cs224n 스터디 1팀

2024.05.07

발표자 : 유금헌, 조다영

스터디원 소개 및 만남 인증

스터디원 1 : 송경준

스터디원 2 : 유금헌

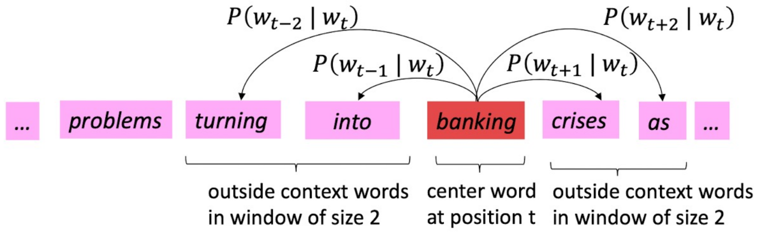
스터디원 3 : 조다영

Word Embedding

: 단어를 벡터화하는 방식

Word2vec

- 단어 간의 유사도를 반영
- 텍스트 전체의 정보를 반영하지 못한다는 단점

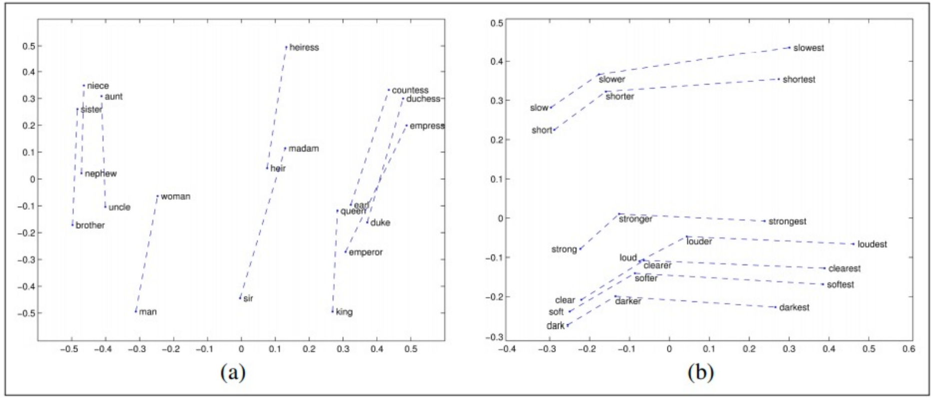


$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\le j\le m\\ j\neq 0}} \log P\left(w_{t+j} \mid w_t;\theta\right)$$

Word Embedding

GloVe

$$w_i \cdot w_j = \log P(i|j)$$



Word Embedding

Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	**88.7**	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	**93.2**	88.3	**82.9**	**82.2**

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Model	Dim.	Size	Sem.	Syn.	Tot.
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7

