Sequence to Sequence Learning with Neural Networks

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le.
Advances in neural information processing systems 27 (2014).

2024.01.18
유하영

- 이전 연구의 한계

DNN

입력과 출력의 차원이 고정된 문제에서만 사용 가능

sequence와 같이 가변적인 길이의 데이터를 처리하는 데에는 제한적
(입력 시퀀스와 출력 시퀀스의 길이가 다른 경우.
ex) speech recognition, machine translation)

RNN

시퀀스를 처리할 수 있는 **DNN**의 **generalization version**
timestep t에서 **RNN**은 입력 x_t와 이전 **timestep**의 **hidden state** h_{t-1}을 함께 고려하여 y_t를 계산

이 역시 가변적인 길이의 데이터 처리에는 제한적

• RNN based seq2seq

두 개의 **RNN**을 연결하는 **Encoder-Decoder** 구조로
가변길이의 입,출력 문제 해결이 가능하다고 봄

- encoder를 통하여 입력 시퀀스 $(x_1, \dots, x_T)$를 고정된 길이의 벡터 C(context vector, 입력 데이터의 요약)로 표현하고 모델의 다음 부분인 decoder로 전달
- decoder에서는 고정된 길이의 벡터 C 를 출력 시퀀스 $(y_1, \dots, y_{T'})$로 mapping한다.

- 각 입출력 시퀀스의 종료는 보통 <EOS> 토큰을 기준으로 정해짐

- **RNN based seq2seq**

  Encoder-Decoder 구조는 일반적인 언어모델을 사용하여 다음과 같은 조건부 확률로 모델링할 수 있다.

  (우변은 모든 **vocabular**에 대한 **softmax probability**로서 표현된다**.)**

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

- 본 논문은 기존 **RNN based seq2seq** 모델의 성능을 다음 세 가지 방법으로 개선한다.

- **LSTM**

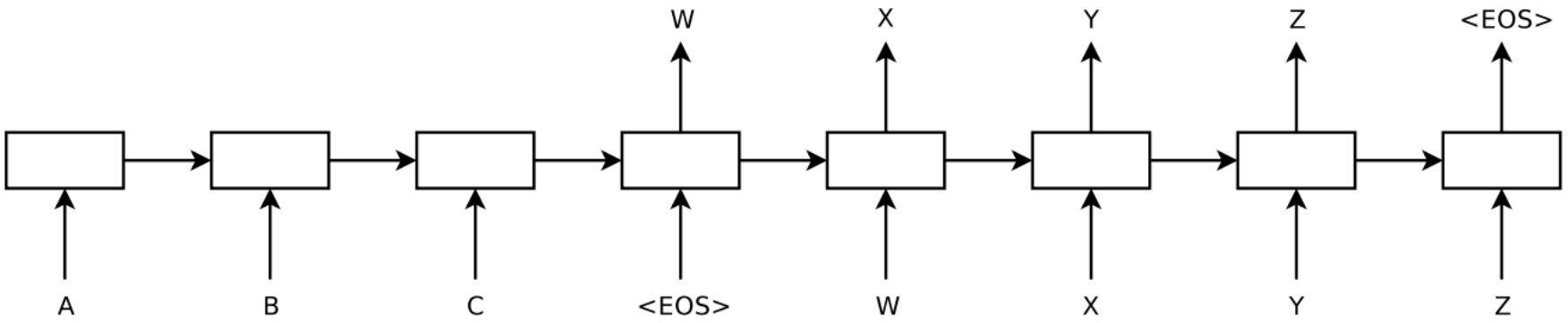
  1. 입력과 출력에 대한 서로 다른 **2개의 LSTM**을 사용한다.

  2. **Deep LSTM**을 사용하여 **Shallow LSTM** 보다 좋은 성능을 제공한다.

