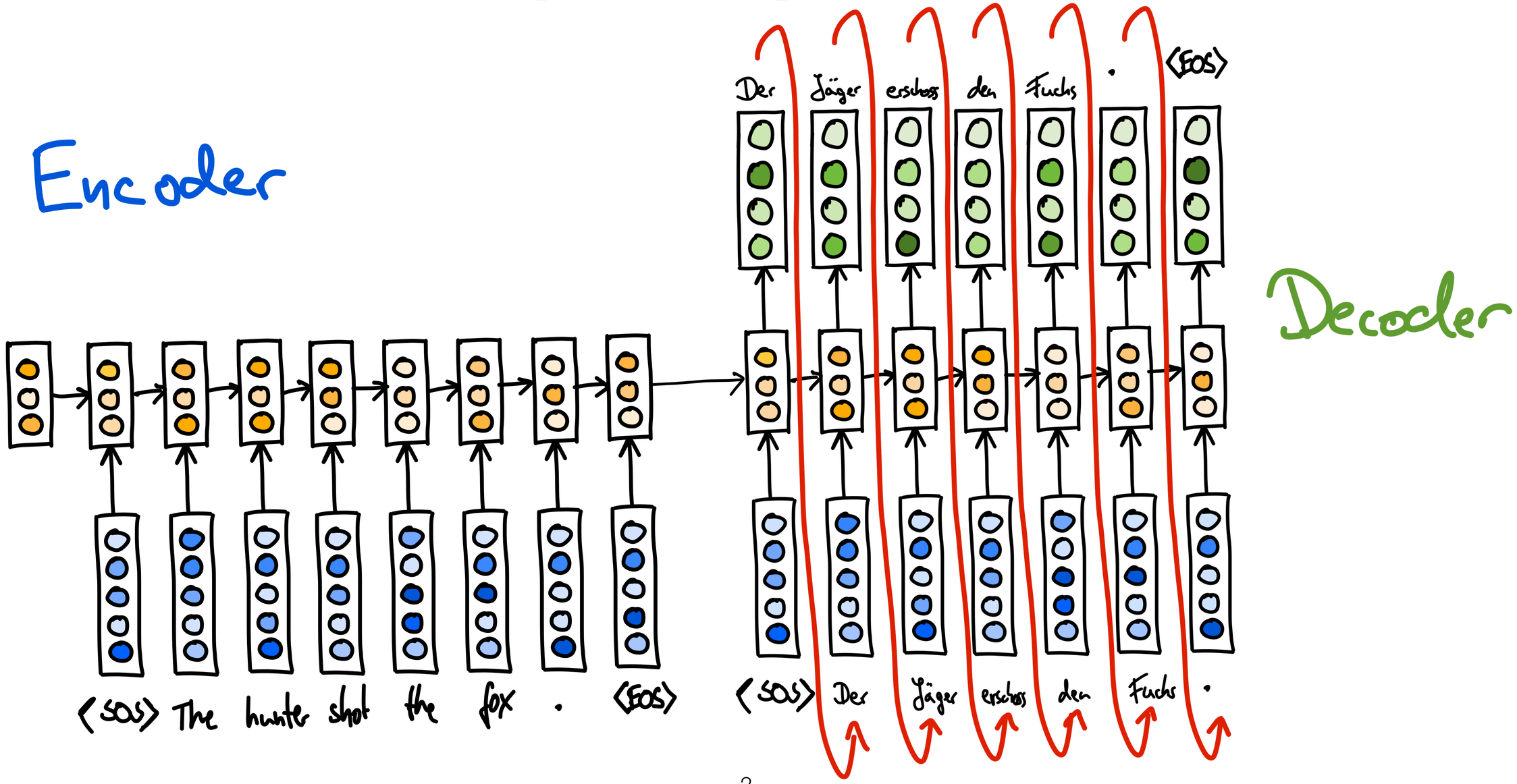
Attention

Tim Rocktäschel & Sebastian Riedel

COMP0087 Natural Language Processing

Seq2Seq Recap



Seq2Seq Recap

Input:

```
j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))
```

Target: 25011.

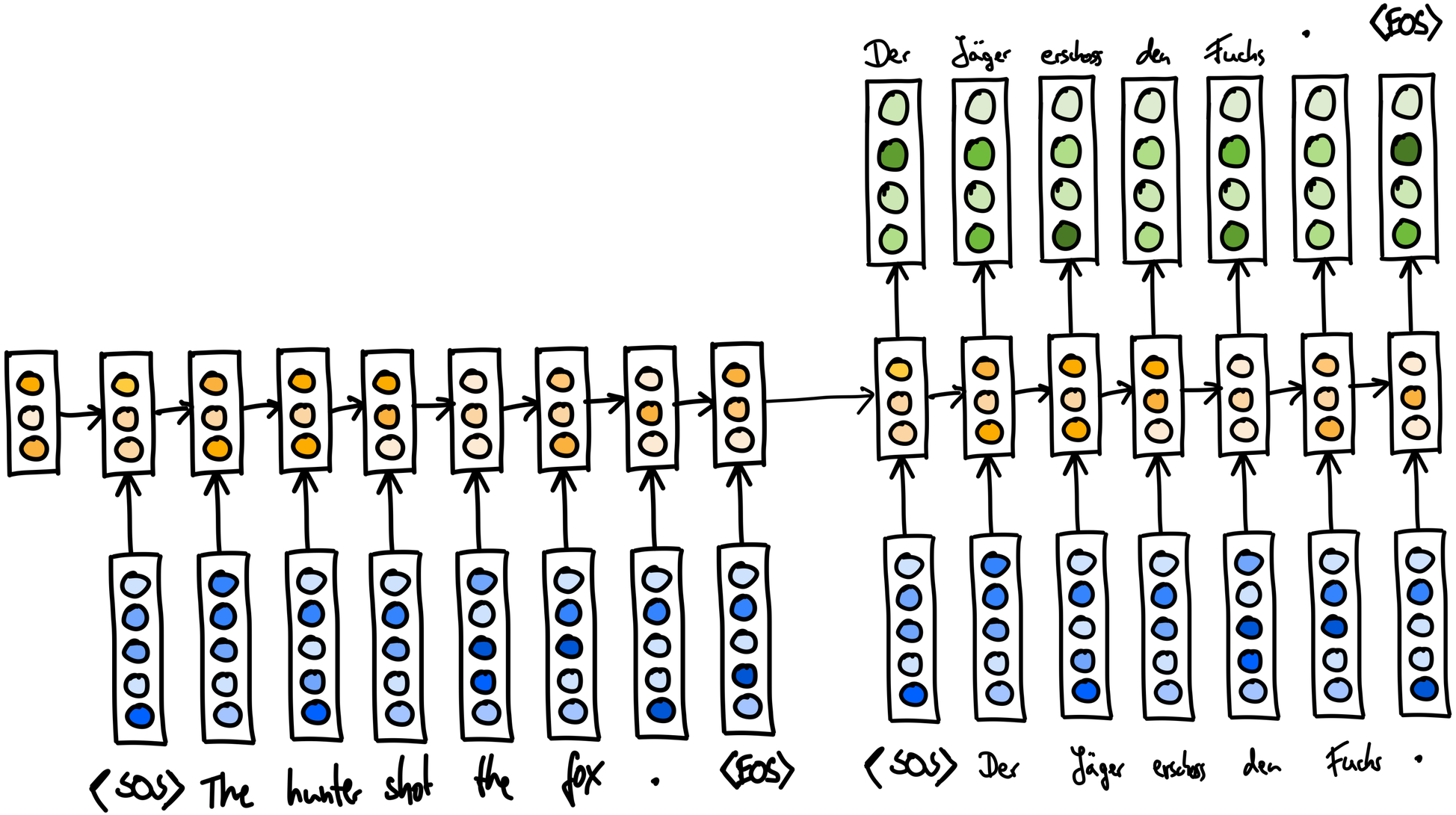
Input:

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else 5308)
```

Target: 12184.

Zaremba et al. Learning to Execute. arXiv 2015.

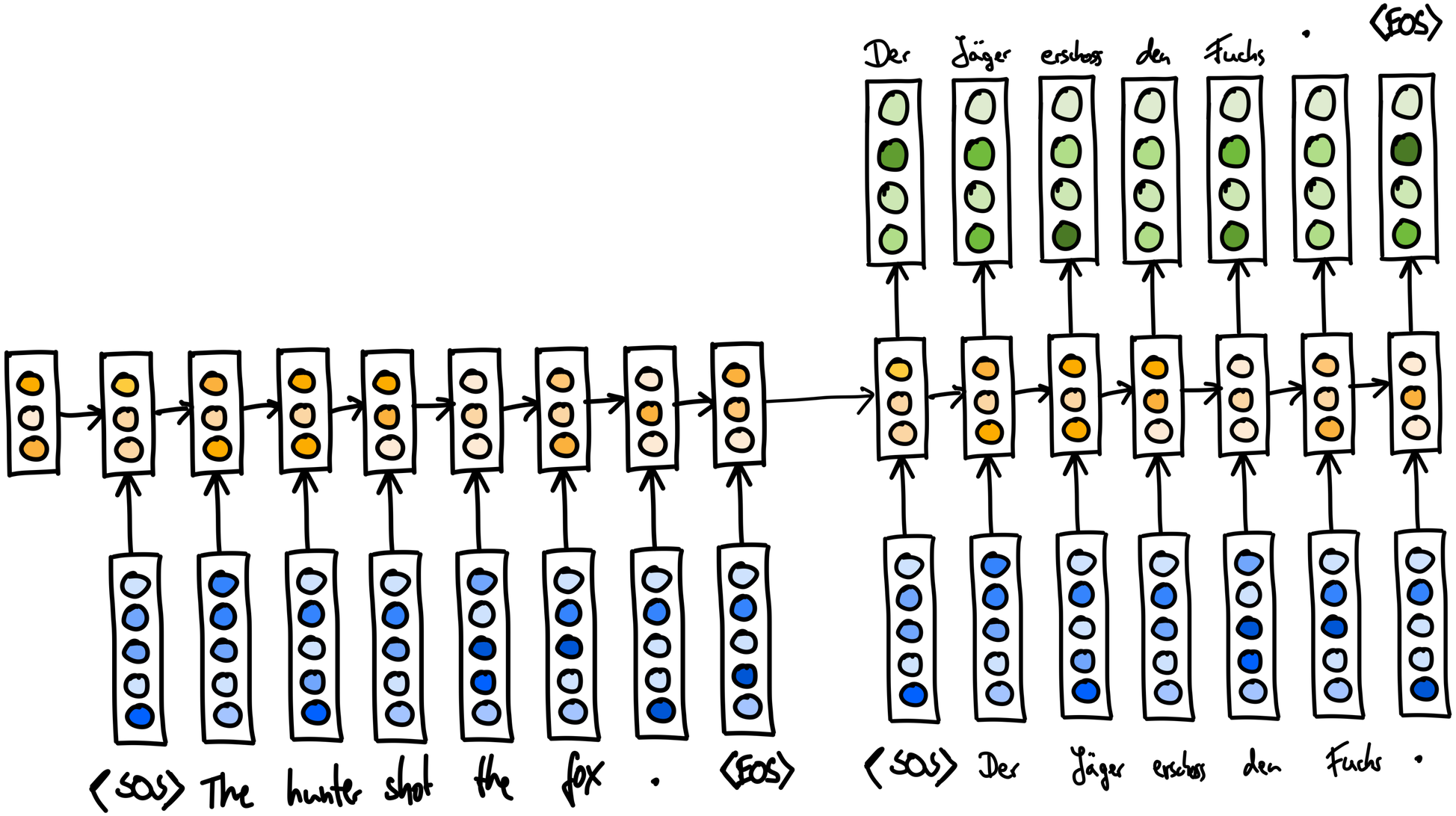
Limits of RNNs



“You can’t cram the meaning of a whole %&!$# sentence into a single $&!#* vector!”

