3장.숫자 세계로 떠난 자연어_파트1

일요일은 AI 김정현

언어모델

순방향 언어모델

그림 3-1 순방향(→) 언어 모델

역방향 언어모델

그림 3-2 역방향(←) 언어 모델

언어모델

마스크 언어모델

어제	카페	갔었어	거기	사람	많더라
어제	카페	갔었어	거기	사람	많더라
어제	카페	갔었어	거기	사람	많더라
어제	카페	갔었어	거기	사람	많더라
어제	카페	갔었어	거기	사람	많더라
어제	카페	갔었어	거기	사람	많더라

그림 3-3 마스크 언어 모델

스킵-그램 모델

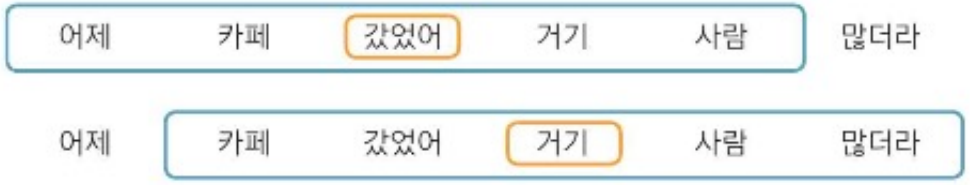
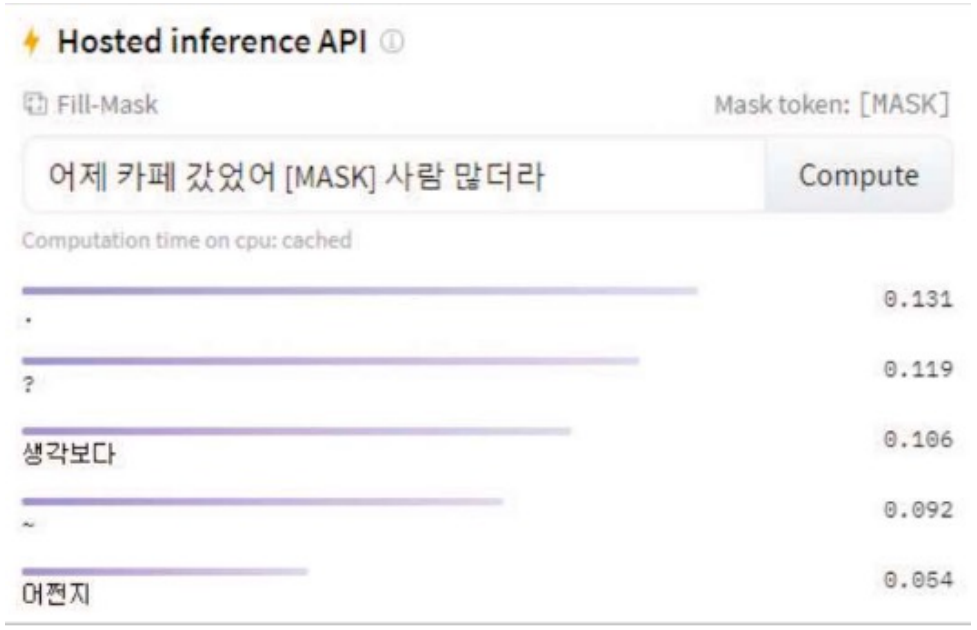


