- NLP Paper review -

Sequence to Sequence Learning with Neural Networks

인천대학교 컴퓨터공학부 강병하

논문소개

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

- 『Sequence to Sequence Learning with Neural Networks』(NIPS 2014)
- 구글팀에서 발표
- 19438회 인용 (2022/7/28 기준)
- RNN 기반(LSTM) Sequence to Sequence 아키텍처를 활용한 기계 번역 (영어→프랑스어)
- 『 Attention is All you Need 』(NIPS 2017)이 발표되기 전까지 기계 번역 등의 시퀀스 문제를 해결한 주역

Sequence to Sequence Examples

1. Speech recognition(음성 인식)

“시리야, 노래 틀어줘"

<음성>

<텍스트>

Sequence to Sequence Examples

2. Question answering(질의 응답)

Q : 몸무게가 22t인 암컷 향고래가
500kg에 달하는 대왕 오징어를 먹고
6시간 뒤에 1.3t 알을 낳았다면
이 암컷 향고래의 몸무게는 얼마인가?

A : 고래는 알을 낳을 수 없다.

<질문>

<대답>

Sequence to Sequence Examples

3. Machine Translation(기계 번역)

“how are you?”

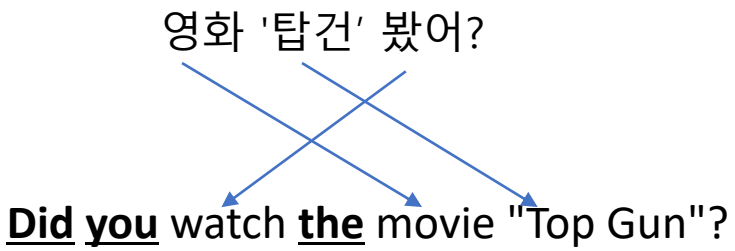
“Comment vas-tu ?”

<영어>

<프랑스어>

Sequence to Sequence Problems

1. 입력 시퀀스와 출력 시퀀스의 길이가 제각각 → 1 대 1로 매핑되지 않음



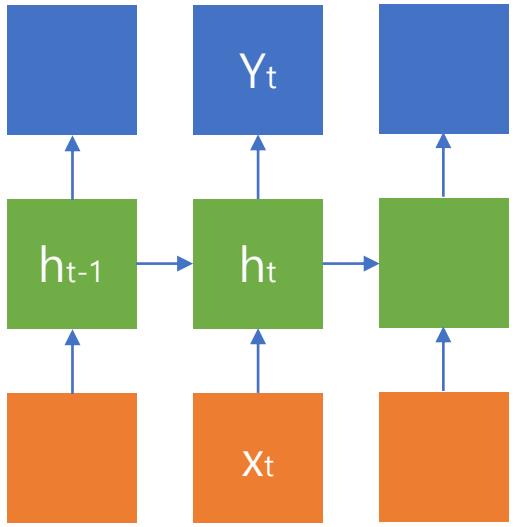
2. 입력 시퀀스와 출력 시퀀스의 영역(domain)이 다름 → 각 영역에 대한 표현 방법 학습 필요

- 음성 → 텍스트
- 질문 → 대답
- 영어 → 프랑스어

RNN

시퀀스와 시퀀스를 매핑하는 데 RNN이 사용됨.

다대다 RNN(Many-to-many)



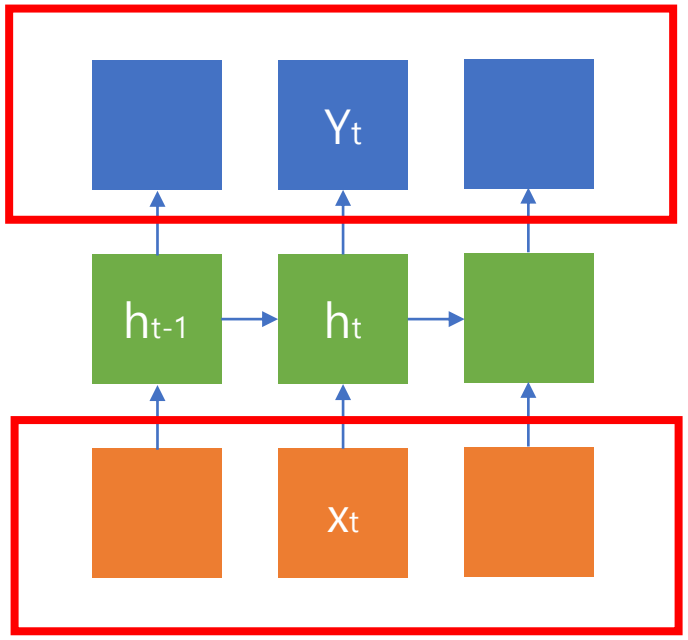
$$y_t = W^{\mathrm{yh}} h_t$$

$$h_t = \mathrm{sigm}\left(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}\right)$$

RNN의 한계

However, it is not clear how to apply an RNN to problems whose input and the output sequences have different lengths with complicated and non-monotonic relationships.

movie ‘Top Gun’ Watch? (X) Did you watch the movie "Top Gun"? (O)



