Does Logic-based Reasoning Work for Dutch?

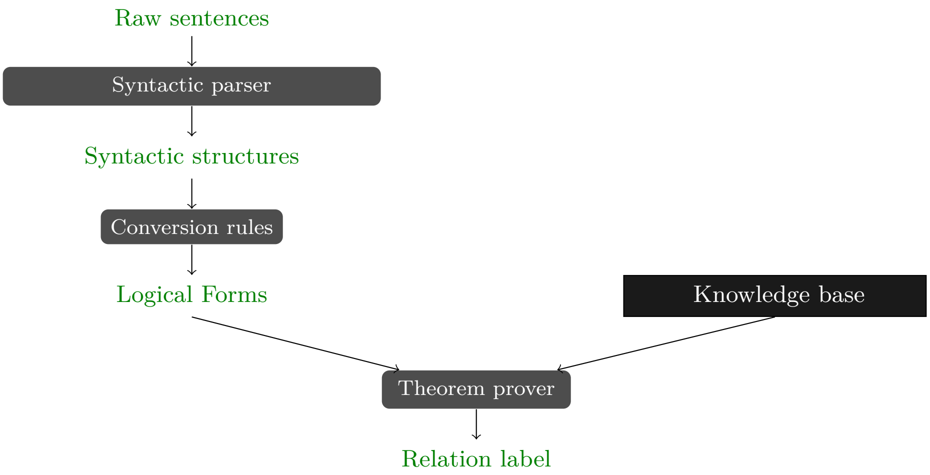
Lasha Abzianidze   Konstantinos Kogkalidis

Utrecht Institute of Linguistics OTS, Utrecht University

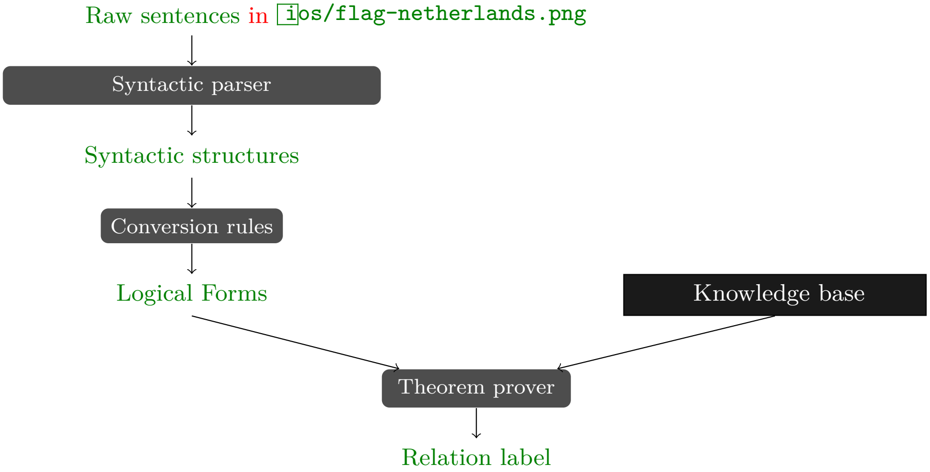
~~06-07-2021 @ CLIN~~

21-06-2021 @ NLP RG

Logic-based approach to NLI



Logic-based approach to NLI



Logic-based approach to NLI

Raw sentences in ios/flag-netherlands.png

Alpino & Neural Proof Nets

$\mathrm{ILL}_{\multimap,\diamond,\Box}$ derivations

Conversion rules

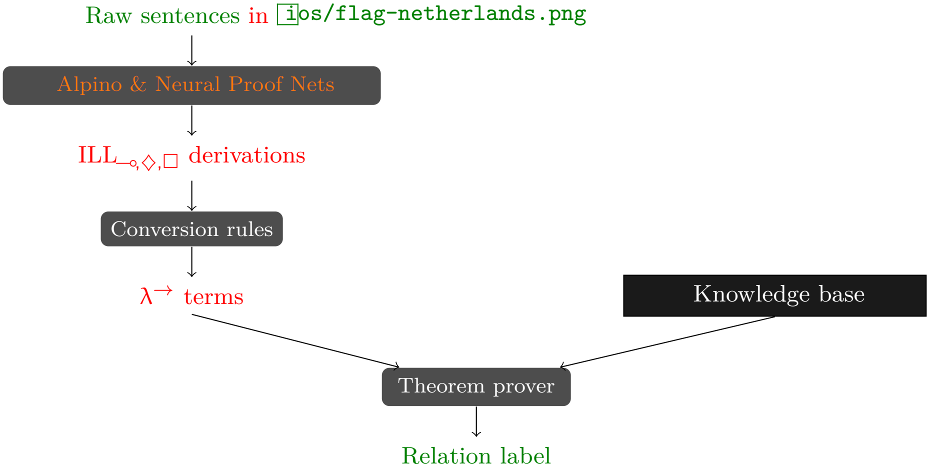
Logical Forms

Knowledge base

Theorem prover

Relation label

Logic-based approach to NLI



Logic-based approach to NLI

Raw sentences in ios/flag-netherlands.png

Alpino & Neural Proof Nets

$\mathrm{ILL}_{\multimap,\diamond,\square}$ derivations

Conversion rules

$\lambda^{\rightarrow}$ terms

Open Dutch WordNet

Theorem prover

Relation label

Logic-based approach to NLI

Raw sentences in ios/flag-netherlands.png

Alpino & Neural Proof Nets

$\mathrm{ILL}_{\multimap,\Diamond,\Box}$ derivations

Conversion rules

$\lambda^{\rightarrow}$ terms

Open Dutch WordNet

LangPro

Relation label

The linear λ-calculus

Words are assigned ILL types, inductively defined as: $\mathcal{T} := a \mid t_1 \multimap t_2 \mid \lozenge^{\delta} t \mid \square^{\alpha} t$ where

- a an atom, from a finite set $\mathcal{A}$:
  $\mathtt{np}$, $\mathtt{s}_{\mathtt{main}}$, $\mathtt{s}_{\mathtt{sub}}$, $\mathtt{pron}$, ...
- $t_1 \multimap t_2$ a linear function that consumes t_1 to produce t_2
  $\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}}$, $\mathtt{np} \multimap (\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}})$, $(\mathtt{np} \multimap \mathtt{s}_{\mathtt{sub}}) \multimap (\mathtt{np} \multimap \mathtt{np})$, ...

