Sequence Models & Attention Mechanism

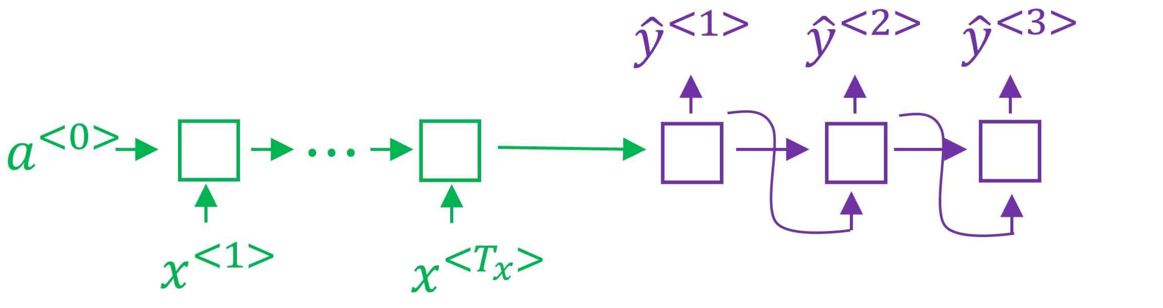
Basic Model

[Sutskever et al., 2014. Sequence to sequence learning with neural networks]

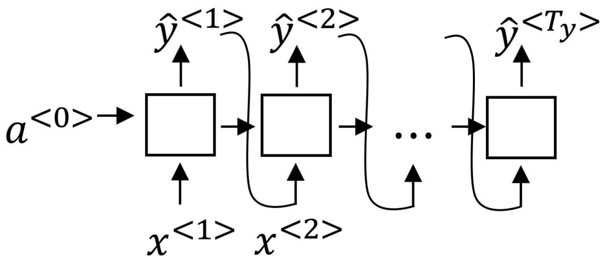
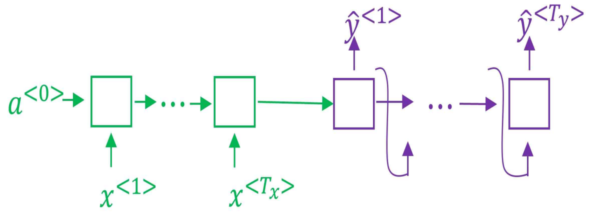
[Cho et al., 2014. Learning phrase representations
using RNN encoder-decoder for statistical machine translation]

$x^{<1>}$ Jane	$x^{<2>}$ visite	$x^{<3>}$ l'Afrique	$x^{<4>}$ en	$x^{<5>}$ septembre.

→	Jane $y^{<1>}$	is $y^{<2>}$	visiting $y^{<3>}$	Africa $y^{<4>}$	in $y^{<5>}$	September. $y^{<6>}$



Encoder	Decoder

Language Model	Machine Translation
 $\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<T_y>}$ $a^{<0>}$ $x^{<1>}$ $x^{<2>}$	 $\hat{y}^{<1>}$ $\hat{y}^{<T_y>}$ $a^{<0>}$ $x^{<1>}$ $x^{<T_x>}$
$P(y^{<1>}, \cdots, y^{<Ty>})$	"Conditional Language Model" $P(y^{<1>}, \cdots, y^{<Ty>} \vert x^{<1>}, \cdots, x^{<Ty>})$

Finding the most likely translation

$P(y^{<1>}, \cdots, y^{<Ty>} \vert x)$, $y^{<t>}$: English, x: English
Jane visite l'Afrique en septembre. → Jane is visiting Africa in September. → Jane is going to be visiting Africa in September. → In September, Jane will visit Africa. → Her African friend welcomed Jane in September,
$\underset{y^{<1>}, \cdots, y^{<T_y>}}{argmax} \; P(y^{<1>}, \cdots, y^{<T_y>} \vert x)$
Random Sampling을 하면 위 예시들처럼 어색할 수 있기 때문에 Random Sampling 하지

않는다. 그래서 조건부 확률을 극대화하는 영어 문장을 찾는 것이다.

