

# Seq2Seq

<Sequence to Sequence Learning with Neural Networks>  
(NIPS 2014)

# 기존 RNN, LSTM의 문제점



입력 문장의 길이와 출력문장의 길이가 같음, fixed 되어있음

# 기존 RNN, LSTM의 문제점



# 문제의식

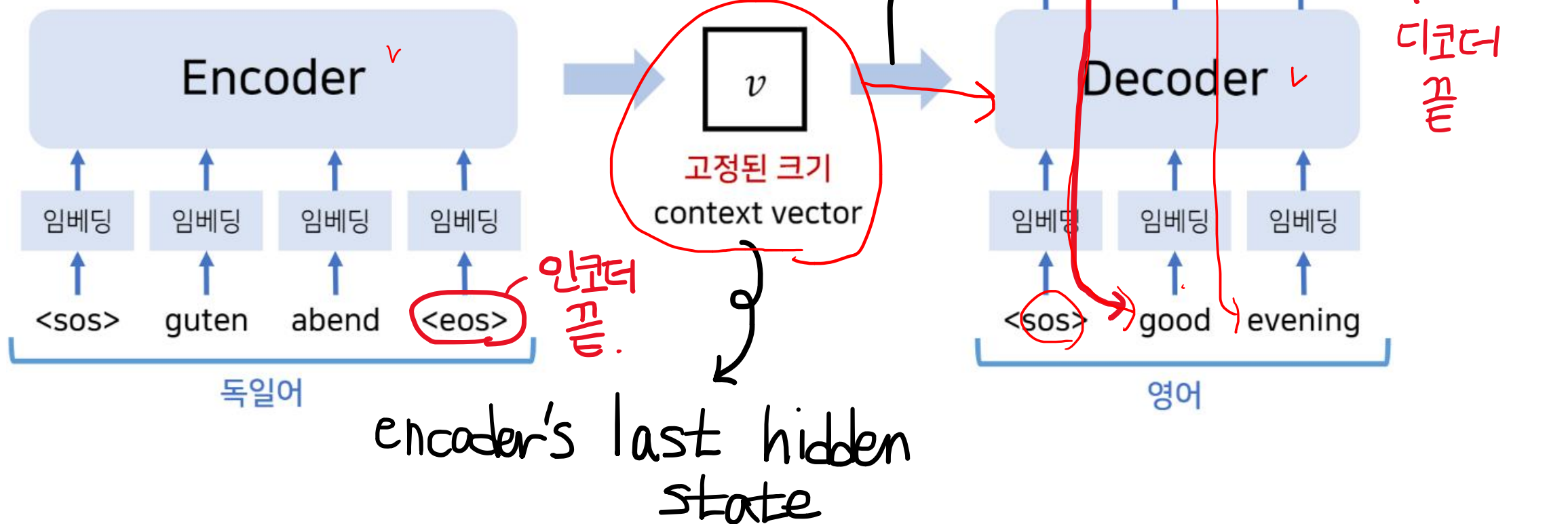
- Domain-independent한 방법이 필요하다
- Monotonic한 관계가 아닌 문장에도 잘 적용할 수 있는 방법이 필요하다

