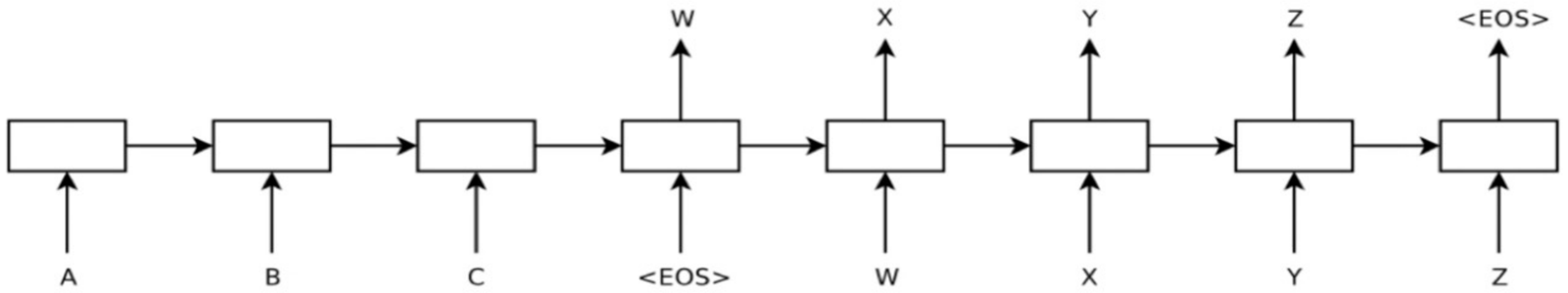
Sequence to Sequence Learning with Neural Networks

발표자 민시온

Background

Sequence to Sequence Learning with Neural Networks

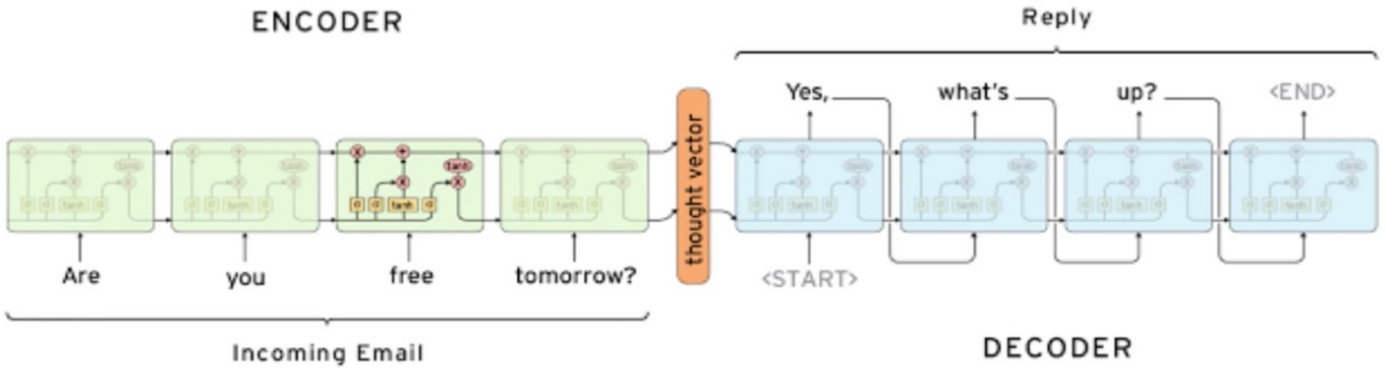
- Limitation of DNN - Sequence 처리에 한계
- Long range temporal dependency - 멀리 떨어진 단어간의 긴밀한 연결관계
- Deal with sequential problems - long sequence에 대한 문제 해결



Abstract

1. 여러층의 **LSTM(4개)**을 사용하여 **Encoder**와 **Decoder** 네트워크를 분리하여 학습 (긴 문장에도 좋은 성능)
2. 입력 문장을 받은 Encoder는 **마지막 hidden state에서 출력된 고정 크기**의 벡터를 출력(context vector)하고, Decoder는 이 벡터를 사용해 문장을 생성
   - **<EOS>** 생성 : 시퀀스의 끝(End-of-Sentence), 여기에 이르면 출력 내용을 생성하는 일을 중단함
   - **Encoder 의 출력(고정 크기 벡터)은 매우 긴 문장에서 성능 하락의 원인**이 됨
3. **WMT' 14** 데이터 셋을 이용, 영어→불어 번역에서 **BLEU** 스코어 34.8점을 달성함 (SMT : 33.3)
   - **BLEU** (bilingual evaluation understudy) : **기계 번역의 품질을 측정**하는데 사용하는 지표, 실제 사람이 한 번역과 기계 번역의 유사성을 계산(n-gram)하는 방식으로 구함
4. **학습** 과정에서 입력 **문장의 순서를 뒤집어서 훈련**하니 더 좋은 성능이 나옴