Dependency parsing


San Jose cops kill man with knife

Close

Text

Paper

Translate

Listen

San Jose cops kill man with knife

경찰이 칼을 든 남자를 죽였다

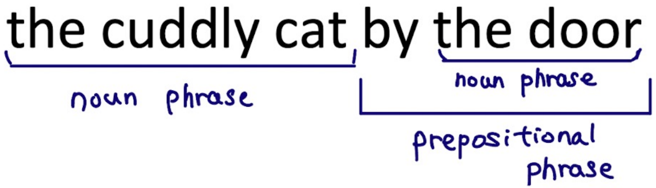
vs

경찰이 칼로 남자를 죽였다

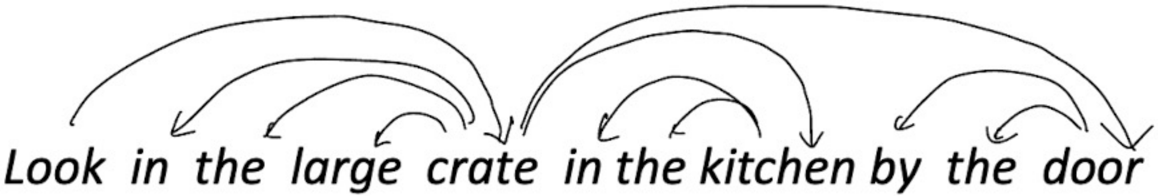
=> 모호성 문제

Dependency parsing

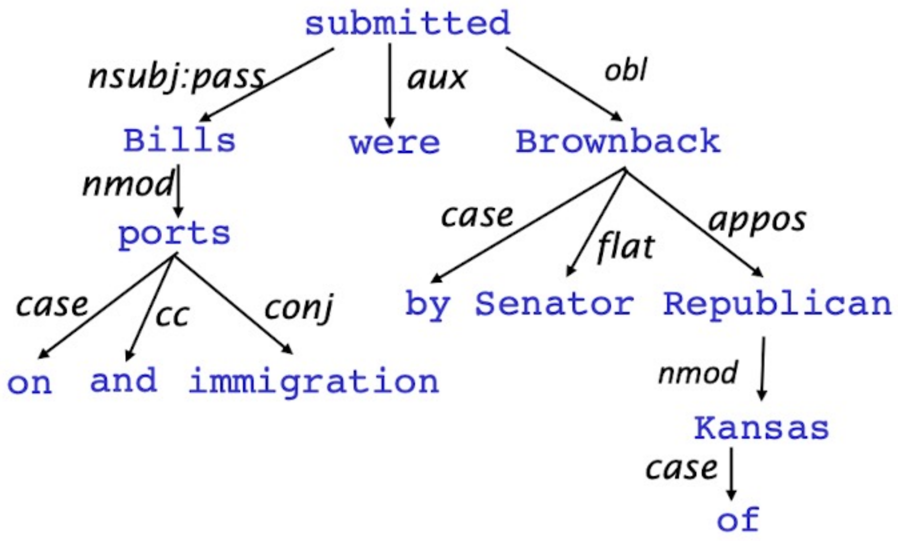
1) Phrase Structure



2) Dependency Structure



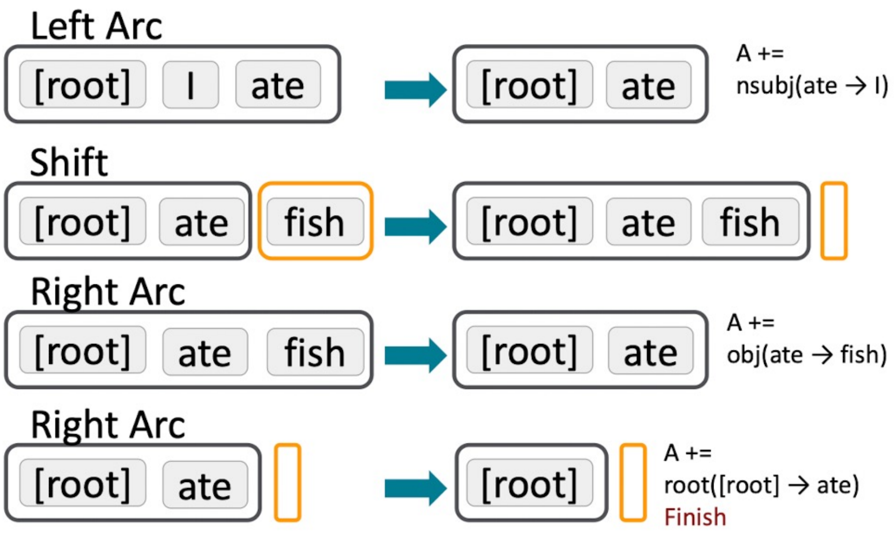
Dependency parsing



- Dynamic programming
- Graph algorithms
- Constraint satisfaction
- Transition-based parsing

