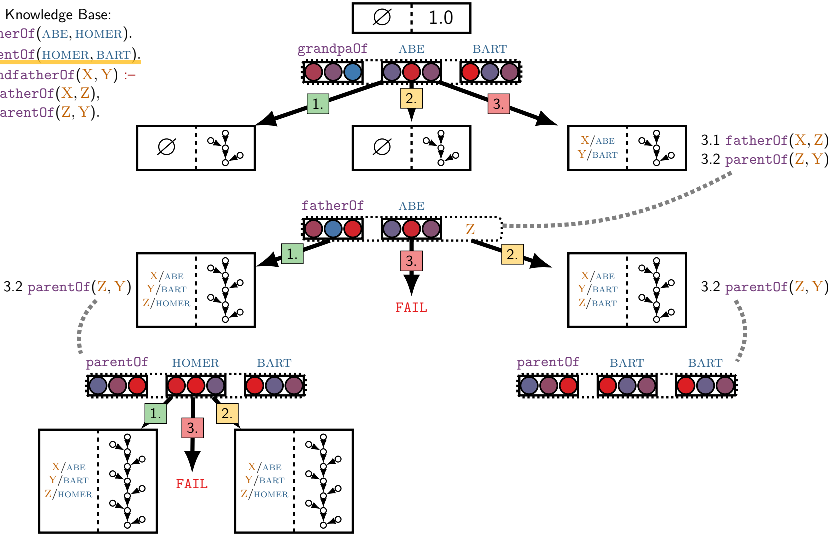
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

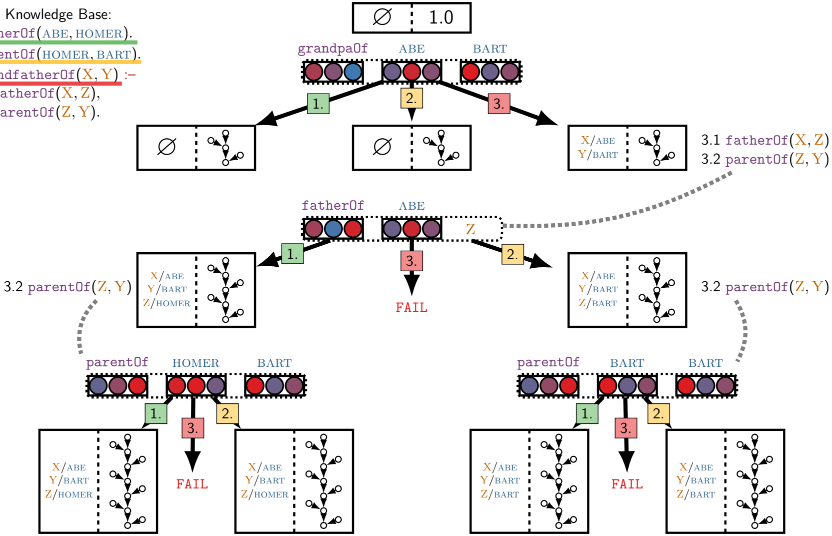
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Differentiable Prover

Example Knowledge Base:

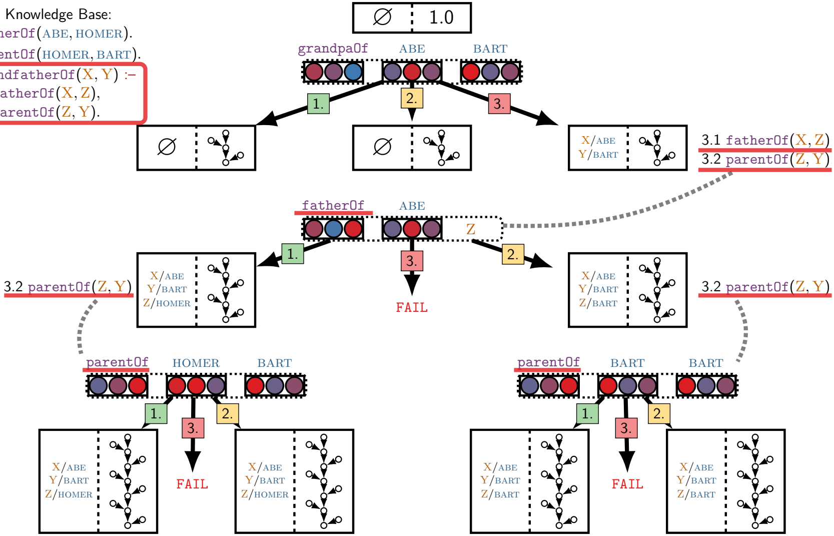
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
   fatherOf(X, Z),
   parentOf(Z, Y).