— Raymond J. Mooney

Limits of RNNs

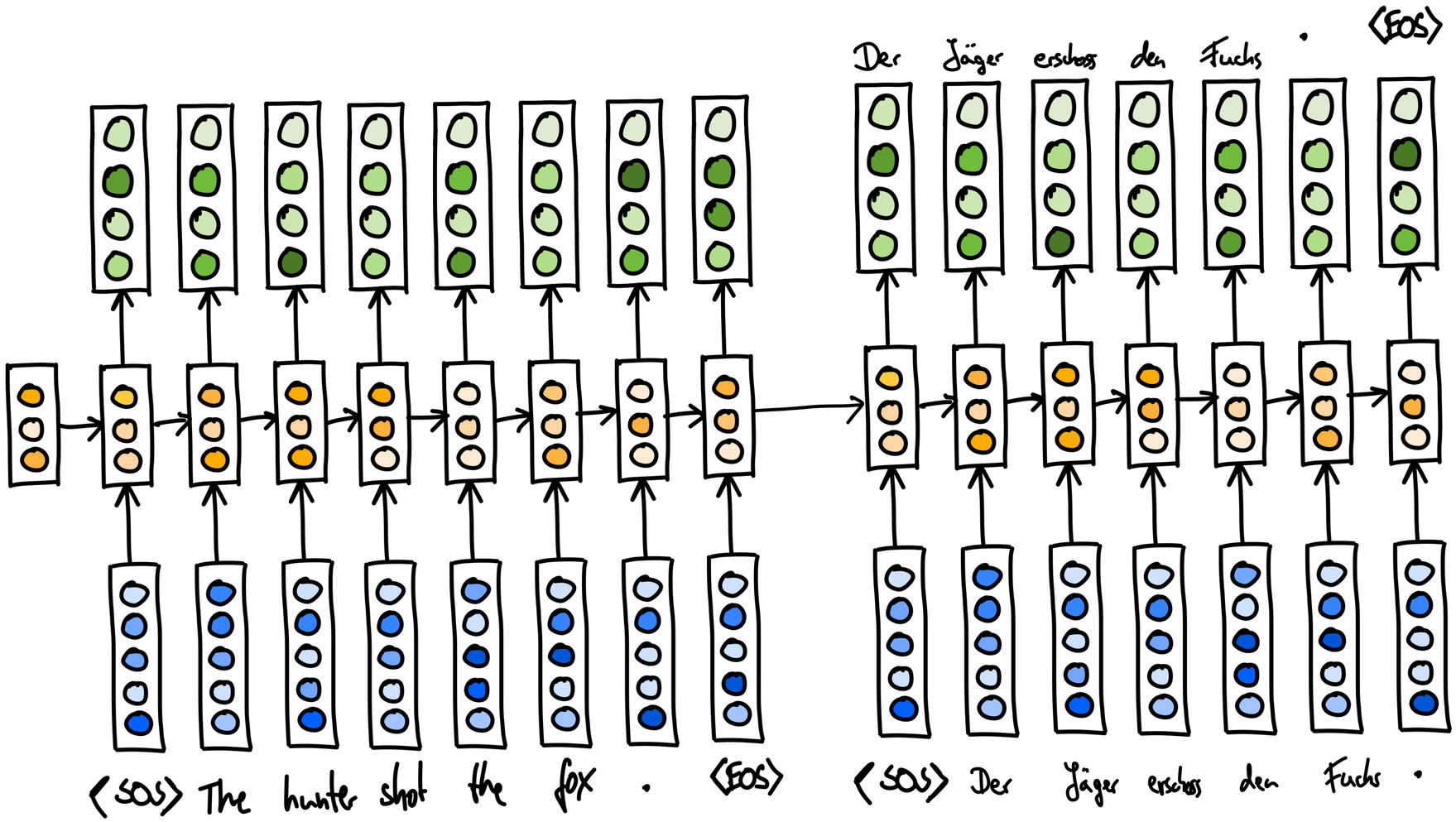


Jointly Learning to Align and Translate

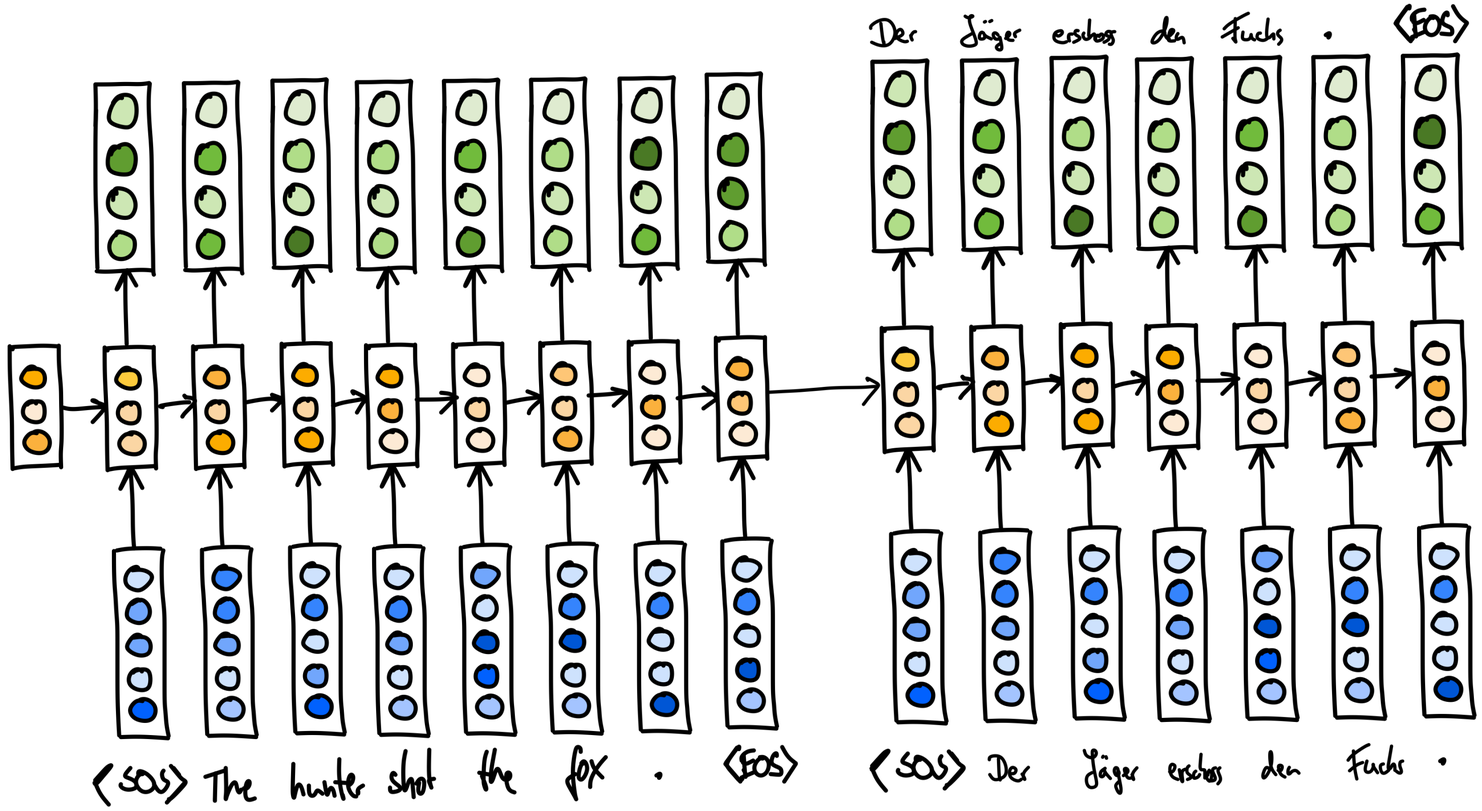
- Keep output vectors during encoding
- At decoding time, learn weighting of encoder outputs
- Use combination of decoder hidden state and linear combination of encoder outputs to decide on the next word

Bahdanau et al. Neural machine translation by jointly learning to align and translate. ICLR 2015.

Sentences as Matrices



Soft Attention



$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$$

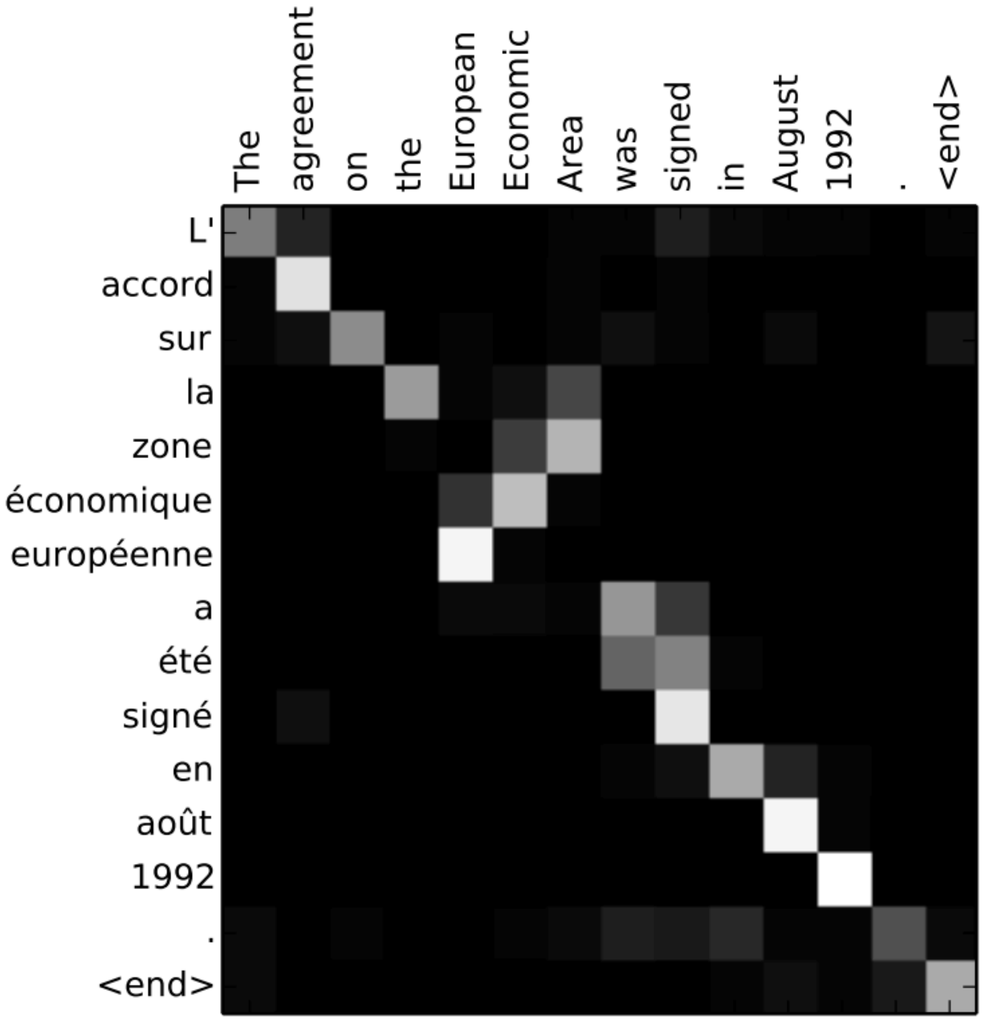
$$m_{ti} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{q}_t; \mathbf{h}_i] + \mathbf{b})$$

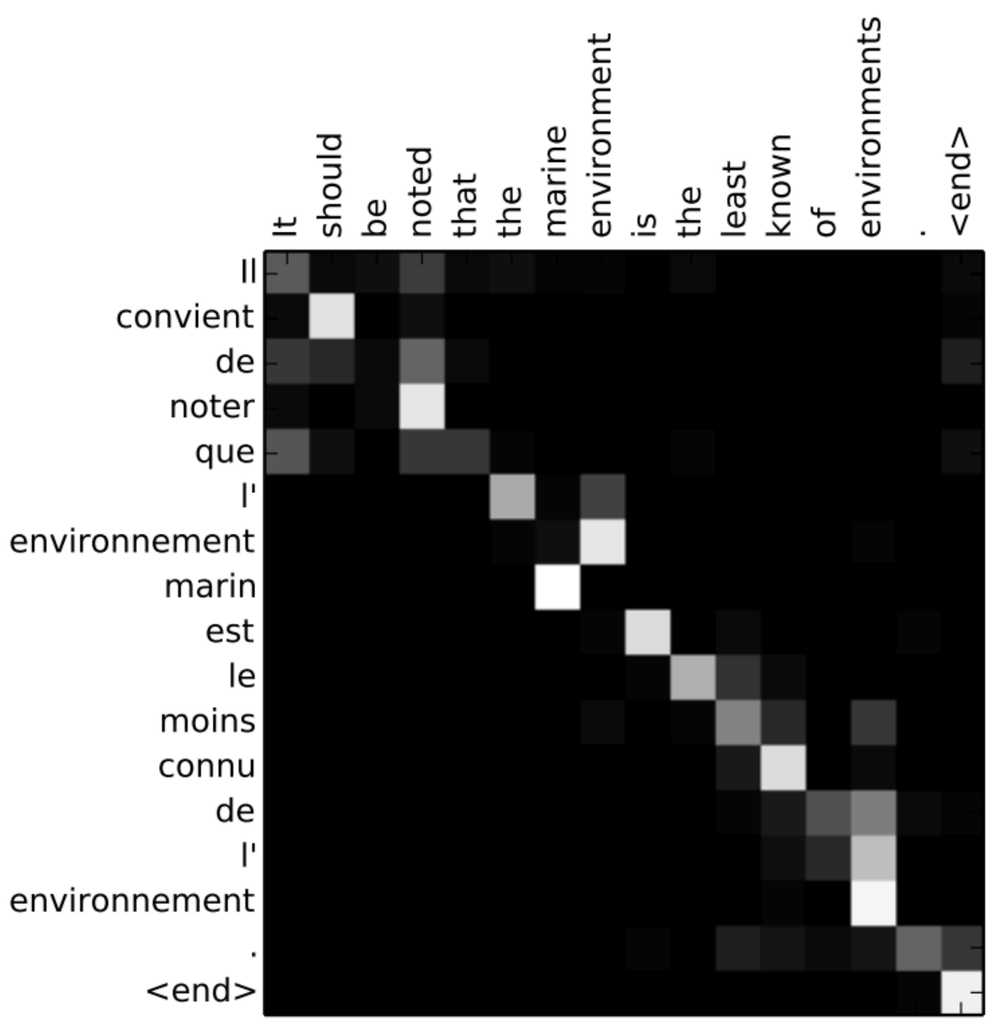
$$\alpha_t = \mathbf{softmax}(\mathbf{m}_{t:})$$

$$\mathbf{q}'_t = \alpha_t^\top \mathbf{H}$$

Bahdanau et al. Neural machine translation by jointly learning to align and translate. ICLR 2015.

Attention Maps





Bahdanau et al. Neural machine translation by jointly learning to align and translate. ICLR 2015.


