Language Model

(spelling correction, speech recognition, machine translation)

김웅희

언어 모델이란?

- 단어 시퀀스에 확률을 할당하는 모델
  - 언어라는 현상을 모델링하고자 단어 시퀀스에 확률을 할당하는 모델
  - 영어 : $p_1 > p_2 > p_3 > p_4$
  - 확률을 통해 보다 적절한 문장을 판단

P_1 = P(”a quick brown dog”)
P_2 = P(“dog quick a brown”)
P_3 = P(“un chien quick brown”)
P_4 = P(“un chien brun rapide”)

  - 기계가 자연스러운 문장을 만들어내도록 하는 것이 언어 모델의 일

언어 모델의 종류

1. 통계적 언어 모델 (Statistical Language Model, SLM)
   - 조건부 확률을 통해 다음 단어를 예측
   - n-gram 언어 모델 등

2. 피드 포워드 신경망 언어 모델 (Neural Network Language Model, NNLM)
   - 뉴럴 네트워크를 통해 다음 단어를 예측
   - RNNLM, LSTM-LM 등

SLM vs NNLM

- 통계적 언어모델은 각 단어에 대한 예측 확률을 곱함

$$P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \ldots, w_{n-1})$$

$P(\text{An adorable little boy is spreading smiles}) =$
$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \times P(\text{boy}|\text{An adorable little}) \times P(\text{is}|\text{An adorable little boy})$
$\times P(\text{spreading}|\text{An adorable little boy is}) \times P(\text{smiles}|\text{An adorable little boy is spreading})$

SLM vs NNLM

- 통계적 언어모델은 각 단어에 대한 예측 확률을 곱함

$$P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \ldots, w_{n-1})$$

$P(\text{An adorable little boy is spreading smiles}) =$
$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \times P(\text{boy}|\text{An adorable little}) \times P(\text{is}|\text{An adorable little boy})$
$\times P(\text{spreading}|\text{An adorable little boy is}) \times P(\text{smiles}|\text{An adorable little boy is spreading})$

~~An adorable little~~ boy is spreading ?

무시됨!

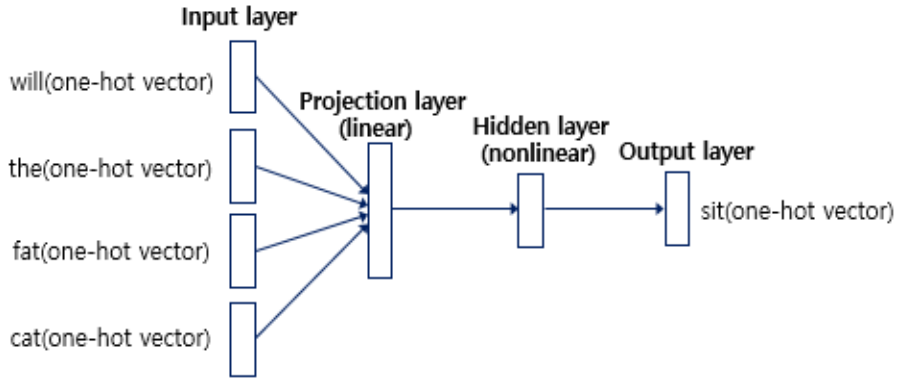
4-gram을 의미 ← n-1개의 단어

$$P(w|\text{boy is spreading}) = \frac{\text{count}(\text{boy is spreading } w)}{\text{count}(\text{boy is spreading})}$$

- 장점 : 직관적임. 구현이 단순.
- 단점 : 의미적 유사도 파악 불가. 희소문제

해당 단어가 훈련 데이터 안에 없을시 고려되지 않는 현상

SLM vs NNLM



단어	원-핫 벡터
what	[1, 0, 0, 0, 0, 0, 0]
will	[0, 1, 0, 0, 0, 0, 0]
the	[0, 0, 1, 0, 0, 0, 0]
fat	[0, 0, 0, 1, 0, 0, 0]
cat	[0, 0, 0, 0, 1, 0, 0]
sit	[0, 0, 0, 0, 0, 1, 0]
on	[0, 0, 0, 0, 0, 0, 1]

$x_{fat} \times W_{V \times M} = e_{fat}$

0	0	0	1	0	0	0

×

0.5	2.1	1.9	1.5	0.8
0.8	1.2	2.8	1.8	2.1
0.1	0.8	1.2	0.9	0.7
2.1	1.8	1.5	1.7	2.7

=

2.1	1.8	1.5	1.7	2.7

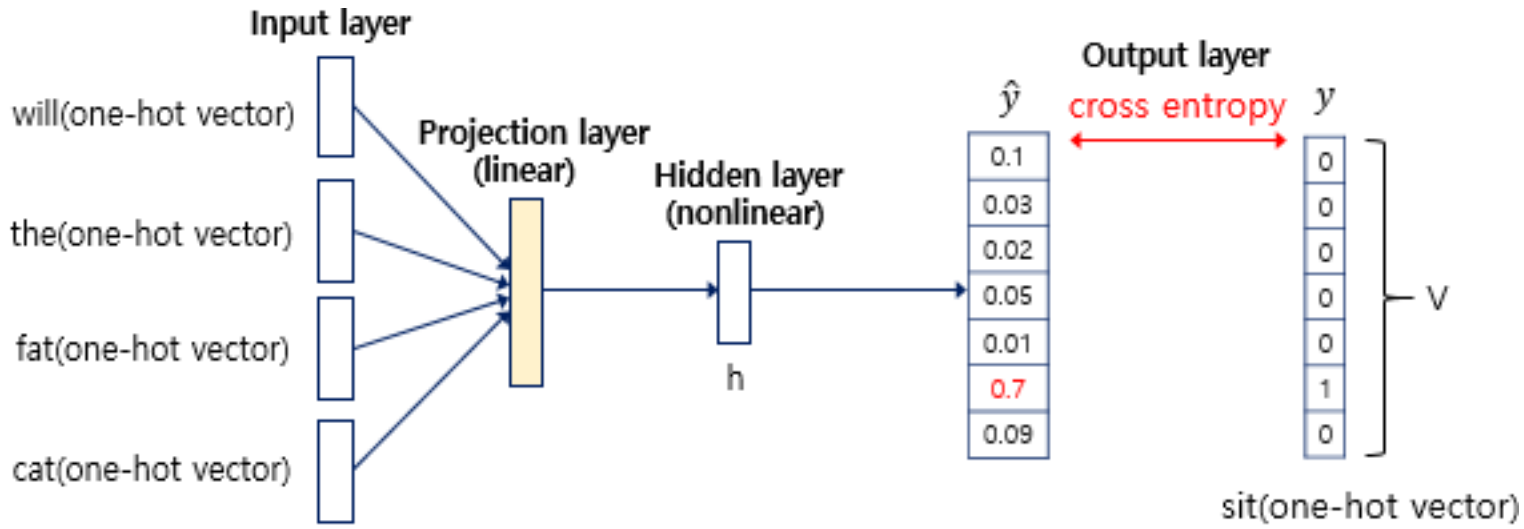
lookup table

- window size : 다음의 단어를 예측하기 위해 사용되는 앞의 n개의 단어
- 입력은 원-핫 벡터
- lookup table : Projection layer를 생성하기 위해 사용되는 테이블. projection layer의 크기가 m이고, 원-핫 벡터의 차원이 7이면 가중치 행렬 W는 7 X m이 됨

투사층 : $p^{layer} = (lookup(x_{t-n}); \ldots; lookup(x_{t-2}); lookup(x_{t-1})) = (e_{t-n}; \ldots; e_{t-2}; e_{t-1})$

SLM vs NNLM

출력층 : $\hat{y} = softmax(W_y h^{layer} + b_y)$



은닉층 : $h^{layer} = tanh(W_h p^{layer} + b_h)$

- 장점 : 밀집 벡터를 이용하여 단어의 유사도를 표현할 수 있고, 모든 n-gram을 저장하지 않아도 되기 때문에 저장 공간의 이점을 가짐
- 단점 : 여전히 정해진 n개의 단어만을 참조

Spelling Correction이란?