Neural Program Induction

Example Knowledge Base:

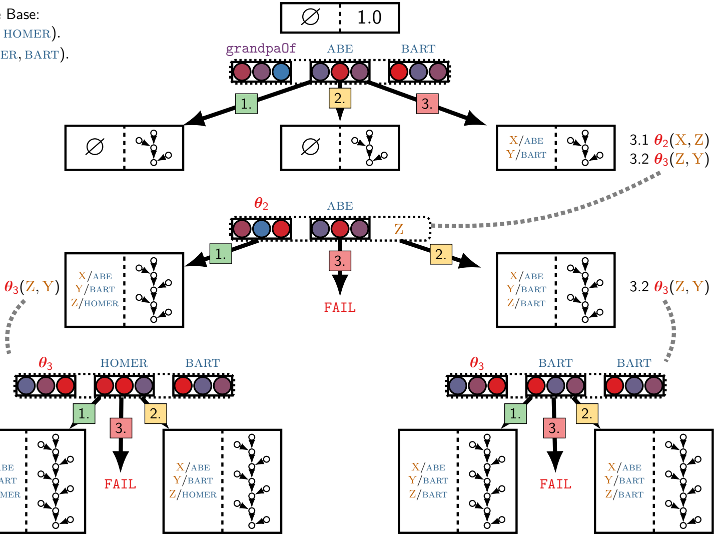
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. grandfatherOf(X, Y) :–
    fatherOf(X, Z),
    parentOf(Z, Y).



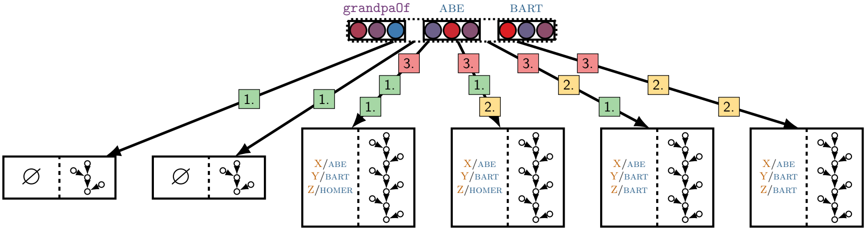
Neural Program Induction

Example Knowledge Base:

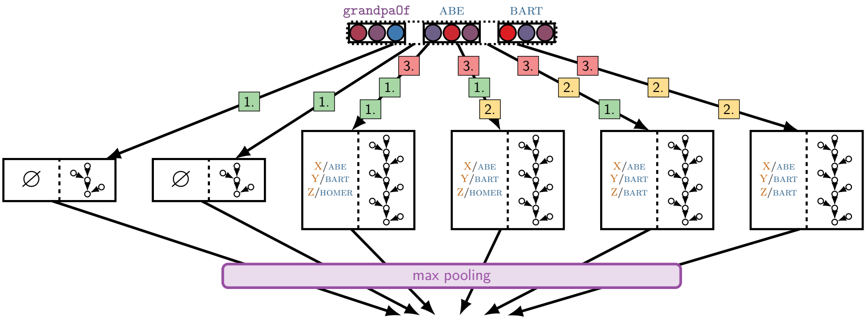
1. fatherOf(ABE, HOMER).
2. parentOf(HOMER, BART).
3. $\theta_1(X, Y)$:–
   $\theta_2(X, Z)$,
   $\theta_3(Z, Y)$.



Training Objective

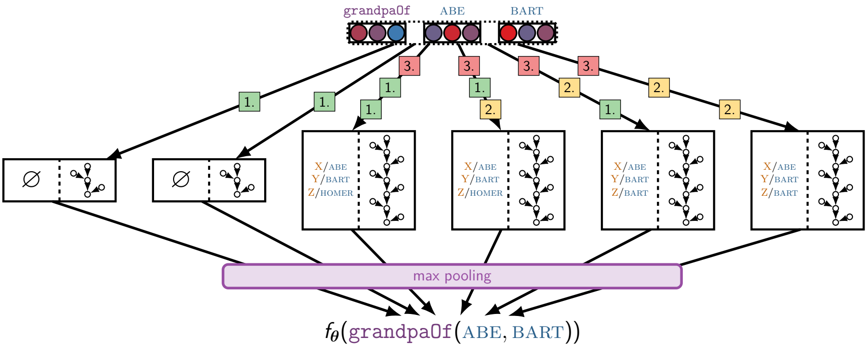


Training Objective



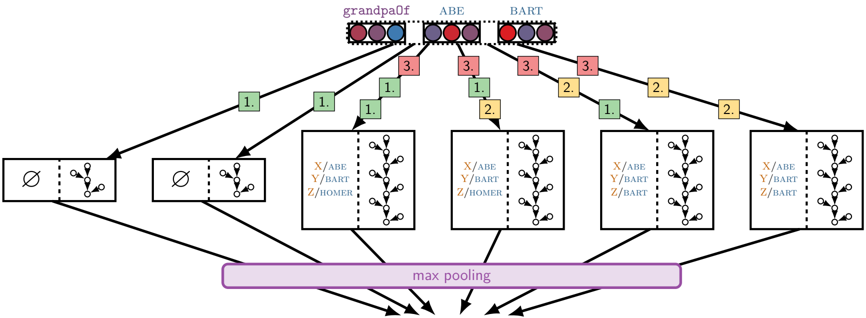
$f_\theta(\texttt{grandpaOf}(\text{ABE}, \text{BART}))$