```
import torch
from einops import rearrange, reduce

x = torch.randn(3, 7, 27)
y = rearrange(x, "batch time emb -> emb () (batch time)",)
y.shape
> torch.Size([27, 1, 21])

x = torch.randn(3, 7, 8)
y1, y2 = rearrange(x, "batch time (split emb) -> split (batch time) emb", split=2)
print(y1.shape)
> torch.Size([21, 4])

x = torch.randn(3, 7, 27)
y = reduce(x, "batch time (emb pool) -> batch time emb", "max", pool=3)
print(y.shape)
> torch.Size([3, 7, 9])
```


https://github.com/arogozhnikov/einops

Implementation Sketch

$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$$

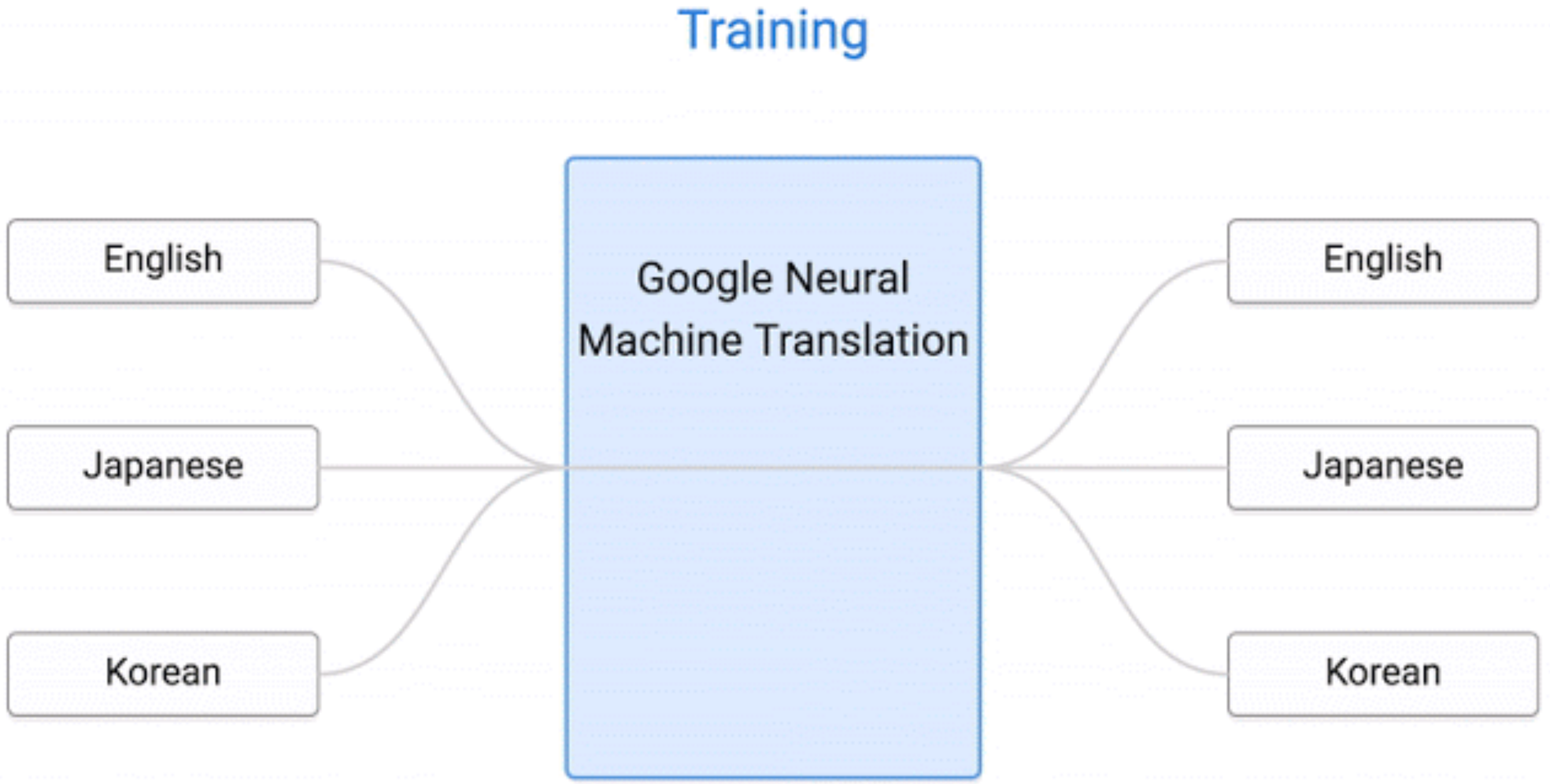
$$m_{ti} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{q}_t; \mathbf{h}_i] + \mathbf{b})$$

$$\alpha_t = \textbf{softmax}(\mathbf{m}_{t:})$$

$$\mathbf{q}'_t = \alpha_t^\top \mathbf{H}$$

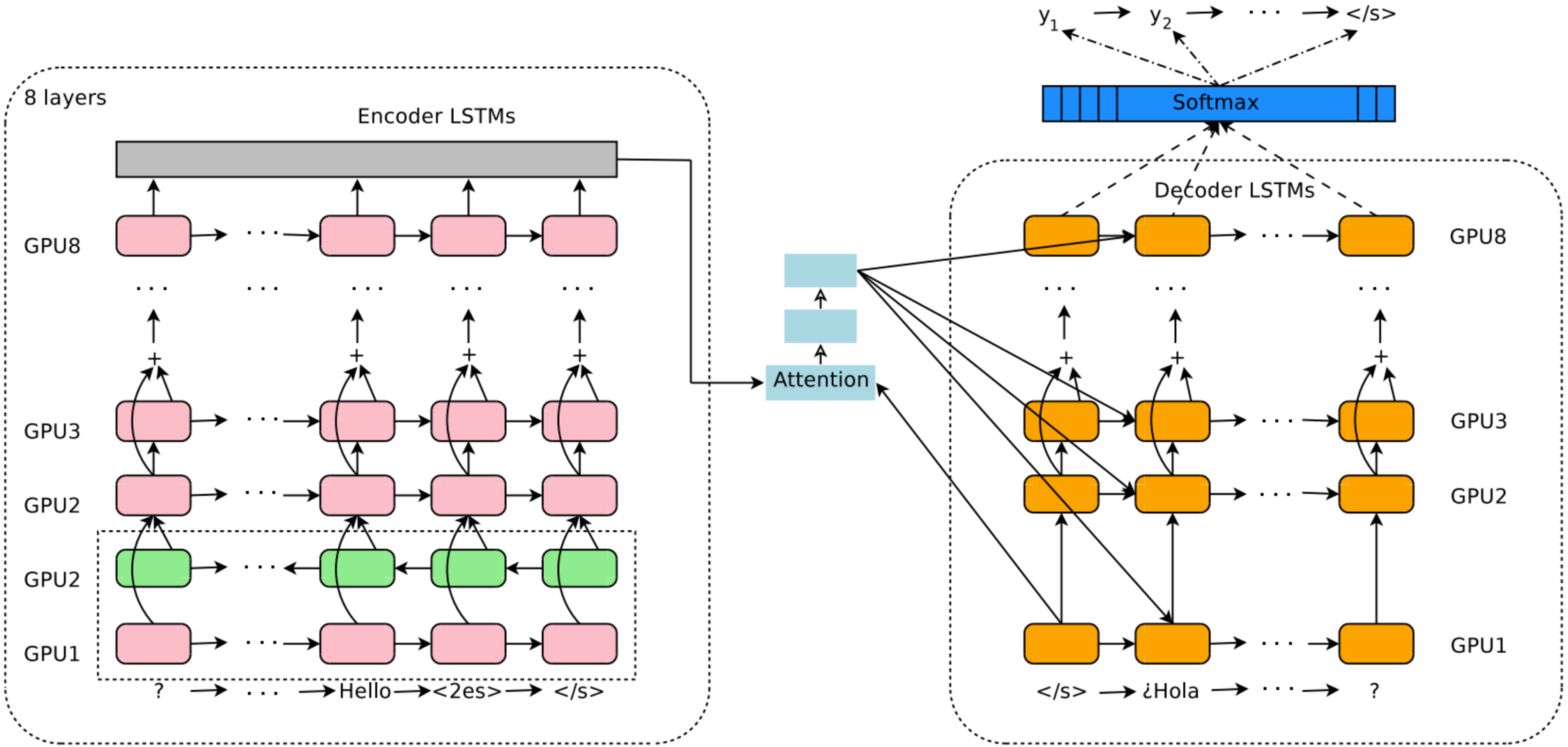
```
import torch
from einops import rearrange, reduce

def attention(H, qt, W, w, b):
  qt = rearrange(qt, "b k -> b () k").expand_as(H)
  u1 = rearrange([H, qt], "x b n k -> b n (x k)")
  u2 = torch.tanh(torch.einsum("bnt,kt->bnk", [u1, W]) + b)
  alpha = torch.softmax(torch.einsum("bnk,k->bn", [u2, w]), 1)
  qt1 = torch.einsum("bn,bnk->bk", [alpha, H])
  return qt1, alpha
```

Zero-shot Machine Translation



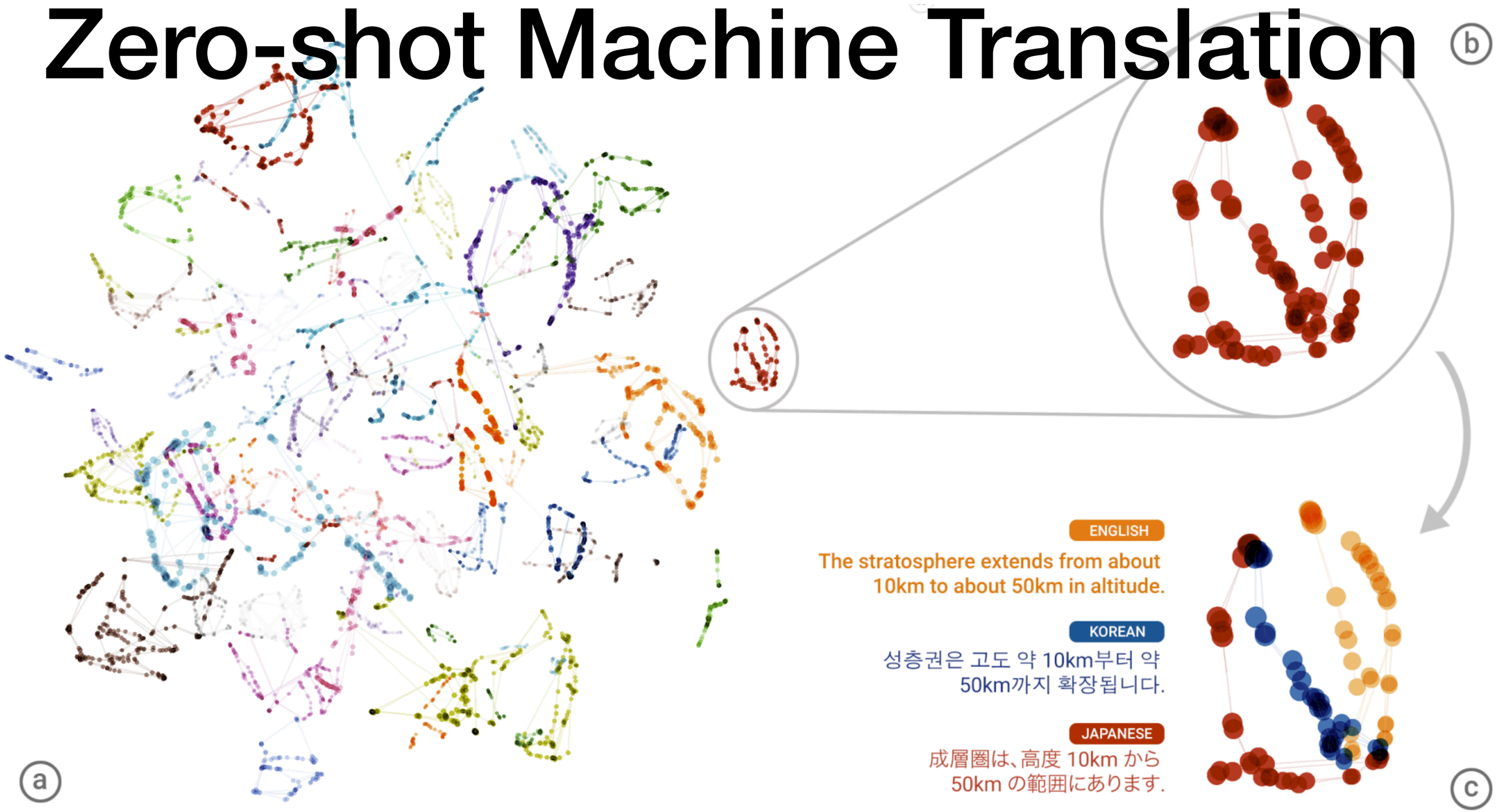
Johnson et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. TACL 2017.

Zero-shot Machine Translation



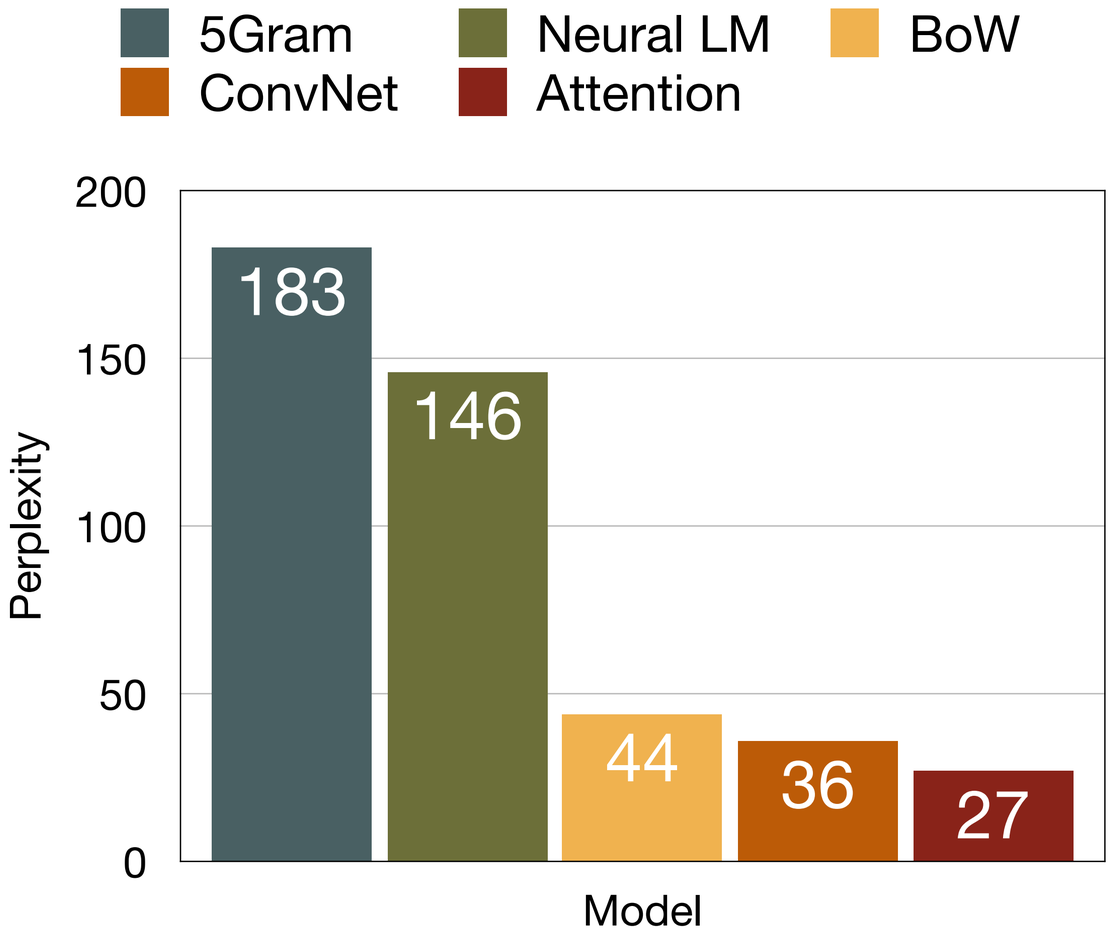
Johnson et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. TACL 2017.