1. 단어가 아닌 스펠링 에러를 탐지하고 보정하거나
   - ex) acress the river -> across the river

2. 문맥에 맞지 않는 실제 단어를 탐지하고 보정하는 것
   - ex) a peace of cake -> a piece of cake

- Spelling Correction에서는 Noisy Channel Model을 사용

Spelling Correction – Noisy Channel

- Noisy Channel Model

$$\hat{w} = argmax_{w \in V} P(w|x)$$

- Bayesian inference의 일종으로 잘못 스펠링된 단어인 관찰 x를 보고, 원래의 단어 w를 찾는 것
- 사전에서 나타나는 모든 단어들에서, 우변식 P(w|x)를 최대화시키는 특정 단어를 찾는 것이 Noisy Channel의 목적

Spelling Correction – Noisy Channel

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

$$\hat{w} = argmax_{w \in V} P(w|x) \quad \Rightarrow \quad \hat{w} = argmax_{w \in V} \frac{P(x|w)P(w)}{P(x)}$$

$$\hat{w} = argmax_{w \in V} P(x|w)P(w)$$

Channel Model

prior

Spelling Correction – Noisy Channel

Error	Correction	Transformation: Correct Letter	Transformation: Error Letter	Transformation: Position (Letter #)	Transformation: Type
acress	actress	t	—	2	deletion
acress	cress	—	a	0	insertion
acress	caress	ca	ac	0	transposition
acress	access	c	r	2	substitution
acress	across	o	e	3	substitution
acress	acres	—	s	5	insertion
acress	acres	—	s	4	insertion

Figure 5.3 Candidate corrections for the misspelling *acress* and the transformations that would have produced the error (after Kernighan et al. (1990)). "—" represents a null letter.

- input word와 유사한 스펠링을 가진 단어 후보자들을 찾음
- Damerau-Levenshtein edit distance : 어떤 문자열 A에서 몇 자를 수정하여 B가 되는지를 숫자로 표현한 것

Spelling Correction – Noisy Channel

Error	Correction	Transformation: Correct Letter	Error Letter	Position (Letter #)	Type
acress	actress	t	—	2	deletion
acress	cress	—	a	0	insertion
acress	caress	ca	ac	0	transposition
acress	access	c	r	2	substitution
acress	across	o	e	3	substitution
acress	acres	—	s	5	insertion
acress	acres	—	s	4	insertion

Figure 5.3 Candidate corrections for the misspelling *acress* and the transformations that would have produced the error (after Kernighan et al. (1990)). "—" represents a null letter.

w	count(w)	p(w)
actress	9,321	.0000231
cress	220	.000000544
caress	686	.00000170
access	37,038	.0000916
across	120,844	.000299
acres	12,874	.0000318

- input word와 유사한 스펠링을 가진 단어 후보자들을 찾음
- Damerau-Levenshtein edit distance : 어떤 문자열 A에서 몇 자를 수정하여 B가 되는지를 숫자로 표현한 것

- 각 보정값 P(w)의 prior 확률은 문맥에서 단어 w가 나타나는 언어 모델의 확률
- 카운트 기반의 접근을 통해 가장 높은 값을 구함

Channel Model을 찾는 방법

- 단어가 잘못 입력될 확률의 모델은 factor들의 모든 정렬 방식을 따르는 것
  - ex) 입력자가 누구인지, 왼손잡이인지 오른손잡이인지 등…
- 하지만 간단한 모델에서는 error의 빈도수를 세는 confusion matrix를 사용
- Confusion Matrix : 알파벳은 26개 문자가 있기 때문에 26 X 26

$$P(x|w) = \begin{cases} \frac{\text{del}[x_{i-1}, w_i]}{\text{count}[x_{i-1} w_i]}, & \text{if deletion} \\ \frac{\text{ins}[x_{i-1}, w_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$

- Confusion Matrix와 Damerau-Levenshtein edit distance를 통해 Channel Model을 구함

Confusion matrix

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

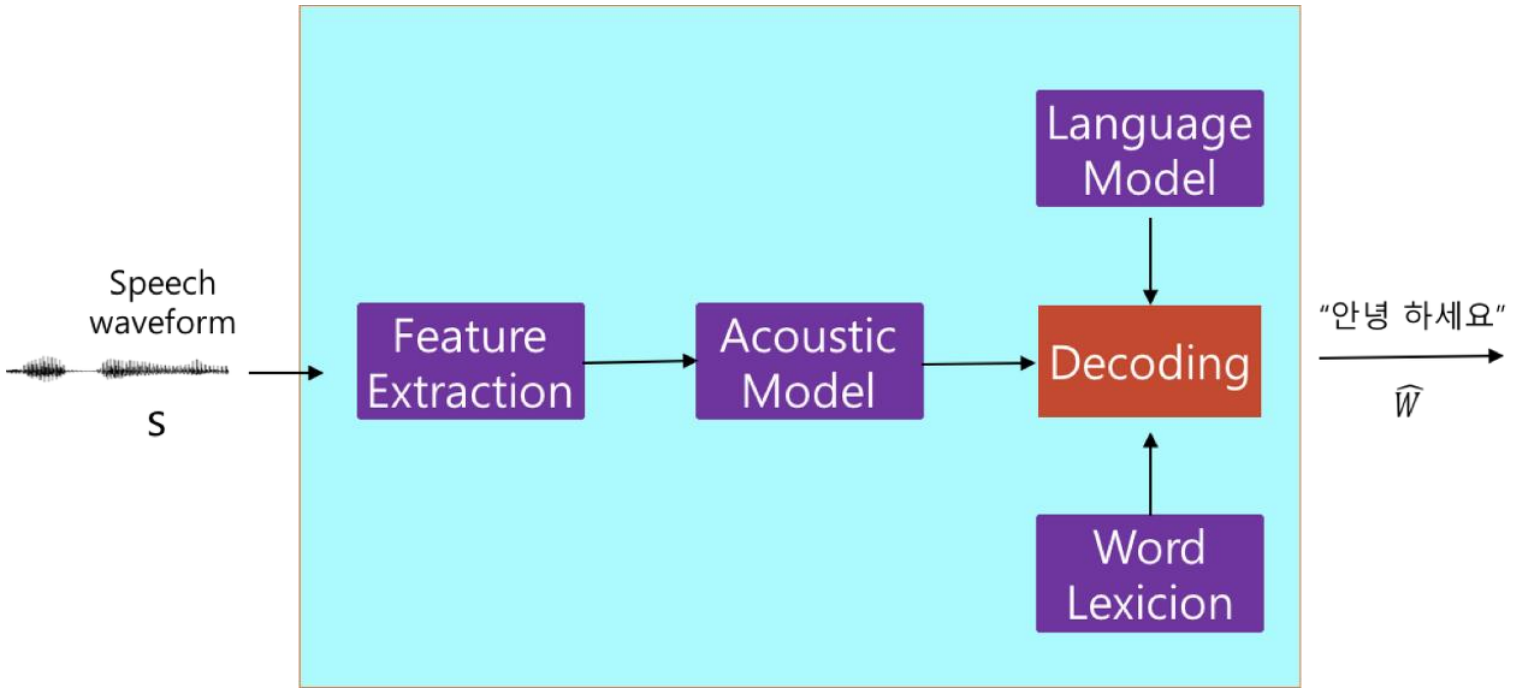
X	Y (correct)																								
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Spelling Correction – Noisy Channel

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|w)	P(w)	10^9*P(x|w)P(w)
actress	t	-	c|ct	.000117	.0000231	2.7
cress	-	a	a|#	.00000144	.000000544	0.00078
caress	ca	ac	ac|ca	.00000164	.00000170	0.0028
access	c	r	r|c	.000000209	.0000916	0.019
across	o	e	e|o	.0000093	.000299	2.8
acres	-	s	es|e	.0000321	.0000318	1.0
acres	-	s	ss|s	.0000342	.0000318	1.0

Figure 5.5 Computation of the ranking for each candidate correction, using the language model shown earlier and the error model from Fig. 5.4. The final score is multiplied by 10^9 for readability.