Why not a greedy search?

greedy search는 가장 높은 확률의 것을 고른다. 즉 가장 높은 확률의 $y^{<1>}$을 고르고 그 다음으로 높은 $y^{<1>}$를 고르고 그다음 $y^{<1>}$를 고르는 그런 방식이지만 원하는 것은 전체 단어 y의 joint probability를 극대화하는 것이다.

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

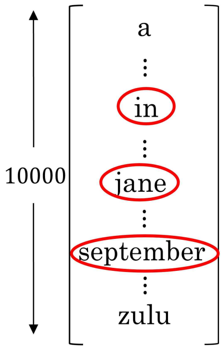
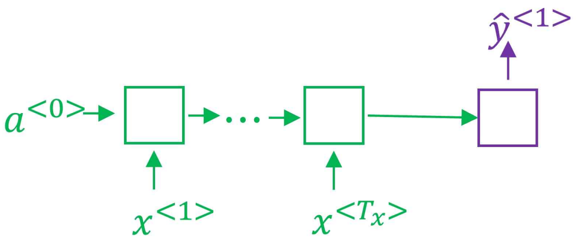
$$P(Jane\,is\,going|x) > P(Jane\,is\,visiting|x)$$

영어 문장은 매우 길어 최댓값을 찾는 데 오랜 시간이 걸리므로 가장 흔한 방법은 approximate search algorithm을 사용하는 것이다. 최댓값을 찾을 것이라 보장하진 못하지만, 충분히 의미있다.

Beam search algorithm: 가장 좋은 문장 찾기

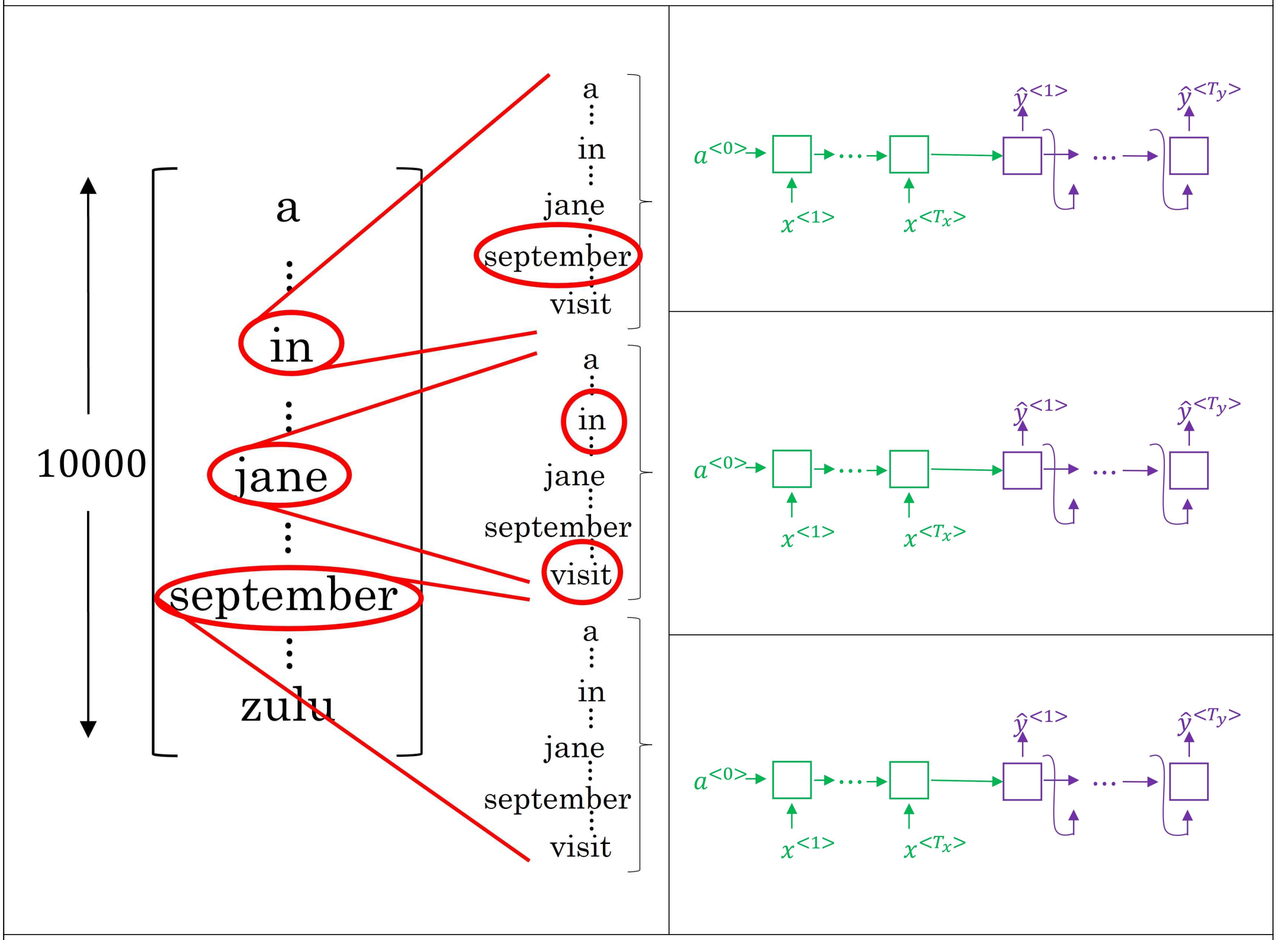
B: beam width (cf. if B = 1 : greedy search)