Syntactic derivations $\equiv$ proofs $\overset{\text{chc}}{\equiv}$ functional programs:

$$\tau := c^t \mid \left(\tau_1^{t_1 \multimap t_2}\ \tau_2^{t_1}\right)^{t_2} \mid \left(\lambda x^{t_1}.\tau^{t_2}\right)^{t_1 \multimap t_2} \mid \delta_{\square}\,\tau \mid \delta^{\square}\,\tau \mid \delta_{\lozenge}\,\tau \mid \delta^{\lozenge}\,\tau$$

A boy plays:

$$\mathrm{play}^{\lozenge^{su}\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}}} \left(su^{\lozenge} \left(\left(\mathrm{det}_{\square}\ \mathrm{a}^{\square^{det}(\mathtt{n} \multimap \mathtt{np})}\right) \mathrm{boy}^{\mathtt{n}}\right)\right)$$

The linear λ-calculus

Words are assigned ILL types, inductively defined as: $\mathcal{T} := a \mid t_1 \multimap t_2 \mid \Diamond^{\delta} t \mid \Box^{\alpha} t$ where

- a an atom, from a finite set $\mathcal{A}$:
  $\mathtt{np}$, $\mathtt{s}_{\mathtt{main}}$, $\mathtt{s}_{\mathtt{sub}}$, $\mathtt{pron}$, ...
- $t_1 \multimap t_2$ a linear function that consumes t_1 to produce t_2
  $\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}}$, $\mathtt{np} \multimap (\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}})$, $(\mathtt{np} \multimap \mathtt{s}_{\mathtt{sub}}) \multimap (\mathtt{np} \multimap \mathtt{np})$, ...

Syntactic derivations $\equiv$ proofs $\overset{\text{chc}}{\equiv}$ functional programs:

$$\tau := c^t \mid \left(\tau_1^{t_1 \multimap t_2}\ \tau_2^{t_1}\right)^{t_2} \mid \left(\lambda x^{t_1}.\tau^{t_2}\right)^{t_1 \multimap t_2} \mid \delta_{\Box}\,\tau \mid \delta^{\Box}\,\tau \mid \delta_{\Diamond}\,\tau \mid \delta^{\Diamond}\,\tau$$

A boy plays:

$$\mathrm{play}^{\Diamond^{su}\mathtt{np} \multimap \mathtt{s}_{\mathtt{main}}} \left(\mathrm{su}^{\Diamond} \left(\left(\mathrm{det}_{\Box}\ \mathrm{a}^{\Box^{det}(\mathtt{n} \multimap \mathtt{np})}\right) \mathrm{boy}^{\mathtt{n}}\right)\right)$$

The linear λ-calculus

Words are assigned ILL types, inductively defined as: $\mathcal{T} := a \mid t_1 \multimap t_2 \mid \diamondsuit^\delta t \mid \Box^\alpha t$ where

- a an atom, from a finite set $\mathcal{A}$:
  $\mathtt{np}$, $\mathtt{s_{main}}$, $\mathtt{s_{sub}}$, $\mathtt{pron}$, ...
- $t_1 \multimap t_2$ a linear function that consumes t_1 to produce t_2
  $\mathtt{np} \multimap \mathtt{s_{main}}$, $\mathtt{np} \multimap (\mathtt{np} \multimap \mathtt{s_{main}})$, $(\mathtt{np} \multimap \mathtt{s_{sub}}) \multimap (\mathtt{np} \multimap \mathtt{np})$, ...

Syntactic derivations ≡ proofs $\overset{\text{chc}}{\equiv}$ functional programs:

$$\tau := c^t \mid \left(\tau_1^{t_1 \multimap t_2}\ \tau_2^{t_1}\right)^{t_2} \mid \left(\lambda x^{t_1}.\tau^{t_2}\right)^{t_1 \multimap t_2} \mid \delta_\Box\, \tau \mid \delta^\Box\, \tau \mid \delta_\diamondsuit\, \tau \mid \delta^\diamondsuit\, \tau$$

A boy plays:

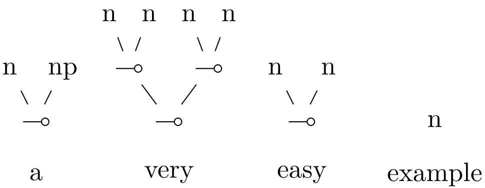
$$\text{play}^{\diamondsuit^{su}\mathtt{np} \multimap \mathtt{s_{main}}} \left(\text{su}^\diamondsuit \left(\left(\text{det}_\Box\ \text{a}^{\Box^{det}(\mathtt{n} \multimap \mathtt{np})}\right) \text{boy}^{\mathtt{n}}\right)\right)$$

Proof Nets

Types are trees, with nodes inductively polarized: + we have − we seek

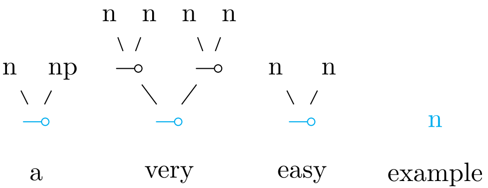
Proof Nets

Types are trees, with nodes inductively polarized: + we have − we seek



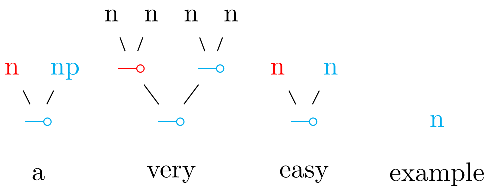
Proof Nets

Types are trees, with nodes inductively polarized: + we have — we seek



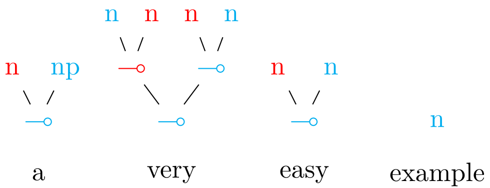
Proof Nets

Types are trees, with nodes inductively polarized: + we have − we seek



Proof Nets

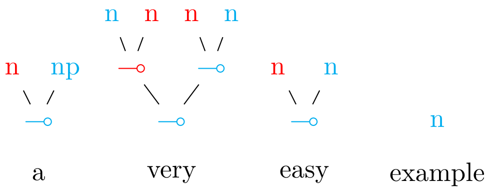
Types are trees, with nodes inductively polarized: + we have − we seek