Speech recognition



$$\underset{W}{\operatorname{argmax}} P(W|X)$$

- 음성인식 : 일정 길이 T 동안 입력된 음성 sequence X에 대해 인식기가 표현할 수 있는 모든 단어들의 조합 중 가장 가능성이 높은 단어열 W를 구하는 것
- 음성인식의 과정은 크게 음성분석, 음향모델 계산, 언어모델 계산, 디코딩 4단계로 이루어짐

Acoustic Model

- Acoustic Model(음향 모델)이란?
  - 음성 신호(audio signal)와 음소 또는 음성을 구성하는 다른 언어 단위 간의 관계를 나타내기 위해 음성 인식에 사용되는 모델
  - 시간축에 따라 움직이며 만든 특징 벡터열 X와 어휘 셋 W에 대해 P(X|W) 확률을 학습
  - 최근의 음향 모델링 방법으로는 DNN-HMM이 쓰임
  - 음향 모델링을 위해서는 음성의 특징되는 부분을 추출해 특징 벡터 생성 (X)

Acoustic Model

- Acoustic Model(음향 모델)이란?
  - 음성 신호(audio signal)와 음소 또는 음성을 구성하는 다른 언어 단위 간의 관계를 나타내기 위해 음성 인식에 사용되는 모델
  - 시간축에 따라 움직이며 만든 특징 벡터열 X와 어휘 셋 W에 대해 P(X|W) 확률을 학습
  - 최근의 음향 모델링 방법으로는 DNN-HMM이 쓰임
  - 음향 모델링을 위해서는 음성의 특징되는 부분을 추출해 특징 벡터 생성 (X)

$\vec{W}: W_1, W_2, W_3, W_4 \ldots\ldots, W_N$ → N개의 단어들로 이루어진 문장

$\vec{X}: X_1, X_2, X_3, X_4 \ldots\ldots, X_T$ → 일정간격으로 추출한 T개의 음성 특징 벡터

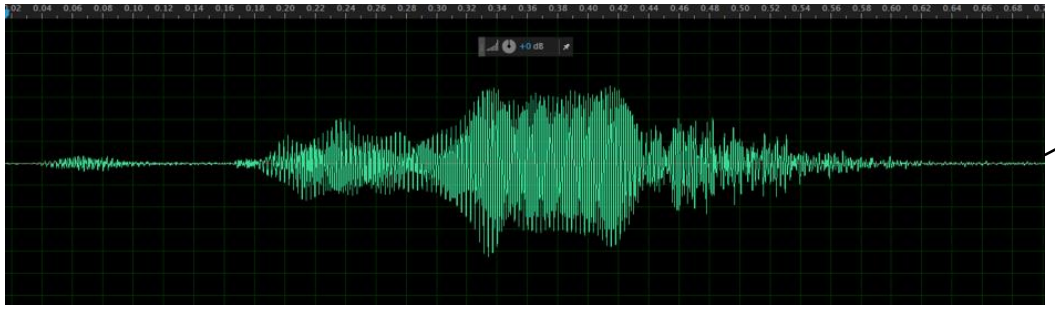
$$\underset{W}{\operatorname{argmax}} P(W|X) = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)} = \underset{W}{\operatorname{argmax}} P(X|W)P(W)$$

음향 모델

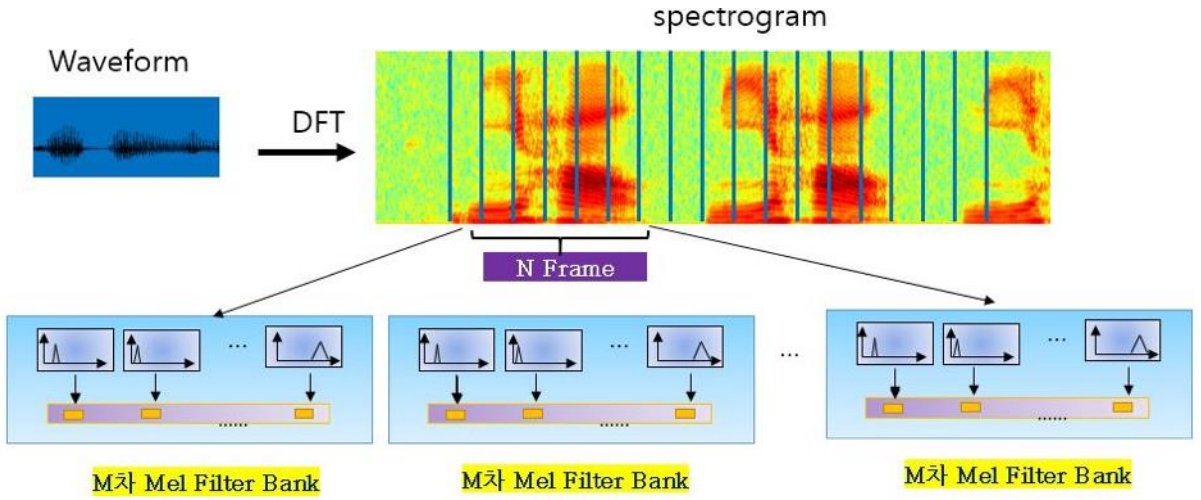
- 음향 모델을 구하는게 Speech recognition의 주목표!

Speech analysis

- 음성신호에서 주파수 분석을 통해 음성의 특징되는 부분을 추출하는 과정
- 1초의 음성이 가질 수 있는 경우의 수는 10^100000 정도이기 때문에 이를 훨씬 작은 차원으로 줄임