B = 3

Step 1	
1만 가지의 가능한 출력 중에서 가장 확률이 높은 3가지 단어를 모두 기억한다.	
	
$P(y^{<1>} \mid x)$	

Step 2

step 1에서 기억한 단어들에 대해 각각 다음 단어로 올 가장 가능성있는 단어를 찾는다.

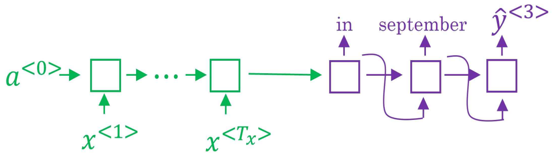
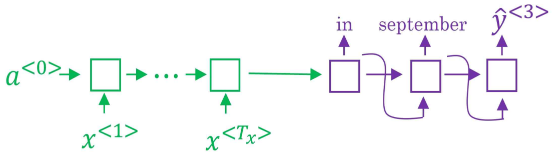
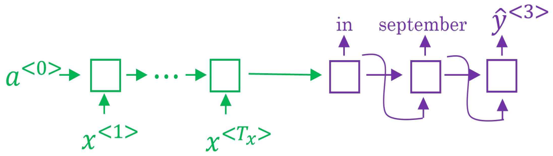


$$P(y^{<1>}, y^{<2>}|x) = P(y^{<1>}|x)P(y^{<2>}|x, y^{<1>})$$

다음으로 $y^{<1>}$과 $y^{<2>}$의 단어 쌍 중에서 가장 어울리는 것을 찾는다. 단순히 가장 가능성 있는 $y^{<2>}$를 찾는 것이 아니라 조건부 확률이 높은 것을 찾는다.

in, jane, septemer에 대한 단어 쌍(총 30,000쌍)의 조건부 확률을 찾고 전체 쌍 중 가장 확률이 높은 쌍 B=3개(in september, dane is, jane visit)를 기억한다. 그리고 다음 단계로 넘어간다.

So just don't instantiate 30,000 copies of the network or three copies of the network to very quickly evaluate all 10,000 possible outputs at that softmax output say for $y^{<2>}$.

