Combining Axiom Injection and Knowledge Base Completion for Efficient Natural Language Inference

Masashi Yoshikawa[◆], Koji Mineshima[★], Hiroshi Noji[♥], Daisuke Bekki[★]
◆ Nara Institute of Science and Technology
★ Ochanomizu University
♥ Artificial Intelligence Research Center, AIST
AAAI-33 2019/1/31

NAIST®

Ochanomizu University

National Institute of Advanced Industrial Science and Technology AIST

Recognizing Textual Entailment

a.k.a. Natural Language Inference

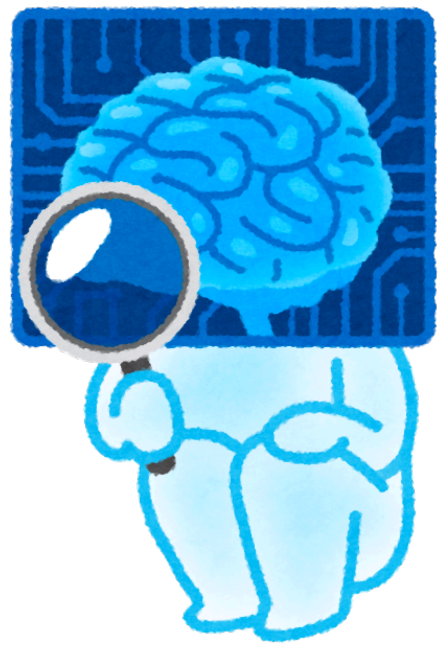
Premise(s)

P1: Clients at the demonstration were all impressed by the system's performance.

P2: Smith was a client at the demonstration.

Hypothesis

H: Smith was impressed by the system's performance.

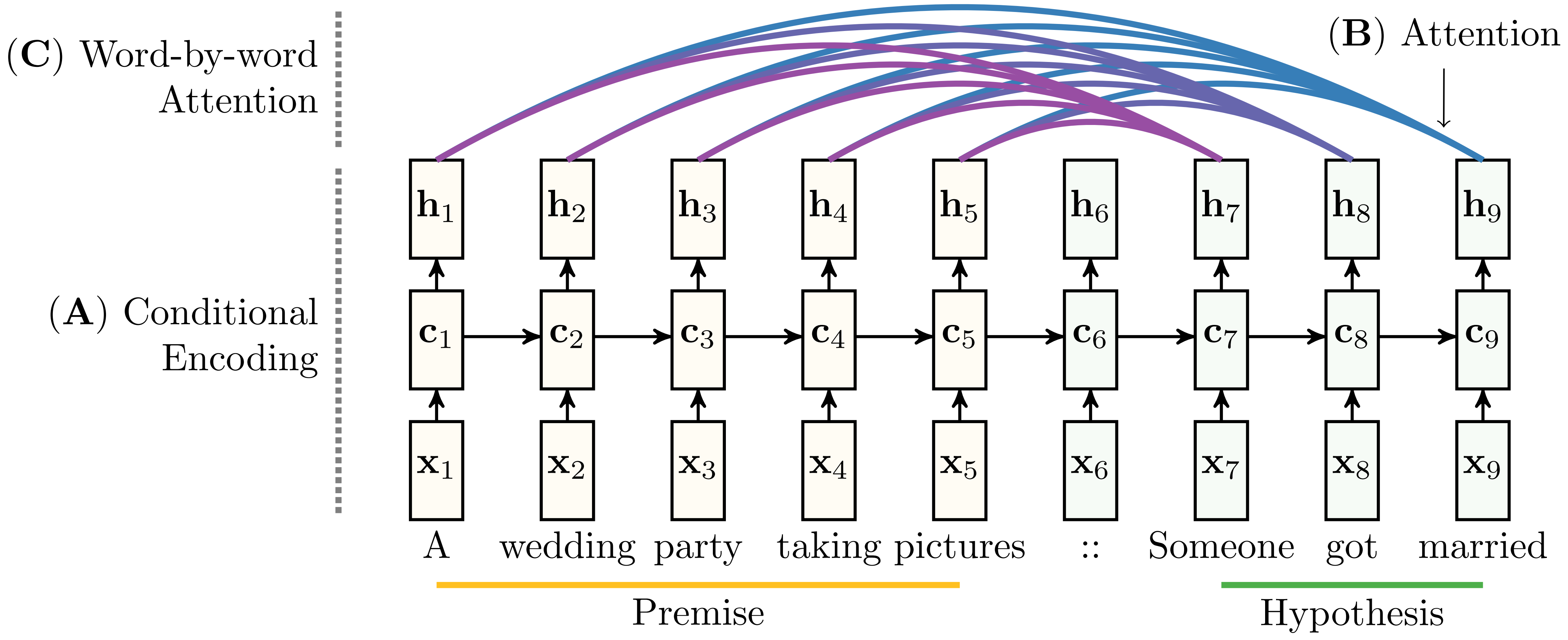


`{entailment, contradiction, unknown}`

- A testbed to evaluate if a machine can reason as we do
  - lexical, logical, syntactic phenomena, etc.
- Elemental technology for improving other NLP tasks
  - Question answering, reading comprehension, etc.

Approaches to RTE

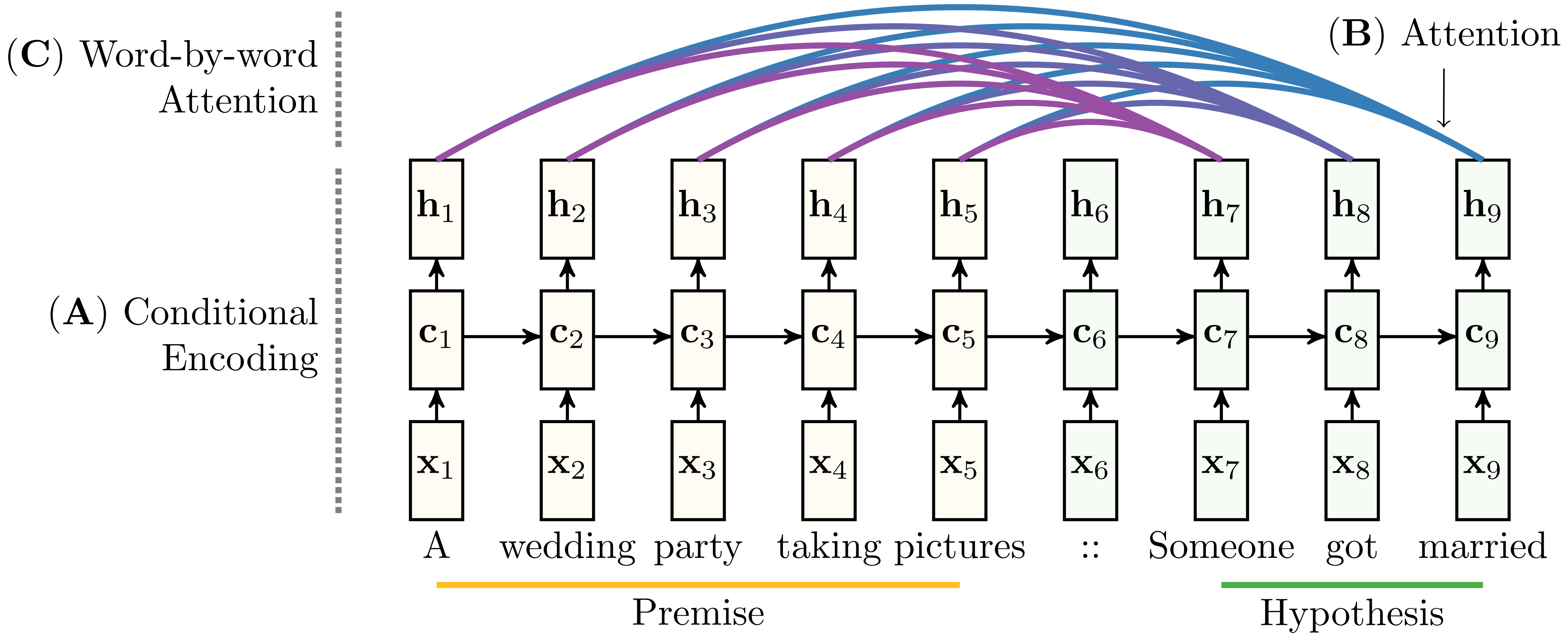
- Machine learning (Rocktäschel et al., 2016, etc.)
  - e.g. Neural Networks



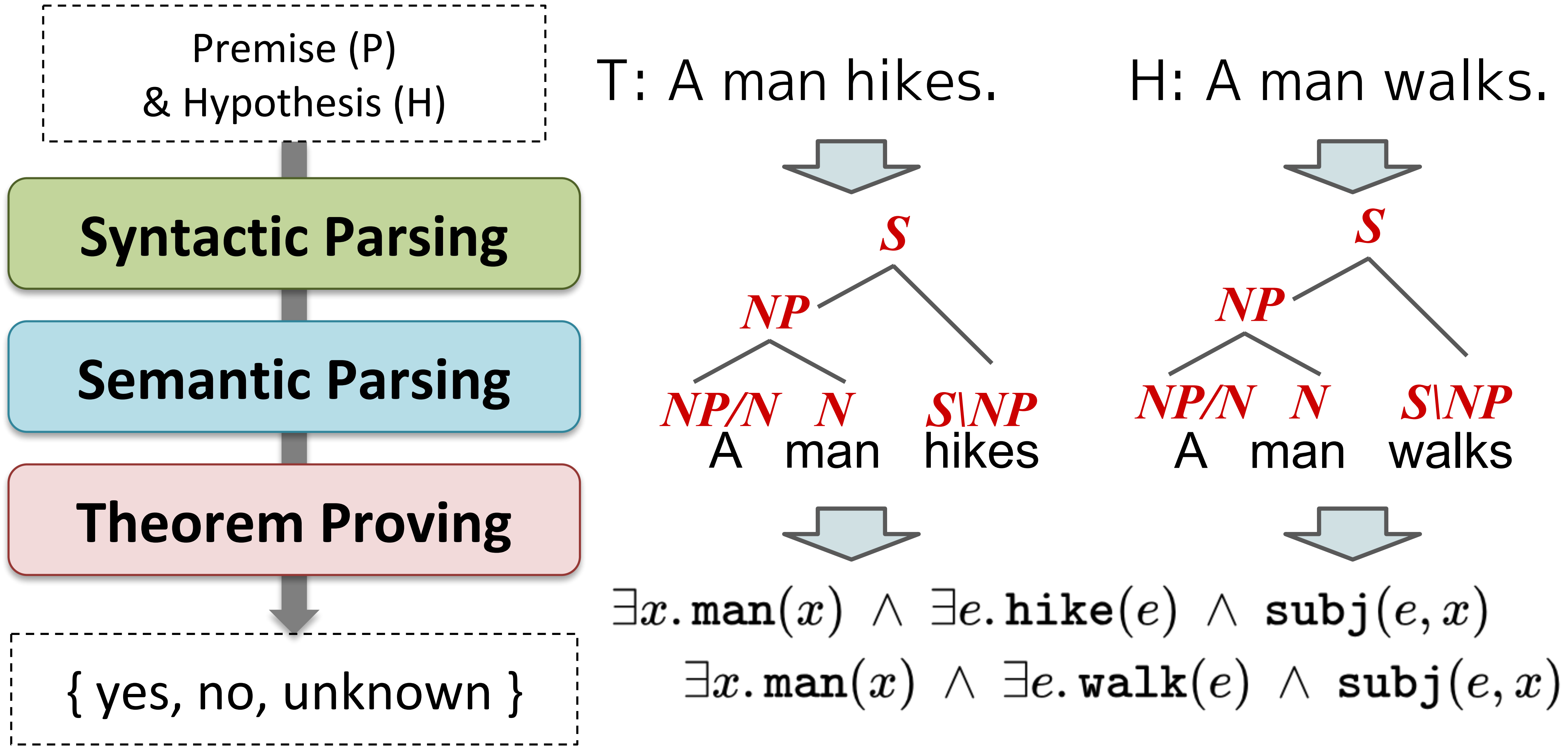
Rocktäschel et al., 2016

Approaches to RTE

- Machine learning (Rocktäschel et al., 2016, etc.)
  - e.g. Neural Networks
- Logic (Mineshima et al., 2015, Abzianidze 2017, etc)
  - Traditional pipeline systems
  - Theorem prover (e.g. Coq)