Dependency parsing

Transition-based parsing



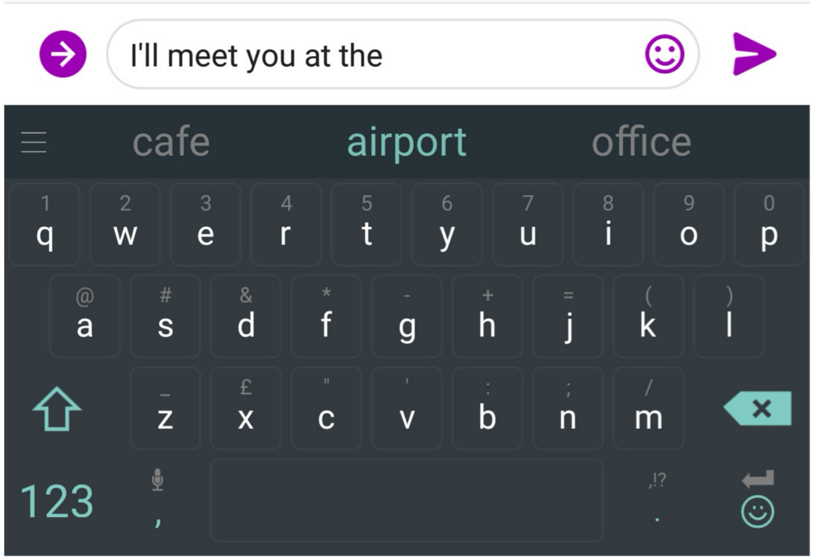
Dependency parsing

Parser	UAS	LAS	sent. / s
MaltParser	89.8	87.2	469
MSTParser	91.4	88.1	10
TurboParser	**92.3**	89.6	8
C & M 2014	92.0	**89.7**	**654**

→ Neural Dependency Parser

Language Model

Language Model : 문장 다음에 올 단어를 예측하는 일



N-gram Language Model

$\cancel{\text{as the proctor started the clock, the}}$ $\underbrace{\text{students opened their}}_{\text{condition on this}}$ ______

discard

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

N-gram Language Model

Problems with n-gram Language Model

Sparsity Problems

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count(students opened their } \boldsymbol{w}) \text{ = 0}}{\text{count(students opened their) = 0}}$$

