2022학년도 1학기 컴퓨터언어학

제19강 기계번역과 부호화기-복호화기 모형 (2)

박수지

서울대학교 인문대학 언어학과

2022년 5월 16일 월요일

오늘의 목표

1. 순환 신경망으로 구성된 부호화기-복호화기 모형에서 사용하는 문맥 벡터의 한계와 대안을 설명할 수 있다.

부호화기-복호화기 모형

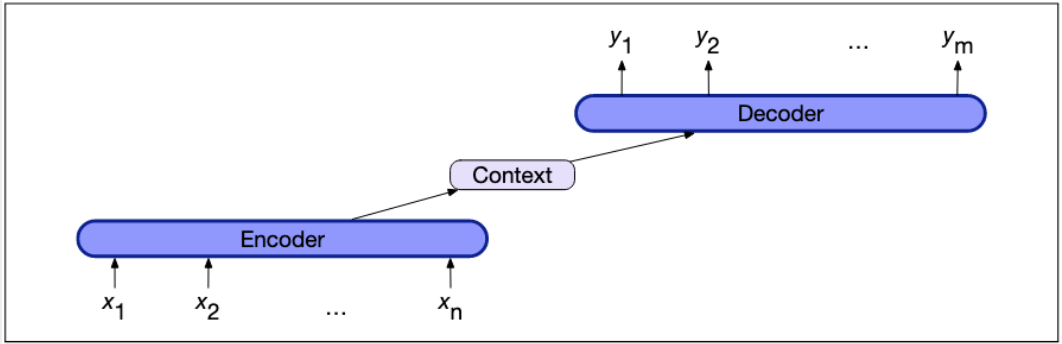


Figure 10.3 The encoder-decoder architecture. The context is a function of the hidden representations of the input, and may be used by the decoder in a variety of ways.

부호화기-복호화기 모형

순환 신경망을 사용하는 경우

부호화기(Encoder)

출발어 문장의 단어 $x_1, x_2, \cdots, x_n$

은닉 상태 $\vec{h}^e_1, \vec{h}^e_2, \cdots, \vec{h}^e_n$

복호화기(Decoder)

문맥 벡터 $\vec{c} = \vec{h}^d_0 = \vec{h}^e_n$

은닉 상태 $\vec{h}^d_i = g\left(\hat{y}_{i-1}, \vec{h}^d_{i-1}, \vec{c}\right)$

도착어 문장 생성 $\vec{h}^d_i \mapsto \hat{y}_i$

부호화기-복호화기 모형

순환 신경망을 사용하는 경우

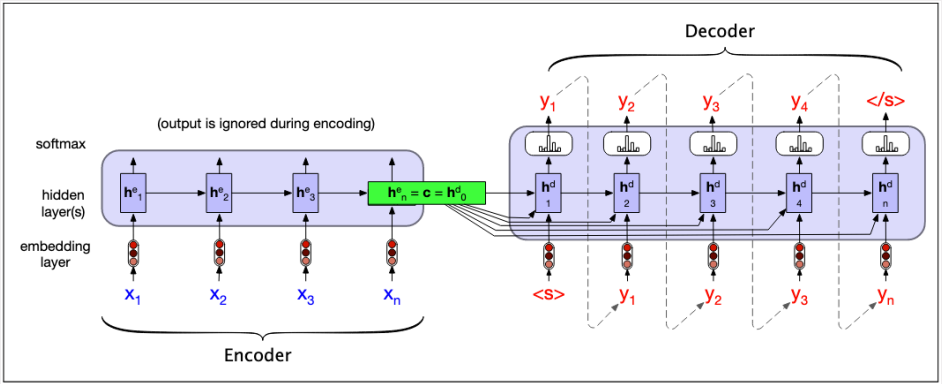


Figure 10.5 A more formal version of translating a sentence at inference time in the basic RNN-based encoder-decoder architecture. The final hidden state of the encoder RNN, h^e_n, serves as the context for the decoder in its role as h^d_0 in the decoder RNN.