Proof Nets

Proof net

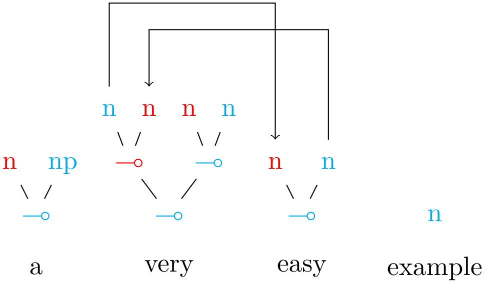
a polarized forest (proof frame) and a bijection between + and − (axiom links)



Proof Nets

Proof net

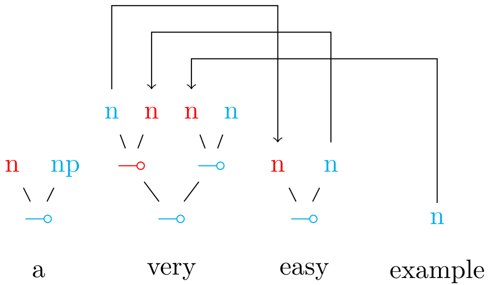
a polarized forest (proof frame) and a bijection between + and − (axiom links)



`very easy`

Proof Nets

Proof net
a polarized forest (proof frame) and a bijection between + and − (axiom links)

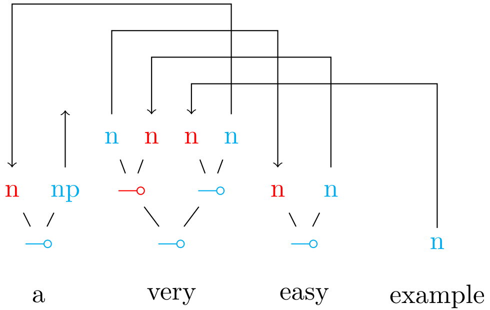


`very easy example`

Proof Nets

Proof net

a polarized forest (proof frame) and a bijection between + and − (axiom links)



`a (very easy example)`

Neural Proof Nets: from sentences to λ-terms

Supertagging

From sentences to proof frames with seq2seq transduction

a very easy example

Neural Proof Nets: from sentences to λ-terms

Supertagging

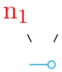
From sentences to proof frames with seq2seq transduction

a very easy example

Neural Proof Nets: from sentences to λ-terms

Supertagging

From sentences to proof frames with seq2seq transduction

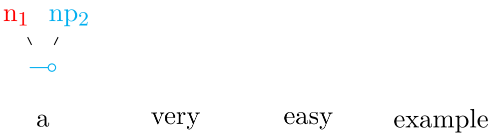


a very easy example

Neural Proof Nets: from sentences to λ-terms

Supertagging

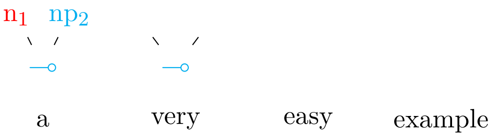
From sentences to proof frames with seq2seq transduction



Neural Proof Nets: from sentences to λ-terms

Supertagging

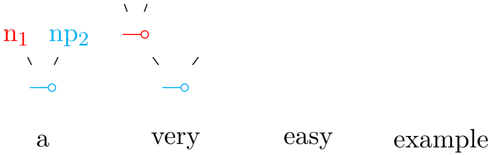
From sentences to proof frames with seq2seq transduction



Neural Proof Nets: from sentences to λ-terms

Supertagging

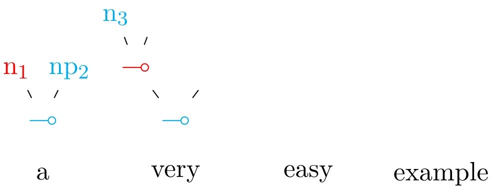
From sentences to proof frames with seq2seq transduction



Neural Proof Nets: from sentences to λ-terms

Supertagging

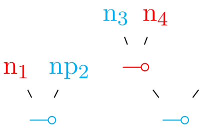
From sentences to proof frames with seq2seq transduction



Neural Proof Nets: from sentences to λ-terms

Supertagging

From sentences to proof frames with seq2seq transduction

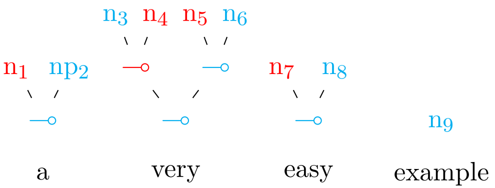


a very easy example

Neural Proof Nets: from sentences to λ-terms

Supertagging

From sentences to proof frames with seq2seq transduction



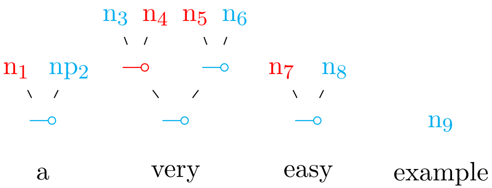
Neural Proof Nets: from sentences to λ-terms

Supertagging

From sentences to proof frames with seq2seq transduction

Proving

From proof frames to axiom links with Sinkhorn-Knopp



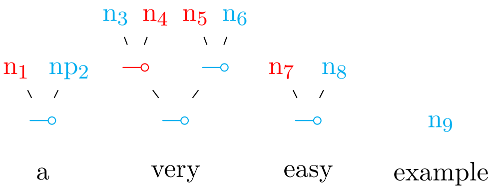
Neural Proof Nets: from sentences to λ-terms