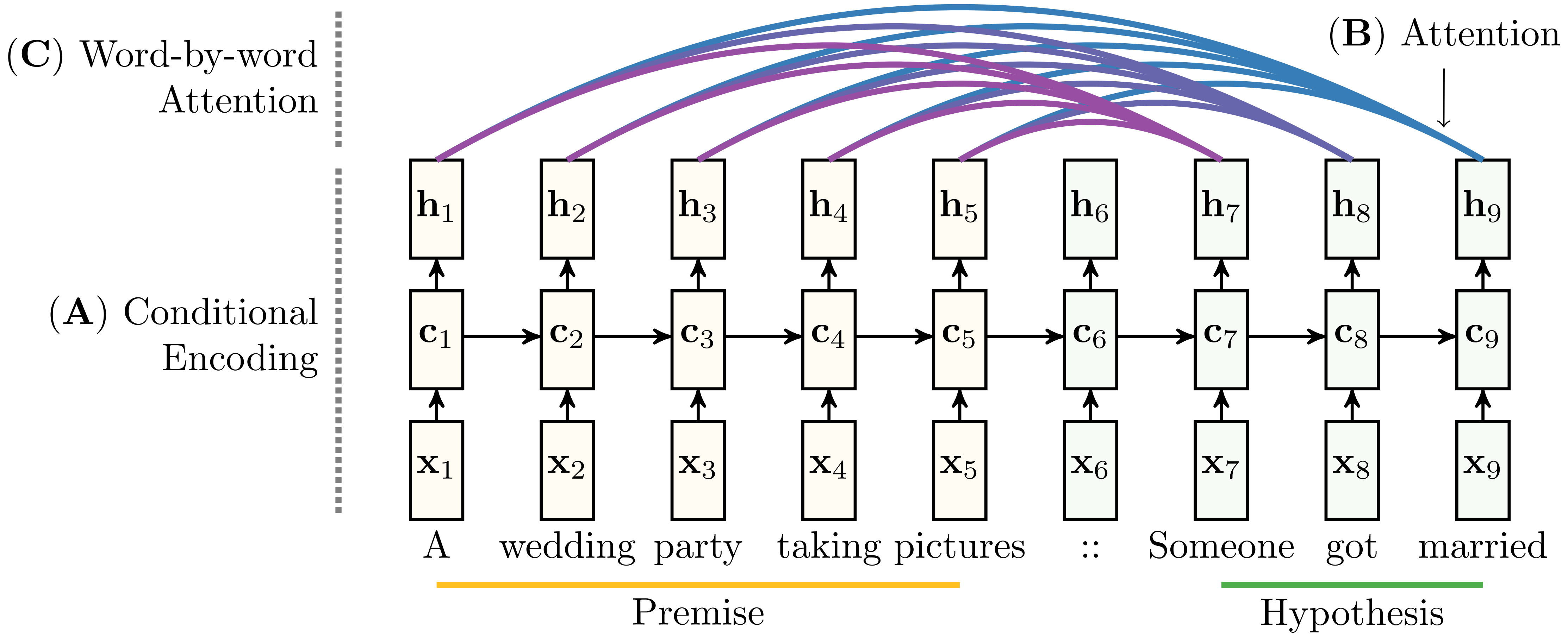
Rocktäschel et al., 2016



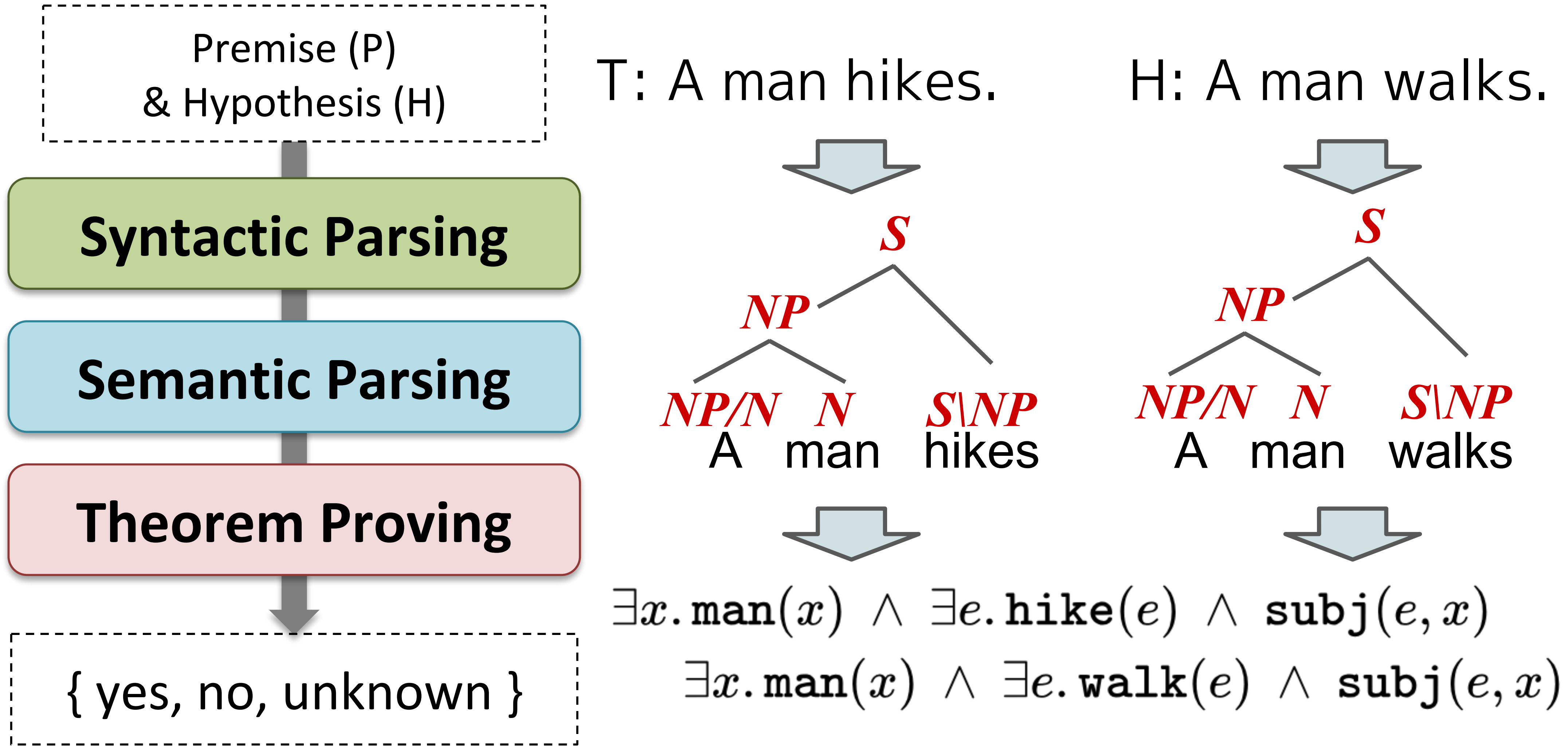
Mineshima et al., 2015

Approaches to RTE

- Machine learning (Rocktäschel et al., 2016, etc.)
  - e.g. Neural Networks
- Logic (Mineshima et al., 2015, Abzianidze 2017, etc)
  - Traditional pipeline systems
  - Theorem prover (e.g. Coq)
- **Ours**: logic-based, extended by ML! (Hybrid)

