Seq2seq Models in NLP

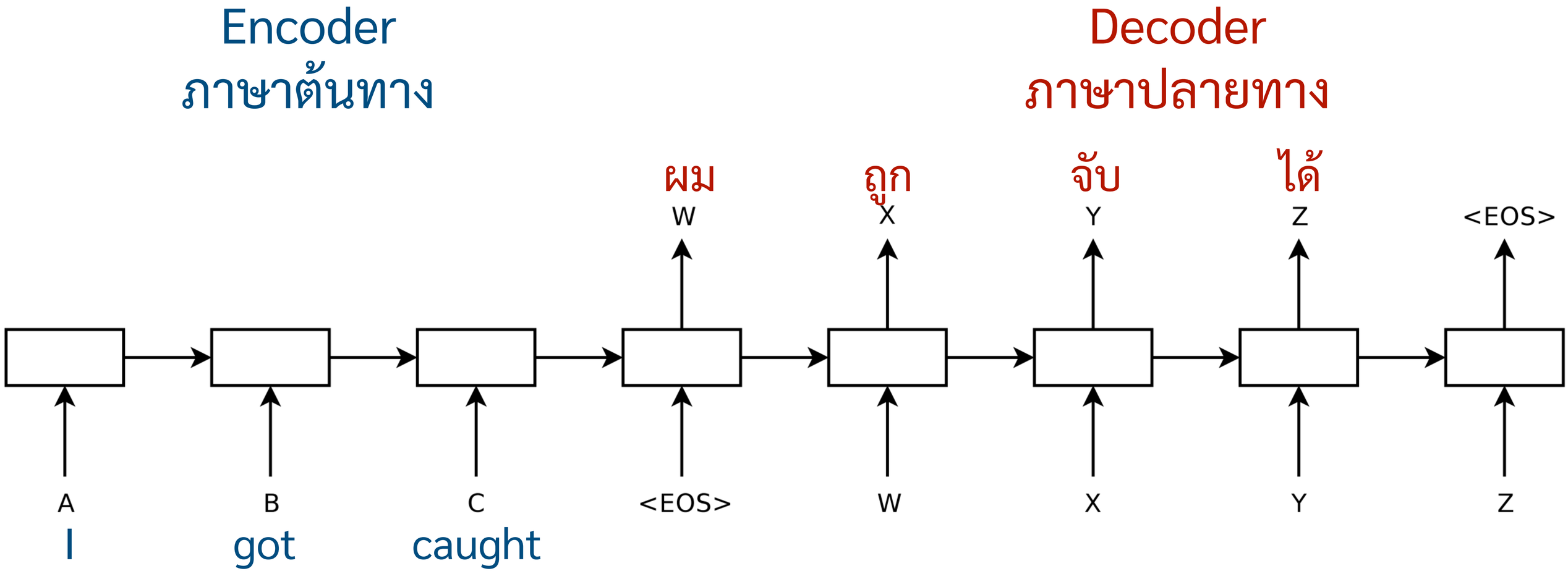
Model อื่นๆ ที่เคยเห็น

- Classification Models: Logistic Regression, Feedforward net
- Sequence Tagging Models: CRF, RNN

Sequence-to-sequence model คืออะไร

- Input: Sequence ของหน่วยทางภาษา
- Output: Sequence ของหน่วยทางภาษา
  (ความยาวไม่จำเป็นต้องเท่ากับ Input)

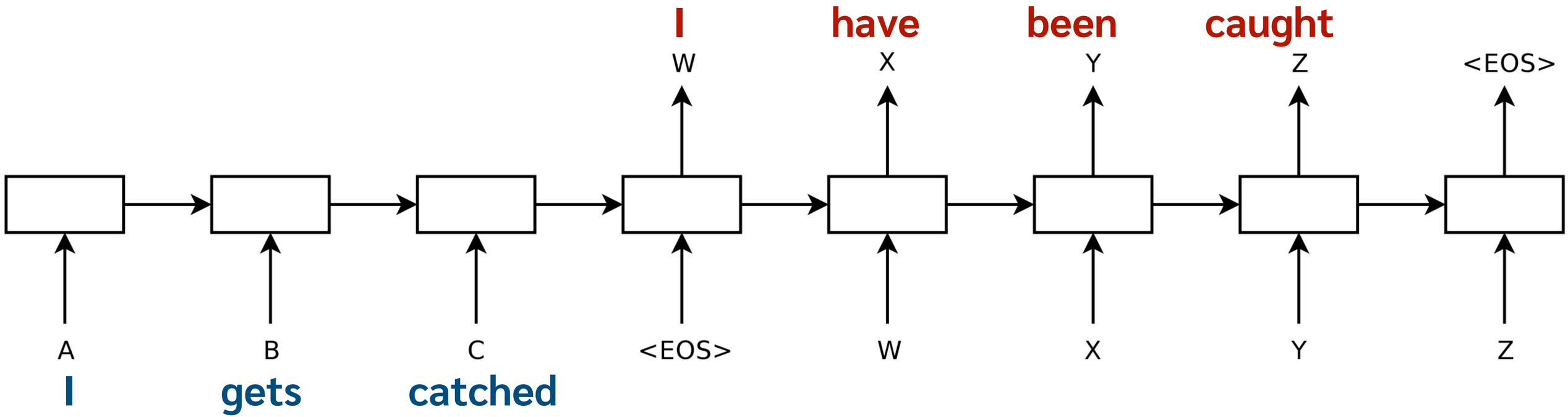
Machine Translation



Grammatical Error Correction

Encoder
ข้อความที่อาจจะมีข้อผิด

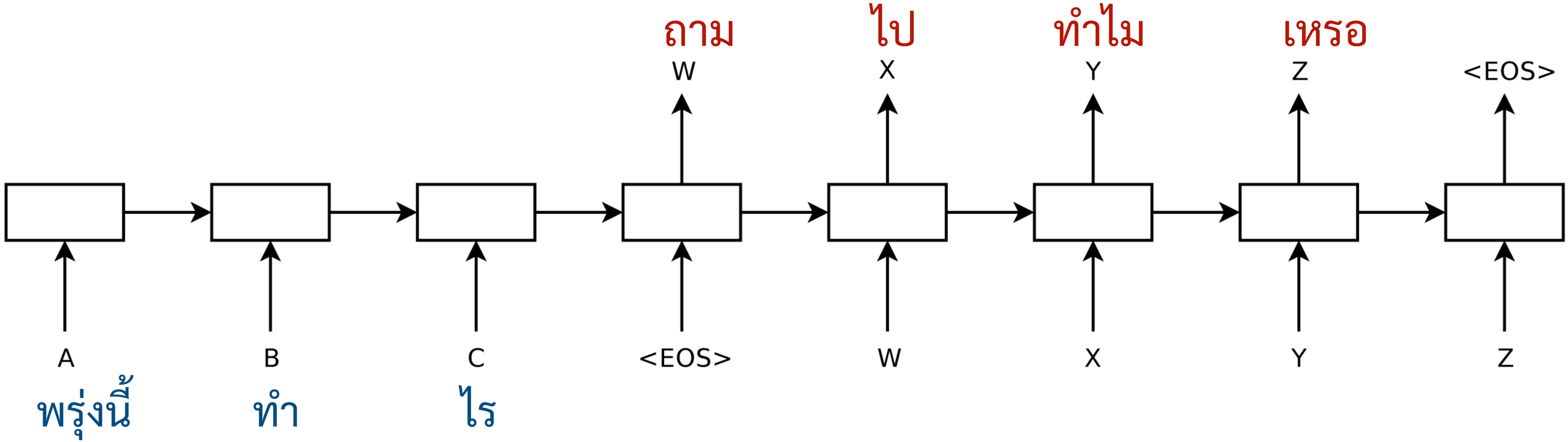
Decoder
ข้อความที่ถูกไม่มีข้อผิด



Chatbot

Encoder
สิ่งที่ได้ฟังมา

Decoder
สิ่งที่จะตอบโต้กลับไป

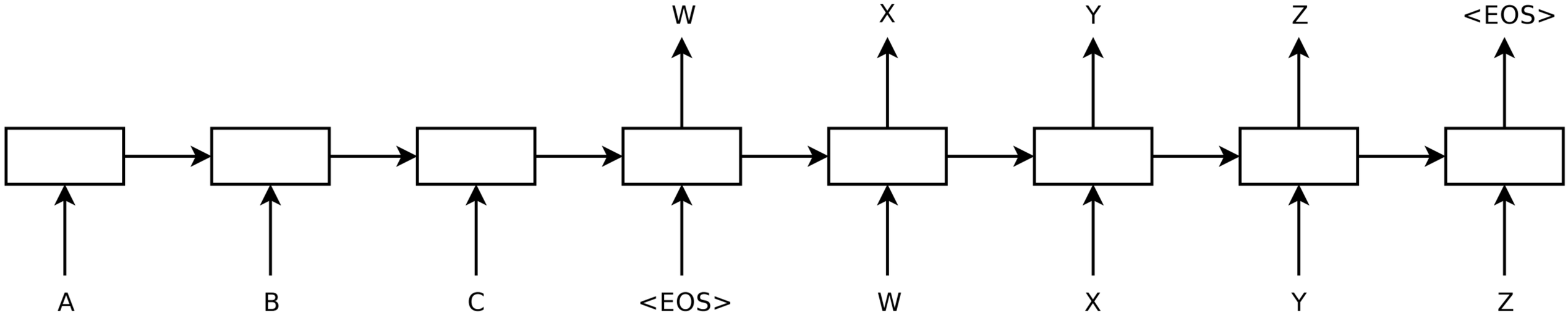


Text Simplification

Encoder
ข้อความที่เขียนด้วยภาษายาก

Decoder
ข้อความที่เขียนด้วยภาษาที่อ่านง่าย

The purpose of education is to develop many skills.

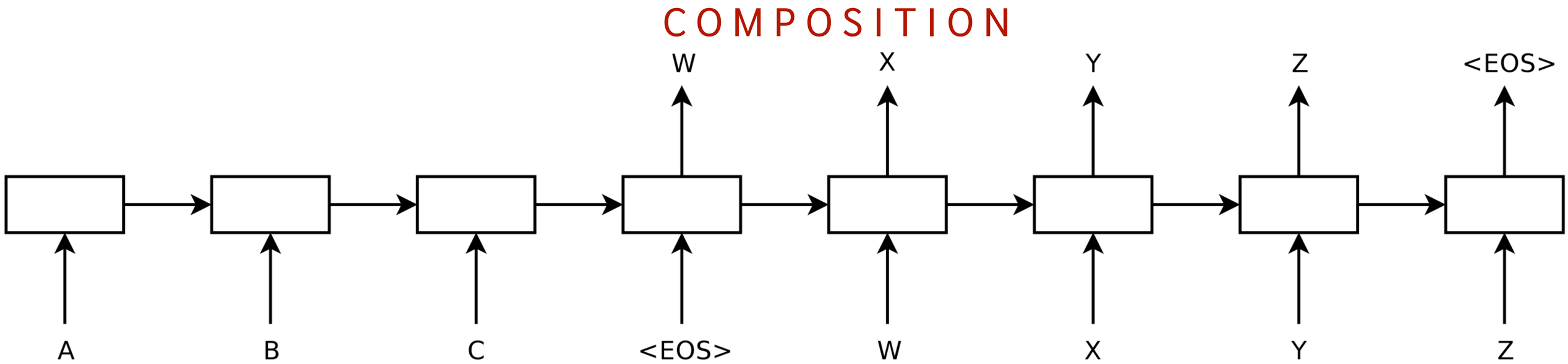


There’s just one major hitch: the primary purpose of education is to develop citizens with a wide variety of skills

Derivational Morphology

Encoder
ตัวอักษรในคำ

Decoder
ตัวอักษรในคำ



C O M P O S E <VerbNominalization>

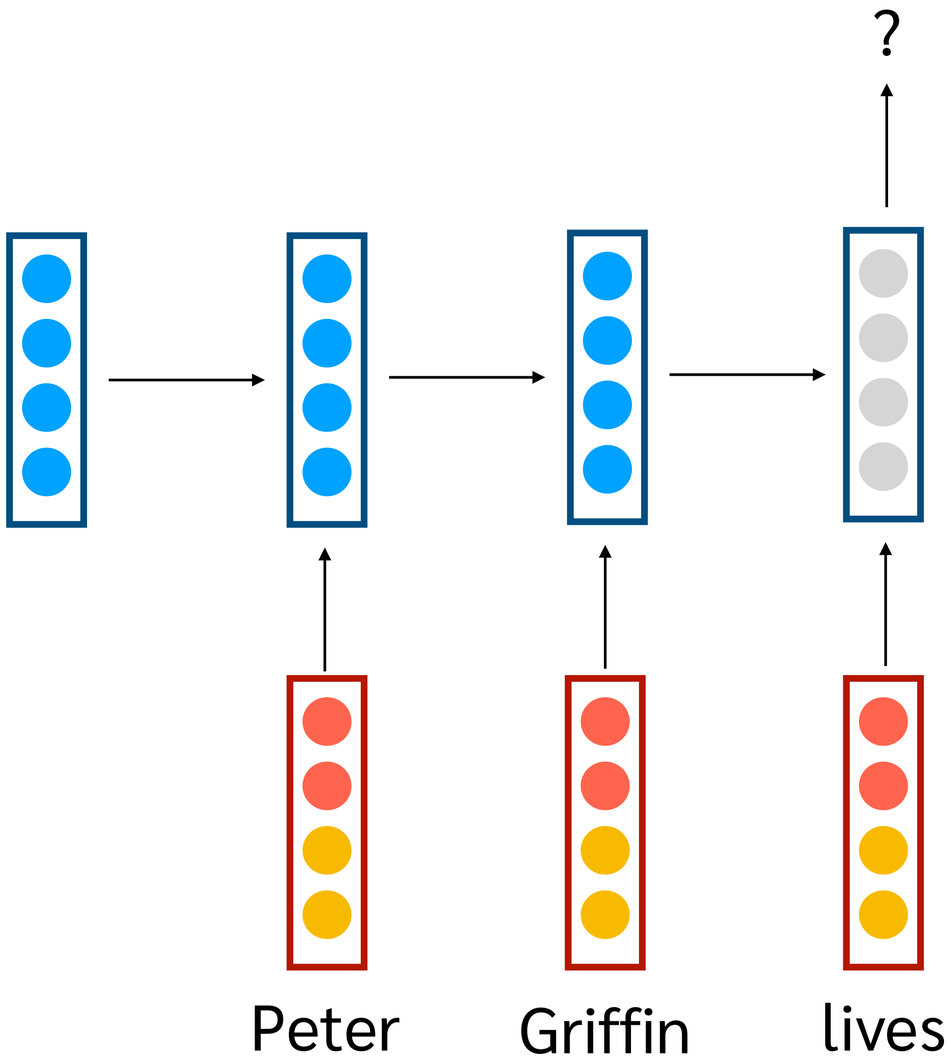
RNN Language Model

Language Modeling

- คำนวณความน่าจะเป็นของประโยค
  P(*Peter Griffin lives in Quahog*) vs
  P(*Peter Griffin lives over Quahog*)

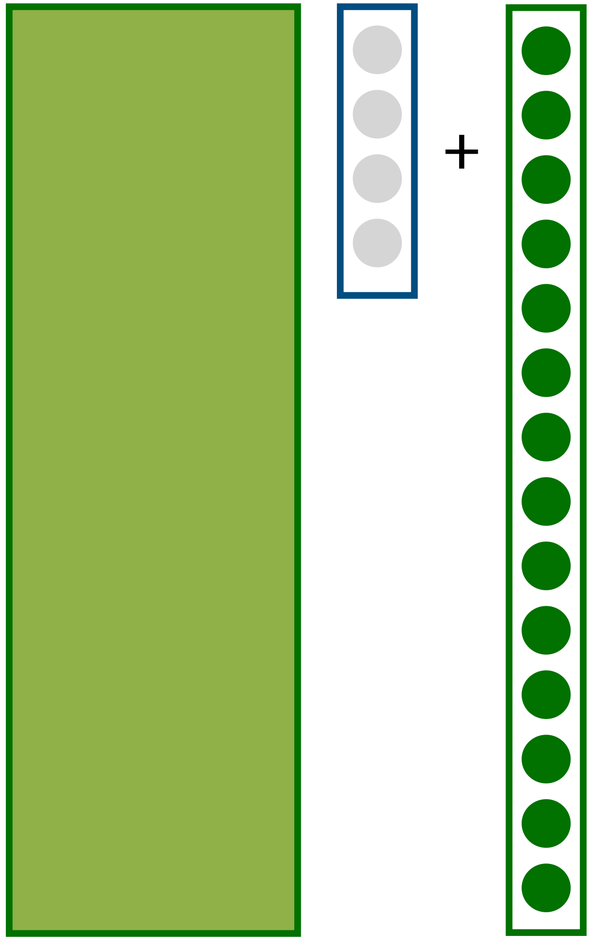
- ทำนายคำที่น่าจะเห็นเป็นคำถัดไป
  Peter Griffin lives _____ .
  Peter Griffin lives in _____ .