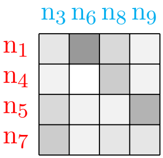
Supertagging

From sentences to proof frames with seq2seq transduction

Proving

From proof frames to axiom links with Sinkhorn-Knopp





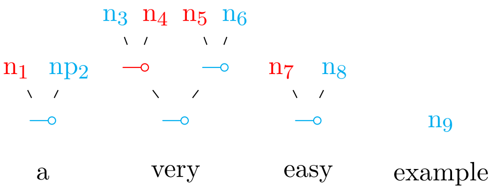
Neural Proof Nets: from sentences to λ-terms

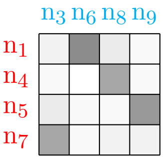
Supertagging

From sentences to proof frames with seq2seq transduction

Proving

From proof frames to axiom links with Sinkhorn-Knopp





Other ingredients: Alpino & ODWN

Alpino:

- Stochastic Attribute Value Grammar (HPSG) for Dutch
- Builds dependency structures
- Used for pre-processing Dutch treebanks

Open Dutch WordNet:

- 52K synsets (vs 117K Princeton WN)
- Derived from the Dutch lexical semantic database Cornetto

- We converted ODWN into the prolog format
- Used relations: synonymy, hypernymy, antonymy, similarity, and derivation.

Other ingredients: Alpino & ODWN

Alpino:

- Stochastic Attribute Value Grammar (HPSG) for Dutch
- Builds dependency structures
- Used for pre-processing Dutch treebanks

Open Dutch WordNet:

- 52K synsets (vs 117K Princeton WN)
- Derived from the Dutch lexical semantic database Cornetto
- We converted ODWN into the prolog format
- Used relations: synonymy, hypernymy, antonymy, similarity, and derivation.

Other ingredients: Alpino & ODWN

Alpino:

- Stochastic Attribute Value Grammar (HPSG) for Dutch
- Builds dependency structures
- Used for pre-processing Dutch treebanks

Open Dutch WordNet:

- 52K synsets (vs 117K Princeton WN)
- Derived from the Dutch lexical semantic database Cornetto

- We converted ODWN into the prolog format
- Used relations: synonymy, hypernymy, antonymy, similarity, and derivation.

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- $\approx$ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

Natural Tableau: proving ios/flag-united-kingdom.png

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$
2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

Natural Tableau: proving ios/flag-united-kingdom.png

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- $\approx$ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

$$
\begin{array}{c}
\boxed{1}\ \text{a hedgehog}\ (\text{be}\ (\lambda x.\ \text{a boy}\ (\lambda y.\ \text{by}\ y\ \text{cradle}\ x))) : \mathbb{T} \\
\boxed{2}\ \text{a person}\ (\lambda x.\ \text{an animal}\ (\lambda y.\ \text{hold}\ y\ x) : \mathbb{F} \\
[1] \mid \\
\boxed{3}\ \text{hedgehog} : [h] : \mathbb{T} \\
\boxed{4}\ \text{a boy}\ (\lambda y.\ \text{by}\ y\ \text{cradle}\ h) : \mathbb{T} \\
[4] \mid \\
\boxed{5}\ \text{boy} : [b] : \mathbb{T} \\
\boxed{6}\ \text{by}\ b\ \text{cradle} : [h] : \mathbb{T}
\end{array}
$$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$

6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$

6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$ | 9 an animal (λy. hold y b) : $\mathbb{F}$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- $\approx$ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction: hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$

6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$ | 9 an animal (λy. hold y b) : $\mathbb{F}$

[5,8]

$\times$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction: hedgehog ⊑ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$
2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$
4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$
6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$

[5,8]

×

9 an animal (λy. hold y b) : $\mathbb{F}$

[9,h]

10 animal : [h] : $\mathbb{F}$

11 hold : [h, b] : $\mathbb{F}$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- $\approx$ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction:
  hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$
2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$
4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$
6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$

[5,8]

×

9 an animal (λy. hold y b) : $\mathbb{F}$

[9,h]

10 animal : [h] : $\mathbb{F}$

[3,10]

×

11 hold : [h, b] : $\mathbb{F}$

Natural Tableau: proving ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction: hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$
2 a person (λx. an animal (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$
4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$
6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$
[5,8]
×

9 an animal (λy. hold y b) : $\mathbb{F}$

[9,h]

10 animal : [h] : $\mathbb{F}$
[3,10]
×

11 hold : [h, b] : $\mathbb{F}$
[7,11]
×

Natural Tableau: proving and learning ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction: hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. a (small animal) (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$

6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$

[5,8]

×

9 a (small animal) (λy. hold y b) : $\mathbb{F}$

[9,h]

10 small animal : [h] : $\mathbb{F}$

[3,10]

open

11 hold : [h, b] : $\mathbb{F}$

[7,11]

×

Natural Tableau: proving and learning ios/flag-united-kingdom.png

- Natural logic + semantic tableau
- Uniform to Ent./Cont.
- Prove with refutation
- Uniform to premise number
- ≈ Syntactic trees
- Native higher-order logic
- Decomposing meaning
- Learning as abduction: hedgehog $\sqsubseteq$ small animal

1 a hedgehog (be (λx. a boy (λy. by y cradle x))) : $\mathbb{T}$

2 a person (λx. a (small animal) (λy. hold y x) : $\mathbb{F}$

[1]

3 hedgehog : [h] : $\mathbb{T}$

4 a boy (λy. by y cradle h) : $\mathbb{T}$

[4]

5 boy : [b] : $\mathbb{T}$

6 by b cradle : [h] : $\mathbb{T}$

[6]

7 cradle : [h, b]$\mathbb{T}$

[2,b]

8 person : [b] : $\mathbb{F}$

[5,8]

×

9 a (small animal) (λy. hold y b) : $\mathbb{F}$

[9,h]

10 small animal : [h] : $\mathbb{F}$

[3,10]

open

11 hold : [h, b] : $\mathbb{F}$

[7,11]