# 제안된 모델

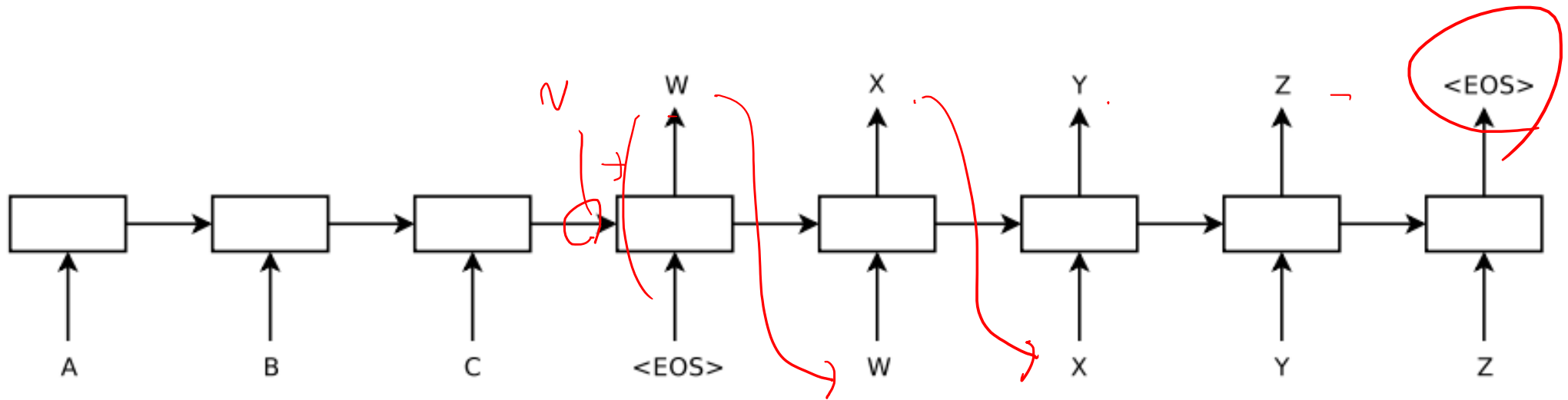
- 첫번째 LSTM이 입력 문장을 순서대로 읽는다
- 그 LSTM은 문장 전체 정보를 나타내는 representation vector를 생성한다
- 또다른 LSTM은 그 벡터로부터 출력 문장을 만들어낸다

# 제안된 모델

※ 인코더와 디코더는 서로 다른 가중치를 가짐.  
(다른 LSTM).

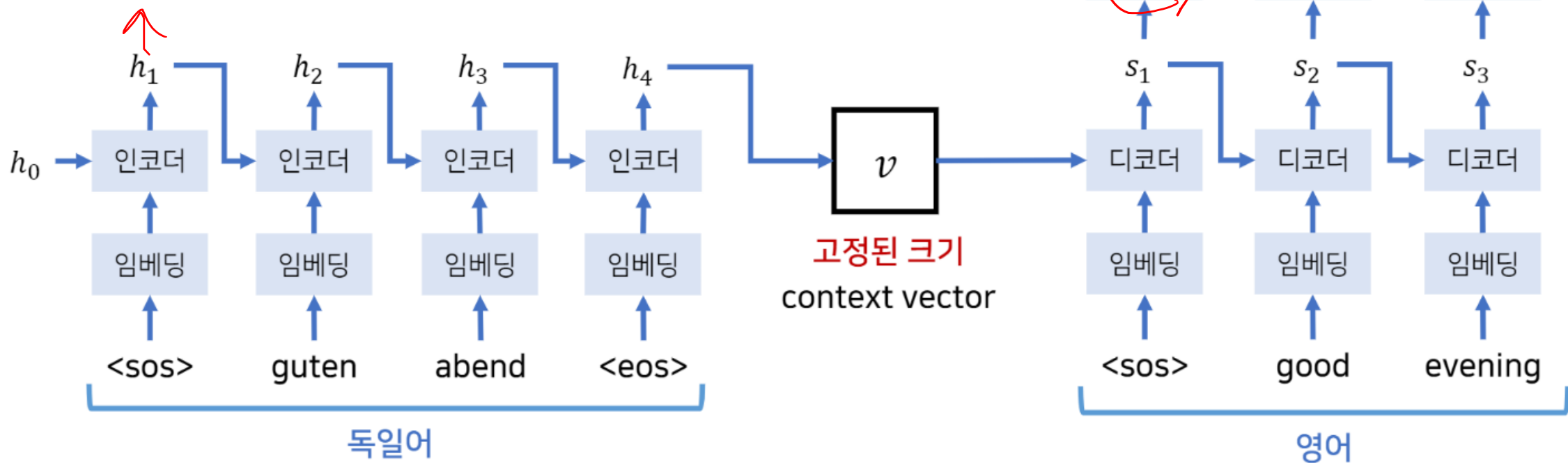


# 제안된 모델



# 제안된 모델

- $x_t$  : 현재의 입력 단어
- $h_t$  : 지금까지 입력된 문장에 대한 정보를 담은 벡터 표현
- $s_t$  : 지금까지 출력된 문장에 대한 정보를 담은 벡터 표현
- $y_t$  : 현재의 출력 단어





