Sequence to Sequence Learning with Neural Networks

저자: Ilya Sutskever, Oriol Vinyals, Quoc V. Le (Google, 2014)

2025.03.31

발표자 : 양희원

목차

연구 배경

✅ 기존의 문제점

- 전통적인 RNN이나 n-gram 모델은 **고정된 길이의 입력/출력만 처리 가능**
- 문장 단위 번역 같은 **가변 길이 입력/출력 문제 해결이 어려움**
- 기존의 방법들은 **장기 의존성**(Long-term dependency)**을 잘 학습하지 못함**
- End-to-end approach를 통한 모델 학습

✅ 이 논문이 해결하려는 문제

- RNN을 활용하여 **입력과 출력 길이가 다른 시퀀스 변환 문제를 해결하는 방법** 제안

모델 소개

✅ 기본 아이디어

- 하나의 RNN이 **입력을 인코딩**하고,
- 또 다른 RNN이 **출력을 디코딩**하는 **Encoder-Decoder 구조** 제안
- **입력 Sequence를 반대로 입력하여 단기 의존성 학습 강화**

✅ 구조 설명

- **Encoder:** 입력 시퀀스를 **고정된 길이의 벡터(Context vector)** 로 변환
- **Decoder:** 이 벡터를 기반으로 **출력 시퀀스를 생성**
- 두 RNN을 연결하여 **입력과 출력의 길이가 달라도 학습 가능**

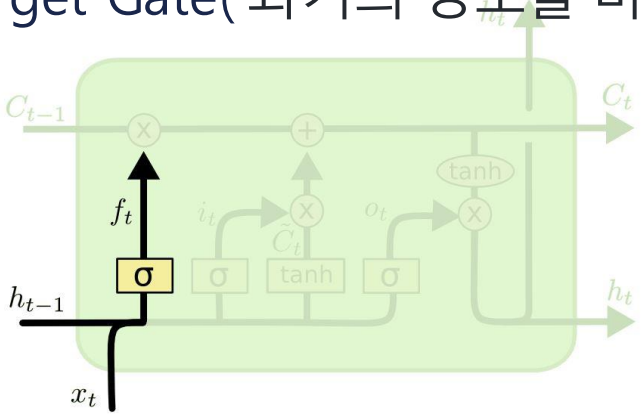
✅ 핵심 특징

- **하나의 문장을 하나의 벡터로 압축**하여 의미를 학습
- **가변 길이 시퀀스를 처리 가능**

LSTM 구조

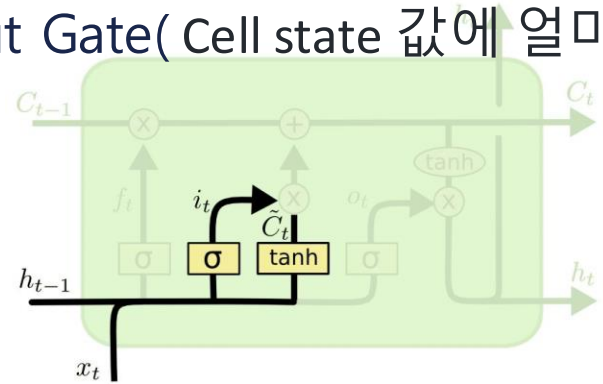
C_t는 t일 때 cell state, x_t는 t일 때 입력 값
h_t는 t일 때 hidden state, i_t는 입력 게이트

Forget Gate(과거의 정보를 버릴지 말지 결정하는 과정



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Input Gate(Cell state 값에 얼마나 더할지 말지 현재 정보 기억)



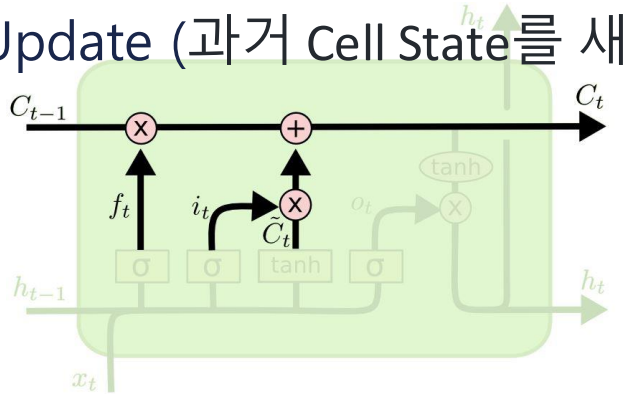
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LSTM 구조