Training Objective



- Loss: negative log-likelihood w.r.t. target proof success

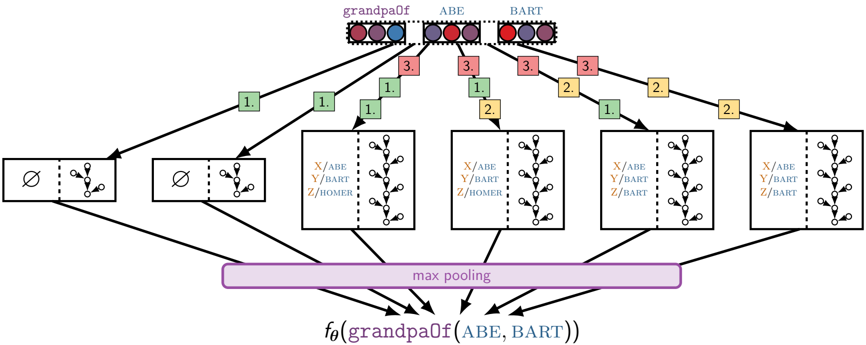
Training Objective



$f_\theta(\texttt{grandpaOf}(\text{ABE}, \text{BART}))$

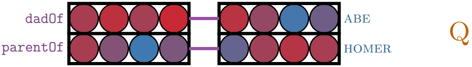
- Loss: negative log-likelihood w.r.t. target proof success
- Trained end-to-end using stochastic gradient descent

Training Objective

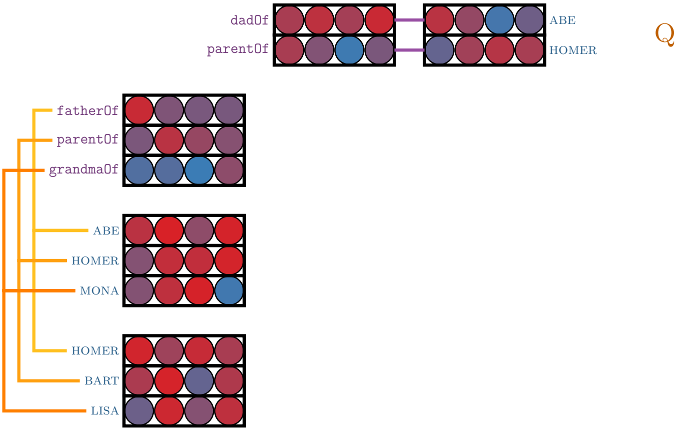


- Loss: negative log-likelihood w.r.t. target proof success
- Trained end-to-end using stochastic gradient descent
- Vectors are **learned such that proof success is high for known facts** and low for sampled negative facts

