UNIVERSITY of WASHINGTON

UWNLP

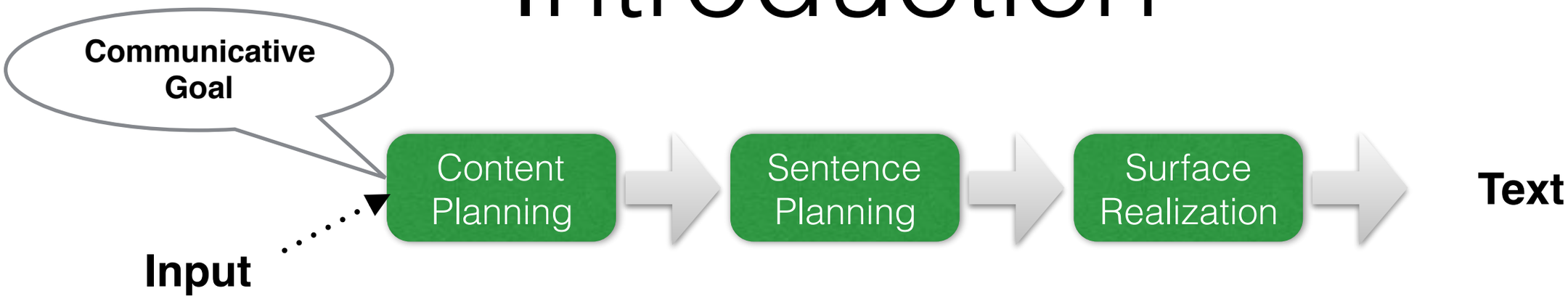
NNLG

Neural Natural Language Generation

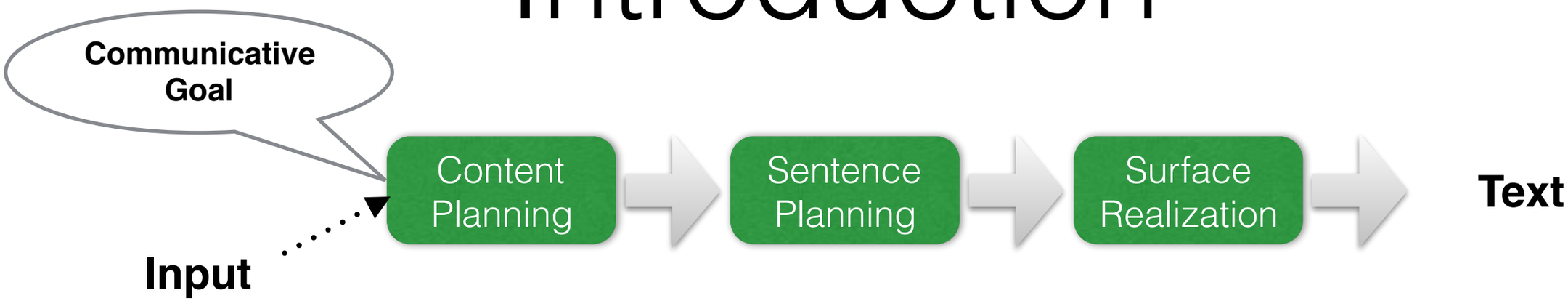
Yannis Konstas

Joint work with
Srinivasan Iyer, Mark Yatskar, Rik Koncel-Kedziorski, Li Zilles,
Luke Zettlemoyer, Yejin Choi, Hannaneh Hajishirzi

Introduction

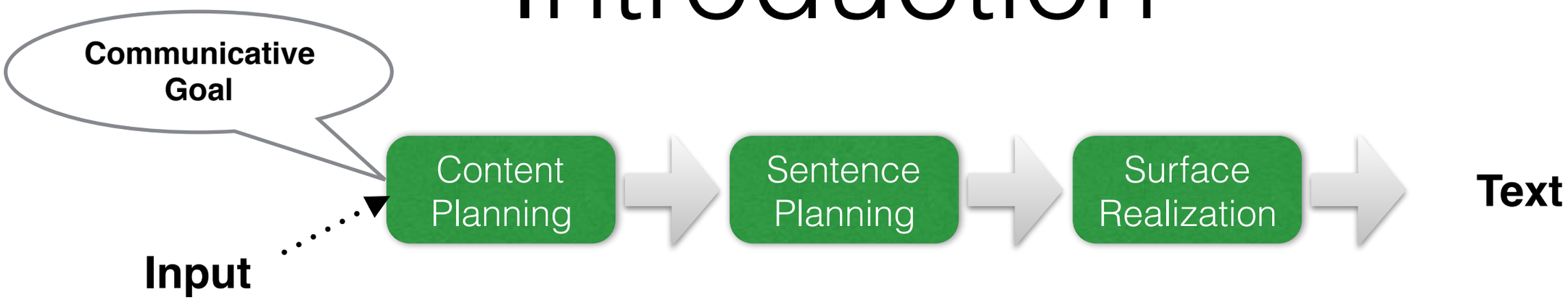


Introduction



- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multi-modal sources

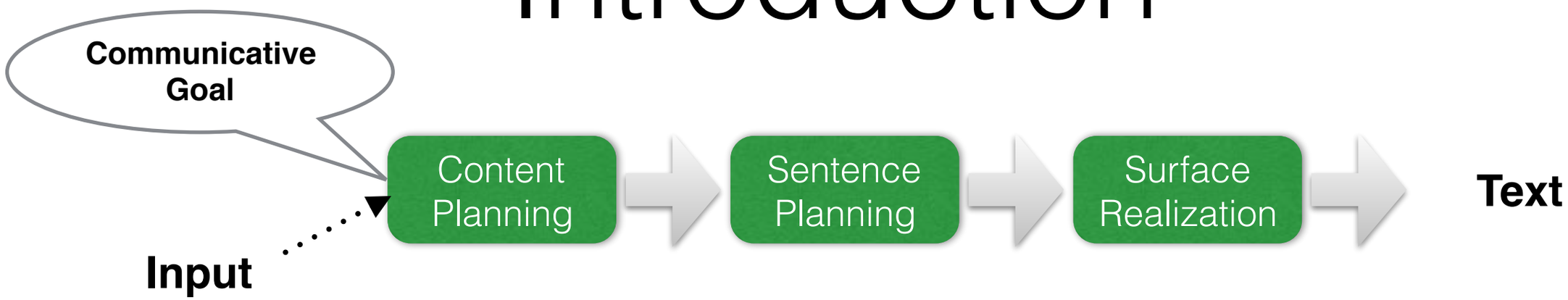
Introduction



- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multi-modal sources

- Single utterance
- Single (complex) sentence
- Multiple sentences
- Multiple paragraphs

Introduction



- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multi-modal sources

- Single utterance
- Single (complex) sentence
- Multiple sentences
- Multiple paragraphs

☑ What is the best input representation?

☑ How can we model document structure?

☑ How do we know that we have done well?

Concept-to-Text Generation

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

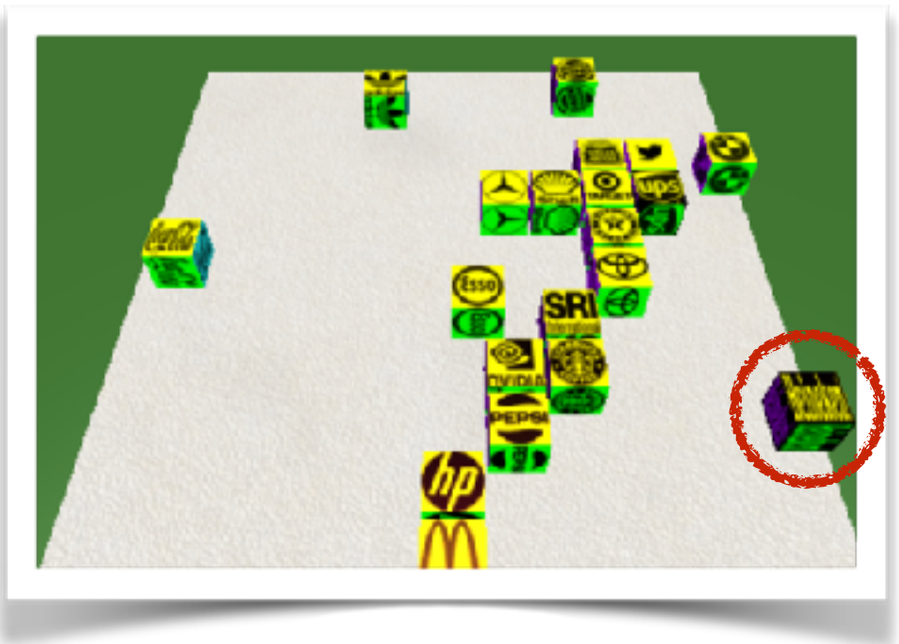
Concept-to-Text Generation

Input: Machine-generated Representation

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

Concept-to-Text Generation

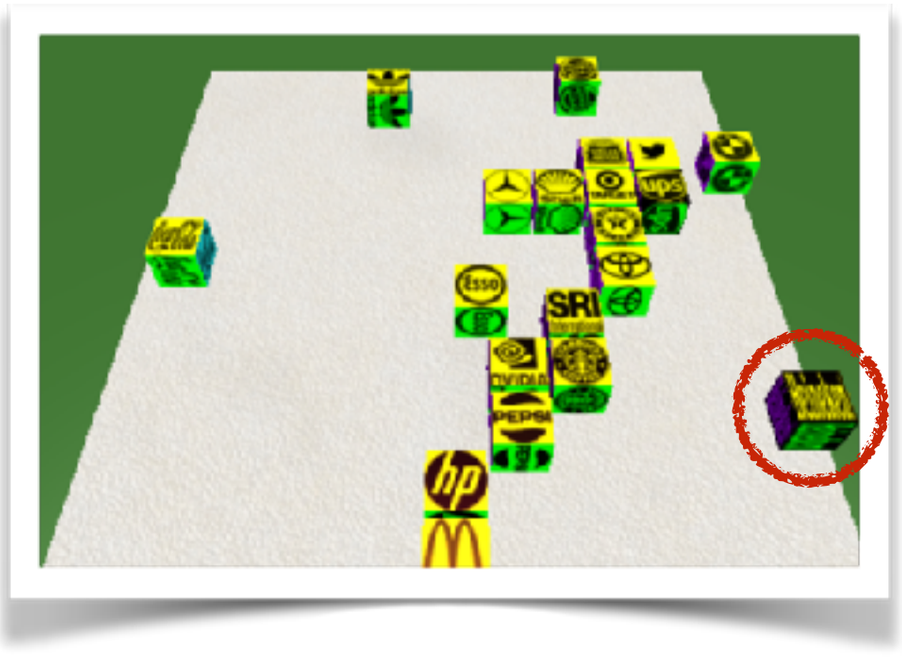
Input: Machine-generated Representation

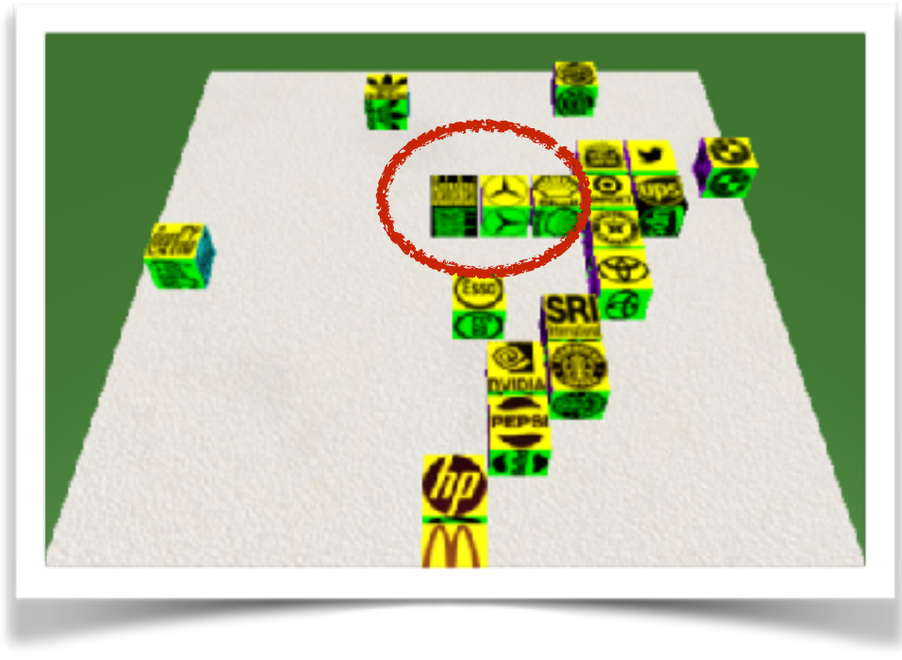


(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

Concept-to-Text Generation

Input: Machine-generated Representation

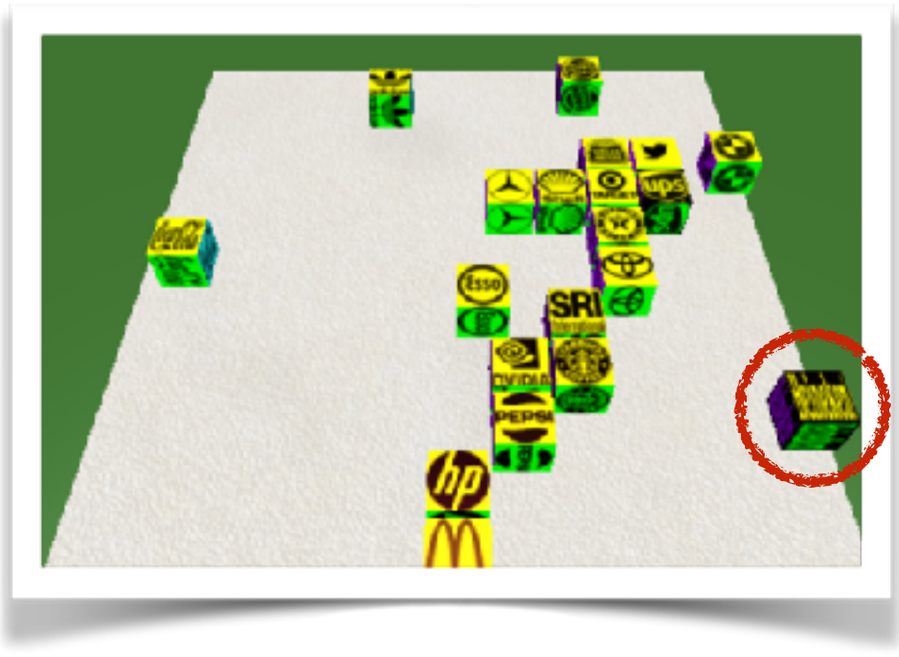


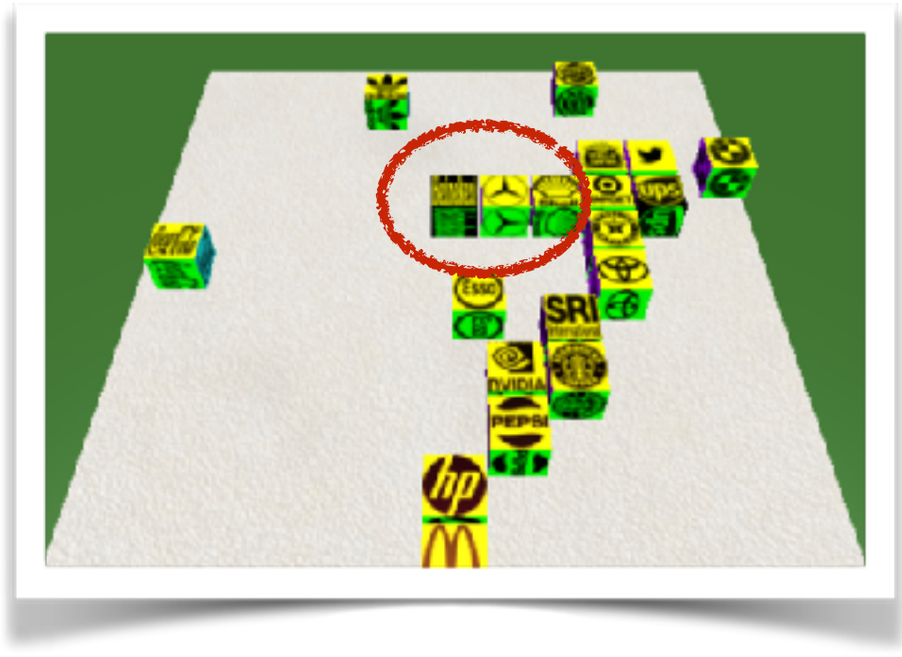


(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

Concept-to-Text Generation

Input: Machine-generated Representation



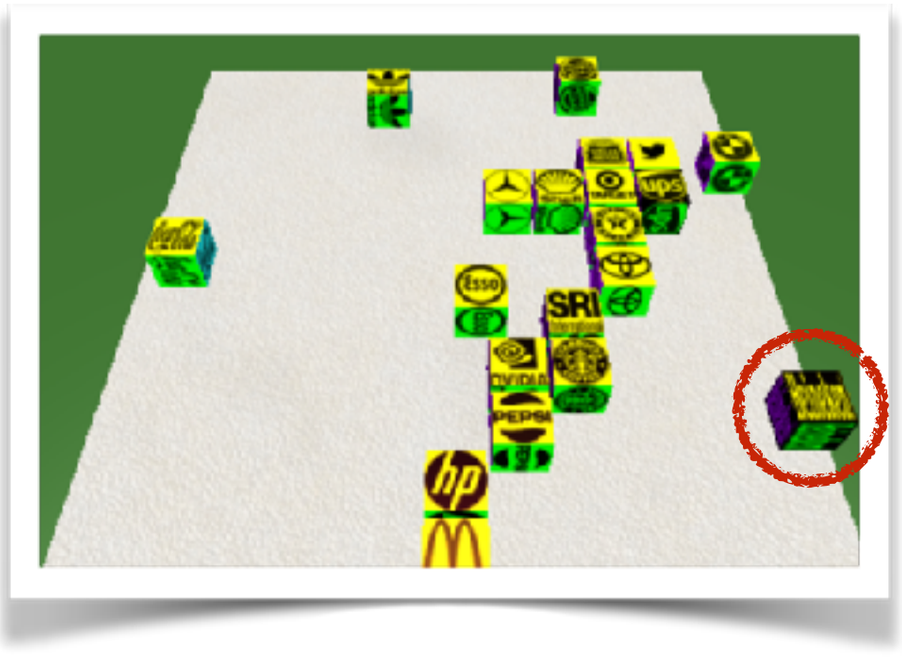


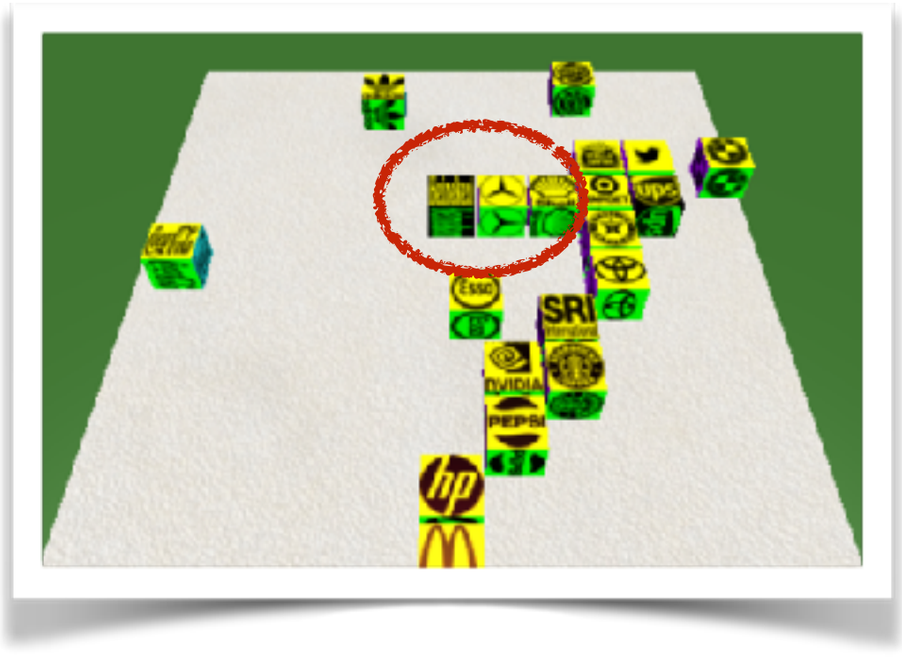
source	block:	hk		
target	block:	ms		
pos	RP:	W	scale:	small

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

Concept-to-Text Generation

Input: Machine-generated Representation





source	block:	hk		
target	block:	ms		
pos	RP:	W	scale:	small

Place the heineken block west of the mercedes block.

