Neural machine translation by jointly learning to align and translate

Bahdanau, Dzmitry and Cho, Kyunghyun and Bengio, Yoshua

경상국립대학교 석사과정 오서영

Abstract

Neural machine translation (NMT)는 Encoder-Decoder 모델 종류에 속하며
- Encoder : 문장 → 고정된 길이의 벡터
- Decoder : 고정된 길이의 벡터 → 번역된 문장

고정된 길이의 벡터 (Fixed-length vector)
- 기존 Encoder-Decoder 아키텍처의 성능을 저하시키는 (bottleneck) 병목 현상
- 모델이 이들 부분들을 명시적으로 hard segment로 형성할 필요 없이, 목표 단어를 예측하는데 관련된 소스 문장의 부분들을 자동으로 (soft search) 검색할 수 있게 함

Introduction

기본 Encoder-Decoder의 성능은 입력 문장의 길이가 길어질수록 급격히 저하됨
→ Encoder-Decoder model의 확장 : "align and translate jointly"

모델이 변환에서 단어를 생성할 때마다, 가장 관련성이 있는 정보가 집중되어 있는 소스 문장에서 position sets를 (soft search)검색함
→ 이러한 source position 와 이전에 생성된 모든 대상 단어와 관련된 context 벡터를 기반으로 대상 단어를 예측함

입력 문장을 일련의 벡터로 Encoding, 변환을 Decoding 하면서 이들 벡터의 하위 집합을 선택함.
→ NMT는 길이에 관계없이 원본 문장의 모든 정보를 고정 길이 벡터로 압축할 필요가 없게 함

RNN Encoder-Decoder

Encoder는 입력 문장 $x = (x_1, \ldots, x_{T_x})$를 c 로 읽어냄

RNN :

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

$$c = q\left(\{h_1, \cdots, h_{T_x}\}\right),$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t,

and c is a vector generated from the sequence of the hidden states.

예를 들어, f = LSTM, $q(\{h_1, \ldots, h_{Tx}\}) = H^T$. (Sutskever et al. (2014))

RNN Encoder-Decoder

Decoder는 context 벡터 c 및 이전에 예측된 모든 단어 $\{y_1, \ldots, y_T\}$가 주어지면 다음 단어 y_t, 를 예측하도록 훈련됨

→ Decoder 는 결합 확률을 순서화된 조건부로 분해함으로써 변환 y에 대한 확률을 정의함

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c), \qquad \text{where y} = \{y_1, \ldots, y_{T_y}\}$$

RNN을 사용한 조건부 확률 모델링:

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

g is nonlinear function that outputs the probability y_t,
s_t is the hidden state of the RNN

Learning to align and translate (New model architecture)

Decoder : General description

- 각 조건부 확률:

$$p(y_i|y_1,\ldots,y_{i-1},\mathbf{x}) = g(y_{i-1},s_i,c_i),$$

where $s_i = f(s_{i-1},y_{i-1},c_i)$, s_i는 시간 i에 대한 RNN hidden state

→ 확률은 각 대상 단어 y_i에 대해 별개의 context 벡터 c_i에 의해 조건화됨.

Context 벡터 c_i는 Encoder가 입력 문장을 매핑하는 **annotations** $(h_1,\ldots,h_{T_x})$ sequence에 의존

→ 각각의 **annotation** h_1는 입력 시퀀스의 i 번째 단어를 둘러싼 부분에 중점을 둔 전체 입력 시퀀스에 대한 정보를 포함

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

weighted sum of these annotations h_i

Learning to align and translate (New model architecture)

The weight α_{ij} of each annotation h_j :

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad \text{where} \quad e_{ij} = a(s_{i-1}, h_j)$$

위치 j 주변의 입력과 위치 i의 출력이 얼마나 일치하는지
점수를 매기는 **alignment model**
→ 점수는 입력 문장의 RNN hidden state s_{i-1}와 j번째 주석 h_j를 기반으로 함

새 모델의 다른 구성요소들과 함께 훈련된 feedforward NN 으로
alignment model을 매개변수화함
→ alignment model 은 soft alignment을 직접 계산하므로 비용 함수의 gradient가
→ 역전파 될 수 있음
→ 이 gradient 를 사용하여 alignment model 뿐만 아니라 전체 translation model을 공동으로 학습할 수 있음

Learning to align and translate (New model architecture)

Encoder : Bidirectional RNN for annotating sequences

- 각 단어의 annotation이 앞의 단어 뿐만 아니라 다음 단어를 요약하게 하려고함

→ Bidirectional RNN 활용 (Schuster and Paliwal, 1997).