N-gram Language Model

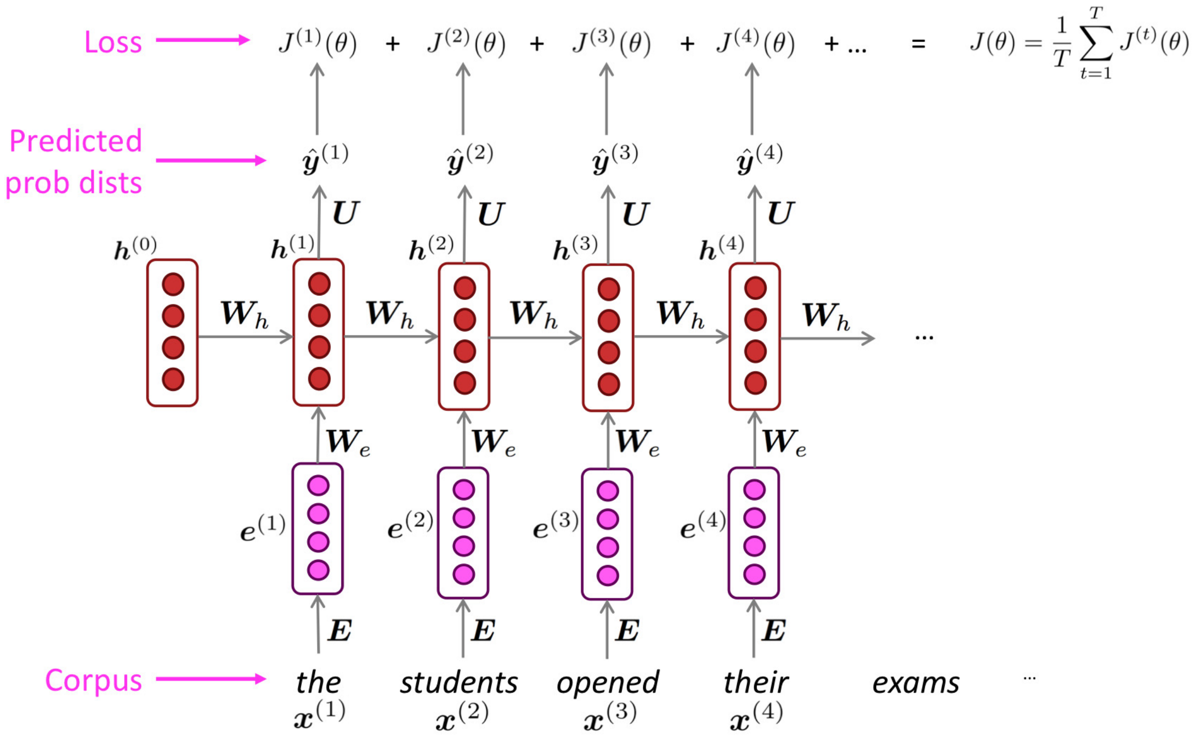
Problems with n-gram Language Model

Storage Problems

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\boxed{\text{count(students opened their } \boldsymbol{w})}}{\text{count(students opened their)}}$$

RNN

Recurrent Neural Network(RNNs)



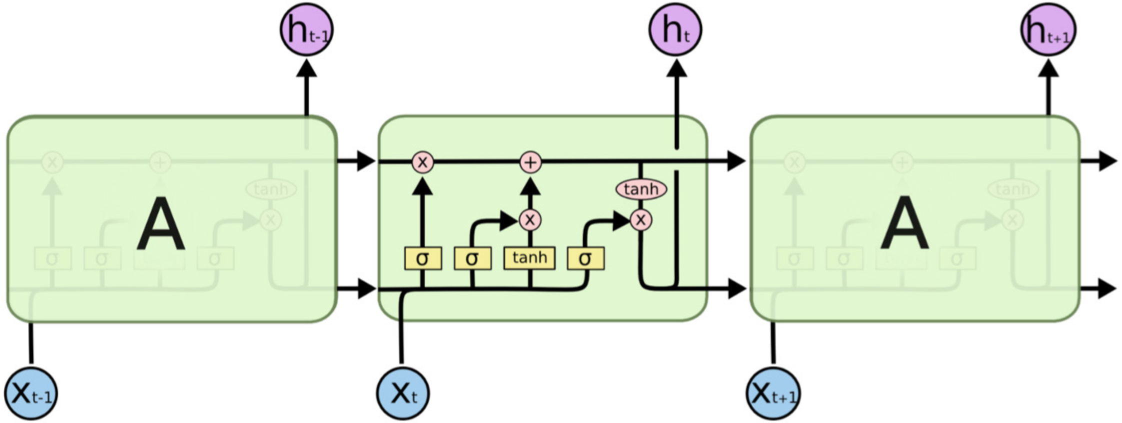
RNN

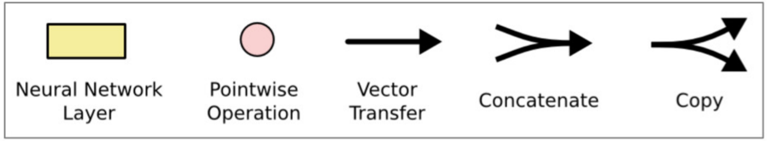
Recurrent Neural Network(RNNs)

장점	단점
Input의 크기에 구애 받지 않음	계산이 느림
Input의 크기가 늘어나도 모델의 크기는 늘어나지 않음	
같은 W를 적용해 input 처리 과정이 대칭적임	아주 먼 단어의 정보를 쓰기가 어려움
아주 먼 단어의 정보도 사용할 수 있음	