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)

Code-to-Text Generation

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Code-to-Text Generation

Input: Source Code

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Code-to-Text Generation

Input: Source Code

CODE-NN

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Code-to-Text Generation

Input: Source Code

CODE-NN

```
public int TextWidth (string text) {
  TextBlock t = new TextBlock();
  t.Text = text;
  return (int) Math.Ceiling(t.ActualWidth);
}
```

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Code-to-Text Generation

Input: Source Code

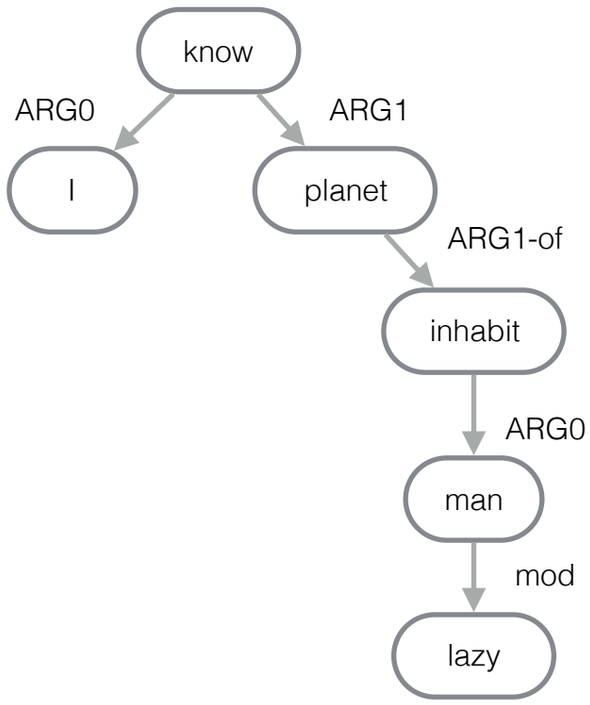
CODE-NN

```
public int TextWidth (string text) {
  TextBlock t = new TextBlock();
  t.Text = text;
  return (int) Math.Ceiling(t.ActualWidth);
}
```

Get rendered width of string rounded up to the nearest integer.

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

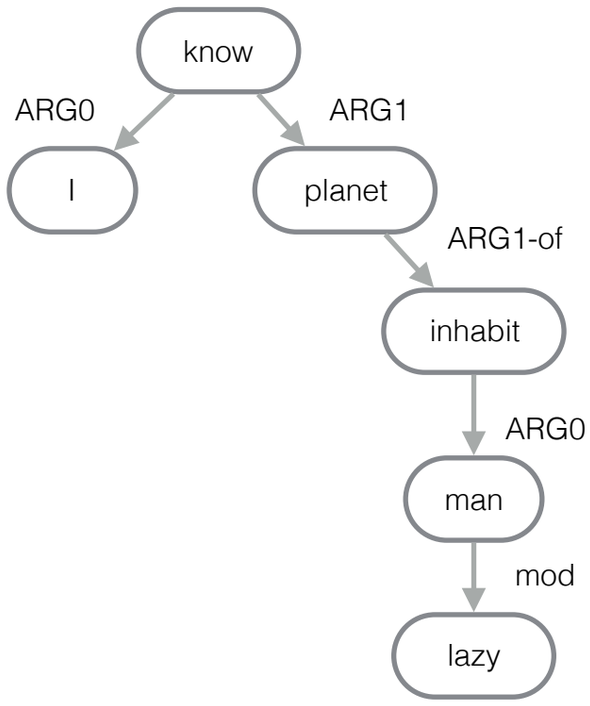
Meaning Representation Generation



Input: Predicate - Argument Structure

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)

Meaning Representation Generation

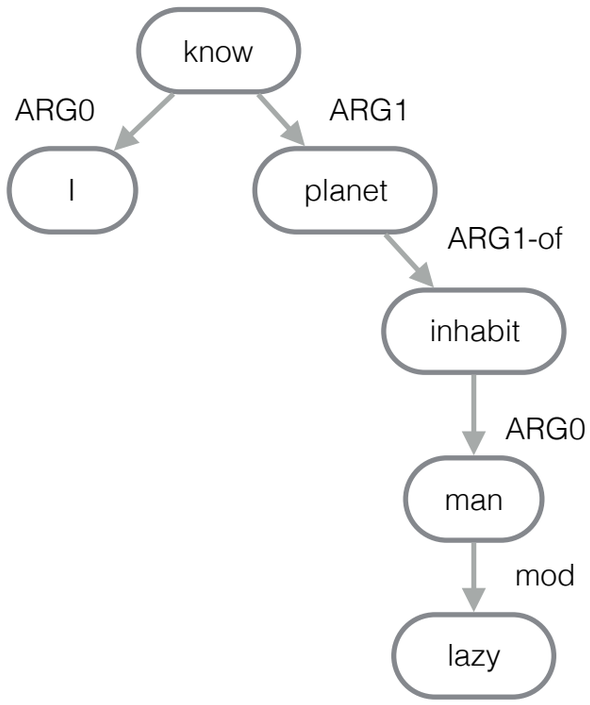


Input: Predicate - Argument Structure

I knew a **planet** that was **inhabited** by a **lazy man**.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)

Meaning Representation Generation



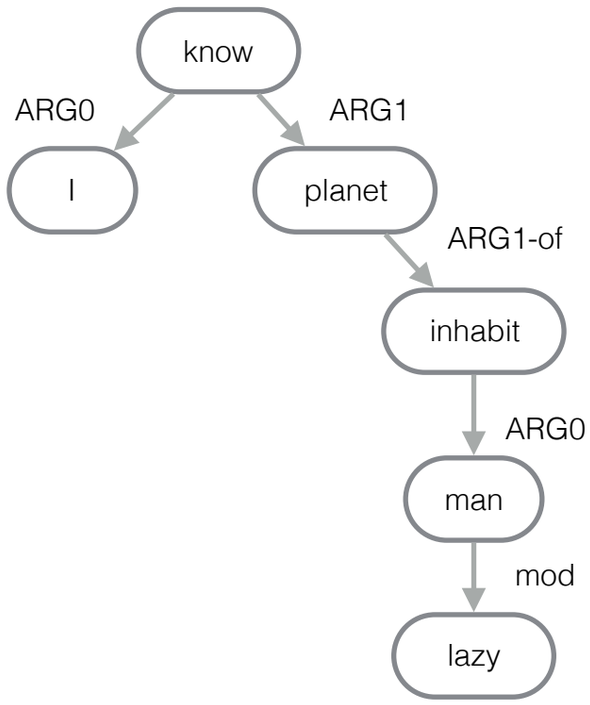
Input: Predicate - Argument Structure

I knew a **planet** that was **inhabited** by a **lazy man**.

I have **known** a **planet** that was **inhabited** by a **lazy man**.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)

Meaning Representation Generation



Input: Predicate - Argument Structure

I knew a **planet** that was **inhabited** by a **lazy man**.

I have **known** a **planet** that was **inhabited** by a **lazy man**.

I know a **planet**. It is **inhabited** by a **lazy man**.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)

Instructional Text Generation

Instructional Text Generation

Input: Goal Cue - Bag of concepts

Instructional Text Generation

Spanakopita (Greek Spinach Pie)

Ingredients

3 tbsp olive oil
1 large onion, chopped
1 bunch green onions
2 cloves garlic, minced
2 pounds spinach
1/2 cup chopped fresh parsley
2 eggs
1/2 cup ricotta cheese
1 cup feta cheese
8 sheets filo dough
1/4 cup olive oil

Input: Goal Cue - Bag of concepts

(Globally Coherent Text Generation with Neural Checklist Models. Kiddon et al, EMNLP 2016.)

Instructional Text Generation

Spanakopita (Greek Spinach Pie)

Ingredients

3 tbsp olive oil
1 large onion, chopped
1 bunch green onions
2 cloves garlic, minced
2 pounds spinach
1/2 cup chopped fresh parsley
2 eggs
1/2 cup ricotta cheese
1 cup feta cheese
8 sheets filo dough
1/4 cup olive oil

Input: Goal Cue - Bag of concepts

Preheat oven to 350 degrees F (175 degrees C). Lightly oil a 9x9 inch square baking pan.

Heat 3 tablespoons olive oil in a large skillet over medium heat. Saute onion, green onions and garlic, until soft and lightly browned. Stir in spinach and parsley, and continue to saute until spinach is limp, about 2 minutes. Remove from heat and set aside to cool.

In a medium bowl, mix together eggs, ricotta, and feta. Stir in spinach mixture. Lay 1 sheet of phyllo dough in prepared baking pan, and brush lightly with olive oil. Lay another sheet of phyllo dough on top, brush with olive oil, and repeat process with two more sheets of phyllo. The sheets will overlap the pan. Spread spinach and cheese mixture into pan and fold overhanging dough over filling. Brush with oil, then layer remaining 4 sheets of phyllo dough, brushing each with oil. Tuck overhanging dough into pan to seal filling.

Bake in preheated oven for 30 to 40 minutes, until golden brown. Cut into squares and serve while hot.

(Globally Coherent Text Generation with Neural Checklist Models. Kiddon et al, EMNLP 2016.)

Story Generation

Input: Script - Text - N/A

Story Generation

Input: Script - Text - N/A

Jim was obsessed with super heroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

Story Generation

Input: Script - Text - N/A

Jim was obsessed with super heroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

He ended up having a great time!

Story Generation

Input: Script - Text - N/A

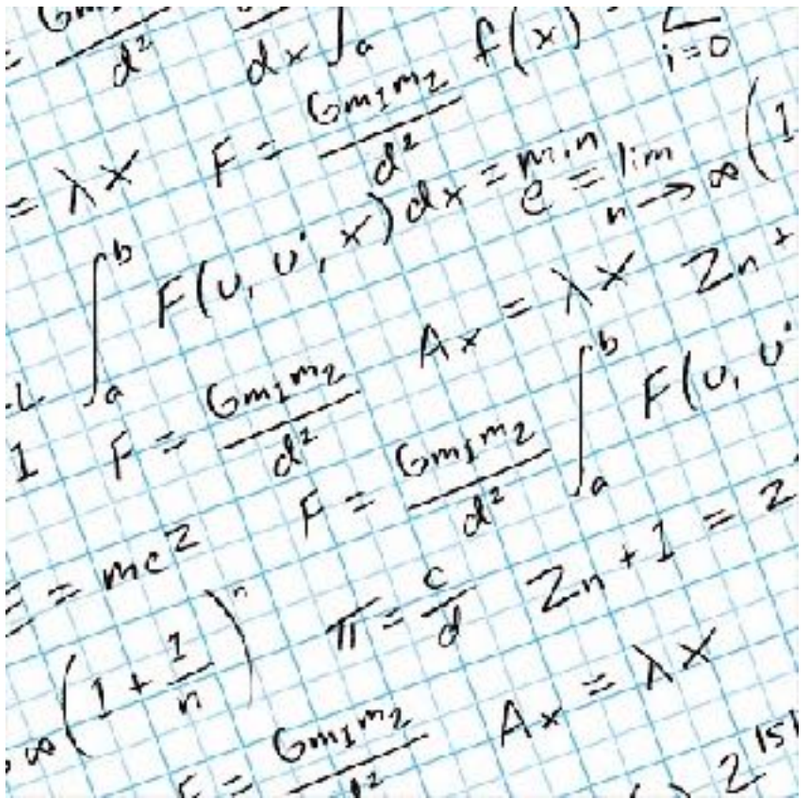
Jim was obsessed with super heroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

He ended up having a great time!

Jim broke his arm and his sister was grounded for a year.

(Work with Li Zilles)

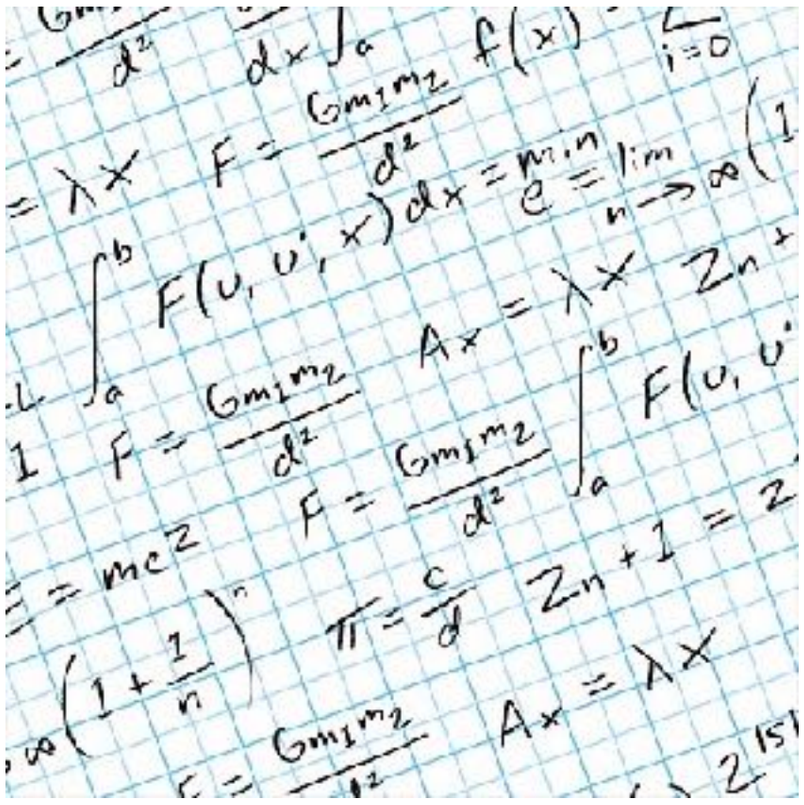
Story Generation (2)



Input: Equation + Theme

(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)

Story Generation (2)

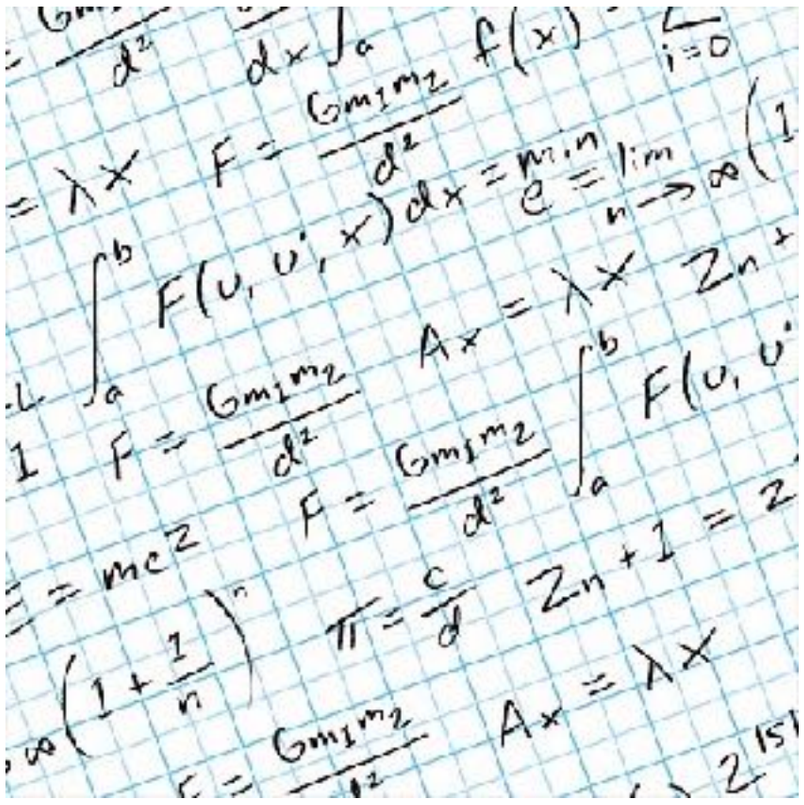


Input: Equation + Theme

504 + x = 639 **+** STAR WARS

(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)

Story Generation (2)



Input: Equation + Theme

504 + x = 639 **+** STAR WARS

Luke Skywalker has 639 blasters. Leia has 504 blasters. How many more blasters does Luke Skywalker have than Leia?

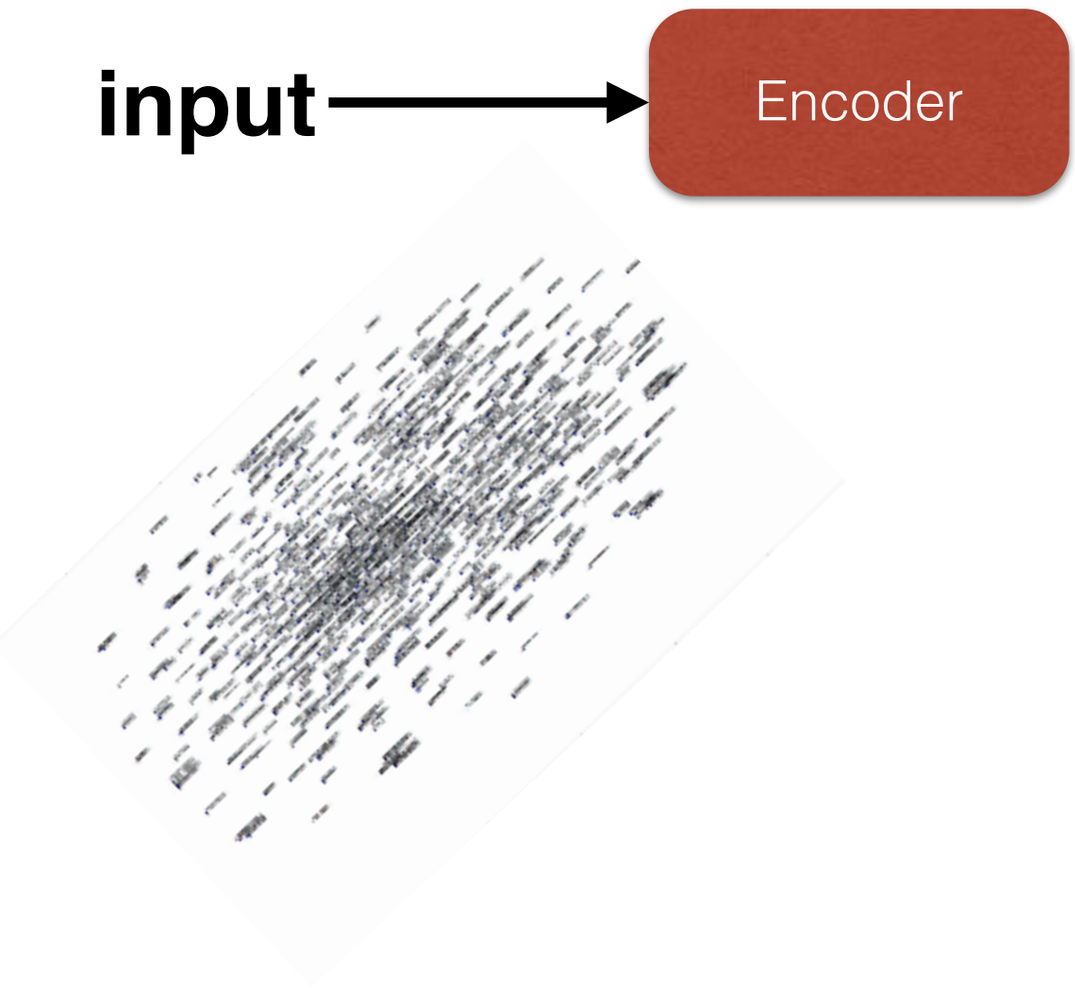
(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)