→ Forward hidden state를 연결하여 각 단어 x_j에 대한 annotation을 얻음:

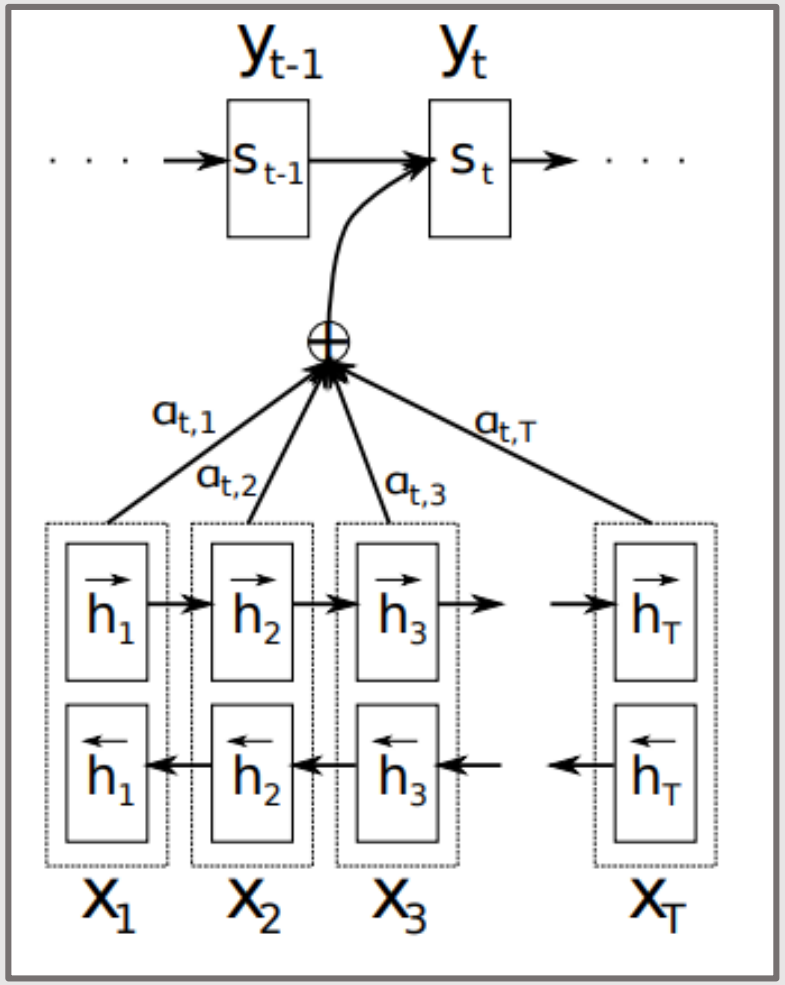
$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top$$

Annotation h_j에는 선행단어와 다음단어의 요약이 포함됨

→ RNN은 최근 입력을 더 잘 나타내는 경향이 있으므로, annotation h_j 는 x_j 주위의 단어에 집중될 것

→ 이러한 annotation sequence는 나중에 context 벡터 계산을 위해 (Decoder, alignment model) 사용됨

Learning to align and translate (New model architecture)



Proposed model (RNNsearch)

Experiments

Dataset

- WMT’ 14 : 영어-프랑스어 corpora
- Tokenization 후, 모델을 훈련시키기 위해 각 언어에서 가장 빈번한 30,000 단어의 후보 목록을 사용