그림 3-4 스킵-그램 모델

KcBERT

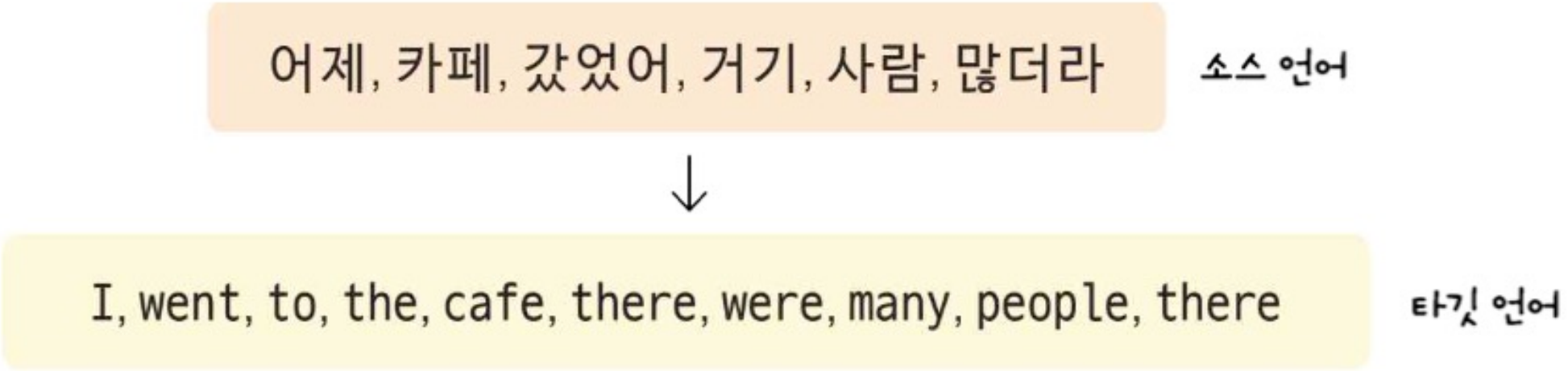
계산결과1

계산결과2



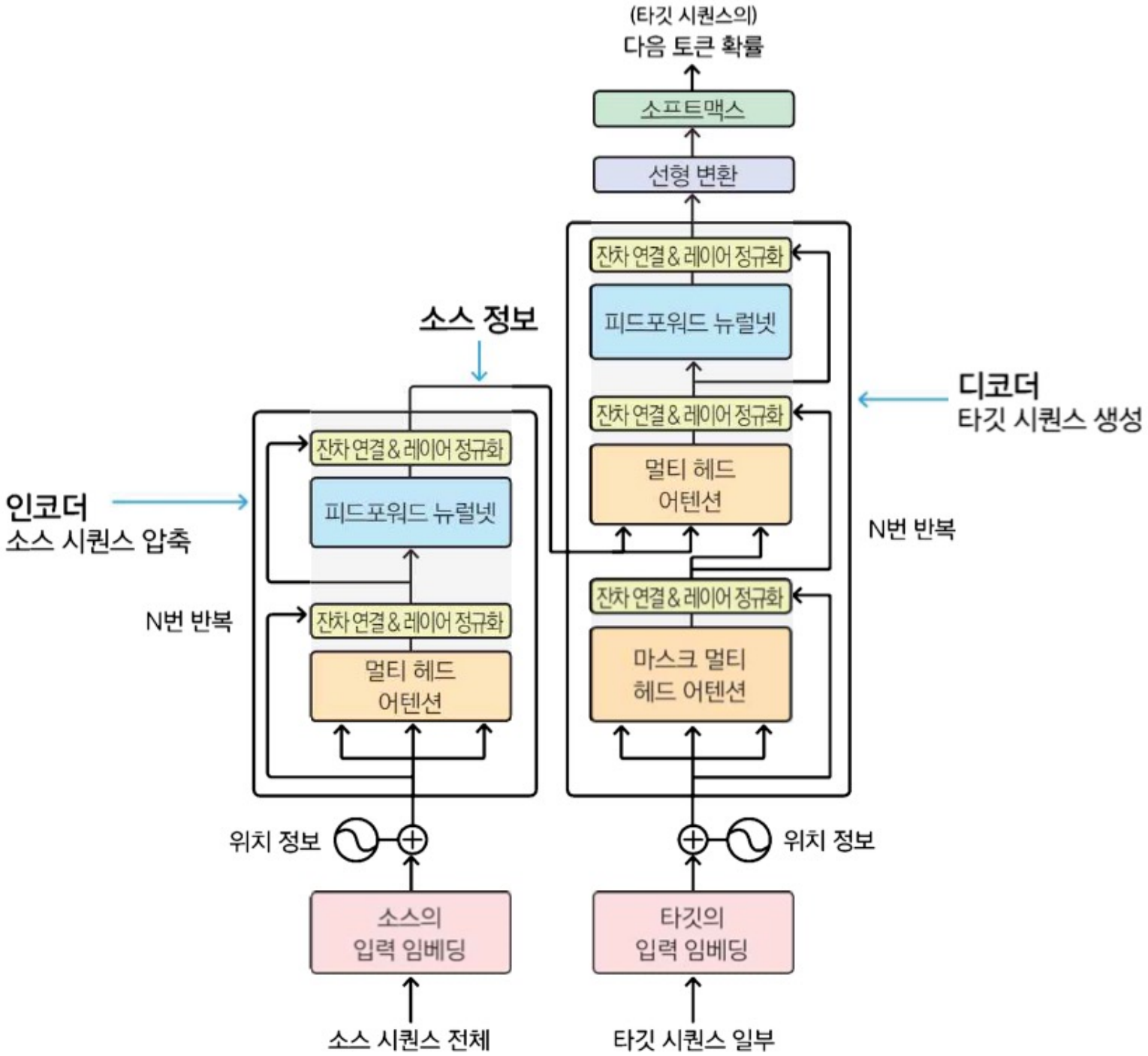
트랜스퍼 러닝 (전이 학습)