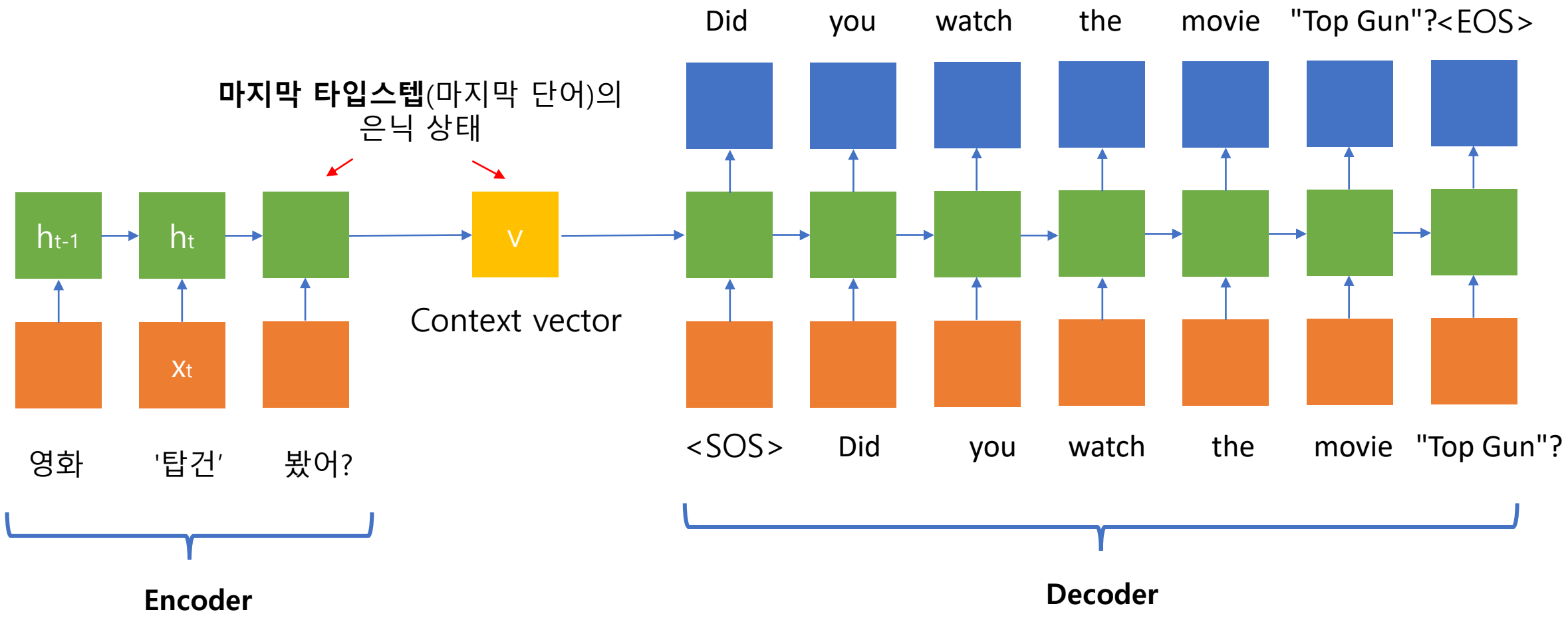
영화 '탑건' 봤어?

→ 기본 RNN 하나만으로 서로 다른 영역의 시퀀스를 매핑하기 어렵다.

영역마다 전혀 다른 **표현 방식(순서)**과 **시퀀스 길이**를 가지기 때문

Seq2Seq : 모델

인코더 RNN과 디코더 RNN으로 구성

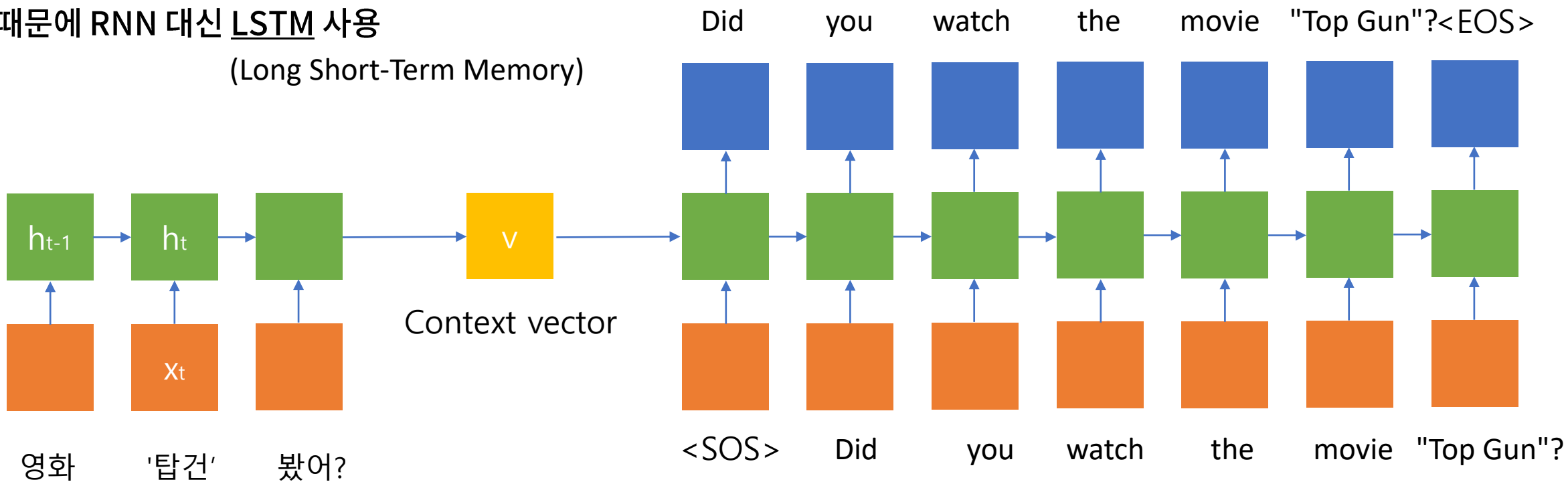


Seq2Seq : 모델

long term dependencies(장기 의존성 문제)