Model(1)



Encoder

- obtain large fixed dimensional vector representation

Decoder

- extract the output sequence

Model(2)

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

- Vanilla RNN : 층이 깊어질 수록 학습이 잘 되지 않는 문제(Vanishing/ Exploding gradients) 발생
- LSTM: forget gate -> 확률적으로 입력값을 통과시켜 문제 해결.
- 예측하고자 하는 것을 조건부 확률로 계산
- x1.. xT(입력), y1.. yT'(출력)
- 확률에 softmax 계산 적용

Model(3)

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

- 입출력 문장을 위한 2개의 LSTM을 동시에 사용
- 성능을 위해 여러층(4) LSTM 층을 구성
- 입력문장의 단어 순서를 뒤집어서 효과적임을 발견.

Experiments

3.1 Dataset details

- WMT’14 : 영어 -> 불어 번역 데이터셋 이용.
- 사전에 없는 단어는 “UNK” 토큰 처리함.

Experiments

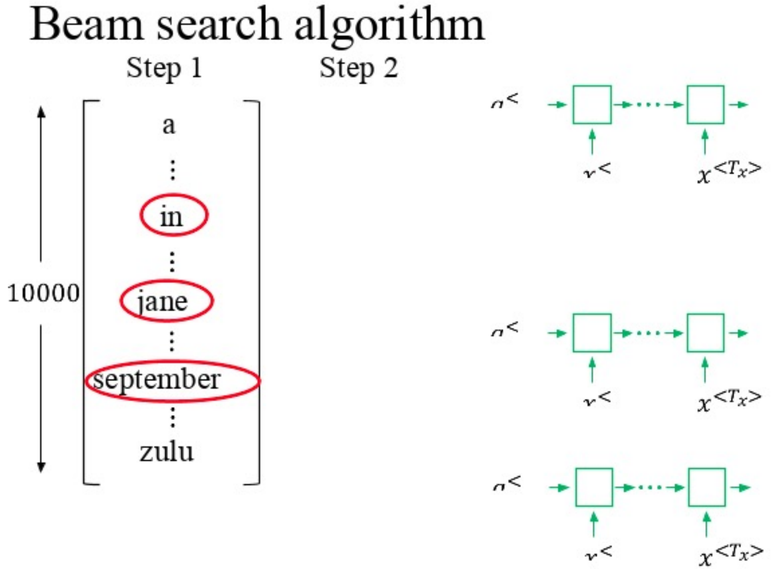
3.2 Decoding and Rescoring

$$1/|\mathcal{S}| \sum_{(T,S)\in\mathcal{S}} \log p(T|S) \qquad \hat{T} = \operatorname{argmax} p(T|S)$$

- 자연어 처리에서 흔히 사용되는 식. 훈련시와 훈련후 결과 생성에 사용된다.
- T: 번역 결과, S: 입력문장(영어), S: 훈련 데이터 셋
- argmax식을 통해 번역 결과를 생성. 번역 과정에는 beam search decoder를 사용

Experiments

3.2 Decoding and Rescoring



- Step 1에서 가장 큰 확률을 beam size 만큼 고른다. 그리고, step 2에서 나온 확률과 step 1에서 나온 가장 큰 확률 beam size 개와 각각 곱해서, 이 중 가장 높은 확률 beam size 개를 취한다.
- 이를 sequence prediction이 끝날 때까지 반복한다.
- beam size를 2로 함.

Experiments

3.3 Reversing the Source Sentences

	Method	test BLEU score (ntst14)
	Bahdanau et al. [2]	28.45
	Baseline System [29]	33.30
reversed	Single forward LSTM, beam size 12	26.17
reversed	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**

1. # of ensemble
2. size of beam search

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

- Forward < Reverse
- beam size 2 < beam size 12

Experiments

3.3 Reversing the Source Sentences

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

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

- 기존 통계기반 번역(SMT)와 함께 사용한 결과 최고 성능을 보이는 score에 근접한 모습.