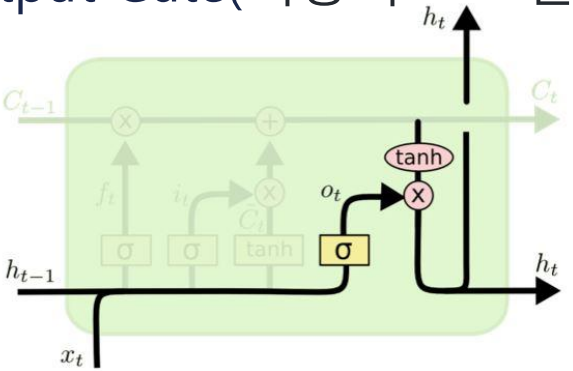
C_t는 t일 때 cell state, x_t는 t일 때 입력 값
h_t는 t일 때 hidden state, i_t는 입력 게이트

Update (과거 Cell State를 새로운 State로 업데이트)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

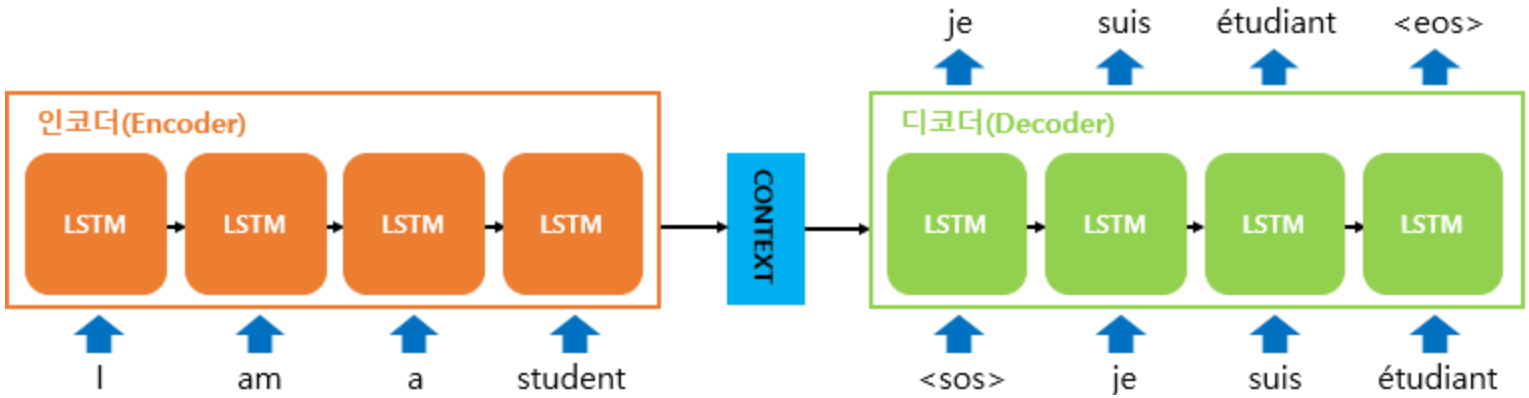
Output Gate(최종적으로 얻어진 Cell State 값을 얼마일지 결정)



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

모델 구조



✅ 모델 아키텍처

- **Encoder:** 여러 개의 RNN 셀 (LSTM) → 마지막 hidden state를 Context Vector로 사용
- **Context**: 입력 문장의 의미를 담고 있는 fixed-dimensional representation
- **Decoder:** Context Vector를 받아 첫 hidden state로 설정 → RNN을 통해 (이전 hidden state와 context vector로 하나씩 단어 생성)

학습 목표

✅ LSTM의 목표는 조건부 확률 구하기!

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

(x1, . . . , xT) 입력 sequence , V는 context vector

(y1, . . . , yT ′) 출력 sequence

입력 sequence를 주었을 때 출력 sequence가 나올 확률을 구하는 것 -> 전체 번역의 확률

각 vector는 독립적이지 않기 때문에 조건부 확률 사용

학습 목표

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

$$p(y_t | v, y_1, \ldots, y_{t-1})$$

y_t는 distribution을 softmax로 사용한다.
디코더는 각 시점 t에서 softmax를 사용하여 다음 단어 y_t의 확률 분포를 계산한다.
하지만 일반적인 RNN 구조에서는 시간이 지남에 따라 과거 정보가 사라지는 **장기 의존성 문제**가 발생할 수 있다.
이 문제는 softmax 때문이 아니라 RNN의 gradient 문제에서 기인한다.
이를 해결하기 위해 LSTM 구조를 도입하여 더 오랜 시간 동안 정보를 유지할 수 있도록 한다.

Experiment (역방향 sequence)

2. 한국어-영어 번역:

- **원문 (한국어):** "고양이가 방석 위에 앉았다."
- **정방향 입력:** `고양이가 방석 위에 앉았다 .`
- **역방향 입력:** `. 앉았다 위에 방석 고양이가`
- **번역 (영어):** "The cat sat on the cushion."

제일 중요한 파트!