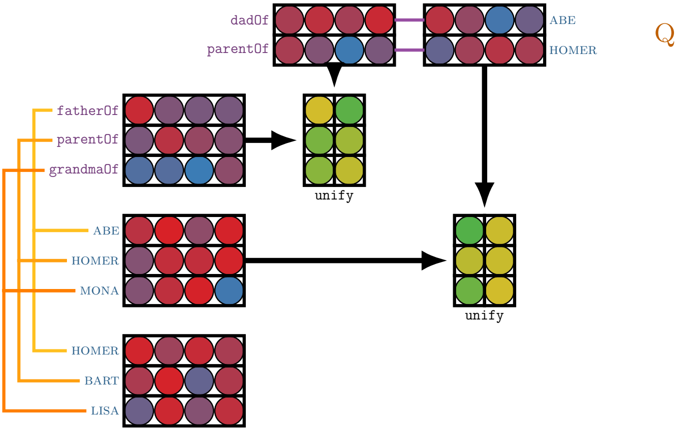
Calculation on GPU



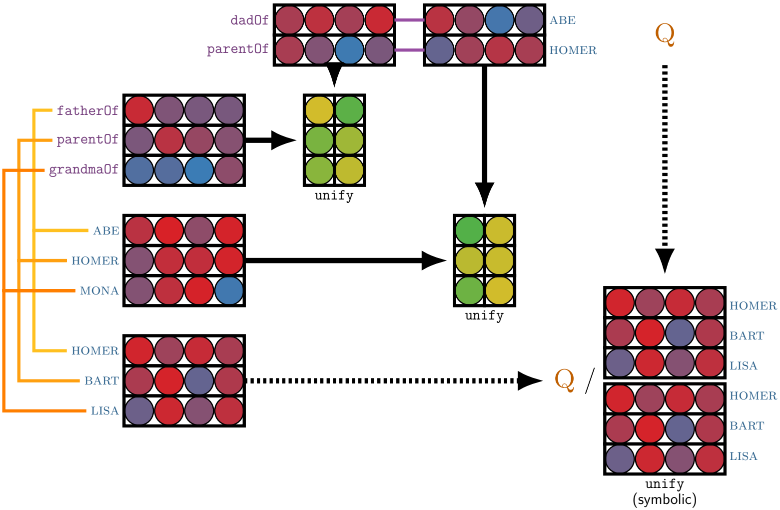
Calculation on GPU



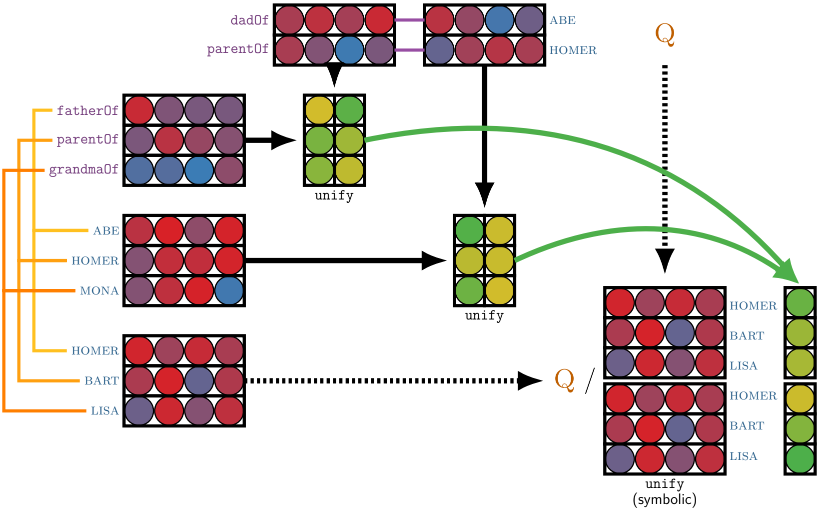
Calculation on GPU



Calculation on GPU

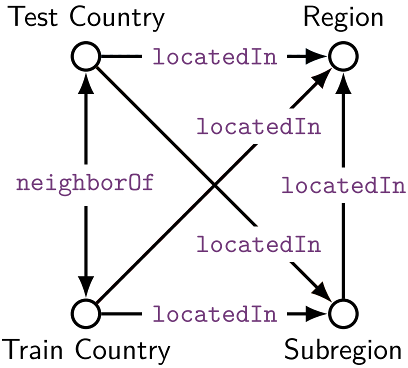


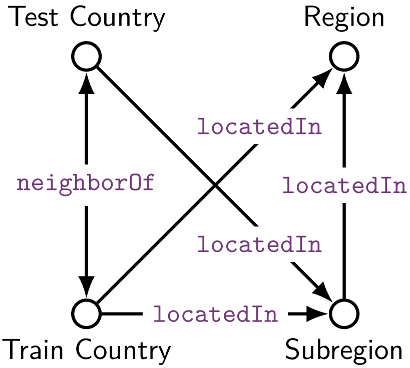
Calculation on GPU



Experiments

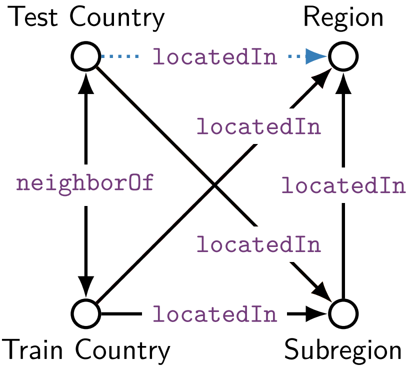
Benchmark Knowledge Bases: **Kinship**, **Nations**, **UMLS** (Kok and Domingos, 2007), and **Countries** (Bouchard et al., 2015)

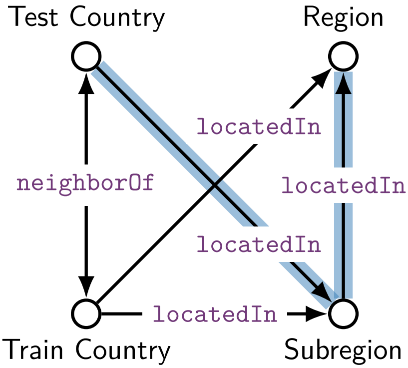




Experiments

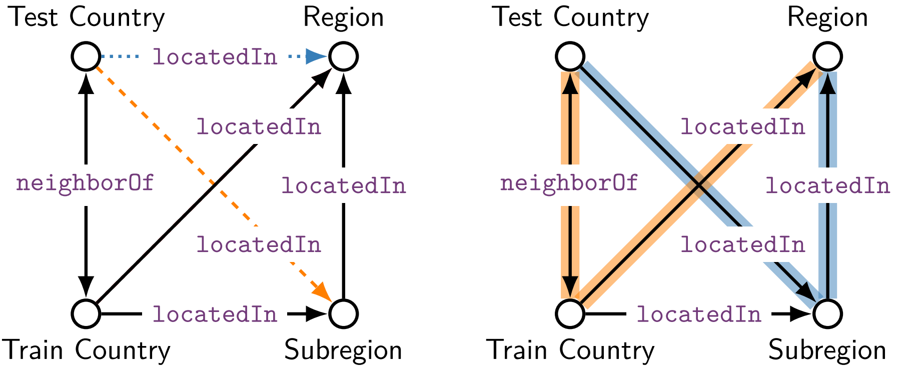
Benchmark Knowledge Bases: **Kinship**, **Nations**, **UMLS** (Kok and Domingos, 2007), and **Countries** (Bouchard et al., 2015)





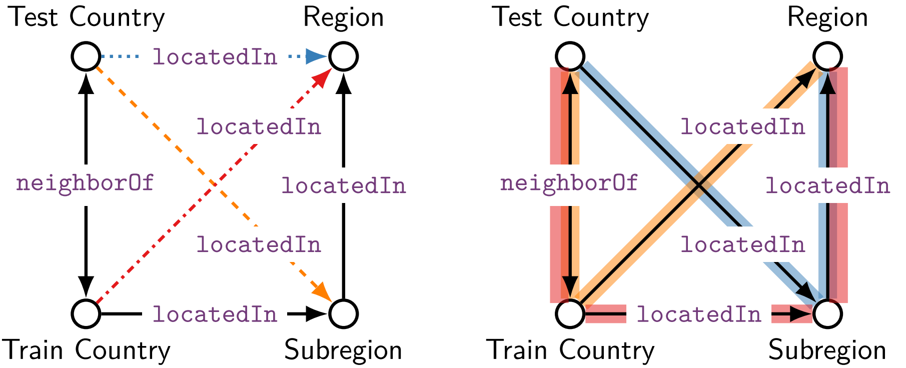
Experiments

Benchmark Knowledge Bases: **Kinship**, **Nations**, **UMLS** (Kok and Domingos, 2007), and **Countries** (Bouchard et al., 2015)