Experiments

3.4 Training Details

- 4 layer LSTMs / 각 layer당 1000cells / 1000 차원의 워드임베딩
- vocabulary size: 160,000(영어) / 80,000(불어)
- 80,000단어의 softmax 분류기
- LSTM의 초기 파라미터 : -0.08~ 0.08로 초기화(uniform distribution)
- SGD without momentum 사용
- learning rate 초기에는 0.7 -> 5 epoch 이후부터 half epoch 진행 시 절반씩 줄여나감.
- batch size : 128문장 단위
- 문장의 길이가 다양하므로, minibatch 안의 비슷한 길이의 문장들을 배치단위로 구성함으로써 성능 2배 향상

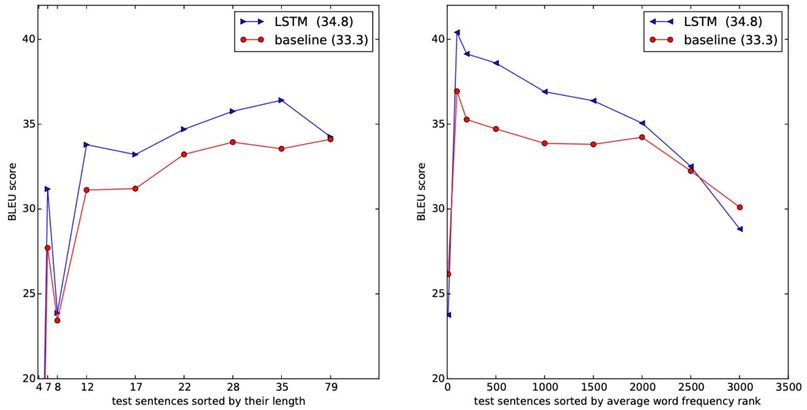
Experiments

3.5 Paralleization

- 8-GPU 머신이 병렬 작업 수행
- 각 LSTM 레이어마다 GPU 할당(4층: 4개)
- 나머지 4개의 GPU 는 softmax에서 병렬적으로 작업하는데 활용
- 훈련에 10일 소요

Experiments

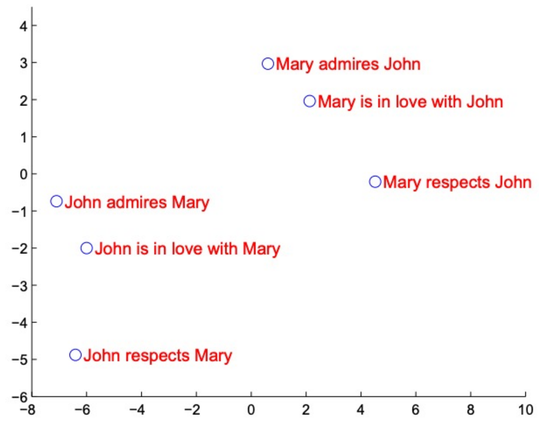
3.6 Performance on long sentences

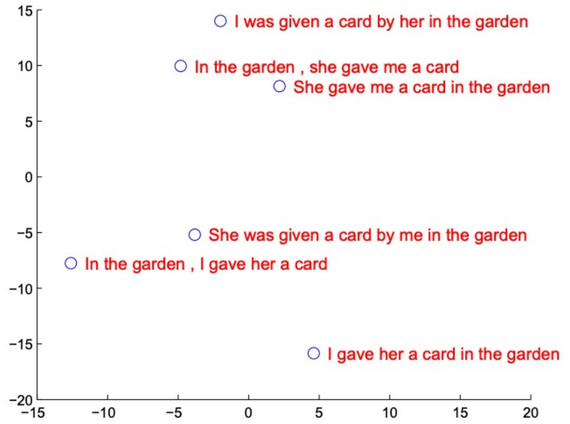


- 전반적으로 SMT baseline < LSTM 모델
- 35단어 이상의 문장을 넘기면 LSTM 모델 성능 감소

Experiments

3.7 Model Analysis





- LSTM의 hidden state를 PCA로 클러스팅한 결과
- 유사 문장 사이 거리가 더 가까움
- 주어/동사별 의미단위로 조밀함. 수/능동태 간의 영향도는 거의 없음.

Conclusion

- Seq2Seq 구조의 4-layer LSTM을 이용한 NMT를 제안함.
- Encoder/ Decoder 두 부분으로 구성됨.
- 35개 단어를 넘기는 경우 정확도가 급격히 하락.
  - 이러한 점을 보완하기 위해 Attention 개념이 등장
  - Character-level로 진행 시 35글자를 넘기는 경우가 많아 Word-level input을 사용함.