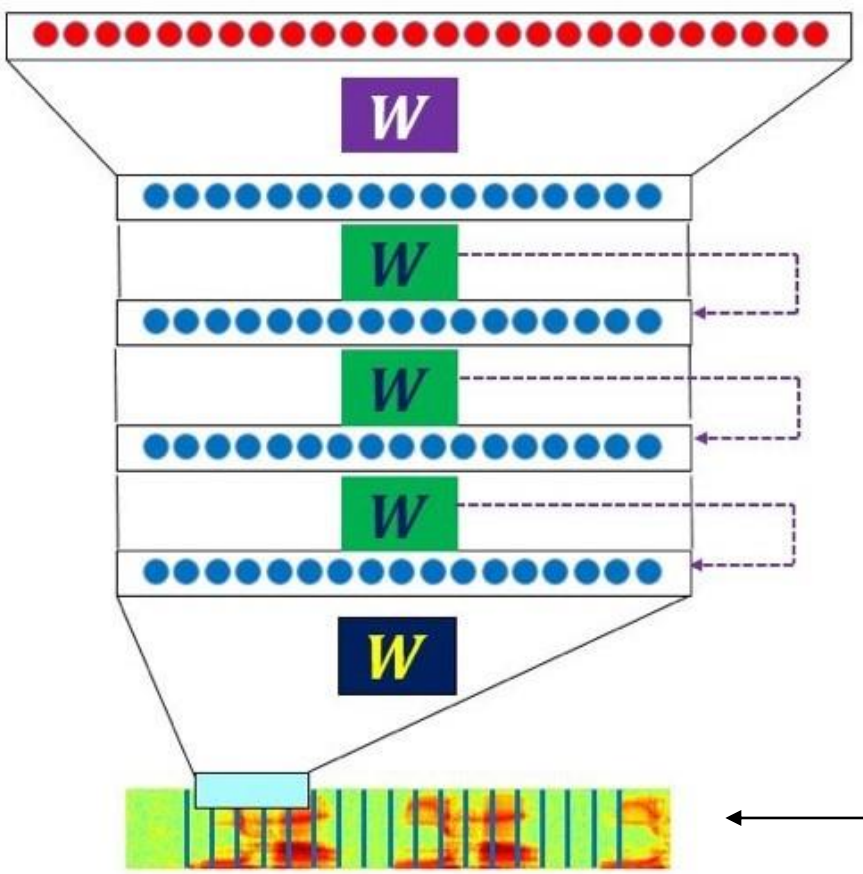


- '카카오' 라고 발성했을 때의 음성 파형

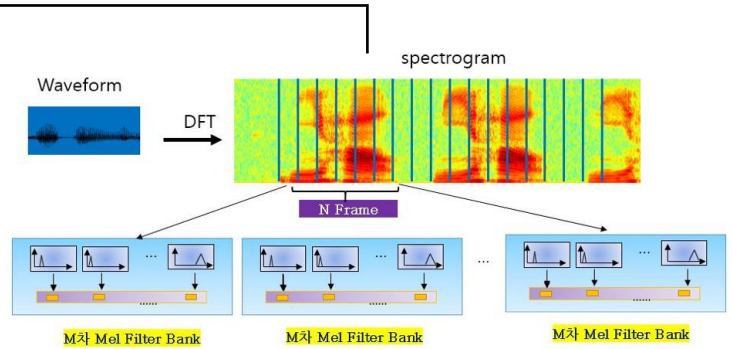


- 특징 벡터 : 음성을 짧은 구간(0.02sec)으로 분석하여 어떤 주파수적 특성을 갖는지 분석해 수십 개의 숫자들로 표현
- 음성 특징 벡터는 x가 된다

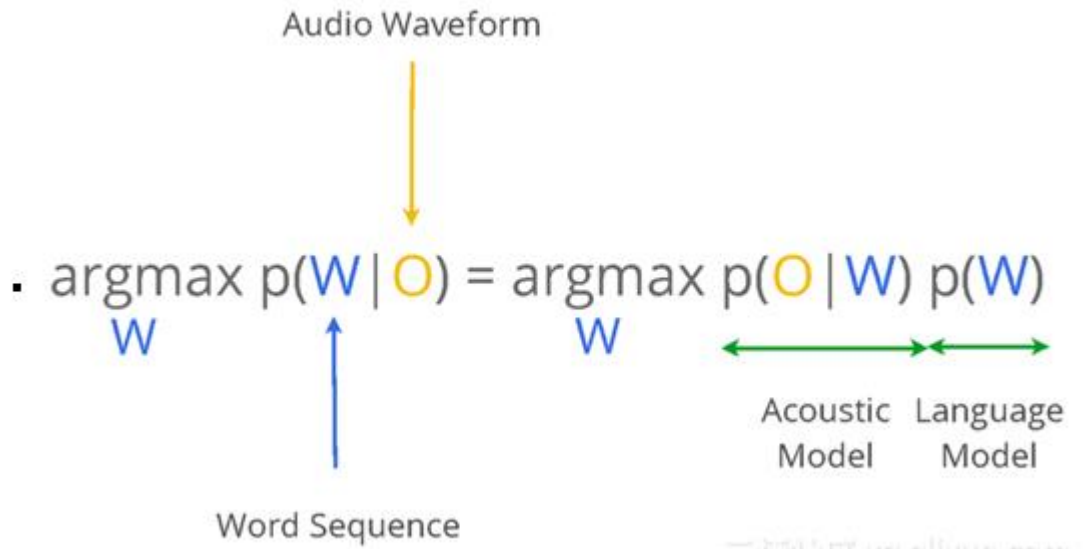
Acoustic Model – DNN-HMM



- 음성 특징 벡터열 X와 어휘 셋 W에 대해 P(W|X) 확률을 학습하는 과정
- DNN-HMM 음소를 DNN으로 모델링하고 이 음소들의 연속적 변화를 HMM으로 예측하는 방식



Speech recognition – Decoding



- 음향모델과 언어모델로 구성된 탐색 공간에서 가장 최적인 경로를 찾는 과정
- Spell Correction과는 Channel Model을 구하는 과정이 다를 뿐 근본적인 공식은 동일

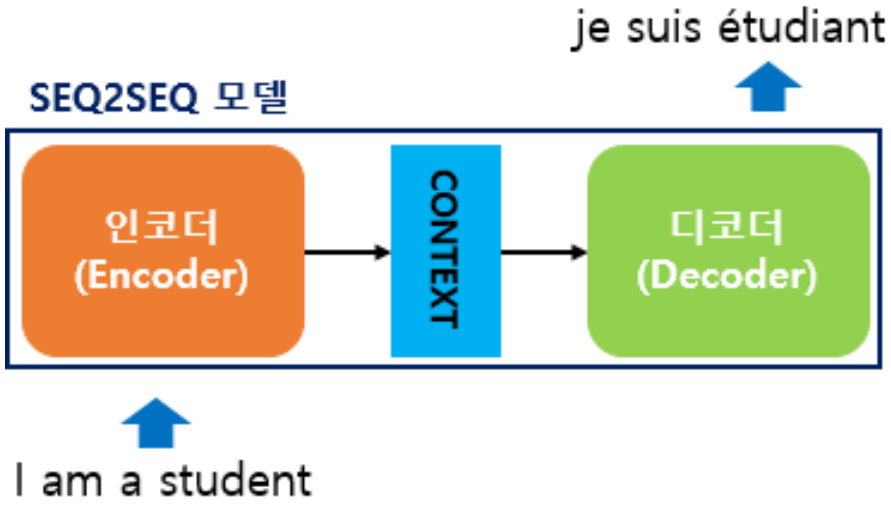
Machine Translation의 종류

- Statistical machine translation
  - 데이터를 기반으로 한 통계학적 접근
  - 한영 번역처럼 어순이 전혀 다른 언어에 잘 대응하지 못함
- Neural Machine Translation
  - 문장 전체를 Context Vector로 표현한 후 이를 기반으로 번역