NNLG Framework

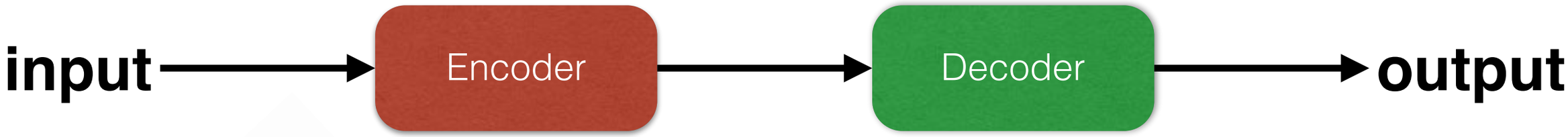
NNLG Framework

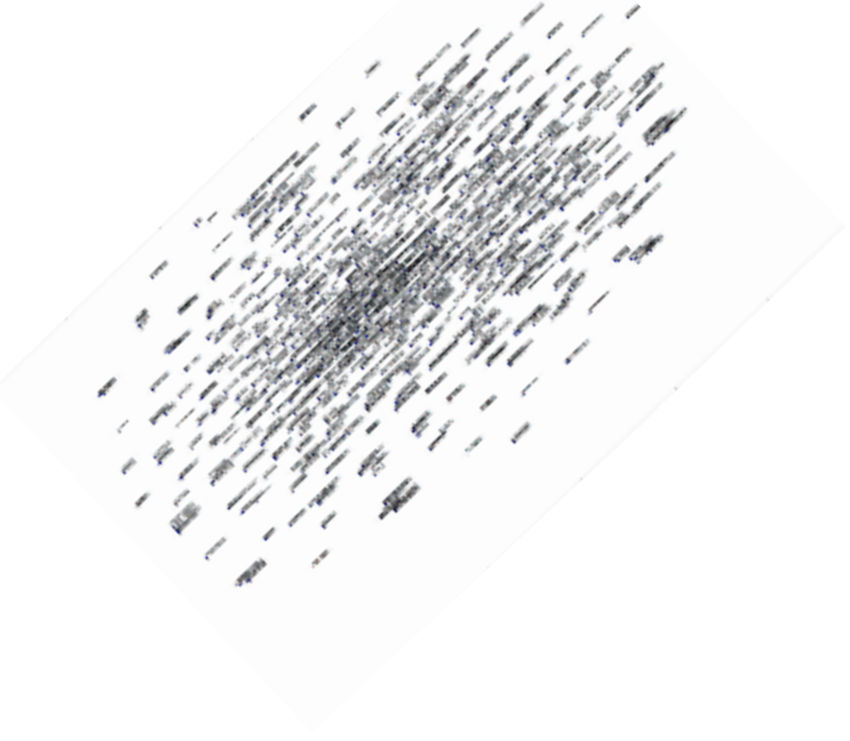
input

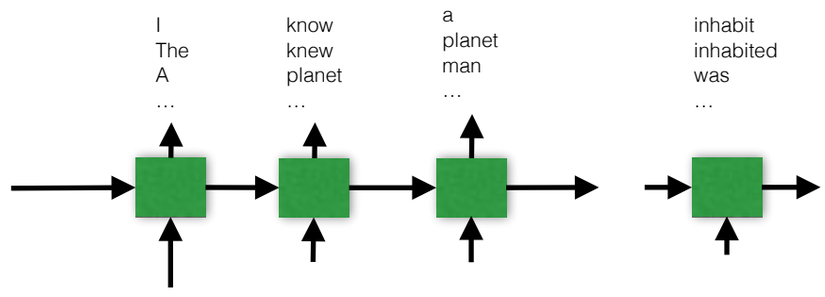
NNLG Framework



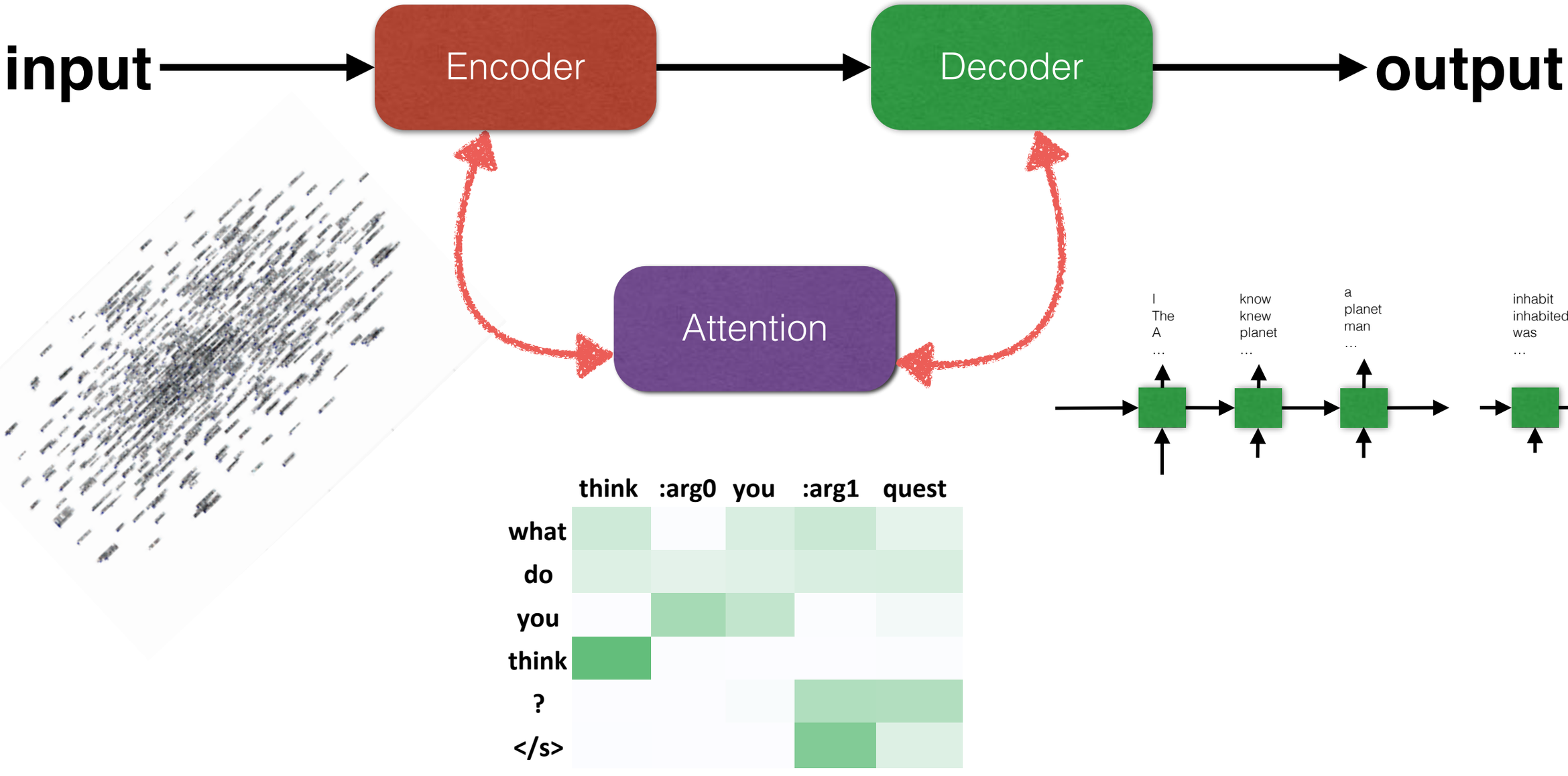
NNLG Framework







NNLG Framework



Encoding

Encoding

Bag of Words

CODE-NN

```
SELECT max(marks) FROM stud_records WHERE marks <
(SELECT max(marks) FROM stud_records);
```

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Encoding

Bag of Words

CODE-NN

```
SELECT max(marks) FROM stud_records WHERE marks <
(SELECT max(marks) FROM stud_records);
```

anonymization

```
SELECT max(col0) FROM tab0 WHERE col0 <
(SELECT max(col1) FROM tab1);
```

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Encoding

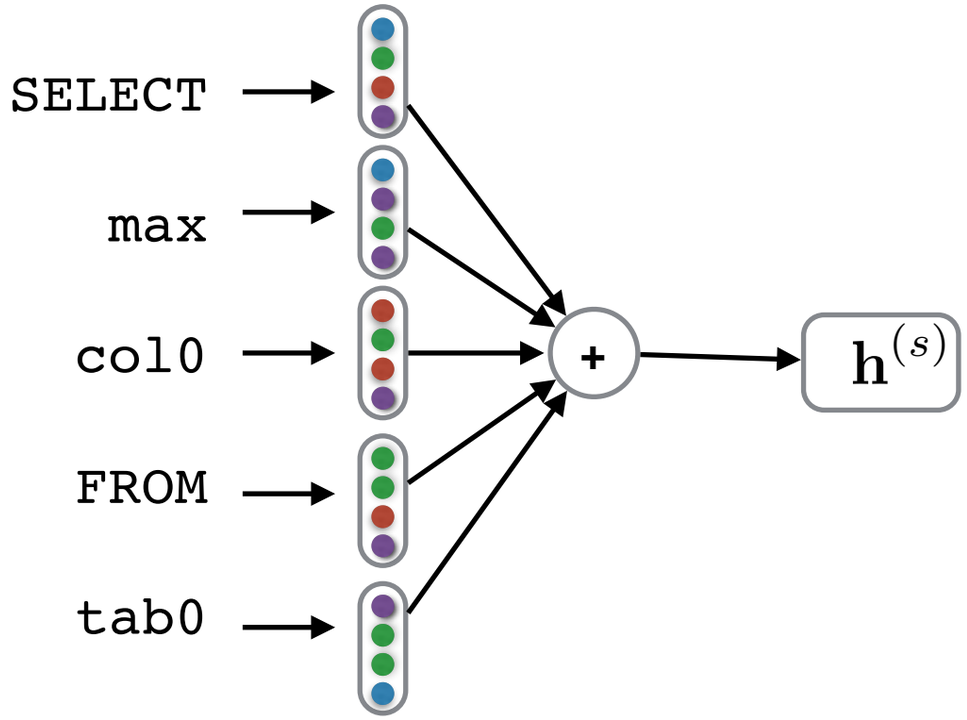
Bag of Words

CODE-NN

```
SELECT max(marks) FROM stud_records WHERE marks <
(SELECT max(marks) FROM stud_records);
```

anonymization

```
SELECT max(col0) FROM tab0 WHERE col0 <
(SELECT max(col1) FROM tab1);
```

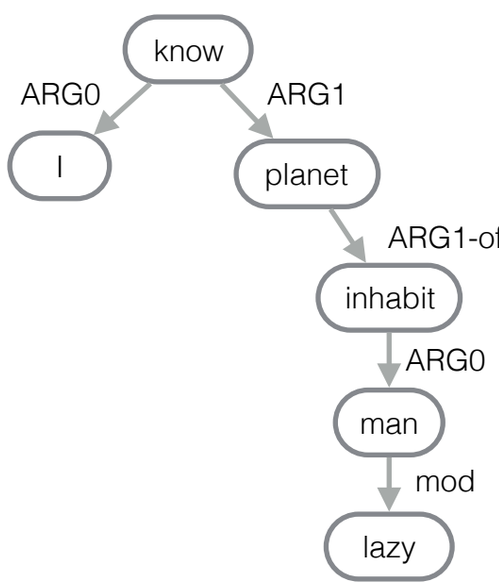


(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)

Encoding

Linearize —> RNN encoding

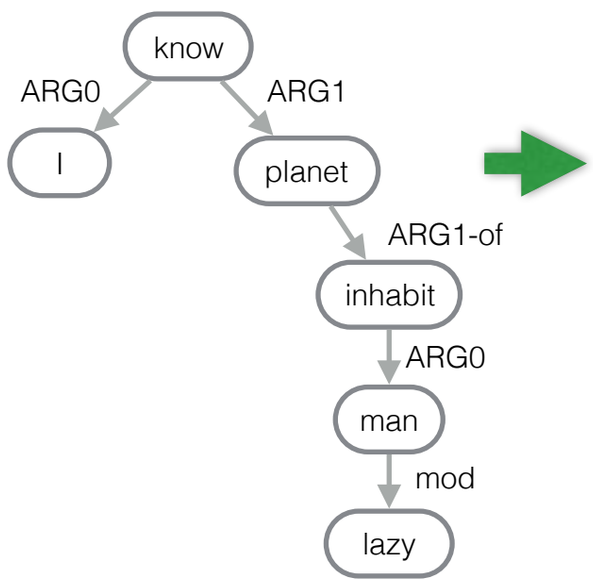
MR Generation



Encoding

Linearize —> RNN encoding

MR Generation

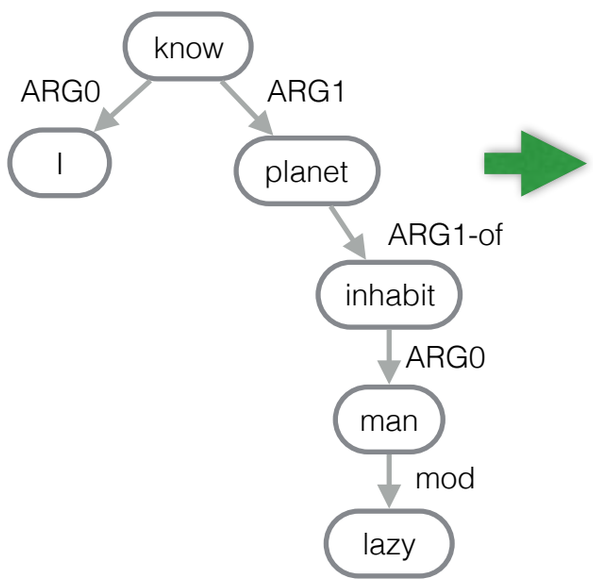



```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```

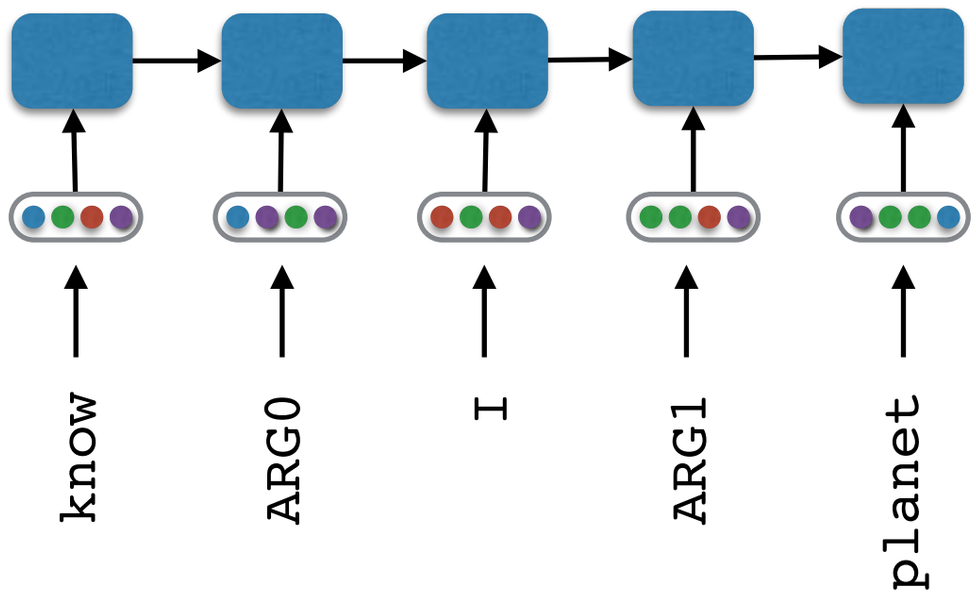
Encoding

Linearize —> RNN encoding

MR Generation



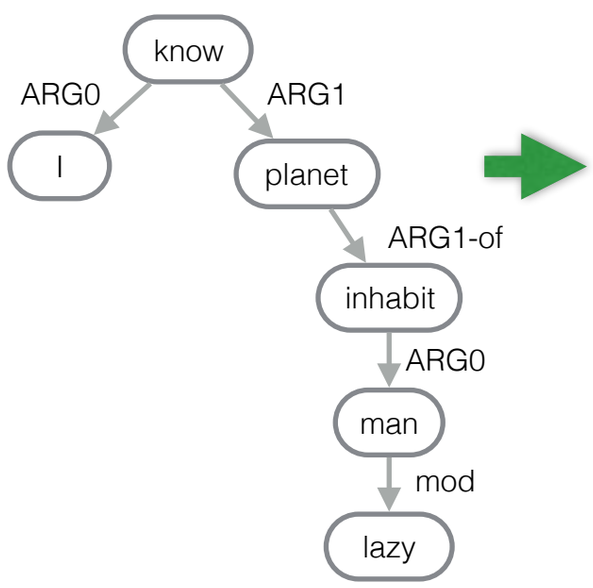

```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```



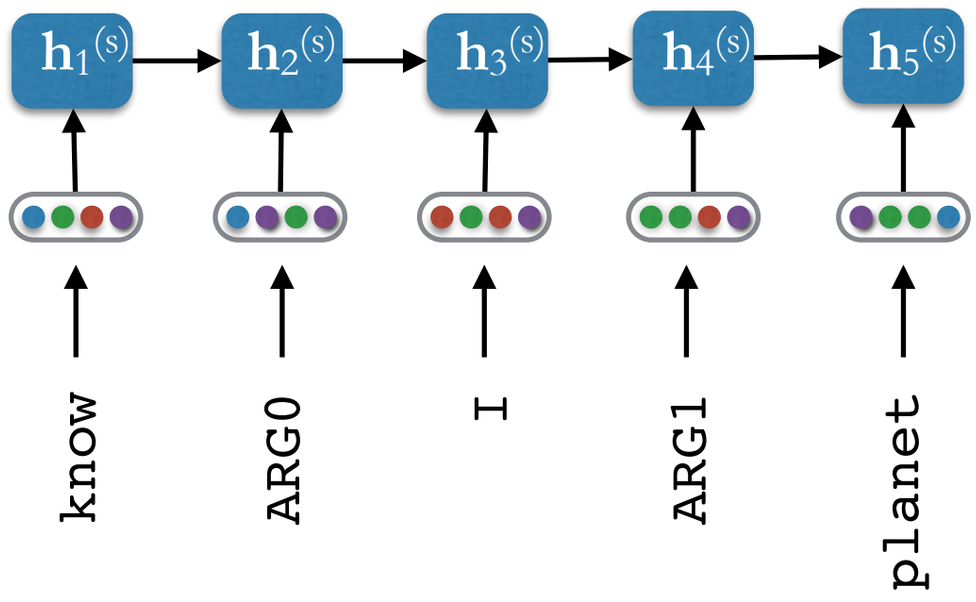
Encoding

Linearize —> RNN encoding

MR Generation



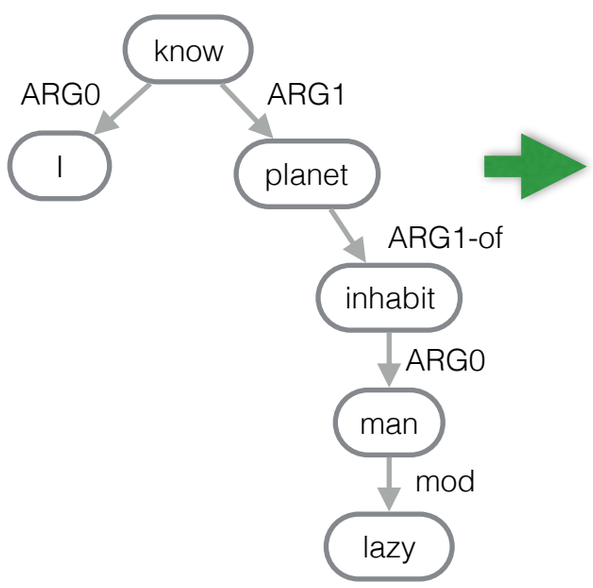

```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```



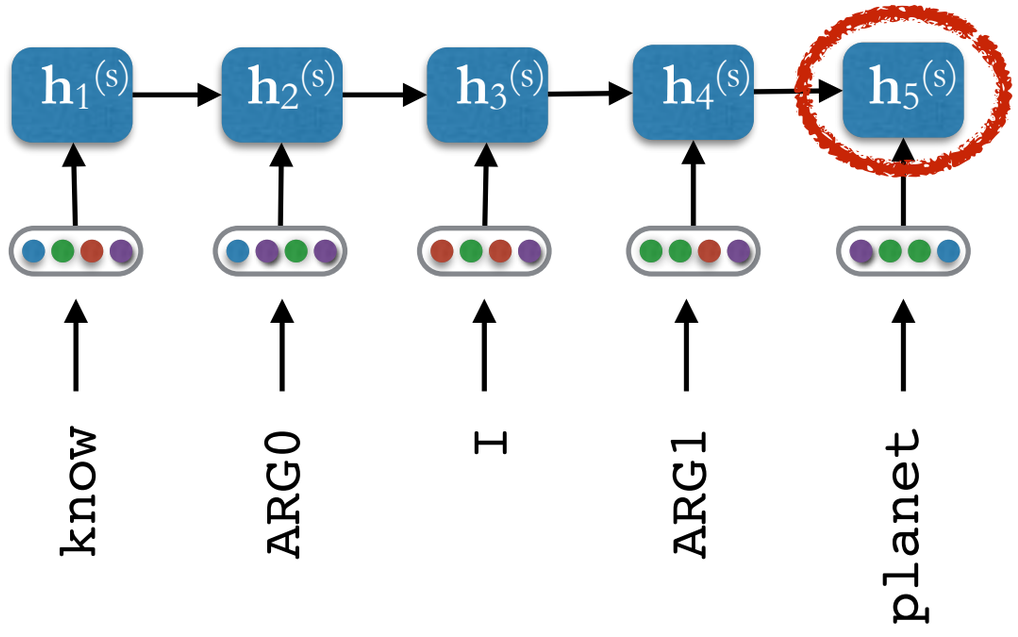
Encoding

Linearize —> RNN encoding

MR Generation



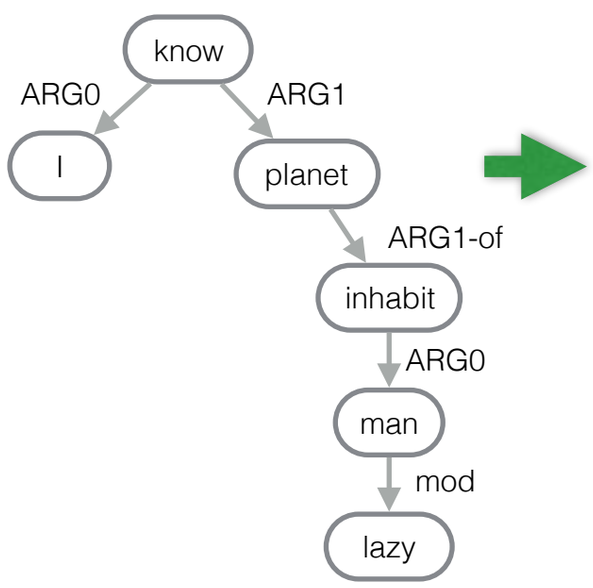