RNN Encoder-decoder 모형의 한계점

정보의 병목 현상

- 출발어 문장 x에 대한 모든 정보를 하나의 벡터 $\vec{c}$로 포착해야 한다.

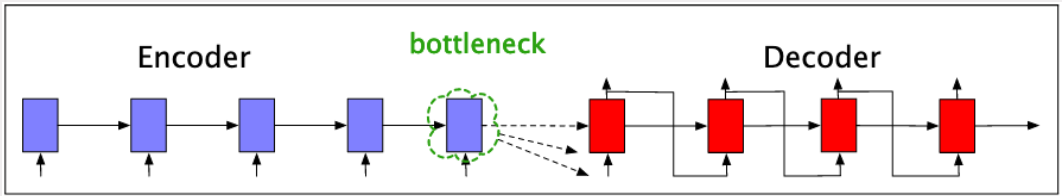


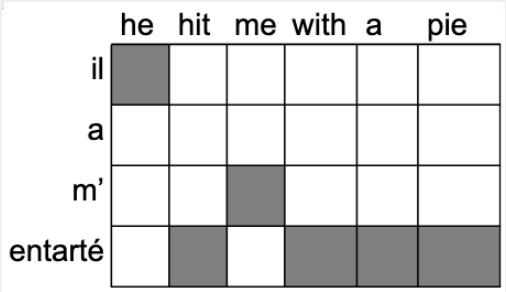
Figure 10.8 Requiring the context *c* to be only the encoder's final hidden state forces all the information from the entire source sentence to pass through this representational bottleneck.

문제

한 문장 **전체**를 표현하는 문맥 벡터 $\vec{c}$가 …
…도착어 문장 y의 각 단어에 똑같은
정도로 반영되어도 괜찮을까?

해법

도착어 문장 y를 생성하는 각 단계 i에서
출발어 x의 각기 다른 부분에 주목한다.



예시: http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture09-nmt.pdf

부호화기-복호화기 모형

Vanilla encoder-decoder

문맥 벡터 $\vec{c} = \vec{h}_0^d = \vec{h}_n^e$

은닉 상태 $\vec{h}_i^d = g\left(\hat{y}_{i-1}, \vec{h}_{i-1}^d, \vec{c}\right)$

모든 i에서 **동일한** 문맥 벡터 $\vec{c}$를 사용한다.

Encoder-decoder with attention

문맥 벡터 $\vec{c}_i = f\left(\vec{h}_1^e, \vec{h}_2^e, \cdots, \vec{h}_n^e\right)$

은닉 상태 $\vec{h}_i^d = g\left(\hat{y}_{i-1}, \vec{h}_{i-1}^d, \vec{c}_i\right)$

i마다 각기 **다른** 문맥 벡터 $\vec{c}_i$를 사용한다.

問 어떻게 다른가?

答 출발어 문장의 단어 $x_1, x_2, \cdots, x_n$ 중 어느 것에 얼마나 주목할지가 i에 따라 달라진다.

부호화기-복호화기 모형

Attention

Encoder-decoder with attention

문맥 벡터 $\vec{c}_i = f\left(\vec{h}^e_1, \vec{h}^e_2, \cdots, \vec{h}^e_n\right)$

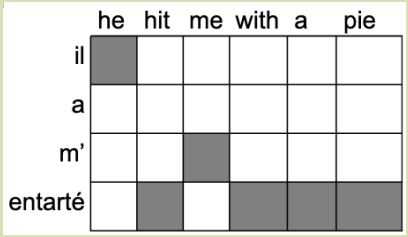
은닉 상태 $\vec{h}^d_i = g\left(\hat{y}_{i-1}, \vec{h}^d_{i-1}, \vec{c}_i\right)$

i마다 각기 **다른** 문맥 벡터 $\vec{c}_i$를 사용한다.

問 어떻게 다른가?