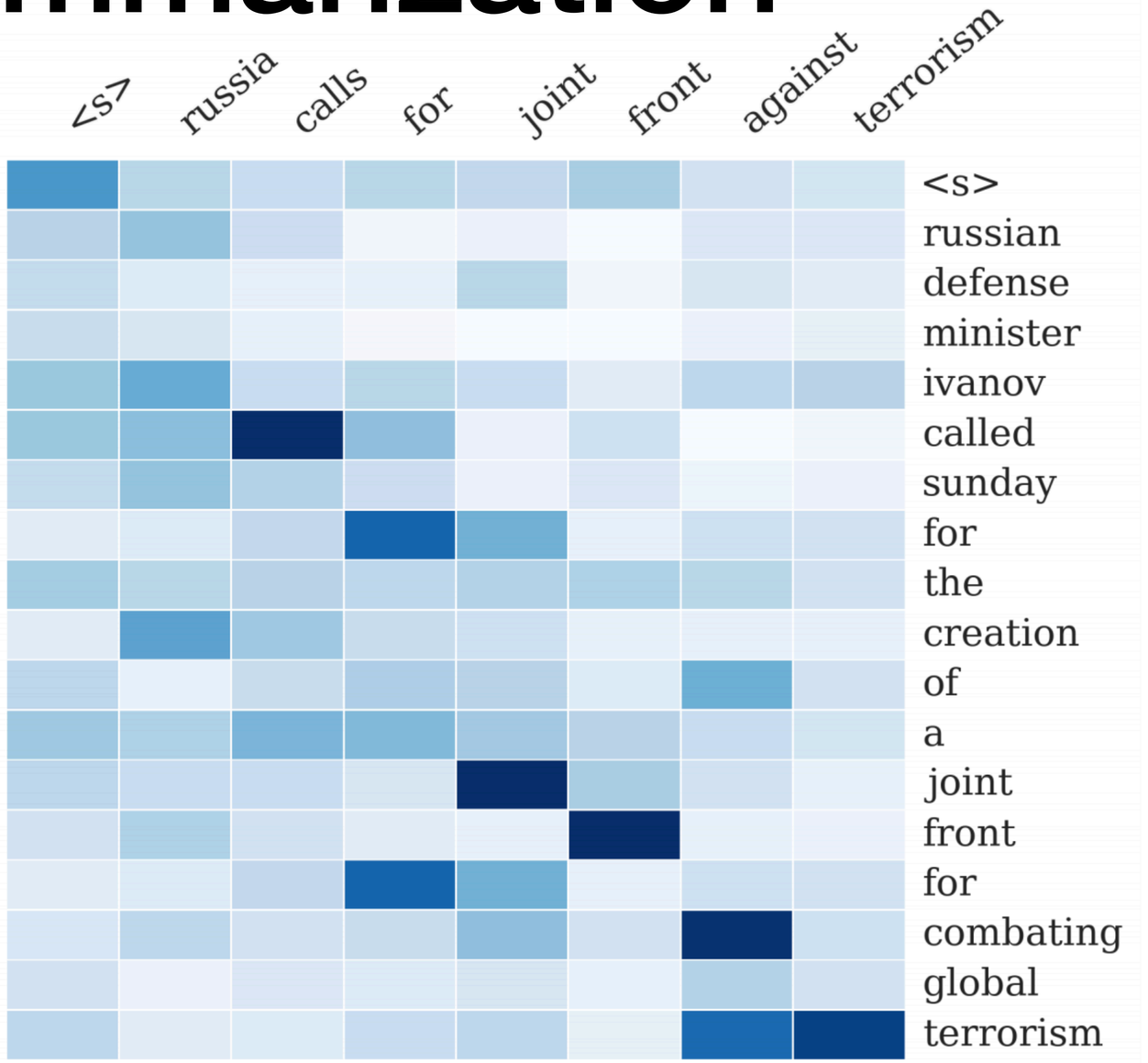
Zero-shot Machine Translation



Johnson et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. TACL 2017.

Sentence Summarization





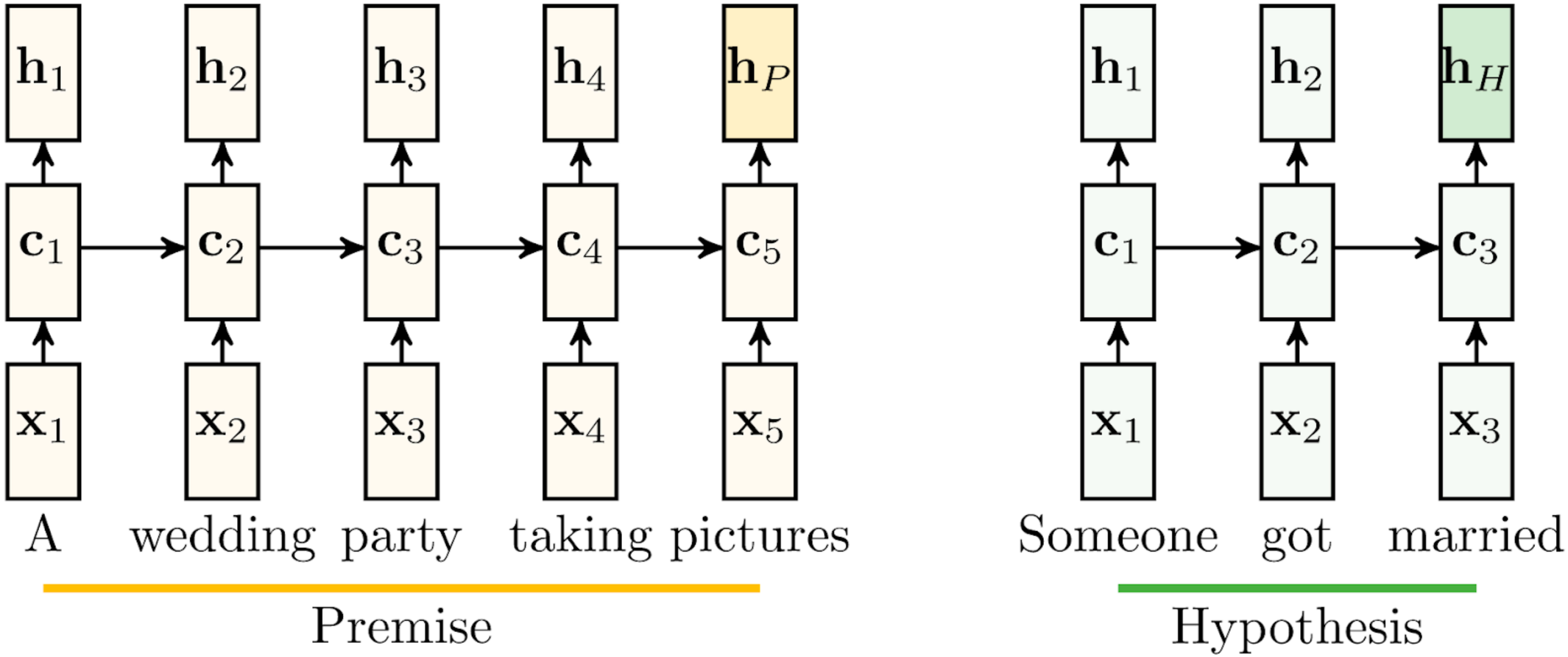
Rush et al. A neural attention model for abstractive sentence summarization. EMNLP 2015.

Recognizing Textual Entailment

A wedding party is taking pictures

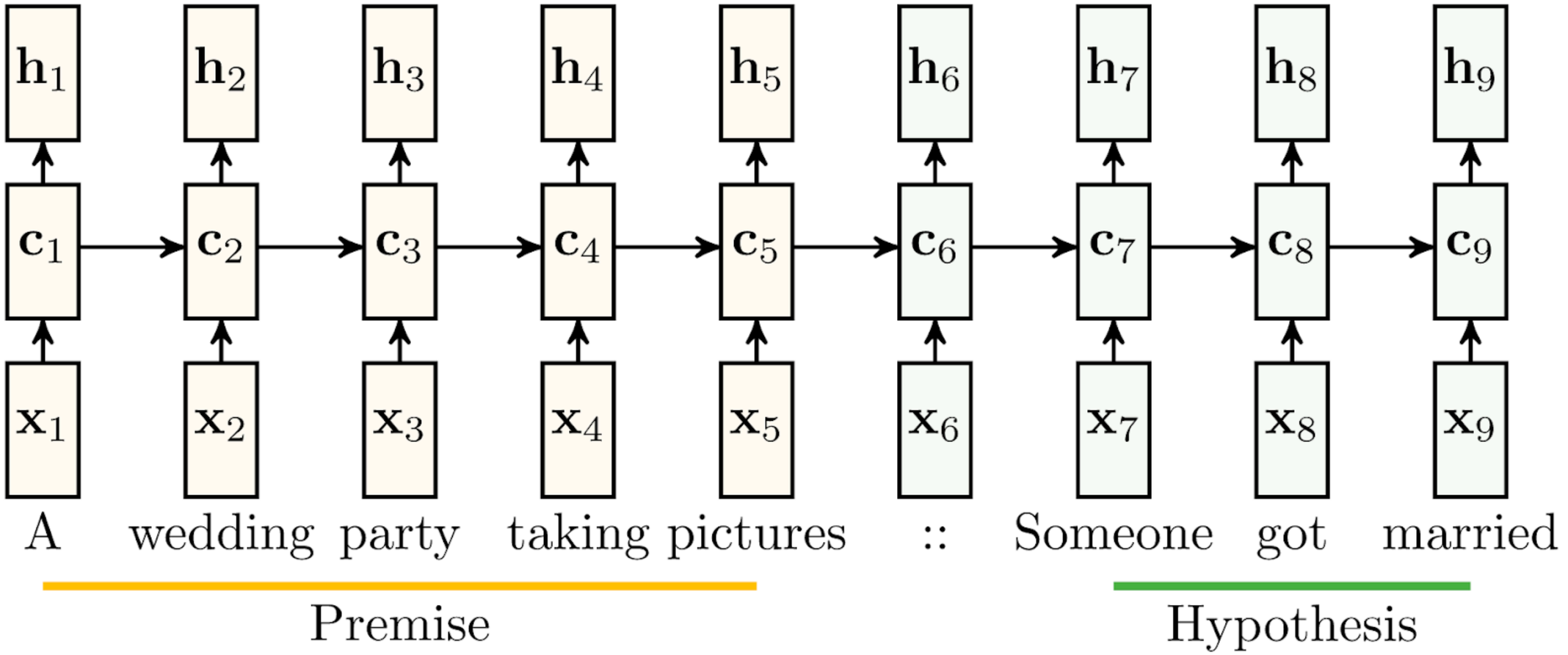
- There is a funeral — **Contradiction**
- They are outside — **Neutral**
- Someone got married — **Entailment**

Independent Sentence Encoding

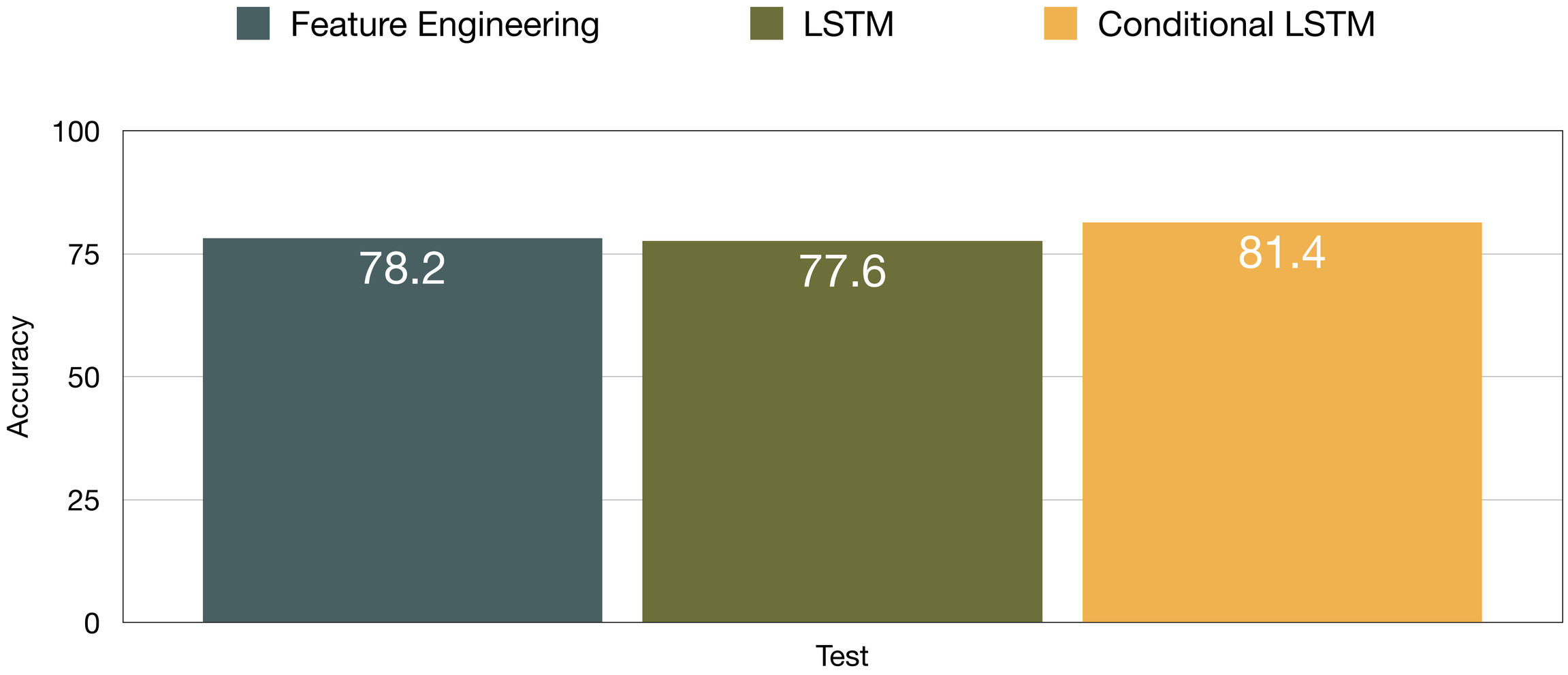


Bowman et al. A large annotated corpus for learning natural language inference. EMNLP 2015.