```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```

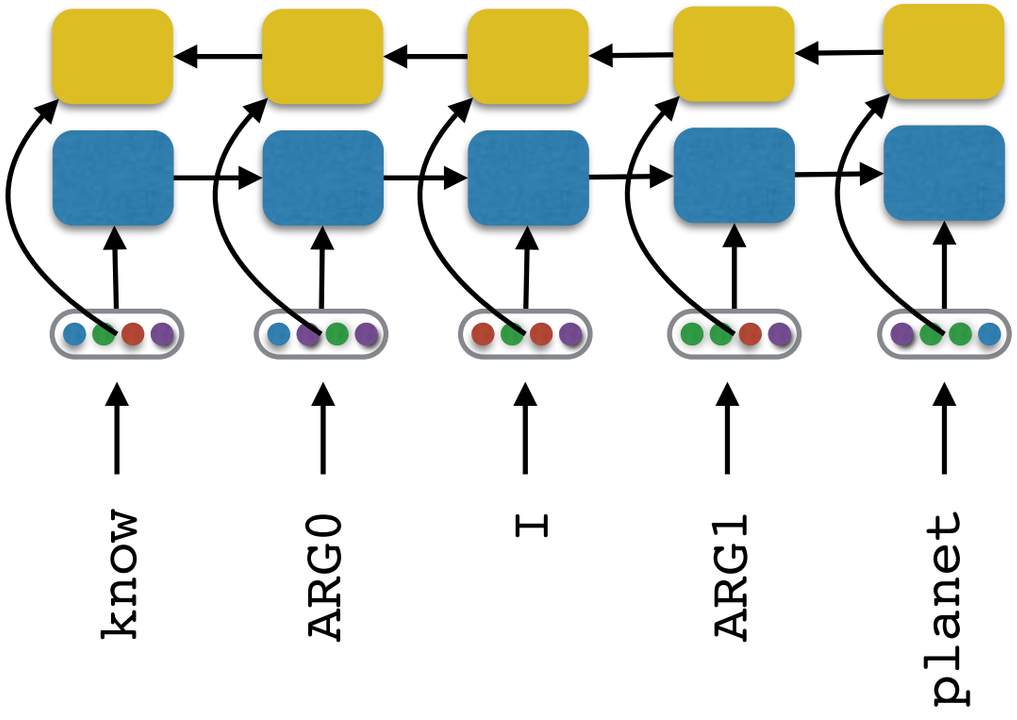


Encoding

Linearize —> RNN encoding

MR Generation

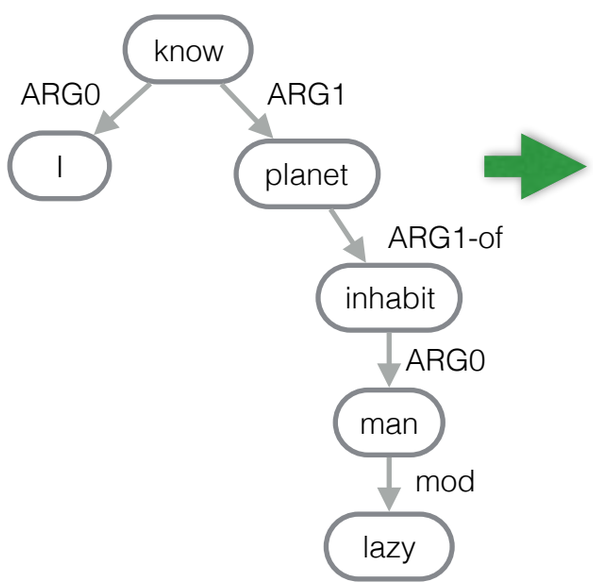




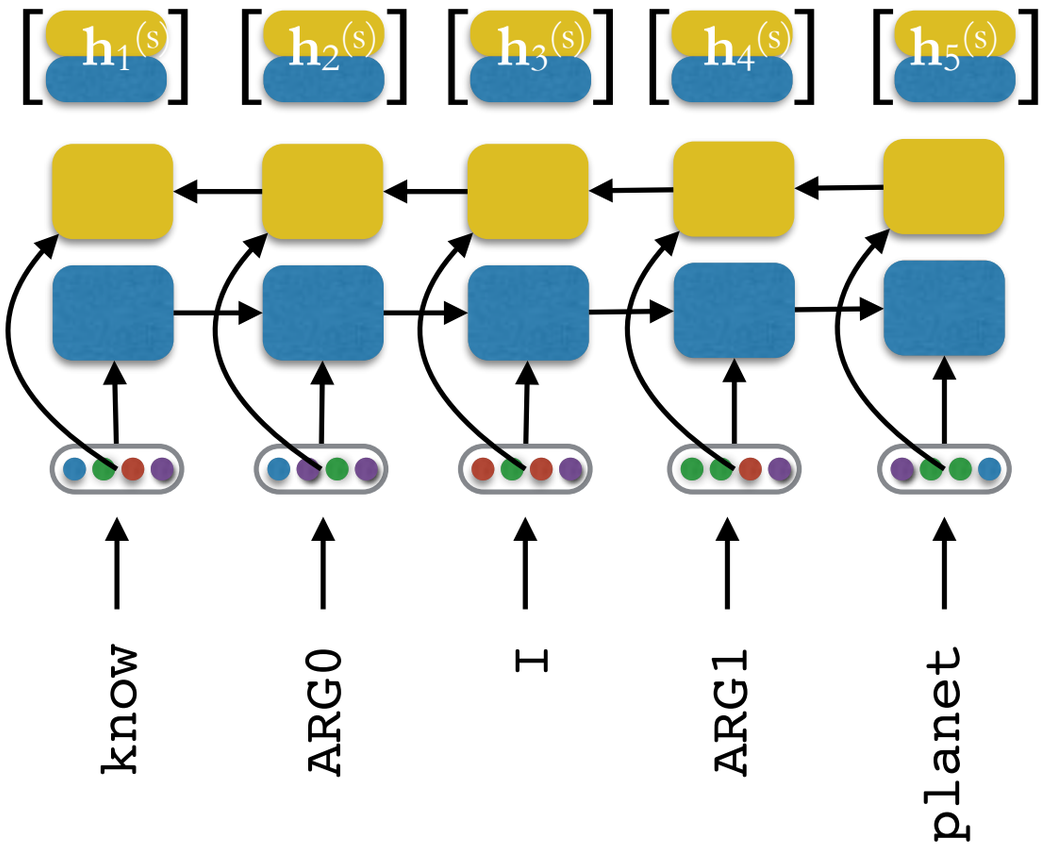
Encoding

Linearize —> RNN encoding

MR Generation



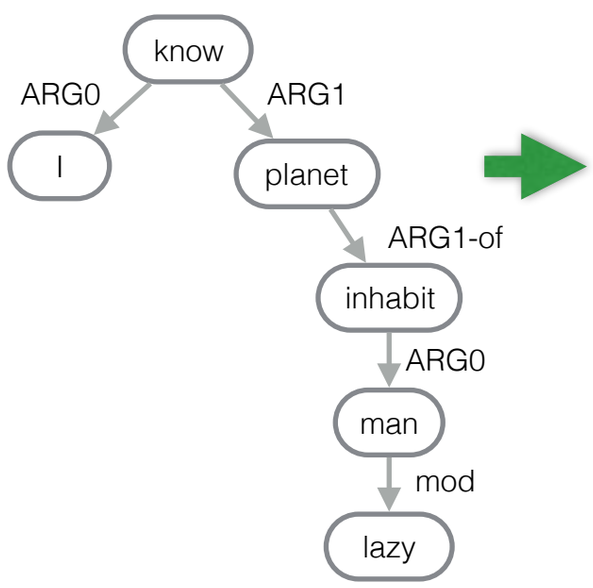

```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```



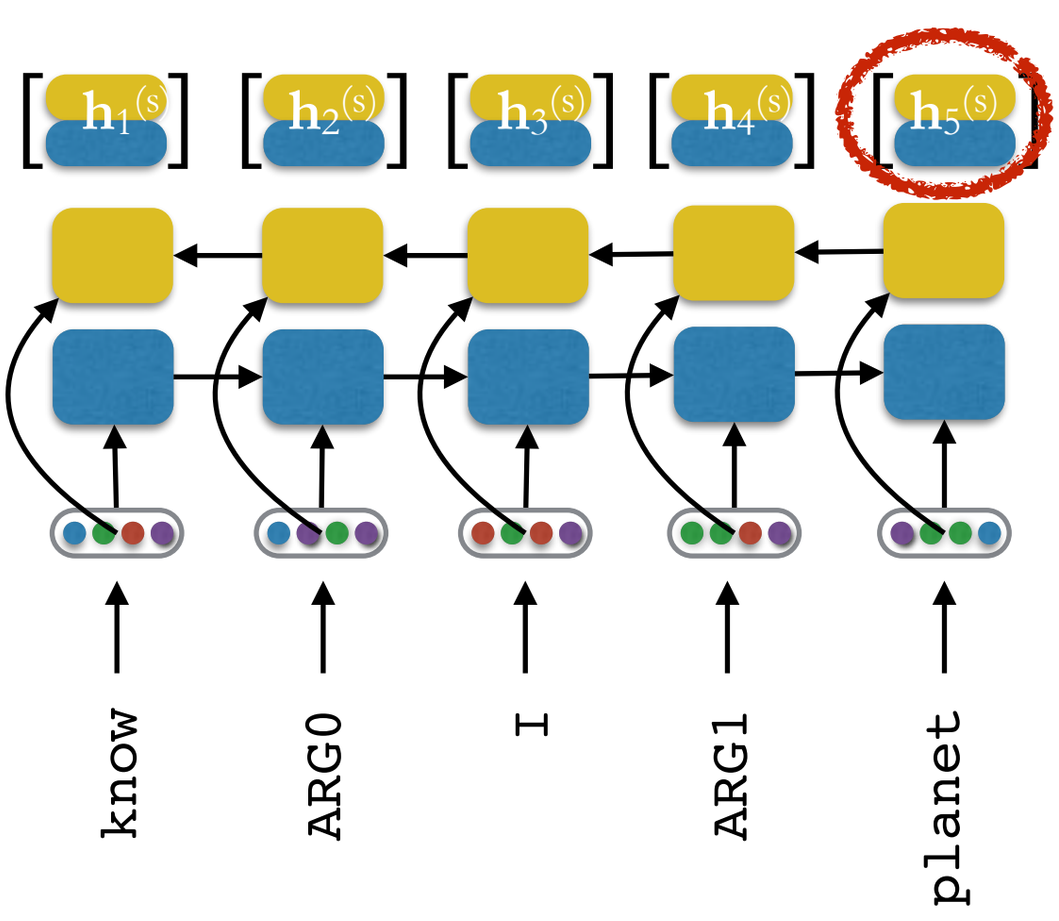
Encoding

Linearize —> RNN encoding

MR Generation