答 출발어 문장의 단어 $x_1, x_2, \cdots, x_n$ 중 어느 것에 얼마나 주목할지가 i에 따라 달라진다.

예시



1. $i = 1$일 때는 il에 주목하고…
2. $i = 2$일 때는 entarté에 주목하고…

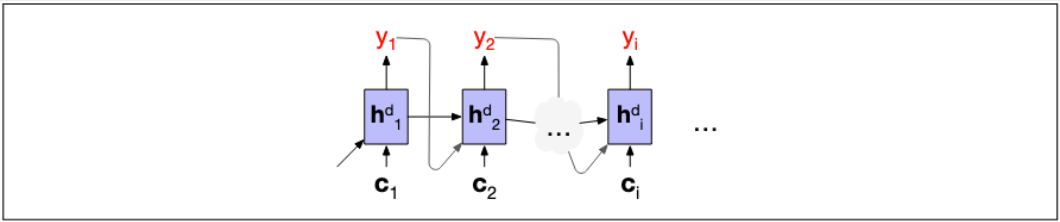


Figure 10.9 The attention mechanism allows each hidden state of the decoder to see a different, dynamic, context, which is a function of all the encoder hidden states.

아이디어

i번째 출력을 위한 복호화기 은닉 상태 $\vec{h}_i^d$를 만들 때

- 부호화기 은닉 상태 $\vec{h}_j^e$를 모두 사용하되
- 각 $\vec{h}_j^e$를 반영하는 정도를 i마다 다르게 한다. $\cdots \cdots \vec{c}_i = f\left(\vec{h}_1^e, \vec{h}_2^e, \cdots, \vec{h}_n^e\right)$

Attention scores

현재 단어를 생성하기 위한 복호화기 은닉 상태 $\vec{h}_i^d$를 만들기 위해…
…직전의 은닉 상태 $\vec{h}_{i-1}^d$에서 부호화기의 각 은닉 상태 $\vec{h}_j^e$를 얼마나 주목할 것인가?

$$a_{ij} = \text{score}\left(\vec{h}_{i-1}^d, \vec{h}_j^e\right) = \vec{h}_{i-1}^d \cdot \vec{h}_j^e \in \mathbb{R} \quad (i\text{는 고정된 시점}, j\text{는 모든 } 1 \leq j \leq n)$$

Attention weights

$\vec{h}_j^e$들의 가중평균을 취하려면 가중치들의 합이 1이 되어야 한다.

$$\alpha_{ij} = \text{softmax}\left(a_{ij}\right) = \frac{\exp\left(a_{ij}\right)}{\exp\left(a_{i1}\right) + \exp\left(a_{i2}\right) + \cdots + \exp\left(a_{in}\right)}$$

Attention score $a_{ij} = \text{score}\left(\vec{h}^{d}_{i-1}, \vec{h}^{e}_{j}\right) = \vec{h}^{d}_{i-1} \cdot \vec{h}^{e}_{j} \in \mathbb{R}$

Attention weight $\alpha_{ij} = \text{softmax}\left(a_{ij}\right) = \dfrac{\exp\left(a_{ij}\right)}{\exp\left(a_{i1}\right) + \exp\left(a_{i2}\right) + \cdots + \exp\left(a_{in}\right)}$

문맥 벡터

$$\vec{c}_i = \sum_{j=1}^{n} \alpha_{ij} \vec{h}^{e}_{j} = \alpha_{i1} \vec{h}^{e}_{1} + \alpha_{i2} \vec{h}^{e}_{2} + \cdots + \alpha_{in} \vec{h}^{e}_{n}$$

이렇게 만든 문맥 벡터를 $\vec{h}^{d}_{i} = g\left(\hat{y}_{i-1}, \vec{h}^{d}_{i-1}, \vec{c}_i\right)$ 에 활용한다.