  3. **입력의 순서를 뒤집어서 제공**한다. **(reversed order Input sequence)**

1. LSTM 사용

- **LSTM**은 **RNN**의 한계인 **long term dependency**을 개선할 수 있다.
  시퀀스가 길어져 **timestep**의 간격이 멀어질수록 **input sequence**의 정보를 모델이 제대로 반영하지 못하는 점



2. Deep LSTM

- **single layer** 대신 **Deep LSTM** 사용
- 본 논문에서는 **layer 4 deep LSTM**을 사용하였다.

3. Reversed order Input sequence

- **Input sequence**의 순서를 거꾸로 하여 **LSTM Encoder**의 새로운 입력으로 사용

A B C -> D E F

C B A -> D E F

- 입력과 출력 시퀀스 간의 평균거리를 유지하면서, 시퀀스의 각 요소 간의 최소 거리는 훨씬 줄어든다.
- **특히, 입력 시퀀스와 출력 시퀀스 간의 직접적인 대응 관계가 있는 경우(예 : 기계번역)에 유용함**

Ex) I am happy → 나는 행복하다
happy am I → 나는 행복하다

- 즉, **시퀀스의 시작 부분과 끝 부분 사이의 거리를 줄이는 것이 이 기법의 핵심**

??? '양방향 LSTM'과 'Reversed order Input sequence' 와의 차이

- **Reversed order Input sequence**는 단순히 입력 시퀀스의 순서를 바꾸는 것
- 즉, 여전히 단방향 **LSTM**을 사용함

A B C → D E F **->** C B A → D E F

- 반면, **Bi-LSTM**은 각 시점에서 입력 시퀀스의 양방향(정방향, 역방향) 정보를 동시에 고려

A B C → D E F

A 시점 – 정방향 **LSTM** : "A"까지의 정보만 고려
– 역방향 **LSTM** : "C B A"의 정보를 고려
(두 정보는 **A** 시점에서 결합)

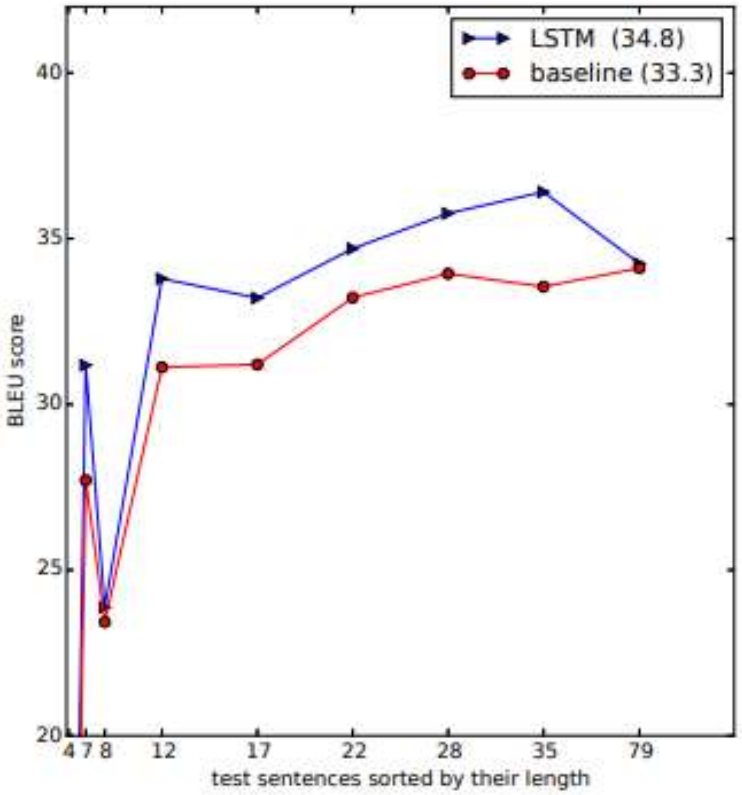
B 시점 – 정방향 **LSTM** : "A B"까지의 정보를 고려
– 역방향 **LSTM** : "C B"까지의 정보를 고려
(두 정보는 **B** 시점에서 결합)

C 시점 – 정방향 **LSTM** : "A B C"까지의 정보를 고려
– 역방향 **LSTM** : "C"까지의 정보를 고려
(두 정보는 **C** 시점에서 결합)

Experimental Results

- 각 **LSTM layer** 마다 서로 다른 초기화 과정을 거치고 **minibatch**를 랜덤하게 섞어 학습한 **LSTM** 앙상블로부터 **best result**를 얻었다.