```
know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
```



Encoding

Hierarchical RNN encoding

Story Generation

Encoding

Hierarchical RNN encoding

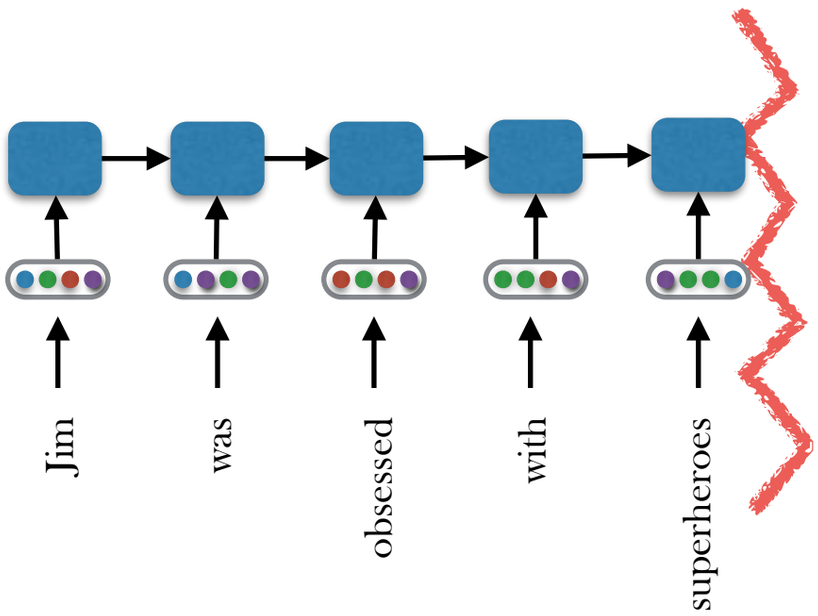
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

Story Generation

Encoding

Hierarchical RNN encoding

Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

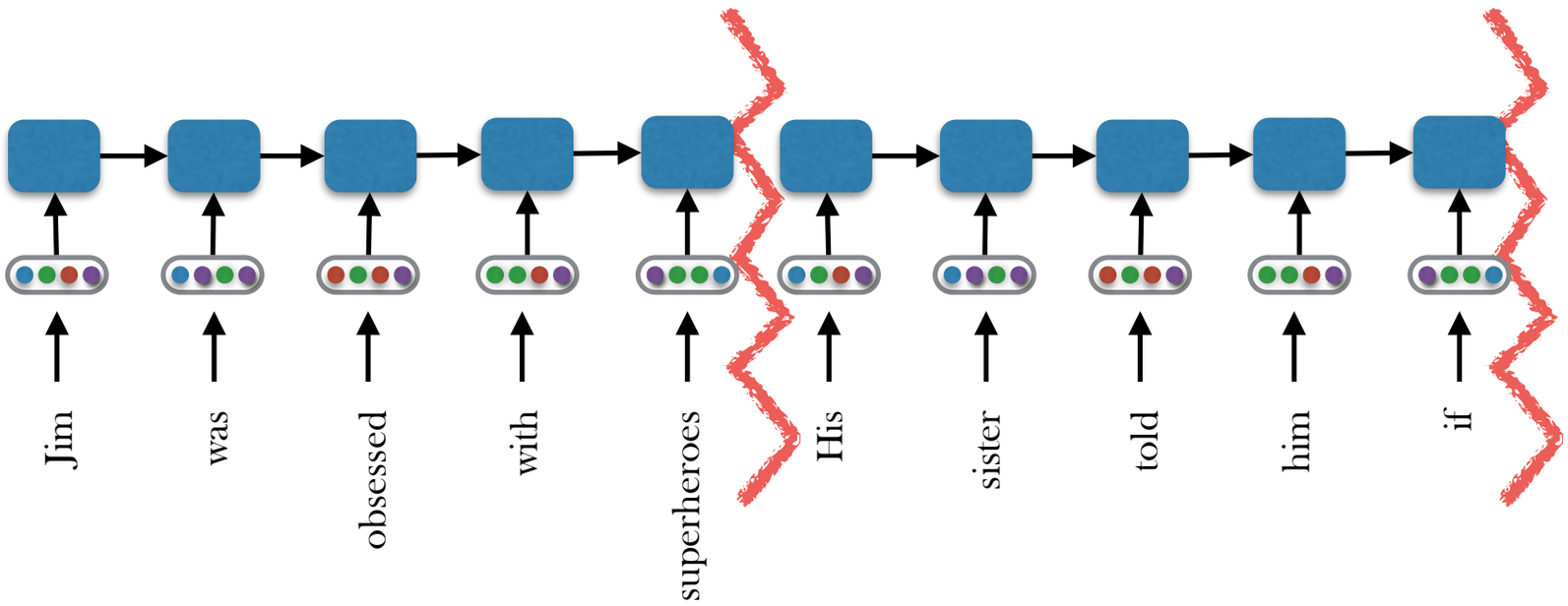


Story Generation

Encoding

Hierarchical RNN encoding

Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.



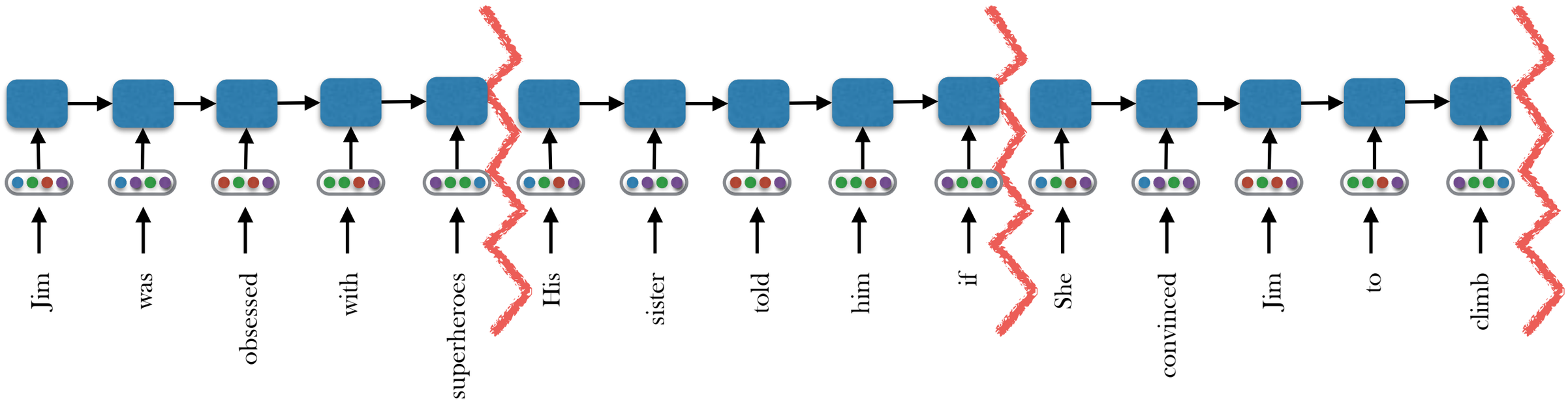
Story Generation

Encoding

Hierarchical RNN encoding

Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

Story Generation

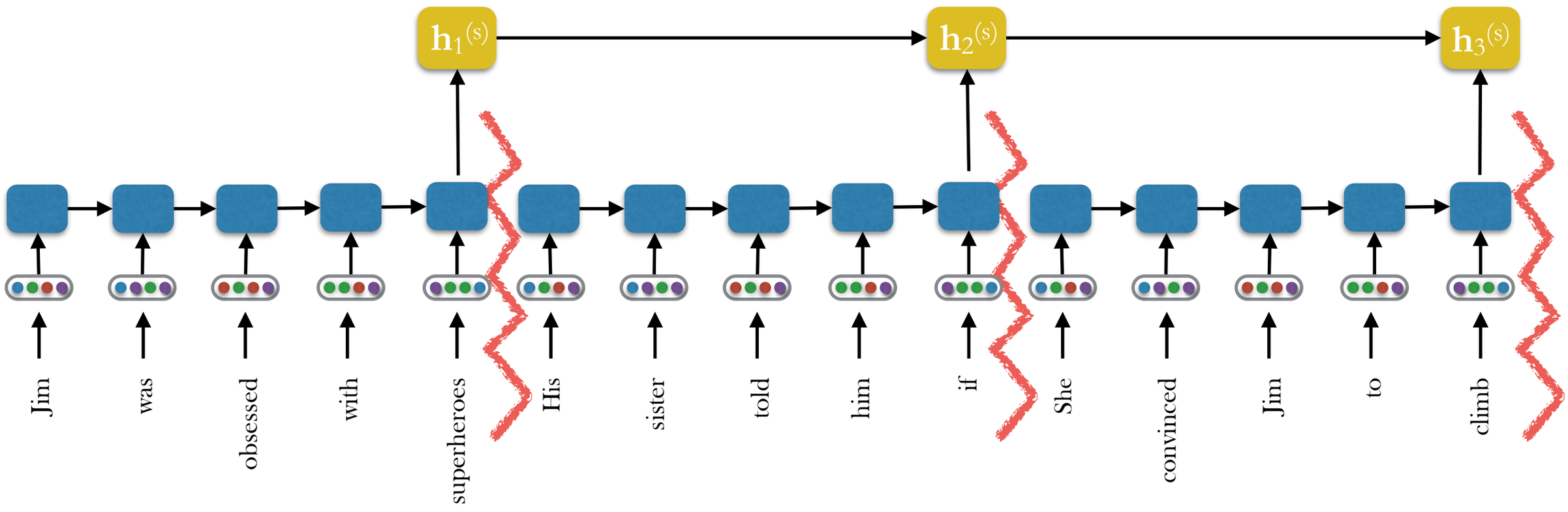


Encoding

Hierarchical RNN encoding

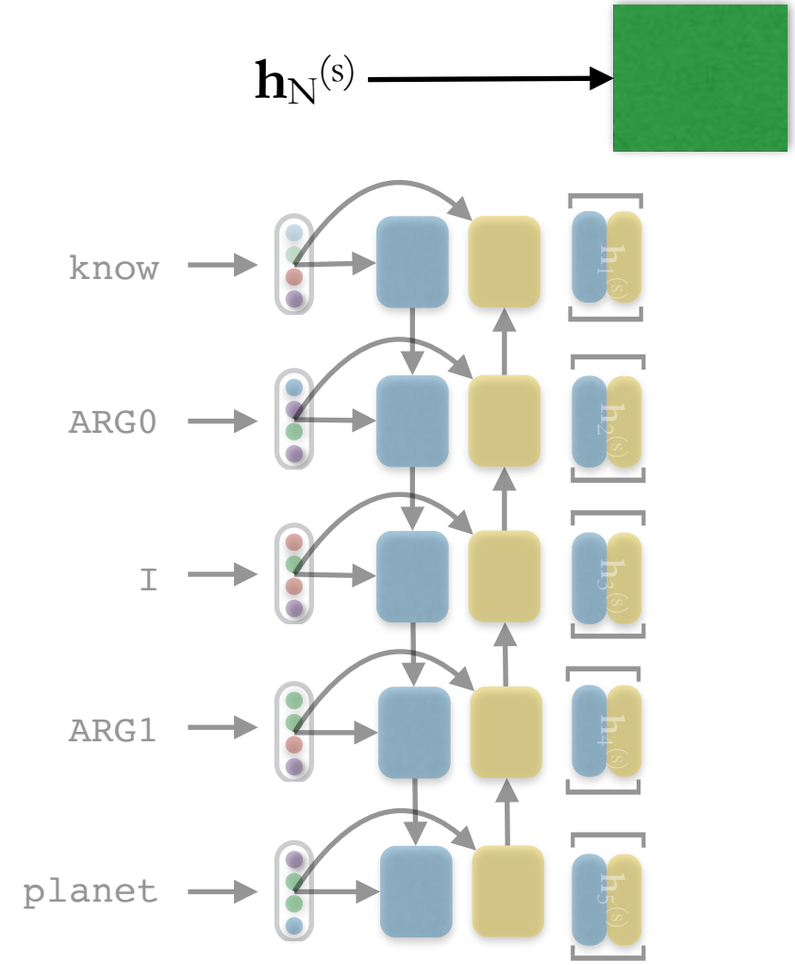
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.

Story Generation



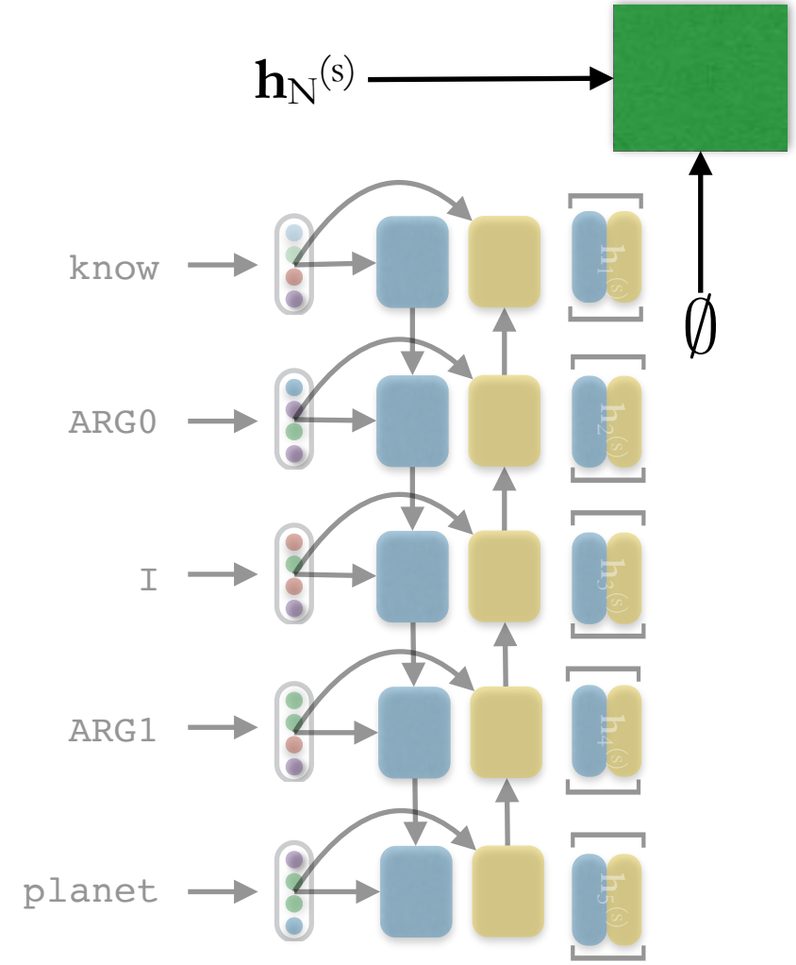
Decoding

Beam search (Left-to-Right)



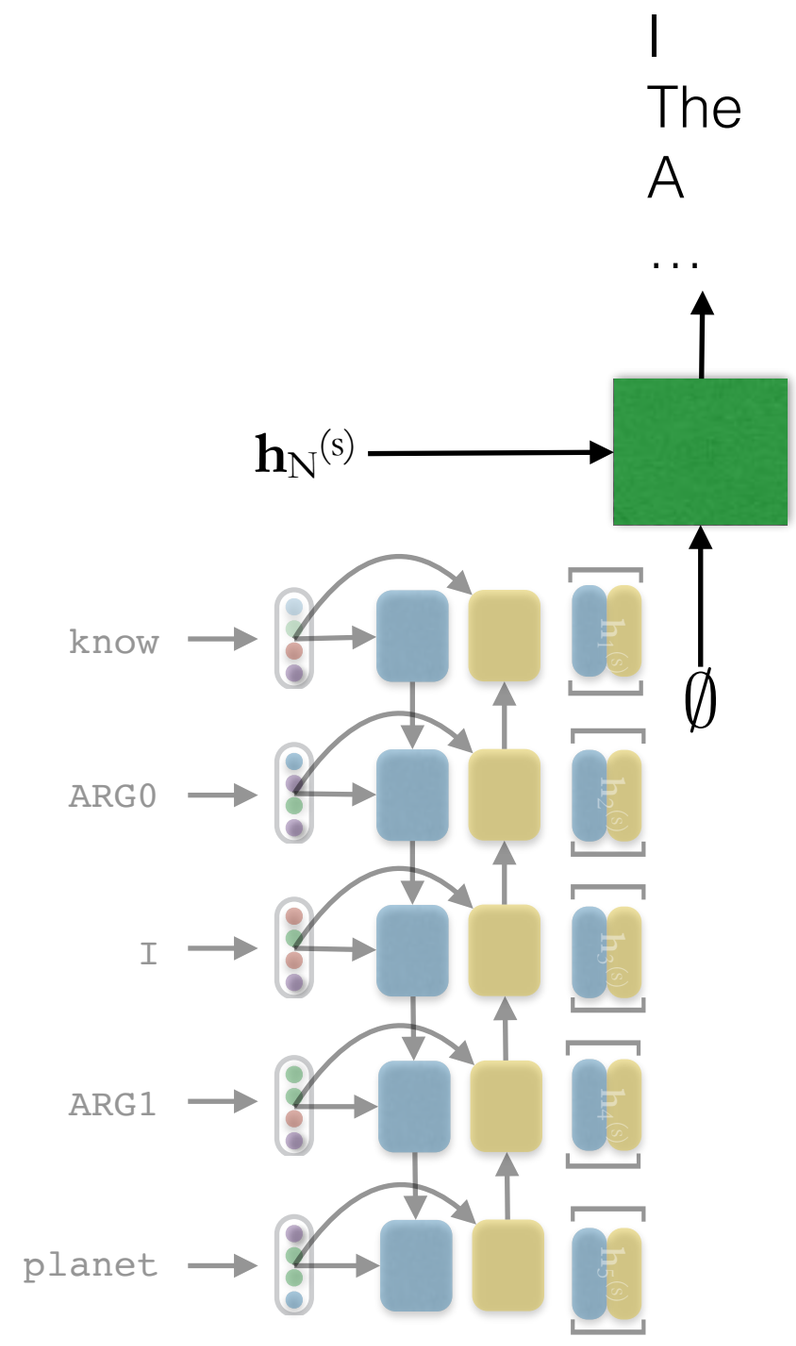
Decoding

Beam search (Left-to-Right)



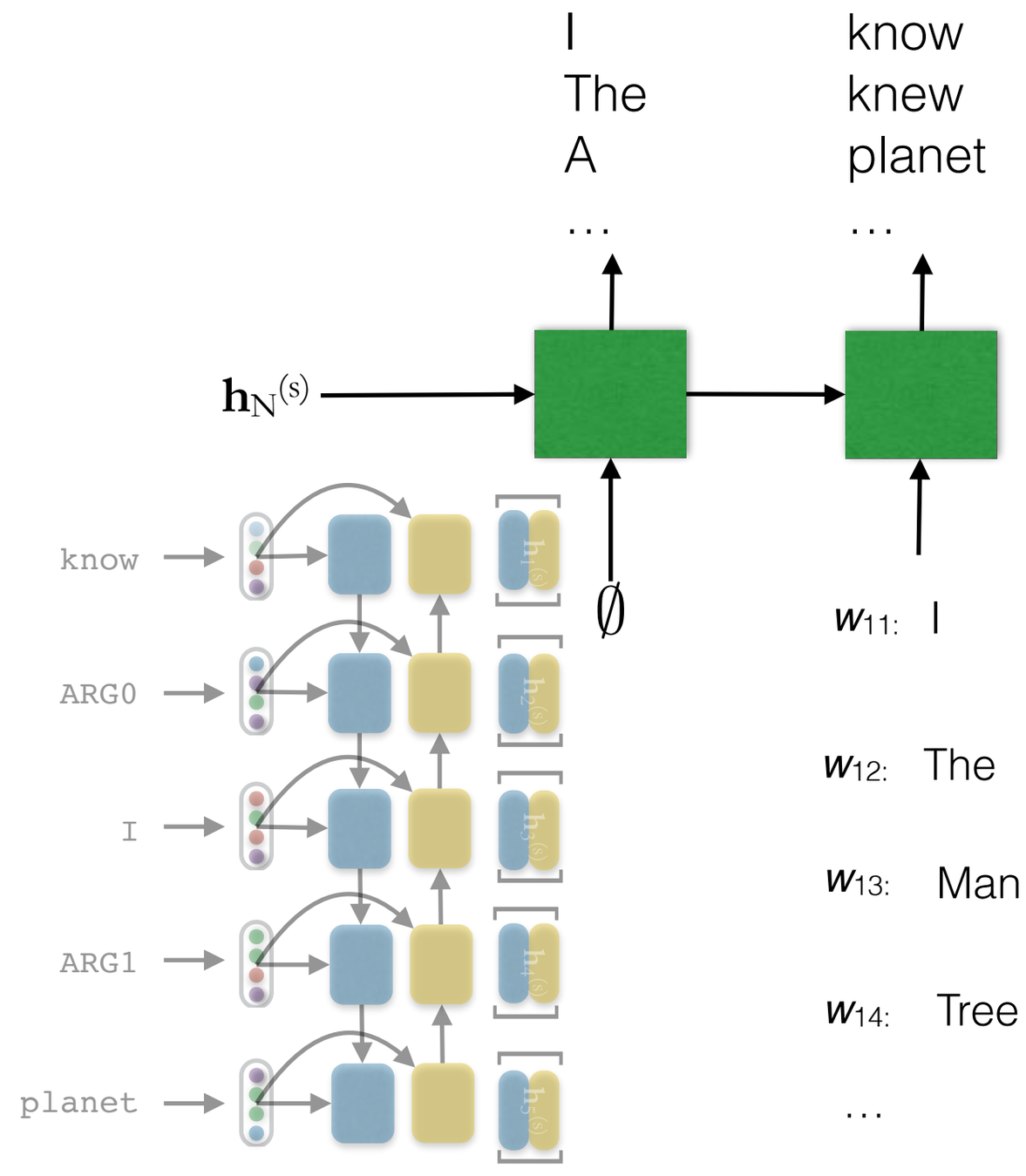
Decoding

Beam search (Left-to-Right)



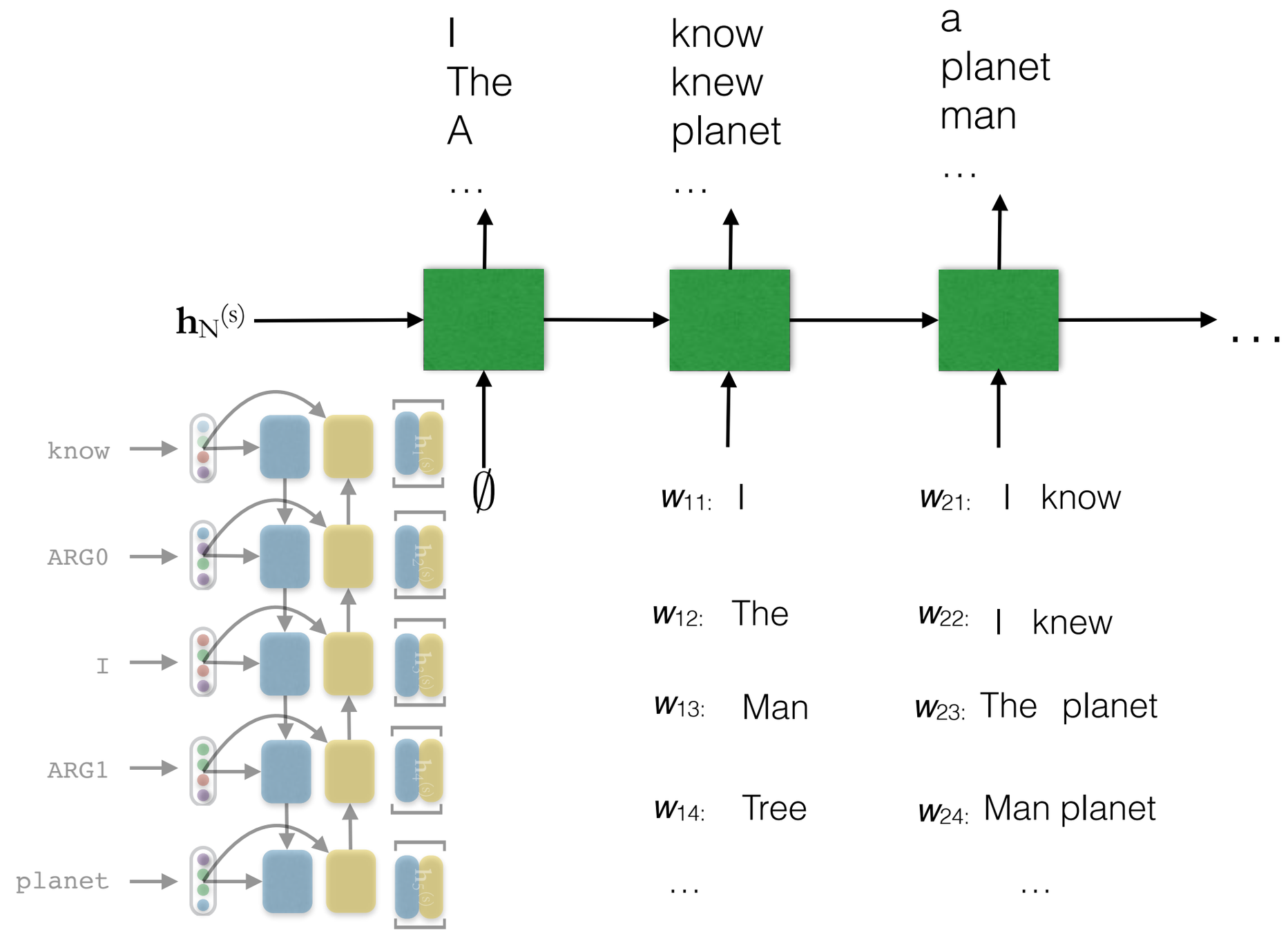
Decoding

Beam search (Left-to-Right)



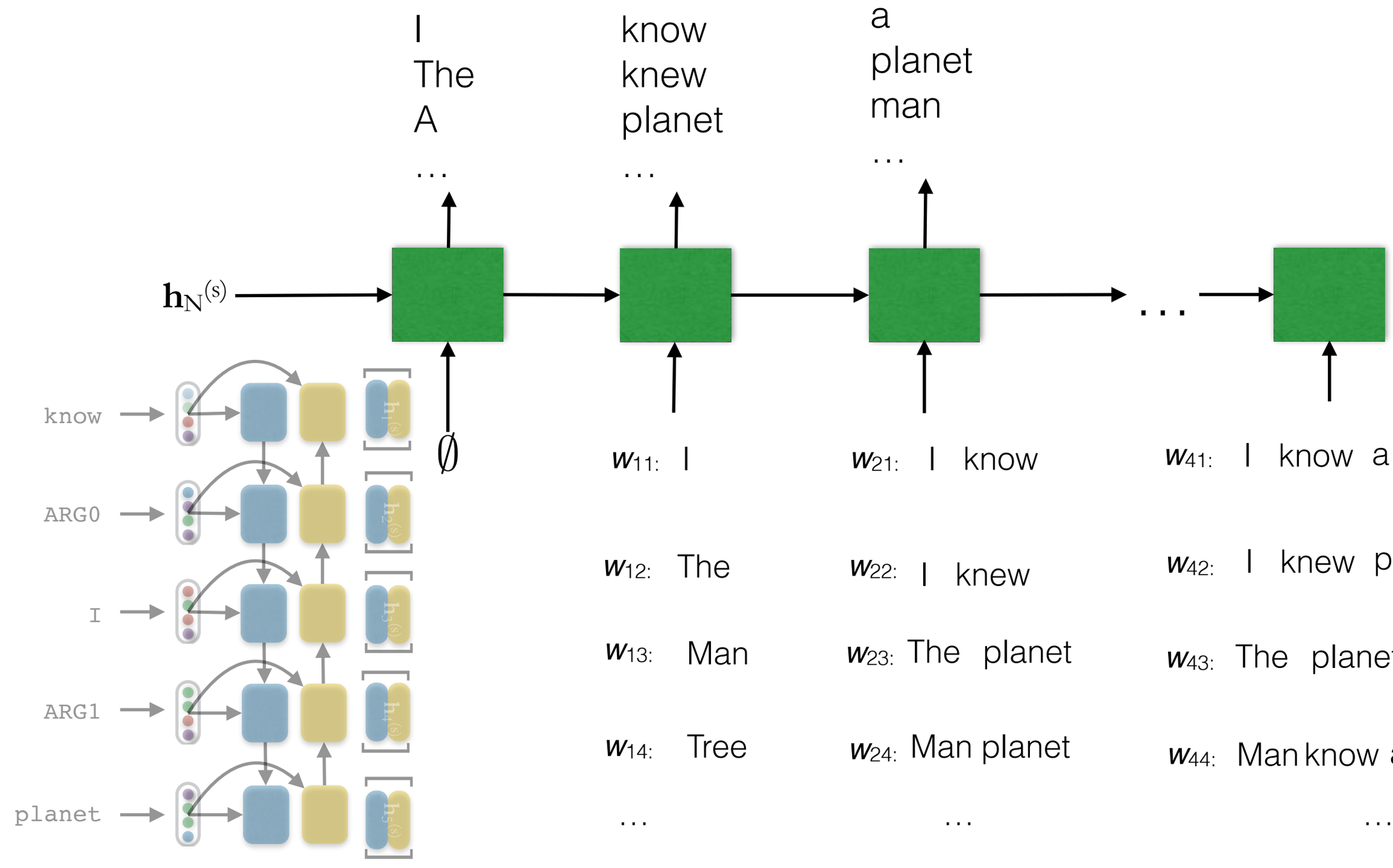
Decoding

Beam search (Left-to-Right)



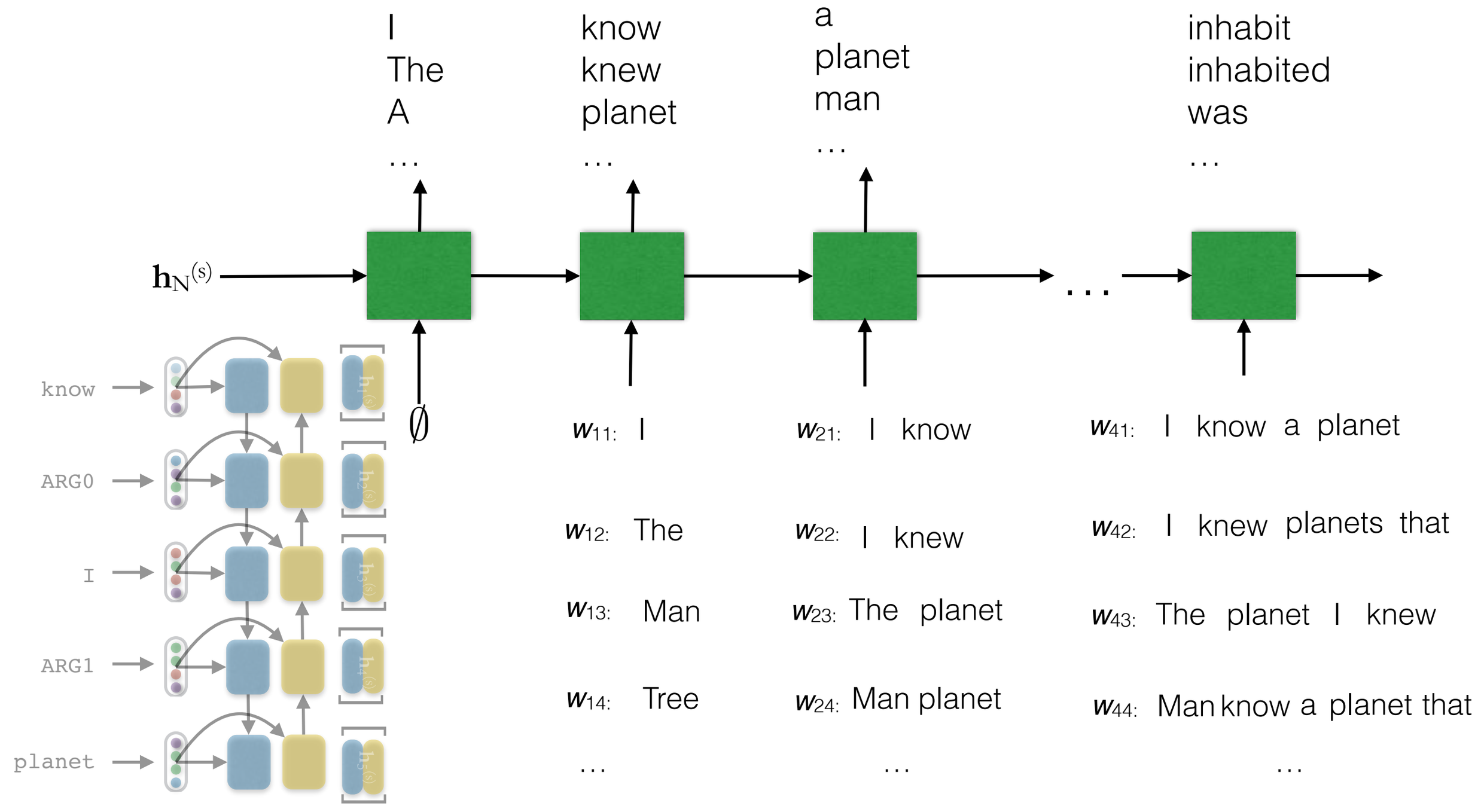
Decoding

Beam search (Left-to-Right)

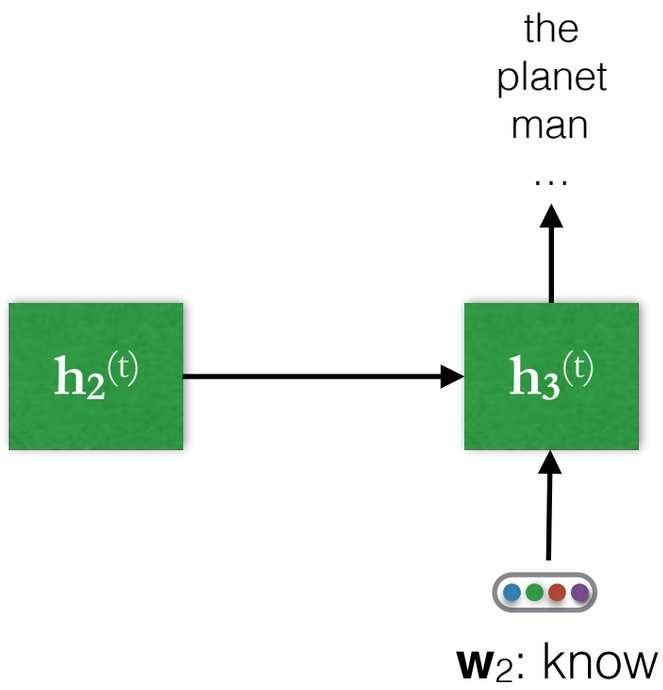


Decoding

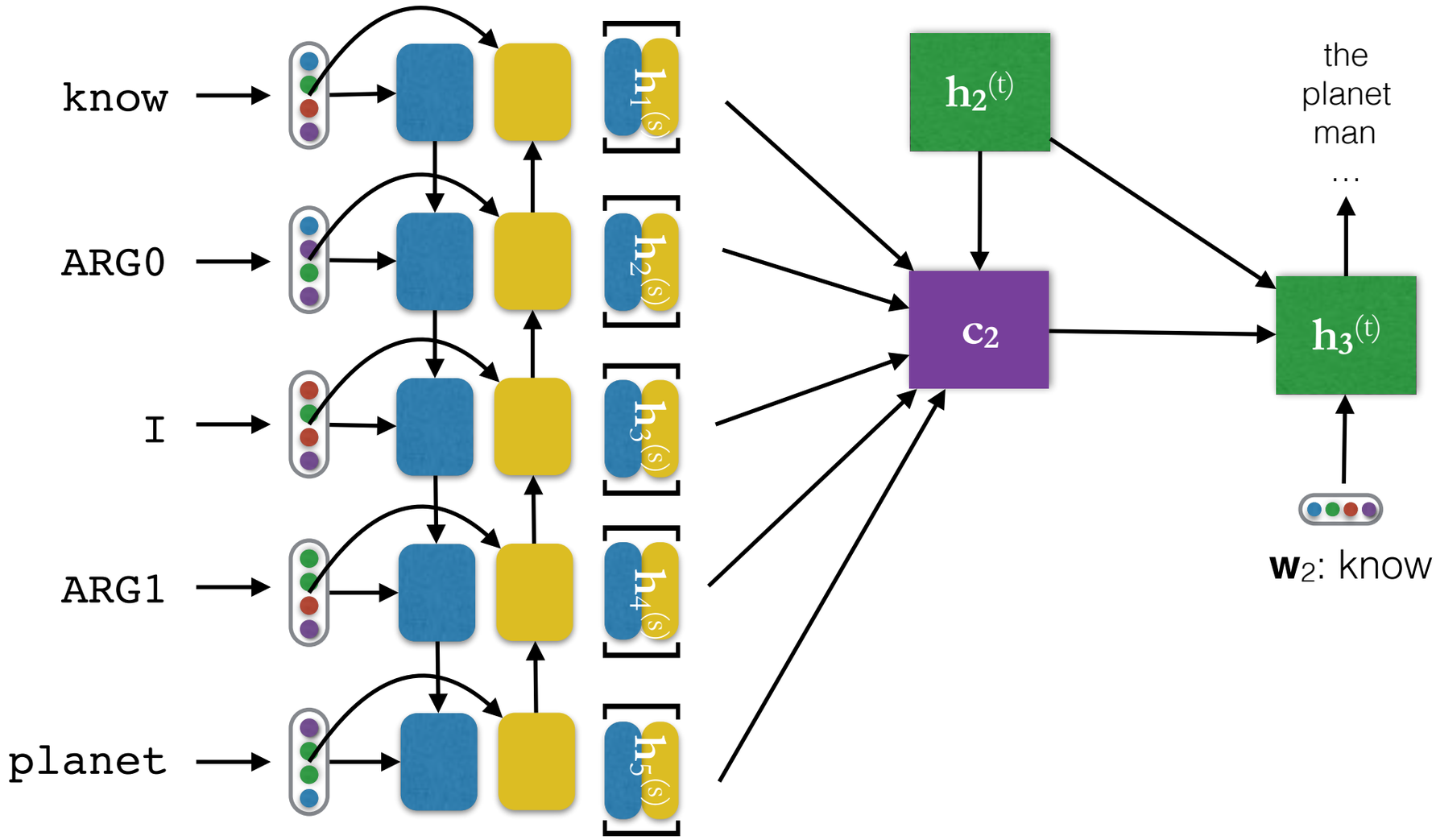
Beam search (Left-to-Right)



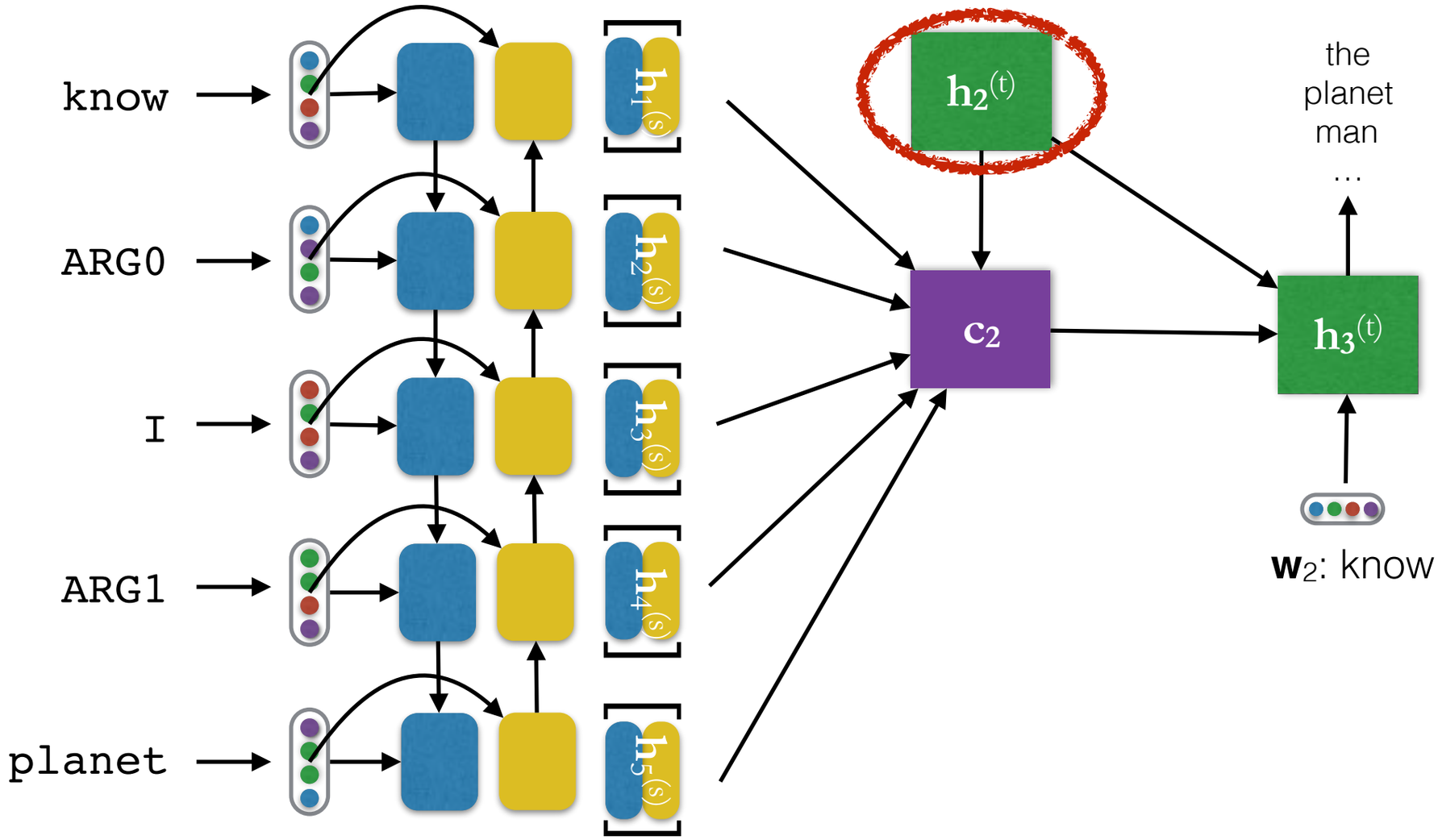
Attention