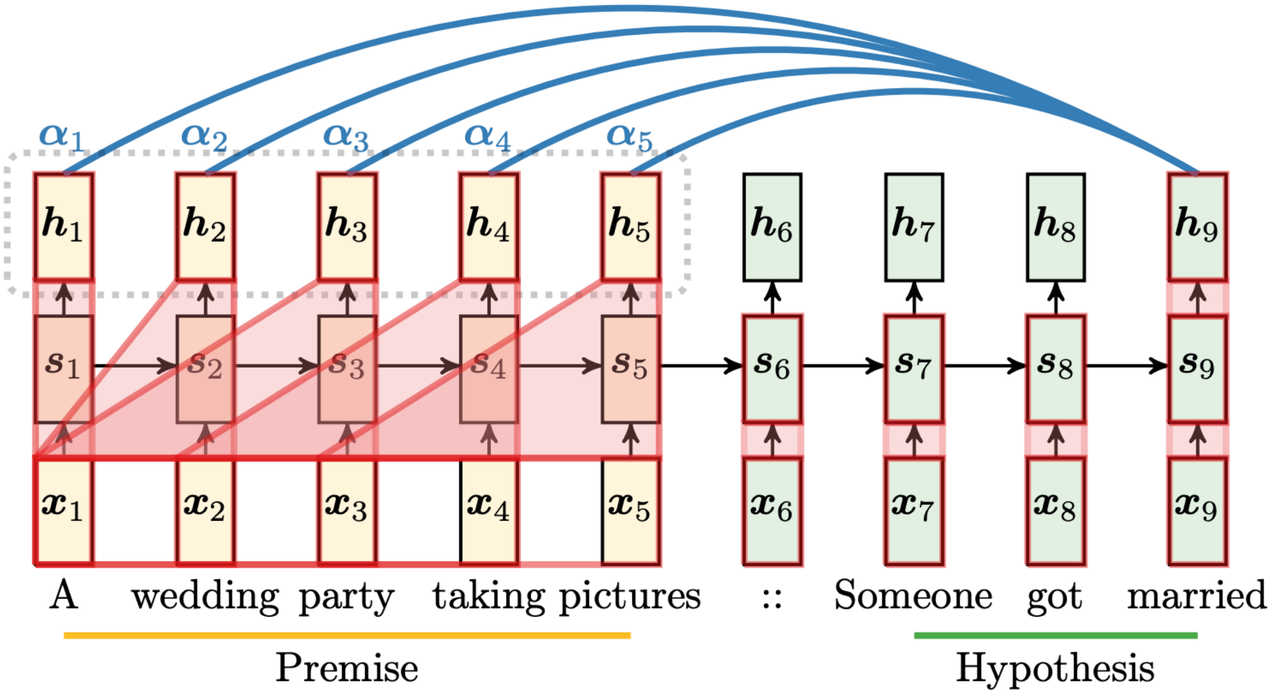
Conditional Encoding



Results


Feature Engineering
LSTM
Conditional LSTM
100
75
50
25
0
Accuracy
78.2
77.6
81.4
Test

Attention



$$\mathbf{P} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$$

$$\mathbf{H} = [\mathbf{h}_{N+1}, \ldots, \mathbf{h}_M]$$

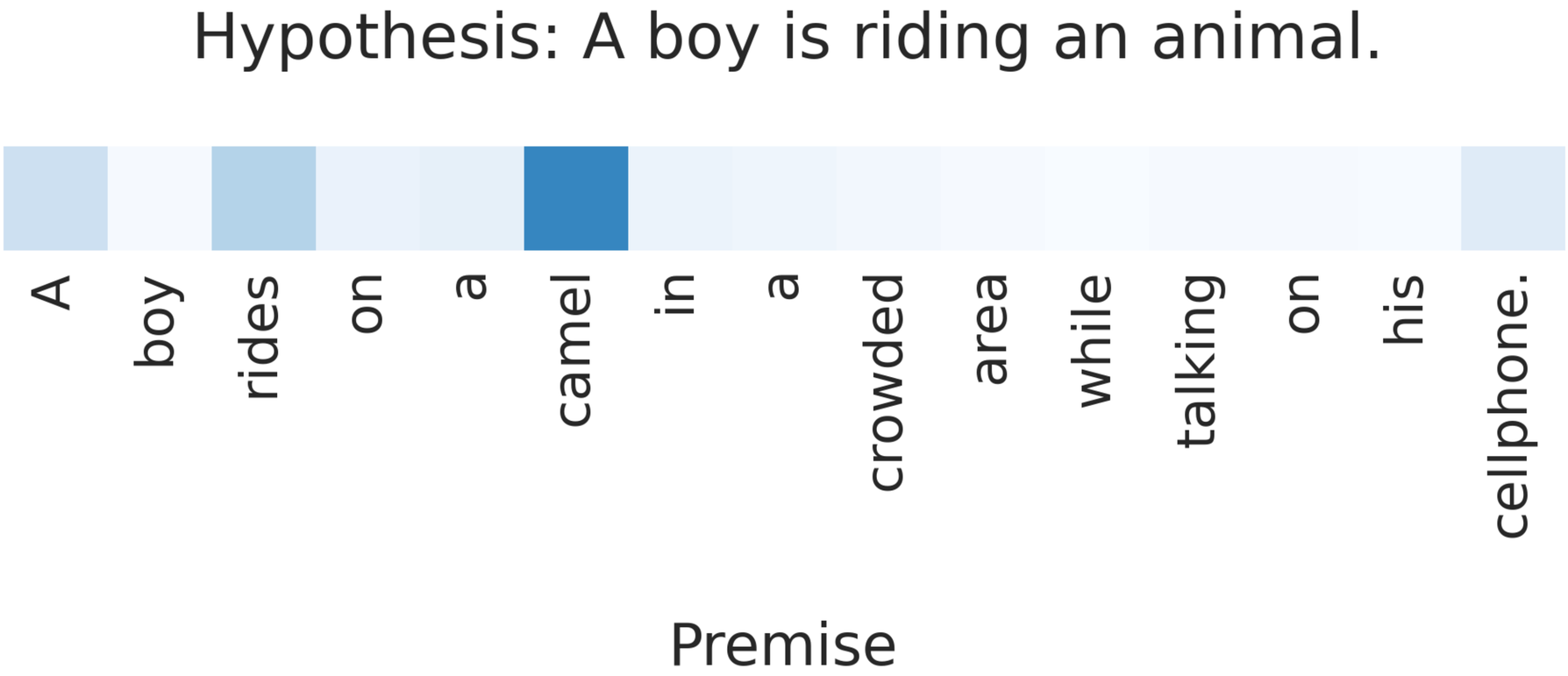
$$m_i = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{h}_M; \mathbf{p}_i] + \mathbf{b})$$

$$\alpha = \mathbf{softmax}(\mathbf{m}_\cdot)$$

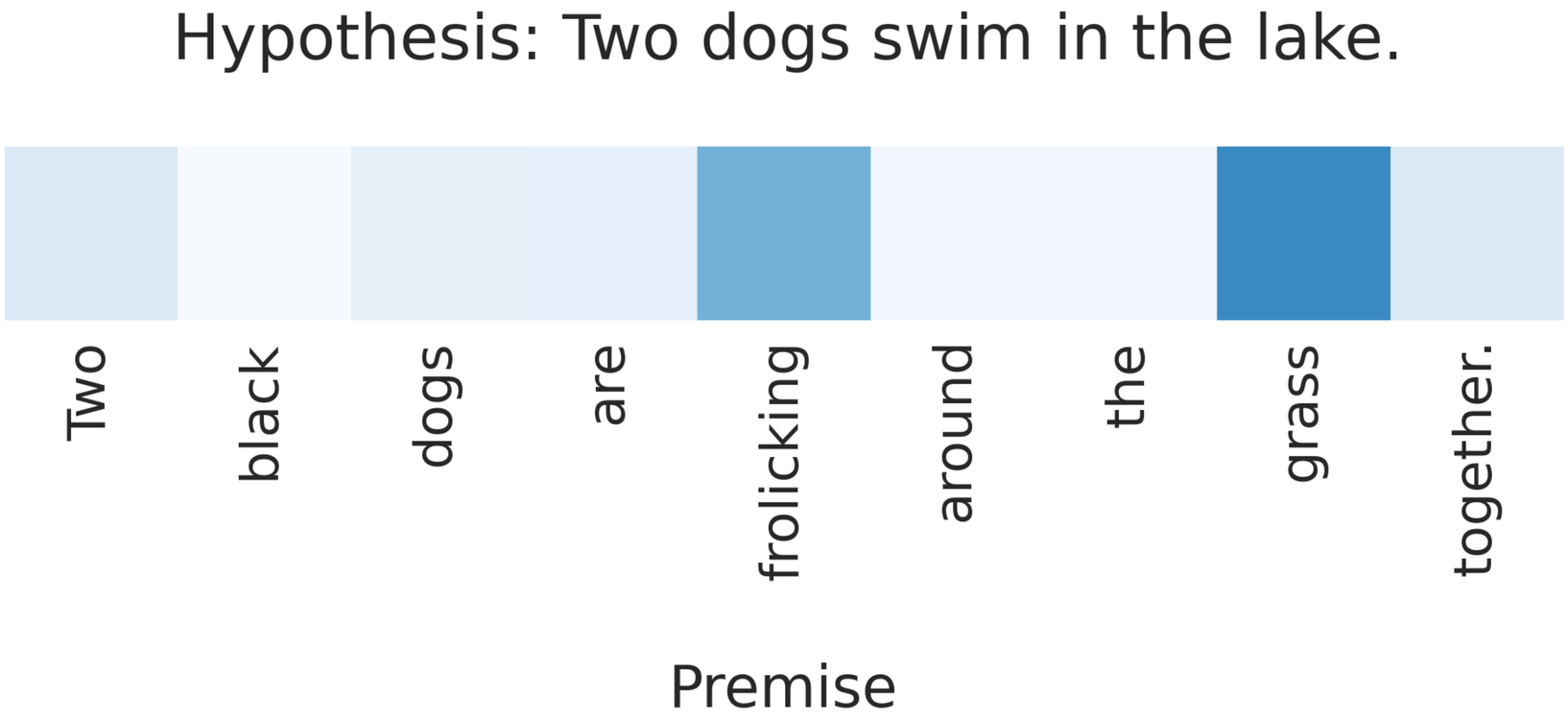
$$\mathbf{p}' = \alpha^\top \mathbf{P}$$

$$\mathbf{h}'_M = \tanh(\mathbf{W}^p \mathbf{p}' + \mathbf{W}^h \mathbf{h}_M)$$

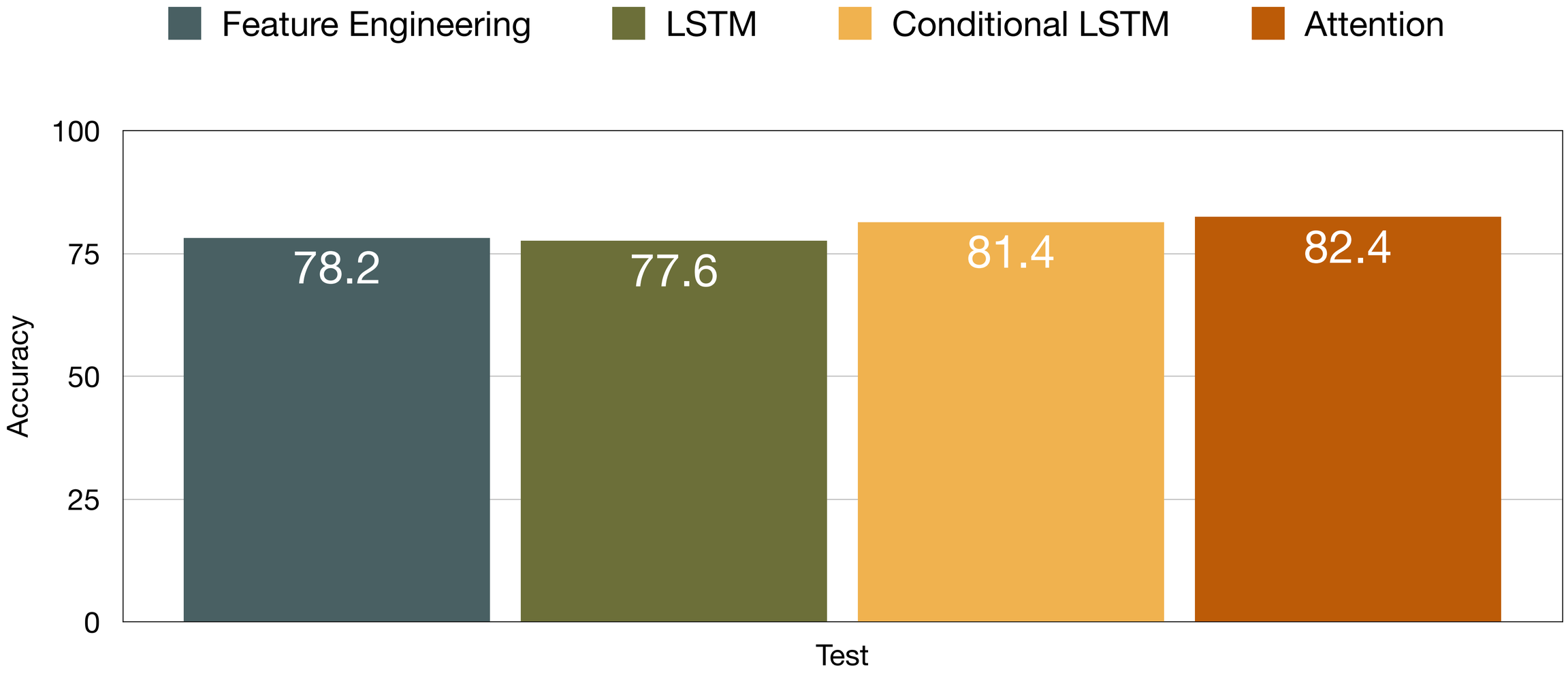
Attention Maps



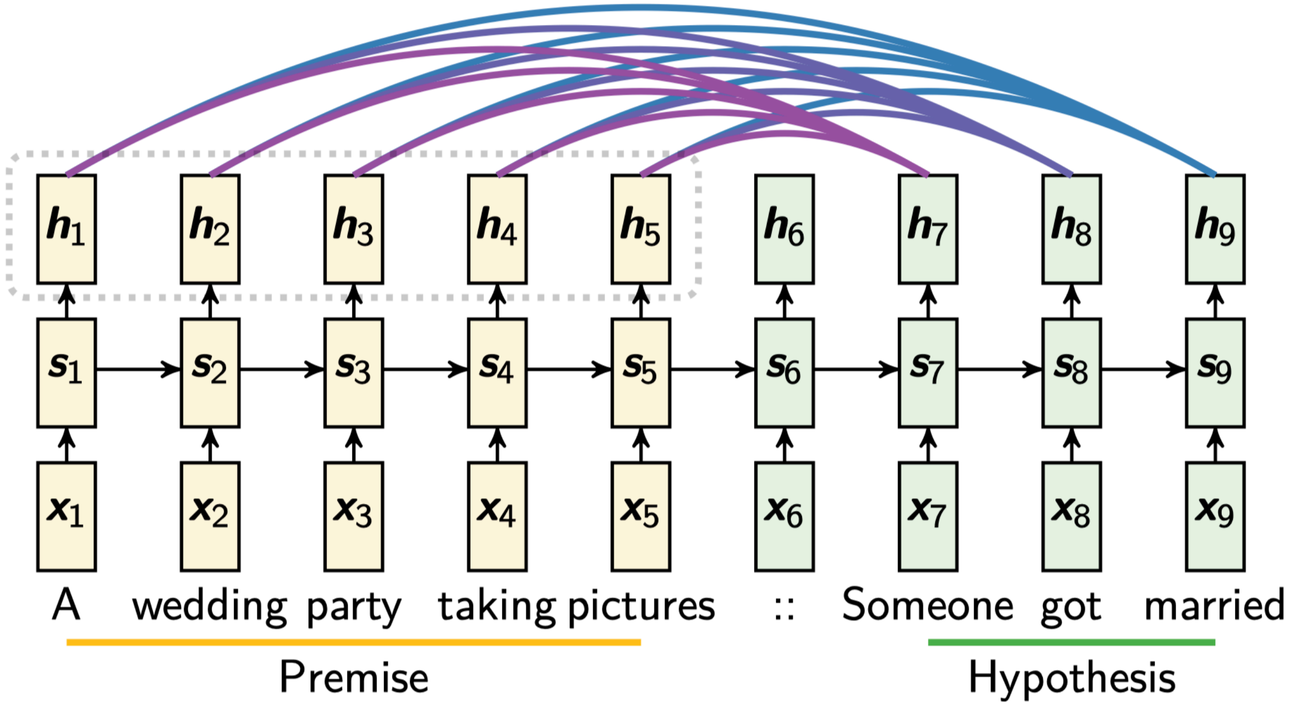
Attention Maps



Results


Feature Engineering
LSTM
Conditional LSTM
Attention
100
75
50
25
0
Accuracy
78.2
77.6
81.4
82.4
Test

Word-by-word Attention



$$\mathbf{P} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$$

$$\mathbf{H} = [\mathbf{h}_{N+1}, \ldots, \mathbf{h}_M]$$

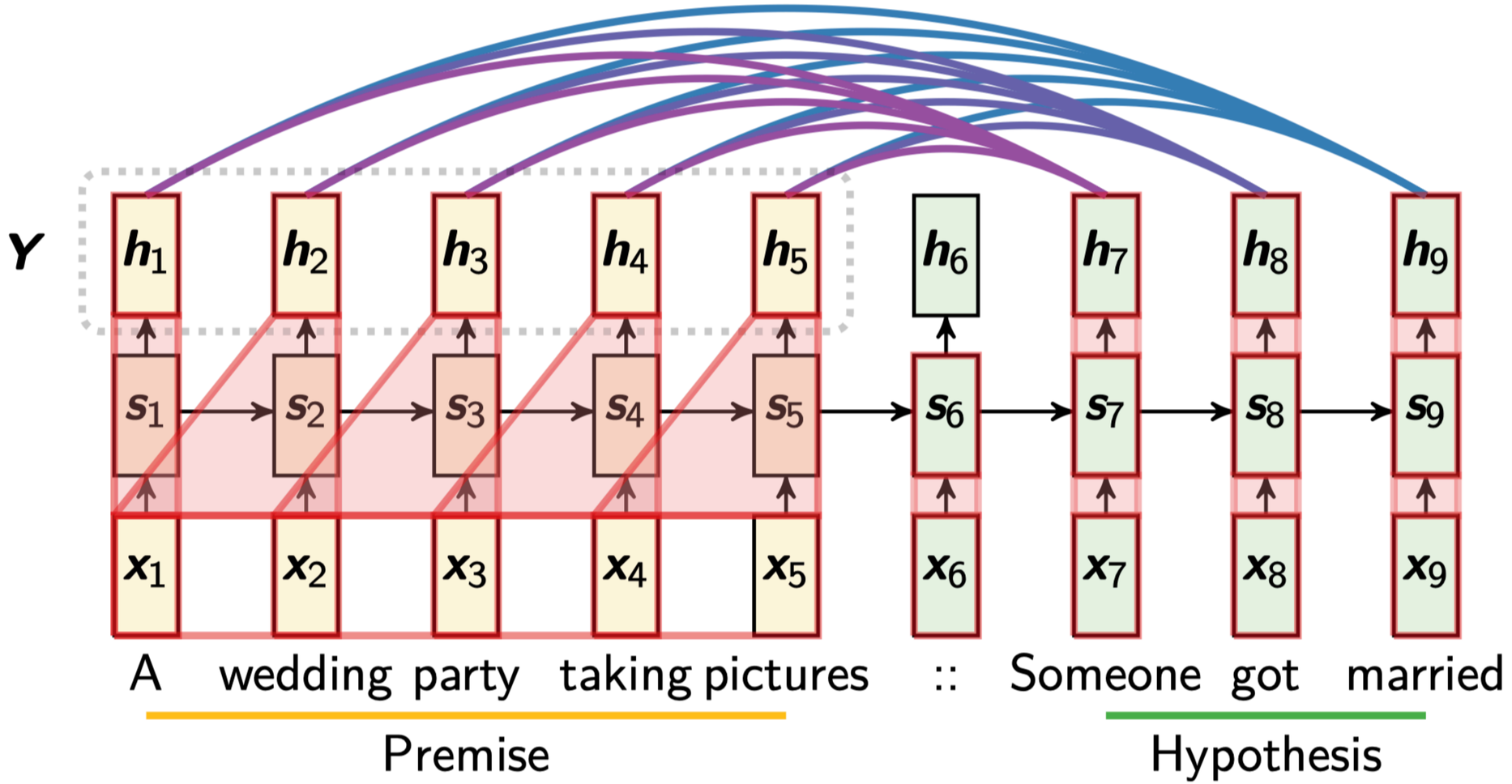
$$m_{ti} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{h}_t; \mathbf{p}_i] + \mathbf{b})$$

$$\alpha_t = \textbf{softmax}(\mathbf{m}_{t:})$$