시퀀스-투-시퀀스


어제, 카페, 갔었어, 거기, 사람, 많더라
소스 언어
I, went, to, the, cafe, there, were, many, people, there
타깃 언어

인코더와 디코더

그림 3-7 인코더, 디코더


(타깃 시퀀스의)
다음 토큰 확률
소프트맥스
선형 변환
잔차 연결 & 레이어 정규화
피드포워드 뉴럴넷
잔차 연결 & 레이어 정규화
멀티 헤드
어텐션
잔차 연결 & 레이어 정규화
마스크 멀티
헤드 어텐션
소스 정보
디코더
타깃 시퀀스 생성
인코더
소스 시퀀스 압축
N번 반복
잔차 연결 & 레이어 정규화
피드포워드 뉴럴넷
N번 반복
잔차 연결 & 레이어 정규화
멀티 헤드
어텐션
위치 정보
위치 정보
소스의
입력 임베딩
타깃의
입력 임베딩
소스 시퀀스 전체
타깃 시퀀스 일부

트랜스포머

모델 학습과 인퍼런스

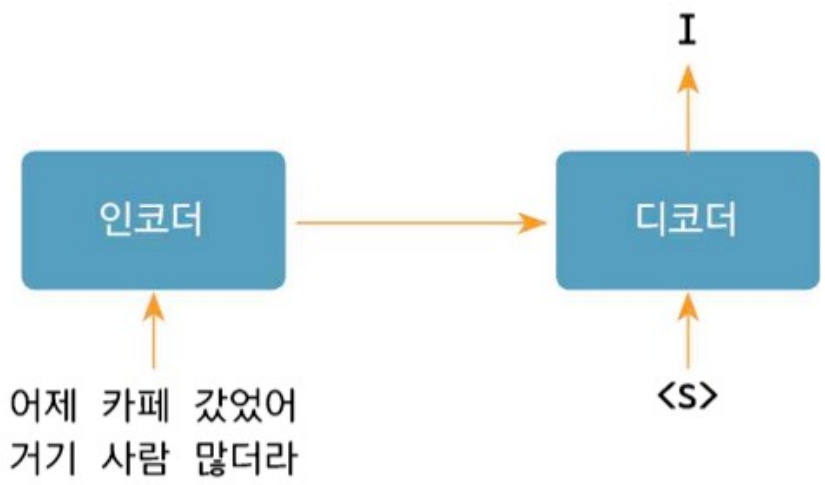


그림 3-9 'i'를 맞히는 학습

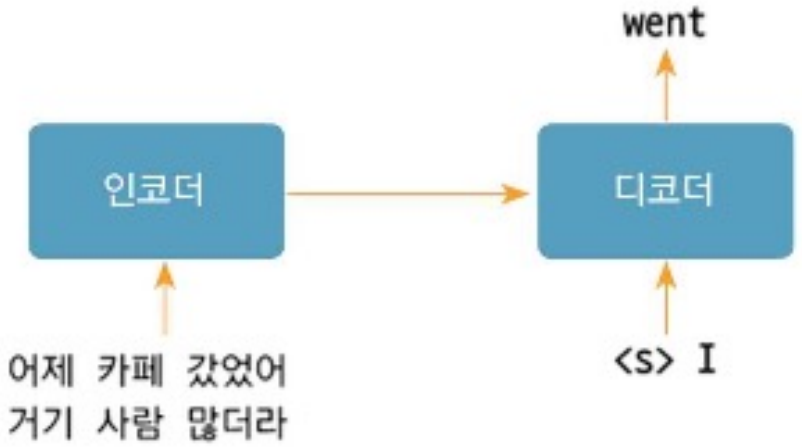


그림 3-11 'went'를 맞히는 학습

학습 과정 중 인코더 입력은 소스 시퀀스 전체, 디코더 입력은 정답 값, 인퍼런스 할 때 디코더 입력은 직전 디코딩 결과

트랜스포머 블록

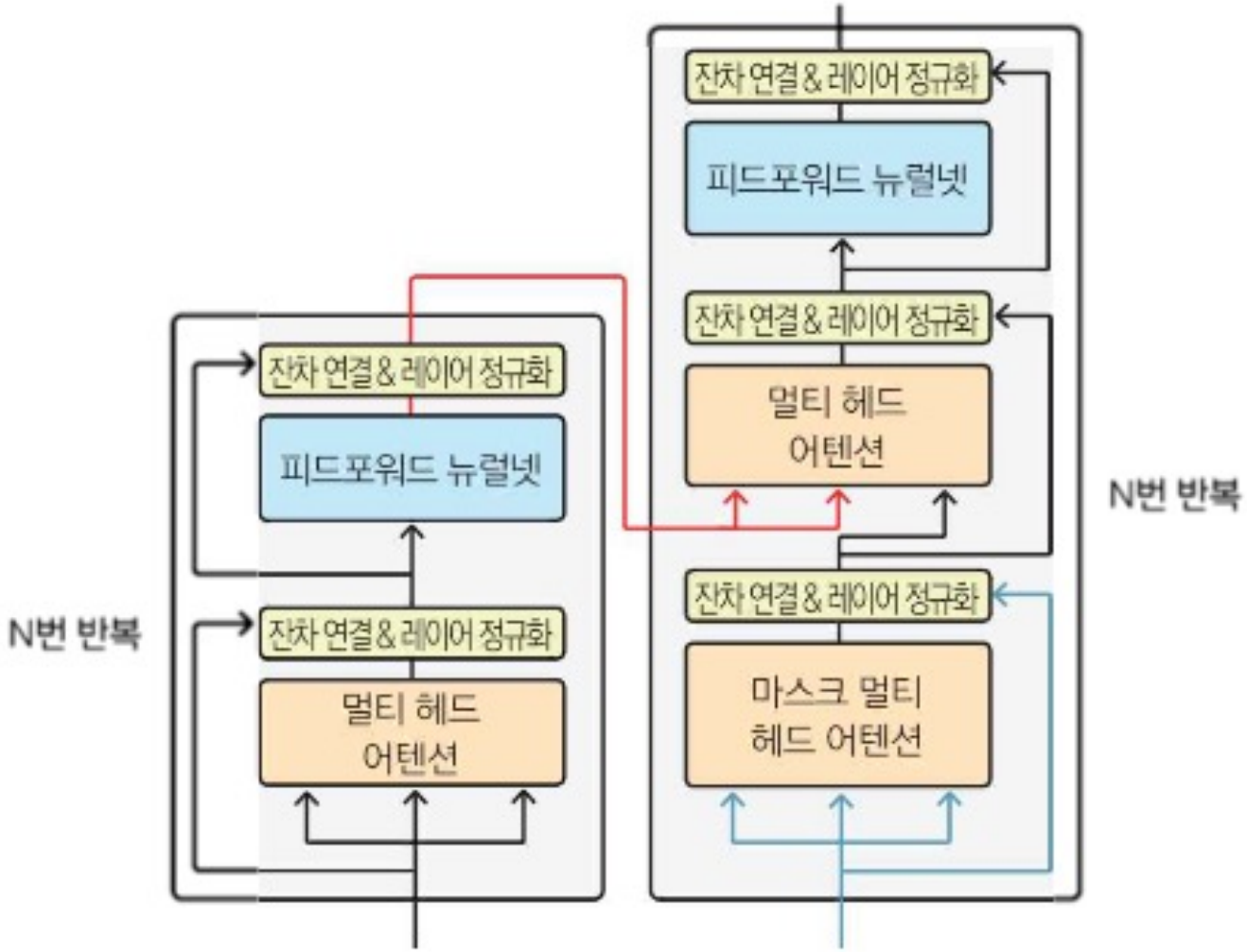


그림 3-31 인코더-디코더

트랜스포머 입력

- 소스 언어의 토큰 인덱스 시퀀스를 이에 대응하는 벡터 시퀀스로 변환해 인코더, 디코더 입력을 만든다.
- 인코더 입력층에서 만들어진 벡터 시퀀스가 최초 인코더 블록의 입력이 되며, 그 출력 벡터 시퀀스가 두번째 인코더 블록의 입력이 된다.
- 다음 인코더 블록의 입력은 이전 블록의 출력이다. (N번 반복)