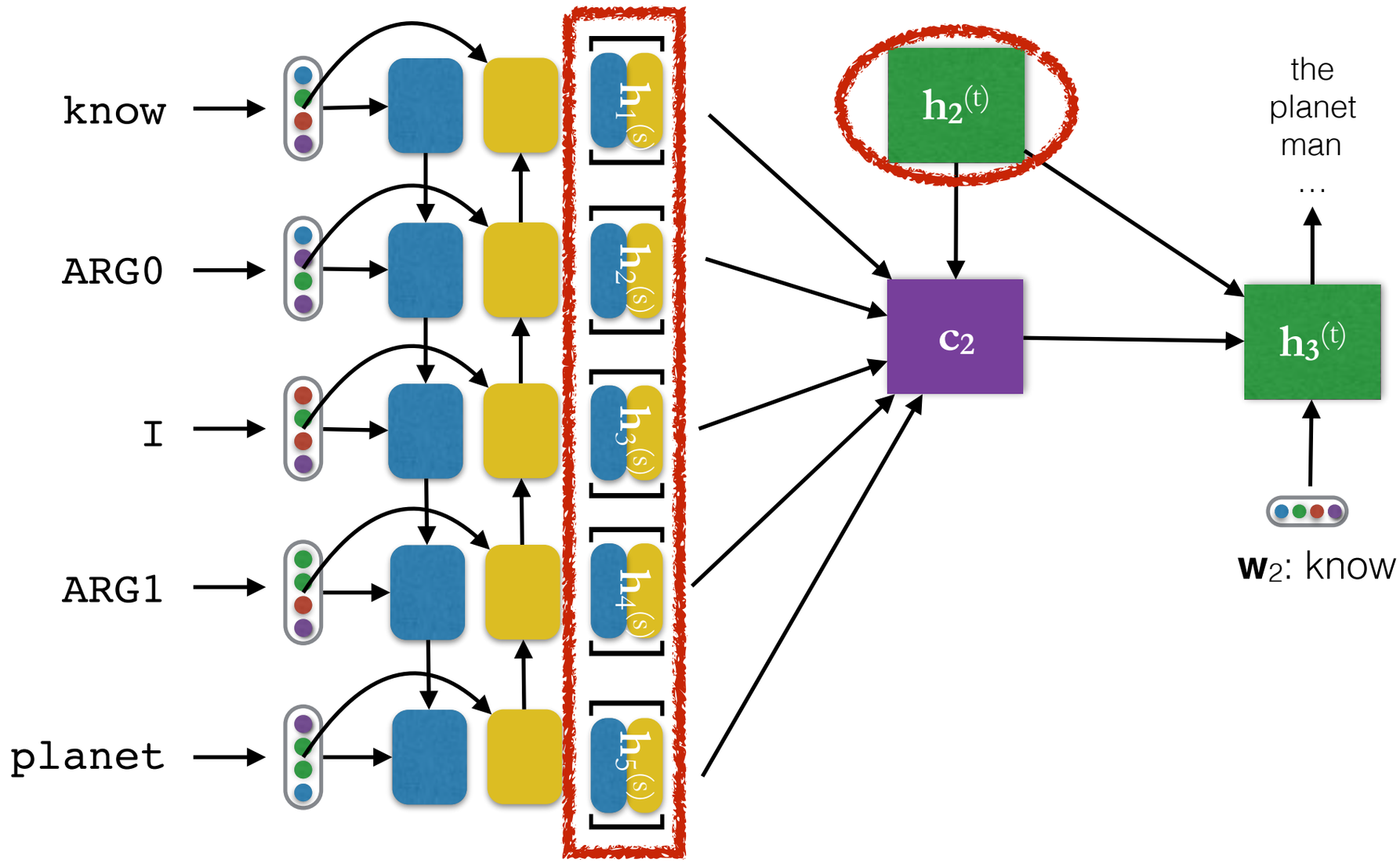
Attention


know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
…
w_2: know

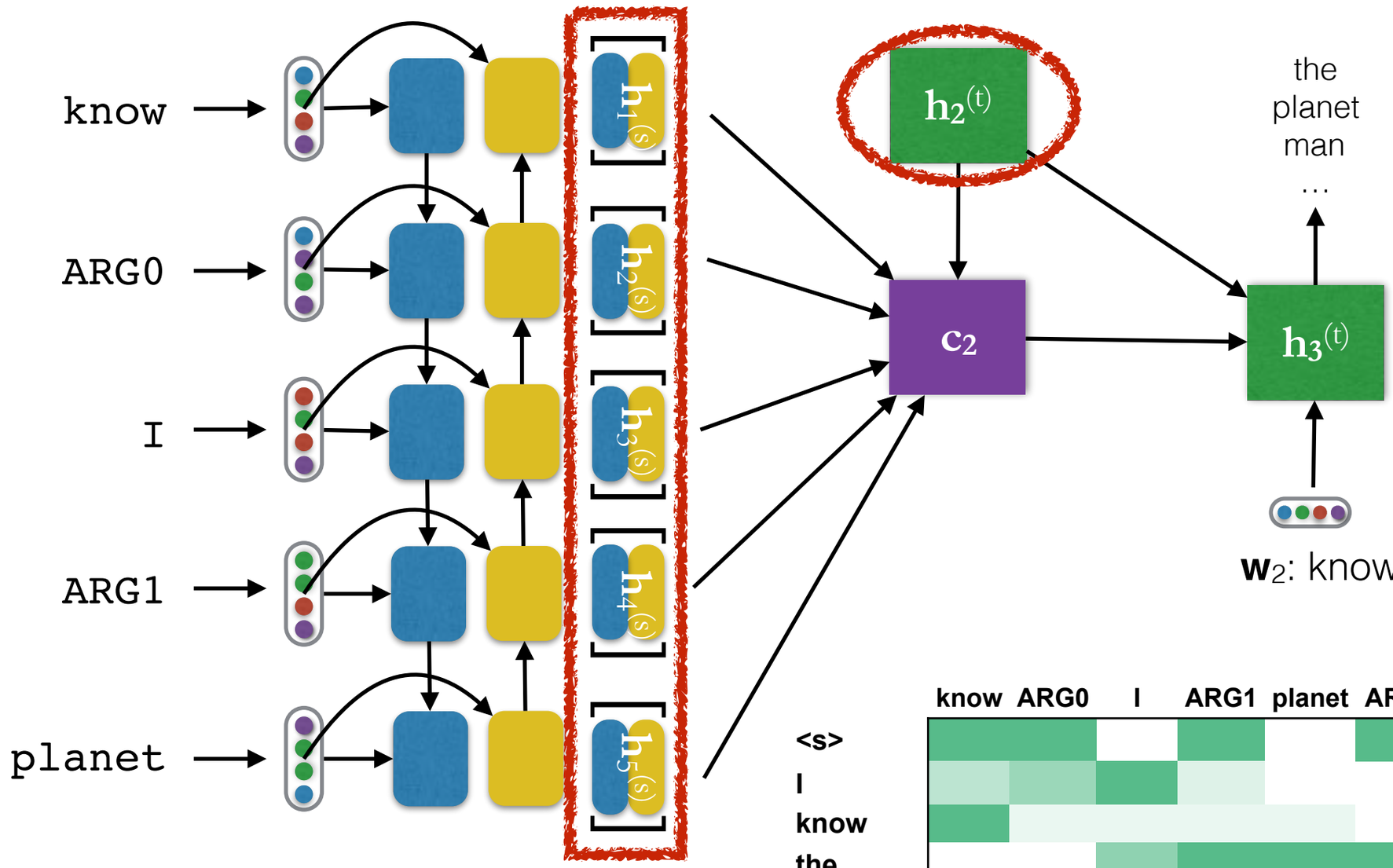
Attention

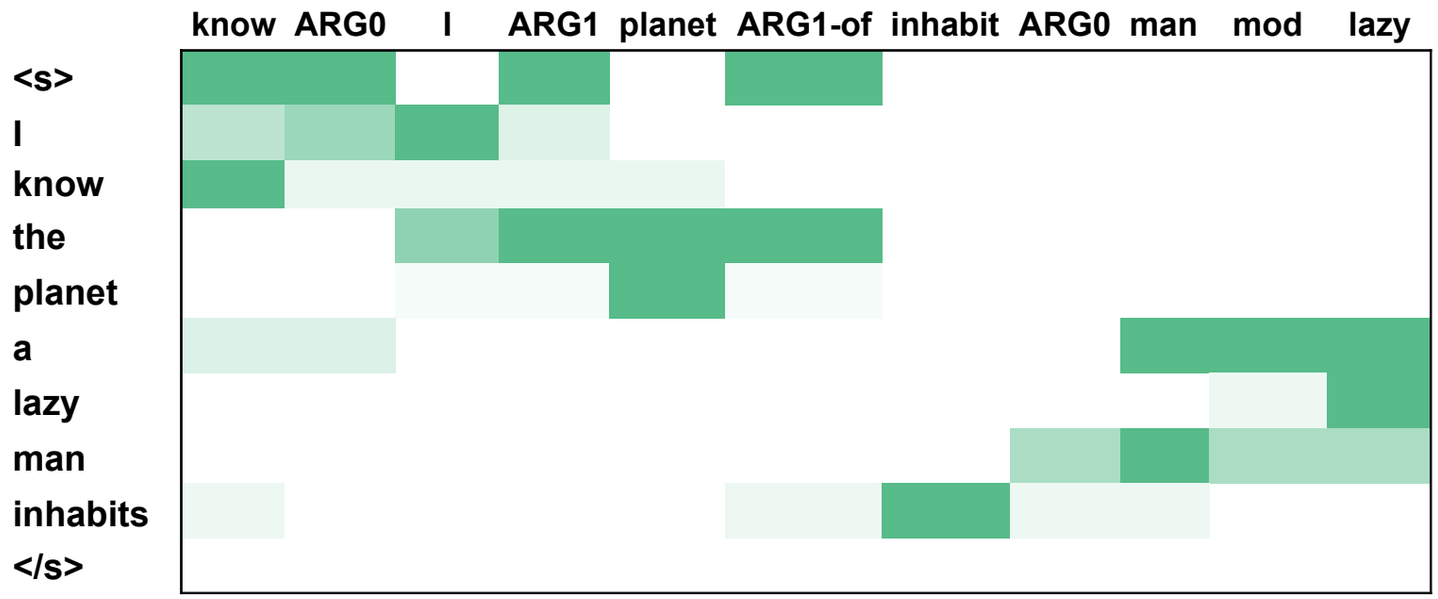

know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
...
w_2: know

Attention

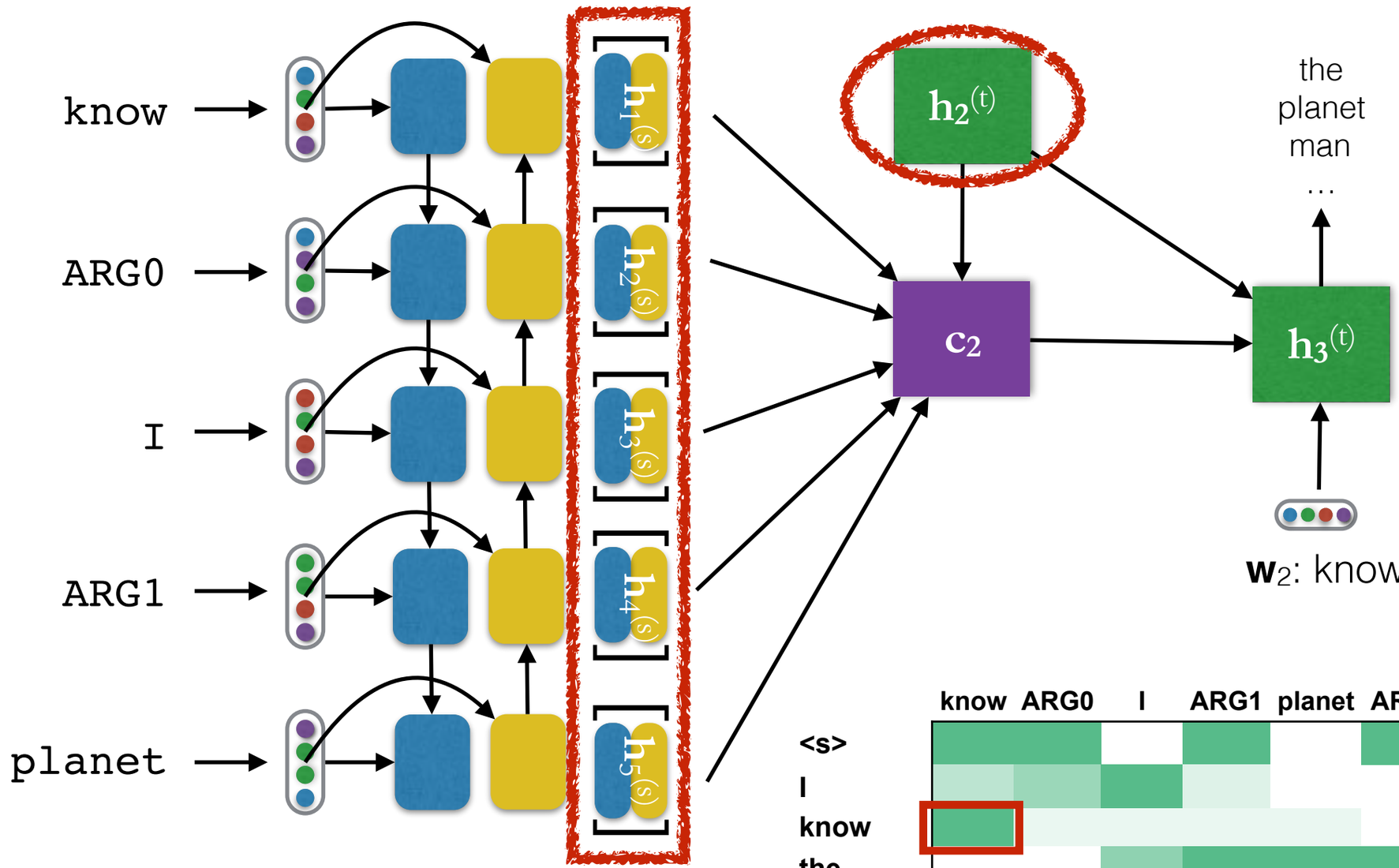

know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
...
w_2: know

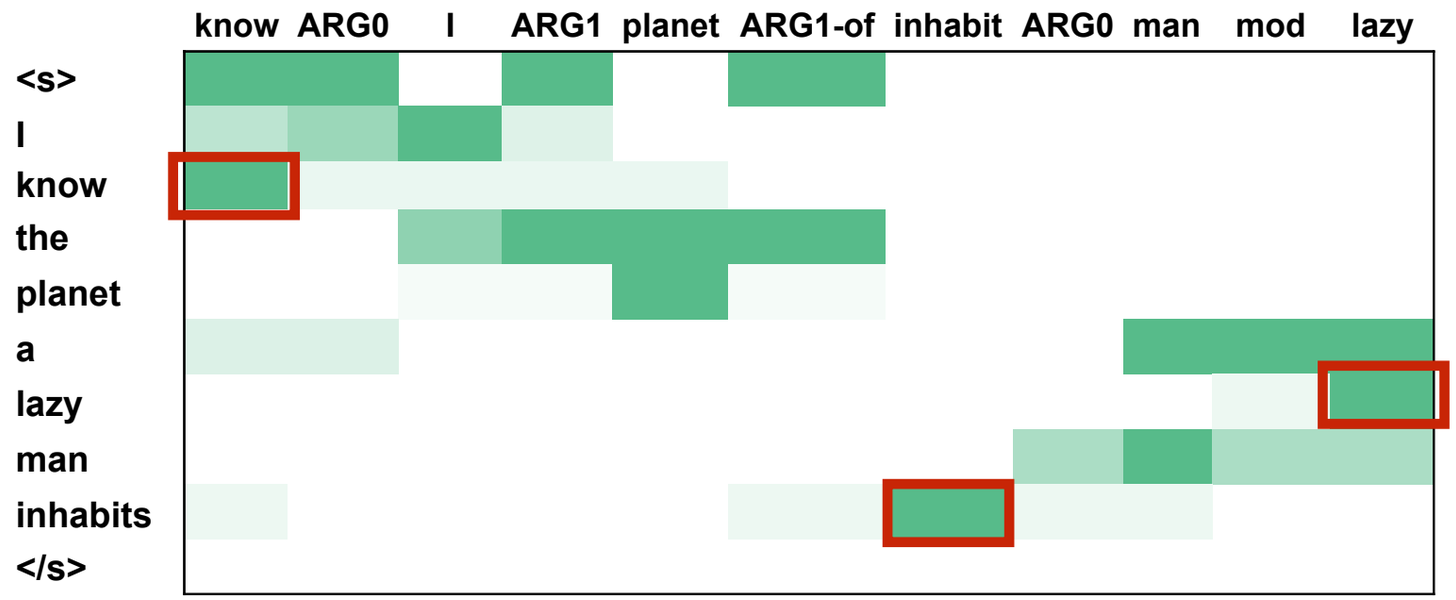
Attention


know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
…
w_2: know

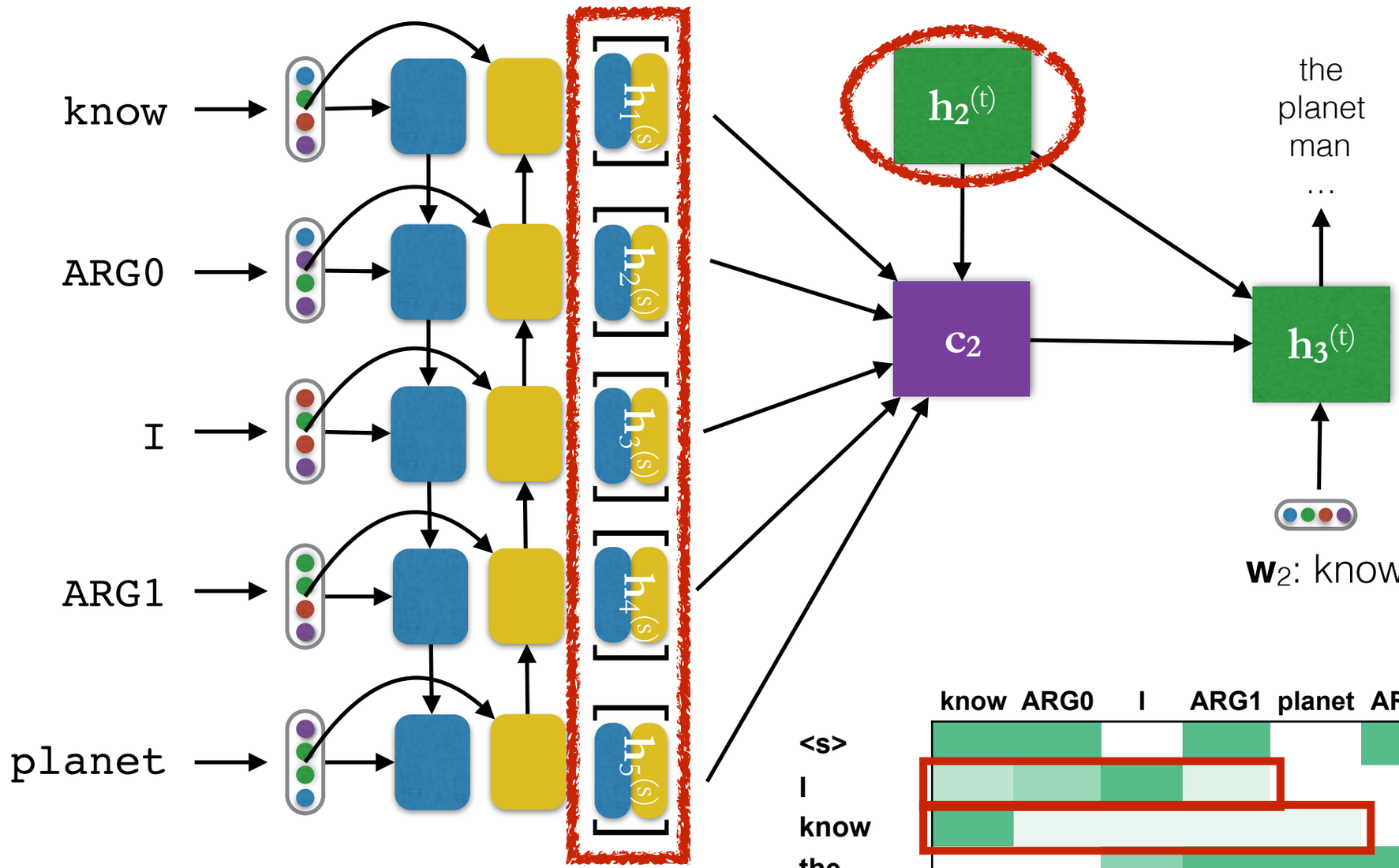

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
<s>
I
know
the
planet
a
lazy
man
inhabits
</s>

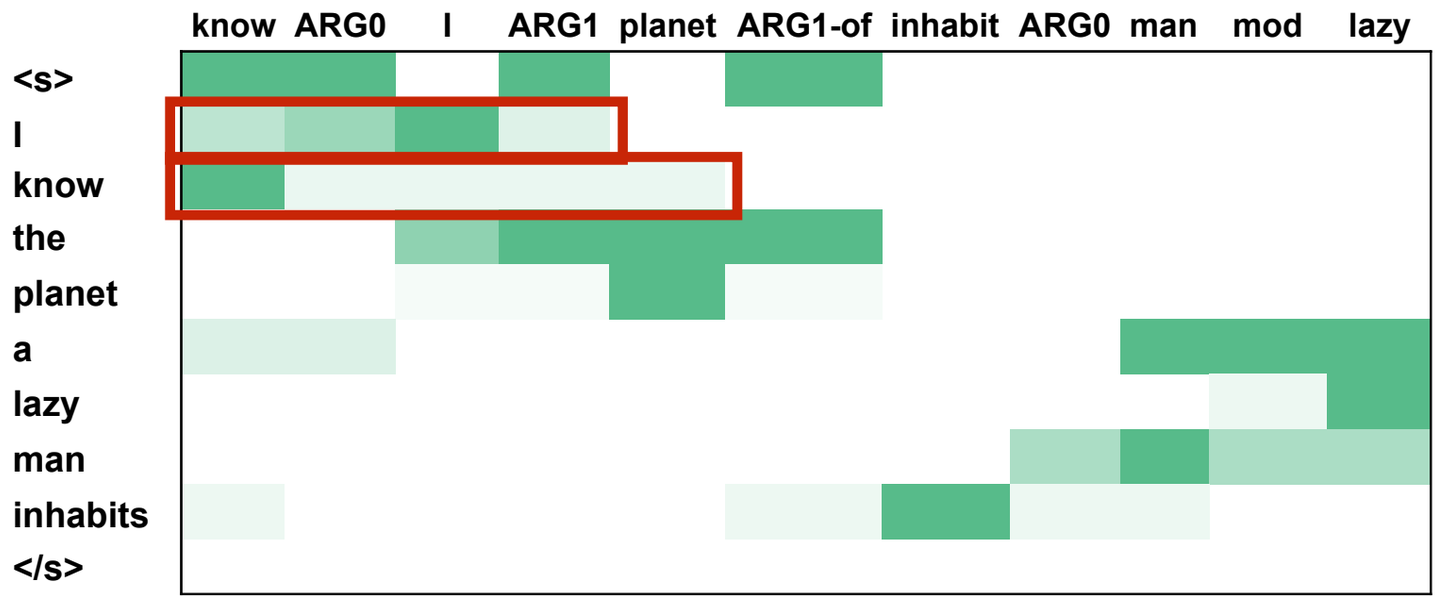
Attention


know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
…
w_2: know

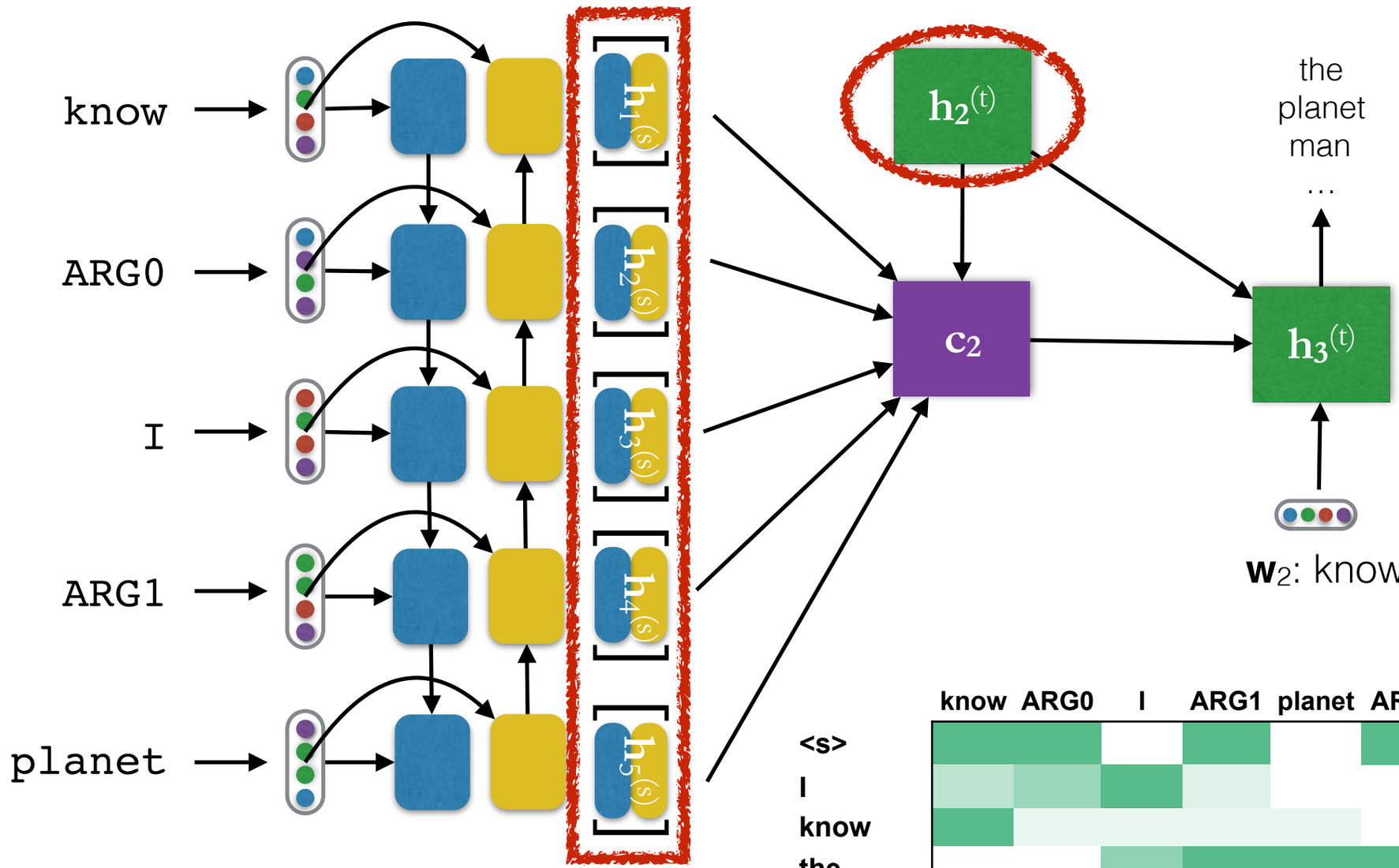

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
<s>
I
know
the
planet
a
lazy
man
inhabits
</s>

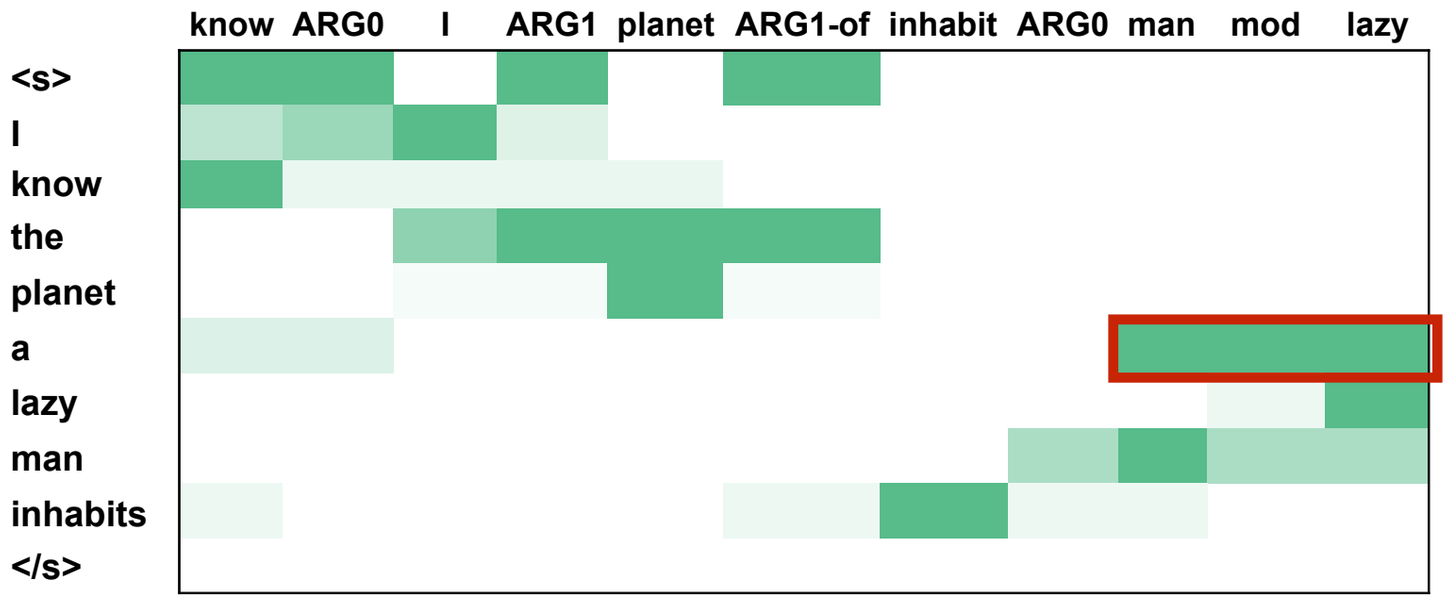
Attention


know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
…
w_2: know


know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
<s>
I
know
the
planet
a
lazy
man
inhabits
</s>

Attention


know
ARG0
I
ARG1
planet
$h_1^{(s)}$
$h_2^{(s)}$
$h_3^{(s)}$
$h_4^{(s)}$
$h_5^{(s)}$
$h_2^{(t)}$
c_2
$h_3^{(t)}$
the
planet
man
…
w_2: know


know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
<s>
I
know
the
planet
a
lazy
man
inhabits
</s>

Issues to Address

Max-probability search

Issues to Address

Max-probability search

Issues

- short / similar outputs
- no guarantee that input is covered