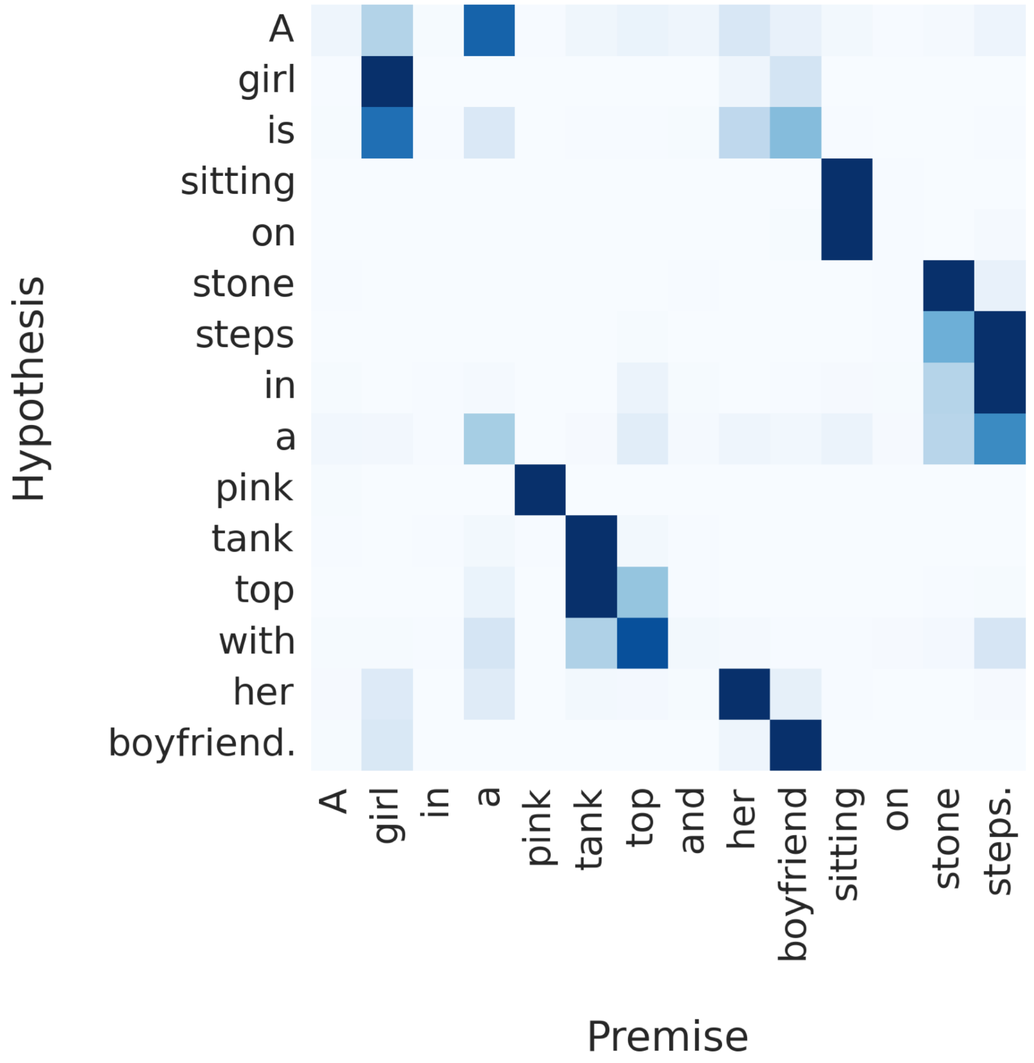
$$\mathbf{p}'_t = \alpha_t^\top \mathbf{P} + \tanh(\mathbf{W}^a \mathbf{p}'_{t-1})$$

$$\mathbf{h}'_t = \tanh(\mathbf{W}^p \mathbf{p}'_t + \mathbf{W}^h \mathbf{h}_t)$$

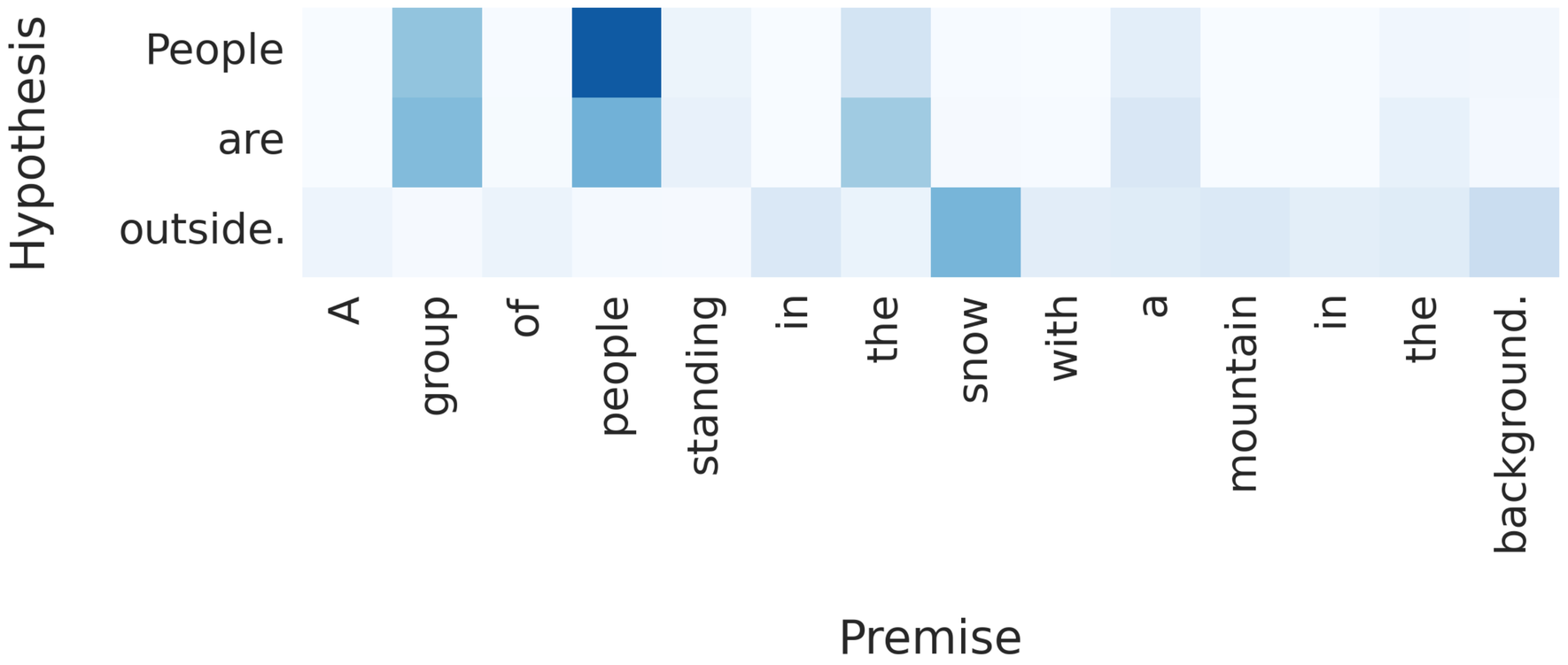
Word-by-word Attention



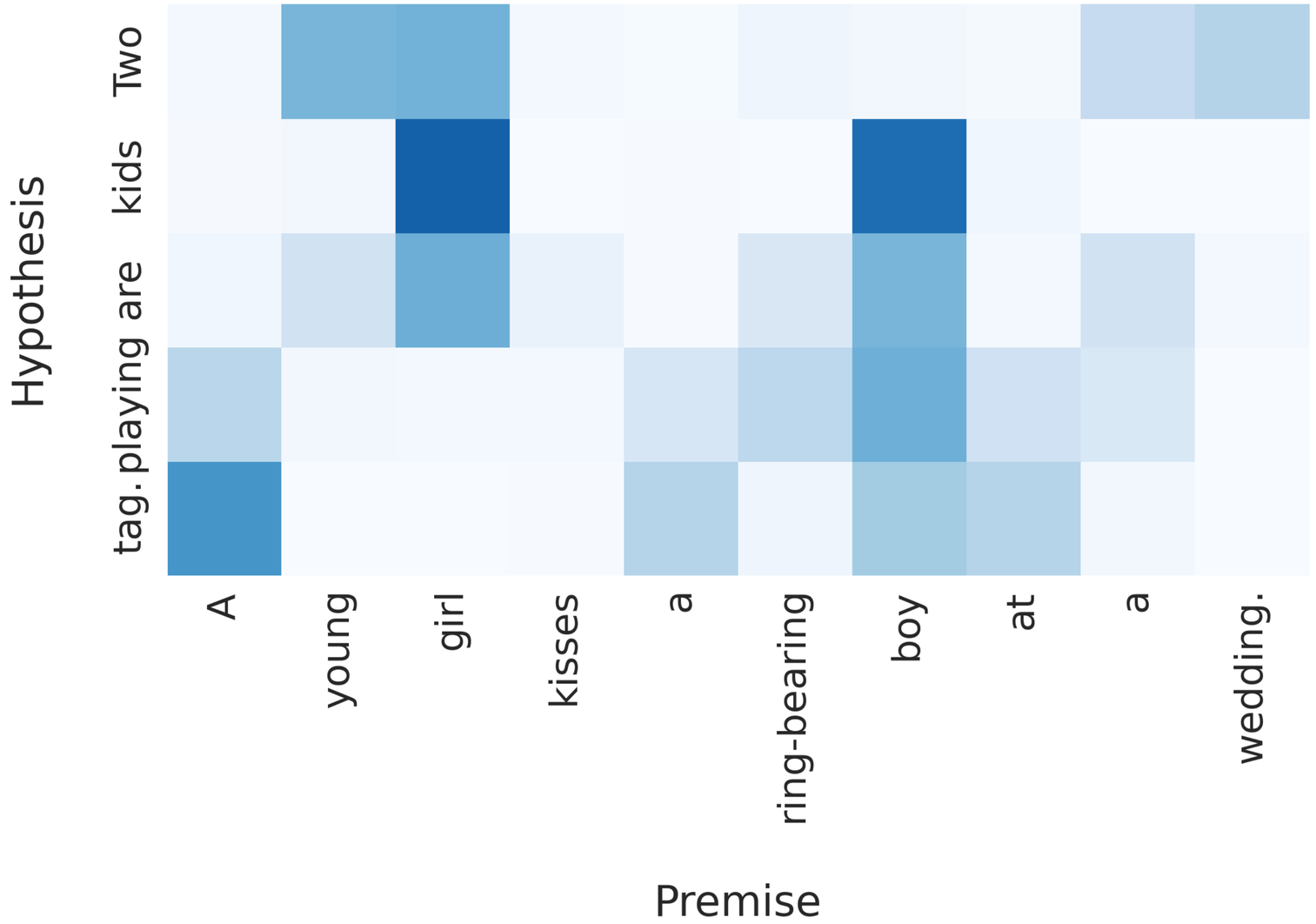
Attention Maps



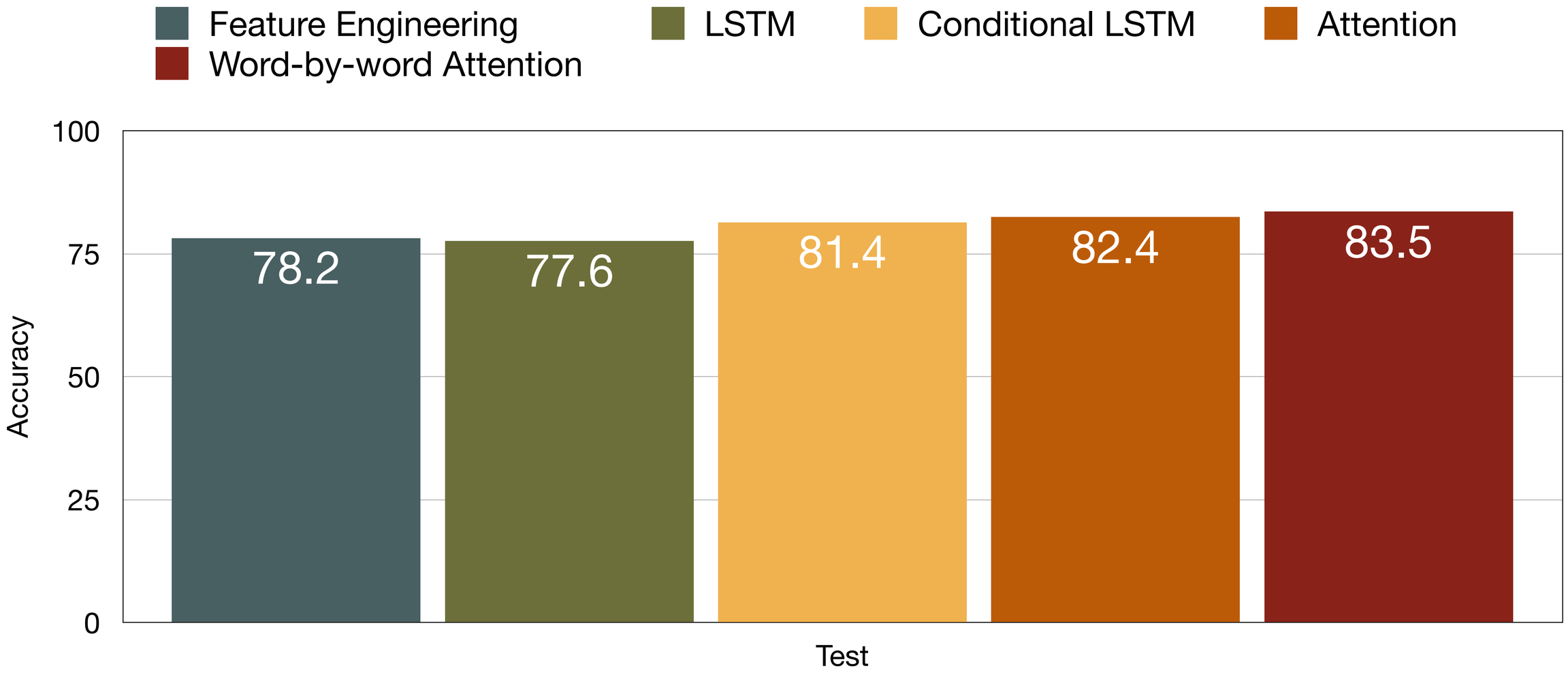
Attention Maps



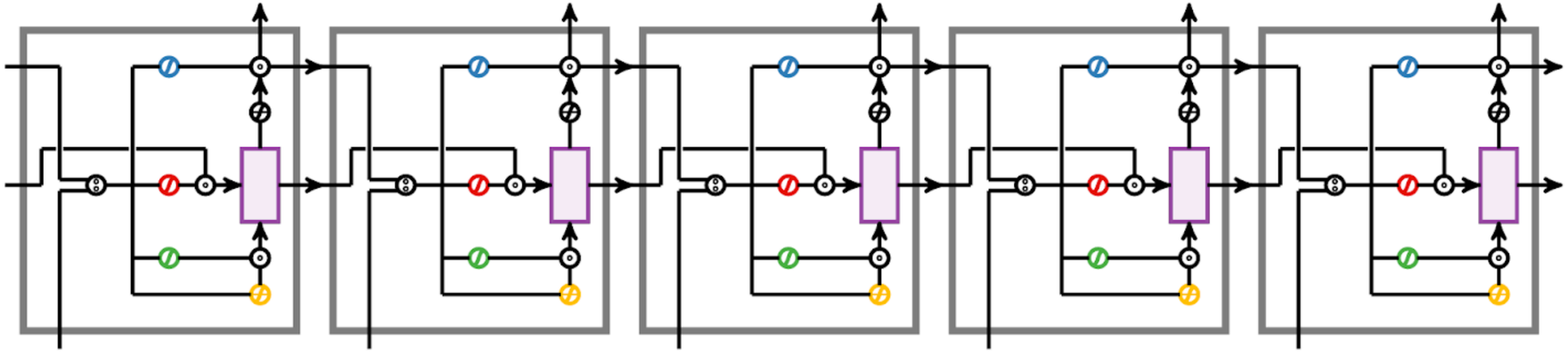
Attention Maps



Results


Feature Engineering
LSTM
Conditional LSTM
Attention
Word-by-word Attention
100
75
50
25
0
Accuracy
78.2
77.6
81.4
82.4
83.5
Test

Long Short-Term Memory-Networks

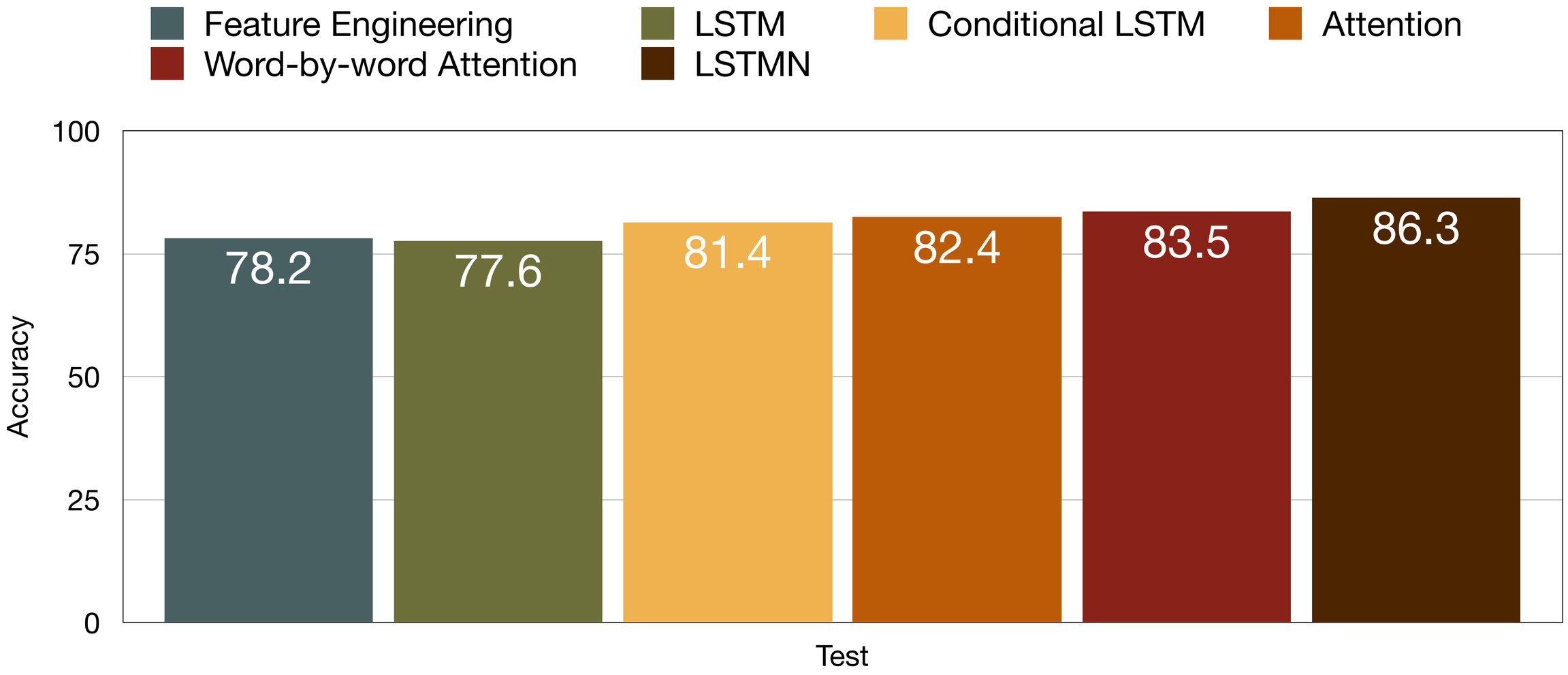


$$\mathbf{C}_t = [\mathbf{c}_1, \ldots, \mathbf{c}_{t-1}]$$
$$\mathbf{H}_t = [\mathbf{h}_1, \ldots, \mathbf{h}_{t-1}]$$
$$m_{ti} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{q}_t; \mathbf{h}_t; \mathbf{h}_i] + \mathbf{b})$$
$$\alpha_t = \mathbf{softmax}(\mathbf{m}_{t:})$$
$$\mathbf{h}'_t = \alpha_t^\top \mathbf{H}_t$$
$$\mathbf{z}_t = [\mathbf{x}_t; \mathbf{h}'_t]$$
$$\vdots$$
$$\mathbf{c}'_t = \mathbf{f}_t \odot (\alpha_t^\top \mathbf{C}_t) + \mathbf{i}_t \odot (\alpha_t^\top \mathbf{C}_t)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}'_t)$$