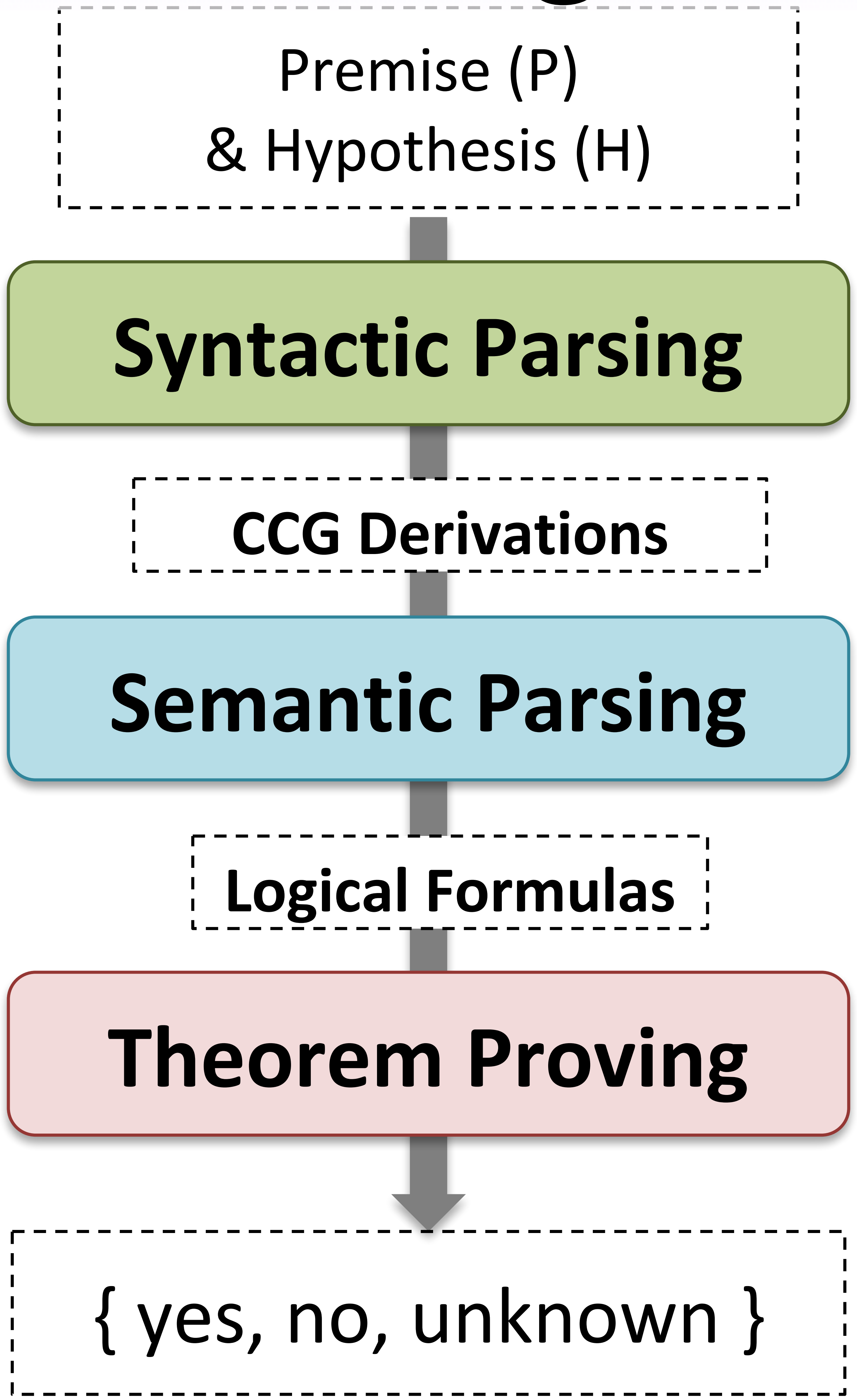


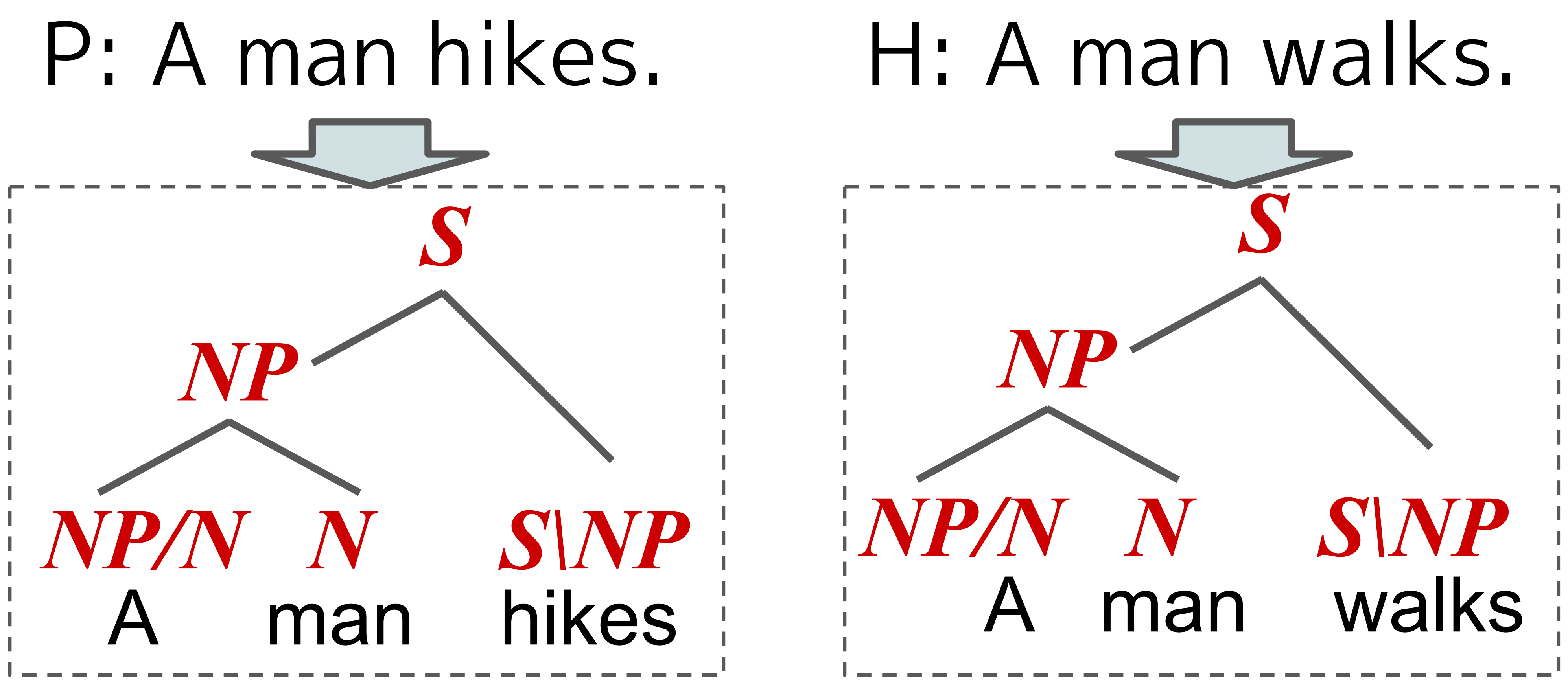
Rocktäschel et al., 2016



Mineshima et al., 2015

ccg2lambda (Mineshima et al., 2015)





$$\exists x.\mathtt{man}(x) \wedge \exists e.\mathtt{hike}(e) \wedge \mathtt{subj}(e,x)$$

$$\exists x.\mathtt{man}(x) \wedge \exists e.\mathtt{walk}(e) \wedge \mathtt{subj}(e,x)$$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq theorem prover

result: unknown

ccg2lambda (Mineshima et al., 2015)

Premise (P)
& Hypothesis (H)

Syntactic Parsing

CCG Derivations

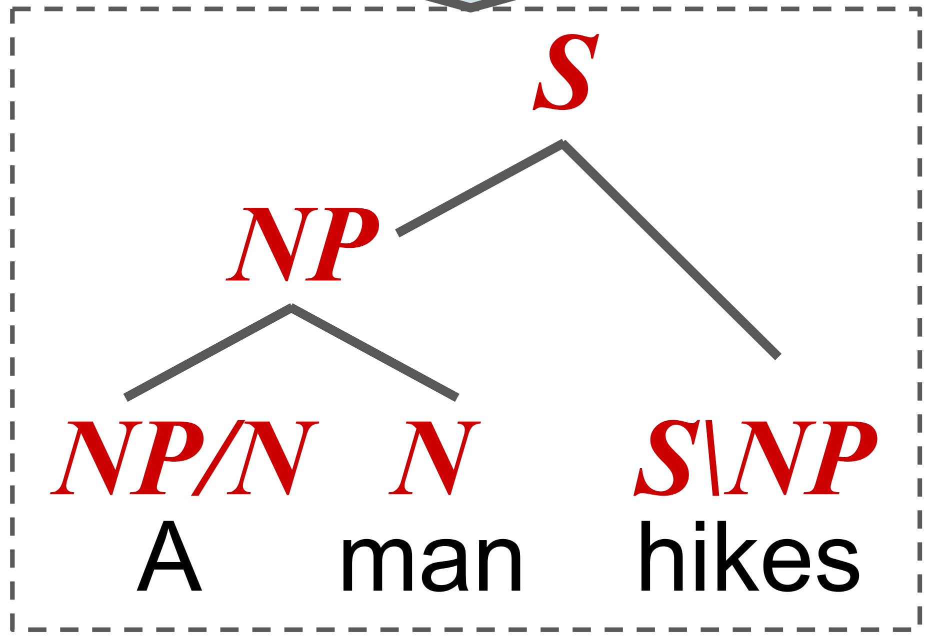
Semantic Parsing

Logical Formulas

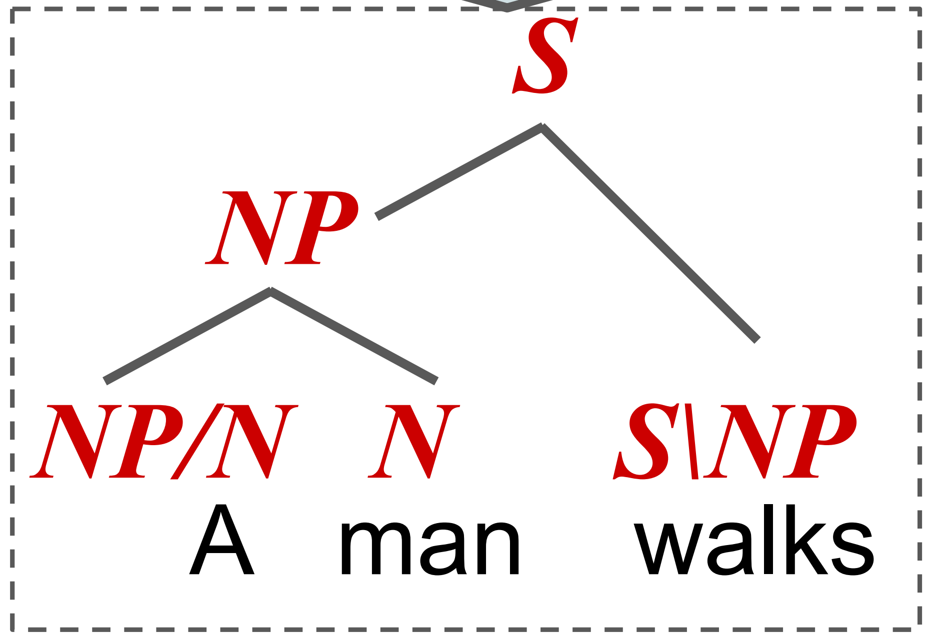
Theorem Proving

{ yes, no, unknown }

P: A man hikes.



H: A man walks.



$$\exists x.\mathbf{man}(x) \wedge \exists e.\mathbf{hike}(e) \wedge \mathbf{subj}(e,x)$$

$$\exists x.\mathbf{man}(x) \wedge \exists e.\mathbf{walk}(e) \wedge \mathbf{subj}(e,x)$$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

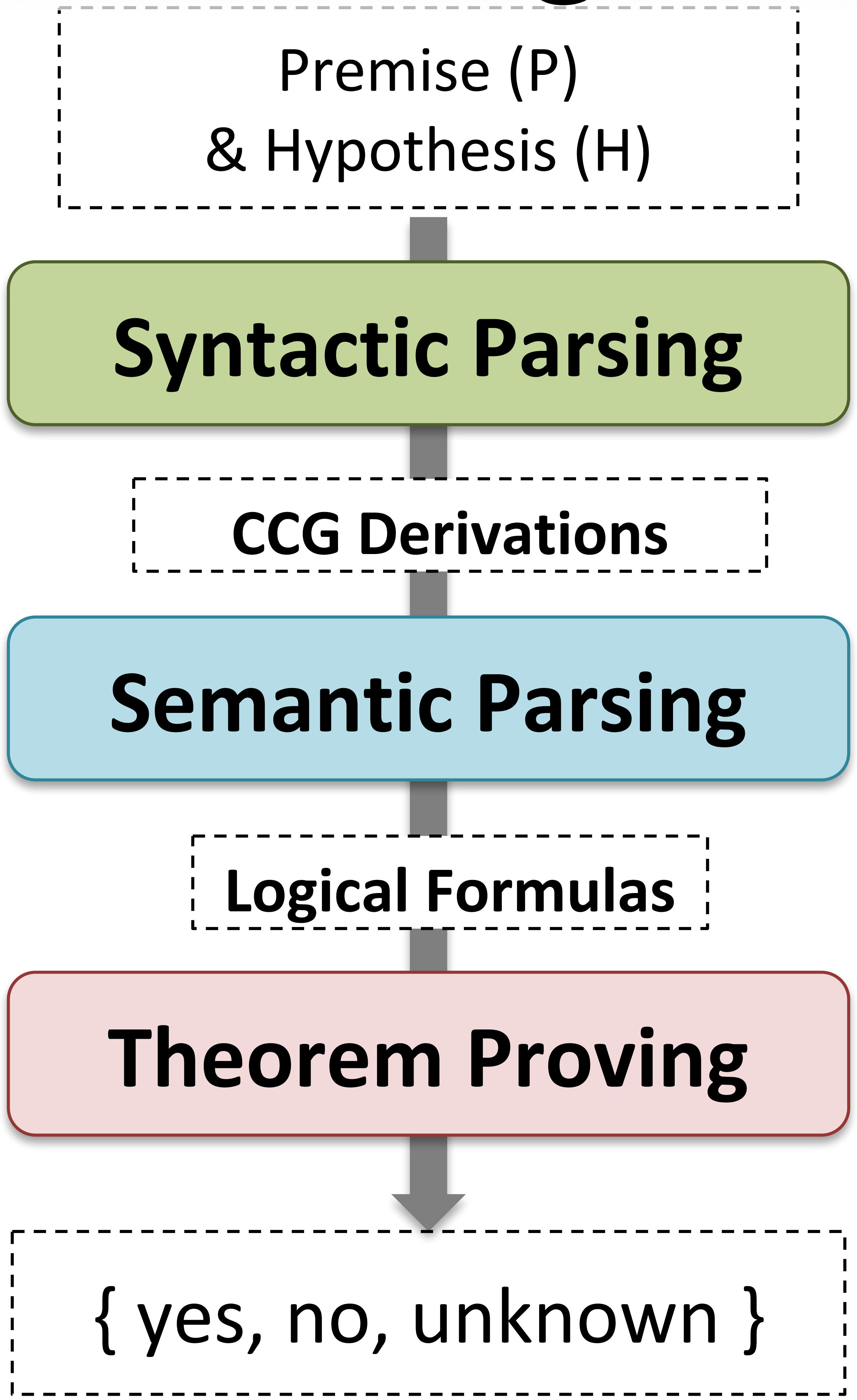
Coq theorem prover

result: unknown

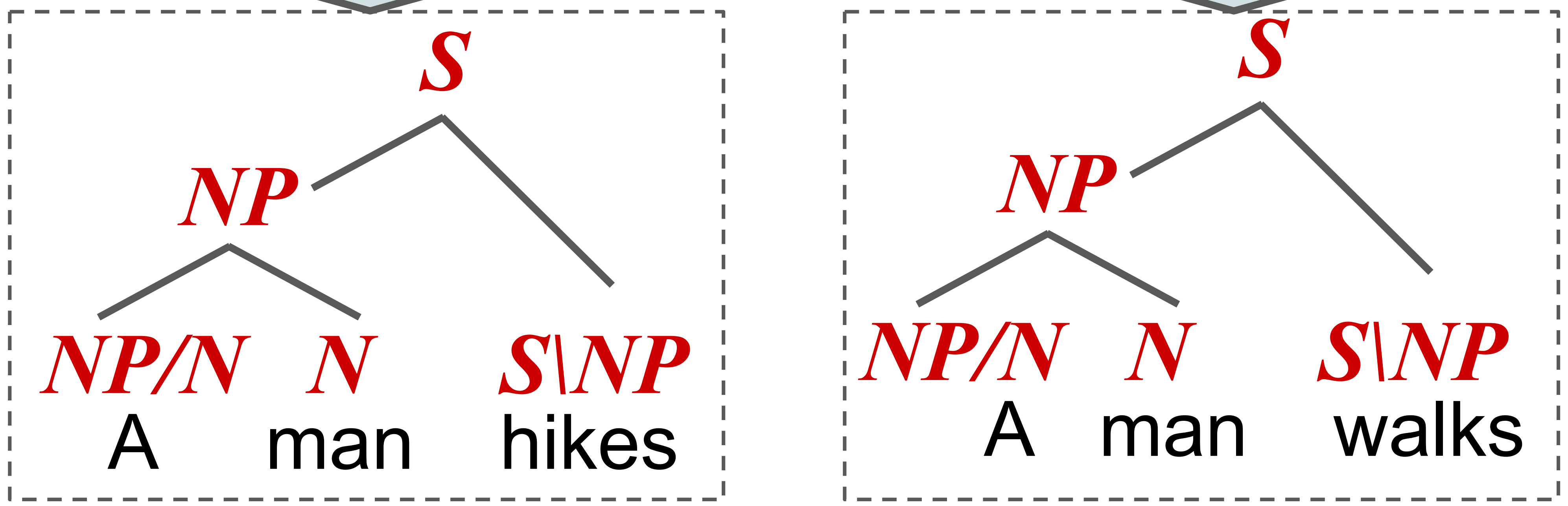
👍 Unsupervised

👍 Captures linguistic phenomena
- 83.6 % accuracy in SICK

ccg2lambda (Mineshima et al., 2015)



P: A man hikes. H: A man walks.