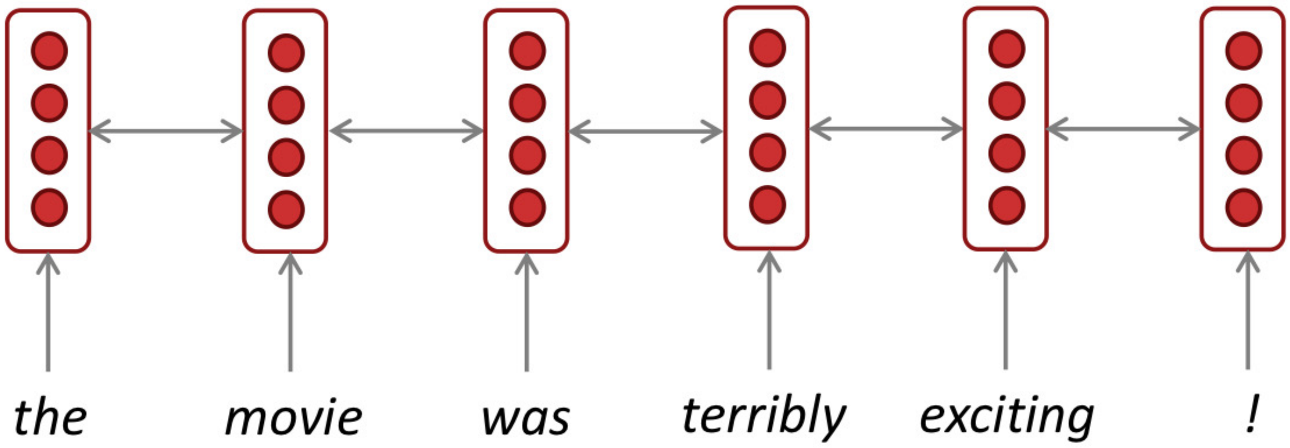
LSTM

Long Short-Term Memory RNNS(LSTMs)

