Seq2Seq:

Sequence to Sequence Learning with Neural Networks

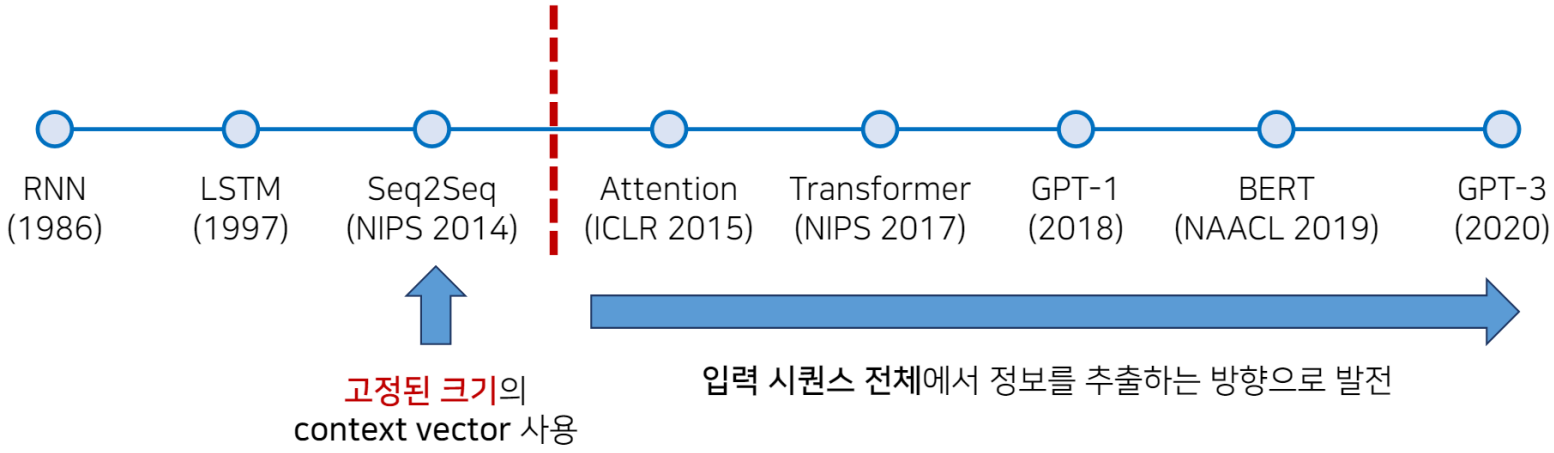
2024.02.02

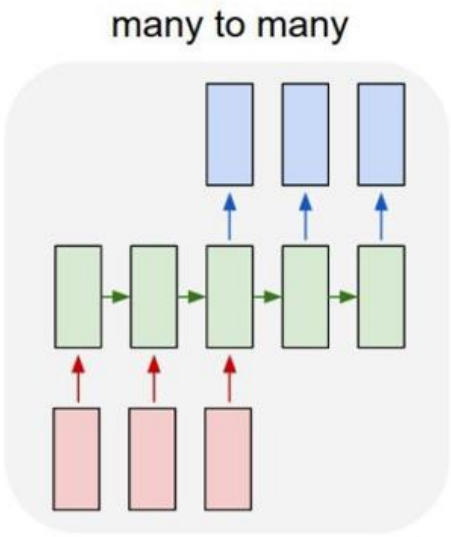
이은주

Intro

Seq2Seq(2014) : 한 시퀀스를 다른 시퀀스로 변환하는 작업을 수행하는 딥러닝 모델
주로 자연어 처리(NLP)분야에서 활용

Transformer(2017)가 나오기 전까지는 성능이 가장 좋았음.
GPT(2018) : Transformer의 디코더 아키텍처 사용
BERT(2019) : Transformer의 인코더 아키텍처 사용





이전 seq2seq(SMT)

전통적인 통계적 언어모델(Statistical Language Model)은 카운트 기반의 접근 사용

- $P(\text{지낸다}|\text{친구와 친하게}) = \frac{count(\text{친구와 친하게 지낸다})}{count(\text{친구와 친하게})}$

한계점:

- 현실적으로 모든 문장에 대한 확률을 가지고 있어야함.

만일, '친구와 친하게'라는 문장이 없으면 확률은 0.