## + ) 언어 모델

- 문장(sequence)에 확률을 부여하는 것

- 연쇄 법칙

$$P(\text{친구와 친하게 지내다}) = P(\text{친구와}) * P(\text{친하게} | \text{친구와}) * P(\text{지내다} | \text{친구와 친하게})$$

$$P(w_1, \dots, w_i) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$

# Seq2Seq 가 구하고자 하는 것

입력문장이 들어왔을 때

$$P(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} P(y_t | v, y_1, \dots, y_{t-1})$$

출력문장이 나올 확률.  
(특정).

representation  $v$ .

→ 그 확률은 연쇄적으로 구한다.

-※.  $T' \neq T$  일 수 있다.

# Experiments

# 큰 LSTM 모델 학습시키기-beam search

- Log probability를 최대로 만들도록 학습

$$1/|\text{set}| \sum_{(T,S) \in \text{set}} \log p(\underline{T}|S)$$

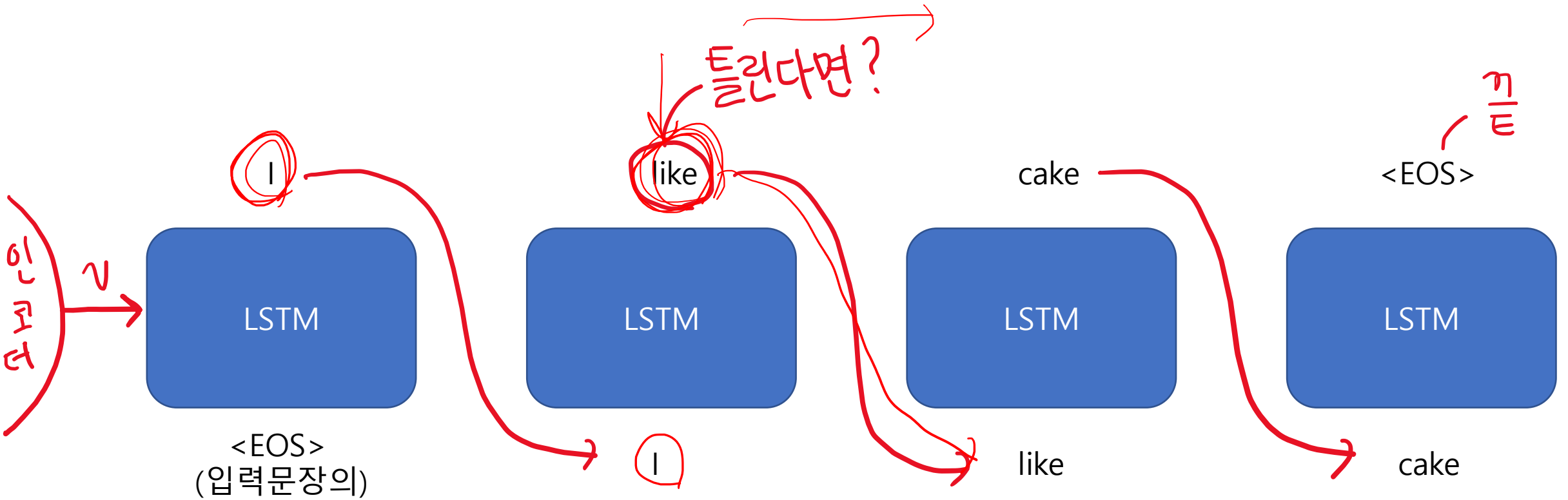
$T$ : correct translation  
 $S$ : source.