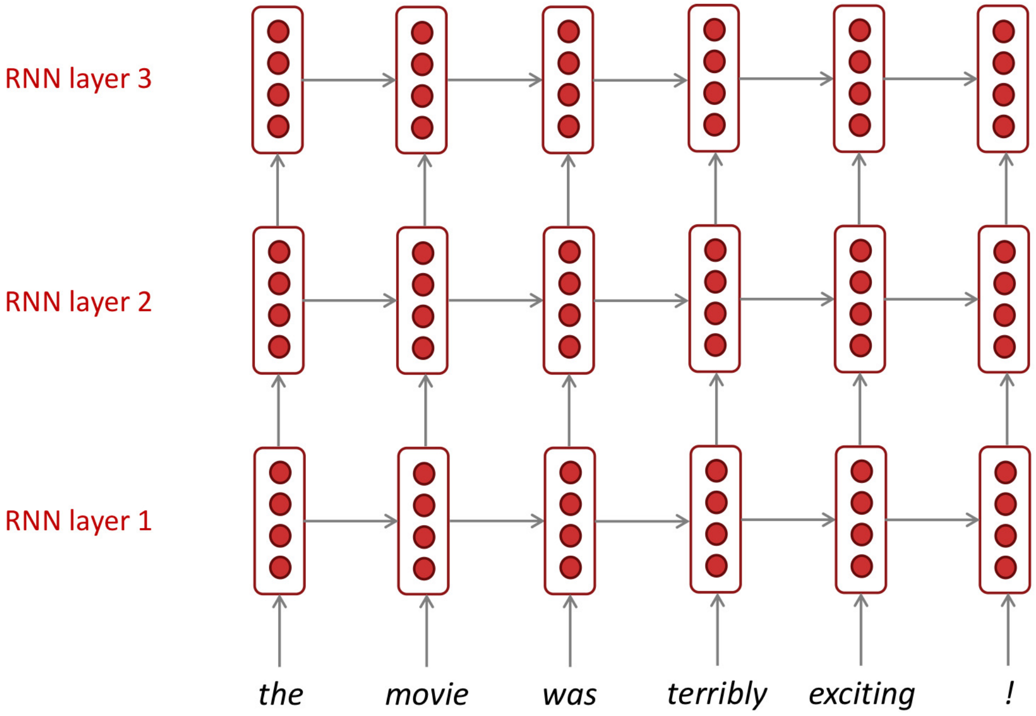




Bidirectional RNNs



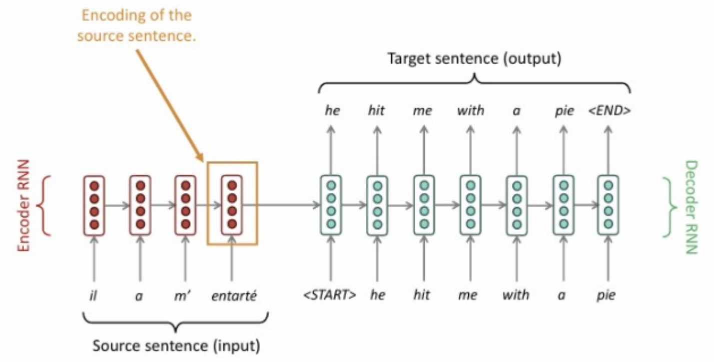
Multi-layer RNNs



기존 RNN의 문제

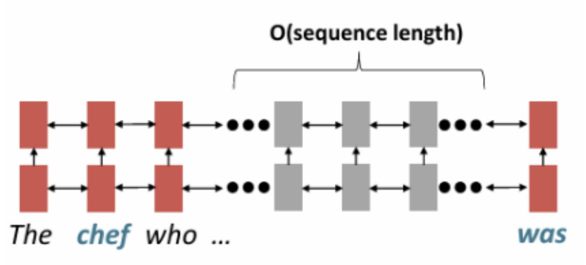
1. Bottleneck problem

- Encoder의 마지막 block은 input sentence의 모든 정보를 담아야 함



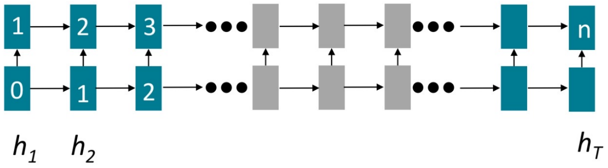
2. Long-distance dependency problem

- 서로 다른 단어가 상호작용하기 위해 O(sequence length) 만큼의 step이 필요함
- LSTM으로도 위 문제를 완벽하게 극복하기 어려움