Experiments

Benchmark Knowledge Bases: **Kinship**, **Nations**, **UMLS** (Kok and Domingos, 2007), and **Countries** (Bouchard et al., 2015)



Details

- Models implemented in TensorFlow

Details

- Models implemented in TensorFlow

  ComplEx Neural link prediction model by Trouillon et al. (2016)

Details

- Models implemented in TensorFlow

  ComplEx Neural link prediction model by Trouillon et al. (2016)

  Prover End-to-end differentiable prover

Details

- Models implemented in TensorFlow

ComplEx	Neural link prediction model by Trouillon et al. (2016)
Prover	End-to-end differentiable prover
Proverλ	Same, but representations trained with ComplEx as auxiliary task

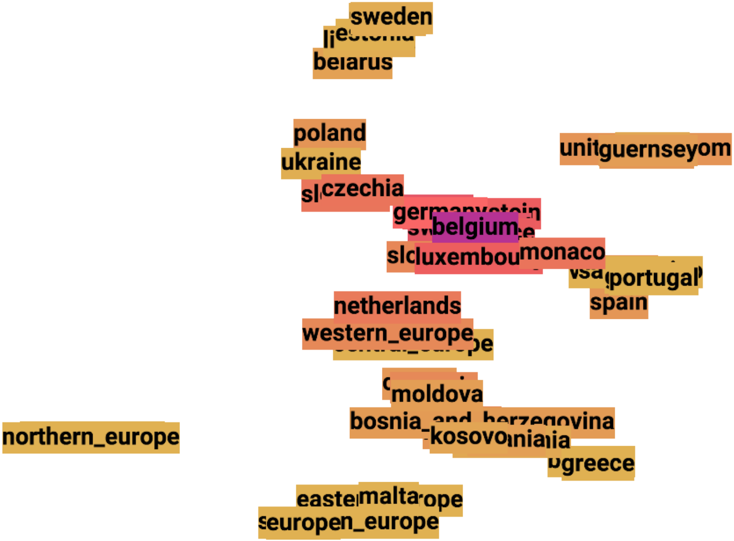
Details

- Models implemented in TensorFlow

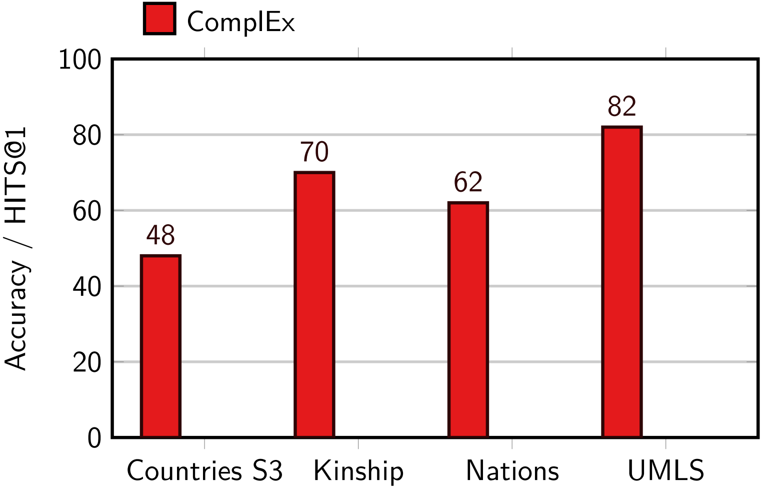
ComplEx	Neural link prediction model by Trouillon et al. (2016)
Prover	End-to-end differentiable prover
Proverλ	Same, but representations trained with ComplEx as auxiliary task

- Rule Templates:

Kinship, Nations & UMLS
20 $\#1(X, Y)$:– $\#2(X, Y)$.
20 $\#1(X, Y)$:– $\#2(Y, X)$.
20 $\#1(X, Y)$:– $\#2(X, Z), \#3(Z, Y)$.
Countries S1
3 $\#1(X, Y)$:– $\#1(Y, X)$.
3 $\#1(X, Y)$:– $\#2(X, Z), \#2(Z, Y)$.
Countries S2
3 $\#1(X, Y)$:– $\#2(X, Z), \#3(Z, Y)$.
Countries S3
3 $\#1(X, Y)$:– $\#2(X, Z), \#3(Z, W), \#4(W, Y)$.

