Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

Cho, 2014

발표: 염혜원

1. Introduction

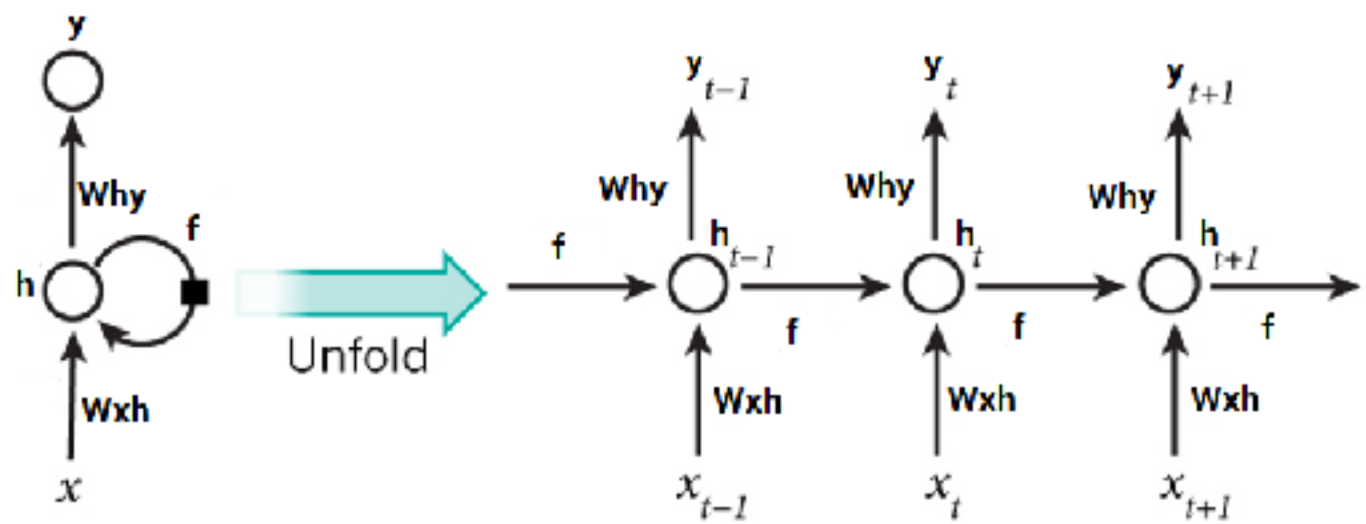
- This paper focuses on a novel neural network architecture that can be used as a **part of** the conventional phrase-based SMT system ; ***"RNN Encoder—Decoder"***
- *RNN Encoder—Decoder* consists of two recurrent networks
  - Encoder maps a variable-length source sequence to a fixed-length vector
  - Decoder maps the vector representation back to a variable-length target sequence

1. Introduction (cont'd)

- Two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence ; 조건부 확률을 최대화!
- Additional suggestion of hidden unit (GRU)
- The model is then used as a part of a standard phrase-based SMT system by scoring each phrase pair in the phrase table ; 기존 번역 시스템을 보조하는 수단으로 활용

2. RNN Encoder—Decoder

2.1 Preliminary: Recurrent Neural Networks



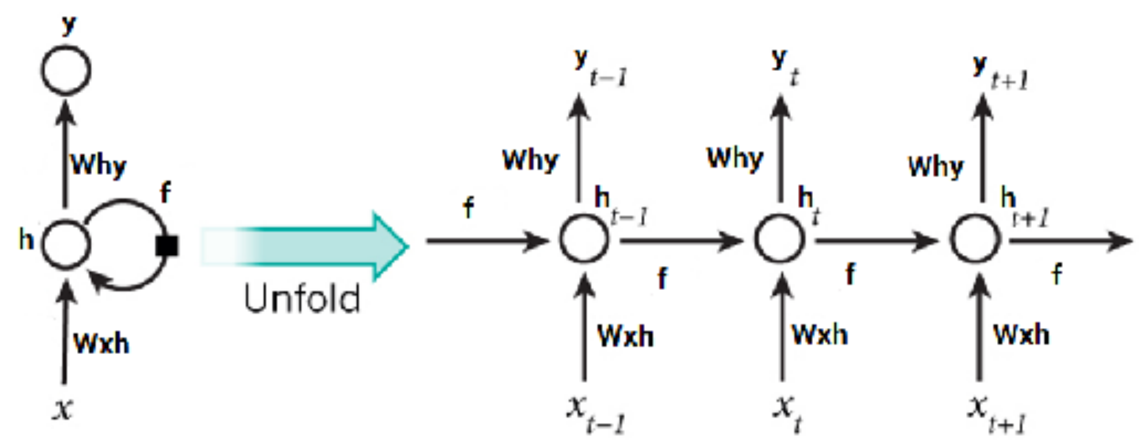
- 구성요소: hidden state **h**, optional output **y**
- Input**:** variable length sequence $x = (x_1, \ldots, x_T)$
- t 시점에 hidden state는 이전 state와 현재 input에 의해 업데이트