- 학습이 완료되면, 가장 적합한 번역을 결과로 도출

$$\hat{T} = \arg \max_T p(T|S)$$



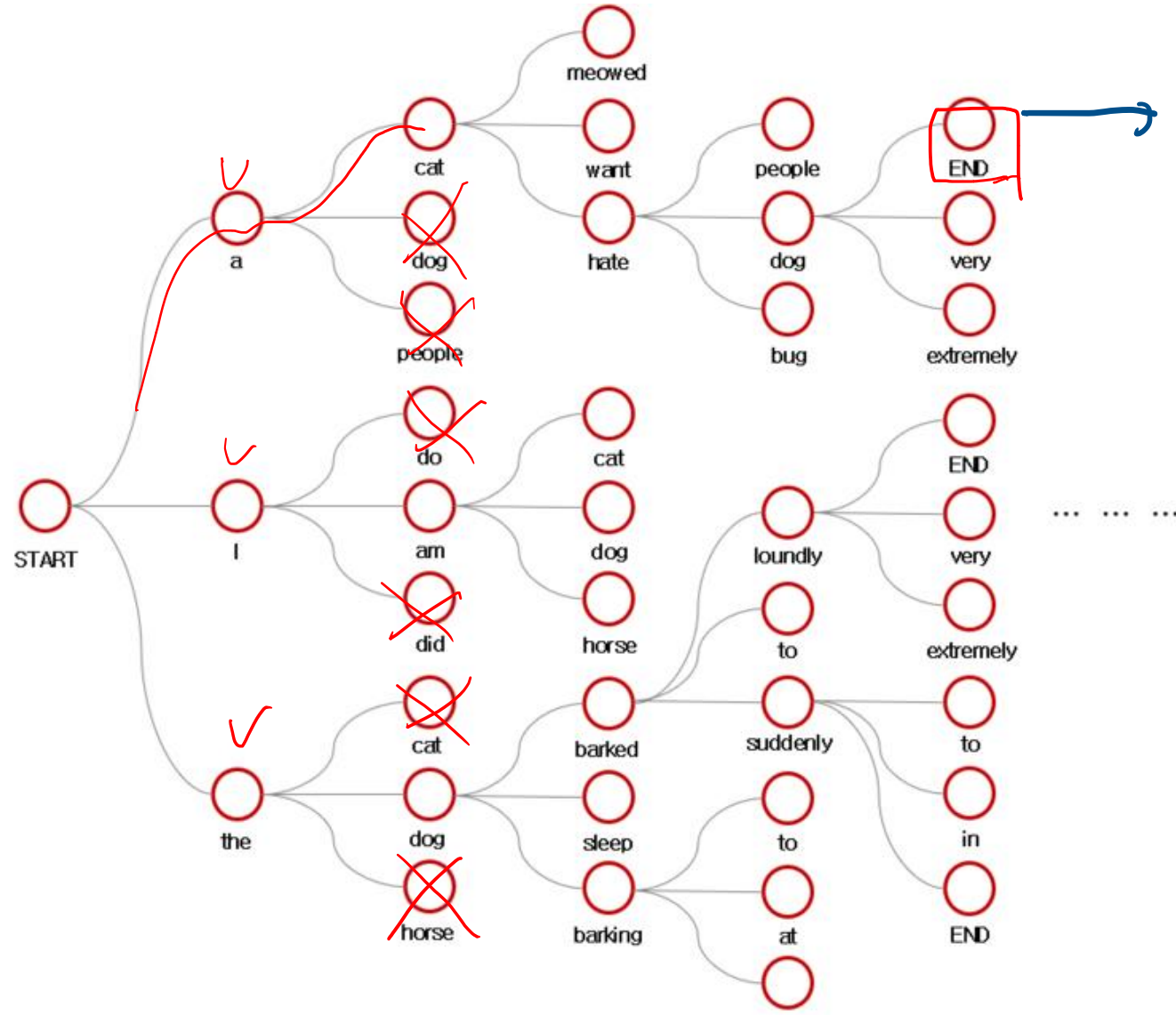
# 큰 LSTM 모델 학습시키기-beam search



greedy decoding의 경우 문제가 생김.