때문에 RNN 대신 LSTM 사용

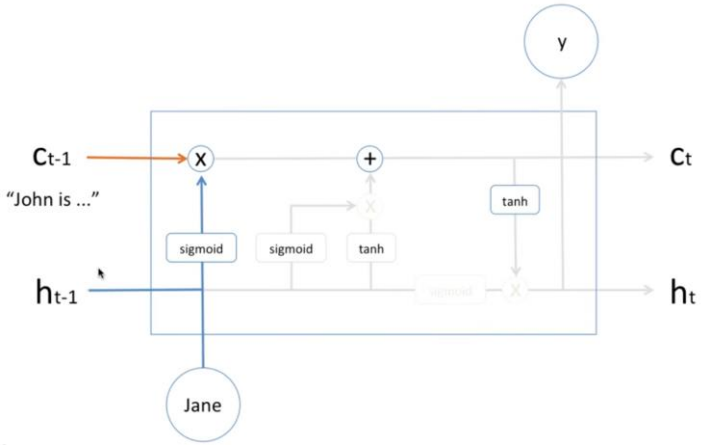
(Long Short-Term Memory)



LSTM(Long Short-Term Memory)

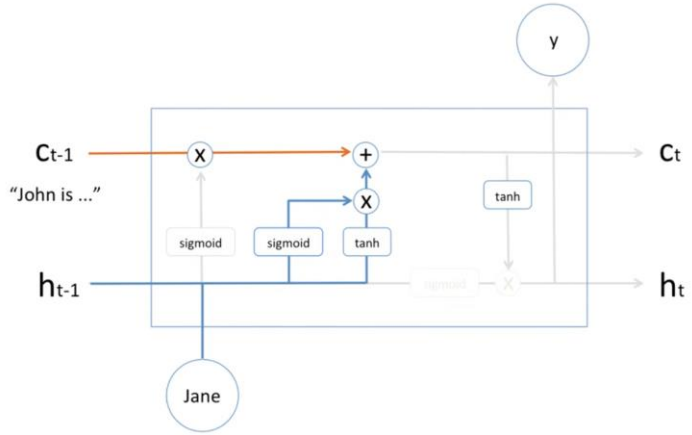
장기 의존성(long-term dependencies) 대안

RNN에 <u>긴 시간 동안의 정보</u>를 기억하는 Cell state 추가



<forget gate>

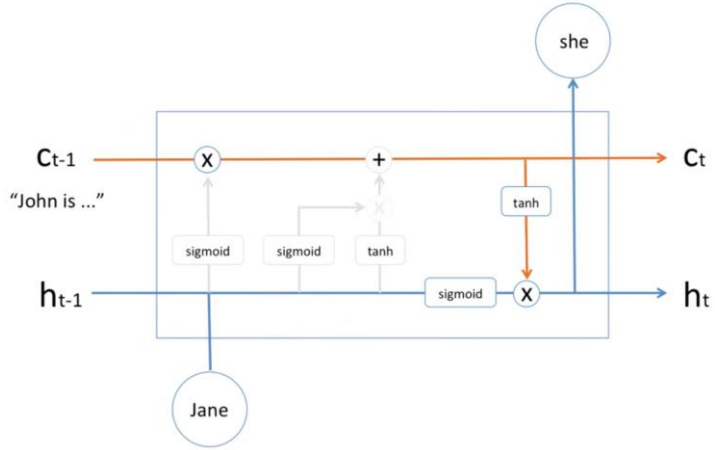
Cell State에서 어떤 정보를 버릴까?



<input gate>

새로운 정보 중 어떤 것을

Cell state에 저장할까?



- 업데이트 된 Ct출력
- Cell State의 어느 부분을 ht로 내보낼까?

LSTM in Seq2Seq

we used two different LSTMs: one for the input sequence and another for the output sequence