- NMT가 압도적인 성능을 보이기 때문에 현재 대부분의 기계번역은 NMT를 통해 이뤄짐

Machine Translation

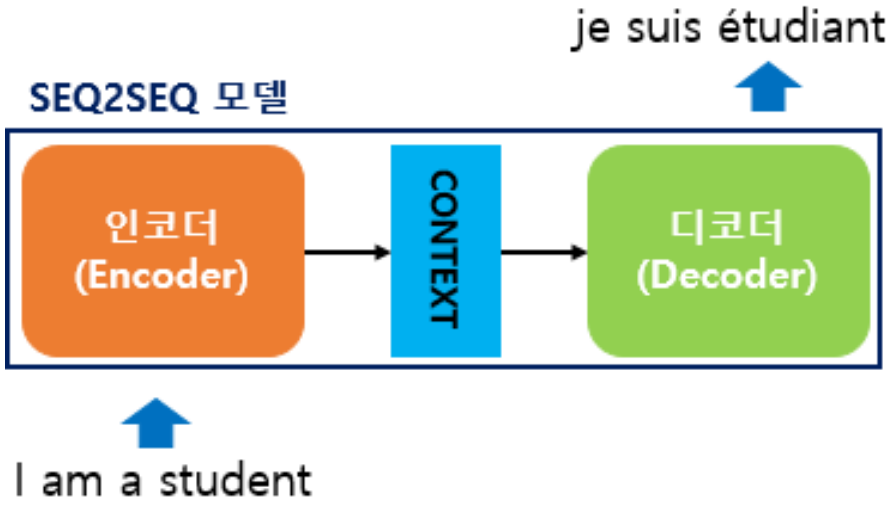
- 기계 번역에서 Language Model의 역할

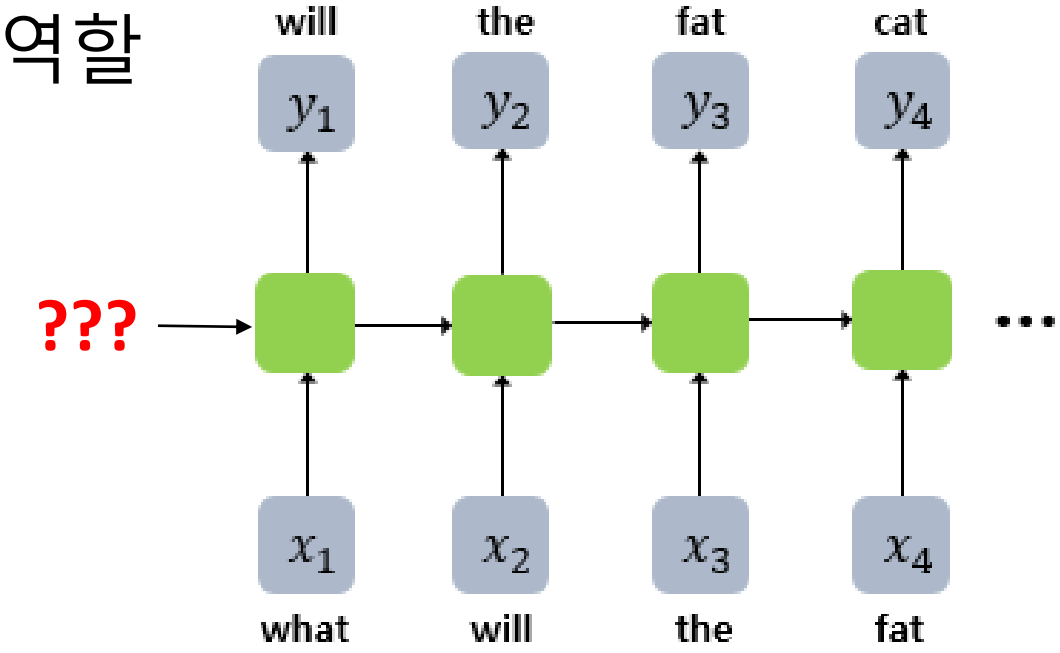


- 기계 번역은 인코더에서 입력 문장을 받아 컨텍스트 벡터에 넘긴뒤, 그걸 이용해 디코더에서 문장을 출력하는데, 이 때 사용되는 언어모델이 RNNLM

Machine Translation

- 기계 번역에서 Language Model의 역할





- 기계 번역은 인코더에서 입력 문장을 받아 컨텍스트 벡터에 넘긴뒤, 그걸 이용해 디코더에서 문장을 출력하는데, 이 때 사용되는 언어모델이 RNNLM

- 기계 번역의 RNNLM이 초기값으로 컨텍스트 벡터를 받는다는게 기본 RNNLM과의 차이

LSTM-LM – teacher forcing

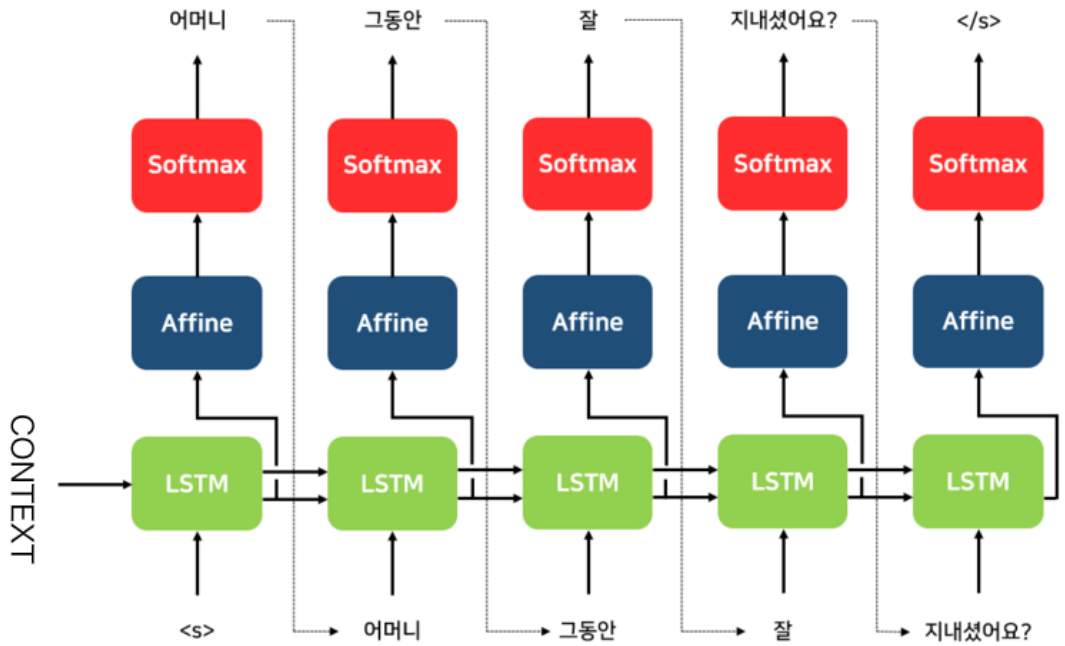
NNLM의 window size 개념 대신 timestep 개념

- RNNLM이나 LSTM-LM 같은 시퀀스 언어 모델은 기존의 NNLM과는 다르게 교사 강요(teacher forcing)를 통해 학습

- '교사 강요'란 테스트 과정에서 t 시점의 출력이 t+1 시점의 입력으로 사용되는 RNN 모델을 훈련시킬 때 사용하는 훈련 기법

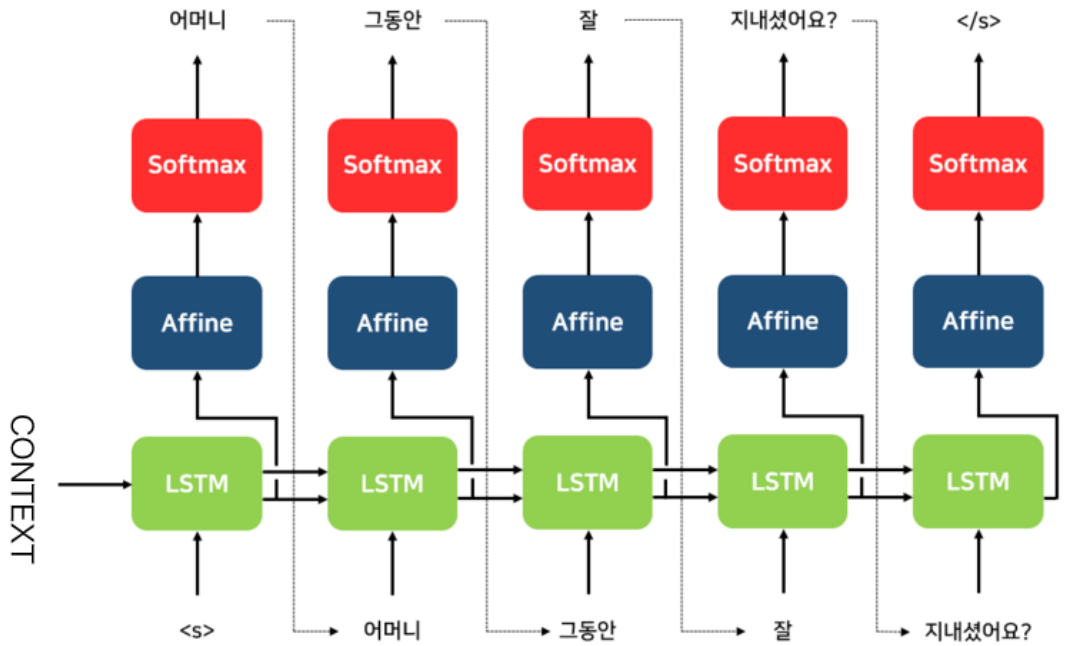
- But 개념만으로는 무슨 뜻인지 모르겠다!!