$\exists x.\mathbf{man}(x) \ \wedge\ \exists e.\mathbf{hike}(e) \ \wedge\ \mathbf{subj}(e,x)$

$\exists x.\mathbf{man}(x) \ \wedge\ \exists e.\mathbf{walk}(e) \ \wedge\ \mathbf{subj}(e,x)$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq theorem prover

result: unknown

👍 Unsupervised

👍 Captures linguistic phenomena
- 83.6 % accuracy in SICK

🤔 How to handle external knowledge?
e.g. $\forall x\,.\,\mathbf{hike}(x) \rightarrow \mathbf{walk}(x)$
- Use WordNet as axioms blows up the search space of theorem proving!

"Abduction" Mechanism (Martínez-Gómez et al., 2017)

Premise (P)
& Hypothesis (H)

Syntactic Parsing

CCG Derivations

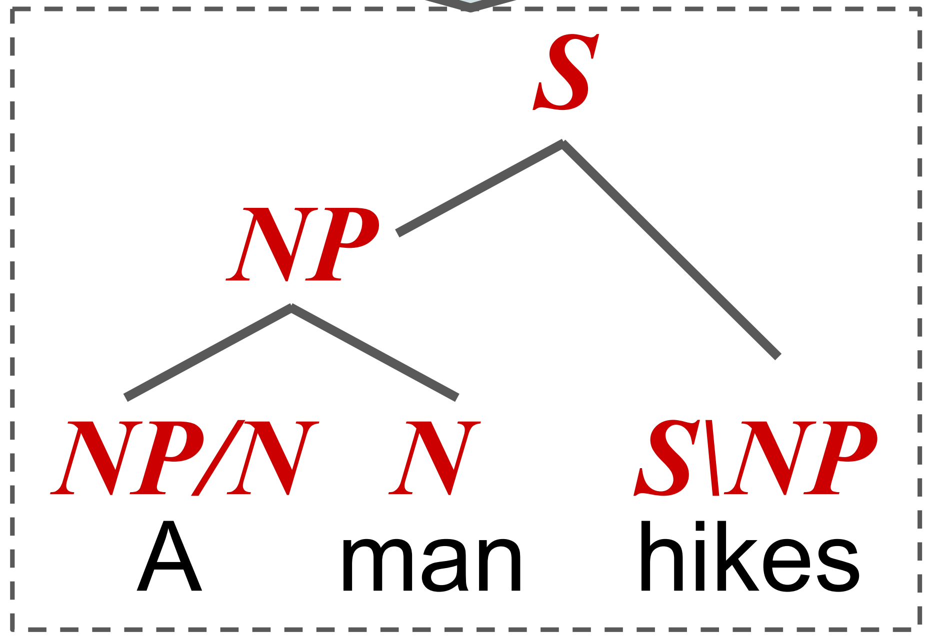
Semantic Parsing

Logical Formulas

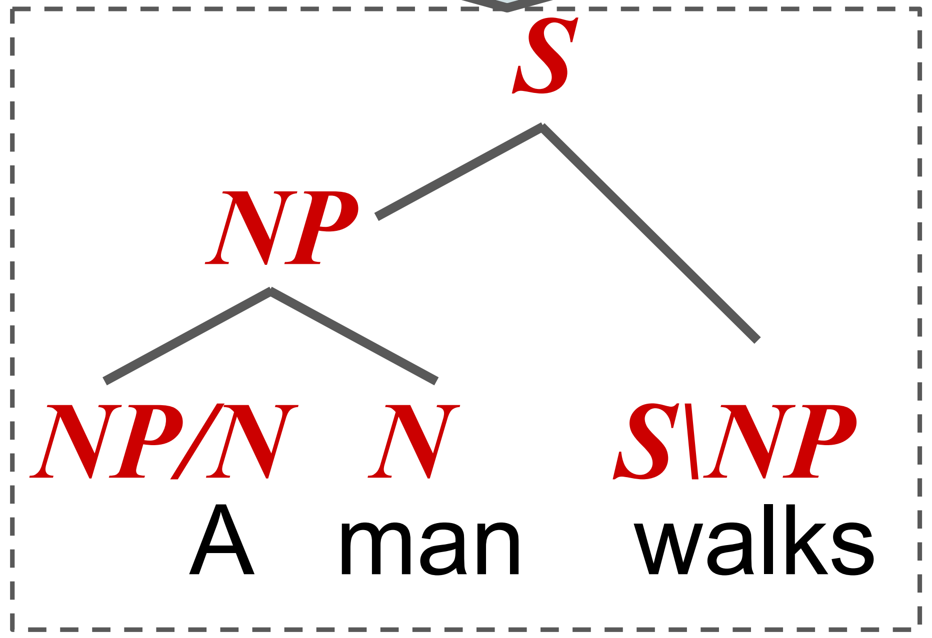
Theorem Proving

{ yes, no, unknown }

P: A man hikes.



H: A man walks.



$$\exists x.\mathbf{man}(x) \wedge \exists e.\mathbf{hike}(e) \wedge \mathbf{subj}(e,x)$$

$$\exists x.\mathbf{man}(x) \wedge \exists e.\mathbf{walk}(e) \wedge \mathbf{subj}(e,x)$$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq theorem prover

result: unknown

"Abduction" Mechanism (Martínez-Gómez et al., 2017)

$\exists x.\mathtt{man}(x) \wedge \exists e.\mathtt{walk}(e) \wedge \mathtt{subj}(e,x)$