×

Syntactic λ-terms to λ-logical forms

$$\text{play}^{\Diamond^{su}\mathtt{np}\multimap \mathtt{s}_{\mathtt{main}}}\left(\text{su}^{\Diamond}\left(\left(\text{det}_{\Box}\ \text{a}^{\Box^{det}(\mathtt{n}\multimap\mathtt{np})}\right)\text{boy}^{\mathtt{n}}\right)\right) \rightsquigarrow \text{play}^{\mathtt{np},\mathtt{s}}\left(\text{a}^{\mathtt{n},\mathtt{np}}\ \text{boy}^{\mathtt{n}}\right)$$

$$\text{large}^{\mathtt{np},\mathtt{np}}\left(\text{brown}^{\mathtt{np},\mathtt{np}}\left(\text{a}^{\mathtt{n},\mathtt{np}}\ \text{dog}^{\mathtt{n}}\right)\right) \rightsquigarrow \text{a}^{\mathtt{n},\mathtt{np}}\left(\text{large}^{\mathtt{n},\mathtt{n}}\left(\text{brown}^{\mathtt{n},\mathtt{n}}\ \text{dog}^{\mathtt{n}}\right)\right)$$

$$\text{and}\ \left(\lambda x.\ \text{brown}(x\ \text{dog})\right)\left(\lambda y.\ \text{black}(y\ \text{dog})\right)\ \text{no}$$

$$\rightsquigarrow \text{and}^{\mathtt{np},\mathtt{np},\mathtt{np}}\left(\text{no}\ (\text{brown}\ \text{dog})\right)\left(\text{no}\ (\text{black}\ \text{dog})\right)$$

$$\text{cut}^{\mathtt{pp},\mathtt{n},\mathtt{np},\mathtt{s}}\left(\text{in}^{\mathtt{n},\mathtt{pp}}\ \text{slice}_{\mathtt{n}}\right)\ \text{meat}_{\mathtt{n}} \rightsquigarrow \text{cut}^{\mathtt{pp},\mathtt{np},\mathtt{np},\mathtt{s}}\left(\text{in}^{\mathtt{np},\mathtt{pp}}\ \{\text{slice}^{\mathtt{n}}\}^{\mathtt{np}}\right)\ \{\text{meat}^{\mathtt{n}}\}^{\mathtt{np}}$$

- Map POS tags and shift to slightly Generalized POS tags: UPOS & Penn
- Use only these syntactic categories: $\mathtt{n}, \mathtt{np}_{x}, \mathtt{s}_{x}, \mathtt{pp}, \mathtt{pr}$
- Function words $\mapsto$ canonical terms (excl. prepositions)

Syntactic λ-terms to λ-logical forms

$$\text{play}^{\Diamond^{su}\mathtt{np}\multimap\mathtt{s}_{\mathtt{main}}}\left(\text{su}^{\Diamond}\left(\left(\text{det}_{\Box}\ \text{a}^{\Box^{det}(\mathtt{n}\multimap\mathtt{np})}\right)\text{boy}^{\mathtt{n}}\right)\right) \rightsquigarrow \text{play}^{\mathtt{np},\mathtt{s}}\left(\text{a}^{\mathtt{n},\mathtt{np}}\ \text{boy}^{\mathtt{n}}\right)$$

$$\text{large}^{\mathtt{np},\mathtt{np}}\left(\text{brown}^{\mathtt{np},\mathtt{np}}\left(\text{a}^{\mathtt{n},\mathtt{np}}\ \text{dog}^{\mathtt{n}}\right)\right) \rightsquigarrow \text{a}^{\mathtt{n},\mathtt{np}}\left(\text{large}^{\mathtt{n},\mathtt{n}}\left(\text{brown}^{\mathtt{n},\mathtt{n}}\ \text{dog}^{\mathtt{n}}\right)\right)$$

$$\text{and}\ (\lambda x.\,\text{brown}(x\ \text{dog}))(\lambda y.\,\text{black}(y\ \text{dog}))\ \text{no}$$

$$\rightsquigarrow \text{and}^{\mathtt{np},\mathtt{np},\mathtt{np}}\left(\text{no}\left(\text{brown dog}\right)\right)\left(\text{no}\left(\text{black dog}\right)\right)$$

$$\text{cut}^{\mathtt{pp},\mathtt{n},\mathtt{np},\mathtt{s}}\left(\text{in}^{\mathtt{n},\mathtt{pp}}\ \text{slice}_{\mathtt{n}}\right)\ \text{meat}_{\mathtt{n}} \rightsquigarrow \text{cut}^{\mathtt{pp},\mathtt{np},\mathtt{np},\mathtt{s}}\left(\text{in}^{\mathtt{np},\mathtt{pp}}\ \{\text{slice}^{\mathtt{n}}\}^{\mathtt{np}}\right)\ \{\text{meat}^{\mathtt{n}}\}^{\mathtt{np}}$$

- Map POS tags and shift to slightly Generalized POS tags: UPOS & Penn
- Use only these syntactic categories: $\mathtt{n}, \mathtt{np}_{\mathrm{x}}, \mathtt{s}_{\mathrm{x}}, \mathtt{pp}, \mathtt{pr}$
- Function words $\mapsto$ canonical terms (excl. prepositions)

Syntactic λ-terms to λ-logical forms

$$\mathrm{play}^{\Diamond^{su}\mathtt{np}\multimap\mathtt{s}_{\mathtt{main}}}\left(\mathrm{su}^{\Diamond}\left(\left(\mathrm{det}_{\Box}\ \mathrm{a}^{\Box^{det}(\mathtt{n}\multimap\mathtt{np})}\right)\mathrm{boy}^{\mathtt{n}}\right)\right) \rightsquigarrow \mathrm{play}^{\mathtt{np},\mathtt{s}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{boy}^{\mathtt{n}}\right)$$

$$\mathrm{large}^{\mathtt{np},\mathtt{np}}\left(\mathrm{brown}^{\mathtt{np},\mathtt{np}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{dog}^{\mathtt{n}}\right)\right) \rightsquigarrow \mathrm{a}^{\mathtt{n},\mathtt{np}}\left(\mathrm{large}^{\mathtt{n},\mathtt{n}}\left(\mathrm{brown}^{\mathtt{n},\mathtt{n}}\ \mathrm{dog}^{\mathtt{n}}\right)\right)$$