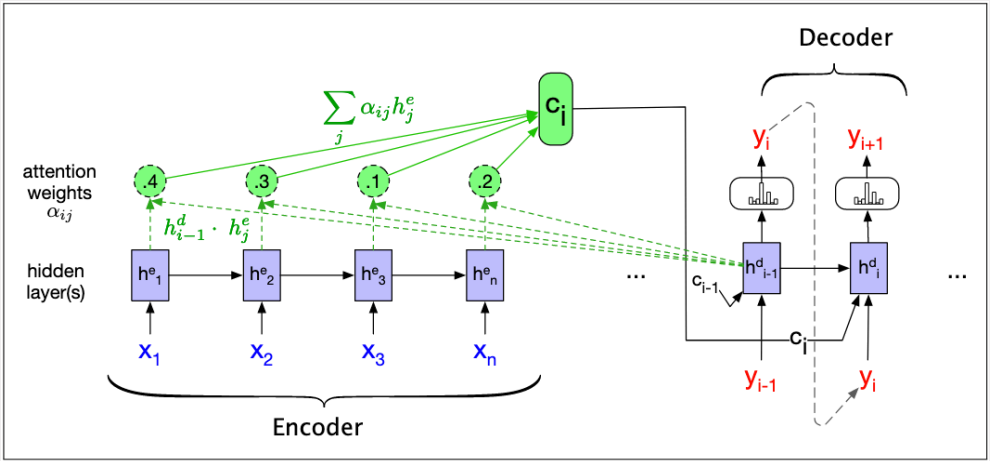


Figure 10.10 A sketch of the encoder-decoder network with attention, focusing on the computation of $\mathbf{c}_i$. The context value $\mathbf{c}_i$ is one of the inputs to the computation of $\mathbf{h}_i^d$. It is computed by taking the weighted sum of all the encoder hidden states, each weighted by their dot product with the prior decoder hidden state $\mathbf{h}_{i-1}^d$.

Attention scores revisited

두 벡터의 내적(점곱, dot product)

$$\text{score}\left(\vec{h}^{d}_{i-1}, \vec{h}^{e}_{j}\right) = \vec{h}^{d}_{i-1} \cdot \vec{h}^{e}_{j} \in \mathbb{R}$$

- 두 벡터의 차원이 같아야 한다.
- 동일한 위치의 성분 사이의 관계만 반영한다.

"More powerful" attention scores

두 벡터의 쌍선형사상(bilinear map)

$$\text{score}\left(\vec{h}^{d}_{i-1}, \vec{h}^{e}_{j}\right) = \vec{h}^{d}_{i-1} \mathbf{W}_{s} \vec{h}^{e}_{j} \in \mathbb{R}$$

- 두 벡터의 차원이 달라도 된다.
- 모든 가능한 성분의 관계를 반영할 수 있다.

$\Rightarrow$ 서로 다른 부호화기와 복호화기를 조합할 수 있다!

예시: 쌍선형사상

$$\vec{y}\mathbf{W}\vec{x} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} w_{11}x_1 + w_{12}x_2 \\ w_{21}x_1 + w_{22}x_2 \\ w_{31}x_1 + w_{32}x_2 \end{bmatrix}$$

$$= y_1 \left(w_{11}x_1 + w_{12}x_2\right) + y_2 \left(w_{21}x_1 + w_{22}x_2\right) + y_3 \left(w_{31}x_1 + w_{32}x_2\right)$$

$$= y_1 w_{11} x_1 + y_1 w_{12} x_2 + y_2 w_{21} x_1 + y_2 w_{22} x_2 + y_3 w_{31} x_1 + y_3 w_{32} x_2$$

오늘 배운 것

Encoder-decoder model with attention

- 왜 필요한가?
- 어떻게 계산하는가?

남은 것

Self-attention → Transformers

- SLP3 9.7 Self-Attention Networks: Transformers
- SLP3 11 Transfer Learning with Pre-trained Language Models and Contextual Embeddings

  BERT Bidirectional Encoder Representations from Transformer

  GPT Generative Pre-trained Transformer