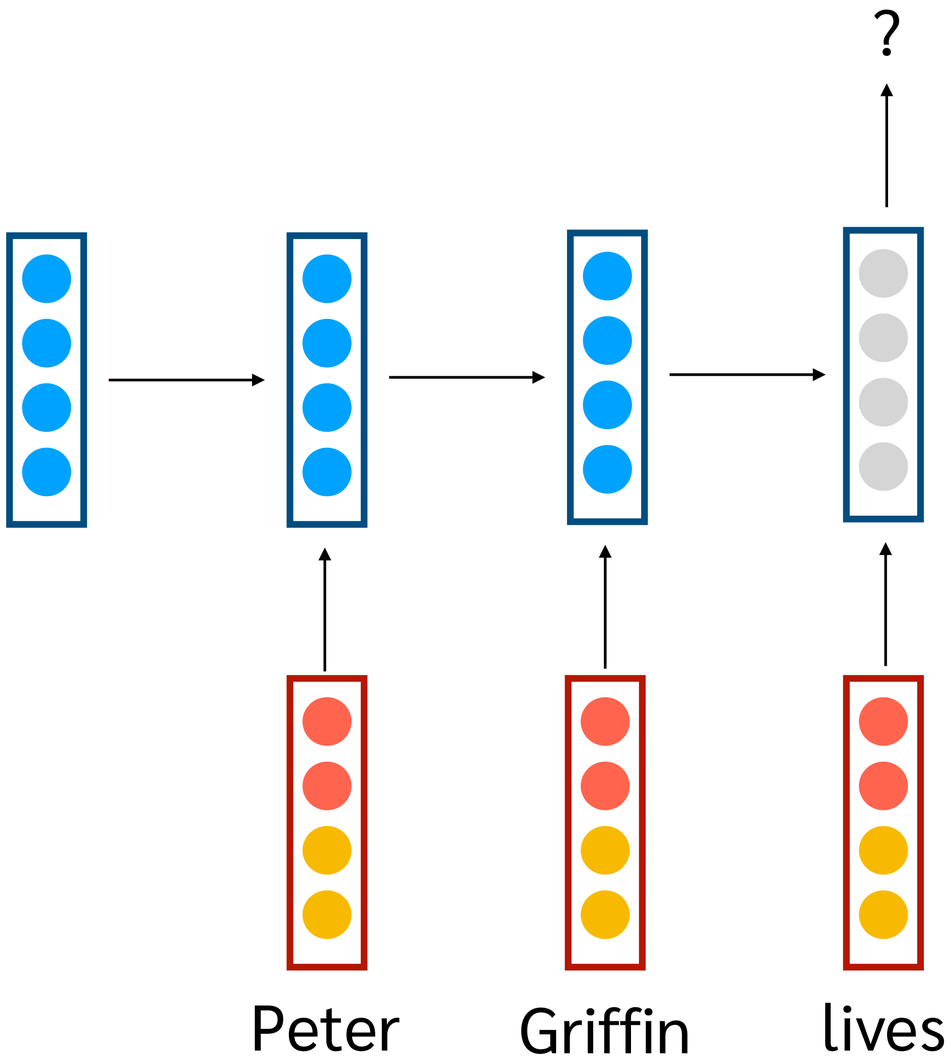
- Generate text ที่มีลักษณะคล้ายกับ training data



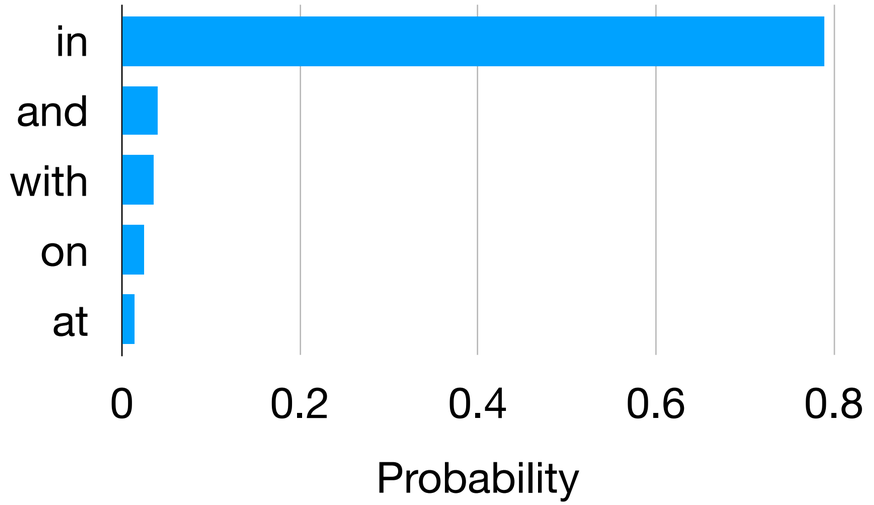
คำต่อไปคือคำว่าอะไร

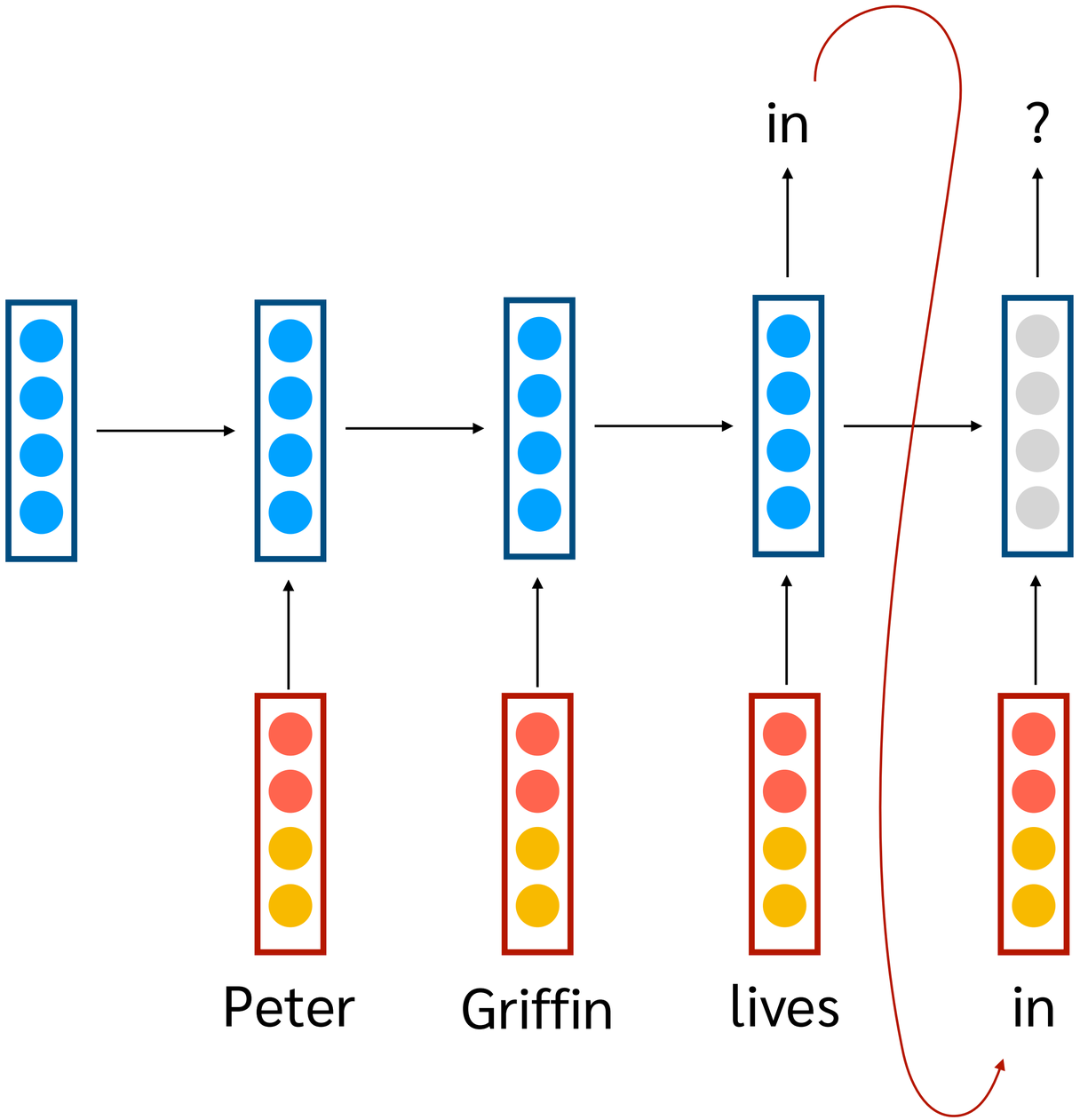
$$y_t = softmax(W_y \cdot h_t + b_y)$$





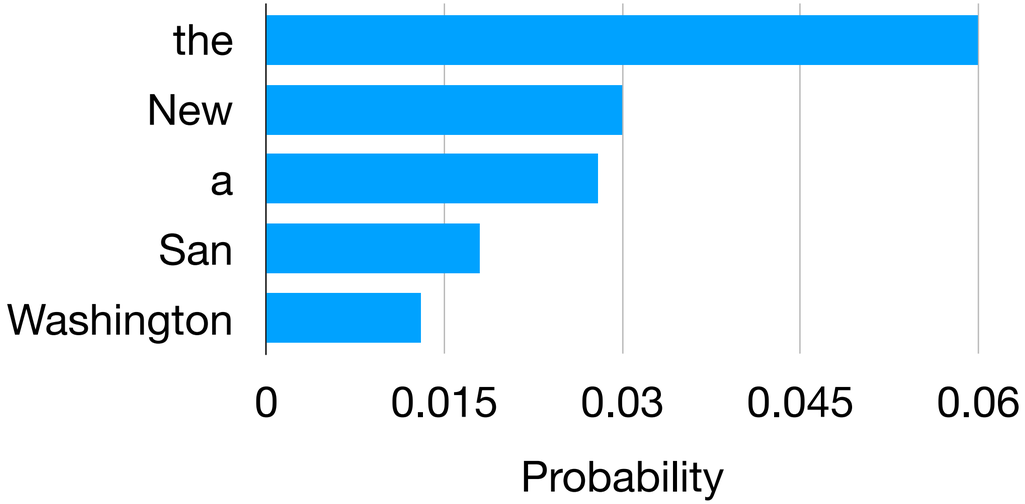
คำต่อไปคือคำว่าอะไร





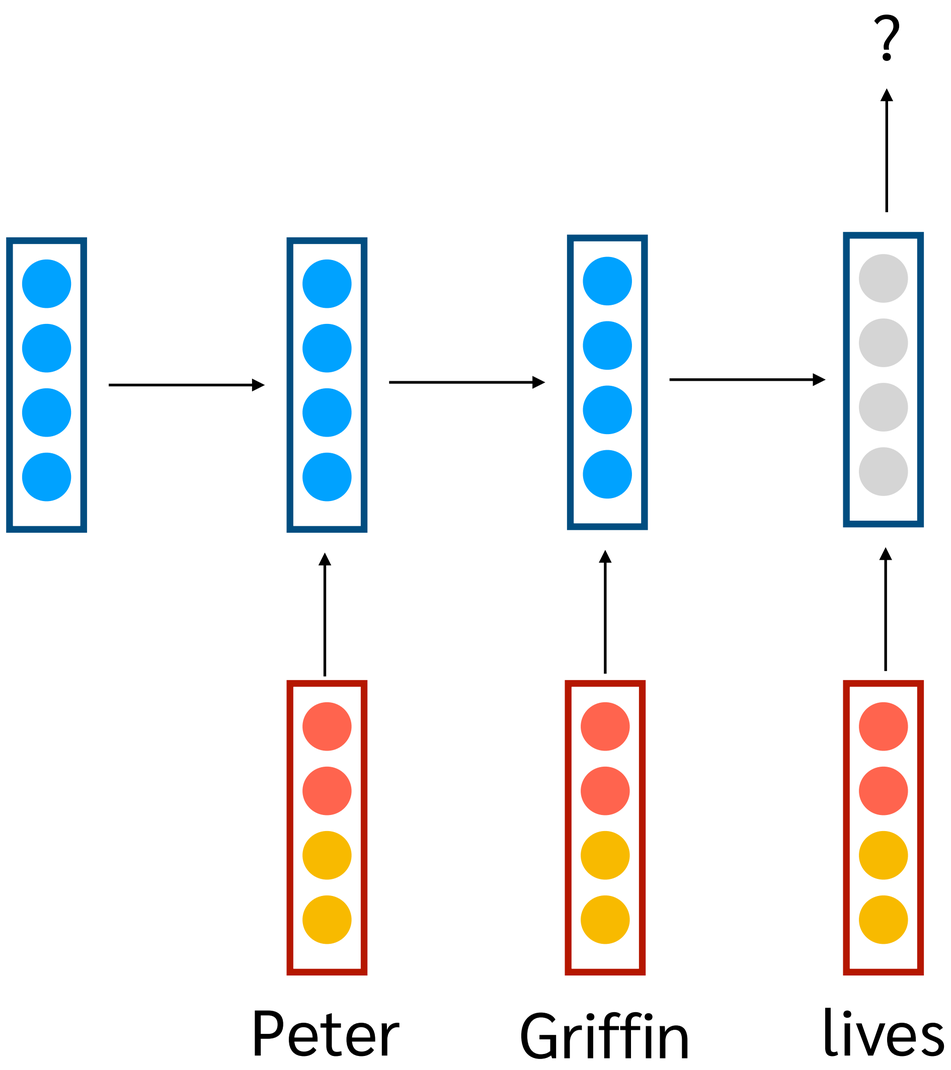
$$y_t = softmax(W_y \cdot h_t + b_y)$$

คำต่อไปคือคำว่าอะไร



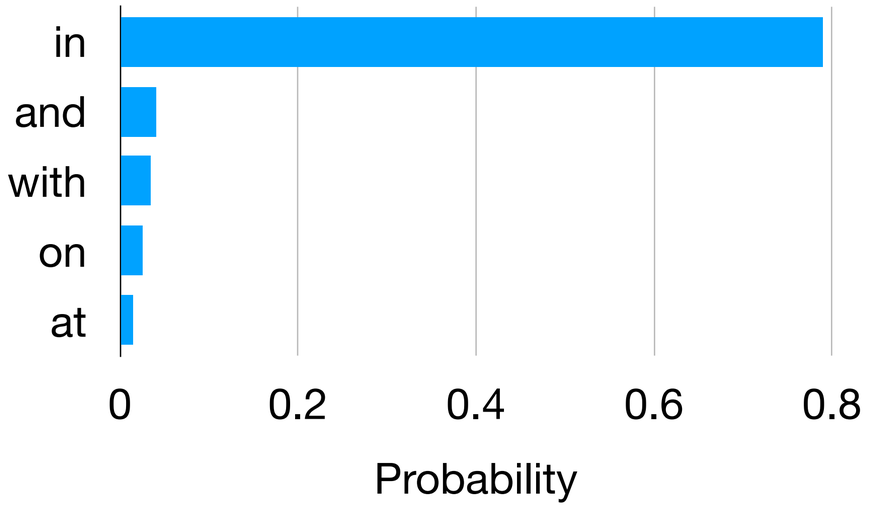
RNN Language Model

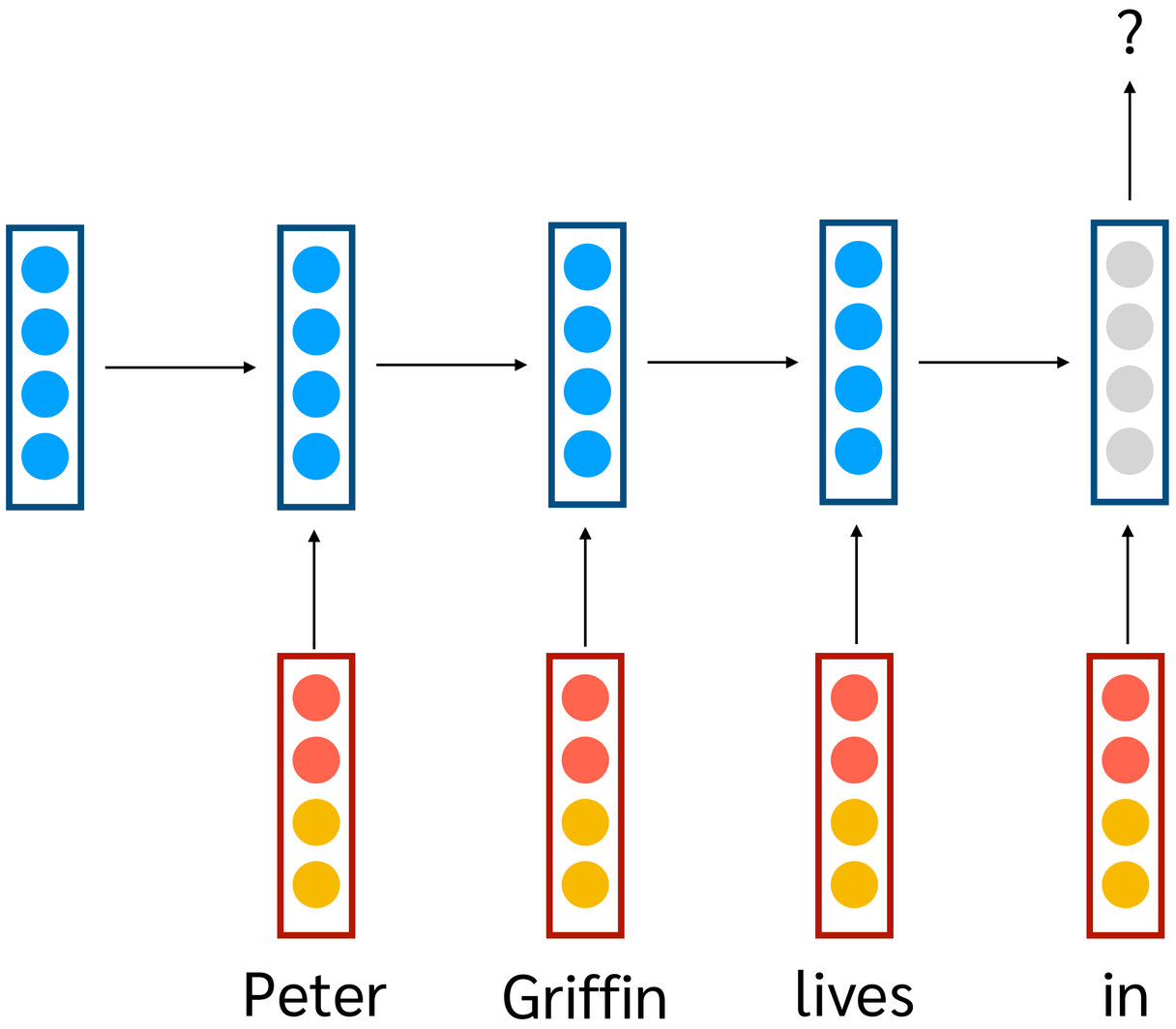
- Input: Context vector (hidden activation)
- Label: 1 ใน 100,000 label



Training Process คือการทำให้คำเจอจริงๆ ได้ probability สูงสุด

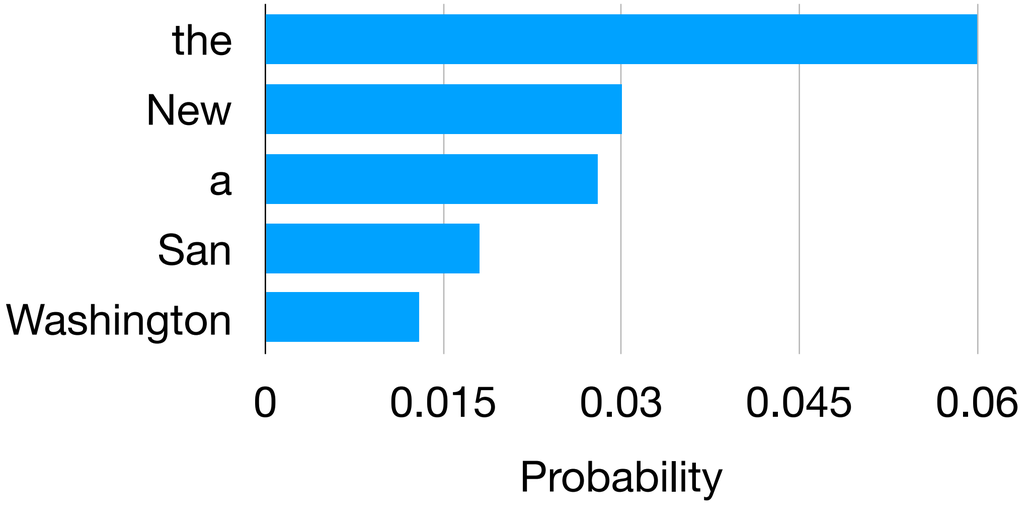
P(in | Peter Griffin lives)