$$h_t = f\left(h_{t-1}, \quad x_t\right)$$

** f : non-linear activation function*

2. RNN Encoder—Decoder

2.1 Preliminary: Recurrent Neural Networks



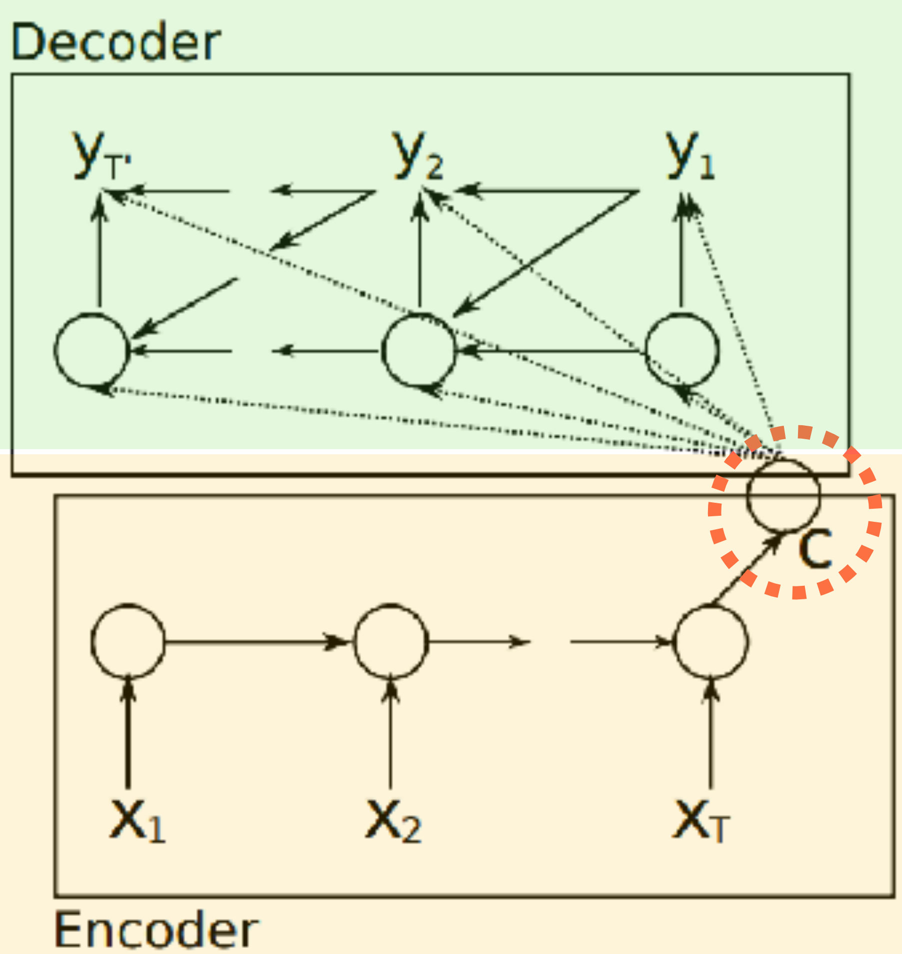
- RNN은 sequence에 대한 probability distribution을 학습할 수 있음
  (Sequence 내에서 다음에 나올 심볼을 예측하도록 학습됨)
  —> 이를 통해 각 시점 *t*에서 새로운 sequence를 생성해낼 수 있음

- Output at each time step t: conditional distribution
  - $p\left(x_t \mid x_{t-1}, \quad \ldots \quad , x_1\right)$

- Probability of the sequence x :
  - $\prod_{t=1}^{t=T} p\left(x_t \mid x_{t-1}, \quad \ldots \quad , x_1\right)$

2. RNN Encoder—Decoder

2.2 RNN Encoder —Decoder



- hidden state h_t가 주어졌을 때
  다음에 나올 심볼 y_t를 예측하도록 학습됨
- 이때 vanilla RNN과 달리,y_{t-1} 과 **c**도 고려

$$h_t = f\left(h_{t-1}, \quad y_{t-1}, \quad c\right)$$

- **summary c** (last hidden state of the encoder) of the whole input sequence

- 순차적으로 sequence 를 읽음
- hidden state update

- *Encoder—Decoder are jointly trained to maximize the conditional log-likelihood*