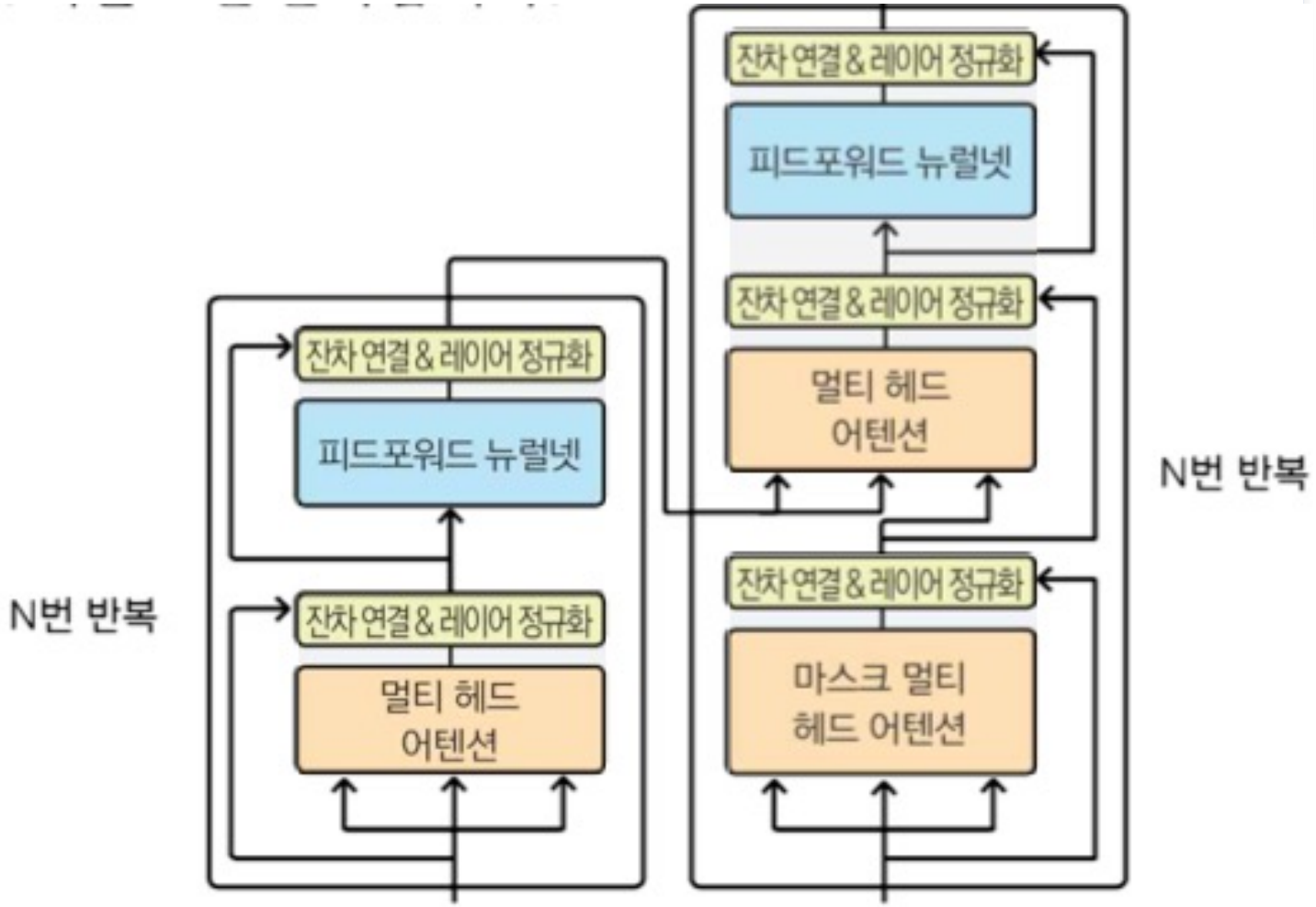


그림 3-25 인코더-디코더

인코더 입력

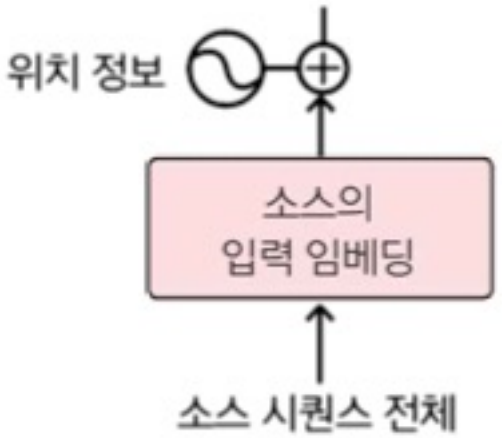


그림 3-23 인코더 입력

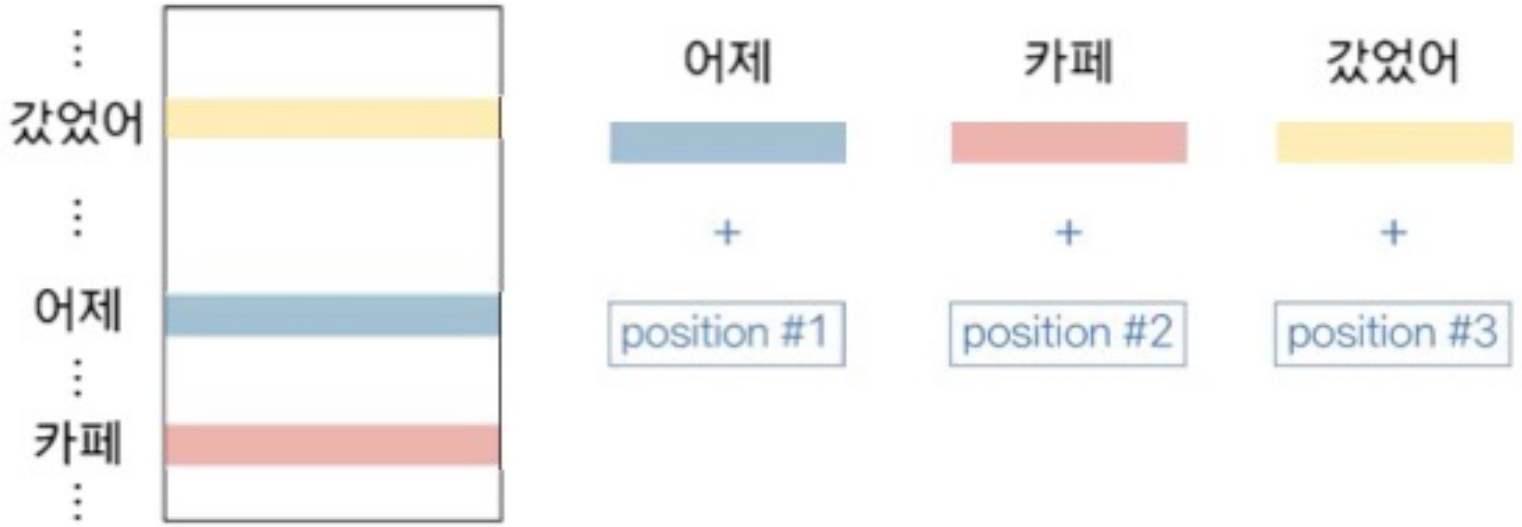


그림 3-24 인코더 입력 예시

트랜스포머 출력

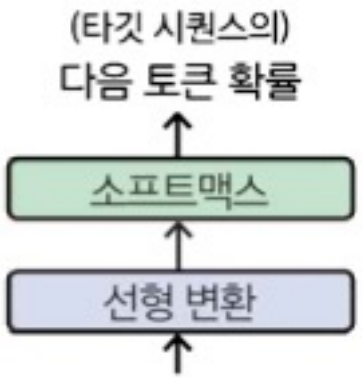


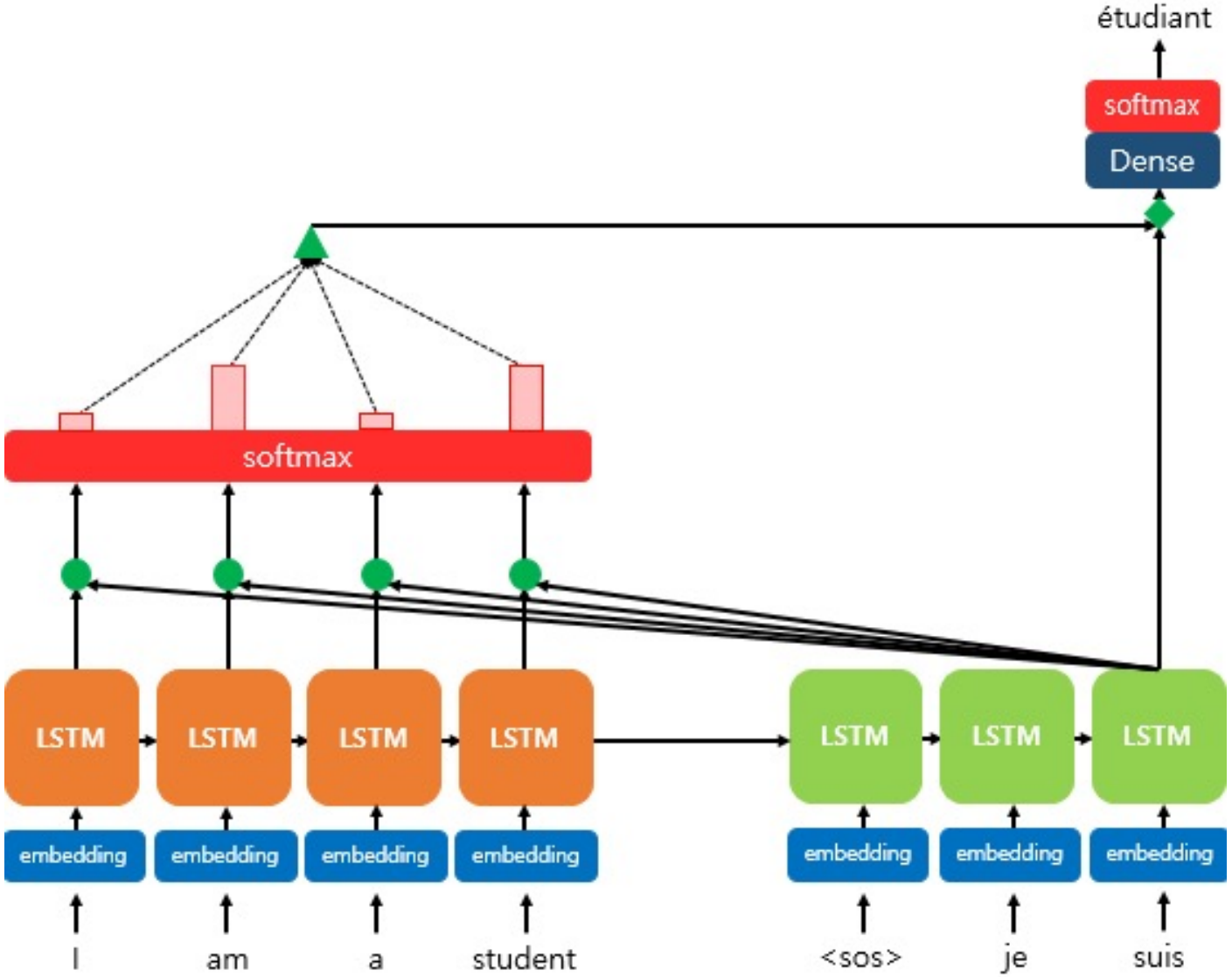
그림 3-26 디코더 출력

- 출력층의 입력은 디코더 마지막 블록의 출력 벡터 시퀀스