Step 3	
in september	 $a^{<0>}$ $x^{<1>}$ $x^{<T_x>}$ in september $\hat{y}^{<3>}$
jane is	 $a^{<0>}$ $x^{<1>}$ $x^{<T_x>}$ in september $\hat{y}^{<3>}$
jane visits	 $a^{<0>}$ $x^{<1>}$ $x^{<T_x>}$ in september $\hat{y}^{<3>}$

$P(y^{<1>}, y^{<2>} \mid x)$	jane visits africa in september. ⟨EOS⟩

Refinements to beam search

Length normalization

$$\underset{y}{argmax} \prod_{t=1}^{T_y} P(y^{<t>} \mid x, y^{<1>}, \cdots, y^{<t-1>}),$$

$$P(y^{<1>}, \cdots, y^{<T_y>} \mid x) = P(y^{<1>} \mid x) P(y^{<2>} \mid x, y^{<1>}) \cdots P(y^{<T_y>} \mid x, y^{<1>}, \cdots, y^{<T_y - 1>})$$

실제로 계산하면 1보다 작은 소수로 컴퓨터가 계산하기에 너무 작아 log를 사용한다.

$$\underset{y}{argmax} \prod_{t=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \cdots, y^{<t-1>})$$

$$\frac{1}{T_y^{\alpha}} \underset{y}{argmax} \prod_{t=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \cdots, y^{<t-1>})$$

Beam search discusstion

large B : better result but slower

small B : worse result but faster

Unlike exact search algorithms like BFS(Breadth First Search) or DFS(Depth First Search), Beam Search runs faster but is not guaranteed to fine exact maximum for $\underset{y}{argmax} P(y \mid x)$.

Error analysis on beam search

Human: Jane visits Africa in September.(y^*) Algorithm: Jane visited Africa last September.($\hat{y}$)
Case 1: $P(y^*\|x) > P(\hat{y}\|x)$ Beam search chose $\hat{y}$. But y^* attains higher $P(y\|x)$. Conclusion: Beam search is at fault.
Case 2: $P(y^*\|x) \le P(\hat{y}\|x)$ y^* is a better translation than $\hat{y}$. But RNN predicted $P(y^*\|x) < P(\hat{y}\|x)$. Conclusion: RNN model is at fault.

Figure out what faction of errors are "due to" beam seach vs. RNN model

Bleu score

[Papineni et. al., 2002. Bleu:
A method for automatic evaluation of machine translation]
bilingual evaluation
the BLEU score is an understudy, could be a substitute for having humans evaluate every output of a machine translation system.

Evaluating machine translation

French: Le chat est sur le tapis	
Reference 1: The cat is on the mat. Reference 2: There is a cat on the mat.	
MT ouput: the the the the the the the.	
Precision: $\frac{7}{7}$	Modified Precision: $\frac{count\_clip(the)}{count(the)} = \frac{2}{7}$

Bleu score on bigrams

Example:	Reference 1: The cat is on the mat.
	Reference 2: There is a cat on the mat.
	MT ouput: the the the the the the the.

	Count	Count_clip
the cat	2	1

cat the	1	1
cat on	1	1
on the	1	1
the mat	1	1
Bleu score on biagrams: $\frac{4}{6}$		

Bleu score on unigrams

Example:	Reference 1: The cat is on the mat.
	Reference 2: There is a cat on the mat.
	MT ouput: the the the the the the the.

$p_1 = \frac{\sum_{unigram \in \hat{y}} count_{clip}(unigram)}{\sum_{unigram \in \hat{y}} count(unigram)}$	$p_n = \frac{\sum_{ngram \in \hat{y}} count_{clip}(ngram)}{\sum_{ngram \in \hat{y}} count(ngram)}$