# 큰 LSTM 모델 학습시키기-beam search

$k = \text{beam 개수}$   
 $= 3$

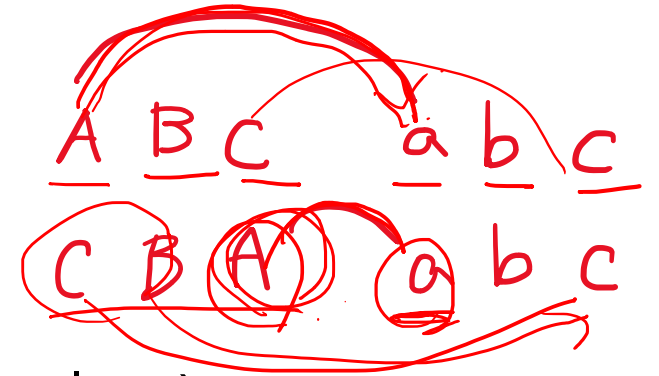


eos를 만난  
beam이 k개가  
될 때 까지.

# ✓문장 순서 바꾸기 (reversing)


short

- 입력 문장의 단어 순서를 거꾸로 함  
=> BLEU 스코어가 25.9 에서 30.6으로 높아짐



- 예상 이유
  - Short term dependency를 해결해줌 ( minimal time lag )
  - Source와 target 간의 평균 거리는 그대로이지만
  - 학습(최적화)이 더 쉬워져서 성능이 좋아진 것 같다고 예측
- 원래는 문장 뒷부분에 대한 판단이 안 좋아질 거라고 생각했는데 오히려 긴 문장도 잘 처리할 수 있게 됨

# Training details

- Deep LSTM layers (4)
- 입력 vocab: 160,000 출력 vocab: 80,000
- 파라미터를  $-0.08 \sim 0.08$  균일분포로 초기화
- ✓ • SGD 사용, learning rate=0.7, epoch에 따라 학습률 감소 *learning rate decay*
- Batch size=128
- 비슷한 길이의 문장끼리 하나의 batch로 *✓* 
- Exploding gradient 해결을 위해 gradient의 크기를 제한