- 서로 다른 파라미터를 가지는 LSTM이 인코더와 디코더에 사용됨

We used deep LSTMs with 4 layers,

- 4개의 은닉층을 가진 깊은 LSTM 사용

each additional layer reduced perplexity by nearly 10%,

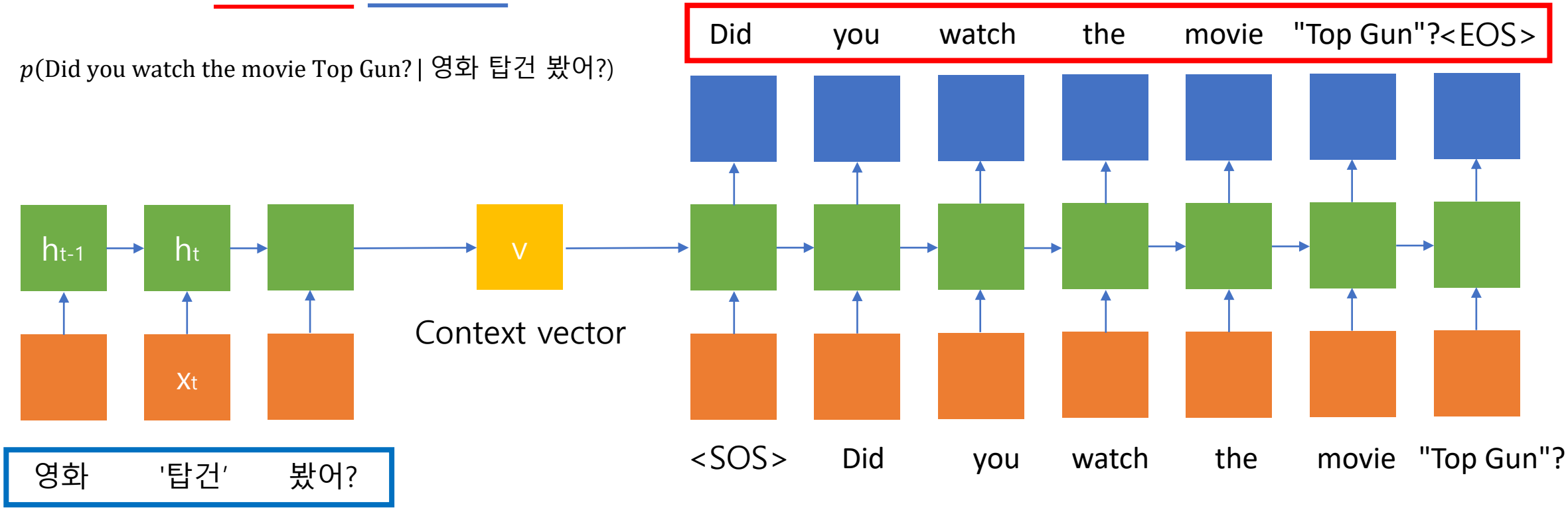
- 추가된 각 은닉층마다 perplexity 10% 감소

헷갈리는 정도
(몇 개의 선택지를 가지고 고민하고 있는지)

Seq2Seq : 모델

조건부 확률 $p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T)$

p(Did you watch the movie Top Gun? | 영화 탑건 봤어?)



Background - 언어 모델(Language model)

단어 시퀀스에 확률을 할당하는 모델

→ 가장 자연스러운 단어 시퀀스를 찾는 것이 목적

- 통계를 이용한 방법 (SMT)
- 인공 신경망을 이용한 방법

p(나는 탑건을 재밌게 봤다) $>$ p(나는 탑건을 재밌게 보였다)

확률을 어떻게 할당할까?

→ 이전 단어들이 주어졌을 때, 모델이 **다음 단어 예측** 하도록 함

Background – 언어 모델(Language model)

p(나는 탑건을 재밌게 봤다) $= p$(나는, 탑건을, 재밌게, 봤다)

- 나는 탑건을 재밌게 봤다

p(봤다 | 나는, 탑건을, 재밌게) → **조건부 확률**

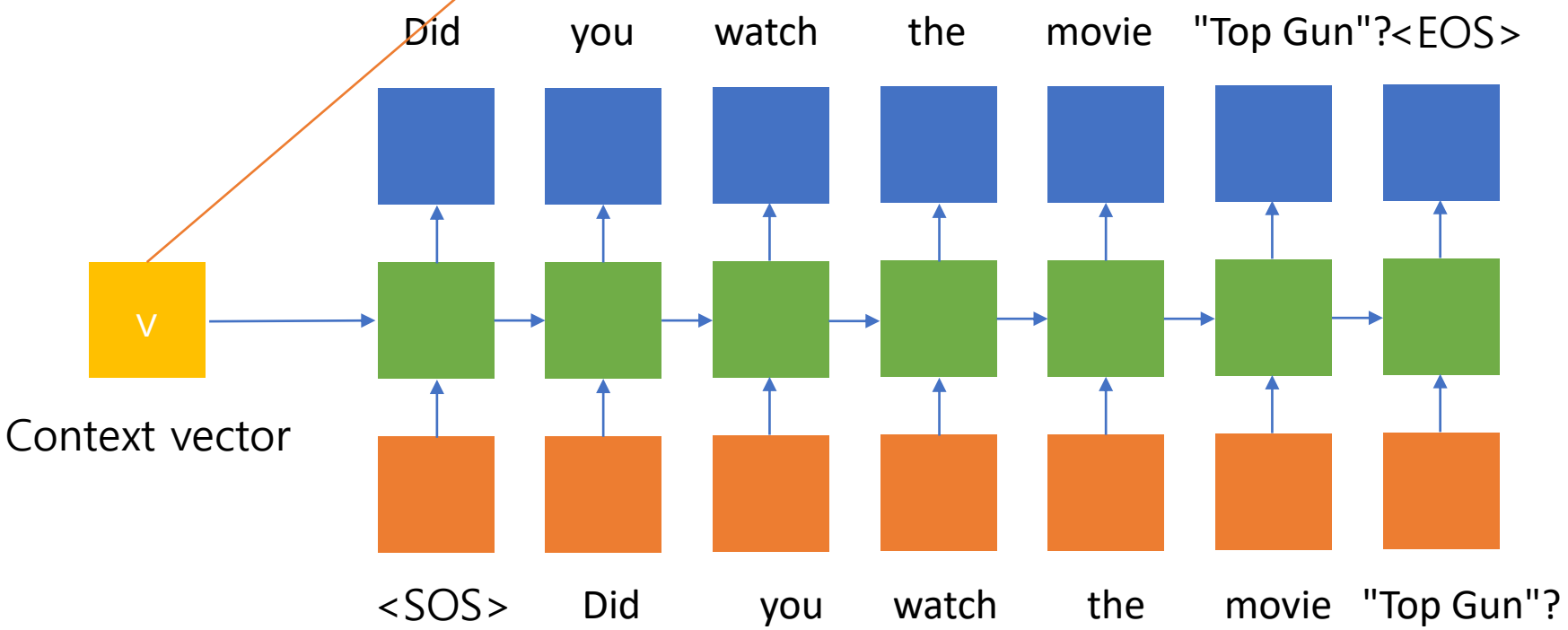
p(나는 탑건을 재밌게 봤다)

