Self Attention & RNN

https://www.edwith.org/deepnlp/lecture/29209/

Self Attention

-RN은 단어들간의 Relation을 본다. CNN은 작은 범위에 있는 관계를 본다.
-CNN 방식을 가중치가 부여된 RN이라고 볼 수 있다. 작은 범위에 있는 단어들은 1, 그렇지 않은 단어들은 0
-그렇다면 가중치가 0과 1이 아닌 그 사이의 값으로도 표현을 한다면?

How to represent a sentence – Self-Attention

- Can we compute those weights instead of fixing them to 0 or 1?
- That is, compute the weight of each pair $(x_t, x_{t'})$

$$h_t = \sum_{t'=1}^{T} \alpha(x_t, x_{t'}) f(x_t, x_{t'})$$

- The weighting function could be yet another neural network
  - Just another subgraph in a DAG: easy to use!
    $\alpha(x_t, x_{t'}) = \sigma(\mathrm{RN}(x_t, x_{t'})) \in [0, 1]$
  - Perhaps we want to normalize them so that the weights sum to one

$$\alpha(x_t, x_{t'}) = \frac{\exp(\beta(x_t, x_{t'}))}{\sum_{t''=1}^{T} \exp(\beta(x_t, x_{t''}))}, \text{where } \beta(x_t, x_{t'}) = \mathrm{RN}(x_t, x_{t'}))$$

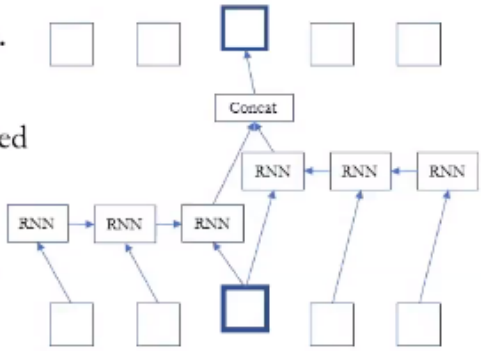
A는 weighting function, 토큰들의 관계에 따라 가중치는 달라진다.
장점 : Long range & short range denpendency 극복할 수 있다. 관계가 높은 토큰은 강조할 수 있다
단점 : 계산 복잡도가 높고 counting 같은 특정 연산이 쉽지 않다. (weight, attention 계산을 같이 해야해서)

Recurrent Neural Network(RNN)

-Self-Attetion의 경우 매번 interation을 보는 반면에 문장 정보를 하나의 백터로 압축을 한다.
-메모리를 가지고 있어서 현재까지 읽은 정보를 저장 할 수 있다.

How to represent a sentence – RNN

- Recurrent neural network: online compression of a sequence $O(T)$
  $h_t = \text{RNN}(h_{t-1}, x_t)$, where $h_0 = 0$.
- Bidirectional RNN to account for both sides.
- Inherently sequential processing
  - Less desirable for modern, parallelized, distributed computing infrastructure.
- LSTM [Hochreiter&Schmidhuber, 1999] and GRU [Cho et al., 2014] have become de facto standard
  - All standard frameworks implement them.
  - Efficient GPU kernels are available.



단점: 문장이 많이 길어질 수록 고정된 메모리에 압축된 정보를 담아야 하기 때문에, 앞에서 학습한 정보를 잊습니다. 이는 곧 정보의 손실을 뜻합니다.
토큰을 순차적으로 하나씩 읽어야 하기 때문에, 훈련 할 때 속도가 기타 네트워크 보다 느립니다.

Long Term Dependency 해결방법: bidirectional network 를 쓴다. LSTM, GRU 등 RNN 의 변형을 사용한다.