I know a planet . </s>
I know the planet . </s>
I know the planet a </s>
I know the planet a lazy man .</s>

Issues to Address

Max-probability search

Issues

- short / similar outputs
- no guarantee that input is covered

I know a planet . </s>
I know the planet . </s>
I know the planet a </s>
I know the planet a lazy man .</s>

Cheap Solutions

- Length penalty
- Coverage penalty (w/ attention weights)

Issues to Address

Issues to Address

Sparsity

Issues to Address

Sparsity

- Anonymize NE tokens

Issues to Address

Sparsity

- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

Issues to Address

Sparsity

- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input

Issues to Address

Sparsity

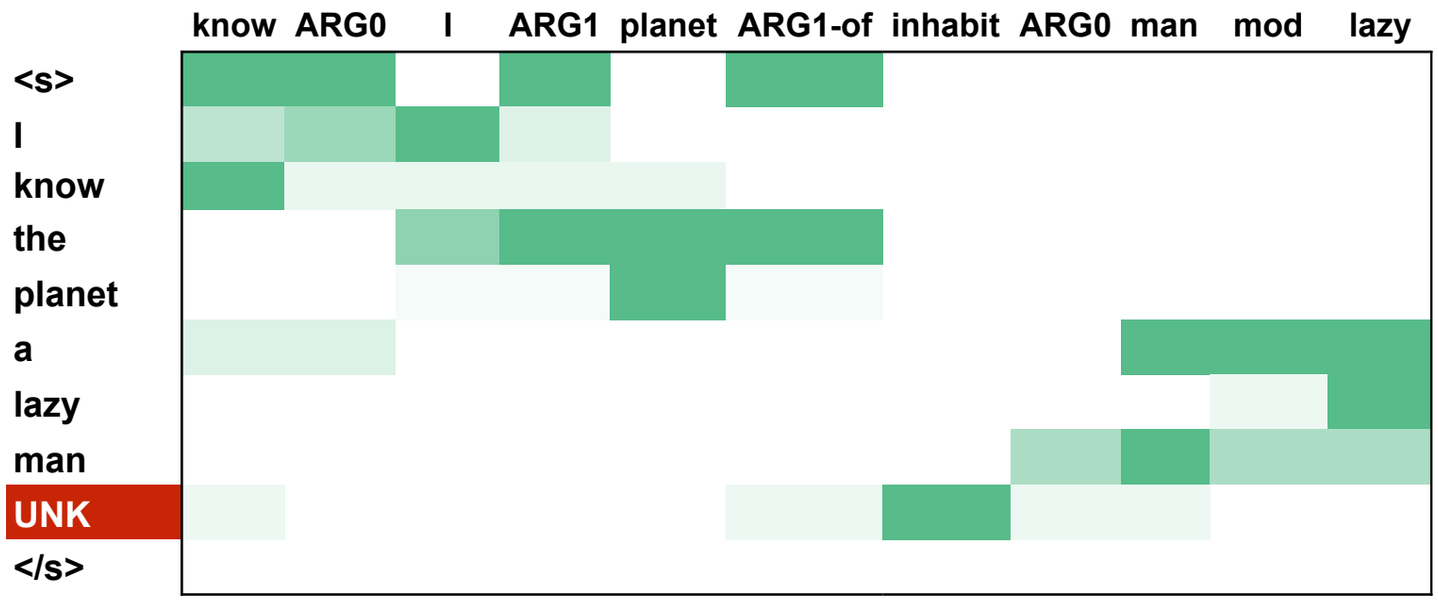
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



Issues to Address

Sparsity

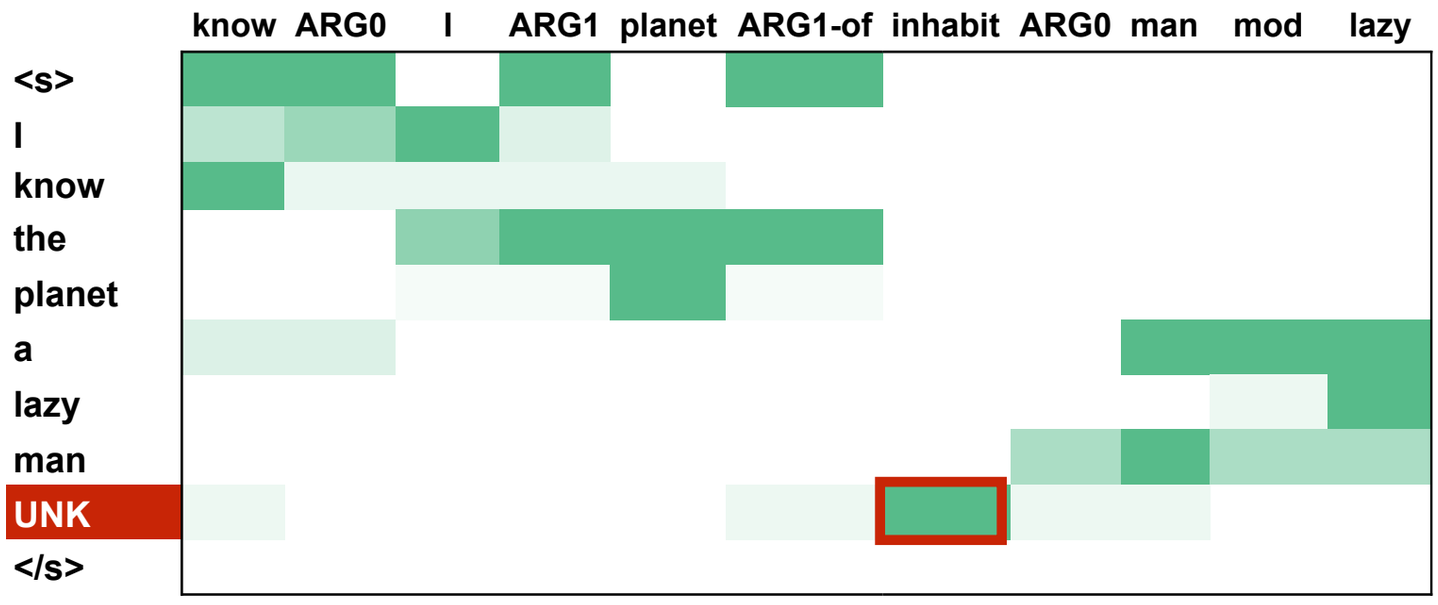
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



Issues to Address

Sparsity

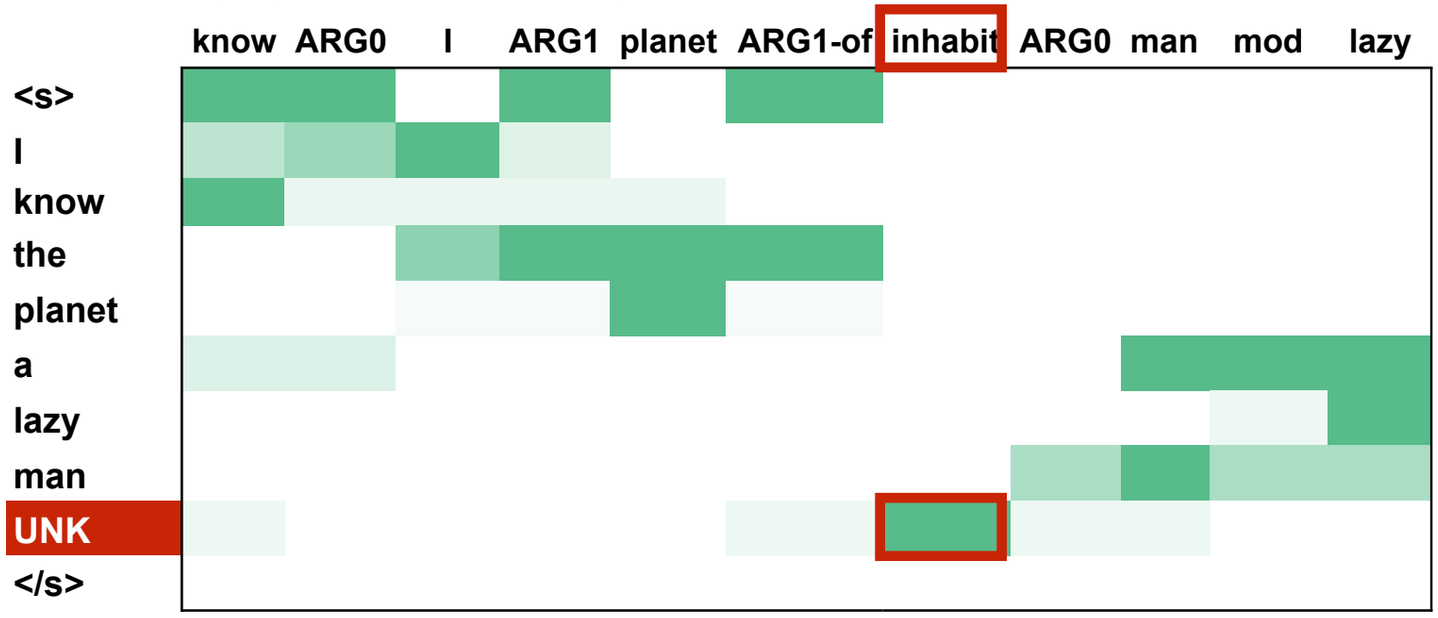
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



Issues to Address

Sparsity

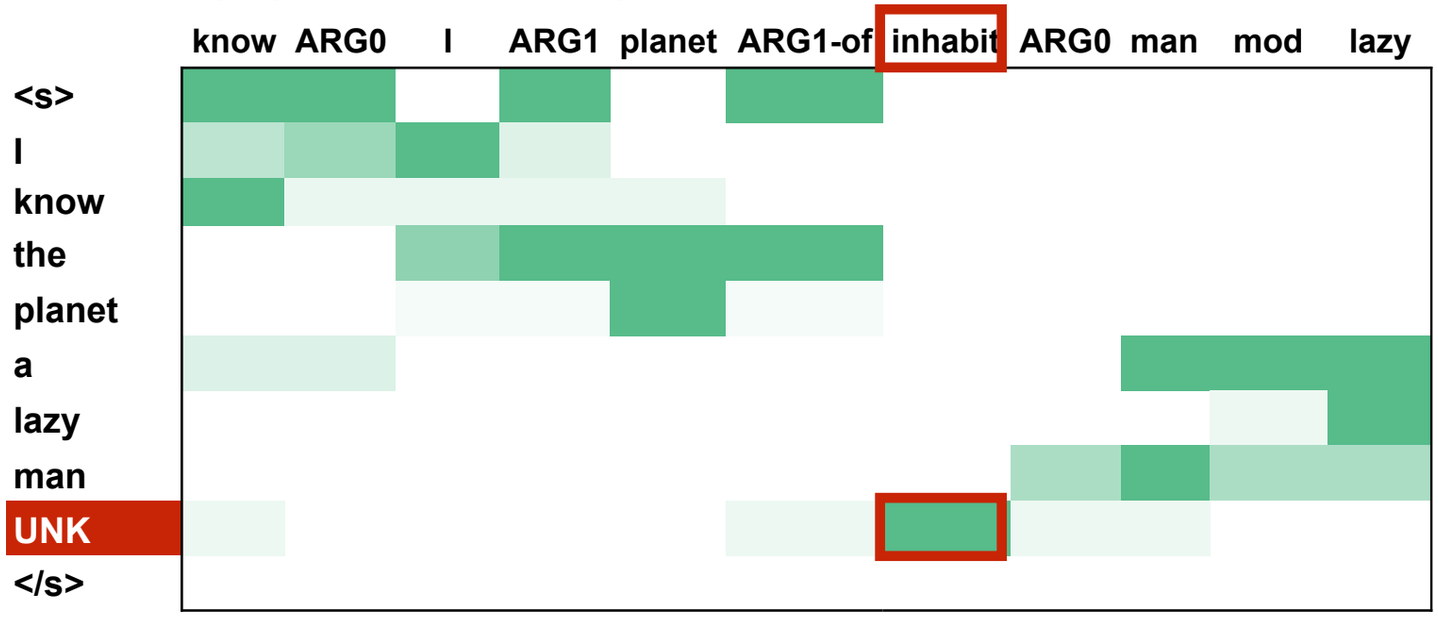
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



input	output	prob
inhabit	**inhabits**	**0.6**
	inhabit	0.2
	inhabiting	0.1
	…	…

Issues to Address

Sparsity

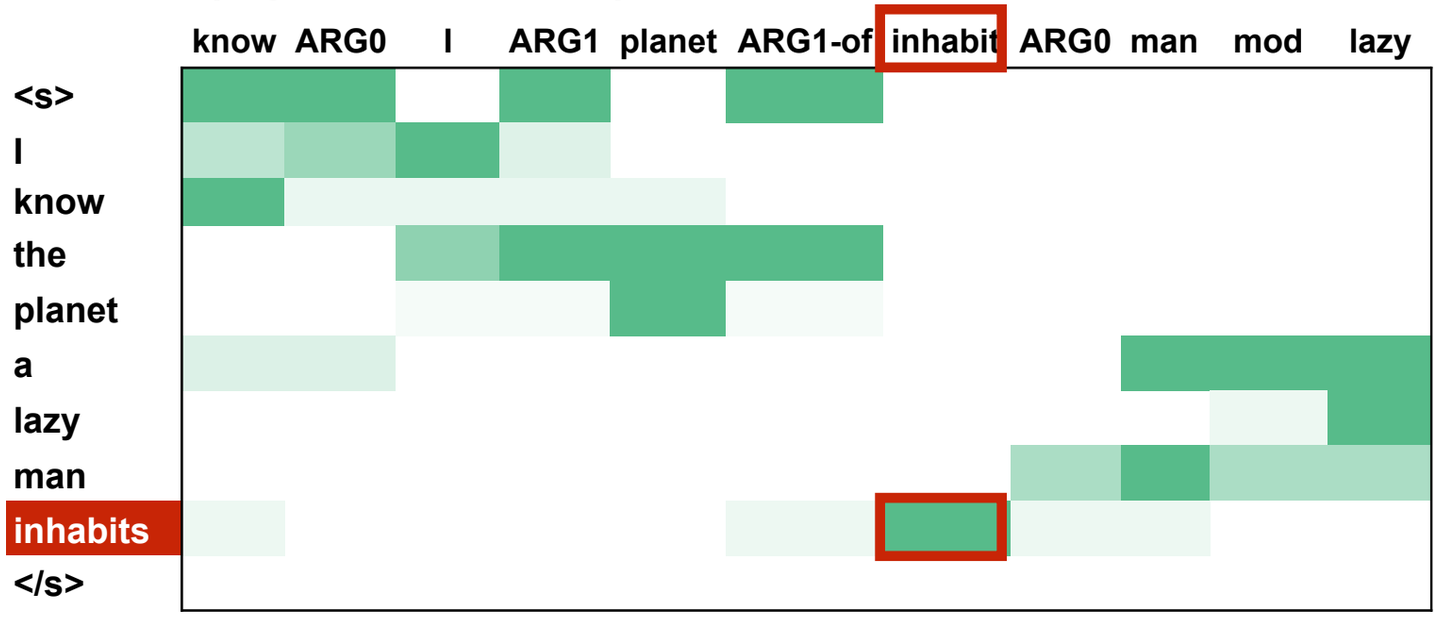
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



input	output	prob
inhabit	**inhabits**	**0.6**
	inhabit	0.2
	inhabiting	0.1
	…	…

Issues to Address

Sparsity

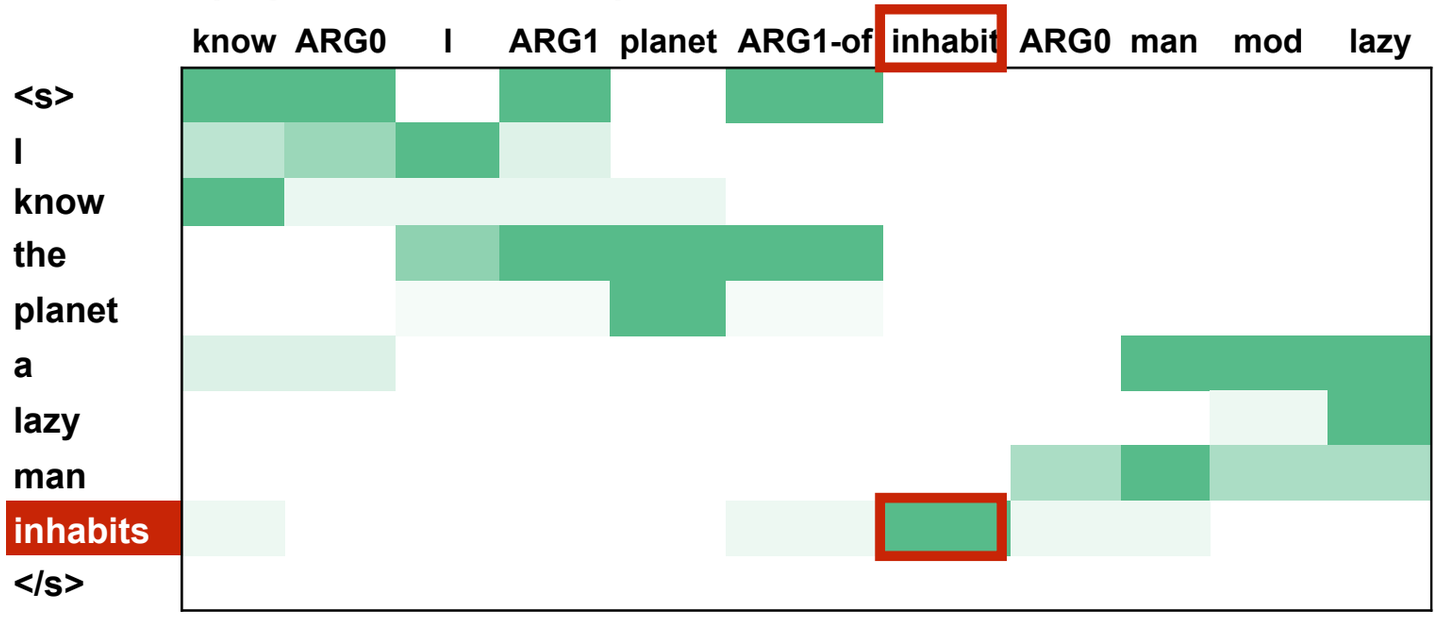