Training Process คือการทำให้คำเจอจริงๆ ได้ probability สูงสุด

P(Quahog | Peter Griffin lives in)

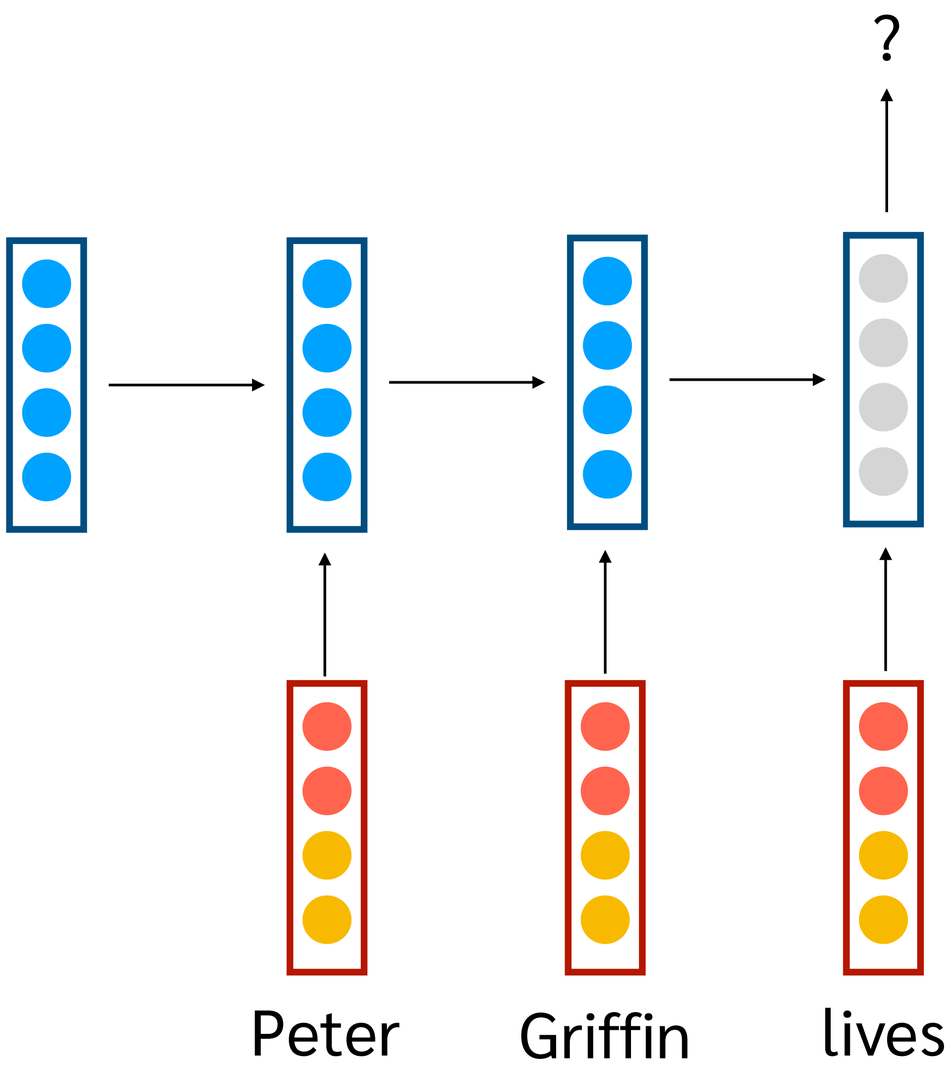


Language Modeling

- คำนวณความน่าจะเป็นของประโยค
  P(*Peter Griffin lives in Quahog*) vs
  P(*Peter Griffin lives over Quahog*)

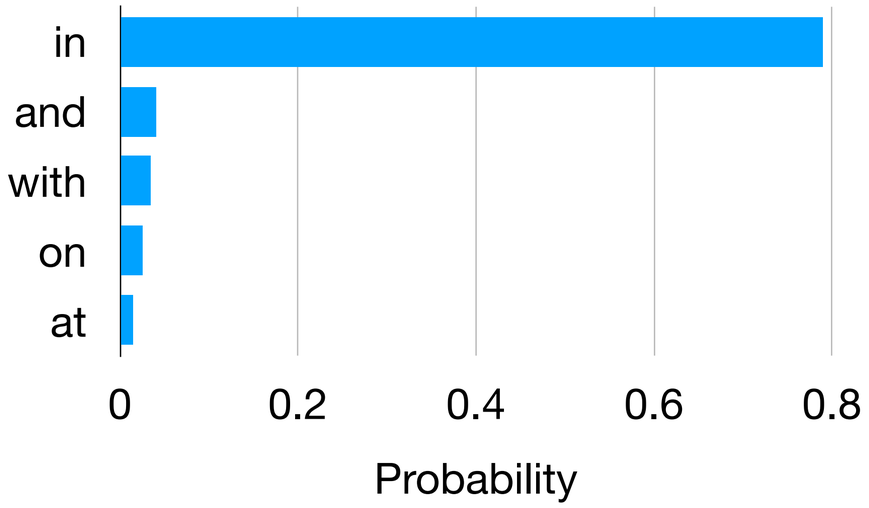
- ทำนายคำที่น่าจะเห็นเป็นคำถัดไป
  Peter Griffin lives _____ .
  Peter Griffin lives in _____ .

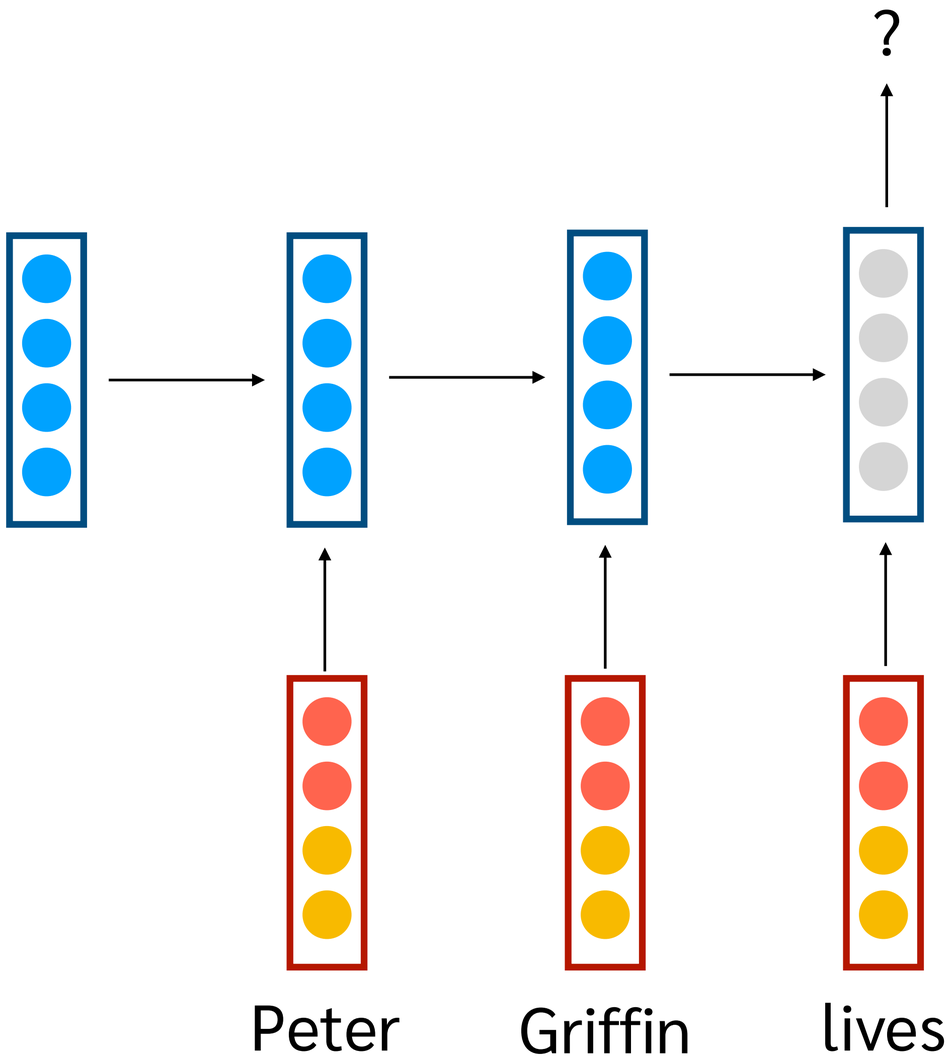
- Generate text ที่มีลักษณะคล้ายกับ training data

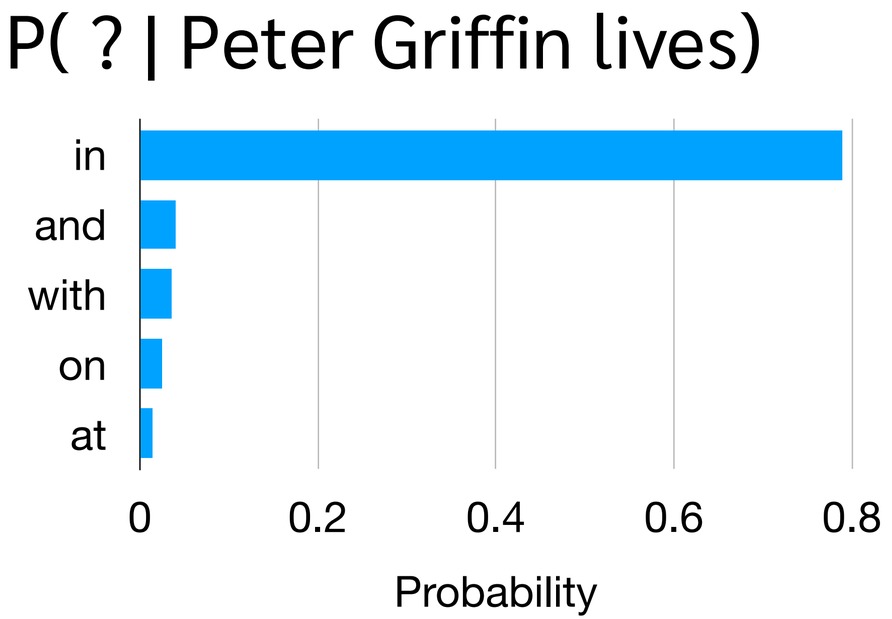


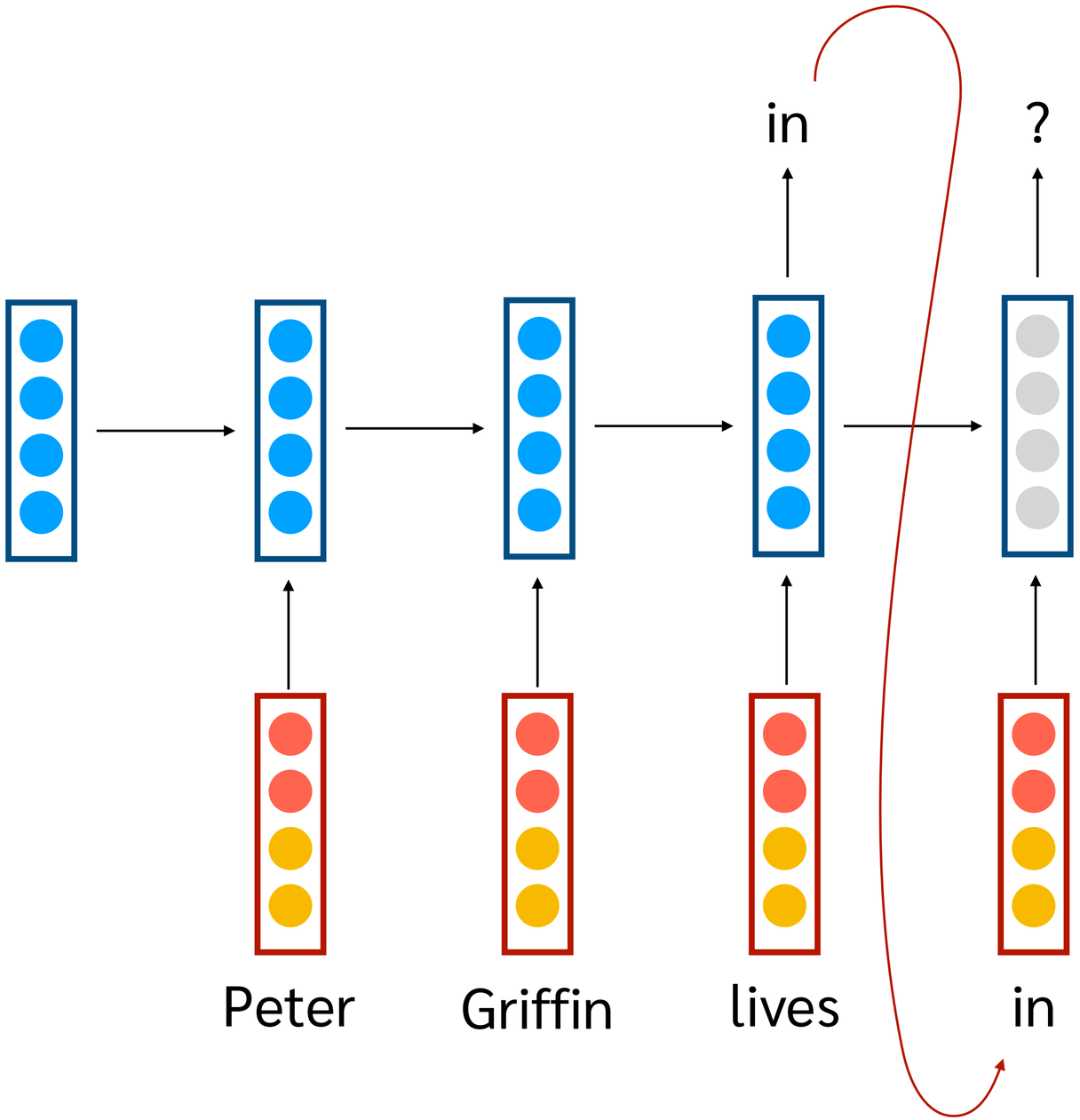
Training Process คือการทำให้คำเจอจริงๆ ได้ probability สูงสุด

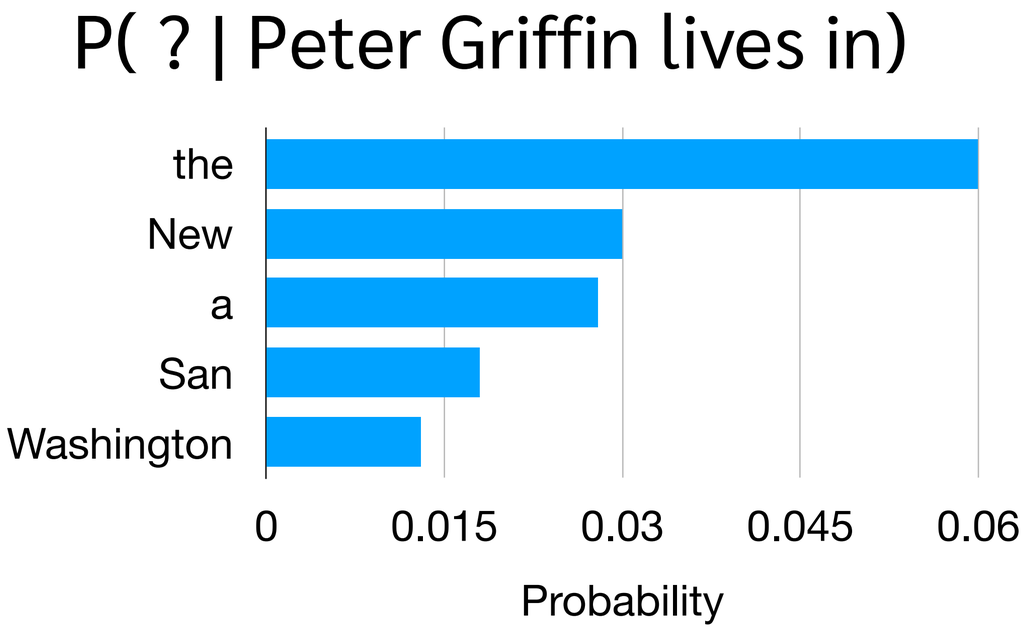
P(in | Peter Griffin lives)