$= p$(나는) $*$ p(탑건을 | 나는) $*$ p(재밌게 | 나는, 탑건을) $*$ p(봤다 | 나는, 탑건을, 재밌게)

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

Seq2Seq : 모델의 목표 공식(formulation)

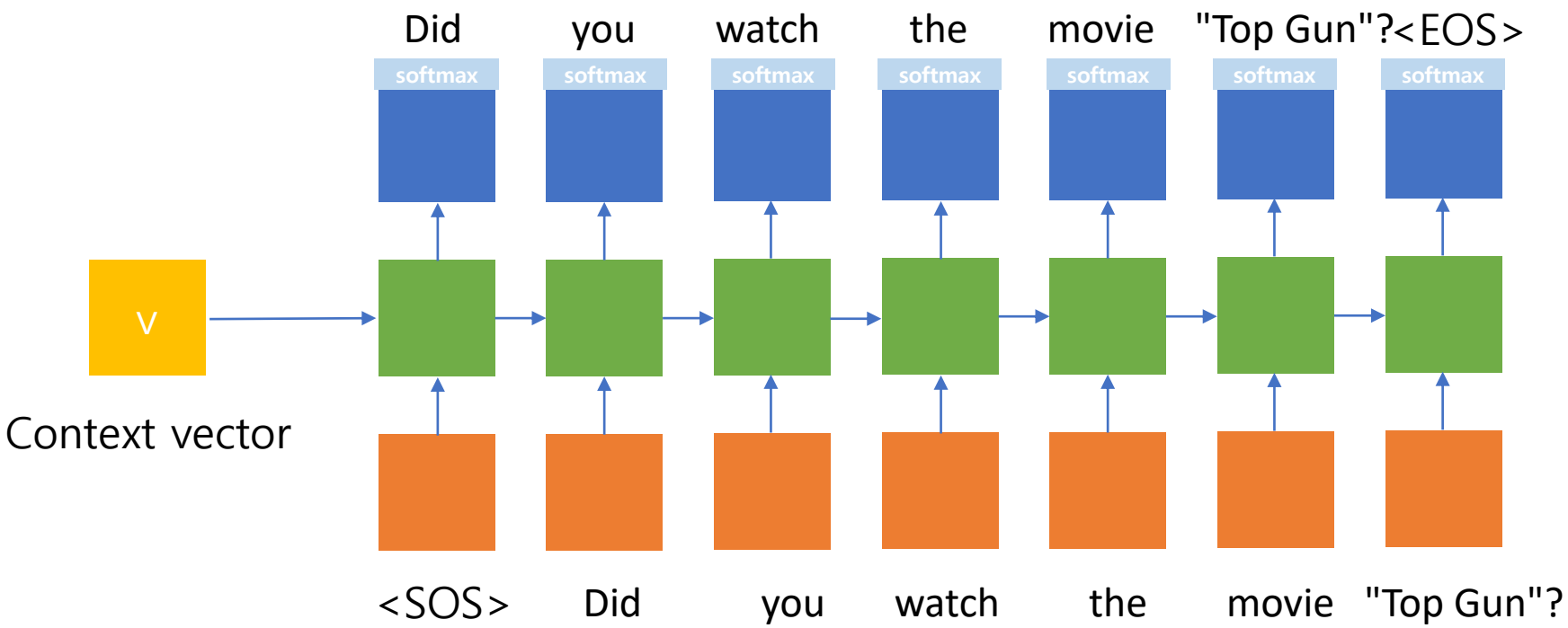
$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$



Seq2Seq : 모델의 목표 공식(formulation)

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

In this equation, each $p(y_t | v, y_1, \ldots, y_{t-1})$ distribution is represented with a softmax (



Seq2Seq : Training objective

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

타겟 문장(올바른 번역)

소스 문장

훈련 세트

- 소스 문장에 대한 타겟 문장의 로그 확률을 최대화하는 것이 목표
- 평균적으로 높은 성능을 위해 훈련 세트로 로그 확률의 총합을 나눔(확률의 평균)