LSTM-LM – teacher forcing



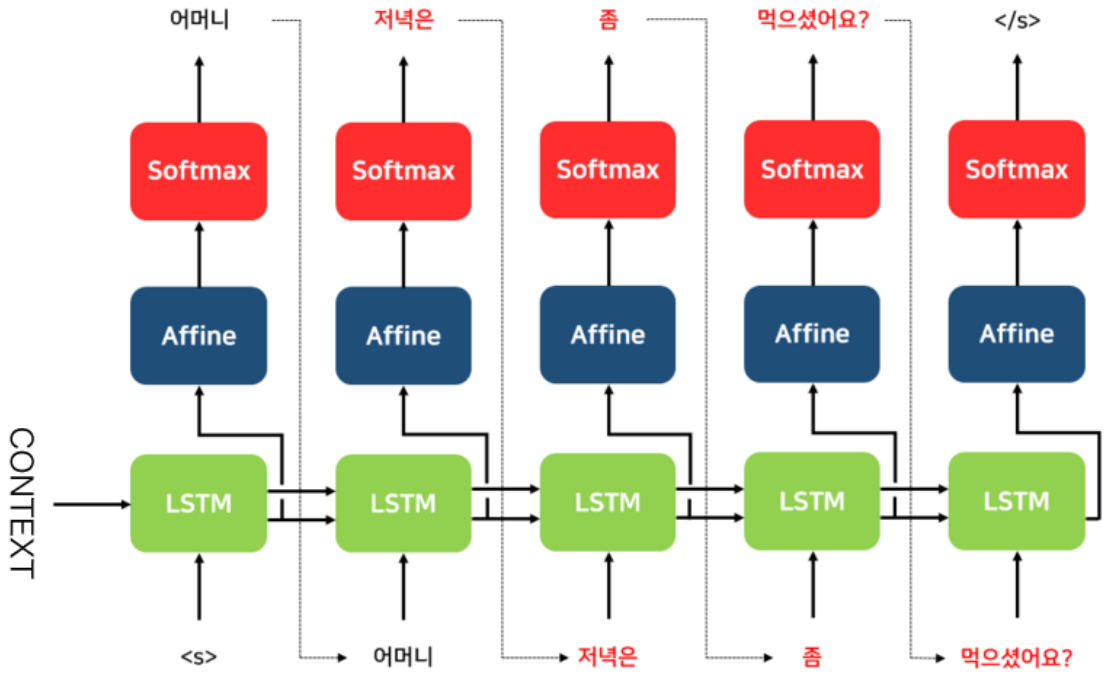
- 정확한 예측이 선행됐다면 상관없지만

LSTM-LM – teacher forcing

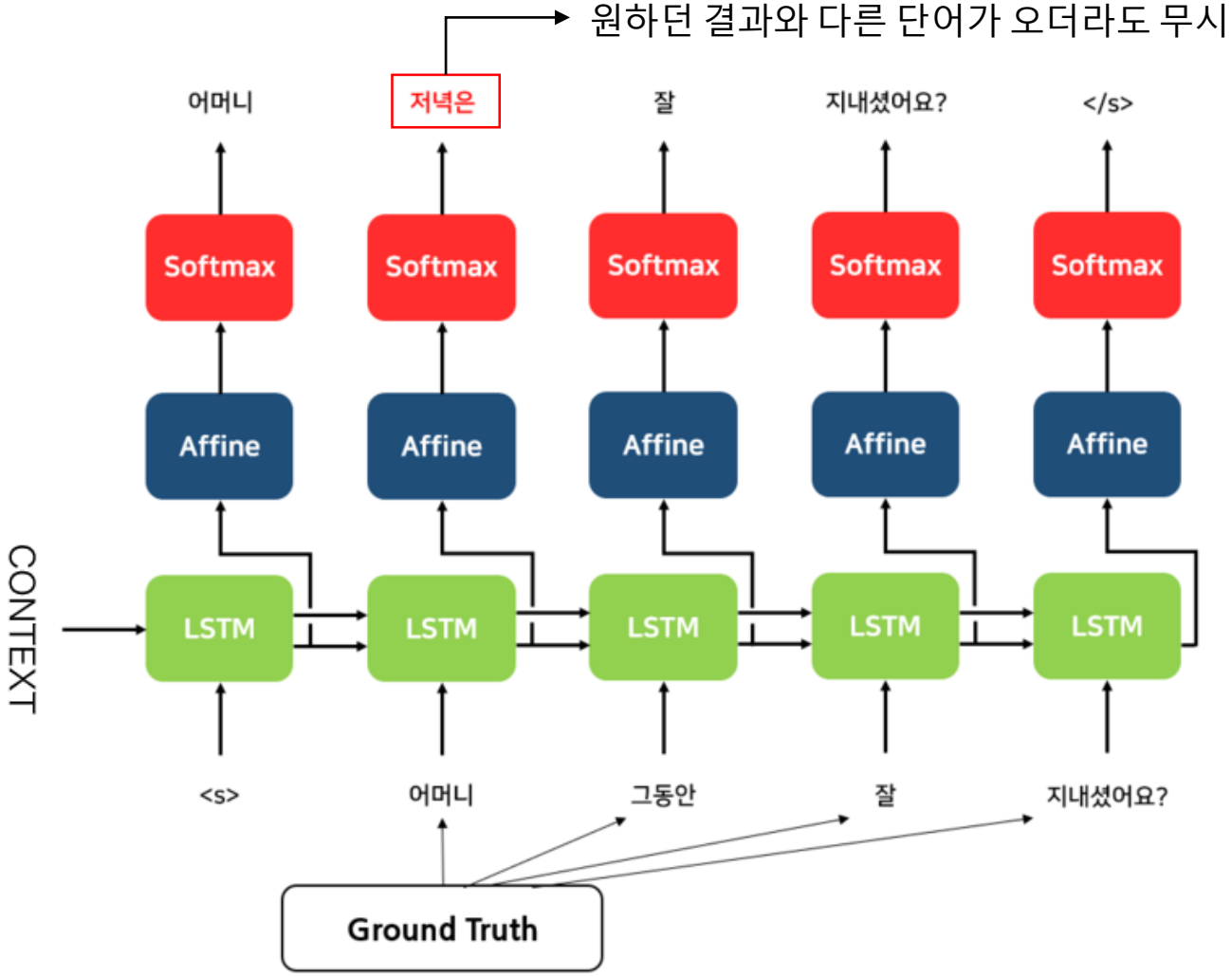


- 잘못된 예측이 이루어지면 그 후의 값들이 전부 잘못된 예측으로 이어짐

- 정확한 예측이 선행됐다면 상관없지만

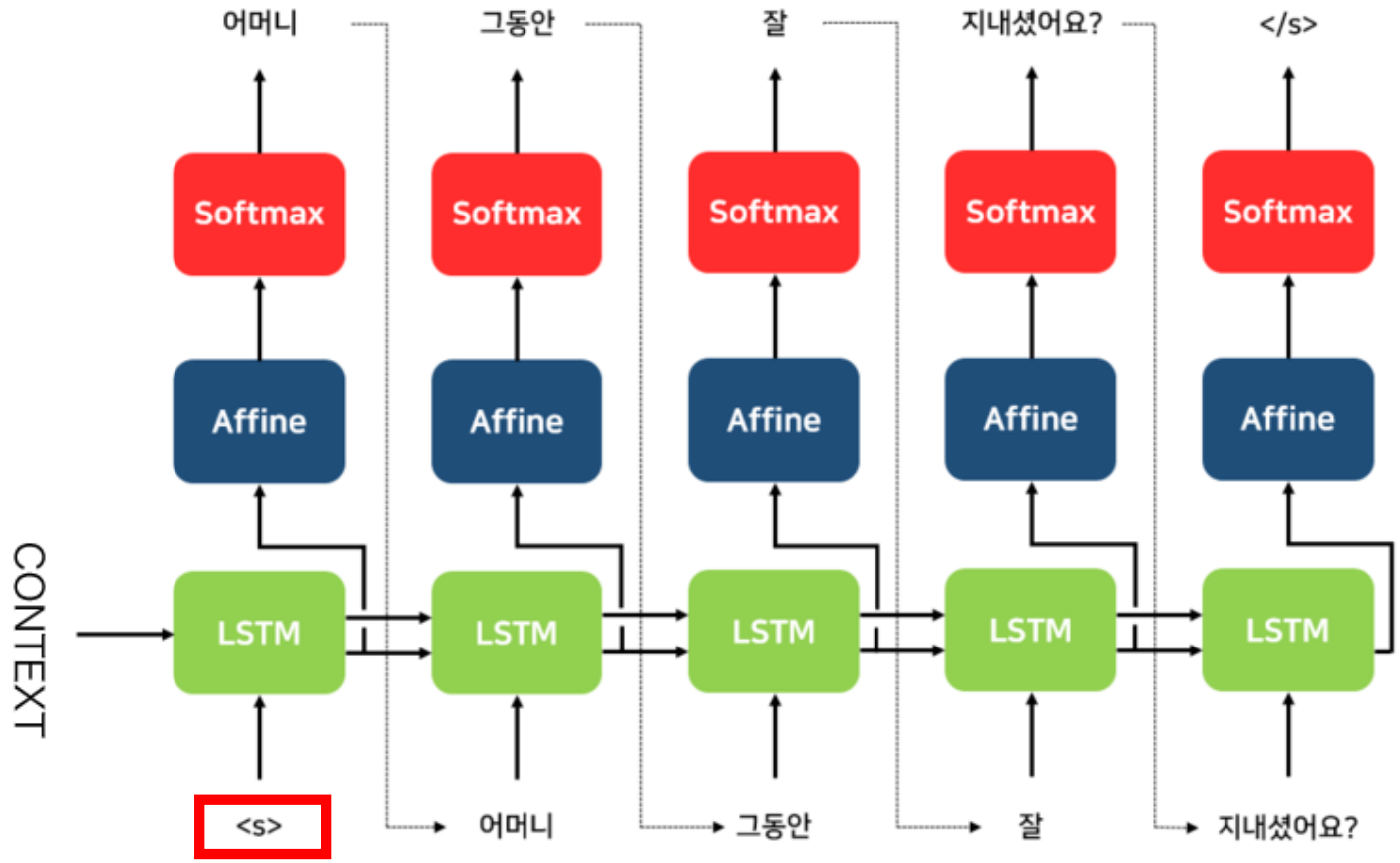


LSTM-LM – teacher forcing



- 때문에 입력으로 알고 있는 정답을 넣어주는 것!
- Mother, how have you been?에 대한 입력은 무조건 '어머니 그동안 잘 지내셨어요?' 이다