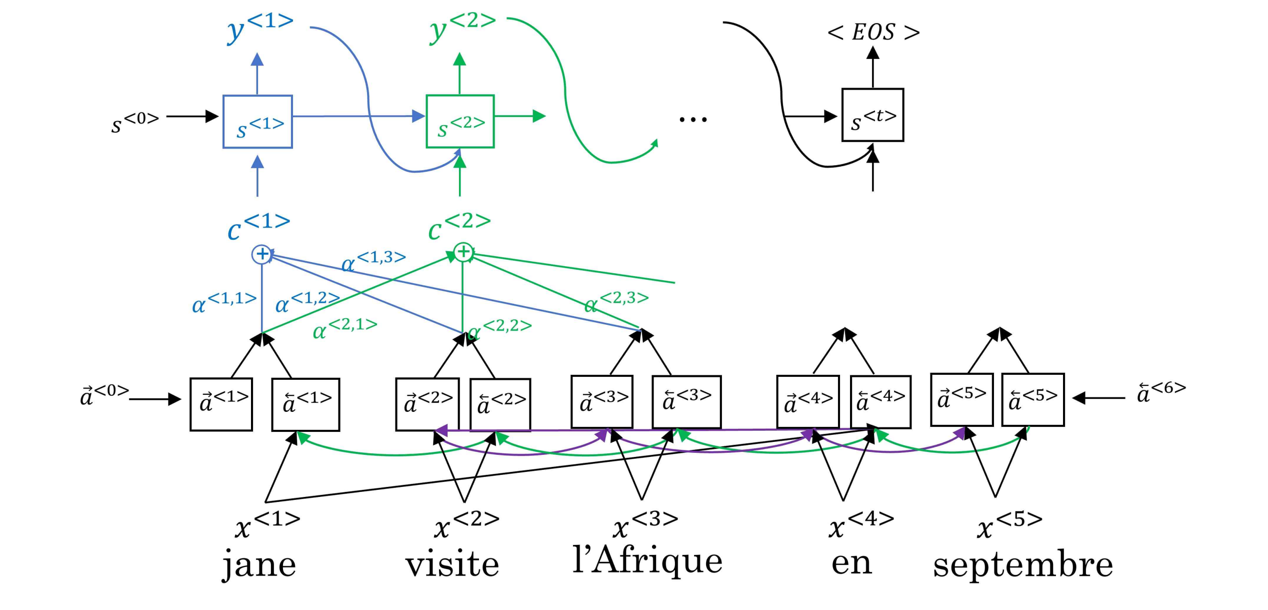
Bleu details

p_n = Bleu score on n-grams only Combined Bleu score: $BPexp\left(\frac{1}{n}\sum_{i=1}^{n} p_i\right)$
BP: brevity penalty $BP = \begin{cases} 1 & \text{if } MT\_output\_length > reference\_output\_length \\ \exp\left(1 - \frac{reference\_ouput\_length}{MT\_output\_length}\right) & otherwise \end{cases}$

Attention model

[Bahdanauet. al., 2014.

Neural machine translation by jointly learning to align and translate]



$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$

$\sum_{t'} \alpha^{<t,t'>} = 1$

$\alpha^{<t,t'>}$ = the amount of "attention" $y^{<t>}$ should pay to $a^{<t'>}$

$c^{<t>} = \sum_{t'} \alpha^{<t,t'>} a^{<t'>}$

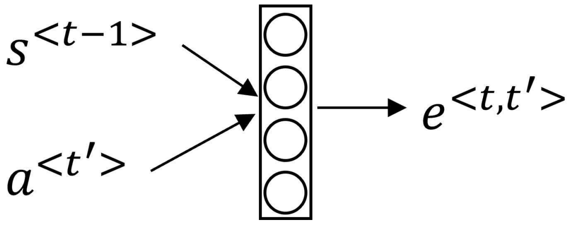
Computing attention $\alpha^{<t,t'>}$

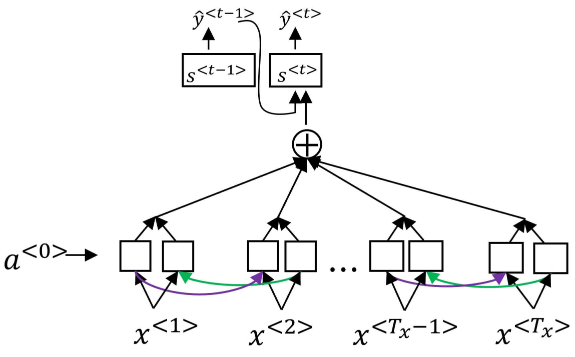
[Xu et. al., 2015. Show, attend and tell:

Neural image caption generation with visual attention]

$\alpha^{<t,t'>}$ = the amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{Tx} \exp(e^{<t,t'>})}$$