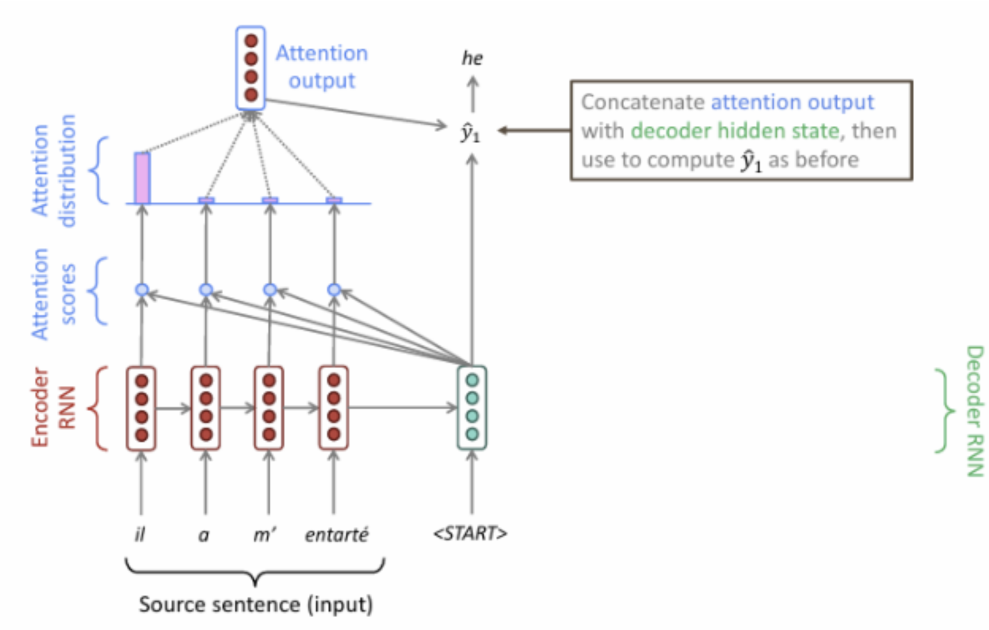
3. Lack of parallelizability

- 연산을 병렬로 수행할 수 없음
- 이전 hidden state가 계산 되기 전까지 그 다음 hidden state를 계산할 수 없기 때문



Attention

Core idea : Decoder에서 encoder에 direct connection을 사용, input sentence의 특정 부분에 집중하자!