- 긴문장 처리 어려움

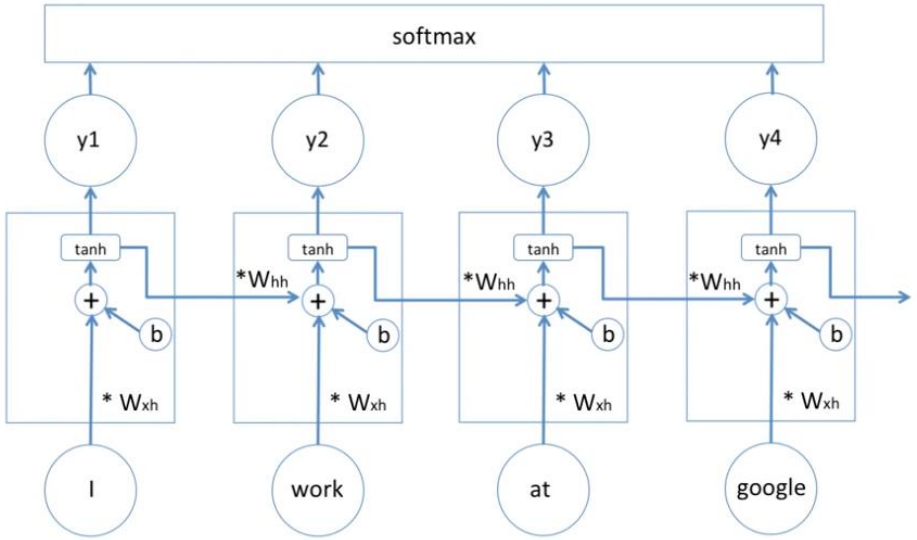
- $P(\text{나는 공부를 마치고 집에서 밥을 먹었다}) = P(\text{나는}) * P(\text{공부를}|\text{나는}) * P(\text{마치고}|\text{나는 공부를}) *$
  $P(\text{집에서}|\text{나는 공부를 마치고}) * P(\text{밥을}|\text{나는 공부를 마치고 집에서}) * P(\text{먹었다}|\text{나는 공부를 마치고 집에서 밥을})$

=> N-gram 언어 모델 사용

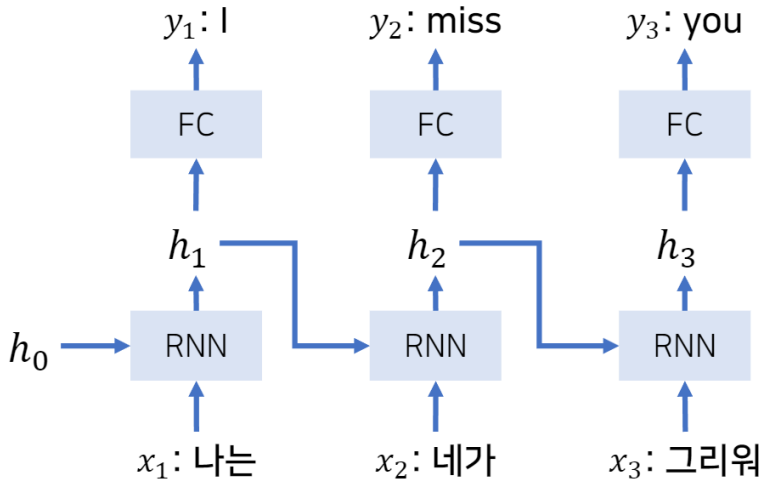
인접한 일부 단어만 고려하는 아이디어

이전 seq2seq(RNN)

RNN과 같은 딥러닝 기법은 강력하지만 일반적인 task에 한계 존재.
초반 딥러닝 모델은 입력과 출력의 dim이 고정되어 있는 경우가 많음.



- 입력: $(x_1, \ldots, x_T)$
- 출력: $(y_1, \ldots, y_T)$
  - $h_t = sigmoid(W^{hx}x_t + W^{hh}h_{t-1})$
  - $y_t = W^{yh}h_t$