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

We search for the most likely translation using a simple left-to-right beam search decoder

Background – Greedy Search

각 타임스텝마다 가장 가능성(확률)이 높은 단어를 선택

- Did you watch the _____

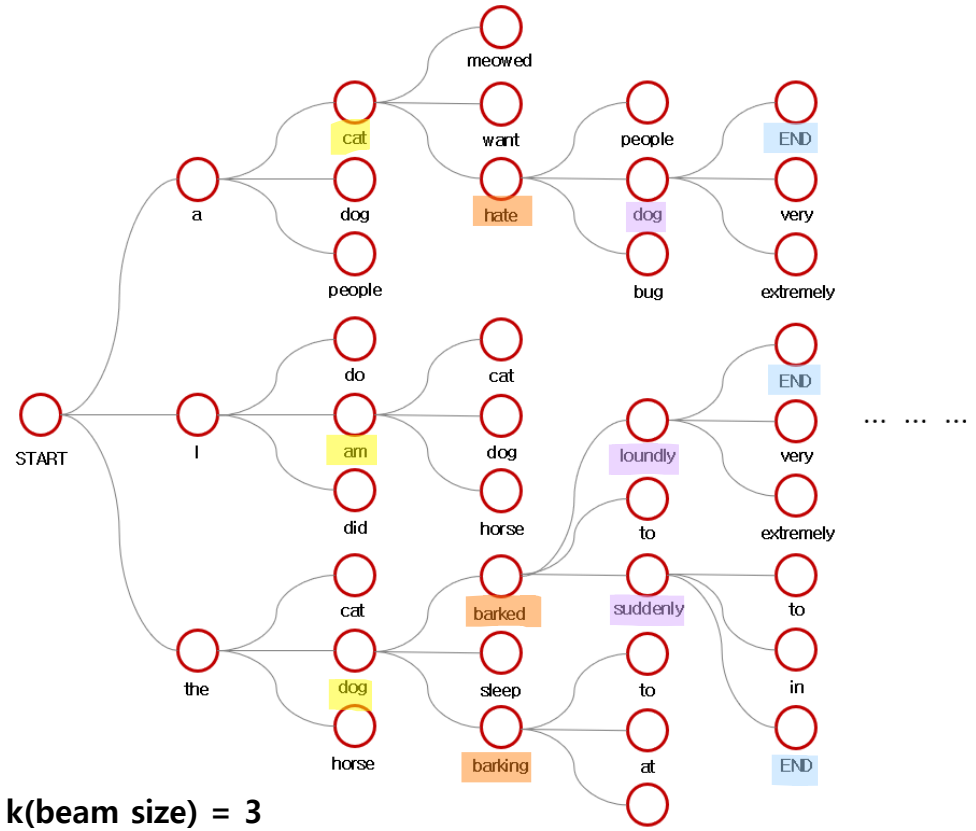
0.48	movie
0.12	happy
0.26	soccer
0.05	run
0.45	sport

확률 분포 상에서 상위 2등은 제외

→ 확률이 근소한 차이라면 2등이 올바른 예측일 경우도 고려해야 한다.

Background – Beam Search

각 타임스텝마다 확률이 높은 k개를 골라 누적확률이 높은 시퀀스를 선택



- k^2의 자식 노드 중 누적확률 순으로 k개 선택
- 뽑힌 k개의 각 노드에서 다시 k개의 자식 노드 선택
- <EOS>를 만난 빔이 k개가 될 때까지 반복
- k개의 후보 중 가장 누적 확률이 높은 빔을 최종 선택