*LSTM*

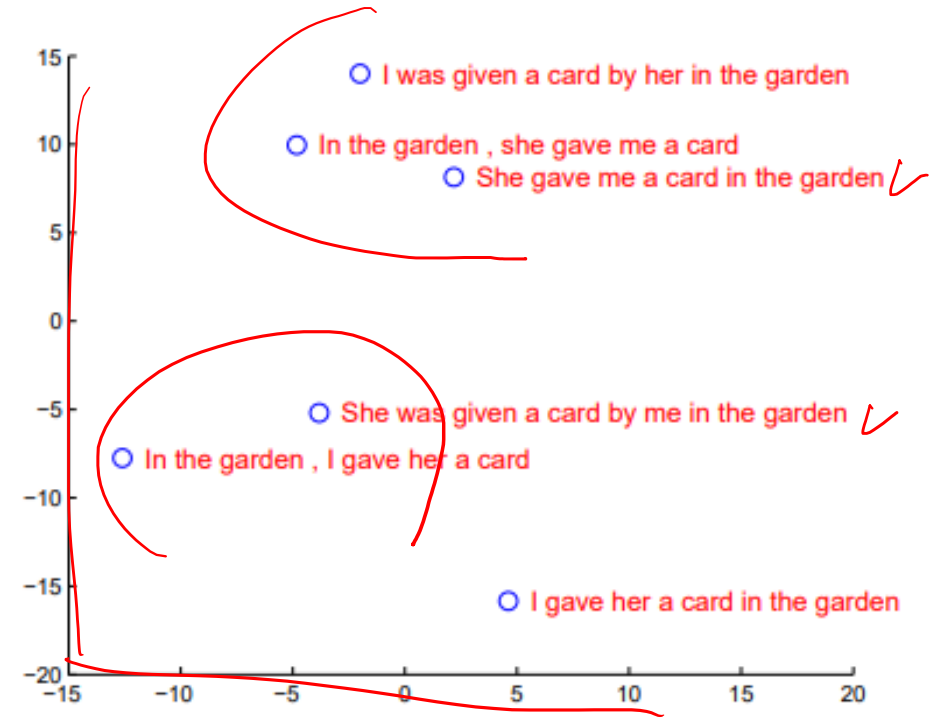
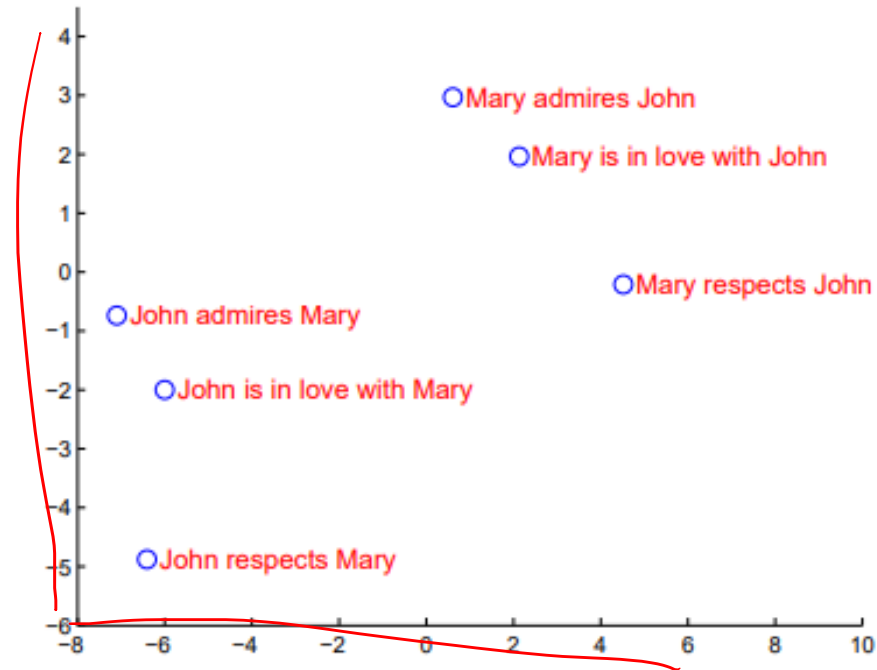
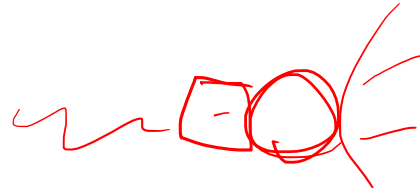


# results

| Method                                     | test BLEU score (ntst14) |
|--|--------------------------|
| Bahdanau et al. [2]                        | 28.45                    |
| Baseline System [29]                       | 33.30                    |
| ○ Single forward LSTM, beam size 12        | 26.17                    |
| Single reversed LSTM, beam size 12         | 30.59                    |
| Ensemble of 5 reversed LSTMs, beam size 1  | 33.00                    |
| Ensemble of 2 reversed LSTMs, beam size 12 | 33.27                    |
| Ensemble of 5 reversed LSTMs, beam size 2  | 34.50                    |
| Ensemble of 5 reversed LSTMs, beam size 12 | <b>34.81</b>             |