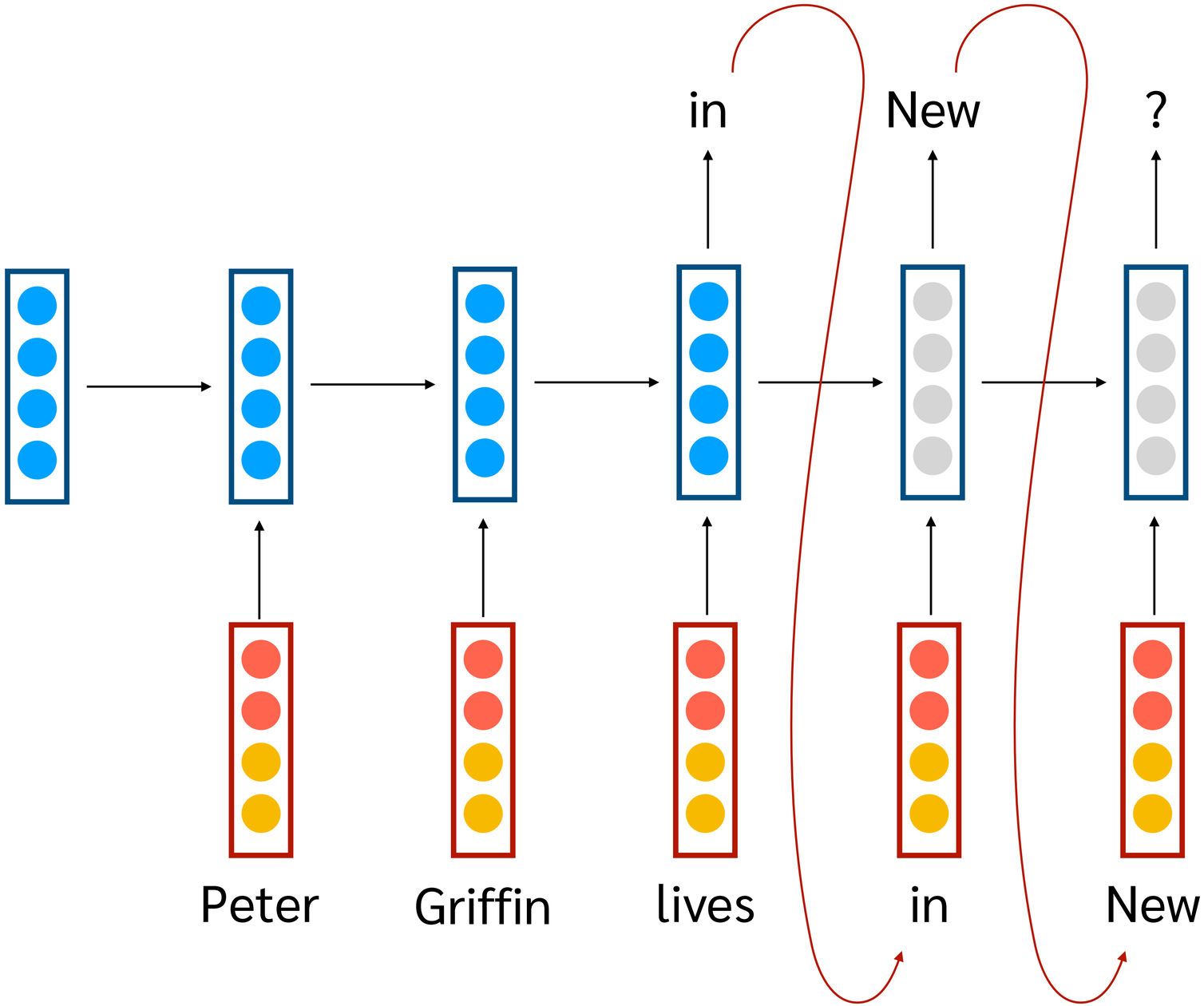


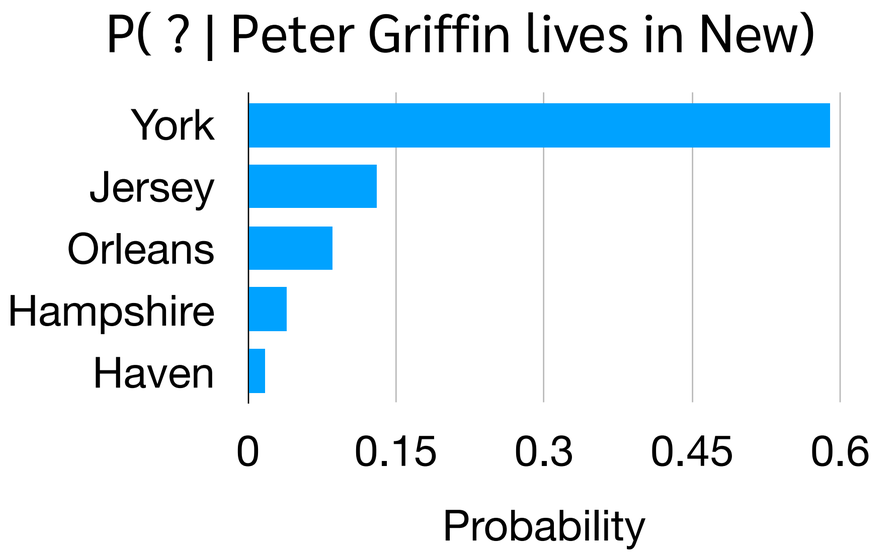

?
Peter
Griffin
lives


P(? | Peter Griffin lives)
in
and
with
on
at
0
0.2
0.4
0.6
0.8
Probability


in
?
Peter
Griffin
lives
in


P(? | Peter Griffin lives in)
the
New
a
San
Washington
0
0.015
0.03
0.045
0.06
Probability

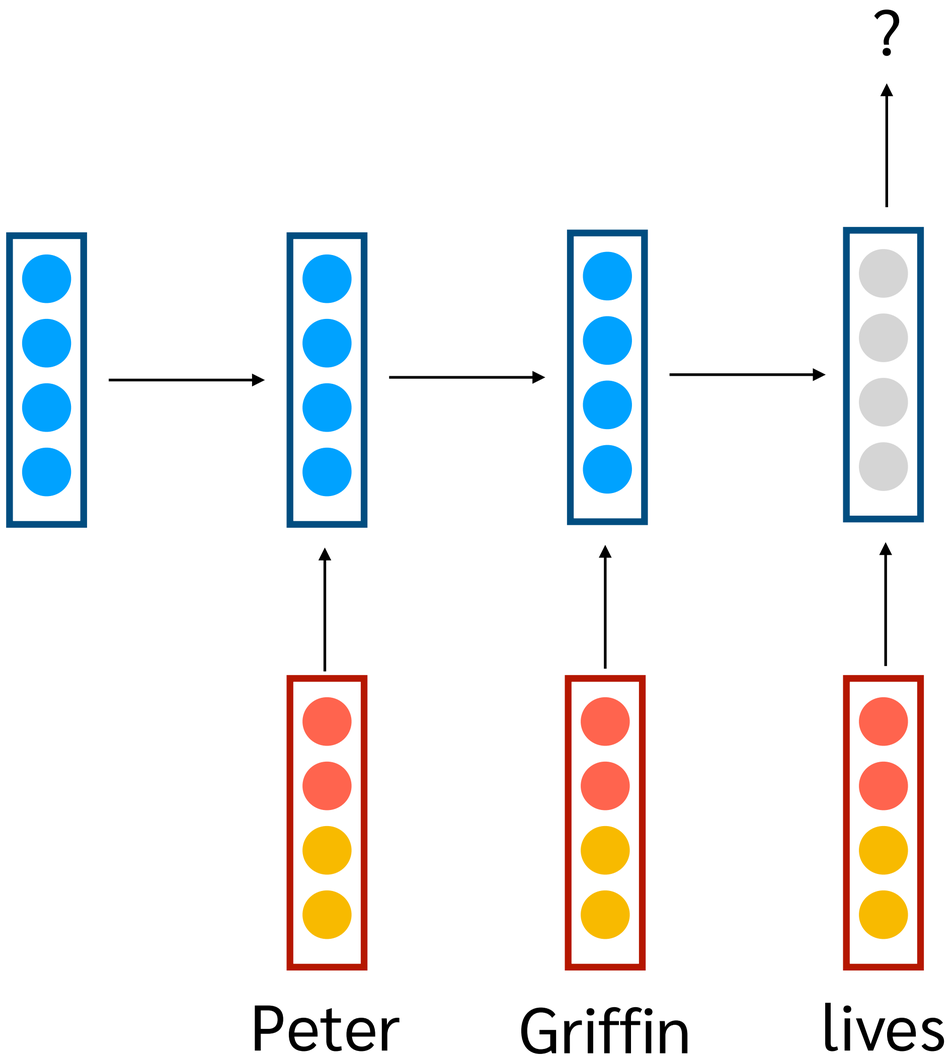

in
New
?
Peter
Griffin
lives
in
New


P(? | Peter Griffin lives in New)
York
Jersey
Orleans
Hampshire
Haven
0
0.15
0.3
0.45
0.6
Probability

RNN Language Model

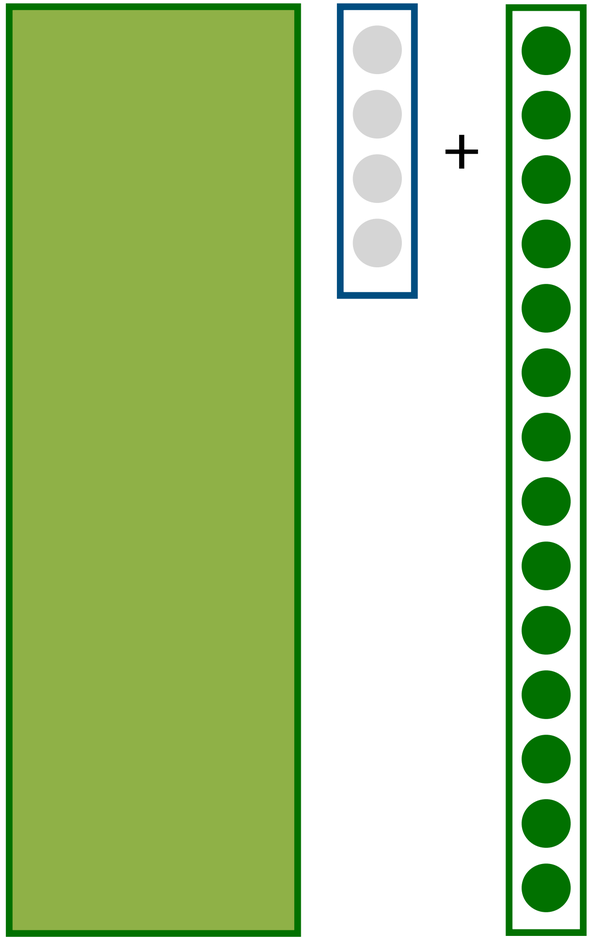
- Generate Text ได้โดยการสุ่มจาก Probability ที่โมเดลคำนวณมาให้
- คำต่อไปถูกทำนายจากบริบทที่อยู่ทางด้านซ้ายมือ
- คำที่ทำนายกลายเป็น input สำหรับ step ถัดไป

Conditional Language Model



คำต่อไปคือคำว่าอะไร

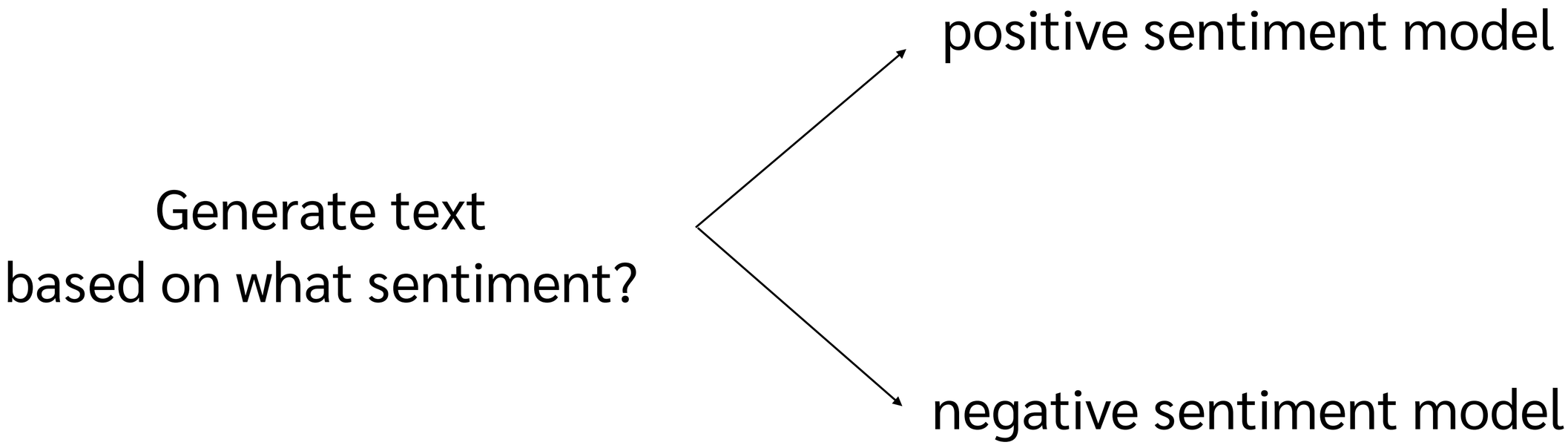
$$y_t = softmax(W_y \cdot h_t + b_y)$$



Conditional Language Model

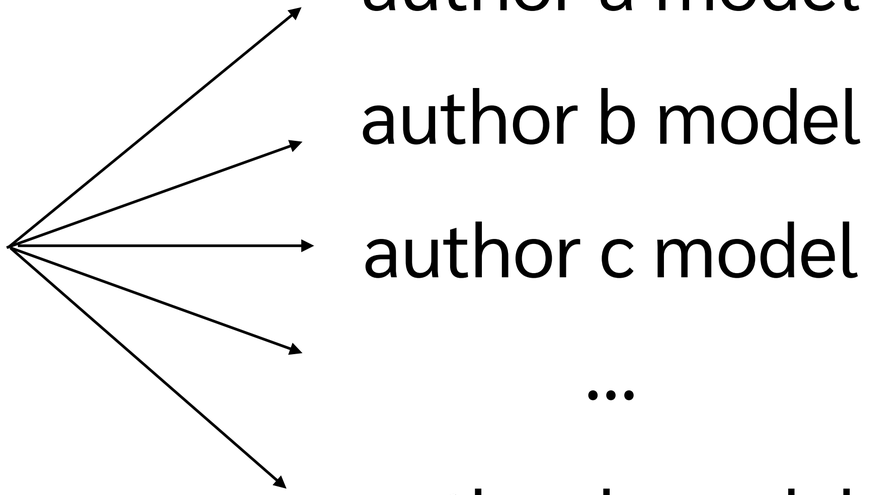
- ใช้ตัวแปรหรือ information อื่น ๆ ในการสร้าง language model ที่ดีขึ้น
  - คำนวณ probability ของประโยค
  - predict คำต่อไปในประโยค
  - generate คำในประโยค

Conditioned on Sentiment



Conditioned on Authors

Generate text
based on which author?



Conditional vs unconditional LM

$$\log P(W) = \sum_{t=1}^{L} \log P(w_t \,|\, w_1, w_2, \ldots, w_{t-1})$$

$$\log P(W \,|\, x) = \sum_{t=1}^{L} \log P(w_t \,|\, x, w_1, w_2, \ldots, w_{t-1})$$

ตัวอย่าง Conditional LM

x (เงื่อนไขหรือ information เพิ่มเติม)	Generated Text
Topic = {technology, กีฬา, การเมือง}	บทความเกี่ยวกับหัวข้อที่กำหนด
ผู้แต่ง	บทความเขียนโดยผู้แต่งที่กำหนด
จุดยืน = {เห็นด้วย, ไม่เห็นด้วย, ไม่มีความเห็น}	Text ที่แสดงจุดยืนที่กำหนด
ประโยคภาษาจีน	คำแปลภาษาไทย
ประโยคภาษาไทย	คำแปลภาษาอังกฤษ
ประโยคที่ทางการเข้าใจยาก	ประโยคที่ย่อยลงเข้าใจง่าย

สรุป

- Conditional Language Model เปิดโอกาสให้เราใส่ตัวแปรอื่นที่ทำให้ LM เราเฉพาะเจาะจงมากขึ้น
- ถ้าตัวแปรนั้นเป็น text จะต้องทำการ encode text ให้เป็น feature ก่อน