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



input	output	prob
inhabit	**inhabits**	**0.6**
	inhabit	0.2
	inhabiting	0.1
	…	…

- CCG Parsing

Issues to Address

Sparsity

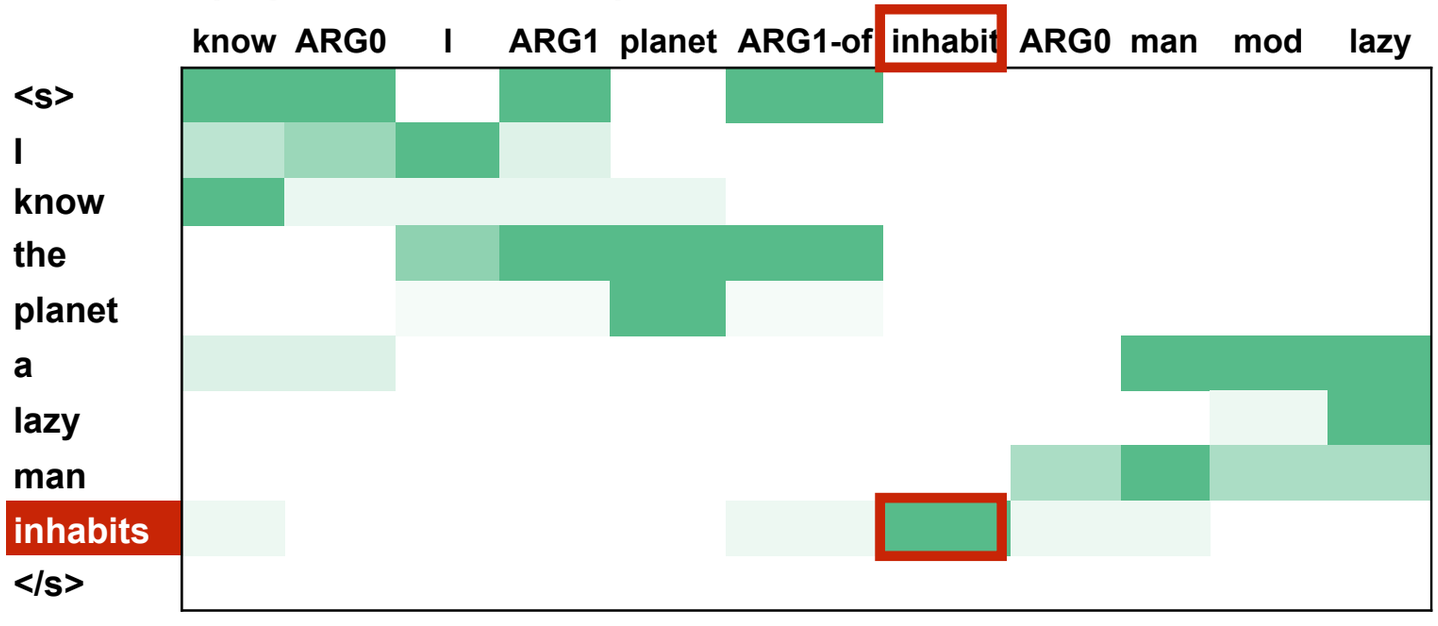
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



input	output	prob
inhabit	**inhabits**	**0.6**
	inhabit	0.2
	inhabiting	0.1
	…	…

- ~~CCG Parsing~~

Issues to Address

Sparsity

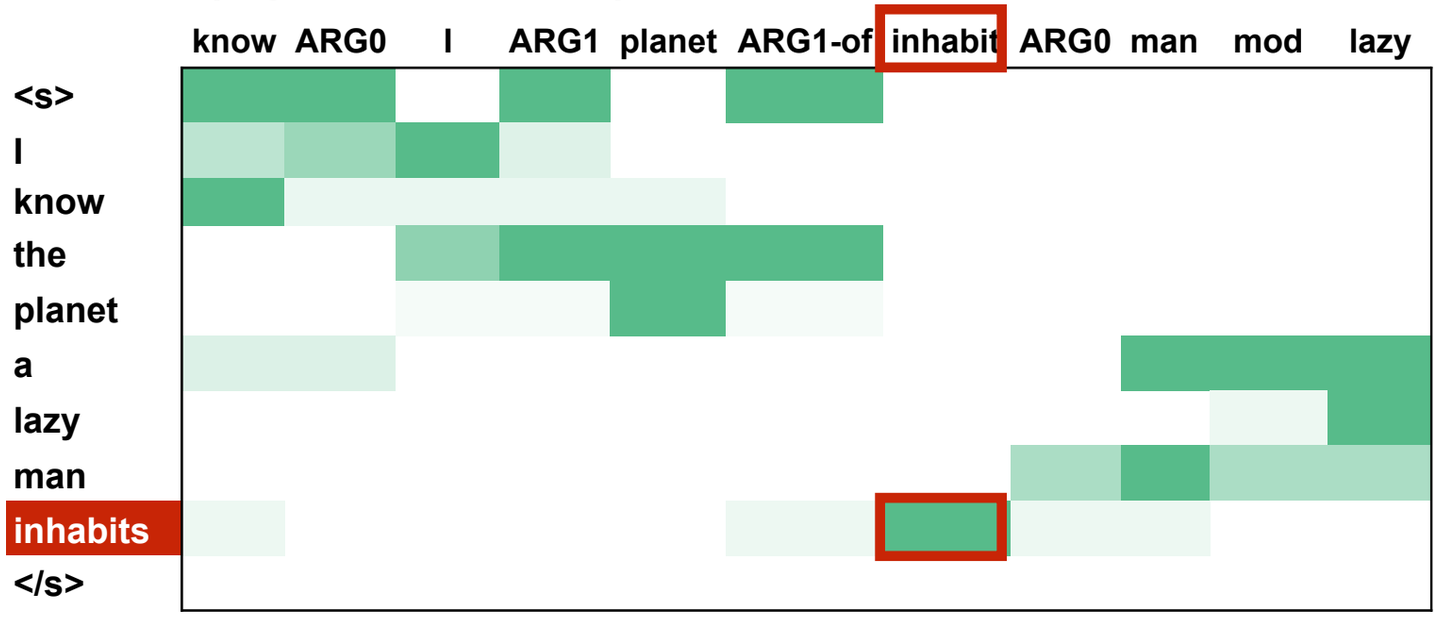
- Anonymize NE tokens

```
state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 …
```

President Obama stated that **UK** should keep …

person_name_0 stated that **country_name_1** should keep …

- Copy from input



input	output	prob
inhabit	**inhabits**	**0.6**
	inhabit	0.2
	inhabiting	0.1
	…	…

- ~~CCG Parsing~~ Data Augmentation

Open Questions

Open Questions

Representations

- Probably shouldn't treat all inputs as strings…

Open Questions

Representations

- Probably shouldn’t treat all inputs as strings…

Document Plans

- Maybe shouldn’t treat output as stream of strings…

Open Questions

Representations

- Probably shouldn't treat all inputs as strings…

Document Plans

- Maybe shouldn't treat output as stream of strings…

Optimize on some goal / Creative evaluation

- Don't want just good-looking string of [X_language]…

Open Questions

Representations

- Probably shouldn't treat all inputs as strings…

Document Plans

- Maybe shouldn't treat output as stream of strings…

Optimize on some goal / Creative evaluation

- Don't want just good-looking string of [X_language]…

THANK YOU