$$\mathrm{and}\ \left(\lambda x.\,\mathrm{brown}(x\ \mathrm{dog})\right)\left(\lambda y.\,\mathrm{black}(y\ \mathrm{dog})\right)\ \mathrm{no}$$

$$\rightsquigarrow \mathrm{and}^{\mathtt{np},\mathtt{np},\mathtt{np}}\left(\mathrm{no}\ (\mathrm{brown}\ \mathrm{dog})\right)\left(\mathrm{no}\ (\mathrm{black}\ \mathrm{dog})\right)$$

$$\mathrm{cut}^{\mathtt{pp},\mathtt{n},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\mathtt{n},\mathtt{pp}}\ \mathrm{slice}_{\mathtt{n}}\right)\ \mathrm{meat}_{\mathtt{n}} \rightsquigarrow \mathrm{cut}^{\mathtt{pp},\mathtt{np},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\mathtt{np},\mathtt{pp}}\ \{\mathrm{slice}^{\mathtt{n}}\}^{\mathtt{np}}\right)\ \{\mathrm{meat}^{\mathtt{n}}\}^{\mathtt{np}}$$

- Map POS tags and shift to slightly Generalized POS tags: UPOS & Penn
- Use only these syntactic categories: $\mathtt{n}, \mathtt{np}_x, \mathtt{s}_x, \mathtt{pp}, \mathtt{pr}$
- Function words ↦ canonical terms (excl. prepositions)

Syntactic λ-terms to λ-logical forms

$$\mathrm{play}^{\diamond^{su}\mathtt{np}\multimap\mathtt{s}_{\mathtt{main}}}\Big(\mathrm{su}^{\diamond}\Big(\Big(\mathrm{det}_{\Box}\ \mathrm{a}^{\Box^{det}(\mathtt{n}\multimap\mathtt{np})}\Big)\ \mathrm{boy}^{\mathtt{n}}\Big)\Big) \rightsquigarrow \mathrm{play}^{\mathtt{np},\mathtt{s}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{boy}^{\mathtt{n}}\right)$$

$$\mathrm{large}^{\mathtt{np},\mathtt{np}}\left(\mathrm{brown}^{\mathtt{np},\mathtt{np}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{dog}^{\mathtt{n}}\right)\right) \rightsquigarrow \mathrm{a}^{\mathtt{n},\mathtt{np}}\left(\mathrm{large}^{\color{red}{\mathtt{n},\mathtt{n}}}\left(\mathrm{brown}^{\color{red}{\mathtt{n},\mathtt{n}}}\ \mathrm{dog}^{\mathtt{n}}\right)\right)$$

$$\mathrm{and}\ \big(\lambda x.\,\mathrm{brown}(x\ \mathrm{dog})\big)\big(\lambda y.\,\mathrm{black}(y\ \mathrm{dog})\big)\ \mathrm{no}$$

$$\rightsquigarrow \mathrm{and}^{\color{red}{\mathtt{np},\mathtt{np},\mathtt{np}}}\big(\color{red}{\mathrm{no}}\ (\color{red}{\mathrm{brown}}\ \mathrm{dog})\big)\ \big(\color{red}{\mathrm{no}}\ (\color{red}{\mathrm{black}}\ \mathrm{dog})\big)$$

$$\mathrm{cut}^{\mathtt{pp},\mathtt{n},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\mathtt{n},\mathtt{pp}}\ \mathrm{slice}_{\mathtt{n}}\right)\ \mathrm{meat}_{\mathtt{n}} \rightsquigarrow \mathrm{cut}^{\mathtt{pp},\color{red}{\mathtt{np}},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\color{red}{\mathtt{np}},\mathtt{pp}}\ \color{red}{\{}\mathrm{slice}^{\mathtt{n}}\color{red}{\}^{\mathtt{np}}}\right)\ \color{red}{\{}\mathrm{meat}^{\mathtt{n}}\color{red}{\}^{\mathtt{np}}}$$

- Map POS tags and shift to slightly Generalized POS tags: UPOS & Penn
- Use only these syntactic categories: $\mathtt{n}, \mathtt{np}_x, \mathtt{s}_x, \mathtt{pp}, \mathtt{pr}$
- Function words $\mapsto$ canonical terms (excl. prepositions)

Syntactic λ-terms to λ-logical forms

$$\mathrm{play}^{\Diamond^{su}\mathtt{np}\multimap\mathtt{s}_{\mathtt{main}}}\left(\mathrm{su}^{\Diamond}\left(\left(\mathrm{det}_{\Box}\ \mathrm{a}^{\Box^{det}(\mathtt{n}\multimap\mathtt{np})}\right)\mathrm{boy}^{\mathtt{n}}\right)\right) \rightsquigarrow \mathrm{play}^{\mathtt{np},\mathtt{s}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{boy}^{\mathtt{n}}\right)$$

$$\mathrm{large}^{\mathtt{np},\mathtt{np}}\left(\mathrm{brown}^{\mathtt{np},\mathtt{np}}\left(\mathrm{a}^{\mathtt{n},\mathtt{np}}\ \mathrm{dog}^{\mathtt{n}}\right)\right) \rightsquigarrow \mathrm{a}^{\mathtt{n},\mathtt{np}}\left(\mathrm{large}^{\mathtt{n},\mathtt{n}}\left(\mathrm{brown}^{\mathtt{n},\mathtt{n}}\ \mathrm{dog}^{\mathtt{n}}\right)\right)$$

$$\mathrm{and}\ \left(\lambda x.\ \mathrm{brown}(x\ \mathrm{dog})\right)\left(\lambda y.\ \mathrm{black}(y\ \mathrm{dog})\right)\ \mathrm{no} \rightsquigarrow \mathrm{and}^{\mathtt{np},\mathtt{np},\mathtt{np}}\left(\mathrm{no}\ (\mathrm{brown}\ \mathrm{dog})\right)\left(\mathrm{no}\ (\mathrm{black}\ \mathrm{dog})\right)$$

$$\mathrm{cut}^{\mathtt{pp},\mathtt{n},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\mathtt{n},\mathtt{pp}}\ \mathrm{slice}_{\mathtt{n}}\right)\ \mathrm{meat}_{\mathtt{n}} \rightsquigarrow \mathrm{cut}^{\mathtt{pp},\mathtt{np},\mathtt{np},\mathtt{s}}\left(\mathrm{in}^{\mathtt{np},\mathtt{pp}}\ \{\mathrm{slice}^{\mathtt{n}}\}^{\mathtt{np}}\right)\ \{\mathrm{meat}^{\mathtt{n}}\}^{\mathtt{np}}$$