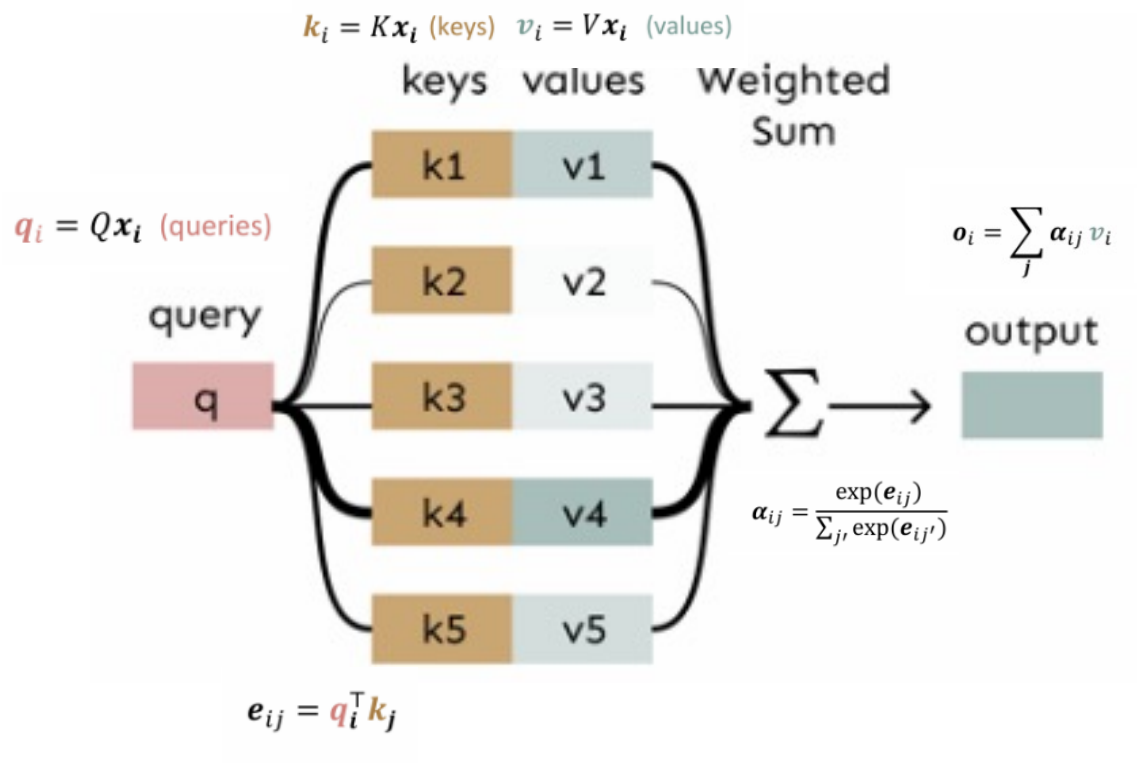


Example of attention structure

이론

Self-Attention

Attention과 차이점 : Query, Key, Value가 모두 동일한 embedding vector에서 도출



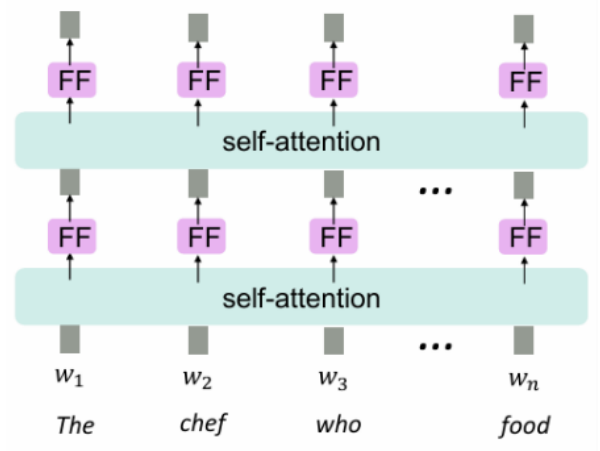
Transformer

1. Positional Representation

$p_i \in \mathbb{R}^d$, for $i \in \{1,2,\ldots,n\}$ are position vectors

$$\tilde{x}_i = x_i + p_i$$

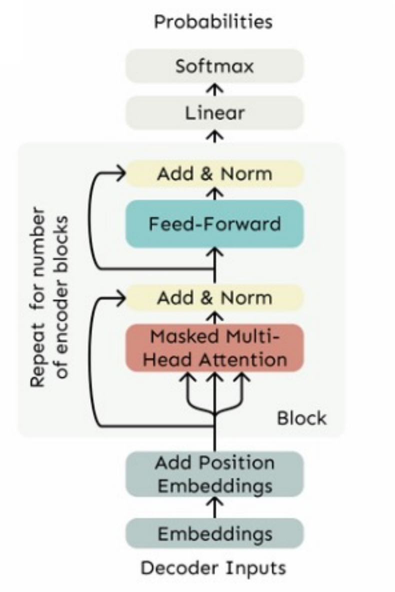
2. Feed-forward Network

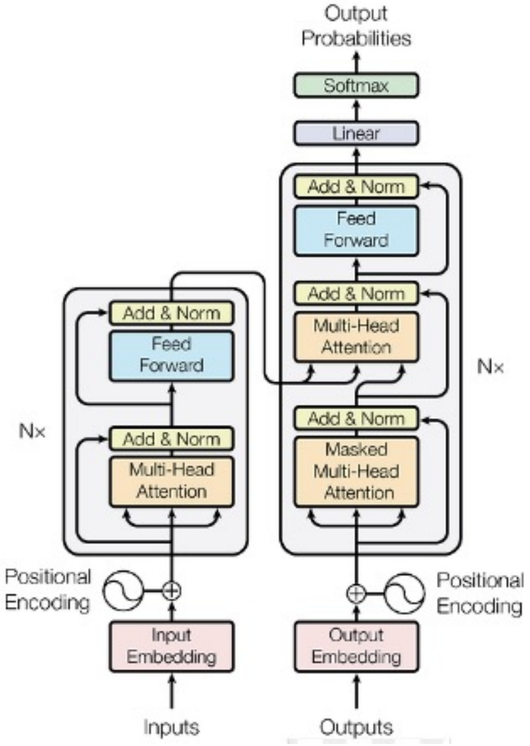


3. Masking

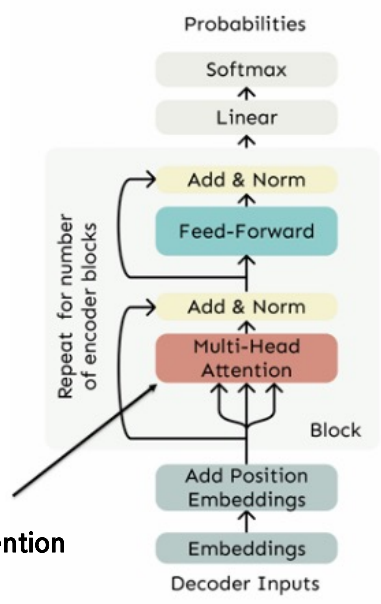
Transformer

Encoder





Decoder



Masked Multi-Head Attention
-> Multi-Head Attention