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**



- **BLEU(Bilingual Evaluation Understudy)**

기계 번역 결과와 사람이 직접 번역한 결과가 얼마나 유사한지 비교하여 번역에 대한 성능을 측정하는 방법

** Greedy Searching

** Beam search

- **Beam search :** 기계 번역이나 자연어 처리에서 사용되는 탐색 기법. 번역이나 텍스트 생성 과정에서 최적의 출력 시퀀스를 찾기 위해 사용
- **Beam size :** 탐색과정에서 한 번에 고려하는 후보 시퀀스의 수
- **--** 최상위 후보들만을 유지하면서 가장 가능성이 높은 출력 시퀀스를 찾아냄./

1. 탐색 시작 : 시작 토큰만 포함된 시퀀스에서 탐색
2. 시퀀스 확장 : 가능한 모든 다음 단어들의 확률을 계산
3. 최상위 후보 선택 : 계산된 확률을 바탕으로, 가장 가능성이 높은 상위 **N**개의 단어 **(=beam size)**를 선택 → 탐색범위 ↑, 계산 복잡도 ↑
4. 최종 시퀀스 선택 : 이과정을 반복하여, 최종적으로 가장 확률이 높은 시퀀스를 선택

Beam size가 크면 더 많은 후보 시퀀스를 고려할 수 있어. 탐색이 보다 포괄적으로 됨

그리디 서치(Greedy Search)나 빔 서치(Beam Search)는 문장 생성, 특히 기계 번역과 같은 시퀀스 생성 작업에서 최적의 결과를 선택하는데 도움을 주는 기법

Experimental Results

- LSTM learns much better when the source sentences are **reversed**
  solving problems with **long term dependencies**

명확한 설명은 X

가설)

- 초반 단어의 영향이 최종 **hidden state**에 미치는 영향 증가
  (short term dependencies을 도입한 것)
- 입력 문장을 뒤집어도 **source word** 와 **target word** 사이 거리의 평균을 동일하게 유지된다.

하지만, **"minimal time lag"** 가 크게 줄어들기 때문에 성능의 향상이 있을 수 있다고 설명

perplexity(혼란도) : 5.8-> 4.7
번역문의 test BLEU 점수 : 25.9-> 30.6

perplexity:
언어 모델 성능 측정 지표 중 하나로 모델이 내놓은 답의 혼란한 정도

Conclusion

- 앙상블 모델 사용

앙상블 모델에서 각각의 LSTM은 같은 입력 데이터에 대해 독립적으로 번역을 수행하고, 이후 이러한 번역들은 다양한 방법으로 결합
여러 LSTM 모델(정확히는 5개)을 조합하여 단일 번역을 생성하는 데 사용되었다는 의미

- **SMT(Statistical Machine Translation)**
  : 통계적 기법을 사용하여 한 언어에서 다른 언어로 텍스트를 번역하는 기계 번역의 한 형태

Conclusion

- 대규모 **deep LSTM**이 대규모 기계 번역 작업에 있어 무제한의 어휘록을 가진 **standard SMT(Statistical Machine Translation)** 기반 시스템 보다 더 높은 성능을 발휘함

- **source sentences**의 단어를 역순으로 배치하는 것이 더 높은 성능을 보임

- **LSTM**은 매우 긴 문장도 거의 올바르게 번역하였다.
  (but, 아주 긴 문장을 역순으로 배치하여 학습할 때는 아직 한계가 보임**)**

- **SMT(Statistical Machine Translation)**
  : 통계적 기법을 사용하여 한 언어에서 다른 언어로 텍스트를 번역하는 기계 번역의 한 형태