The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Cheng et al. Long Short-Term Memory-Networks for Machine Reading. EMNLP 2016.

Results


Feature Engineering
LSTM
Conditional LSTM
Attention
Word-by-word Attention
LSTMN
100
75
50
25
0
Accuracy
78.2
77.6
81.4
82.4
83.5
86.3
Test

Attention Only: Do we need RNNs at all?

- Premise: Bob is in his room, but because of the thunder and lightning outside, he cannot sleep.
- Hypothesis 1: Bob is awake.
- Hypothesis 2: It is sunny outside.

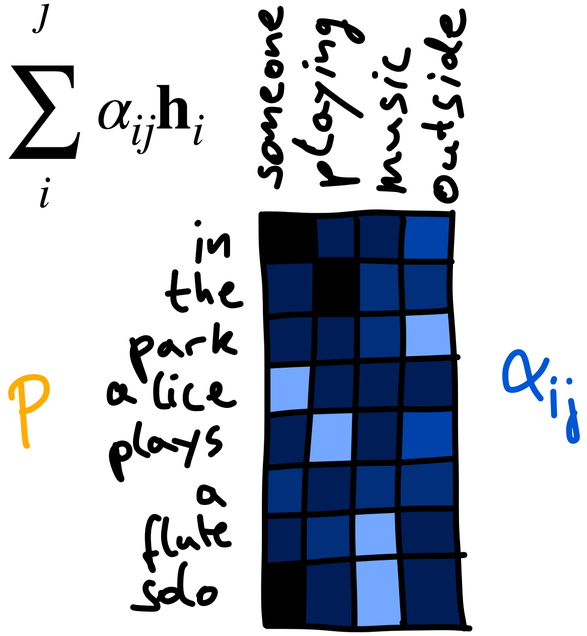
⇒ Problem could be decomposed into aligning subphrases

Parikh et al. A Decomposable Attention Model for Natural Language Inference. EMNLP 2016.

Attention Only: Do we need RNNs at all?

$$\mathbf{h}'_i = \sum_j \alpha_{ij} \mathbf{p}_j$$

$$\mathbf{p}'_j = \sum_i \alpha_{ij} \mathbf{h}_i$$



Attend

$$\mathbf{v}_i^h = \mathbf{mlp}_{\theta_1}([\mathbf{h}_i, \mathbf{h}'_i])$$

$$\mathbf{v}_j^p = \mathbf{mlp}_{\theta_1}([\mathbf{p}_j, \mathbf{p}'_j])$$

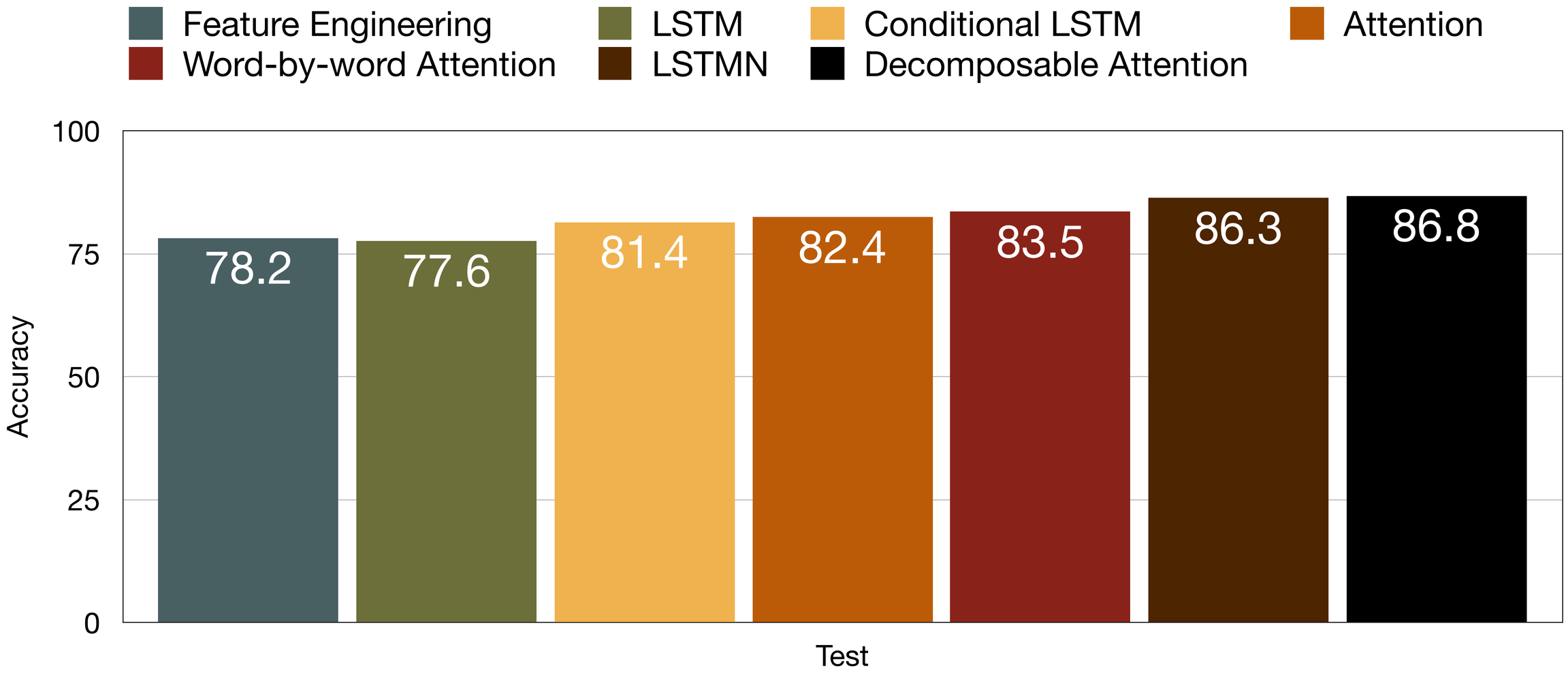
Compare

$$\mathbf{y} = \mathbf{mlp}_{\theta_2}\left(\left[\sum_i \mathbf{v}_i^h; \sum_j \mathbf{v}_j^p\right]\right)$$

Aggregate

Parikh et al. A Decomposable Attention Model for Natural Language Inference. EMNLP 2016.