Dataset

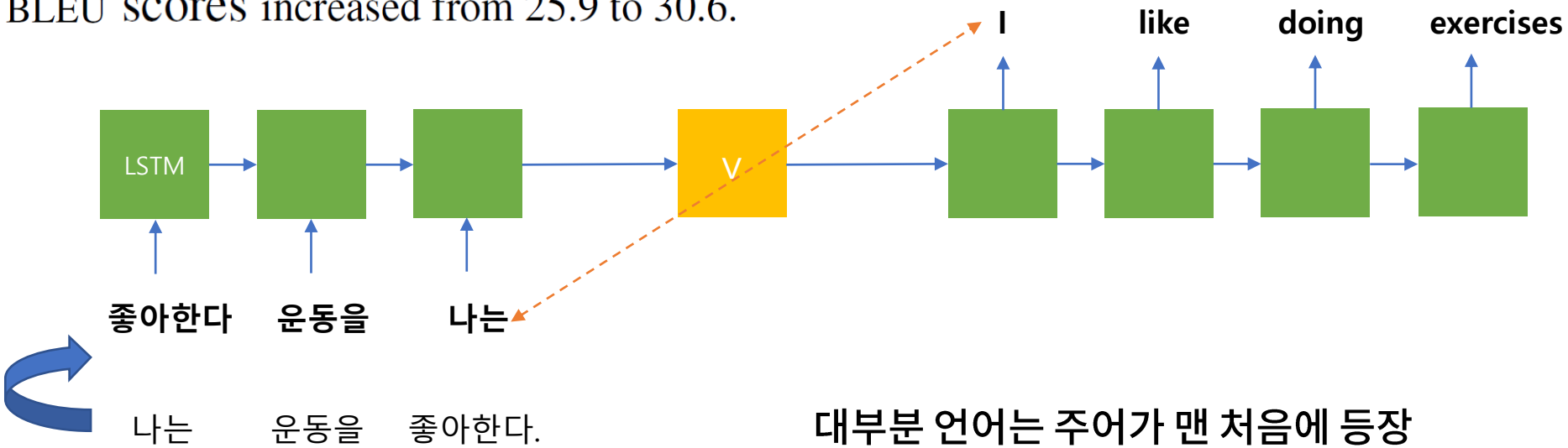
WMT'14 English to French dataset

- 2백만 문장(3억 4천 8백만개의 프랑스 단어, 3억 4백만개의 영어 단어로 구성)
- 고정된 크기의 어휘 사전 사용
  - 소스 어휘사전 – 가장 자주 등장한 160,000 단어
  - 타겟 어휘사전 – 가장 자주 등장한 80,000 단어
- **어휘 사전에 없는 단어는 "UNK" 토큰으로 교체**

소스 문장 순서 뒤집기(Reversing the Source Sentences)

문장의 순서를 거꾸로 입력했을 때 더 좋은 성능

perplexity dropped from 5.8 to 4.7
BLEU scores increased from 25.9 to 30.6.



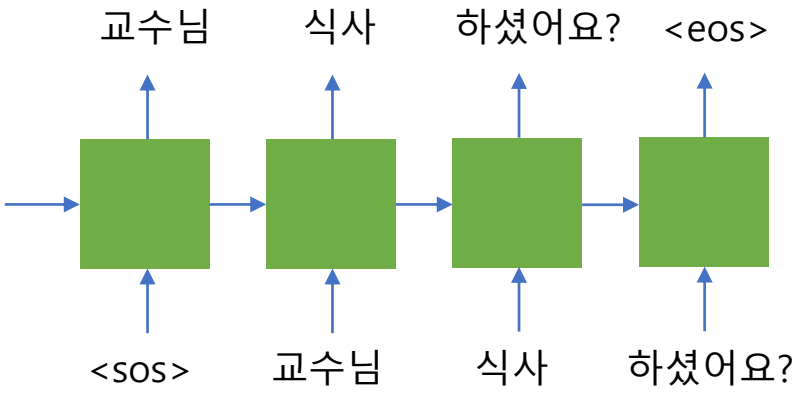
대부분 언어는 주어가 맨 처음에 등장

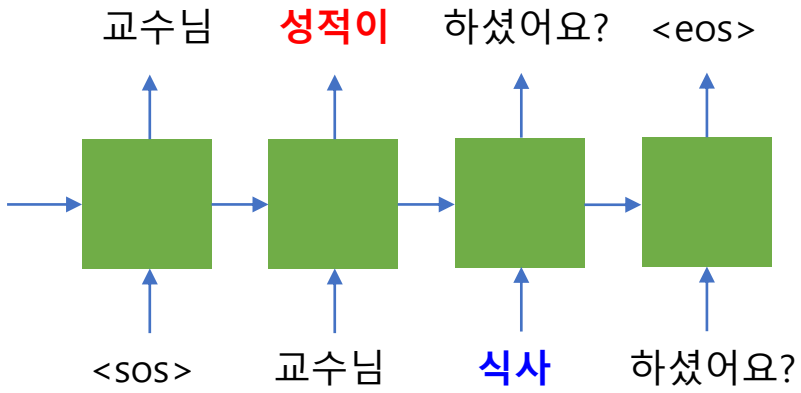
→ 많은 short term dependencies를 도입

Teacher-Force Training

한 번 예측을 잘못하면 전체 시퀀스의 예측이 엉망이 될 가능성 (조건부 확률이므로)

→ 강제로 정답(실제 목표 출력, Ground Truth)을 입력





학습 초기에는 모델 성능이 낮기 때문에 잘못된 예측값으로 은닉 상태가 업데이트 됨

→ Teacher-Force을 사용하면 **학습 속도 빨라짐**

기계 번역 성능 평가

BLUE score 비교

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**

- Forward LSTM < Reversed LSTM
- Single LSTM < Ensemble LSTM
- Beam size 2 < Beam size 12

비용측면에서는 Ensemble of 5 reversed LSTMs, beam size 2 이

Single reversed LSTM, beam size 12 보다 저렴

기계 번역 성능 평가

다른 방법들과 BLUE score 비교

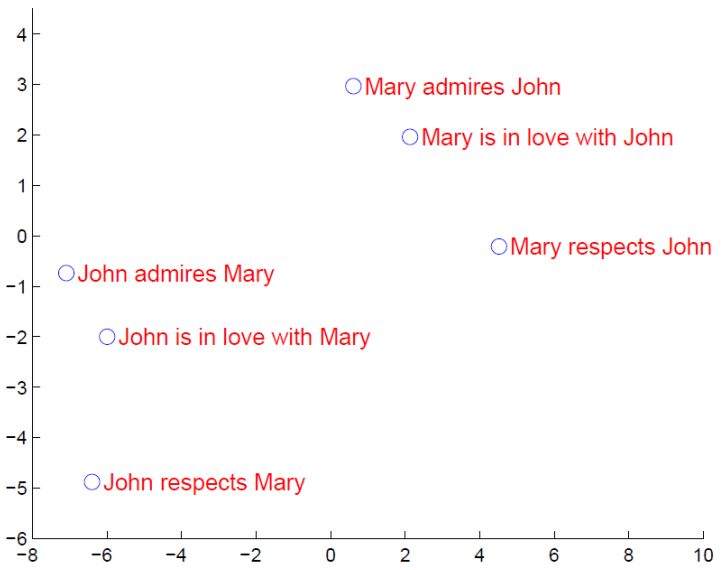
Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	∼45