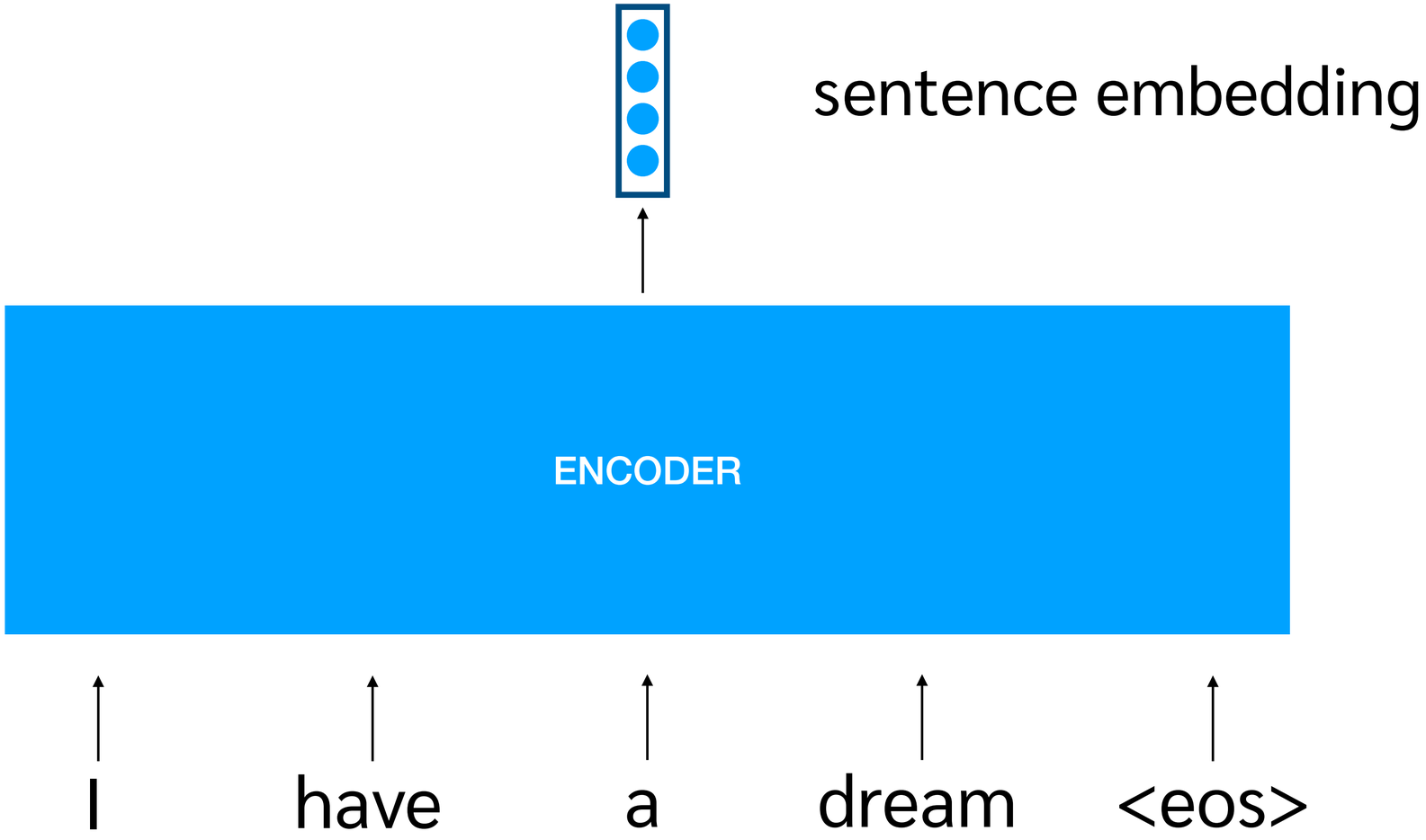
Encoder-Decoder Model

คือ conditional language model ที่เงื่อนไขเป็น Text

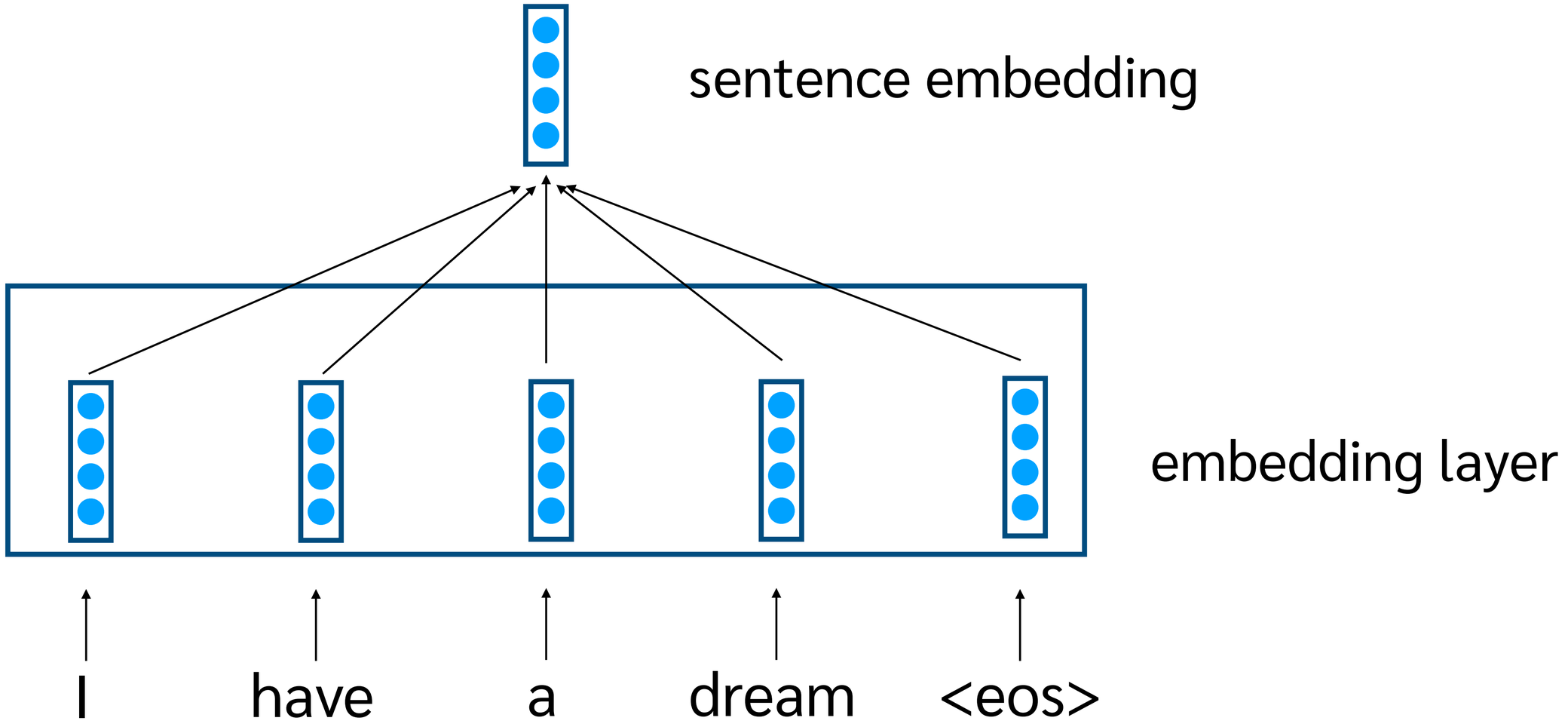
Neural Machine Translation

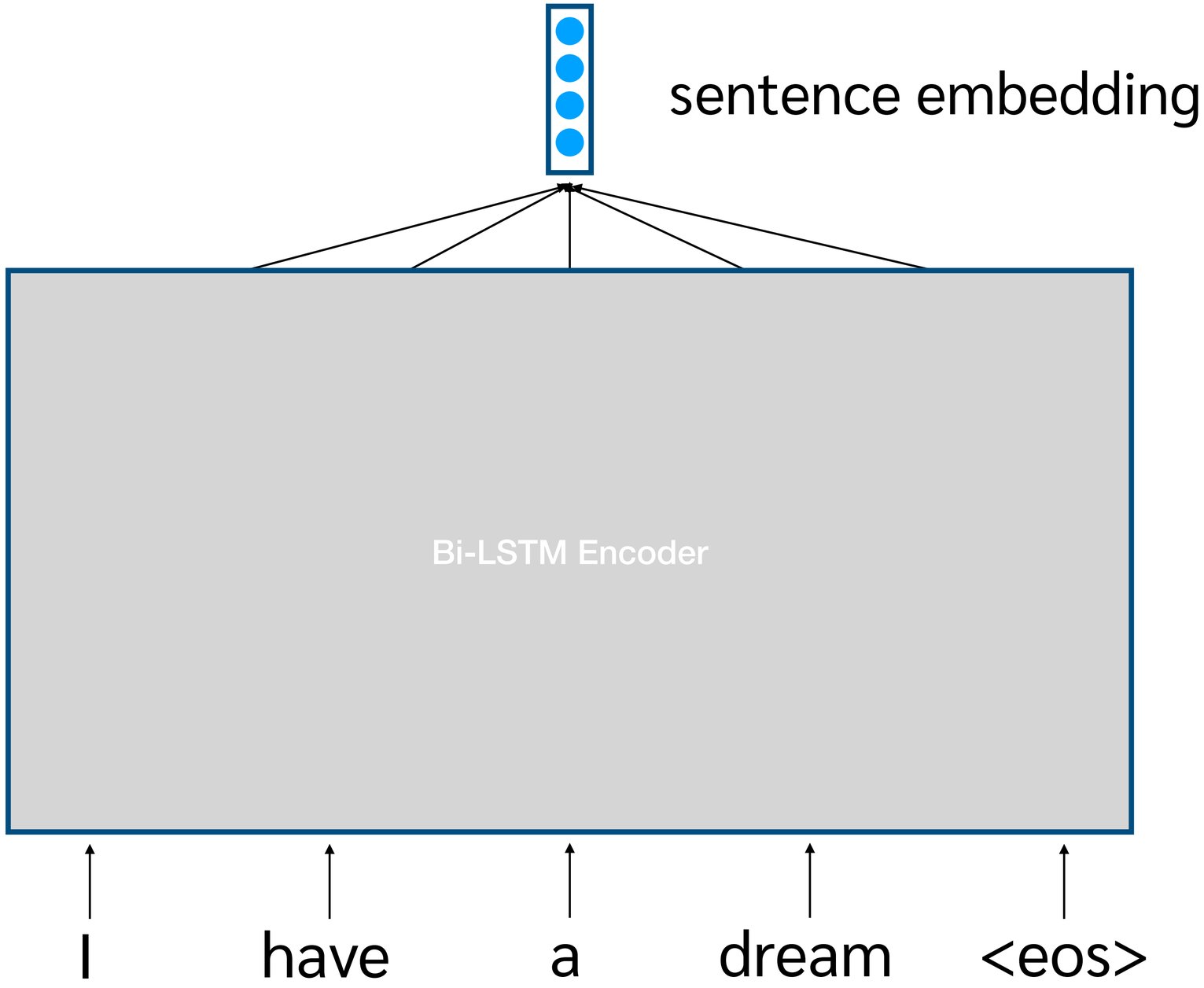
- Encoder-decoder model คือ conditional language model ที่
  - เงื่อนไข: Text เป็นภาษาต้นฉบับ
  - output: Text คำแปล

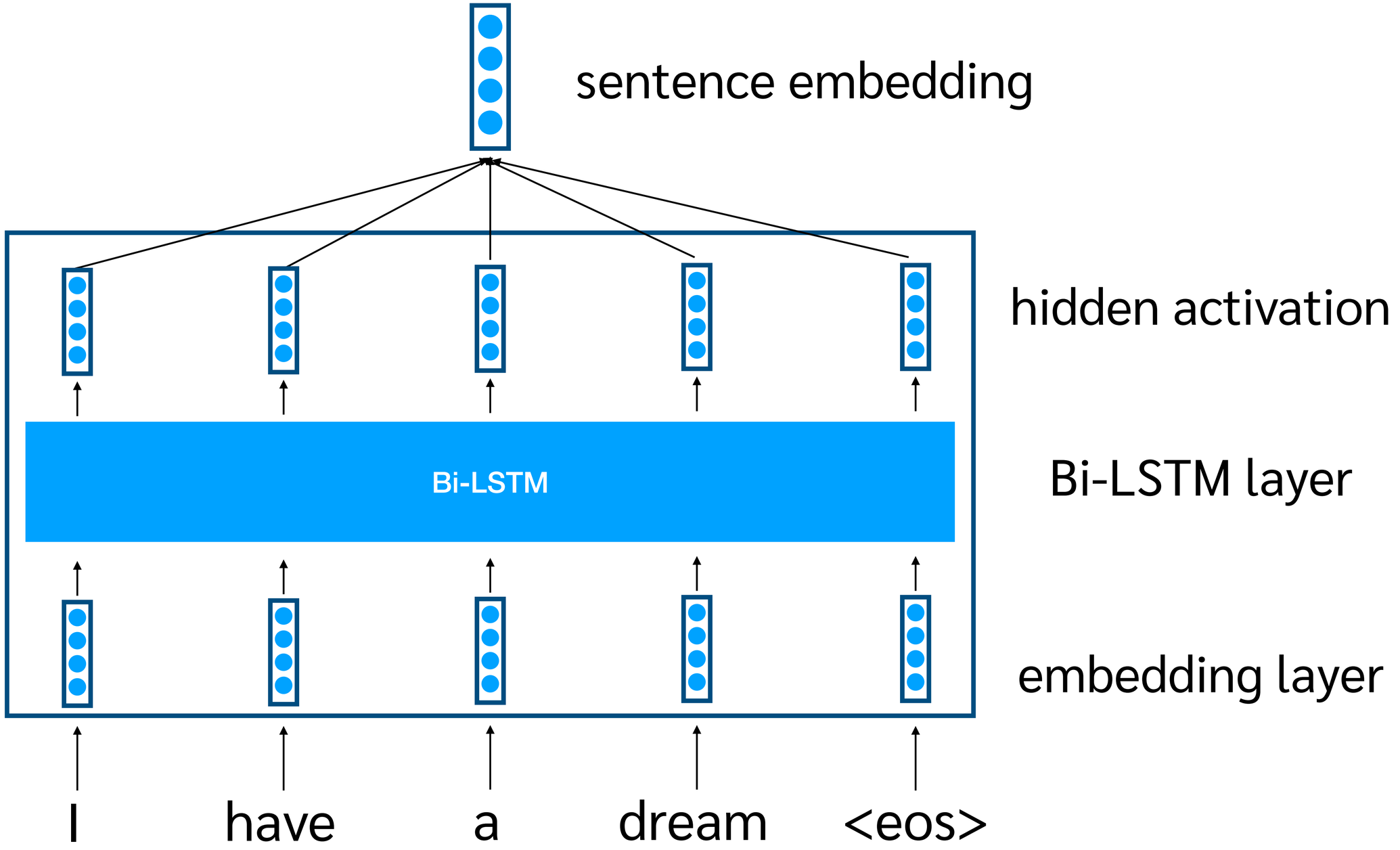
Encode = แปลงประโยคเป็น embedding



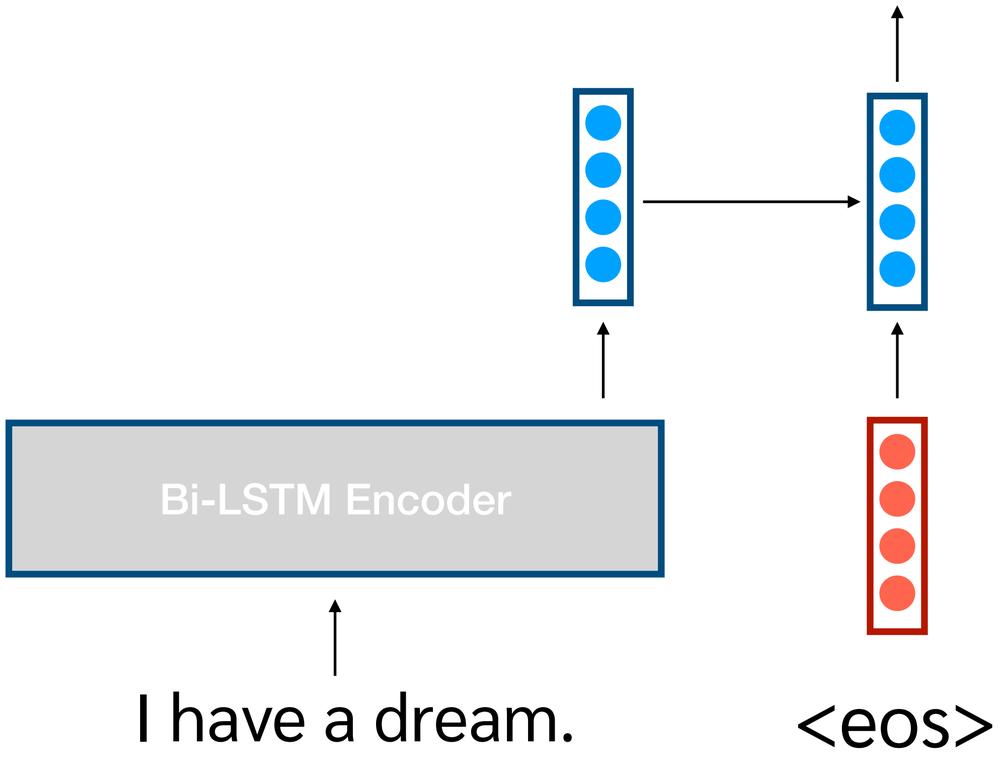
Average Pooling



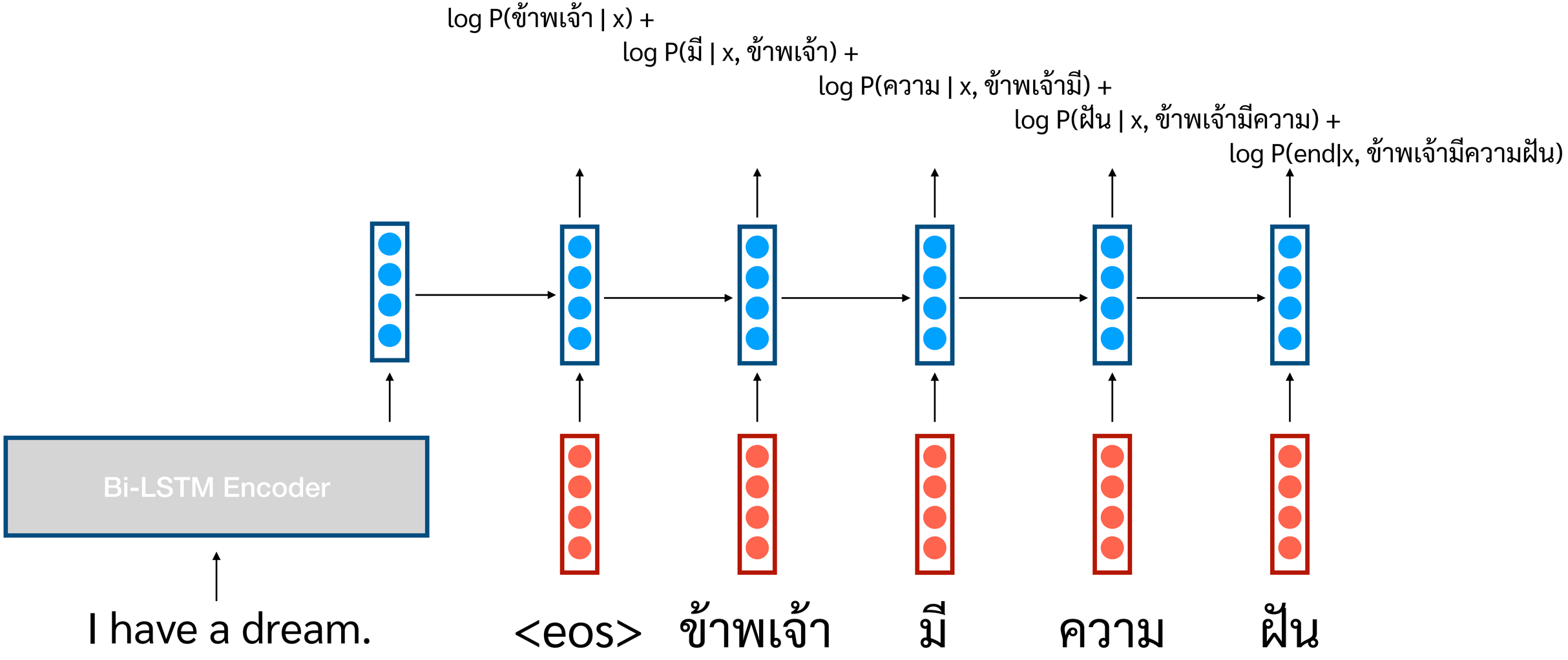

sentence embedding
Bi-LSTM Encoder
I
have
a
dream
<eos>


sentence embedding
hidden activation
Bi-LSTM
Bi-LSTM layer
embedding layer
I
have
a
dream
<eos>