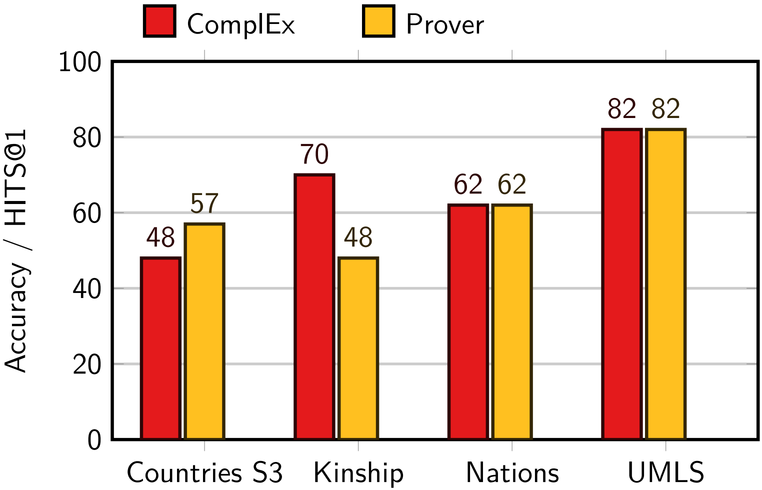

sweden
belarus
poland
ukraine
czechia
belgium
monaco
guernsey
portugal
spain
netherlands
western_europe
moldova
kosovo
greece
northern_europe
malta

Results



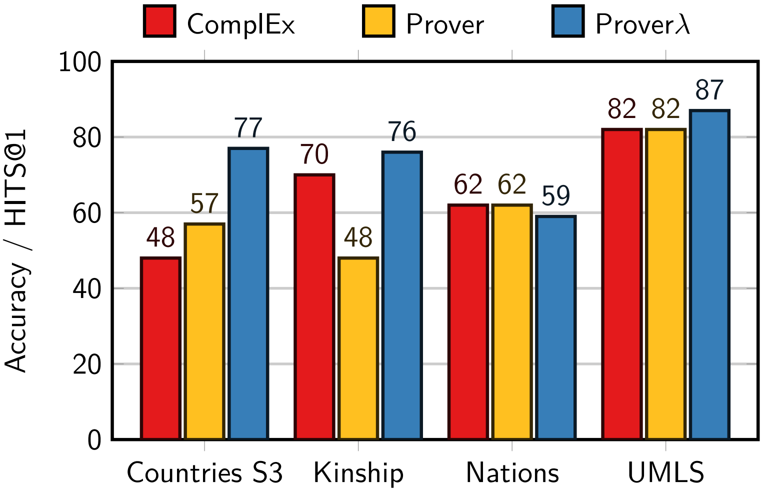
ComplEx
Accuracy / HITS@1
100
80
60
40
20
0
48
70
62
82
Countries S3
Kinship
Nations
UMLS

Results



ComplEx
Prover
100
80
60
40
20
0
Accuracy / HITS@1
48
57
70
48
62
62
82
82
Countries S3
Kinship
Nations
UMLS

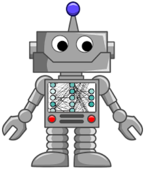
Results


ComplEx
Prover
Proverλ
Accuracy / HITS@1
0
20
40
60
80
100
48
57
77
70
48
76
62
62
59
82
82
87
Countries S3
Kinship
Nations
UMLS

Examples of Induced Rules

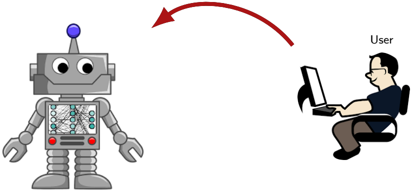
Corpus		Induced rules and their confidence
Countries	S1	0.90 `locatedIn`(X,Y) :– `locatedIn`(X,Z), `locatedIn`(Z,Y).
	S2	0.63 `locatedIn`(X,Y) :– `neighborOf`(X,Z), `locatedIn`(Z,Y).
	S3	0.32 `locatedIn`(X,Y) :– `neighborOf`(X,Z), `neighborOf`(Z,W), `locatedIn`(W,Y).
Nations		0.68 `blockpositionindex`(X,Y) :– `blockpositionindex`(Y,X).
		0.46 `expeldiplomats`(X,Y) :– `negativebehavior`(X,Y).
		0.38 `negativecomm`(X,Y) :– `commonbloc0`(X,Y).
		0.38 `intergovorgs3`(X,Y) :– `intergovorgs`(Y,X).
UMLS		0.88 `interacts_with`(X,Y) :– `interacts_with`(X,Z), `interacts_with`(Z,Y).
		0.77 `isa`(X,Y) :– `isa`(X,Z), `isa`(Z,Y).
		0.71 `derivative_of`(X,Y) :– `derivative_of`(X,Z), `derivative_of`(Z,Y).

Outlook

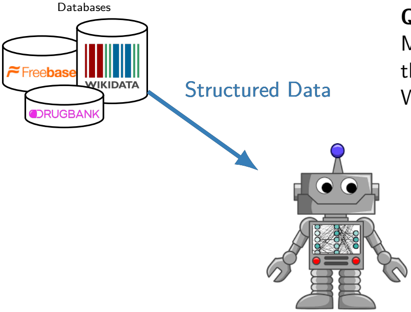


Outlook

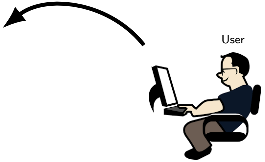
Question
My patient is not responding after three days of codeine treatment. What could have happened?