Logical Formulas

Theorem Proving

{ yes, no, unknown }

Search on KBs

PRINCETON UNIVERSITY
WordNet
A lexical database for English

New Axioms

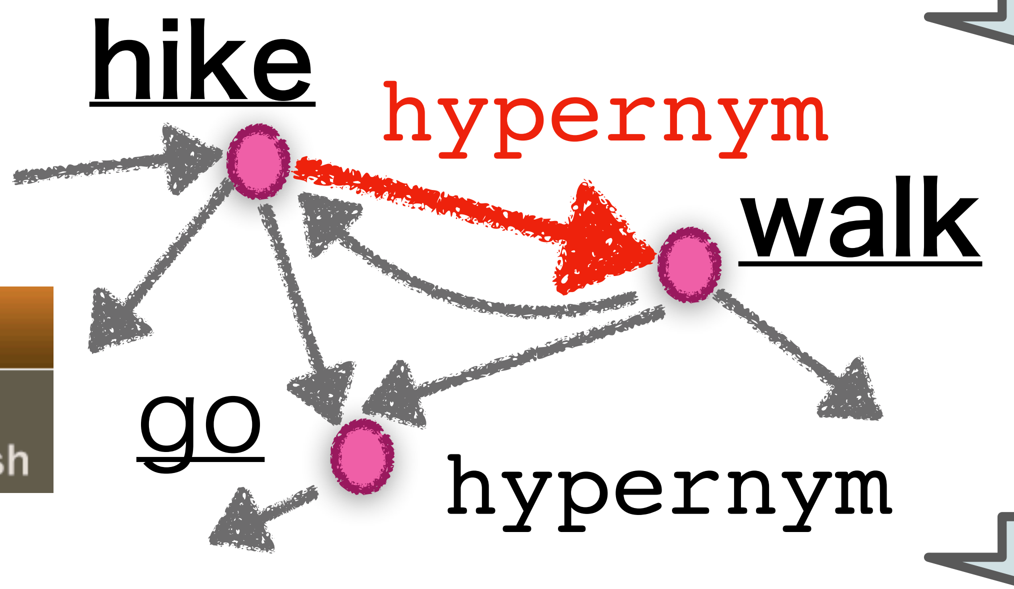
Theorem Proving

{ yes, no, unknown }

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq theorem prover

result: unknown



$\forall e.\,\mathtt{hike}(e) \rightarrow \mathtt{walk}(e)$

```
Coq < Axiom ax1: forall x: Event, hike e -> walk e.
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq theorem prover

result: yes

More steps when the 1st theorem proving is unsuccessful

1. Search KBs (e.g. WordNet) for useful lexical relations
2. Rerun Coq with additional axioms

"Abduction" Mechanism (Martínez-Gómez et al., 2017)

- Promising approach to handling external knowledge within a logic-based system

"Abduction" Mechanism (Martínez-Gómez et al., 2017)

- Promising approach to handling external knowledge within a logic-based system
- (However,) **Practical issues**:
  - We want to **add more knowledge** to increase the coverage of reasoning
  - We want the **KBs to be compact** for efficient inference & memory usage

"Abduction" Mechanism (Martínez-Gómez et al., 2017)

- Promising approach to handling external knowledge within a logic-based system
- (However,) **Practical issues**:
  - We want to **add more knowledge** to increase the coverage of reasoning
  - We want the **KBs to be compact** for efficient inference & memory usage
- Do not want to run Coq again and again for real applications 😣
  - Ideally, the mechanism should be tightly integrated with the inference for effciency

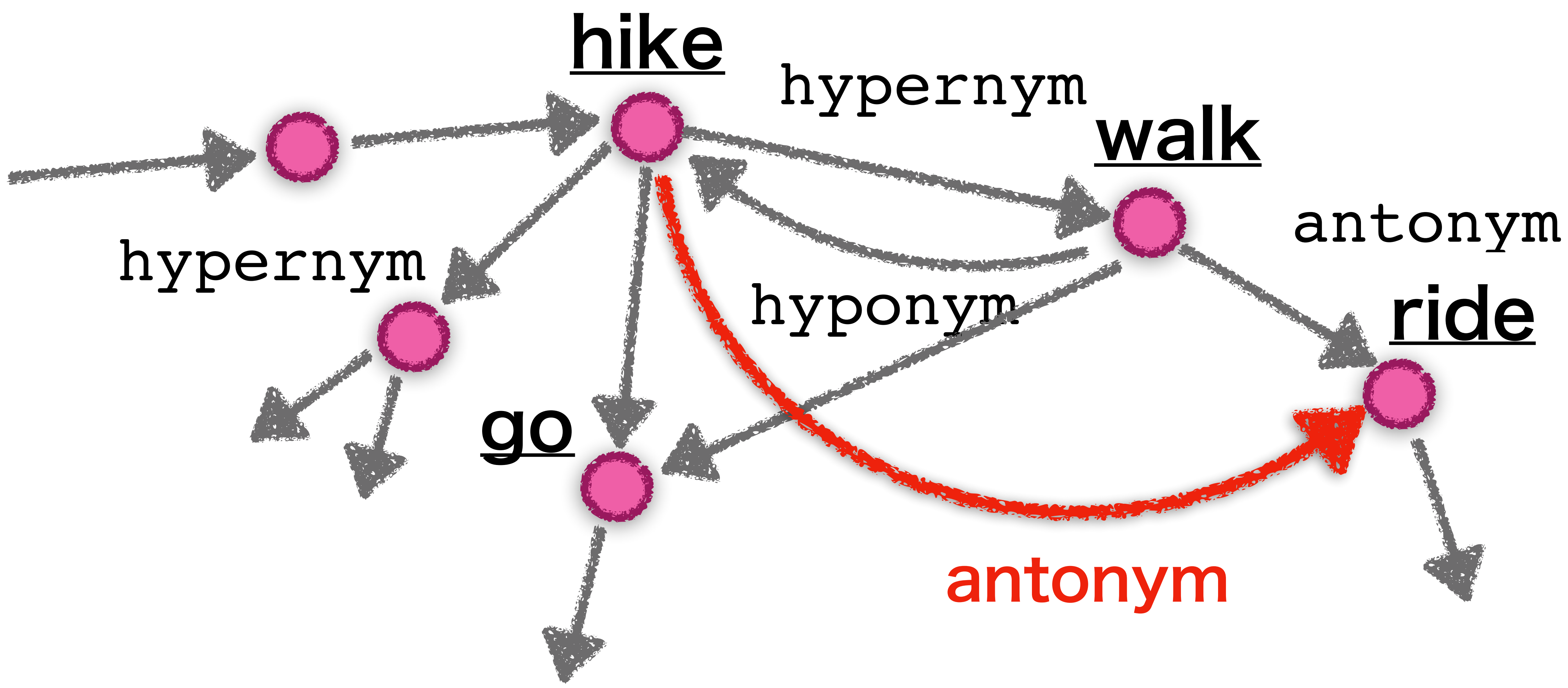
"Abduction" Mechanism (Martínez-Gómez et al., 2017)

- Promising approach to handling external knowledge within a logic-based system
- (However,) **Practical issues**:
  - We want to **add more knowledge** to increase the coverage of reasoning
  - We want the **KBs to be compact** for efficient inference & memory usage
- Do not want to run Coq again and again for real applications 😣
  - Ideally, the mechanism should be tightly integrated with the inference for effciency

👉 **We solve these issues by:**

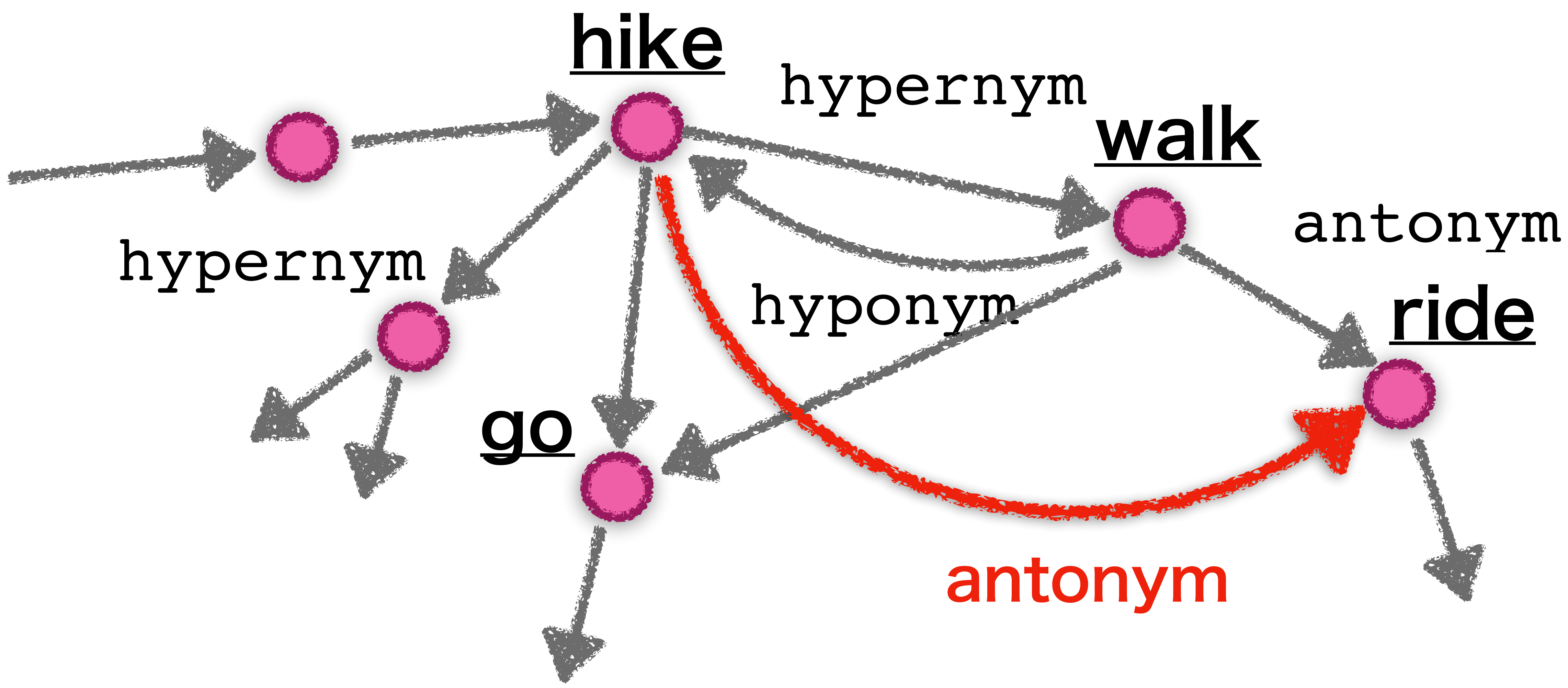
1. **Replacing search on KBs by techniques of "Knowledge Base Completion"**