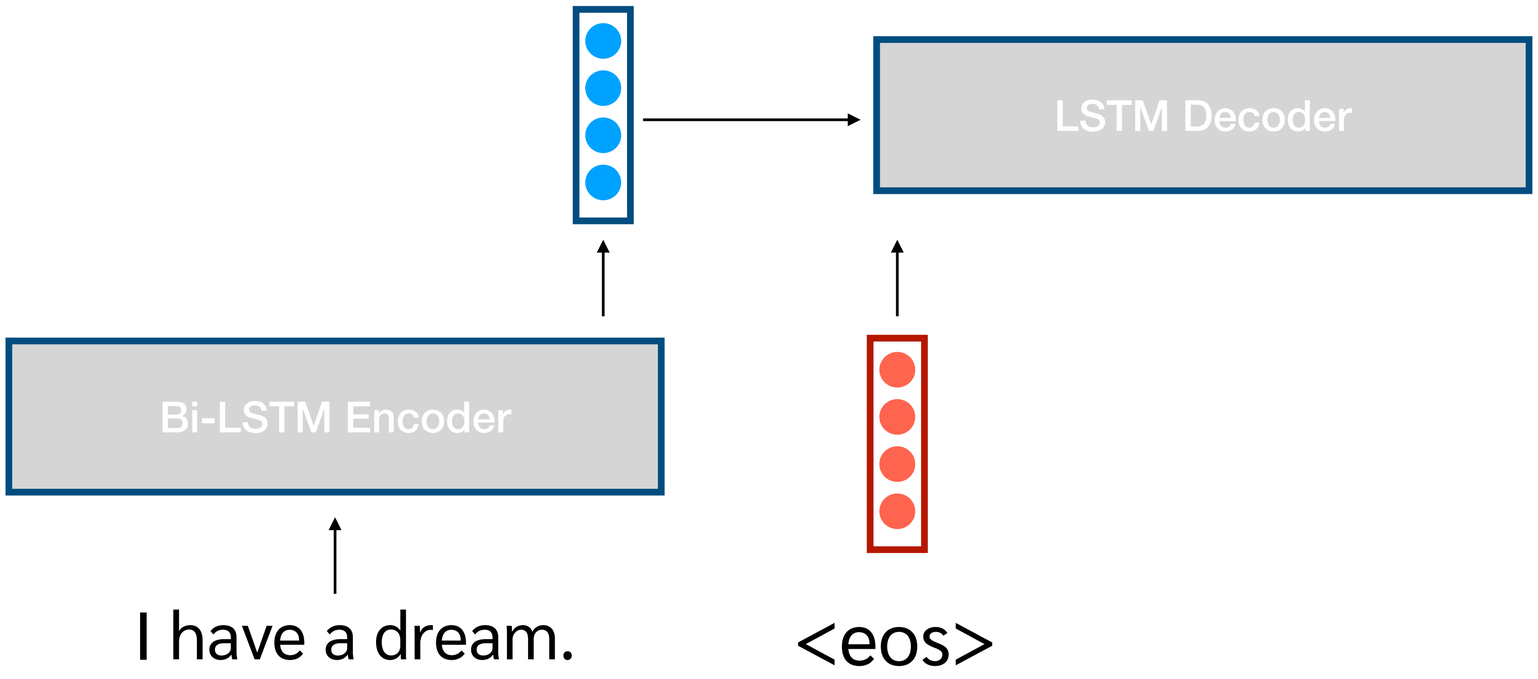
Neural MT with Conditional LM



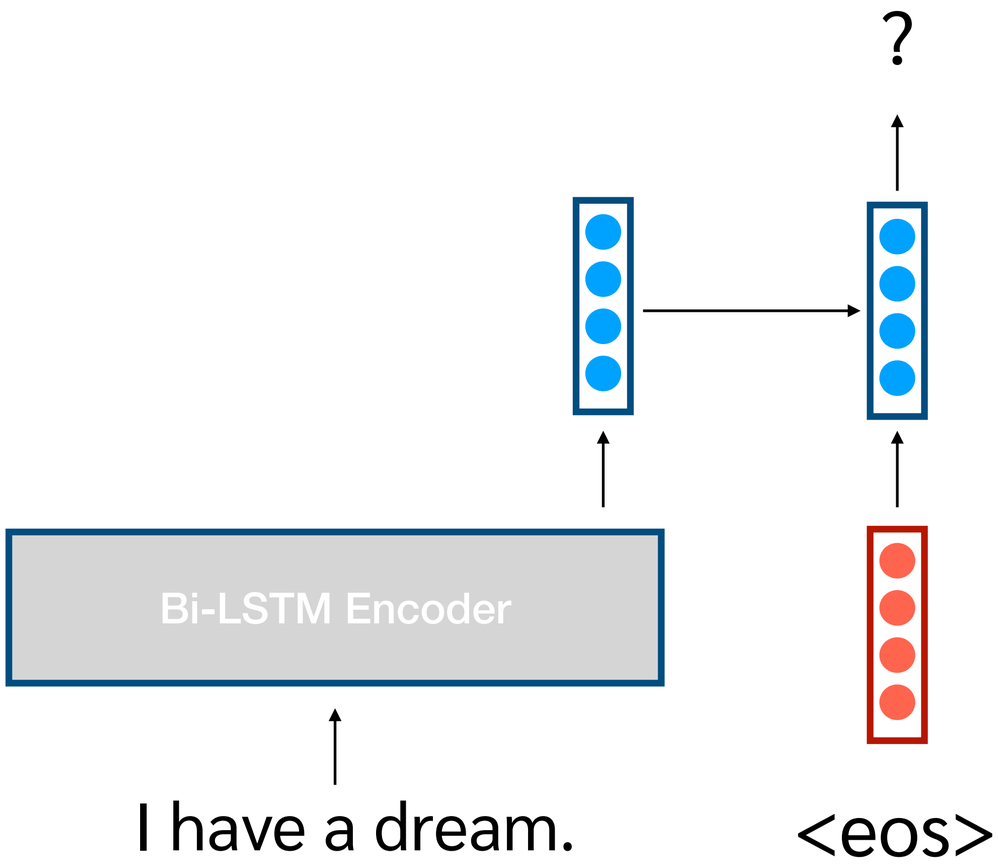
Training Neural MT

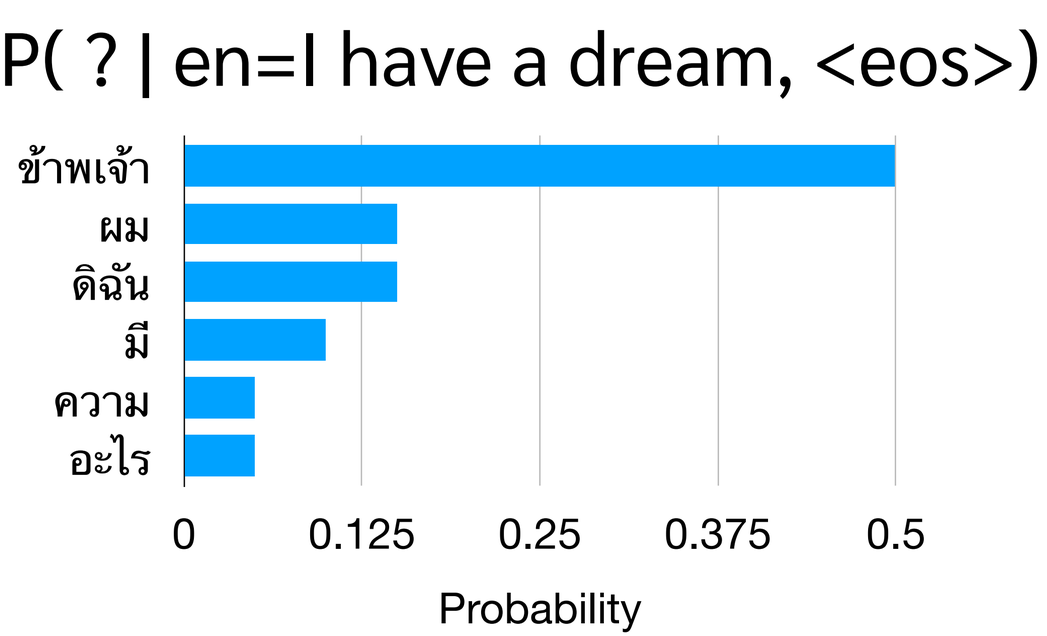


Decoding with Neural MT

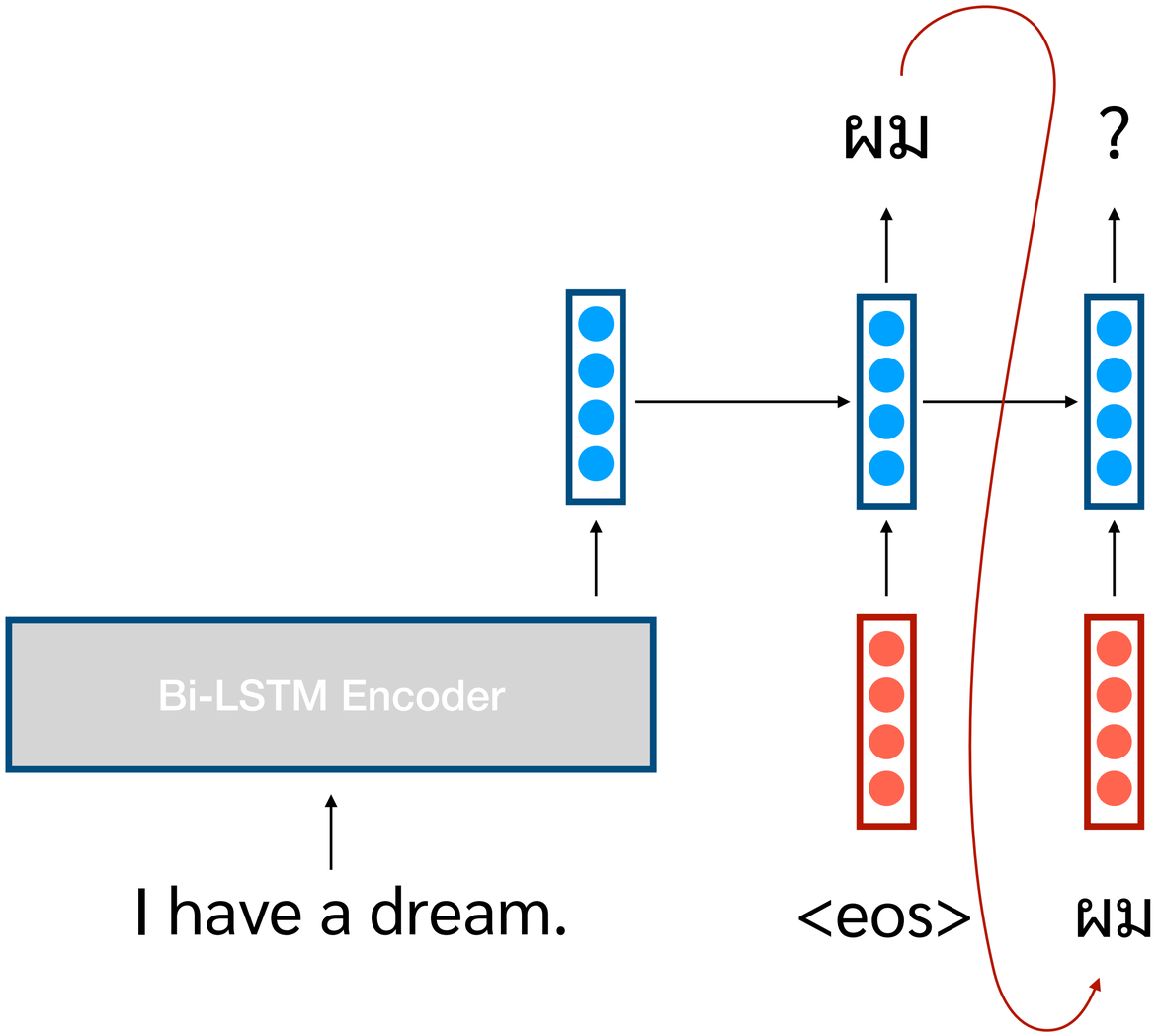


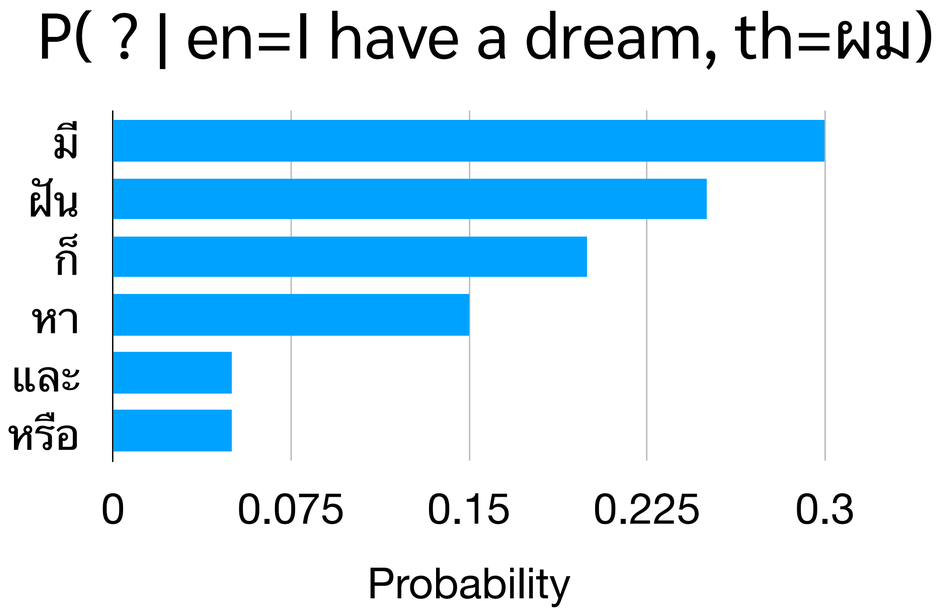
Decoding with Neural MT





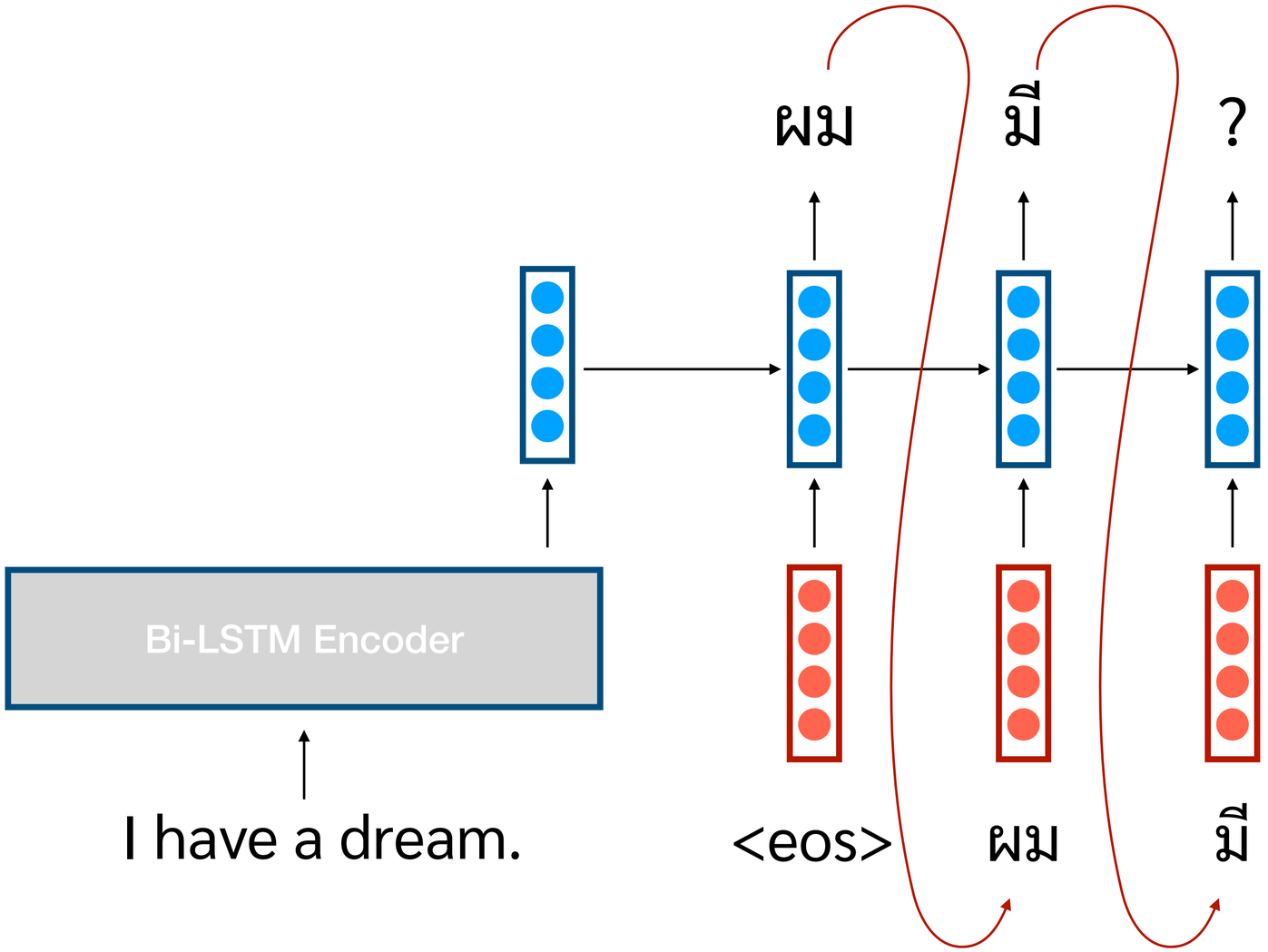
Decoding with Neural MT





ของจริงต้องใช้ Beam search
เก็บ context vector ไว้ใน hypothesis

Decoding with Neural MT

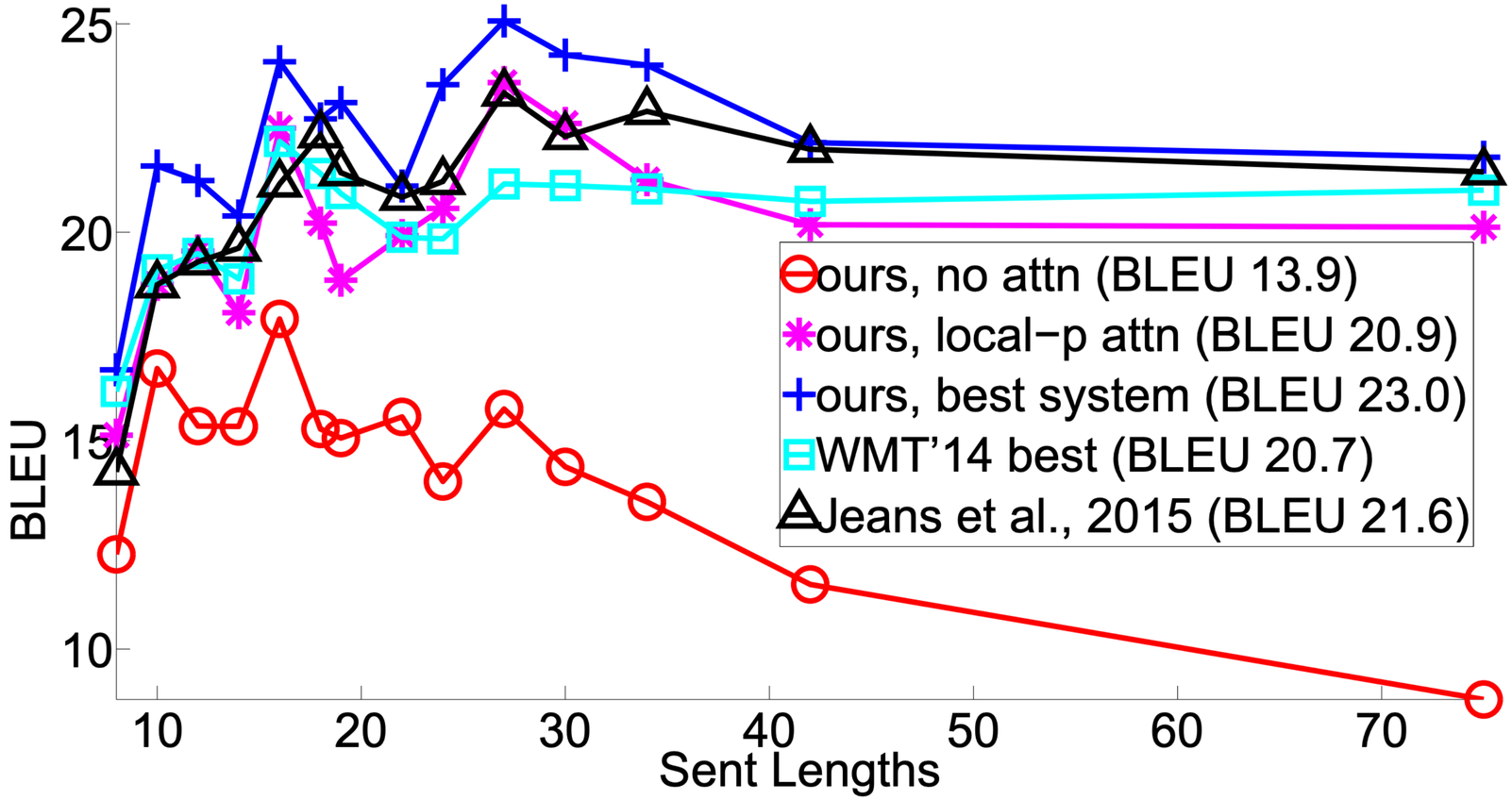


สรุป: Encoder-Decoder Model

- Encoder - เปลี่ยนภาษาต้นฉบับให้เป็น vector
- Decoder - เป็น conditional language model ใช้ vector จาก encoder เป็น feature เพิ่มเติม