- 출력층의 출력은은 타깃 언어의 어휘 수만큼의 차원을 갖는 벡터
- 만약 타깃 언어의 어휘가 3만개이면 이 벡터의 차원수는 3만이 되며, 3만개 요솟값을 모두 더하면 그 합은 1이 된다.
- 이 벡터는 디코더에 입력된 타깃 시퀀스의 다음 토큰 확률 분포를 가리킨다.
- 트랜스포머의 학습은 인코더와 디코더 입력이 주어졌을 때 모델 최종 출력에서 정답에 해당하는 단어의 확률값을 높이는 방식으로 수행된다.

출처 : https://wikidocs.net/22893

어텐션

디코더의 세번째 LSTM 셀은 출력 단어를 예측하기 위해 인코더의 모든 입력단어들의 정보를 다시 한번 참고한다.



셀프 어텐션

셀프 어텐션 vs 합성곱 신경망

그림 3-17 합성곱 신경망

합성곱 필터의 크기를 넘어서는 문맥은 읽어내기 어렵다.

셀프 어텐션은 문맥 전체를 고려하여 지역적인 문맥만 보는 CNN보다 강점이 있다.

셀프 어텐션 vs 순환 신경망

그림 3-18 순환 신경망

순환 신경망은 입력정보를 차례대로 처리하고 오래전에 읽었던 단어는 잊어버린다.

셀프 어텐션은 시퀀스 길이가 길어지더라도 정보를 잊어버리지 않는다.

셀프 어텐션 vs 어텐션

그림 3-19 'cafe'의 어텐션

디코더 쪽 RNN에 어텐션을 추가하면, 디코더가 타깃 시퀀스를 생성할 때 소스 시퀀스 전체에서
어떤 요소에 주목해야 할지를 알려준다.

어센션은 소스 시퀀스 전체 단어와 타켓 시퀀스 단어 하나 사이를 연결하는데,