2. **Developing "`abduction`" Coq plugin**

1. Extending Abduction Mechanism with KBC



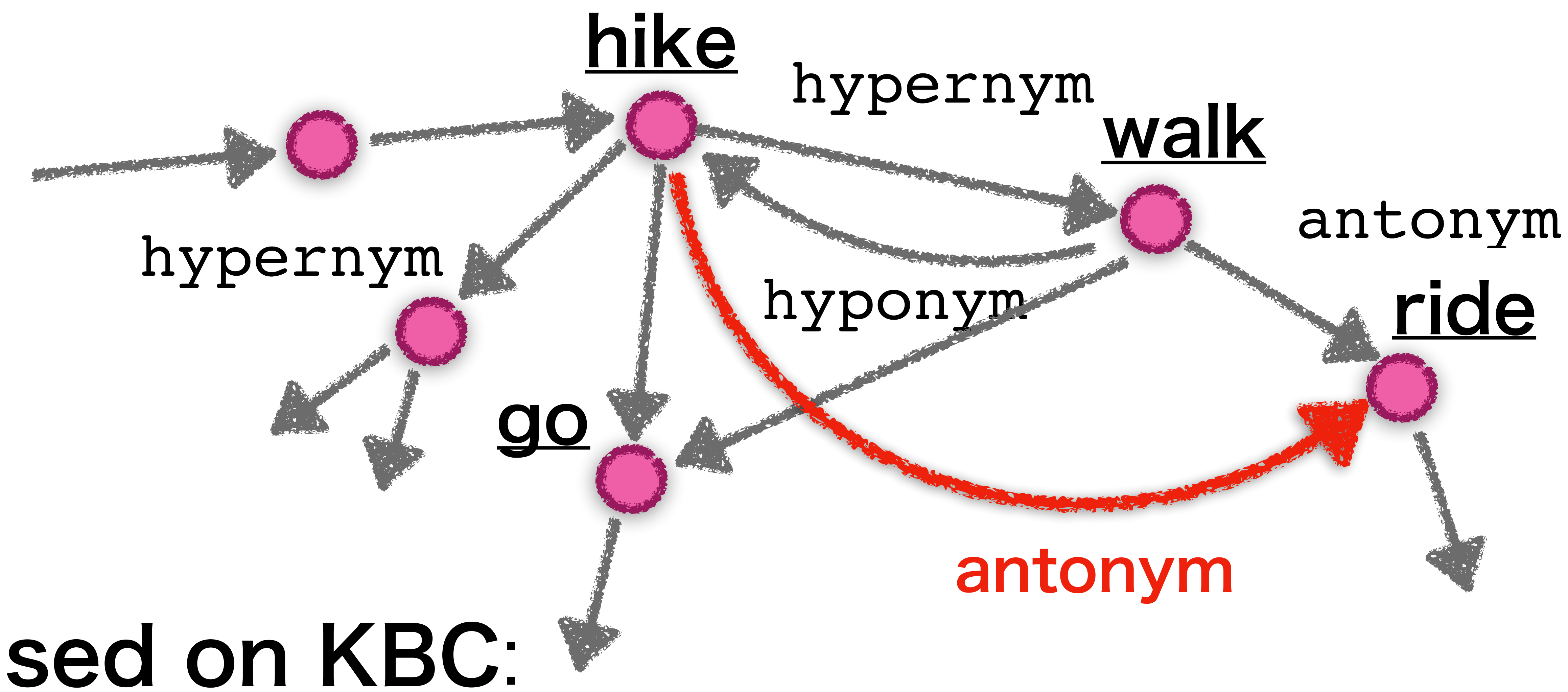
1. Extending Abduction Mechanism with KBC

- **Knowledge Base Completion**:
  - A task to complement missing relations
  - recent huge advancement

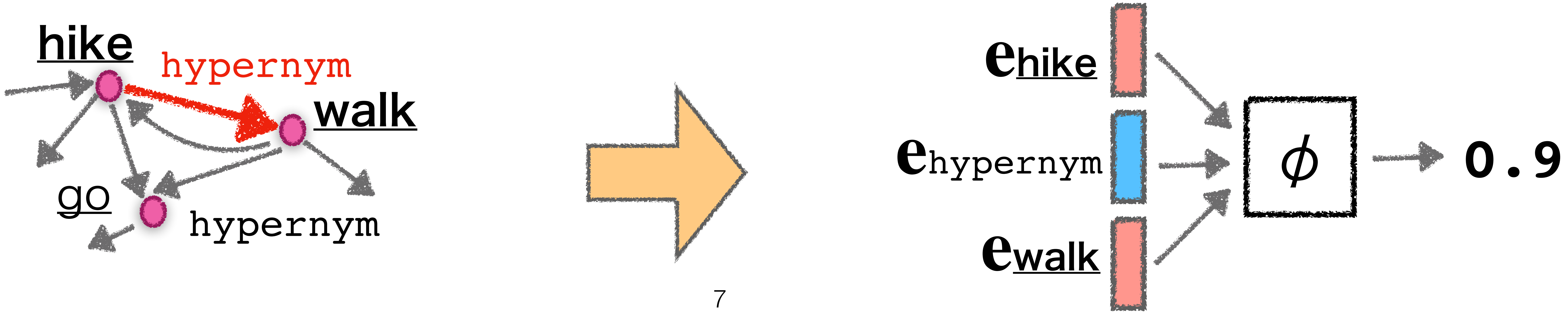


1. Extending Abduction Mechanism with KBC

- **Knowledge Base Completion**:
  - A task to complement missing relations
  - recent huge advancement

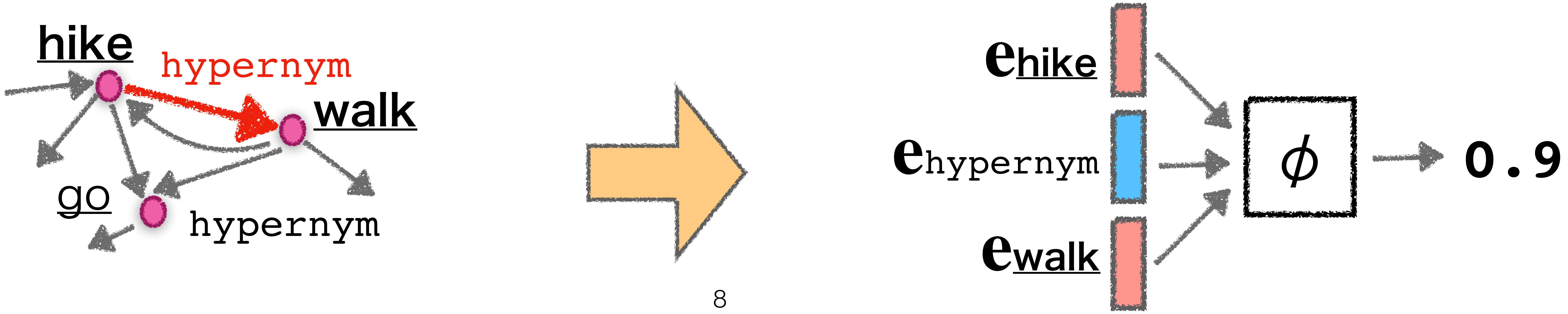


- We propose an **abduction mechanism based on KBC**:
  - If (s, r, o) is missing, use it as axiom if $\phi(s, r, o) \geq \delta$ (threshold)
  - ComplEx (Trouillon et al., 2016): $\phi(s, r, o) = \sigma(Re(\langle \mathbf{e}_s, \mathbf{e}_r, \mathbf{e}_o \rangle)), \forall \mathbf{e}_v \in \mathbb{C}^n$



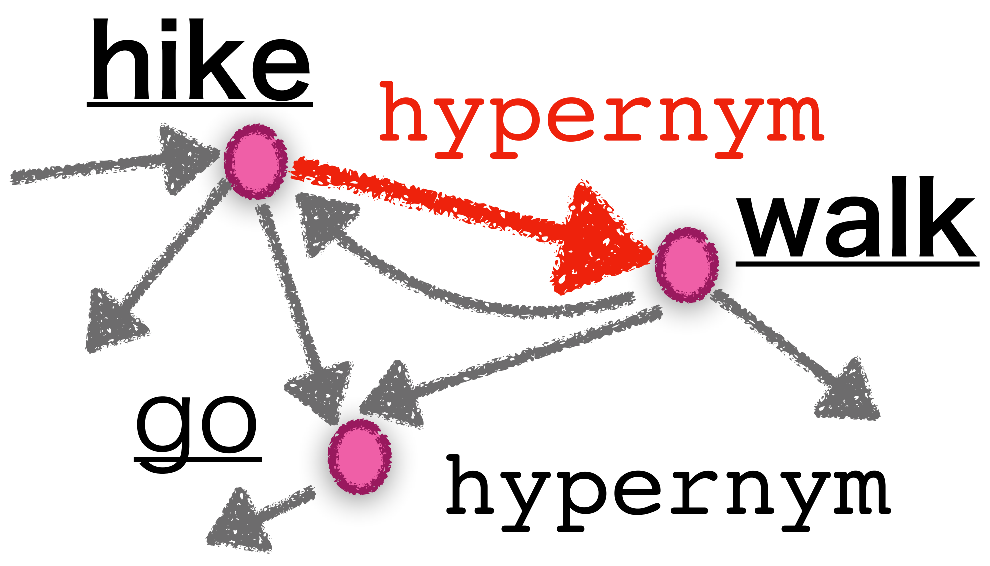
1. Extending Abduction Mechanism with KBC

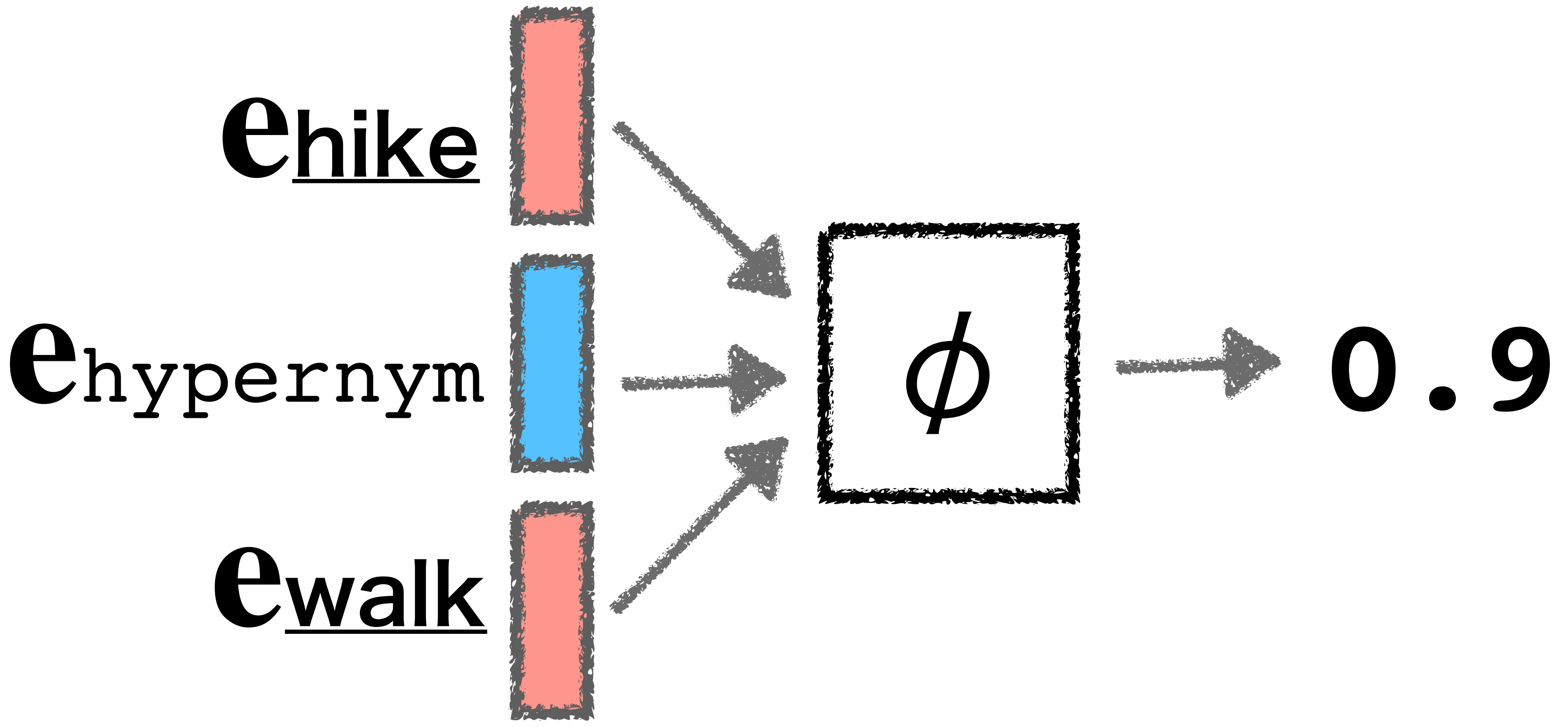
	Search on KB	KBC
Latent Knowledge	Hand-crafted rules (e.g. transitive closure of `hypernym`)	KBC models learn accurately
Efficiency	Multi-hop reasoning takes time	One dot product (ComplEx)
Scalability	Adding more knowledge harms the search time	Knowledge from VerbOcean (Chklovski et al., 2004) are added for free



1. Extending Abduction Mechanism with KBC

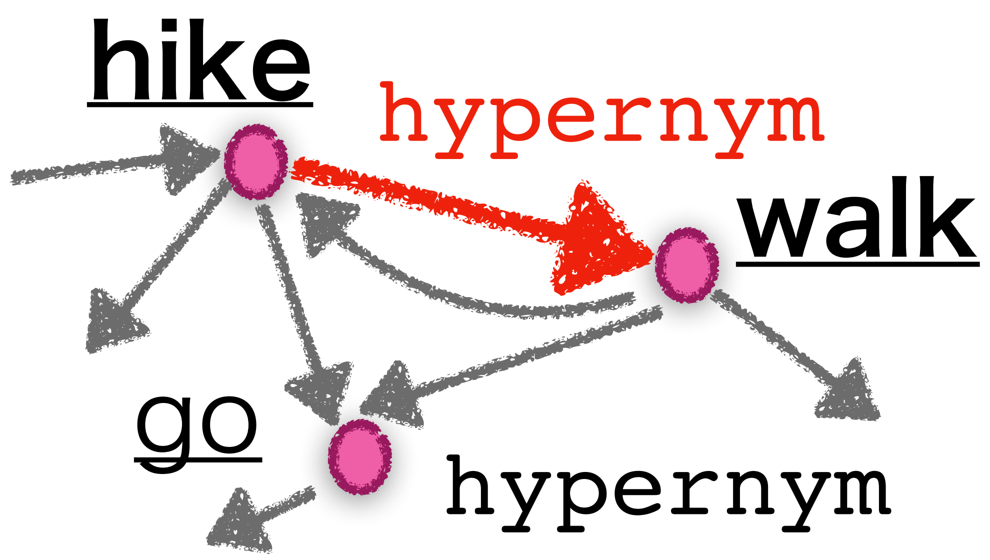
	Search on KB	KBC
Latent Knowledge	Hand-crafted rules (e.g. transitive closure of `hypernym`)	KBC models learn accurately
Efficiency	Multi-hop reasoning takes time	One dot product (ComplEx)
Scalability	Adding more knowledge harms the search time	Knowledge from VerbOcean (Chklovski et al., 2004) are added for free

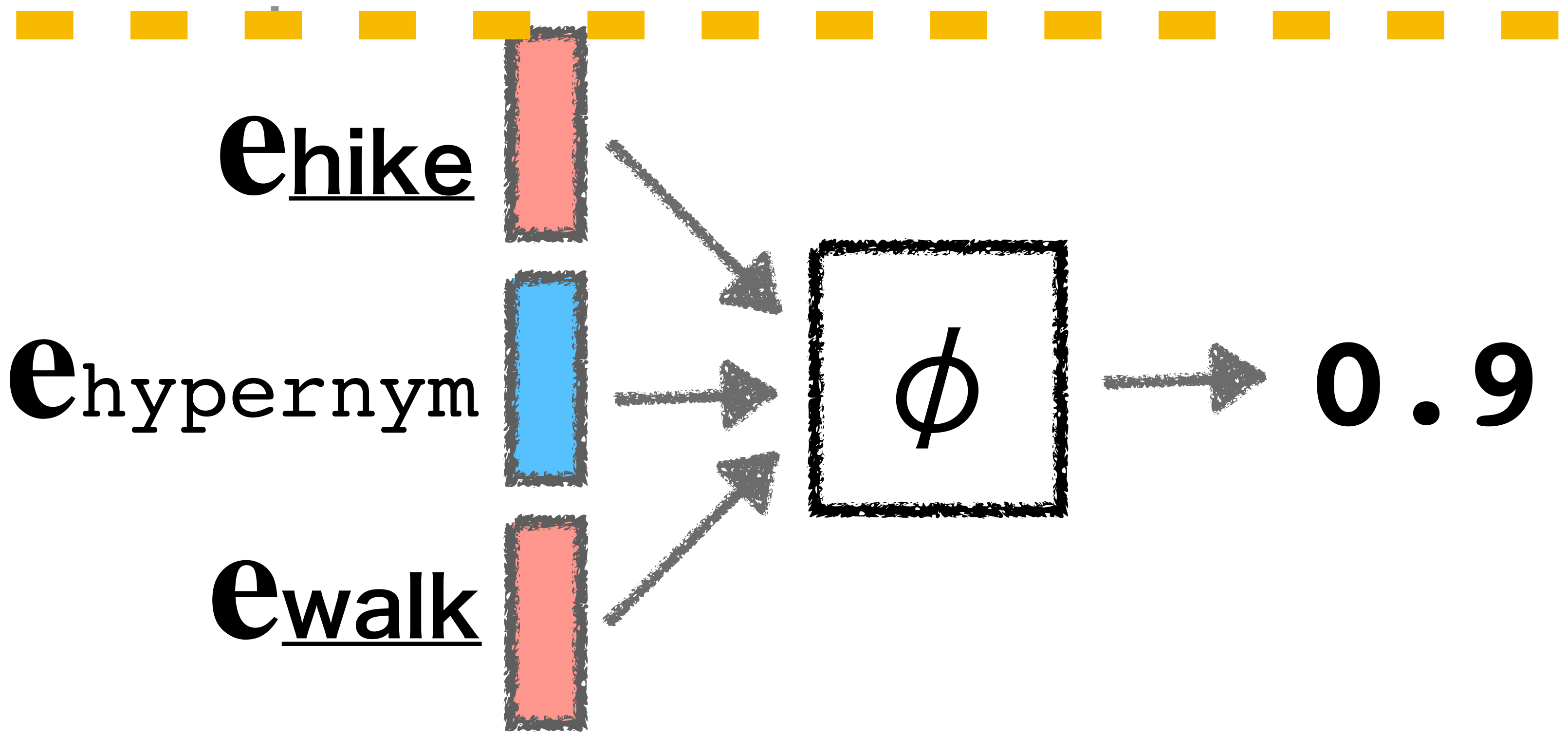




1. Extending Abduction Mechanism with KBC

	Search on KB	KBC
Latent Knowledge	Hand-crafted rules (e.g. transitive closure of `hypernym`)	KBC models learn accurately
Efficiency	Multi-hop reasoning takes time	One dot product (ComplEx)
Scalability	Adding more knowledge harms the search time	Knowledge from VerbOcean (Chklovski et al., 2004) are added for free





2. Faster Reasoning with "**abduction**" Coq plugin

Coq Interactive Session

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 ============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x)
```