Models

- RNN Encoder -Decoder (Rnnencdec, Cho et al.), RNNSearch (제안 된 모델)
- 최대 30 개의 단어의 문장으로 훈련: (RNNencdec-30, RNNsearch-30)
- 최대 50 개의 단어의 문장으로 훈련: (RNNencdec-50, RNNsearch-50)

- SGD with Adadelta : 각 SGD 업데이트 방향은 80 문장의 미니 배치를 사용하여 계산함

→ 모델이 학습되면 beam search를 사용하여 조건부 확률을 대략적으로 최대화하는 translation 을 찾음

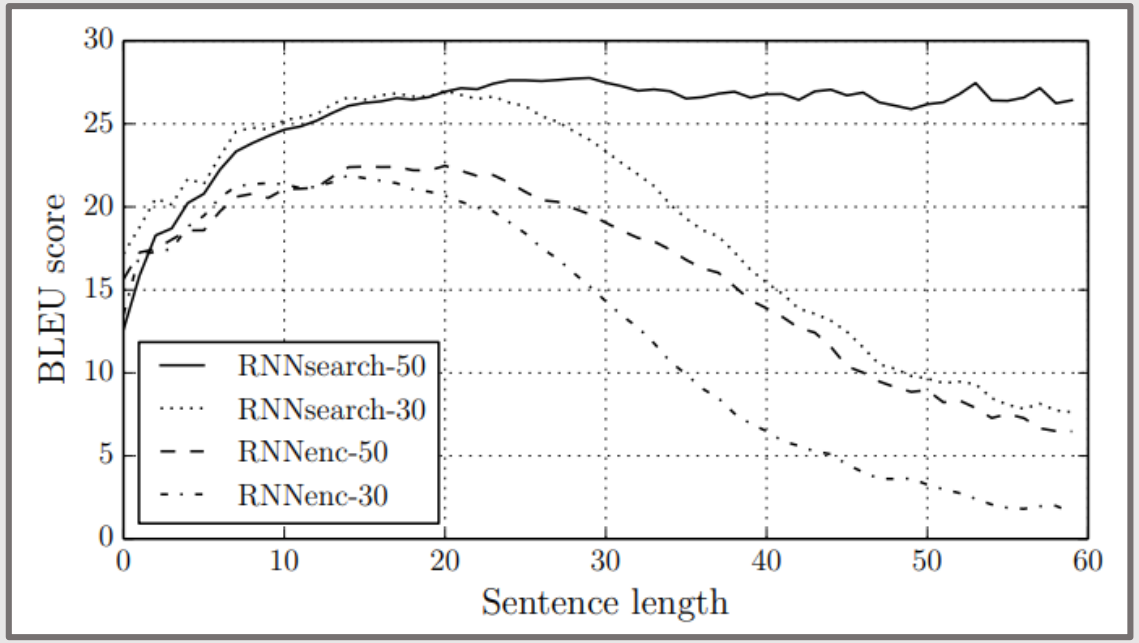
Experiments

Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

BLEU score table

RNNSearch의 성능은 기존의 문구 기반 번역 시스템 (MOSE) 의 성능만큼 높음

Experiments



- 문장의 길이가 증가함에 따라 RNNencdec의 성능이 크게 떨어짐
But, RNNsearch-30과 RNNsearch-50 모두 문장 길이에 더 robust 함.
→ 특히 RNNsearch-50 은 길이가 50 이상인 문장에서도 성능 저하 X

Conclusion

1. 각 대상 단어를 생성할 때 모델이 입력 단어 또는 encode로 계산된 annotation을 (soft search) 검색 하여 기본 Encoder-decoder를 확장함

2. 이로 인해 모델이 전체 소스 문장을 고정된 길이의 벡터로 encoding 할 필요 X
→ 모델이 다음 단어 생성과 관련된 정보에만 초점을 맞추도록 함

3. Alignment mechanism을 포함하여 번역 시스템의 모든 부분이 올바른 번역을 생성하게 하는 로그 확률을 향해 공동으로 훈련됨