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y_n \mid x_n)$$

▸ Minimize Cross Entropy Error for all target words conditioned on source words

2. RNN Encoder—Decoder

2.2 RNN Encoder —Decoder

- RNN Encoder—Decoder 가 학습되면 모델은 두 가지로 활용될 수 있음
  - Input sequence 가 주어졌을 때 target sentence 생성
  - (input, output) pair가 주어졌을 때 조건부 확률을 기반으로 점수를 매김

2. RNN Encoder—Decoder

2.3 Hidden Unit that Adaptively Remembers and Forgets

- New type of hidden unit: GRU

$$r_j = \sigma\left(\left[W_r x\right]_j + \left[U_r h_{t-1}\right]_j\right)$$

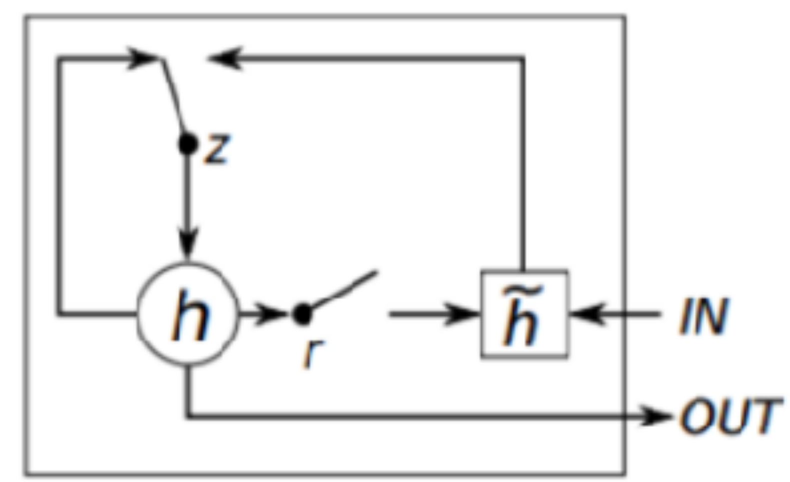
reset gate
: 이전 hidden 무시할 것인지

$$z_j = \sigma\left(\left[W_z x\right]_j + \left[U_z h_{t-1}\right]_j\right)$$

update gate
: 새로운 hidden state $\widetilde{h}_j^t$ 로 업데이트 할 것인지

$$\widetilde{h}_j^t = \phi\left([Wx]_j + \left[U(r \odot h_{t-1}\right]_j\right)$$