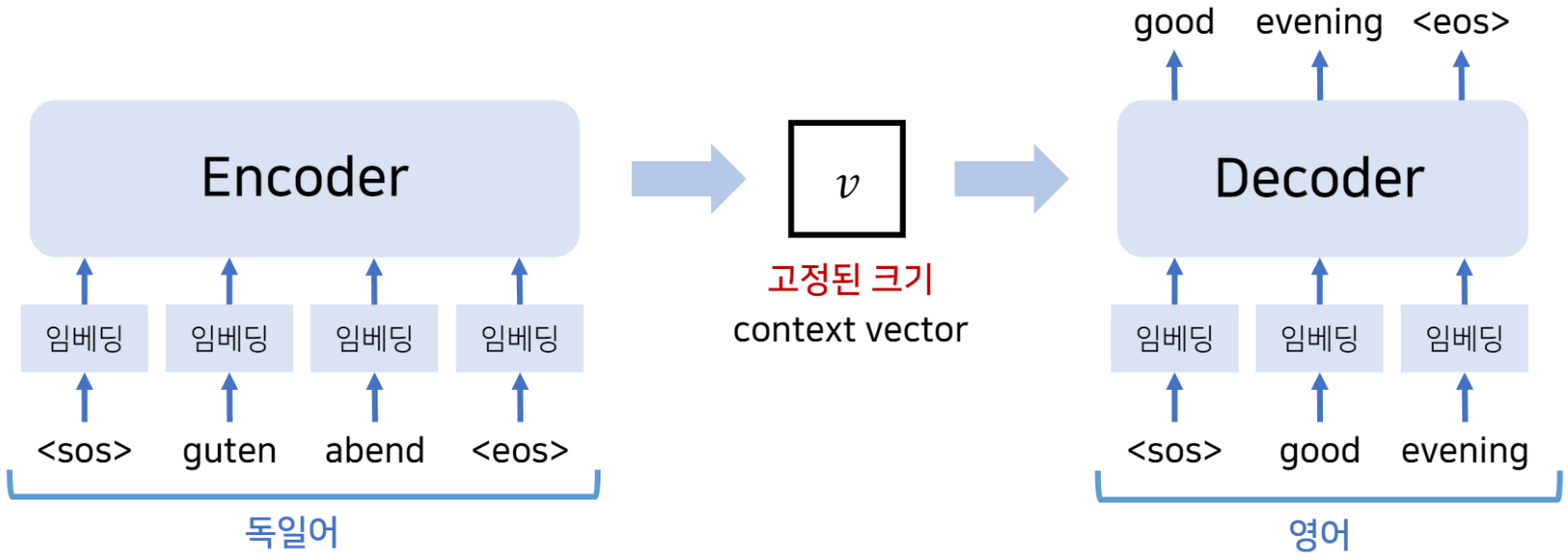


학교	갔었니?	≠	Did	you	go	to	School?

->음성인식, 번역과 같은 sequential data에서 한계 => LSTM으로 해결

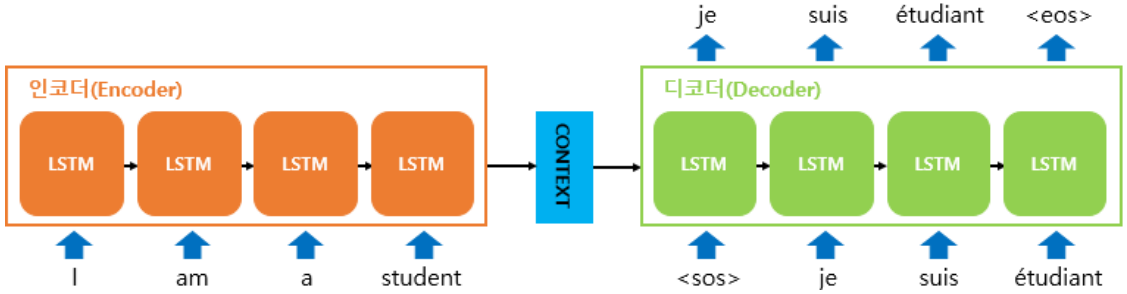
The Model

입력 시퀀스가 하나의 고정된 크기의 벡터로 바꾸는 방법 사용.
인코더가 고정된 크기의 문맥벡터(context vector)를 추출
인코더를 위한 RNN, 디코더를 위한 RNN 따로 사용.