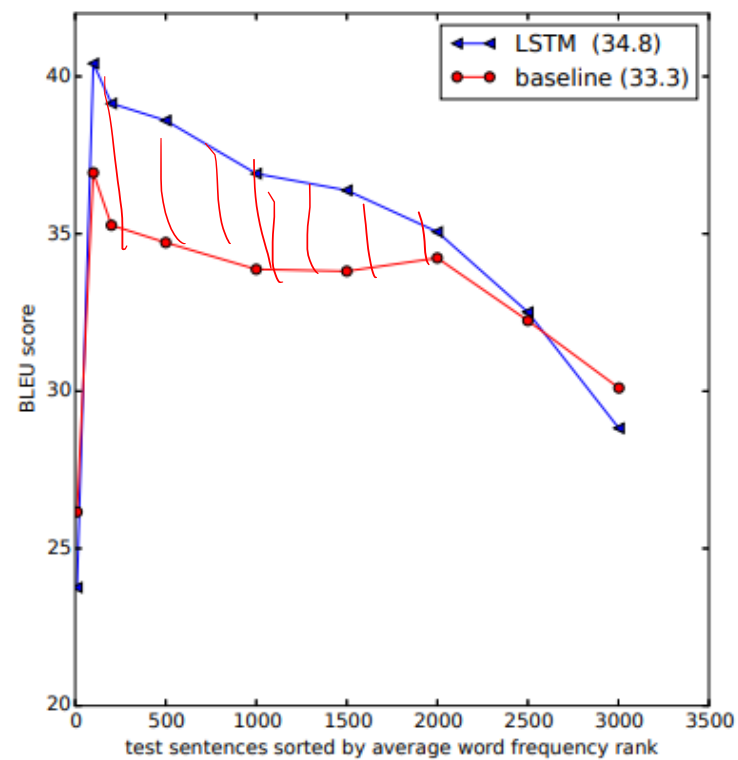
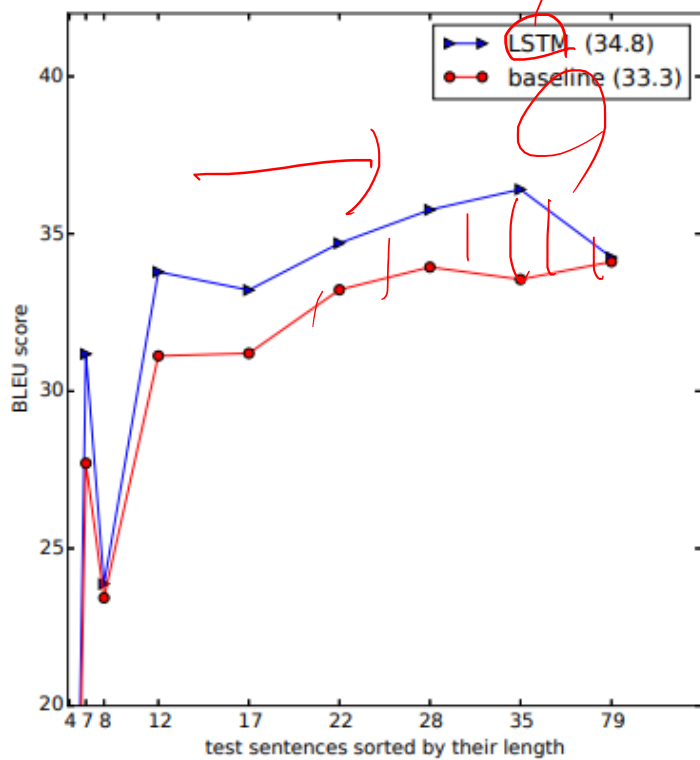
| Method  | test BLEU score (ntst14) |
|---|--------------------------|
| ○ Baseline System [29]  | 33.30                    |
| Cho et al. [5]  | 34.54                    |
| // <u>Best WMT'14 result [9]</u>                                      | <b>37.0</b>              |
| Rescoring the baseline 1000-best with a single forward LSTM           | 35.61                    |
| Rescoring the baseline 1000-best with a single reversed LSTM          | 35.85                    |
| Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs | <b>36.5</b>              |
| Oracle Rescoring of the Baseline 1000-best lists                      | ~45                      |

# 분석



순서에 따른 의미 변화도 잘 감지  
=> 기존의 bag of words 로는 파악할 수 없는 부분

# 분석



긴 문장도 ok  
단어 빈도가 낮아도 좋은 성능

# 결론

80k vocab.

- limited vocab의 Seq2Seq는 기존의 unlimited SMT를 뛰어넘음
  - 아직 심플하고 unoptimized 한 모델의 성능이므로,
  - 앞으로 더욱 발전할 가능성이 높음

통계적인

- ☆ Source를 reverse 했을 때 놀라운 성능을 보임
  - (• 긴 문장에 대해서도 좋은 성능 (이전엔 불가능)
  - RNN도 reversed data를 쓴다면 학습가능할 것이다