2. Faster Reasoning with "**abduction**" Coq plugin

Coq Interactive Session

Lexical gap!

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 ============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x)
```

2. Faster Reasoning with "abduction" Coq plugin

Coq Interactive Session

Lexical gap!

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 ============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x)

t < abduction.
```

2. Faster Reasoning with "abduction" Coq plugin

Coq Interactive Sessio

Lexical gap!

(man, walk)
(man, hike)
(hike, walk)

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 ============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x)

t < abduction.
```

2. Faster Reasoning with "abduction" Coq plugin

Construct a list of predicate pairs from context and goal

Coq Interactive Sessio

Lexical gap!

(man, walk)
(man, hike)
(hike, walk)

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 ============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x)

t < abduction.
```

2. Faster Reasoning with "abduction" Coq plugin

Coq Interactive Sessio

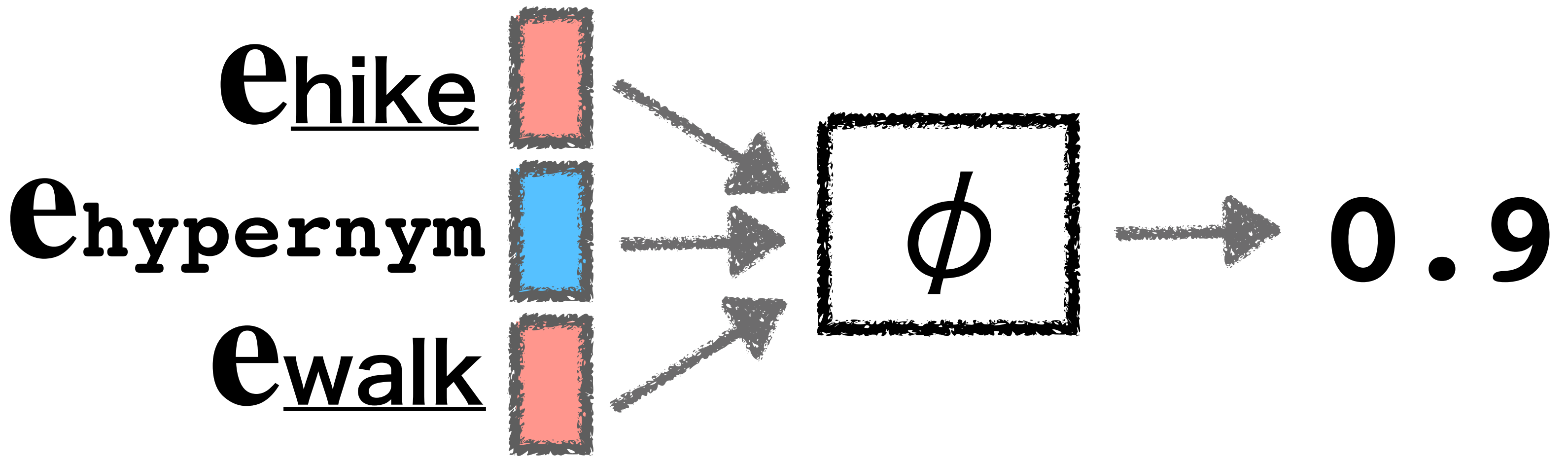
```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj
 ================================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x

t < abduction.
```

Lexical gap!

(man, walk)
(man, hike)
(hike, walk)

Construct a list of predicate pairs from context and goal

Evaluate all the predicate pairs using ComplEx



Filter them by score

2. Faster Reasoning with "abduction" Coq plugin

Coq Interactive Sessio

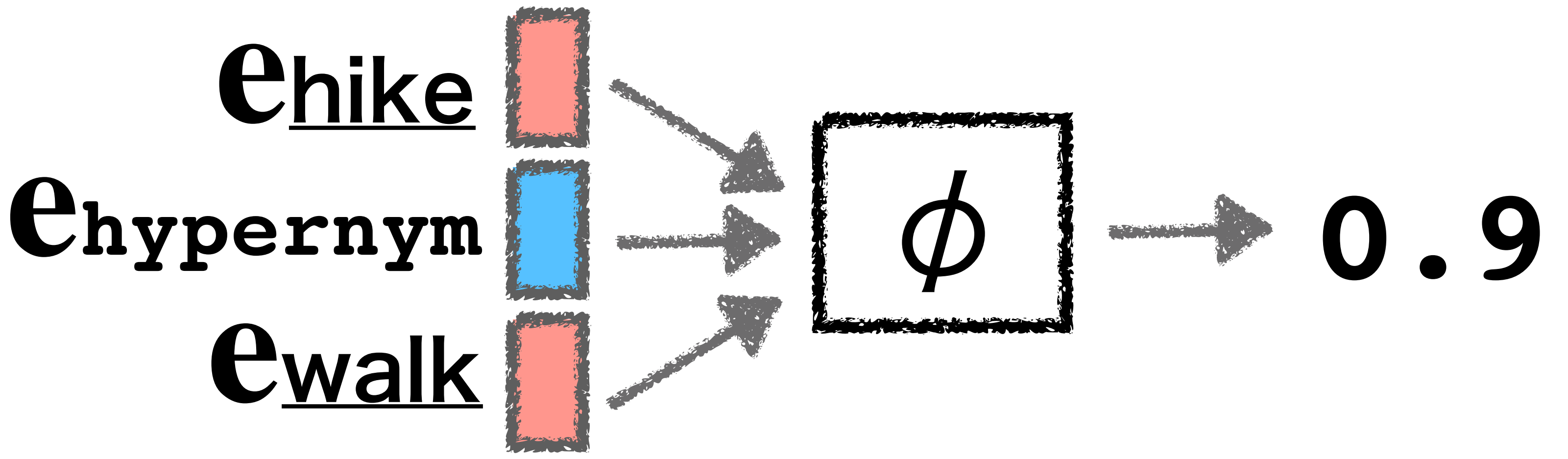
Lexical gap!

(man, walk)
(man, hike)
(hike, walk)

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj
 ==============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x

t < abduction.
```

Construct a list of predicate pairs from context and goal

Evaluate all the predicate pairs using ComplEx



Filter them by score

Add them as axioms

(hike, hypernym, walk)

$$\forall x\,.\,\mathbf{hike}(x) \rightarrow \mathbf{walk}(x)$$

2. Faster Reasoning with "abduction" Coq plugin

Coq Interactive Sessio

```
1 subgoal

 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj
 ===============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x