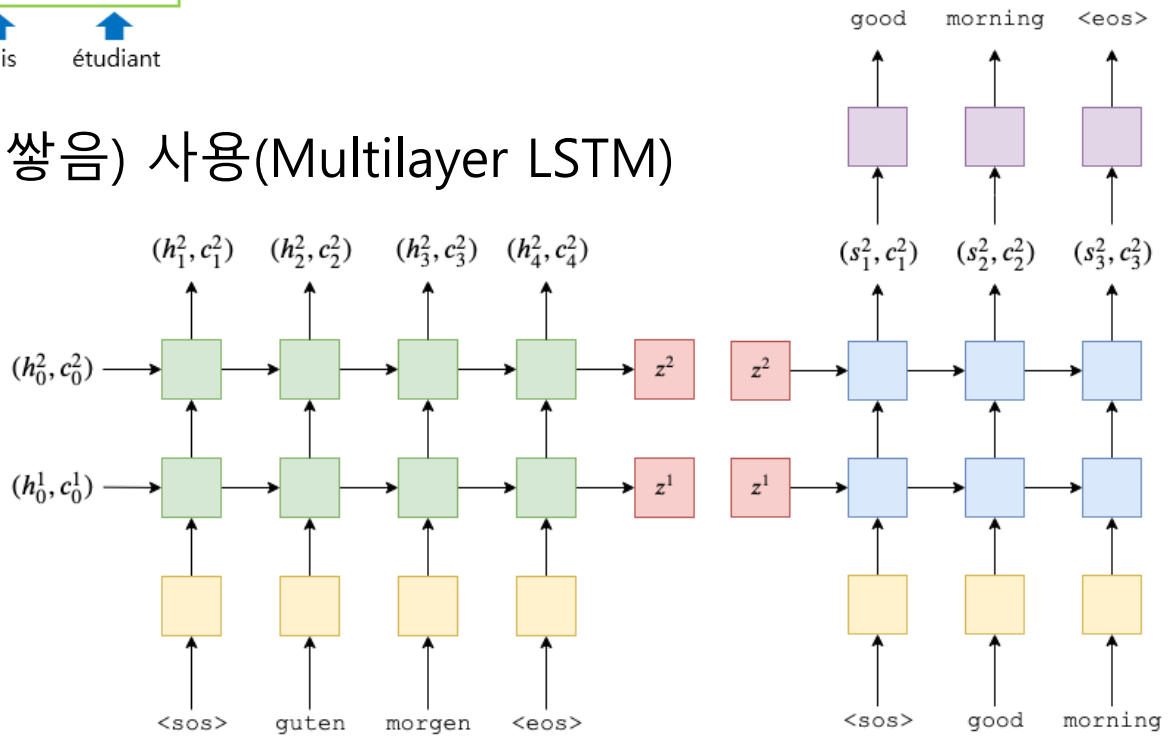


The Model

1. 인코더, 디코더 파트의 각각의 LSTM은 서로 다른 파라미터 사용



2. LSTM은 총 4개의 레이어를 겹쳐(위로 쌓음) 사용(Multilayer LSTM)
옆 사진은 layer를 2층 쌓음



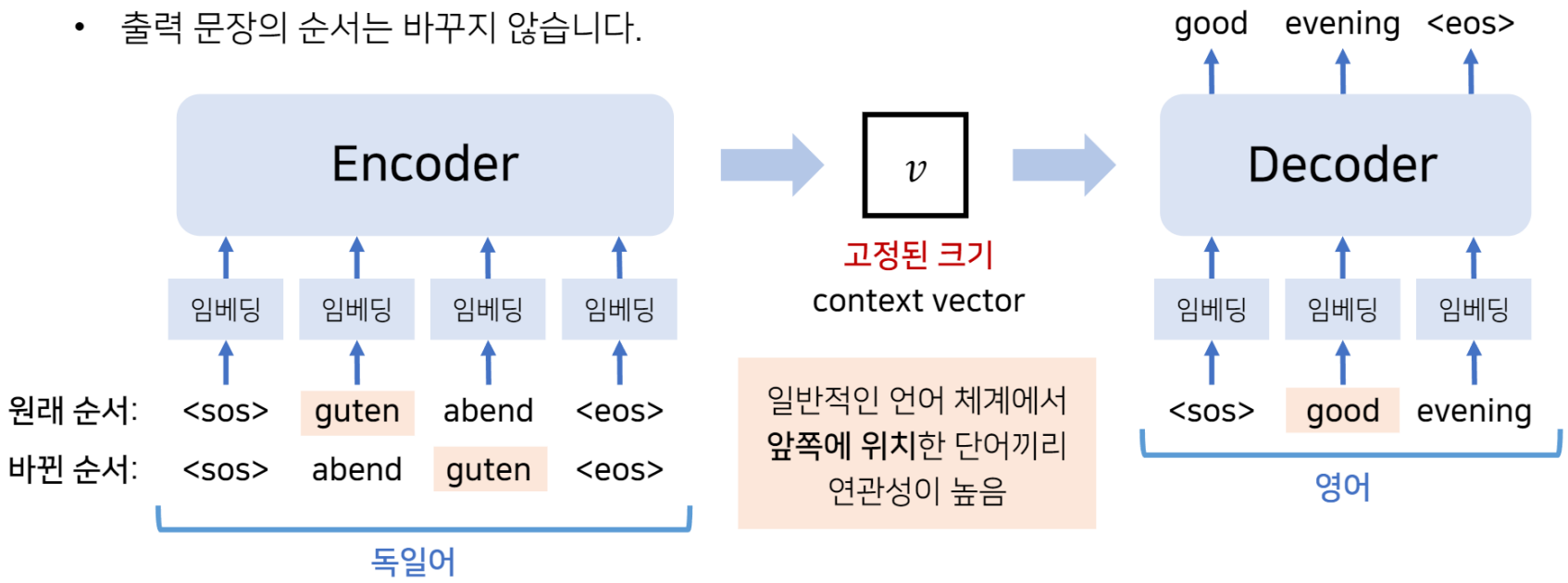
The Model

3. 입력 token의 순서를 바꿨을 때 성능이 더 향상

Input : a, b, c 이고 실제로 모델에 들어갈 때는 c, b, a순으로 들어가게끔.

Decoder 부분의 input이 α, β, γ라고 했을 때. 즉, a는 α와 비슷하게 되므로 상대적으로 a와 α가 높은 연관성을 가져 매핑됨.

학습난이도 낮추므로 좋은 성능 얻음.



Experiments

- English to French(논문)
(실습 : 독일어(src) – 영어(trg))
- BLEU score(기계번역 성능지표)
- WMT14 dataset
160만개의 token(단어). 입력 단어는 160만개의 token 중 하나.
Src와 trg 순서 반대

```
# 학습 데이터 중 하나를 선택해 출력
print(vars(train_dataset.examples[30])['src'])
print(vars(train_dataset.examples[30])['trg'])
```

```