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\widetilde{h}_j^t$$



σ : logistic sigmoid $[]_j$: j-th element of a vector

2. RNN Encoder—Decoder

2.3 Hidden Unit that Adaptively Remembers and Forgets

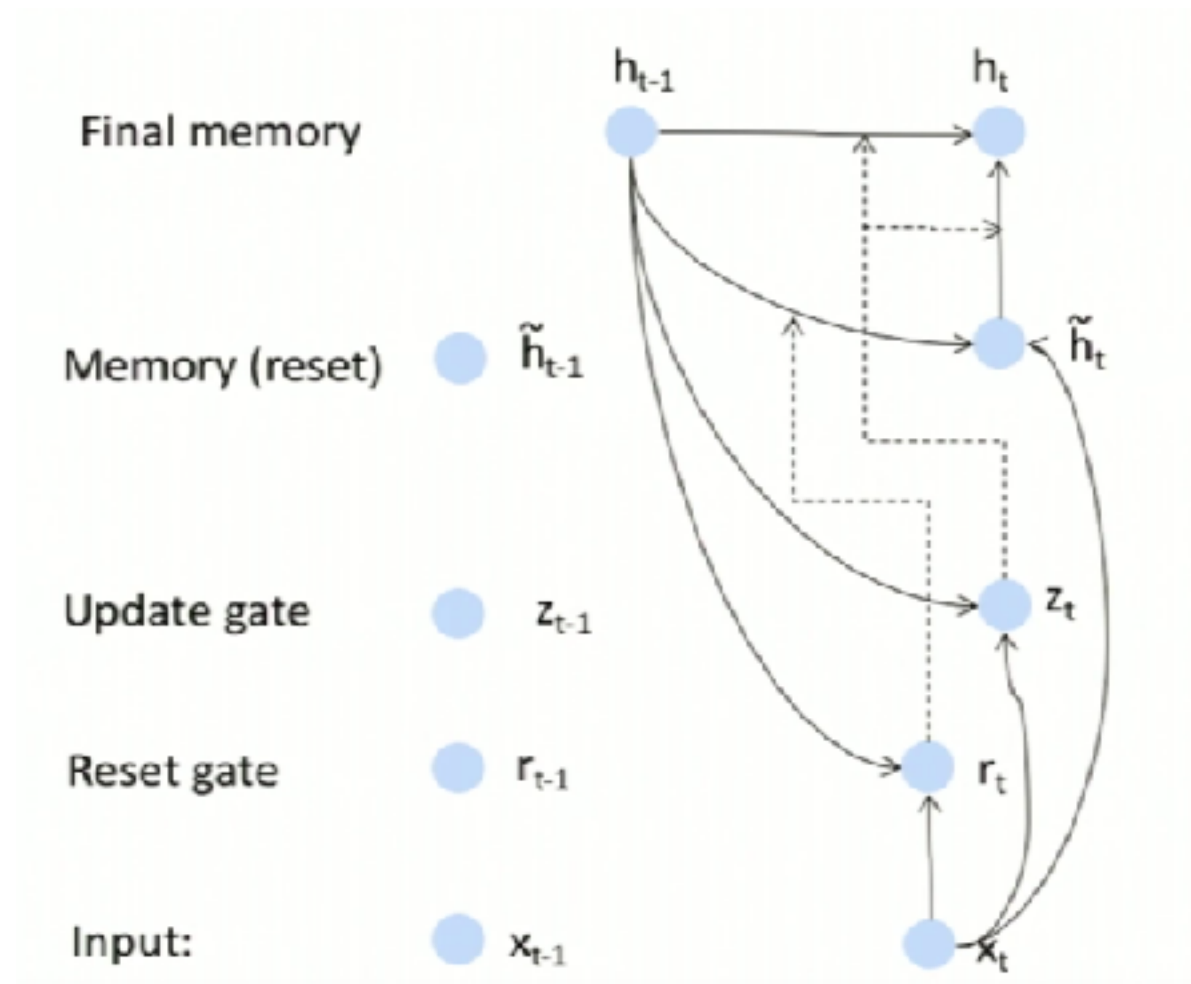
- New type of hidden unit: GRU

$$r_j = \sigma\left(\left[W_r x\right]_j + \left[U_r h_{t-1}\right]_j\right)$$

$$z_j = \sigma\left(\left[W_z x\right]_j + \left[U_z h_{t-1}\right]_j\right)$$

$$\tilde{h}_j^t = \phi\left([Wx]_j + \left[U(r \odot h_{t-1}\right]_j\right)$$

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\tilde{h}_j^t$$



* source : CS 224 lecture #9

3. Statistical Machine Translation

- SMT의 목적은 source sentence **e**가 주어졌을 때 ***p*(f|e)** 를 maximize 하는 translation f를 찾는 것임

- 실제 SMT는 다음의 log-linear model를 활용함

$$\log p\left(f \mid e\right) = \sum_{n=1}^{N} w_n f_n(f, e) + \log Z(e)$$

w_n, f_n : n-th feature and weight

$\log Z(e)$: normalization constant

- weight는 BLEU score를 maximize 하도록 최적화 됨

* BLEU(bilingual evaluation understudy) :기계 번역의 품질을 측정하는데 사용하는 지표.
실제 사람이 한 번 역과 기계 번역의 유사성을 계산하는 방식으로 구함. 간단하고 쉽게 구할 수 있다는 장점이 있음

3. Statistical Machine Translation

3.1 Scoring Phrase Pairs with RNN Encoder—Decoder

- 본 논문에서는 RNN Encoder—Decoder를 phrase pair에 대해 학습시키고, 결과 score를 additional feature로 활용

$$\log p\left(f \mid e\right) = \sum_{n=1}^{N} w_n f_n(f, e) + \log Z(e)$$

w_n, f_n : n-th feature and weight

$\log Z(e)$: normalization constant

4. Experiments

4.1 Data and Baseline System

- Translation에 있어 가능한 모든 data를 concat하는 것은 optimal 하지 않을 뿐만 아니라 모델을 핸들링하기도 어려움
- 각 task 에 가장 적합한 subset에 focus 해야 함
  - subset of 418M words out of more than 2G words for language modeling
  - subset of 348M out of 850M words for training the RNN Encoder—Decoder
  - for training the neural networks, limit source/target vocal to the most frequent 15,000 words for both English/French
  - out-of-vocabulary words were mapped to [UNK]