t < abduction.
1 subgoal
 H : exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)
 NLax1 : forall x : Event, hike x -> walk x
 ===============================
 exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x
```

Lexical gap!

(man, walk)
(man, hike)
(hike, walk)

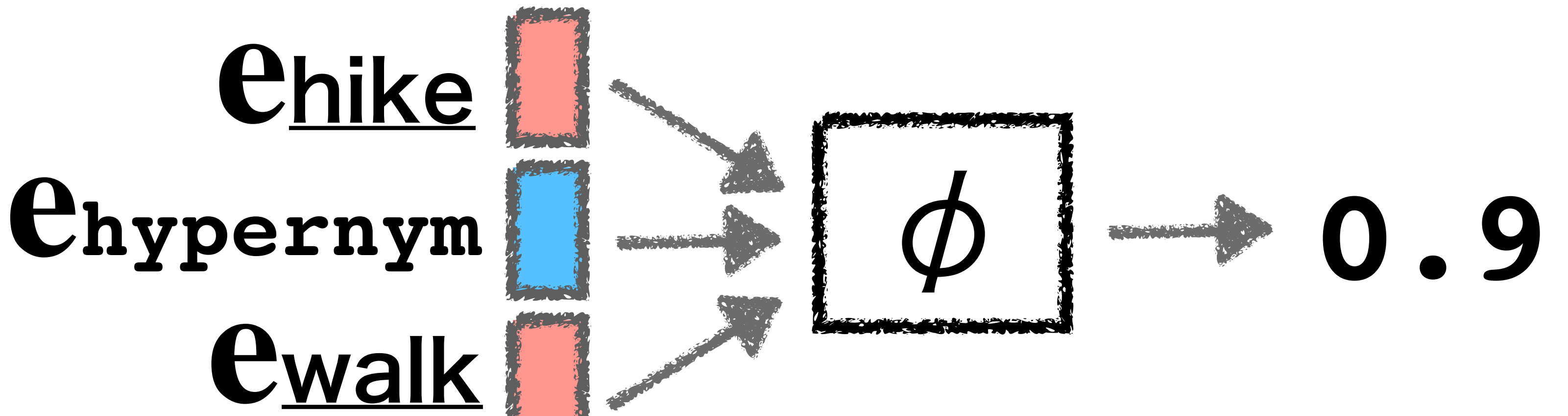
(hike, hypernym, walk)

$$\forall x\,.\,\mathbf{hike}(x) \rightarrow \mathbf{walk}(x)$$

Construct a list of predicate pairs from context and goal

↓

Evaluate all the predicate pairs using ComplEx



↓

Filter them by score

↓

Add them as axioms

Summary so far…

Syntactic Parsing

CCG Derivations

Semantic Parsing

Logical Formulas

Theorem Proving

{ yes, no, unknown }

NP NP/N N S\NP A man hikes

NP NP/N N S\NP A man walks

$$\exists x.\mathtt{man}(x) \wedge \exists e.\mathtt{hike}(e) \wedge \mathtt{subj}(e,x)$$

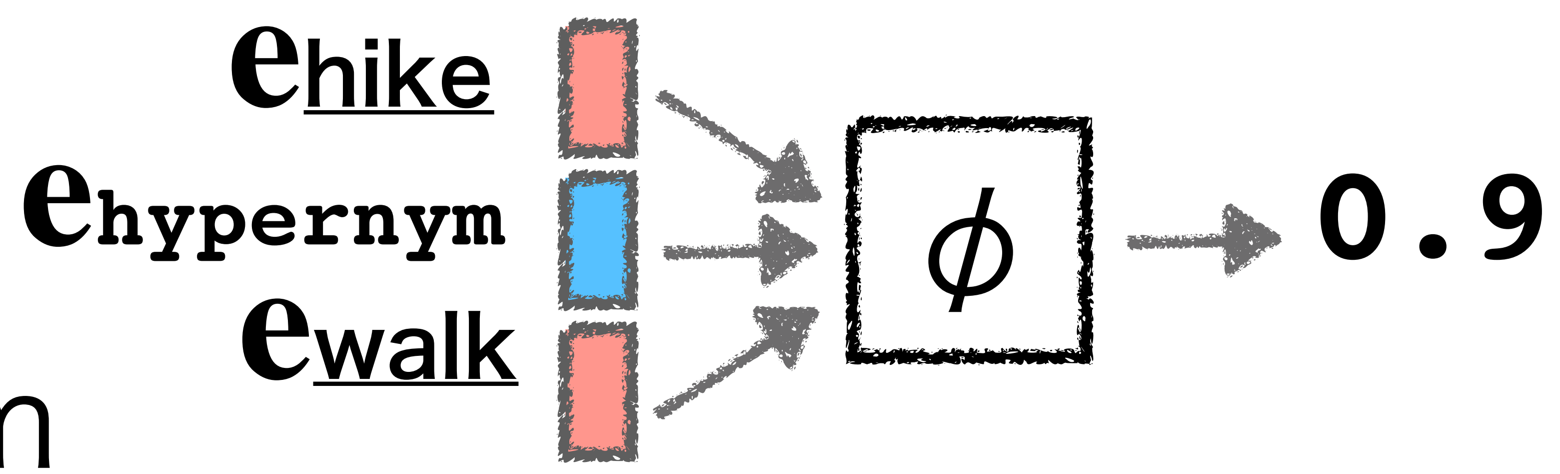
$$\exists x.\mathtt{man}(x) \wedge \exists e.\mathtt{walk}(e) \wedge \mathtt{subj}(e,x)$$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
  exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

Coq

+abduction

result: yes



👍 Efficient and scalable abduction mechanism

👍 No need to rerun Coq in abduction

- Our method is applicable to other logic-based systems
  - e.g. Modern Type Theory (Bernandy and Chatzikyriakidis, 2017)

Experiments

- SICK RTE dataset (Marelli et al., 2014)
- Metrices: accuracy and processing time
- ComplEx is trained on logistic loss: $\sum_{((s,r,o),t)\in\mathcal{D}} t \log f(s,r,o) + (1-t)\log(1-f(s,r,o))$
- The training data is constructed using WordNet
  - synonym, antonym, hyponym, hypernyms, etc.
  - The trained ComplEx model achieves MRR of 77.68%

P: A flute is being played in a lovely way by a girl.

H: One woman is playing a flute.

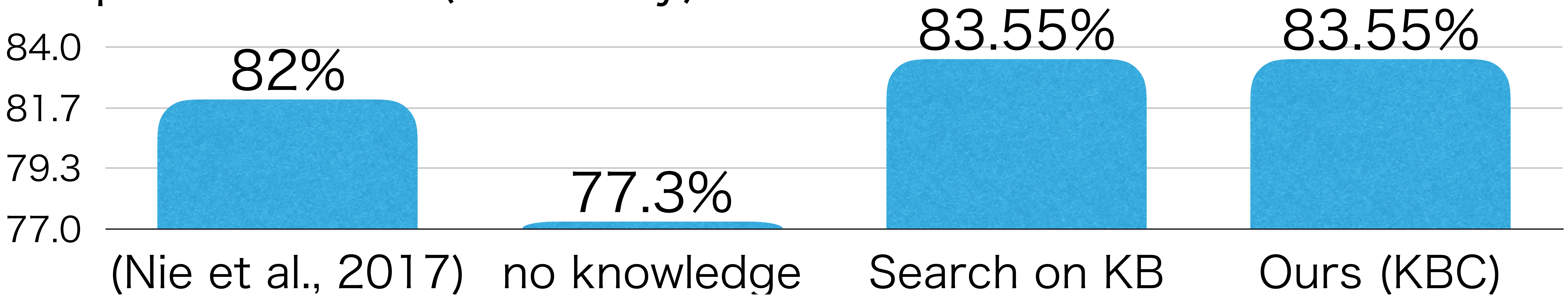
logical

lexical phenomena

syntactic

Experimental Results on SICK

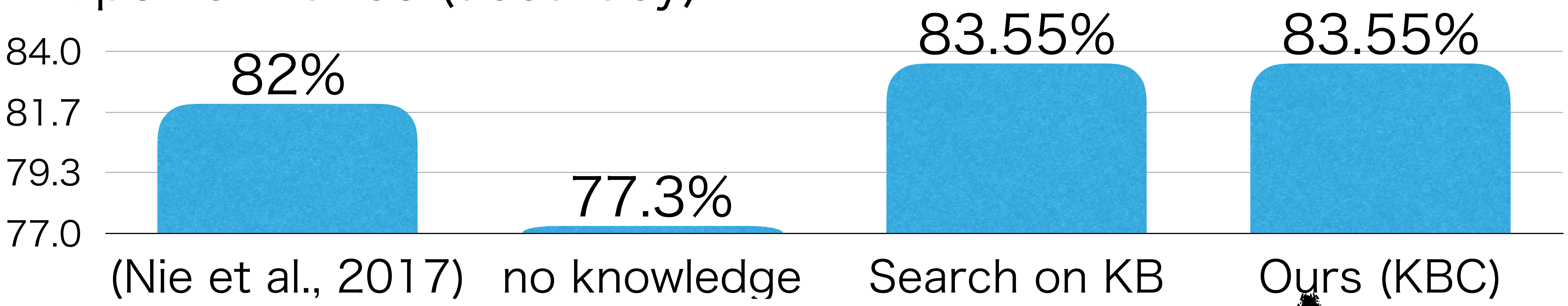
- RTE performance (accuracy)



- Baselines: Search on KB (Martínez-Gómez et al., 2017), NN-based (Nie et al., 2017)

Experimental Results on SICK

- RTE performance (accuracy)

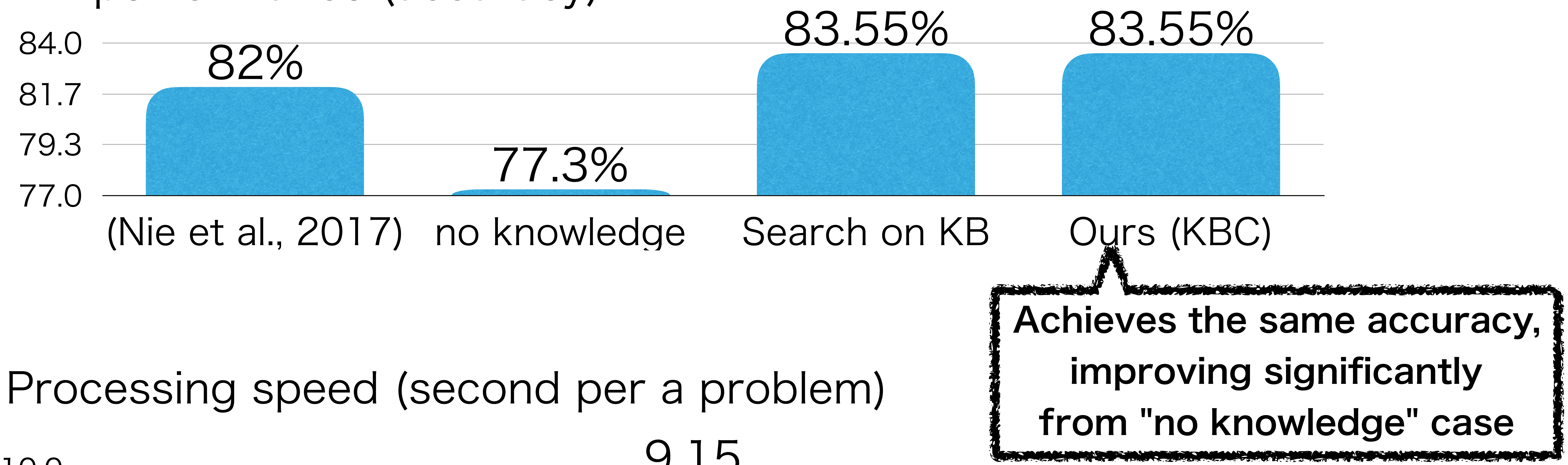


Achieves the same accuracy, improving significantly from "no knowledge" case

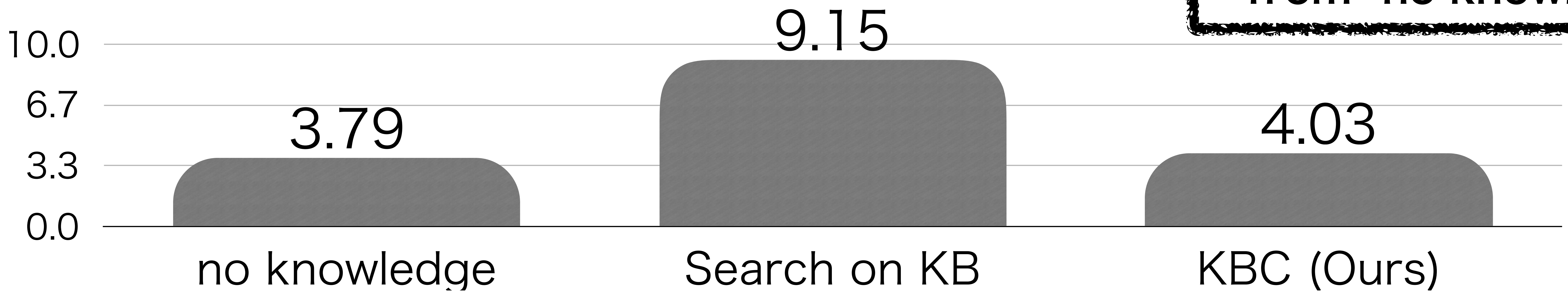
- Baselines: Search on KB (Martínez-Gómez et al., 2017), NN-based (Nie et al., 2017)

Experimental Results on SICK

- RTE performance (accuracy)



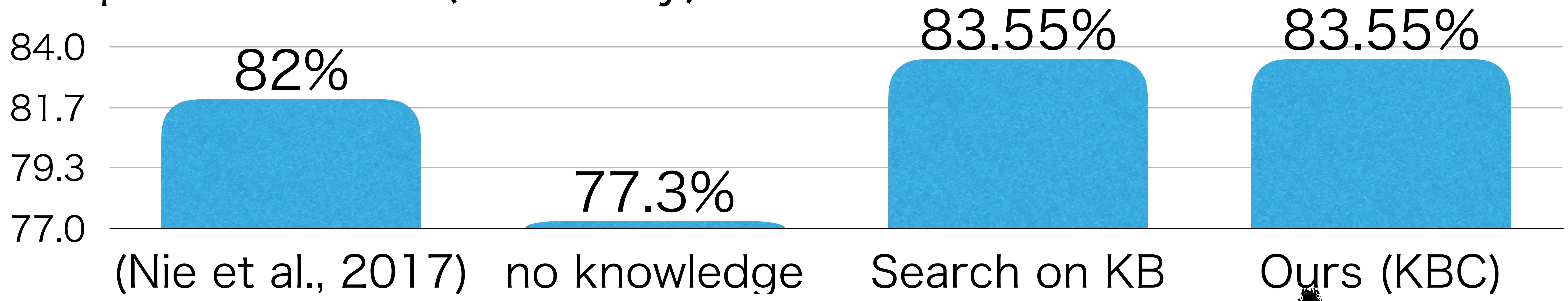
- Processing speed (second per a problem)



- Baselines: Search on KB (Martínez-Gómez et al., 2017), NN-based (Nie et al., 2017)

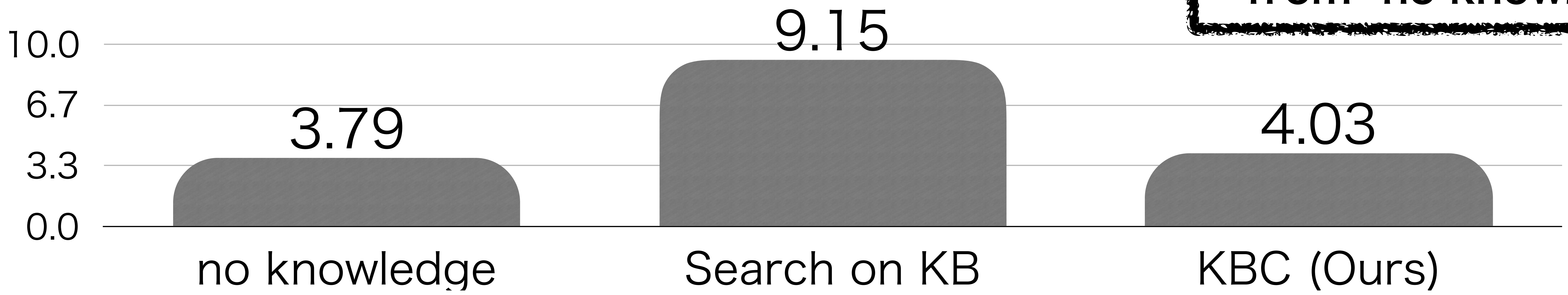
Experimental Results on SICK

- RTE performance (accuracy)



Achieves the same accuracy, improving significantly from "no knowledge" case

- Processing speed (second per a problem)



- Baselines: Search on KB (Martínez-Gómez et al., 2017), NN

Our method halves the time to process an RTE problem!

Thank you!

- A KBC-based axiom injection for logic-based RTE systems
  - Efficient, scalable, and it provides latent knowledge
- `abduction` tactic for further faster reasoning
- **Come to my poster (#1319) for other topics**:
  - Adding other KB (VerbOcean) without losing efficiency
  - Evaluating learned latent knowledge in terms of RTE (LexSICK dataset)
- **All the codes, dataset and slides are available**:
  - https://masashi-y.github.io