Results


Feature Engineering
LSTM
Conditional LSTM
Attention
Word-by-word Attention
LSTMN
Decomposable Attention
100
75
50
25
0
Accuracy
78.2
77.6
81.4
82.4
83.5
86.3
86.8
Test

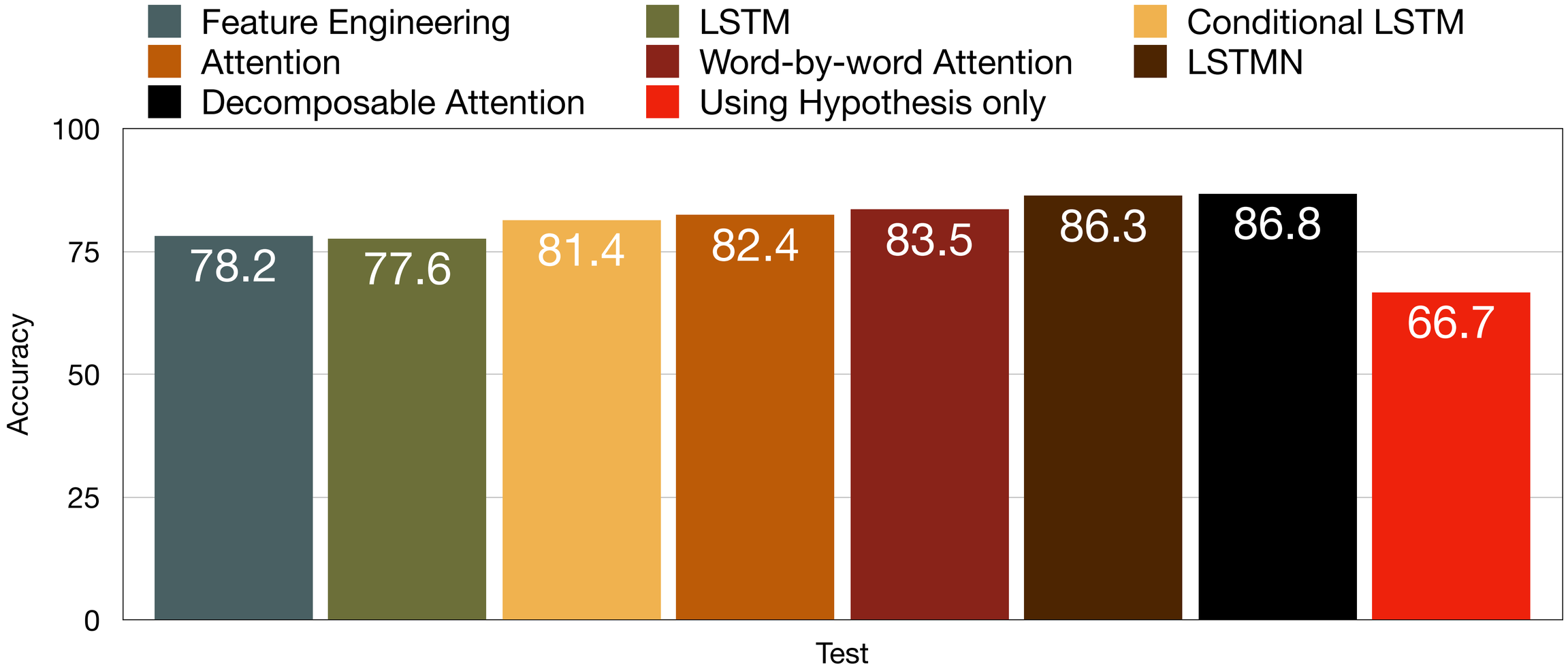
Caution: Biases in the data

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are **at least** three **people** on a loading dock.
Neutral	A woman is selling bamboo sticks **to help provide for her family.**
Contradiction	A woman is **not** taking money for any of her sticks.

Table 1: An instance from SNLI that illustrates the artifacts that arise from the annotation protocol. A common strategy for generating entailed hypotheses is to remove gender or number information. Neutral hypotheses are often constructed by adding a purpose clause. Negations are often introduced to generate contradictions.

Gururangan et al. Annotation Artifacts in Natural Language Inference Data. NAACL 2018.

Caution: Biases in the data



AAAS: Machine learning 'causing science crisis'

By Pallab Ghosh
Science correspondent, BBC News, Washington

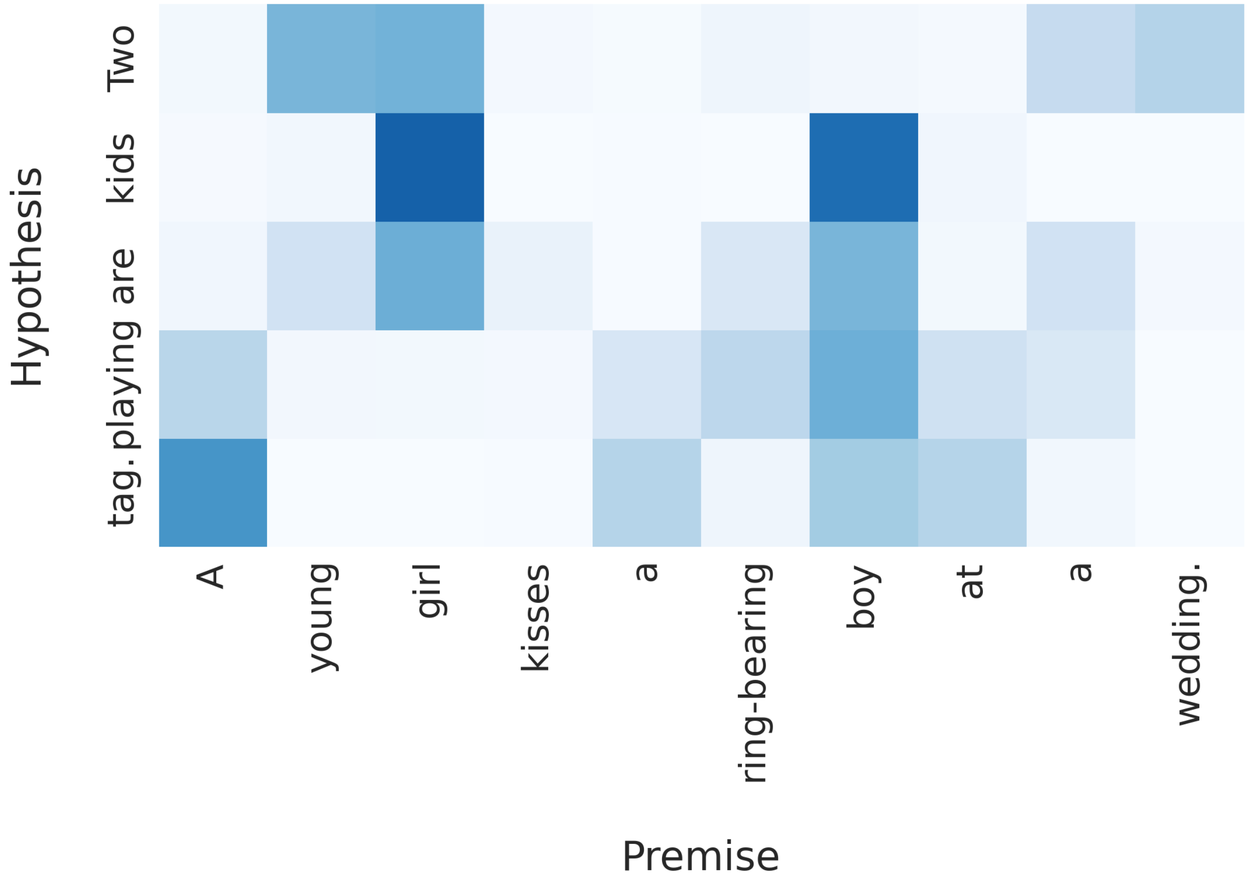
16 February 2019 | Science & Environment

“Often these studies are not found out to be inaccurate until there's another real big dataset that someone applies these techniques to and says ‘oh my goodness, the results of these two studies don't overlap‘," she said.

“There is general recognition of a **reproducibility crisis in science right now**. I would venture to argue that a **huge part of that does come from the use of machine learning** techniques in science.”

Machine learning systems and the use of big data sets has accelerated the crisis, according to Dr Allen. That is because **machine learning algorithms have been developed specifically to find interesting things in datasets and so when they search through huge amounts of data they will inevitably find a pattern**.

Caution: Attention and Interpretability



$$\mathbf{P} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$$

$$\mathbf{H} = [\mathbf{h}_{N+1}, \ldots, \mathbf{h}_M]$$

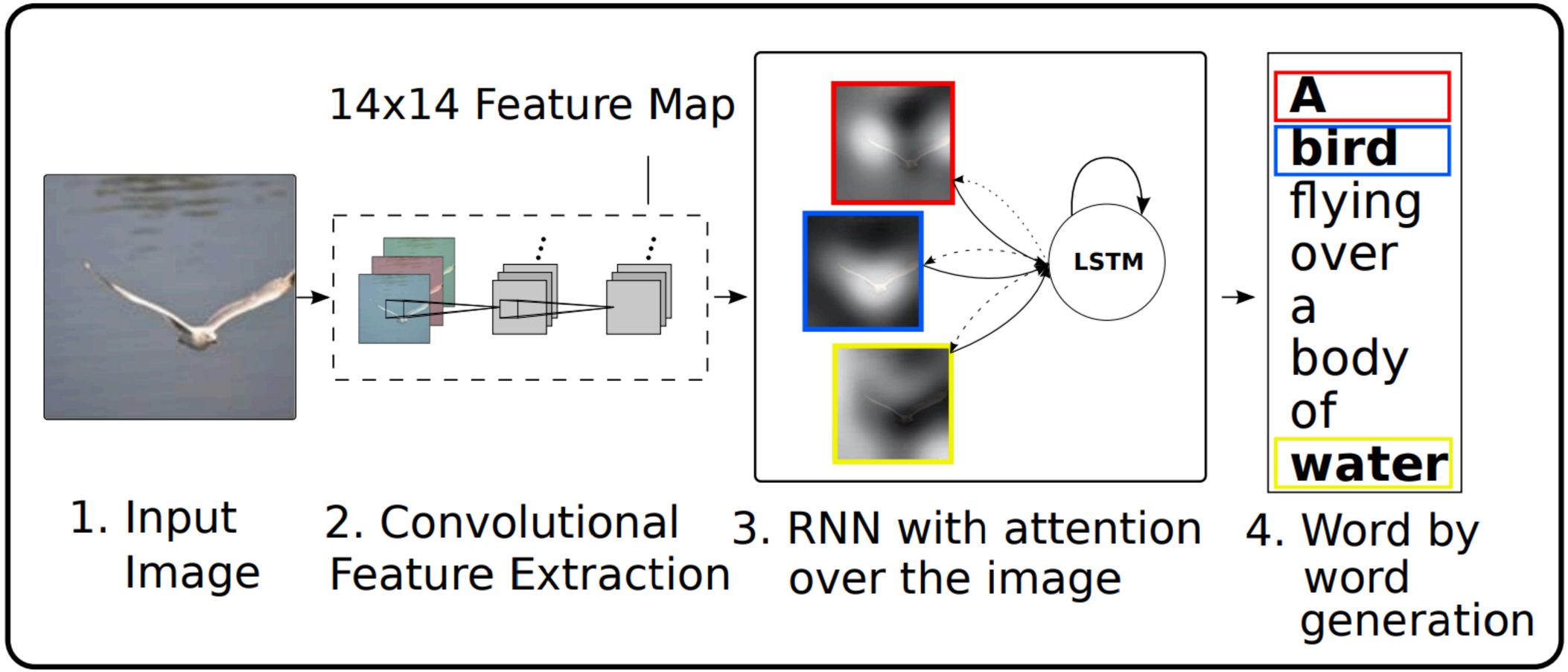
$$m_{ti} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{h}_t; \mathbf{p}_i] + \mathbf{b})$$

$$\alpha_t = \mathbf{softmax}(\mathbf{m}_{t:})$$

$$\mathbf{p}'_t = \alpha_t^\top \mathbf{P} + \tanh(\mathbf{W}^a \mathbf{p}'_{t-1})$$

$$\mathbf{h}'_t = \tanh(\mathbf{W}^p \mathbf{p}'_t + \mathbf{W}^h \mathbf{h}_t)$$

Multimodal



Xu et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. JMLR 2015.

Multimodal

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Xu et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. JMLR 2015.

Pointer Networks

- What if the vocabulary is large but many outputs in **y** are copies from **x**?

$$u_{it} = \mathbf{w}^\top \tanh(\mathbf{W}[\mathbf{h}_i^x; \mathbf{h}_t^y]) \qquad i \in (1, \ldots, N)$$

$$\alpha_t = \textbf{softmax}(\mathbf{u}_{:t})$$

$$\mathbf{h}'_t = \alpha^\top \mathbf{H}^x$$

Idea: Directly sample position from this distribution

Vinyals et al. Pointer Networks. NIPS 2015.