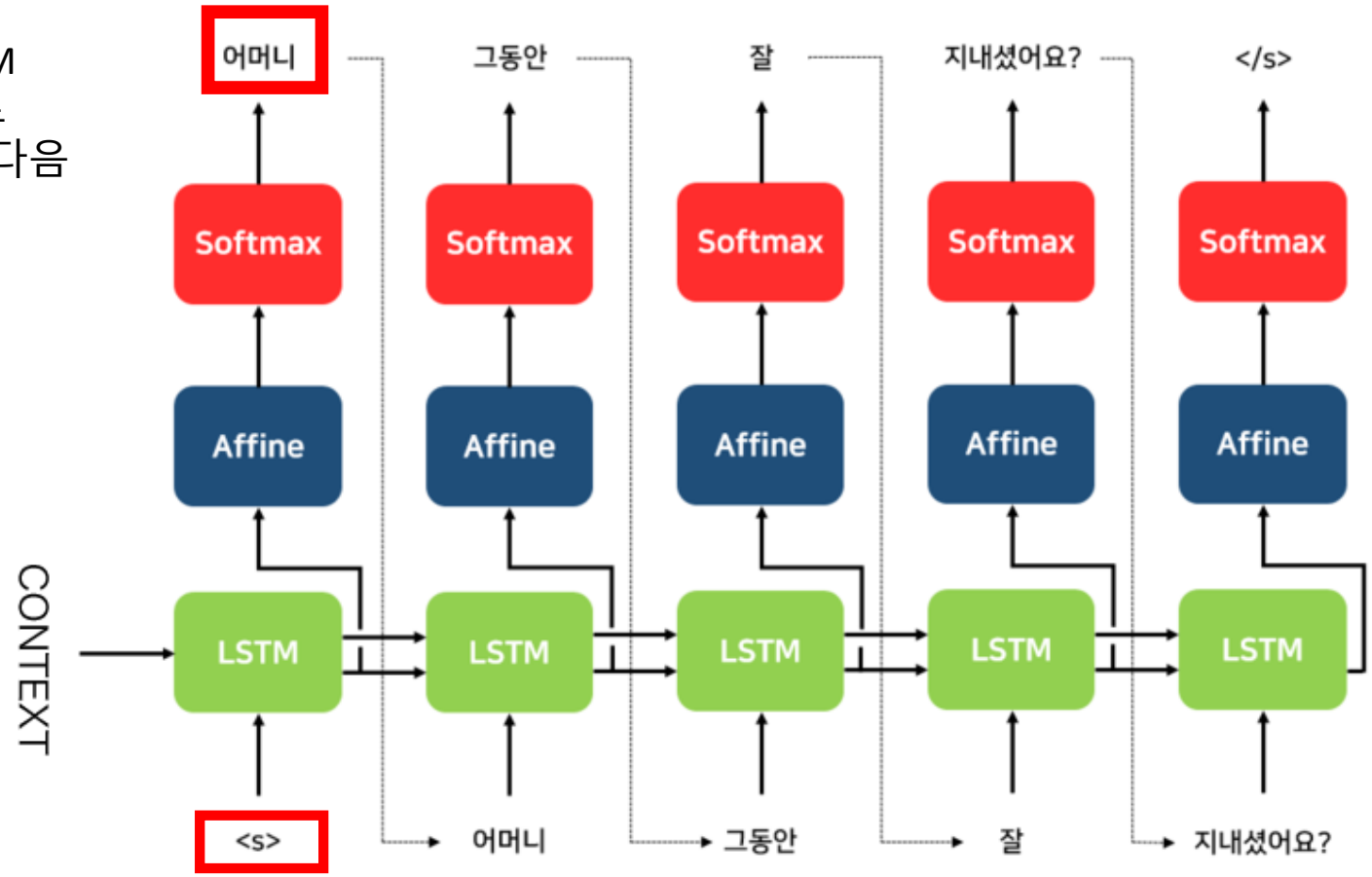
Machine Translation – LSTM-LM



- 디코더의 초기값인 <s>가 입력되면, 다음에 등장할 확률이 가장 높은 단어를 예측

Machine Translation – LSTM-LM

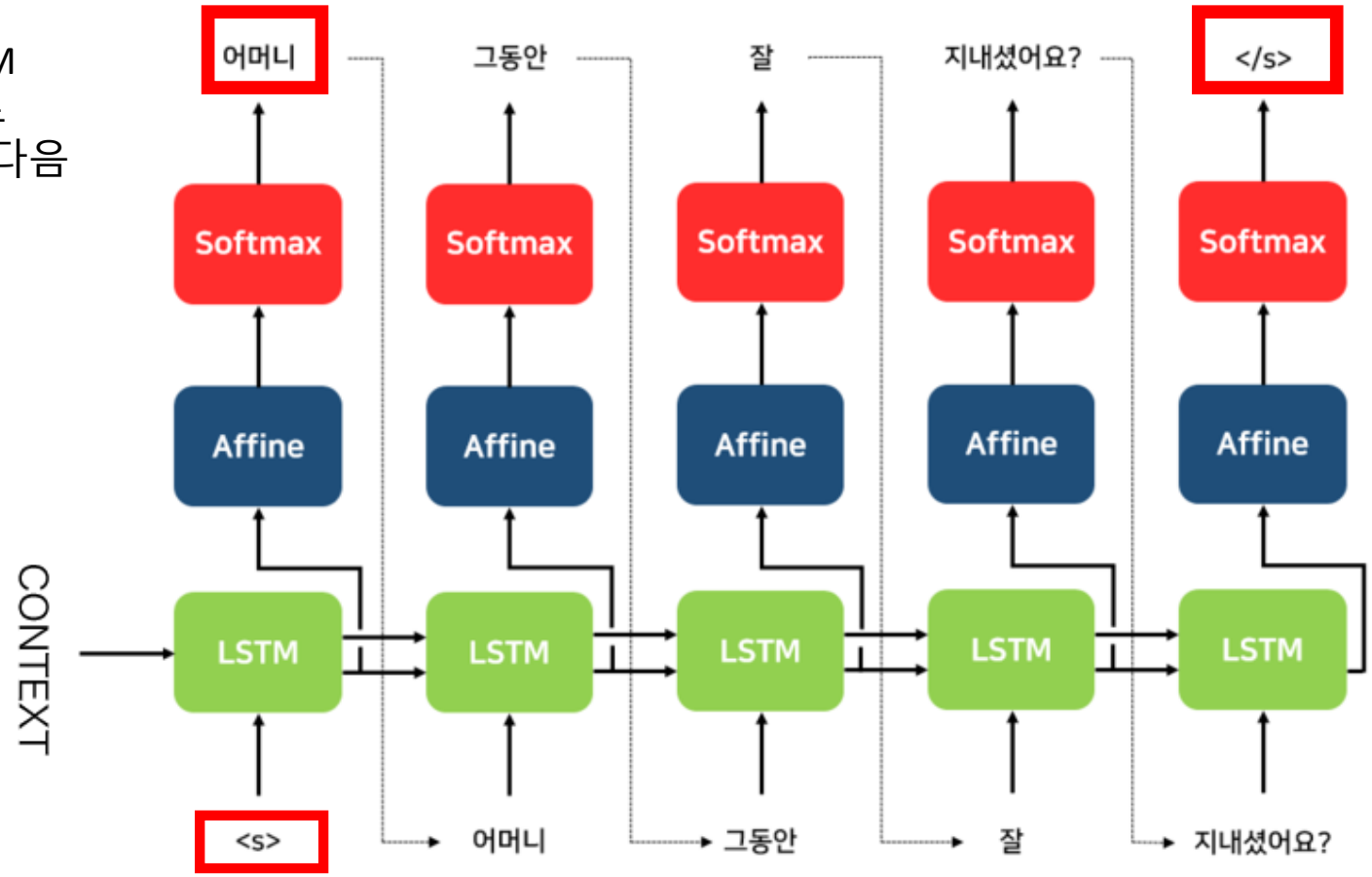
- 첫번째 시점의 디코더 LSTM 셀은 다음에 등장할 단어로 '어머니'를 예측하고 이를 다음 시점의 LSTM 셀로 입력



- 디코더의 초기값인 <s>가 입력되면, 다음에 등장할 확률이 가장 높은 단어를 예측

Machine Translation – LSTM-LM

- 첫번째 시점의 디코더 LSTM 셀은 다음에 등장할 단어로 '어머니'를 예측하고 이를 다음 시점의 LSTM 셀로 입력