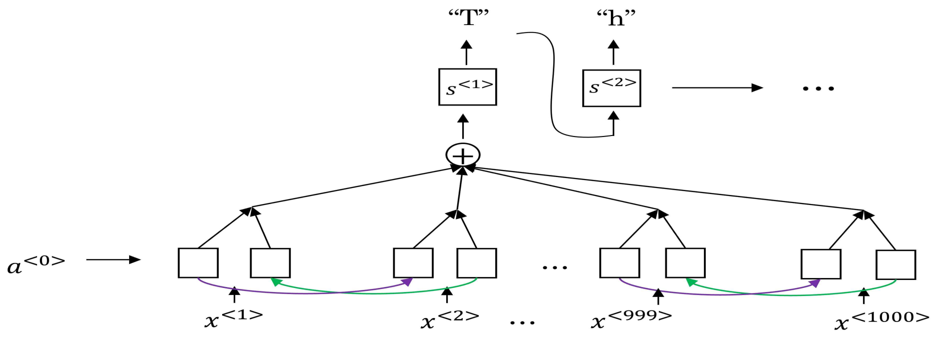




Speech recognition

Speech recognition problem: phonemes

Attention model for speech recognition

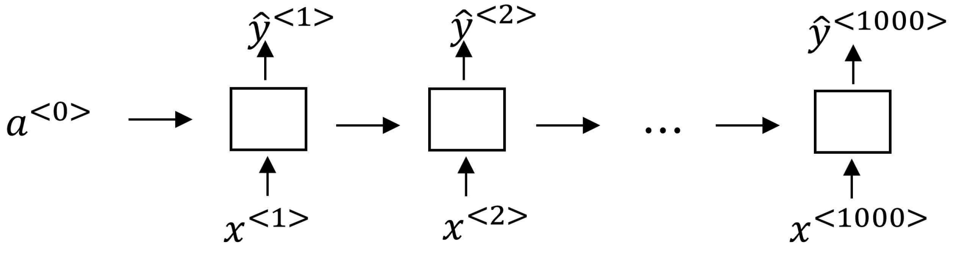


CTC cost for speech recognition

(Connectionist temporal classification)

[Graves et al., 2006. Connectionist Temporal Classification:
Labeling unsegmented sequence data with recurrent neural networks]

"the quick brown fox"



t t t _ h _ e e e _ _ _ ⎵ _ _ _ g g g _ _

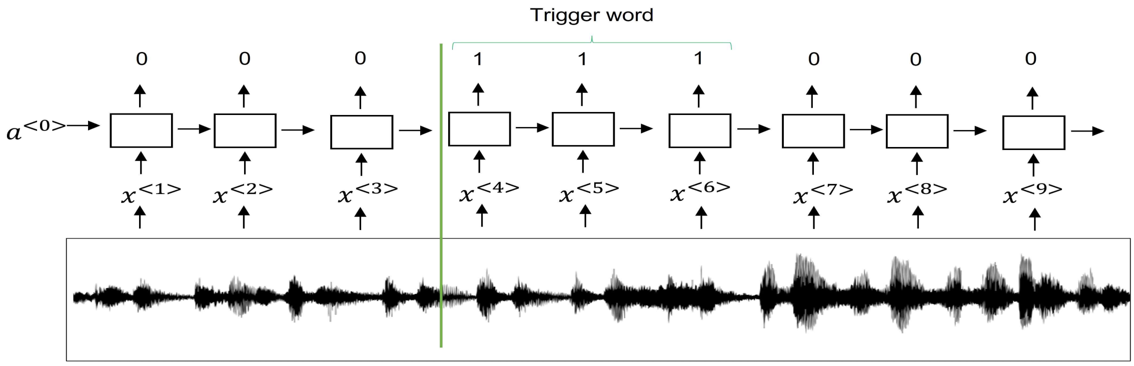
_ : "blank“

⎵: "sparse"

Basic rule: collapse repeated characters not separated by "blank"

Trigger word detection

"Hi Bixby", "Hey Clova", "Hey Kakao"


Trigger word
0
0
0
1
1
1
0
0
0
$a^{<0>}$
$x^{<1>}$
$x^{<2>}$
$x^{<3>}$
$x^{<4>}$
$x^{<5>}$
$x^{<6>}$
$x^{<7>}$
$x^{<8>}$
$x^{<9>}$