- Map POS tags and shift to slightly Generalized POS tags: UPOS & Penn
- Use only these syntactic categories: $\mathtt{n}$, $\mathtt{np}_x$, $\mathtt{s}_x$, $\mathtt{pp}$, $\mathtt{pr}$
- Function words $\mapsto$ canonical terms (excl. prepositions)

Natural Tableau for Dutch ios/

vrolijk (een jongen (λx. een piano (λy. spelen y x))) : $\mathbb{T}$

1 happily (a boy (λx. a piano (λy. play y x))) : $\mathbb{T}$

een piano (worden (λy. een persoon (λx. door x bespelen y))) : $\mathbb{F}$

2 a piano (be (λy. a person (λx. by x play y))) : $\mathbb{F}$

[1]

3 een jongen (λx. een piano (λy. spelen y x)) : $\mathbb{T}$

[3]

4 jongen : [j] : $\mathbb{T}$

5 een piano (λy. spelen y j) : $\mathbb{T}$

[5]

6 piano : [p] : $\mathbb{T}$

7 spelen : [p, j] : $\mathbb{T}$

[2,6]

8 worden (λy. een persoon (λx. door x bespelen y)) : [p] : $\mathbb{F}$

[8]

9 λy. een persoon (λx. door x bespelen y) : [p] : $\mathbb{F}$

[9]

10 een persoon (λx. door x bespelen p) : $\mathbb{F}$

[10,j]

11 persoon : [j] : $\mathbb{F}$

[4,11]

×

12 λx. door x bespelen p : [j] : $\mathbb{F}$

[12]

13 door j bespelen p : $\mathbb{F}$

[13]

14 bespelen[p, j] : $\mathbb{F}$

[7,14]

×

1771 Entailment

Een jongen speelt vrolijk piano

en piano wordt bespeeld door een persoon

1771 Entailment

A boy is happily playing the piano

A piano is being played by a person

Experiments & Results

Parser / pos
Alpino / Alpino
Alpino / spaCy
NPN / Alpino
NPN / spaCy
LangPro 2×2

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	Tϵ
Alpino / Alpino	72.7
Alpino / spaCy	74.8
NPN / Alpino	72.0
NPN / spaCy	74.3
LangPro 2×2	76.0

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	Tϵ	Tα:Tϵ
Alpino / Alpino	72.7	(+9.3)
Alpino / spaCy	74.8	(+9.5)
NPN / Alpino	72.0	(+8.6)
NPN / spaCy	74.3	(+9.1)
LangPro 2×2	76.0	(+9.8)

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	Tϵ	Tα:Tϵ	Eϵ
Alpino / Alpino	72.7	(+9.3)	74.1
Alpino / spaCy	74.8	(+9.5)	75.9
NPN / Alpino	72.0	(+8.6)	72.8
NPN / spaCy	74.3	(+9.1)	75.0
LangPro 2×2	76.0	(+9.8)	77.1

- POS tagging: spaCy is $1.7_\leqslant$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_\leqslant$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	Tϵ	Tα:Tϵ	Eϵ	Tα:Eϵ
Alpino / Alpino	72.7	(+9.3)	74.1	(+1.8) 75.9
Alpino / spaCy	74.8	(+9.5)	75.9	(+1.7) 77.6
NPN / Alpino	72.0	(+8.6)	72.8	(+1.5) 74.3
NPN / spaCy	74.3	(+9.1)	75.0	(+1.4) 76.4
LangPro 2×2	76.0	(+9.8)	77.1	(+1.6) 78.7

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	$T\epsilon$	$T\alpha{:}T\epsilon$	$E\epsilon$	$T\alpha{:}E\epsilon$
Alpino / Alpino	72.7	(+9.3)	74.1	(+1.8) 75.9
Alpino / spaCy	74.8	(+9.5)	75.9	(+1.7) 77.6
NPN / Alpino	72.0	(+8.6)	72.8	(+1.5) 74.3
NPN / spaCy	74.3	(+9.1)	75.0	(+1.4) 76.4
LangPro 2×2	76.0	(+9.8)	77.1	(+1.6) 78.7

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	$T\epsilon$	$T\alpha{:}T\epsilon$	$E\epsilon$	$T\alpha{:}E\epsilon$
Alpino / Alpino	72.7	(+9.3)	74.1	(+1.8) 75.9
Alpino / spaCy	74.8	(+9.5)	75.9	(+1.7) 77.6
NPN / Alpino	72.0	(+8.6)	72.8	(+1.5) 74.3
NPN / spaCy	74.3	(+9.1)	75.0	(+1.4) 76.4
LangPro 2×2	76.0	(+9.8)	77.1	(+1.6) 78.7

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	$T\epsilon$	$T\alpha$:$T\epsilon$	$E\epsilon$	$T\alpha$:$E\epsilon$	
Alpino / Alpino	72.7	(+9.3)	74.1	(+1.8)	75.9
Alpino / spaCy	74.8	(+9.5)	75.9	(+1.7)	77.6
NPN / Alpino	72.0	(+8.6)	72.8	(+1.5)	74.3
NPN / spaCy	74.3	(+9.1)	75.0	(+1.4)	76.4
LangPro 2×2	76.0	(+9.8)	77.1	(+1.6)	78.7

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Experiments & Results

Parser / pos	Tϵ	Tα:Tϵ	Eϵ	Tα:Eϵ
Alpino / Alpino	72.7	(+9.3)	74.1	(+1.8) 75.9
Alpino / spaCy	74.8	(+9.5)	75.9	(+1.7) 77.6
NPN / Alpino	72.0	(+8.6)	72.8	(+1.5) 74.3
NPN / spaCy	74.3	(+9.1)	75.0	(+1.4) 76.4
LangPro 2×2	76.0	(+9.8)	77.1	(+1.6) 78.7

- POS tagging: spaCy is $1.7_{\leqslant}$ better than Alpino;
- Parsers: Alpino > NPN? Sentences parsed 98.6% vs 94.9%;
- The ensemble is $1.2_{\leqslant}$ better than the best monoLP;
- The number of false proofs increases by 4% after αbductive learning.