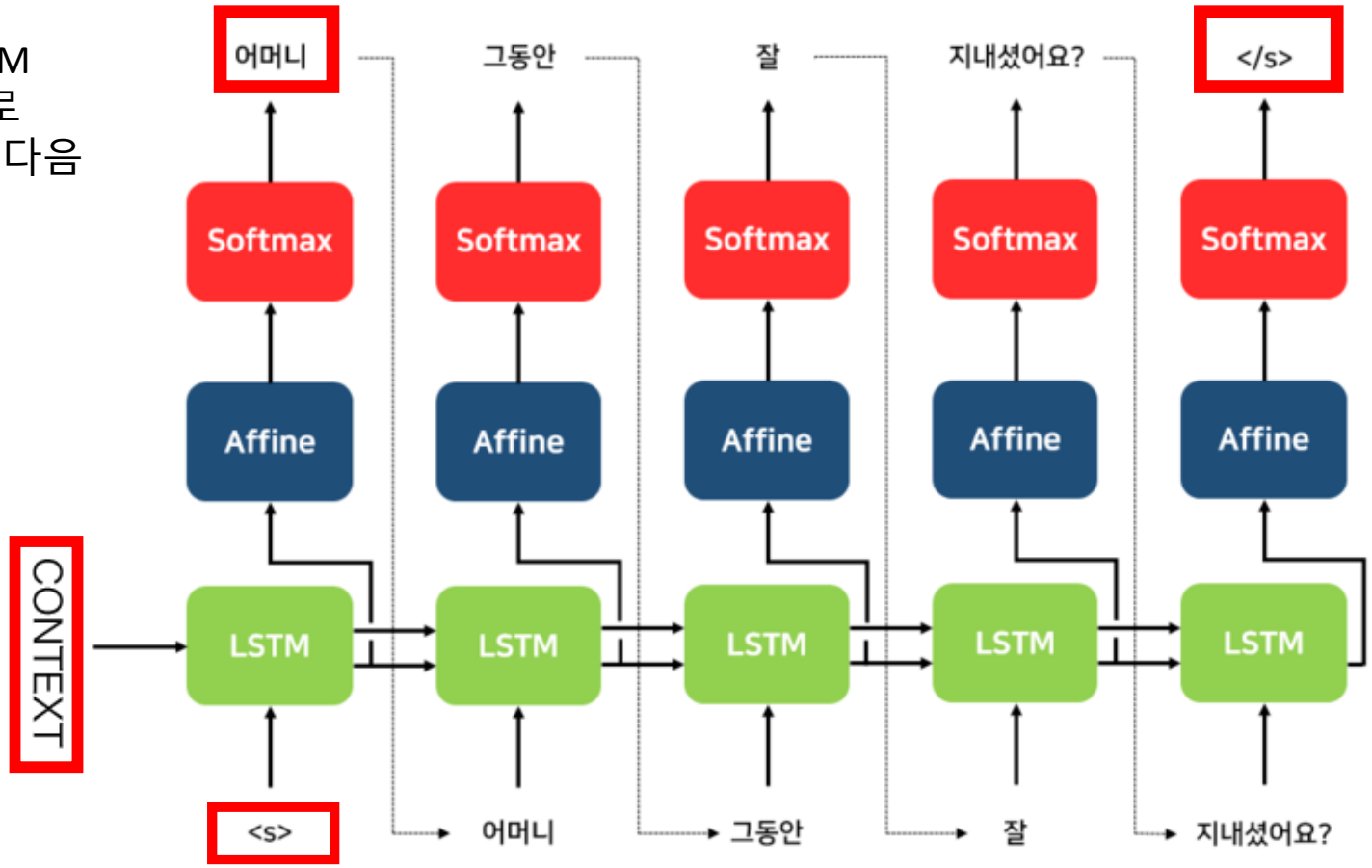


- 예측한 단어를 다음 시점의 RNN 셀의 입력으로 넣는 행위를 반복하며, </s>가 나올 때까지 반복

- 디코더의 초기값인 <s>가 입력되면, 다음에 등장할 확률이 가장 높은 단어를 예측

Machine Translation – LSTM-LM

- 첫번째 시점의 디코더 LSTM 셀은 다음에 등장할 단어로 '어머니'를 예측하고 이를 다음 시점의 LSTM 셀로 입력



- 하지만 컨텍스트 벡터는 크기가 한정되어 있기 때문에 정확도에는 한계가 있음 -> 어텐션

- 예측한 단어를 다음 시점의 RNN 셀의 입력으로 넣는 행위를 반복하며, </s>가 나올 때까지 반복

- 디코더의 초기값인 <s>가 입력되면, 다음에 등장할 확률이 가장 높은 단어를 예측

Thank you