1) 입력 Sequence를 역방향으로 입력함에 따라 단기 의존성을 더 잘 학습하도록 돕고, 기울기 소실 문제를 해결함
2) 입력과 출력 간의 establish communication(서로 매치 되는 벡터들끼리 가까워짐)이 생성을 위해 필요한 정보를 빨리 얻어서 Memory활용도가 높아짐

Experiment (Decoding 과정)

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

$$\downarrow$$

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

A. 학습 목표는 소스문장 S와 번역 T가
나올 확률을 최대한 높이기 (학습 목표는 주어진
원문에 대해 올바른 번역문이 생성될 확률을 최대화하는 것입니다.)

B. Beam search를 통해 출력 sequence를 조절
(B=3)이라고 설정
1. 첫 번째 단어로 "the", "a", "an“ 이 선택
2. (|V|) => 10000개는 어휘 집합의 크기를 통한 가능한 조합을 생성! 3*10000개 라고 생각
3. 30,000개의 조합 중에서 가장 유망한 (B)개의 조합만 선택
<EOS>토큰 나올 때까지 반복
4. 토큰 나오면 complete set에 추가

Experiment (Training details)

모델 구조: 4개의 LSTM 층, 각 층은 1000개의 셀, 1000차원 단어 임베딩 사용.
입력 어휘 160,000개, 출력 어휘 80,000개.

초기화: LSTM 파라미터는 -0.08 ~ 0.08 사이의 uniform distribution으로 초기화.

최적화 알고리즘: 모멘텀 없는 stochastic gradient descent 사용. Learning rate = 0.7 (5 epoch 후 0.5 epoch마다 절반으로 감소). 총 7.5 epoch 학습.

배치 크기: 128개 시퀀스(문장)를 배치로 사용. Gradient는 배치 크기로 나누어 계산.

Gradient Clipping: Exploding gradients 방지를 위해 gradient norm이 5를 넘으면 scaling

Minibatch 구성: Minibatch 내 문장 길이를 비슷하게 유지하여 계산 효율성 향상.

Experiment Results

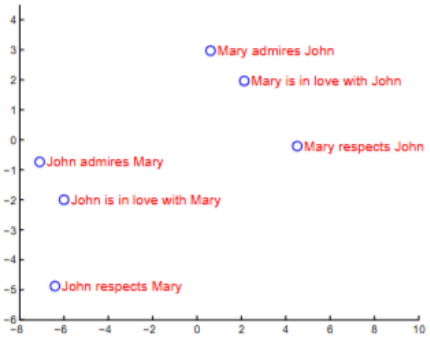
Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size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

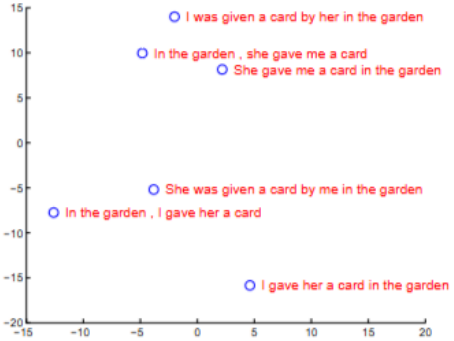
Beam size랑 reverse 입력 Sequence를 사용하면 성능이 좋아짐

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT'14 result [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

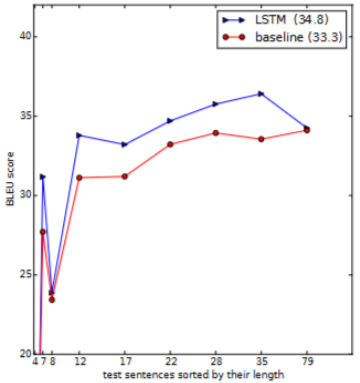
Forward , Reverse, Multi-layer일 수록 성능이 좋아짐

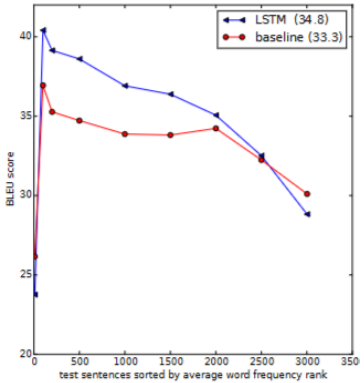
Experiment Results





<PCA>능동태와 수동태(주어, 목적어)가 바뀐 것도 학습을 잘하고 비슷한 의미를 가진 문장들끼리 clustering잘함





길이와 단어 빈도를 통한 학습을 했을 때 LSTM이 성능이 좋았음

결론

핵심 발견: 제한된 어휘와 최소한의 가정만으로도 대규모 딥 LSTM이 기존 SMT 시스템을 능가할 수 있음을 입증했습니다.

주요 시사점: LSTM 기반 접근 방식은 다른 시퀀스 학습 문제에도 적용 가능하며, 소스 문장 단어 순서 반전이 성능 향상에 크게 기여합니다.

향후 전망: LSTM의 긴 문장 처리 능력과 추가적인 최적화를 통해 번역 정확도를 더욱 향상시킬 수 있습니다.