['.', 'steht', 'urinal', 'einem', 'an', 'kaffee', 'tasse', 'einer', 'mit', 'der', ',', 'mann', 'ein']
['a', 'man', 'standing', 'at', 'a', 'urinal', 'with', 'a', 'coffee', 'cup', '.']
```

Conclusion

Result 1.
Baseline model(SMT): 33.3%
LSTM : 34.8%
=> 딥러닝이 통계적 모델보다 성능이 높다는 결과

SMT + LSTM : 36.5%
입력문장을 바꾸는 것이 성능향상에 도움을 준다는 결과

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	**34.81**

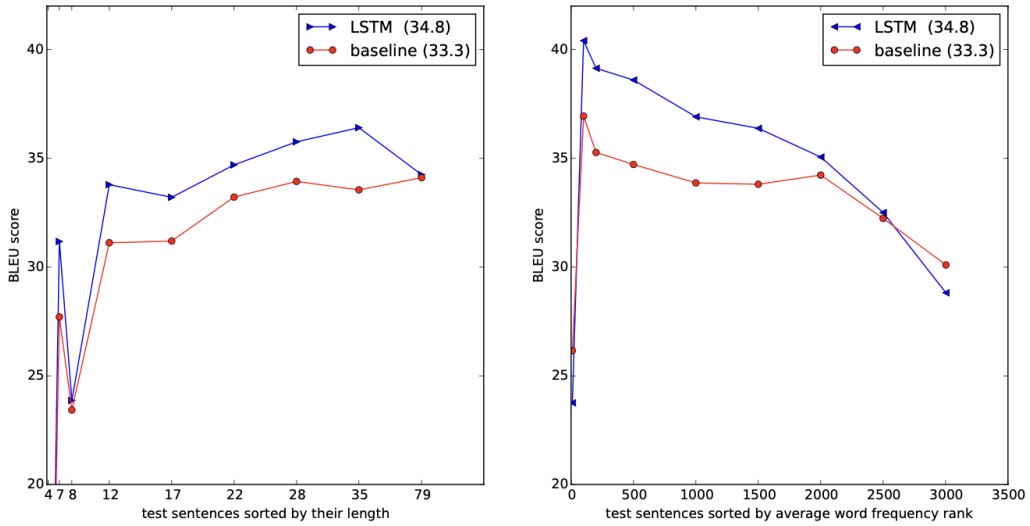
Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	**37.0**
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	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	~45

Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

Conclusion

Result 2.
긴 문장에서도 좋은 성능을 보임



Result 3.
PCA결과 단어의 순서에 따라 민감하지만,
문장의 수동, 능동 형태에는 큰 영향 받지 x