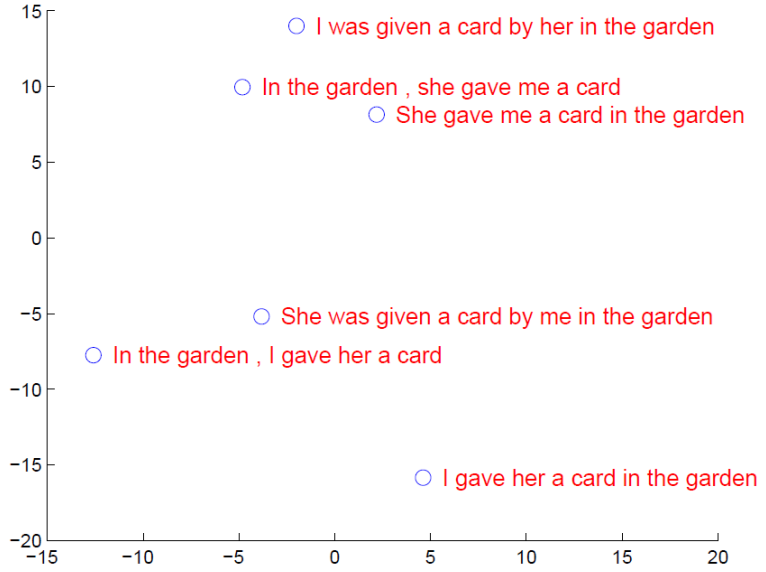
Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

➔ 제약(제한된 크기의 어휘사전)에도 불구하고 SOTA에 근접한 BLUE score 기록

모델 분석 – 문장 순서, 수동태-능동태 민감도

2-dimensional PCA projection → 학습된 표현(고정된 크기의 벡터) 시각화

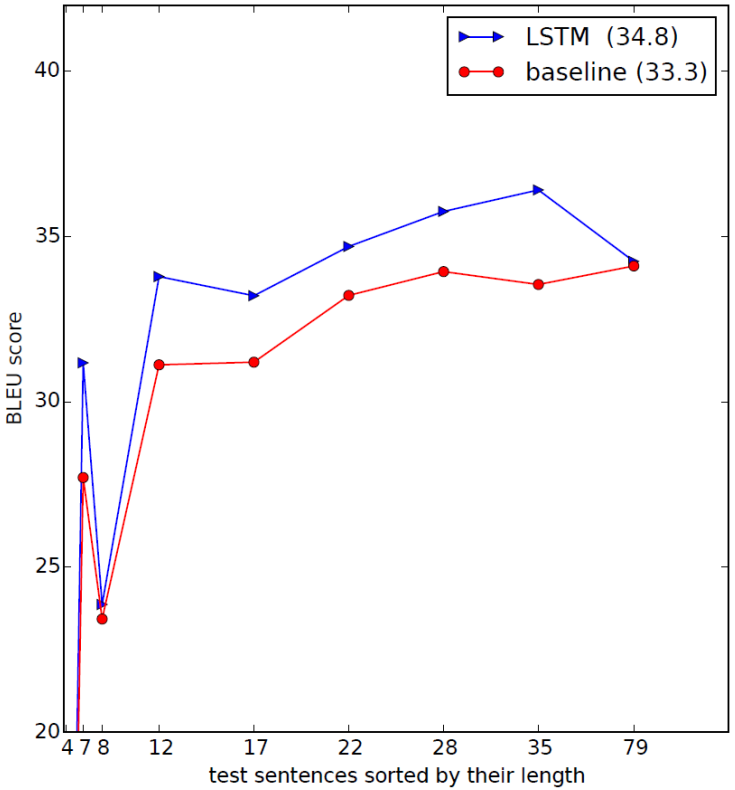




문장 순서의 변화 = 의미의 변화 → 민감하게 반응

수동 <-> 능동의 변화 = 표현 형태의 변화 → 둔감하게 반응

모델 분석 – 긴 문장에 대한 성능



단어 개수에 따른 BLUE score 변화

→ 35개의 단어 밑으로는 번역 성능이 감소하지 않음

→ RNN 기반 LSTM이 비교적 긴 문장을 처리할 수 있음을 보임

결론

- 비교적 간단한(제한된 어휘사전) LSTM 기반 기계 번역이 기존 SMT(통계적 기계 번역)을 능가할 수 있음을 보임

  → 실제로 기계 번역을 급속도로 발전시키는 데 기여(Transformer(2017) 전까지 SOTA)

- RNN 구조에서 입력의 시퀀스의 순서를 바꾸는 것이 성능을 향상 시킬 수 있음을 발견
- LSTM이 장기 의존성에 강하다는 것을 보임
- 기계 번역 뿐만 아니라 다양한 분야의 시퀀스 투 시퀀스 문제 해결에 적용
  - 음성인식
  - 챗봇
  - 내용 요약