Encoder-Decoder Model with Attention

ประโยคยาวแปลแย่



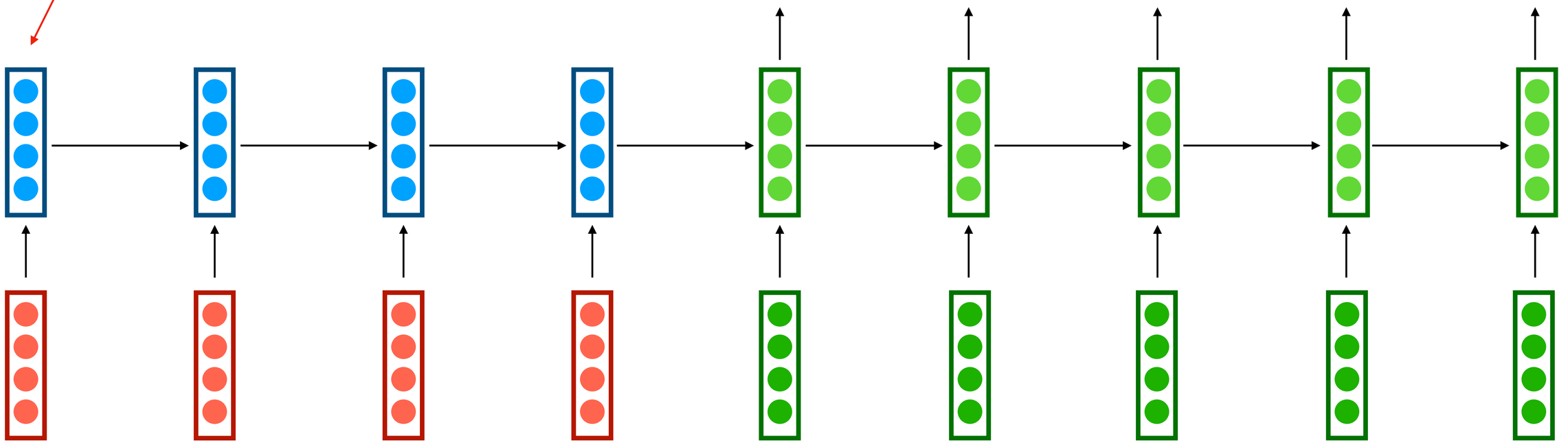
โมเดลเราไม่เหมาะกับประโยคยาว

- I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character.
- ข้าพเจ้ามีความฝันว่าวันหนึ่งลูกทั้งสี่คนของข้าพเจ้าจะอยู่ในประเทศที่คนไม่ตัดสินกันด้วยสีผิว แต่ตัดสินกันด้วยเนื้อแท้ของนิสัยใจคอ

Idea ของ Attention

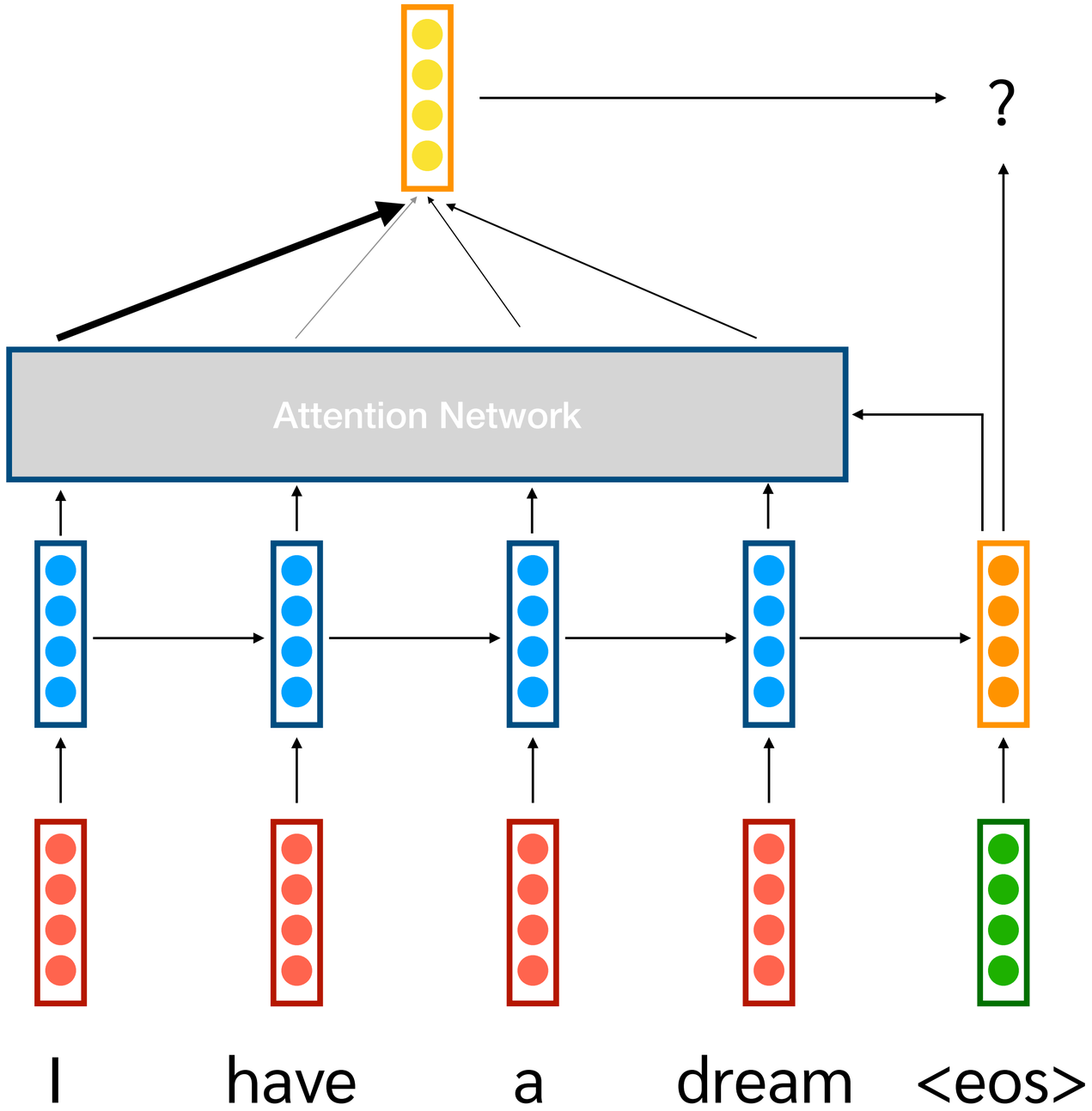
- เราควรแบ่งความสนใจ (attention) กับคำที่มีความสำคัญในการแปลในจุดนั้น