Comparison to the transformer-based NLI models

Models	All	Ent	Cont
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0	86.2	86.7
mBERT	79.9	79.0	81.9
RobBert	81.7	76.9	85.3

Comparison to the transformer-based NLI models

Models	All $\pm\Delta$	Ent	Cont
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0 -0.3	86.2	86.7
mBERT	79.9 $+0.7$	79.0	81.9
RobBert	81.7 $+0.9$	76.9	85.3

Comparison to the transformer-based NLI models

Models	All $\pm\Delta$	Ent $\pm\Delta$	Cont
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0 -0.3	86.2 $+2.0$	86.7
mBERT	79.9 $+0.7$	79.0 $+4.7$	81.9
RobBert	81.7 $+0.9$	76.9 $+6.4$	85.3

Comparison to the transformer-based NLI models

Models	All $\pm\Delta$	Ent $\pm\Delta$	Cont $\pm\Delta$
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0 −0.3	86.2 +2.0	86.7 +0.8
mBERT	79.9 +0.7	79.0 +4.7	81.9 +3.1
RobBert	81.7 +0.9	76.9 +6.4	85.3 +1.1

Comparison to the transformer-based NLI models

Models	All $\pm\Delta$	Ent $\pm\Delta$	Cont $\pm\Delta$
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0 −0.3	86.2 +2.0	86.7 +0.8
mBERT	79.9 +0.7	79.0 +4.7	81.9 +3.1
RobBert	81.7 +0.9	76.9 +6.4	85.3 +1.1

Problems failed by all three DL models but solved by LangPro:

$\boxed{1556}$ Entailment

A man is carrying a tree

A man is carrying a plant

$\boxed{175}$ Entailment

A family is watching a little boy who is hitting a baseball

A boy is hitting a baseball

Comparison to the transformer-based NLI models

Models	All ±Δ	Ent ±Δ	Cont ±Δ
LangPro 2×2	78.7	50.6	66.3
BERTje	82.0 −0.3	86.2 +2.0	86.7 +0.8
mBERT	79.9 +0.7	79.0 +4.7	81.9 +3.1
RobBert	81.7 +0.9	76.9 +6.4	85.3 +1.1

Problems failed by all three DL models but solved by LangPro:

1556 Entailment
A man is carrying a tree
A man is carrying a plant

175 Entailment
A family is watching a little boy who is hitting a baseball
A boy is hitting a baseball

4092 Contradiction
The person is not drawing
A man is drawing a picture

6356 Contradiction
A woman in a red dress is putting away an instrument
A woman in a red dress is playing an instrument

Findings: SICK NL & Open Dutch WN

SICK NL is more challenging that the original dataset due to MT:

- Transferred gold labels are not gold:

$\boxed{3181}$ Neutral?
A man is trekking in the woods
The man is not hiking in the woods

$\mapsto$

$\boxed{3181}$ Neutral
Een man is aan het wandelen in het bos
De man is niet aan het wandelen in het bos

- Extra reasoning due to translation shifts:
drawing a picture $\mapsto$ een tekening maakt | tekent een foto
dirt bike race $\mapsto$ crossmotorwedstrijd | crossmotorrace.

Lexical relations learned from the training set:

lopen $\equiv$ rennen
pizza $\sqsubseteq$ voedsel/food
halter/dumbbell $\sqsubseteq$ gewicht/weight
leeg/empty | vol/full

Findings: SICK NL & Open Dutch WN

SICK NL is more challenging that the original dataset due to MT:

- Transferred gold labels are not gold:

3181 Neutral?		3181 Neutral
A man is trekking in the woods	$\mapsto$	Een man is aan het wandelen in het bos
The man is not hiking in the woods		De man is niet aan het wandelen in het bos

- Extra reasoning due to translation shifts:
  drawing a picture $\mapsto$ een tekening maakt | tekent een foto
  dirt bike race $\mapsto$ crossmotorwedstrijd | crossmotorrace.

Lexical relations learned from the training set:

lopen $\equiv$ rennen

halter/dumbbell $\sqsubseteq$ gewicht/weight

pizza $\sqsubseteq$ voedsel/food

leeg/empty | vol/full

Findings: SICK NL & Open Dutch WN

SICK NL is more challenging that the original dataset due to MT:

- Transferred gold labels are not gold:

3181 Neutral?		3181 Neutral
A man is trekking in the woods	$\mapsto$	Een man is aan het wandelen in het bos
The man is not hiking in the woods		De man is niet aan het wandelen in het bos

- Extra reasoning due to translation shifts:
  drawing a picture $\mapsto$ een tekening maakt | tekent een foto
  dirt bike race $\mapsto$ crossmotorwedstrijd | crossmotorrace.

Lexical relations learned from the training set:

lopen $\equiv$ rennen

halter/dumbbell $\sqsubseteq$ gewicht/weight

pizza $\sqsubseteq$ voedsel/food

leeg/empty | vol/full

Conclusion & Future work

- First Dutch NLI system based on logic
- YES! Logic-based Reasoning Works for Dutch: promising results
- Automatic translation makes the NLI data more challenging

Conclusion & Future work

- First Dutch NLI system based on logic
- YES! Logic-based Reasoning Works for Dutch: promising results
- Automatic translation makes the NLI data more challenging

- Employ the Dutch CCG parser
- Qualitative comparison of the Dutch syntactic parsers
- Multilingual LangPro: SICK + CCG parsers for Logical Forms