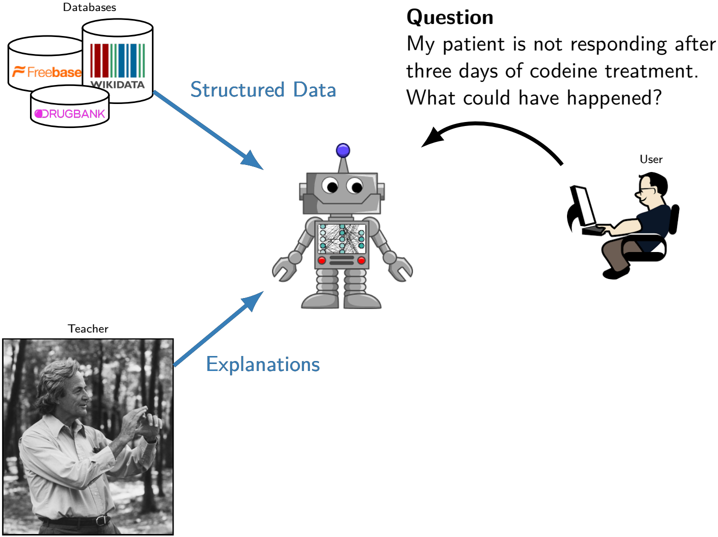
Outlook



Question
My patient is not responding after three days of codeine treatment. What could have happened?



Outlook


Databases
Freebase
WIKIDATA
DRUGBANK
Structured Data
Question
My patient is not responding after three days of codeine treatment. What could have happened?
User
Teacher
Explanations

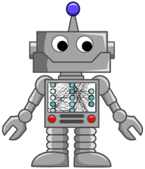
Outlook

Databases

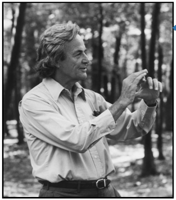
Structured Data

Publications

Text



Teacher

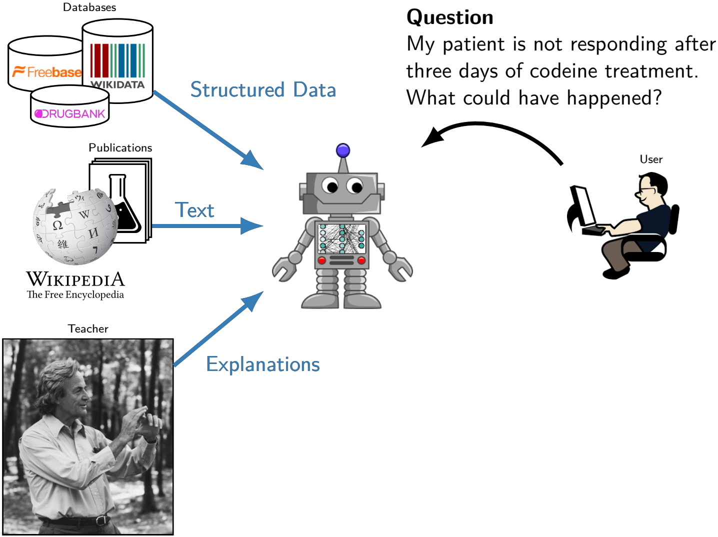


Explanations

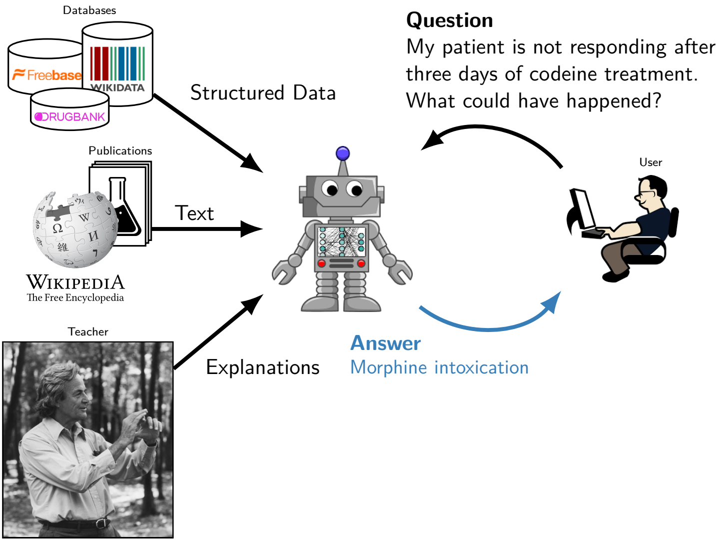
Question
My patient is not responding after three days of codeine treatment. What could have happened?

User

Outlook


Databases
Freebase
WIKIDATA
DRUGBANK
Structured Data
Publications
WIKIPEDIA
The Free Encyclopedia
Text
Teacher
Explanations
Question
My patient is not responding after three days of codeine treatment. What could have happened?
User

Outlook

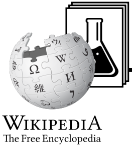

Databases
Freebase
WIKIDATA
DRUGBANK
Structured Data
Publications
WIKIPEDIA
The Free Encyclopedia
Text
Teacher
Explanations
Question
My patient is not responding after three days of codeine treatment. What could have happened?
User
Answer
Morphine intoxication

Outlook

Databases

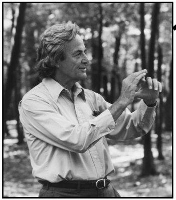
Structured Data

Publications

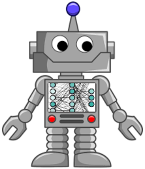


Text

Teacher



Explanations



Question
My patient is not responding after three days of codeine treatment. What could have happened?