4. Experiments

4.1.1 RNN Encoder—Decoder

- 1000 hidden units with the proposed gates at the encoder and at the decoder
- activation function used for $\tilde{h}$: tanh
- from the hidden state in the decoder to output : implemented as a deep neural network w/ single intermediate layer having 500 maxout units each pooling 2 inputs (??)
- Adadelta / sgd to train RNN Encoder—Decoder
- 64 phrase pairs used per each update
- most frequent 15,000 words (both English and French)

4. Experiments
(Appendix) RNN Encoder

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\widetilde{h}_j^t$$

* initial hidden state is fixed to 0

$$\widetilde{h}_j^t = tanh\left(\left[We(x_t)\right]_j + \left[U(r \odot h_{t-1})\right]_j\right)$$

$$z_j = \sigma\left(\left[W_z e(x_t)\right]_j + \left[U_z h_{t-1}\right]_j\right)$$

$$r_j = \sigma\left(\left[W_r e(x_t)\right]_j + \left[U_r h_{t-1}\right]_j\right)$$

$$c = tanh\left(Vh^N\right)$$

* c: representations of the source phrase

4. Experiments

(Appendix) RNN Decoder

$$h'^{0} = tanh(V'c)$$

* initialization

$$h'^{t}_{j} = z'_{j}h'^{t-1}_{j} + (1 - z'_{j})\widetilde{h'}^{t}_{j}$$

$$\widetilde{h'}^{t}_{j} = tanh\left(\left[W'e(y_{t-1})\right]_{j} + r_{j}\left[U'h'_{t-1} + Cc\right]\right)$$

$$z'_{j} = \sigma\left(\left[W'_{z}e(y_{t-1})\right]_{j} + \left[U'_{z}h'_{t-1}\right]_{j} + [C_{z}c]_{j}\right)$$

$$r'_{j} = \sigma\left(\left[W'_{r}e(y_{t-1})\right]_{j} + \left[U'_{r}h'_{t-1}\right]_{j} + [C_{r}c]_{j}\right)$$

$$r'_{j} = \sigma\left(\left[W'_{r}e(y_{t-1})\right]_{j} + \left[U'_{r}h'_{t-1}\right]_{j} + [C_{r}c]_{j}\right)$$

- t 시점마다 decoder는 j-th word에 대한 확률을 계산

$$p(y_{t,j} = 1 \mid y_{t-1}, \ldots, y_{1}, X) = \frac{exp(g_{j}s_{t})}{\sum_{j'=1}^{K} exp(g_{j'}s_{t})}$$

* i-elemet of s^{t} :

$$s_{i}^{t} = max\left(s'^{t}_{2i-1}, s'_{2i}\right)$$

—> *maxout* unit

$$s'^{t} = O_{h}h'^{t} + O_{h}y_{t-1} + O_{h}c$$

4. Experiments

4.1.2 Neural Language Model

- CSLM (Continuous Space Language Model)
: traditional approach of using a neural network for learning a target language model (Schwenk, 2007)
    - RNN과 CSLM 을 같이 사용했을 때와, RNN만 사용했을 때를 비교해서 RNN과 CSLM의 contribution이 구별됨을 확인하고자 함

4. Experiments
4.2 Quantitative Analysis

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

* Word Penalty (??)

4. Experiments

4.3 Qualitative Analysis

- 기존의 translation model은 통계량에 기반하므로 자주 나오는 phrase를 덜 나오는 phrase대비 잘 예측 할 것
- RNN Encoder—Decoder는 frequency 정보 없이 훈련 되었으므로 통계 보다는 언어학적 규칙성에 의해 scoring할 것으로 기대
  - 대부분의 케이스에서 RNN Encoder—Decoder의 선택이 실제 번역과 유사했으며, RNN Encoder—Decoder는 짧은 구문을 선호하는 현상 관찰됨
- RNN Encoder—Decoder 만으로 generation 결과 phrase table을 참조하지 않고서도 well-formed target phrases 생성함

4. Experiments

4.4 Word and Phrase Representations

- RNN Encoder—Decoder 는 phrase 의 semantic & syntactic structure 모두 캡처

5. Conclusion

- RNN Encoder—Decoder 모델 제시:
  - arbitrary length sequence to another sequence from a different set, of arbitrary length
- 새로운 hidden unit 제시:
  - includes a reset / update gate
- 새로운 모델은 언어학적 규칙성을 찾아낼 수 있고, 말이 되는 target phrases를 제안할 수도 있음
- RNN Encoder—Decoder가 scoring에 활용되었을 때 BLEU score 향상

End of Document