เราต้องการให้คำตรงนี้ถูกยิงไปไกลๆ



I have a dream <eos> ข้าพเจ้า มี ความ ฝัน

Encoder Decoder



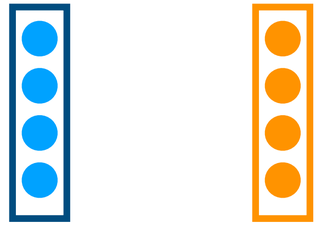
Predict ด้วย

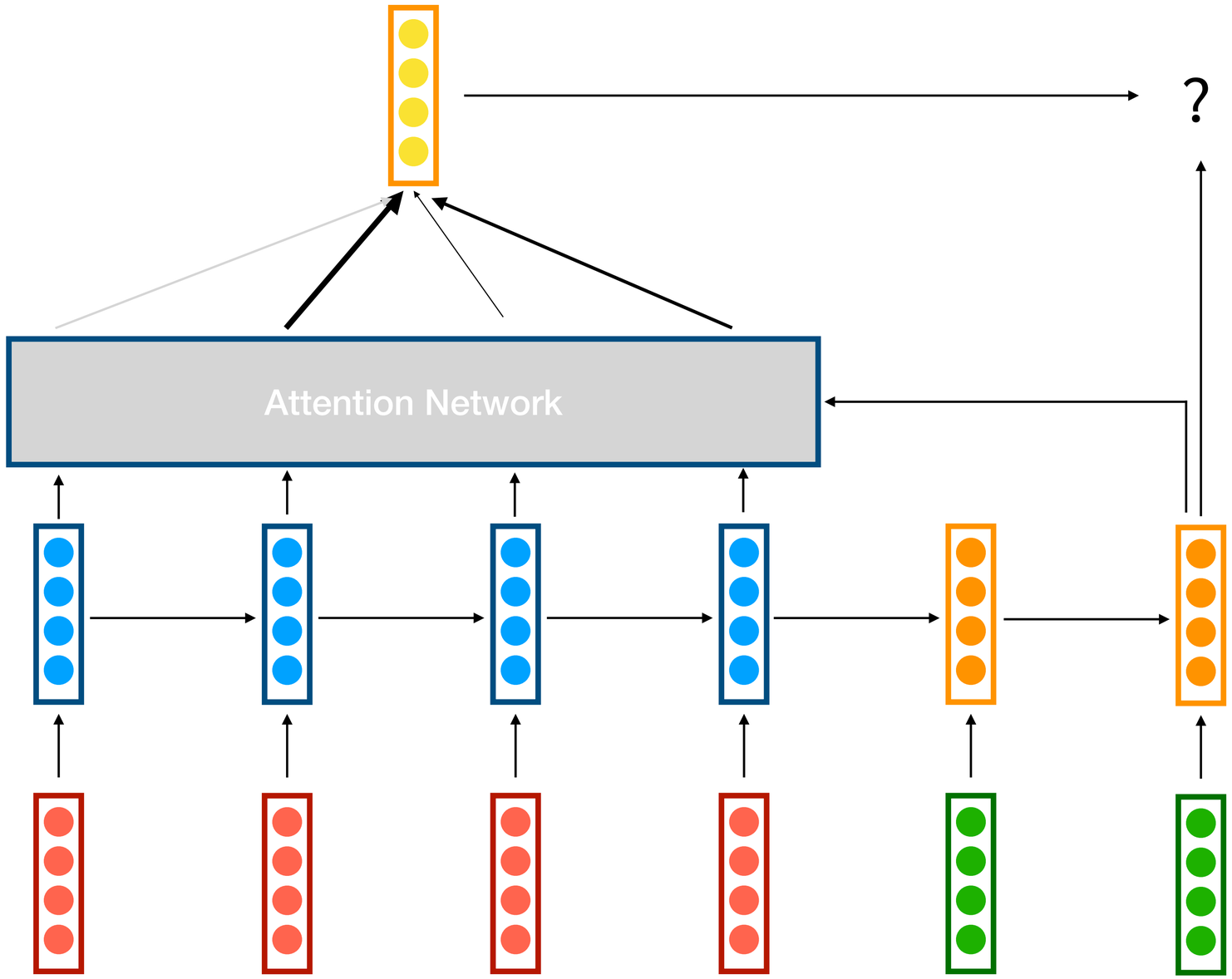
Attention(I , <eos>) = 2.1

Attention(have, <eos>) = 0.2

Attention(a, <eos>) = 0.4

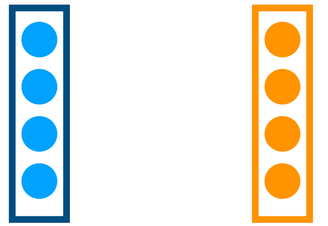
Attention(dream, <eos>) = 0.5



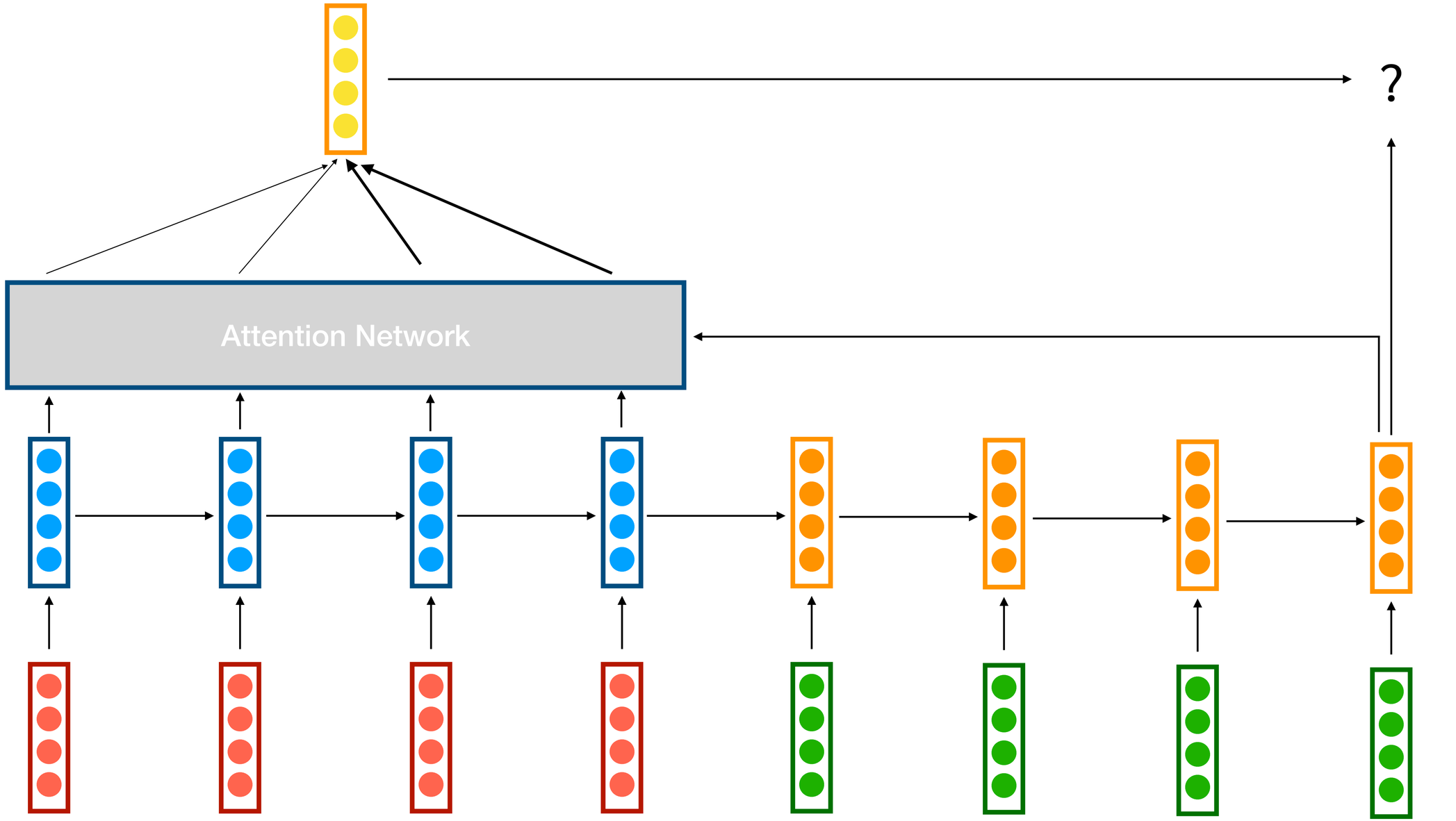


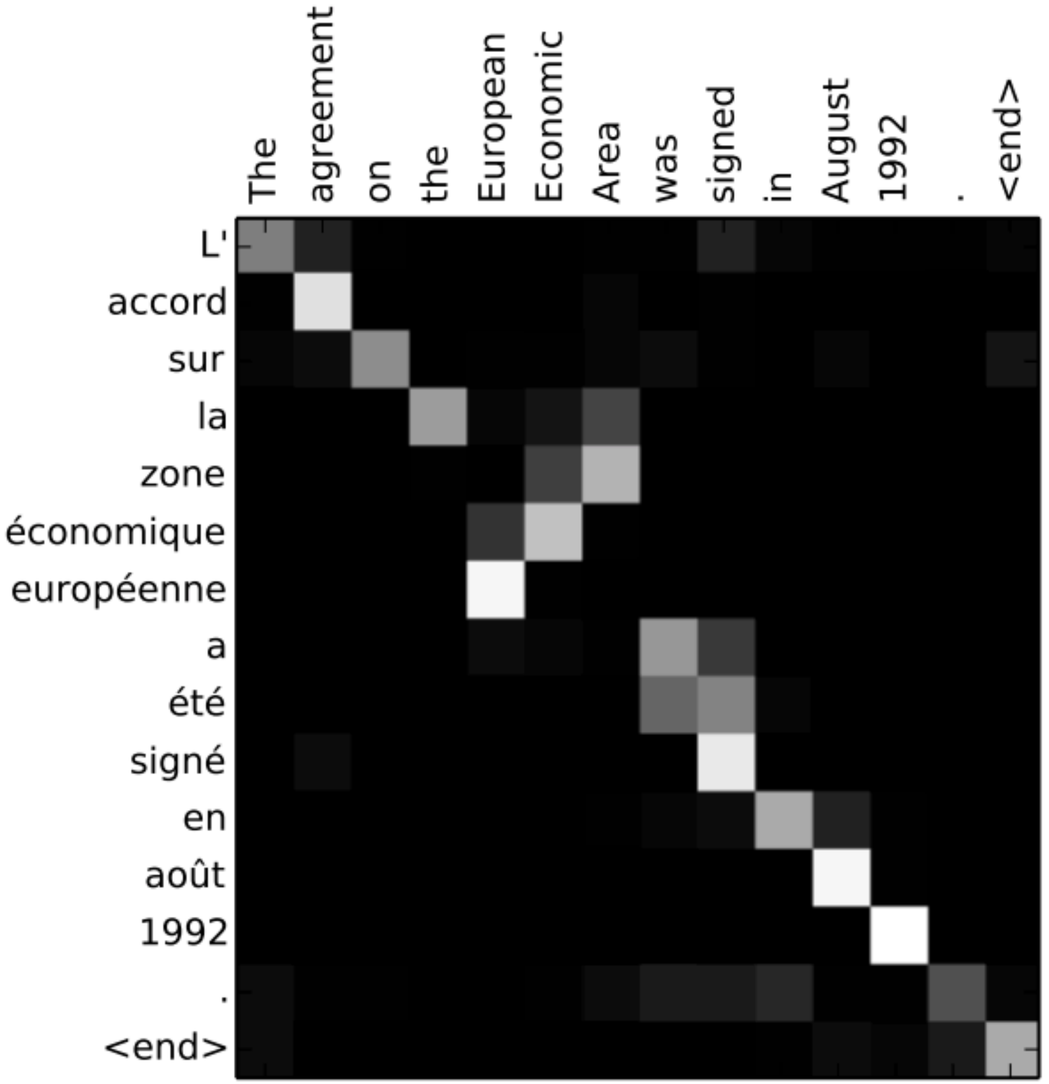
Predict ด้วย

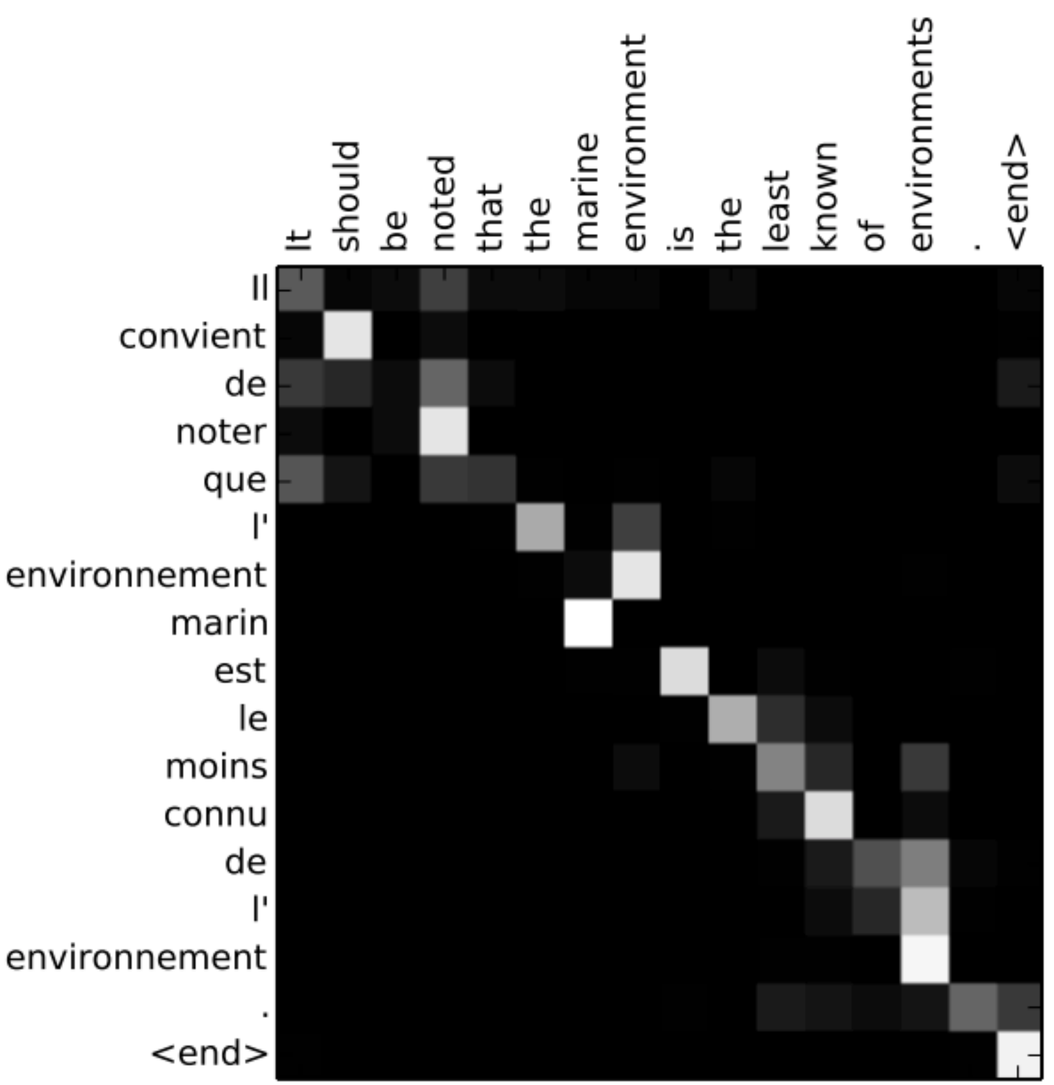
Attention(I , ข้าพเจ้า) = 0.1
Attention(have, ข้าพเจ้า) = 1.8
Attention(a, ข้าพเจ้า) = 0.2
Attention(dream, ข้าพเจ้า) = 0.8



I have a dream <eos> ข้าพเจ้า


?
Attention Network
I
have
a
dream
<eos>
ข้าพเจ้า
มี
ความ



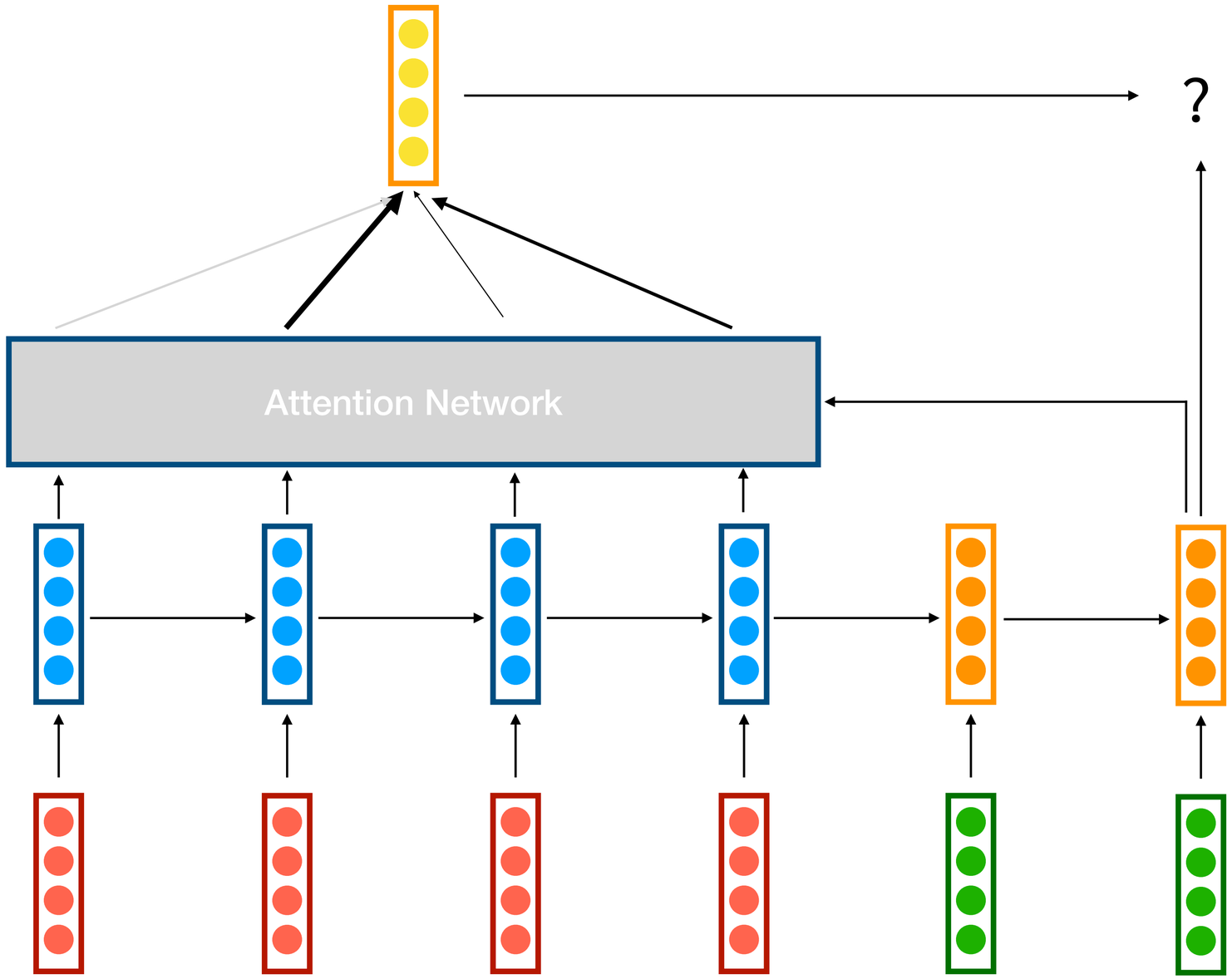


Bahdanau, et al. (2015) Neural Machine Translation by Jointly Learning to Align and Translate

Attention Model

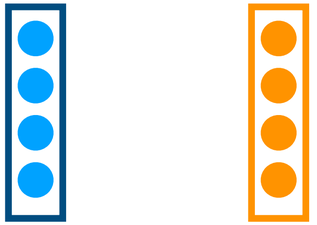
- Context vector (vector ของประโยคต้นฉบับ) เป็น weighted-average ของ hidden activation
- Attention network เป็นตัวคำนวณหาว่าคำไหนจะได้น้ำหนักเยอะน้อยแค่ไหน (คล้ายๆ word alignment)
- Attention model ช่วยเพิ่มความแม่นยำให้กับ MT encoder-decoder model

Attention Network



Predict ด้วย

Attention(I , ข้าพเจ้า) = 0.1
Attention(have, ข้าพเจ้า) = 1.8
Attention(a, ข้าพเจ้า) = 0.2
Attention(dream, ข้าพเจ้า) = 0.9



I have a dream <eos> ข้าพเจ้า

Softmax คะแนนดิบเป็น attention

Attention(I , ข้าพเจ้า) = 0.1
Attention(have, ข้าพเจ้า) = 1.8
Attention(a, ข้าพเจ้า) = 0.2
Attention(dream, ข้าพเจ้า) = 0.9

```
In [5]: attention_score = np.array([0.1, 1.8, 0.2, 0.9])

In [6]: attention = np.exp(attention_score)/ np.exp(attention_score).sum()

In [7]: attention
Out[7]: array([0.10199233, 0.55830063, 0.11271895, 0.2269881 ])
```

Softmax คะแนนดิบเป็น attention

Attention(I , ข้าพเจ้า) = 0.1
Attention(have, ข้าพเจ้า) = 1.8
Attention(a, ข้าพเจ้า) = 0.2
Attention(dream, ข้าพเจ้า) = 0.9

คะแนนดิบพวกนี้คำนวณยังไง?

```
In [5]: attention_score = np.array([0.1, 1.8, 0.2, 0.9])

In [6]: attention = np.exp(attention_score)/ np.exp(attention_score).sum()

In [7]: attention
Out[7]: array([0.10199233, 0.55830063, 0.11271895, 0.2269881 ])
```

ตัวอย่าง attention function

Attention(I , ข้าพเจ้า) = 1
Attention(have, ข้าพเจ้า) = 1
Attention(a, ข้าพเจ้า) = 1
Attention(dream, ข้าพเจ้า) = 1

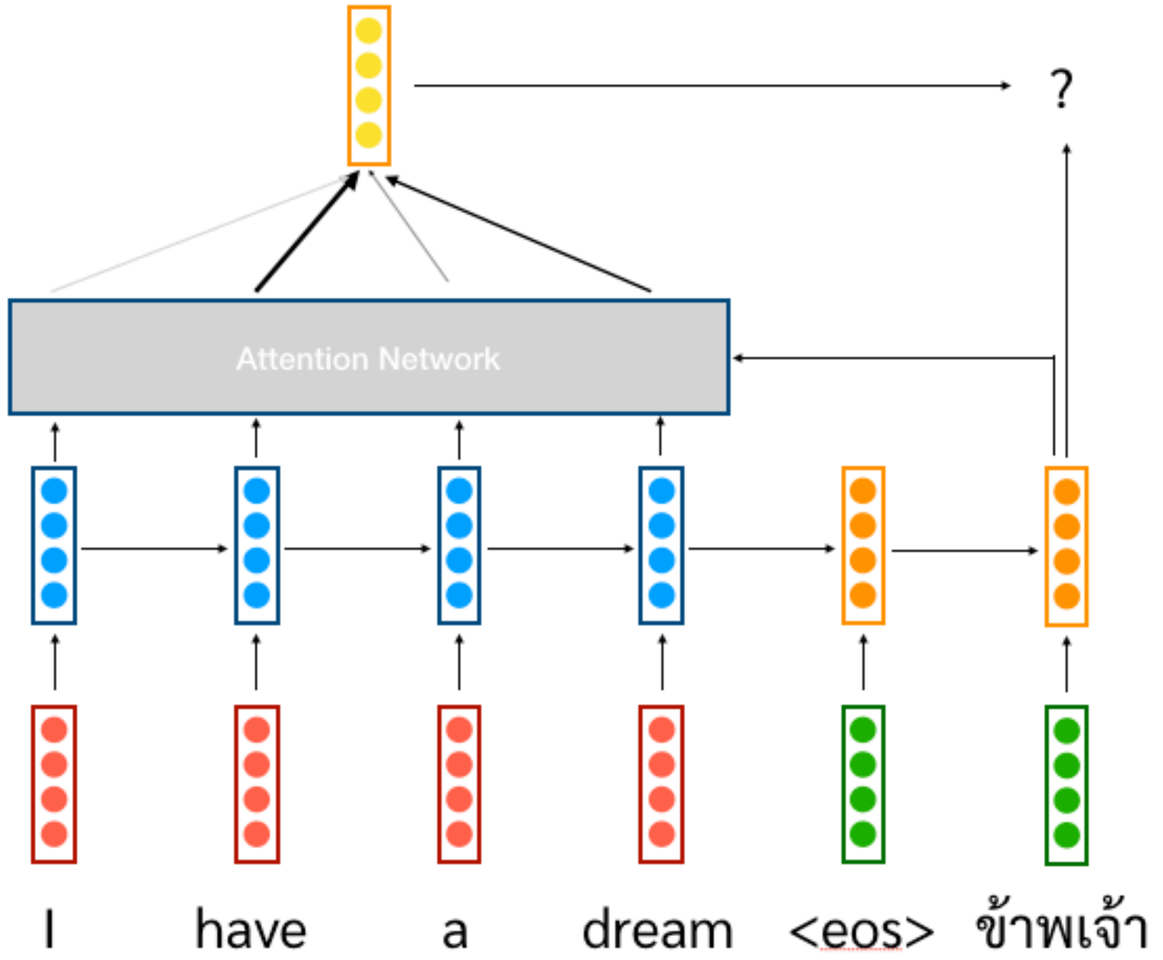
$$a(x_1, s_2) = 1$$
$$a(x_2, s_2) = 1$$
$$a(x_3, s_2) = 1$$
$$a(x_4, s_2) = 1$$

$$\alpha(x_1, s_2) = 1/4$$

Attention Network

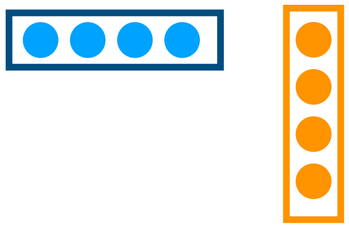
$$c_t = \sum_{i}^{N} \alpha(x_i, s_t) x_i$$

$$\alpha(x_i, s_t) = \frac{\exp(a(x_i, s_t))}{\sum_{j}^{N} \exp(a(x_j, s_t))}$$

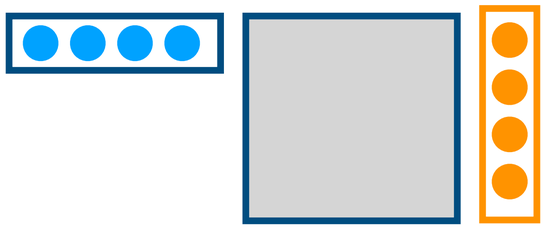


ตัวอย่าง attention function

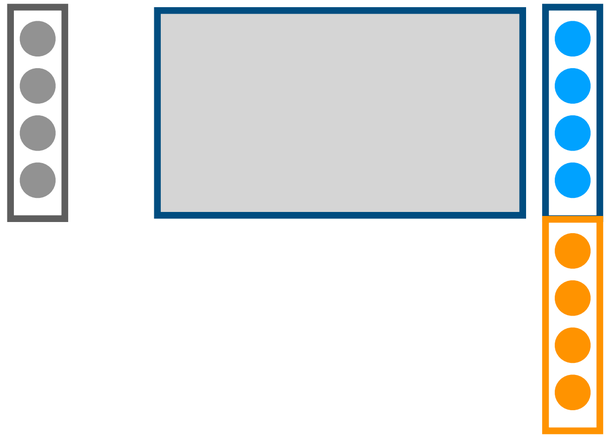
$$a(x_i, s_j) = x_i^T \cdot s_j$$



$$a(x_i, s_j) = x_i^T \cdot W_a \cdot s_j$$



$$a(x_i, s_j) = v_a^T \cdot \tanh(W_a \cdot [x_i; s_j])$$



สรุป

- Attention ช่วยให้รวบ vector หลายๆ อันเข้ามาเป็นอันเดียวเพื่อใช้เป็น feature ได้
- Attention score คำนวณได้หลายแบบ แต่เน้นการคำนวณค่าความเข้ากันได้ของสอง vector