User

Answer
Morphine intoxication

Proof

- Codeine is metabolized to morphine
- Mutation in CYP2D6 can cause ultrarapid metabolization
- Ultrarapid metabolization can lead to morphine overdose
- Morphine overdose is an intoxication

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent
- **Induces interpretable rules** from data via gradient descent

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent
- **Induces interpretable rules** from data via gradient descent
- Various computational optimizations: batch proving, tree pruning etc.

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent
- **Induces interpretable rules** from data via gradient descent
- Various computational optimizations: batch proving, tree pruning etc.
- Future research:

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent
- **Induces interpretable rules** from data via gradient descent
- Various computational optimizations: batch proving, tree pruning etc.
- Future research:
  - **Scaling up** to larger knowledge bases

Summary

- We proposed various ways of **regularizing vector representations of symbols using rules**
- We used Prolog's backward chaining as recipe for recursively constructing a neural network to prove queries to a knowledge base
- Proof success differentiable w.r.t. vector representations of symbols
- **Symbolic rule application but neural unification**
- **Learns vector representations of symbols** from data via gradient descent
- **Induces interpretable rules** from data via gradient descent
- Various computational optimizations: batch proving, tree pruning etc.
- Future research:
  - **Scaling up** to larger knowledge bases
  - **Connecting to RNNs** for proving with natural language statements

Thank you!

http://rockt.github.com
tim.rocktaschel@cs.ox.ac.uk
Twitter: @_rockt

References I

T. R. Besold, A. S. d'Avila Garcez, S. Bader, H. Bowman, P. M. Domingos, P. Hitzler, K. Kühnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902, 2017. URL http://arxiv.org/abs/1711.03902.

G. Bouchard, S. Singh, and T. Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *Proceedings of the 2015 AAAI Spring Symposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 2015.

W. W. Cohen. Tensorlog: A differentiable deductive database. *CoRR*, abs/1605.06523, 2016. URL http://arxiv.org/abs/1605.06523.

R. Das, A. Neelakantan, D. Belanger, and A. McCallum. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017. URL http://arxiv.org/abs/1607.01426.

A. S. d'Avila Garcez and G. Zaverucha. The connectionist inductive learning and logic programming system. *Appl. Intell.*, 11(1):59–77, 1999. doi: 10.1023/A:1008328630915. URL http://dx.doi.org/10.1023/A:1008328630915.

A. S. d'Avila Garcez, K. Broda, and D. M. Gabbay. *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media, 2012.

T. Demeester, **T. Rocktäschel**, and S. Riedel. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1389–1399, 2016. URL http://aclweb.org/anthology/D/D16/D16-1146.pdf.

L. Ding. Neural prolog-the concepts, construction and mechanism. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 4, pages 3603–3608. IEEE, 1995.

R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *CoRR*, abs/1711.04574, 2017. URL http://arxiv.org/abs/1711.04574.

S. Hölldobler. A structured connectionist unification algorithm. In *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, July 29 - August 3, 1990, 2 Volumes.*, pages 587–593, 1990. URL http://www.aaai.org/Library/AAAI/1990/aaai90-088.php.

References II

S. Kok and P. M. Domingos. Statistical predicate invention. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 433–440, 2007. doi: 10.1145/1273496.1273551. URL http://doi.acm.org/10.1145/1273496.1273551.

E. Komendantskaya. Unification neural networks: unification by error-correction learning. *Logic Journal of the IGPL*, 19(6):821–847, 2011. doi: 10.1093/jigpal/jzq012. URL http://dx.doi.org/10.1093/jigpal/jzq012.

P. Minervini, T. Demeester, **T. Rocktäschel**, and S. Riedel. Adversarial sets for regularised neural link predictors. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

T. Rocktäschel and S. Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3791–3803, 2017. URL http://papers.nips.cc/paper/6969-end-to-end-differentiable-proving.

T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1119–1129, 2015. URL http://aclweb.org/anthology/N/N15/N15-1118.pdf.

L. Serafini and A. S. d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. In *Proceedings of the 11th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy'16) co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence (HLAI 2016), New York City, NY, USA, July 16-17, 2016.*, 2016. URL http://ceur-ws.org/Vol-1768/NESY16_paper3.pdf.

L. Shastri. Neurally motivated constraints on the working memory capacity of a production system for parallel processing: Implications of a connectionist model based on temporal synchrony. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society: July 29 to August 1, 1992, Cognitive Science Program, Indiana University, Bloomington*, volume 14, page 159. Psychology Press, 1992.

J. W. Shavlik and G. G. Towell. An approach to combining explanation-based and neural learning algorithms. *Connection Science*, 1(3): 231–253, 1989.

References III

G. Sourek, V. Aschenbrenner, F. Zelezný, and O. Kuzelka. Lifted relational neural networks. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches co-located with the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015), Montreal, Canada, December 11-12, 2015.*, 2015. URL http://ceur-ws.org/Vol-1583/CoCoNIPS_2015_paper_7.pdf.

G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artif. Intell.*, 70(1-2):119–165, 1994. doi: 10.1016/0004-3702(94)90105-8. URL http://dx.doi.org/10.1016/0004-3702(94)90105-8.

T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2071–2080, 2016. URL http://jmlr.org/proceedings/papers/v48/trouillon16.html.

B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*, 2015. URL http://arxiv.org/abs/1412.6575.