셀프 어텐션은 소스 시퀀스 전체 단어들 사이를 연결한다.
어텐션은 RNN 구조위에서 동작, 셀프 어텐션은 RNN 없이 동작
타깃 언어의 단어를 1개 생성할 때 어텐션은 1회 수행, 셀프어텐션은 인코더, 디코더 블록의 갯수 만큼 반복 수행

셀프 어텐션 예시 - 거기

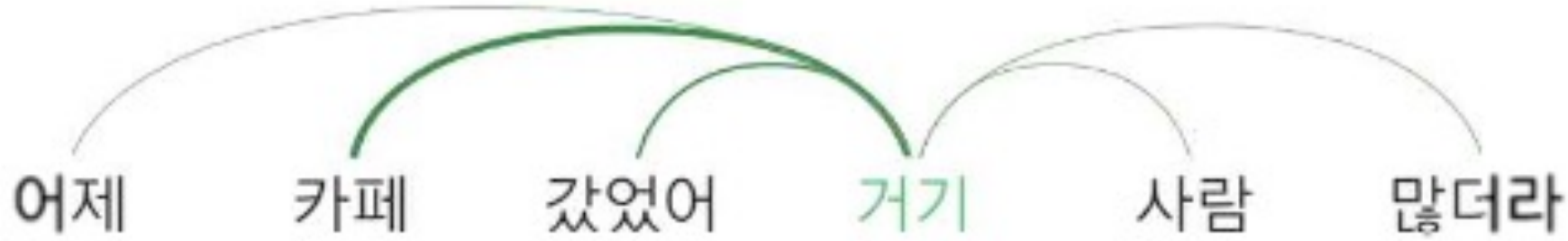


그림 3-20 '거기'의 셀프 어텐션

거기 에 대응하는 장소는 카페 라는 사실을 알아 챌 수 있다. 거기는 갔었어와 연관이 있음을 확인할 수 있다.

트랜스포머의 인코더 블록에서는 거기 라는 단어를 인코딩 할 때 카페, 갔었어 라는 단어의 의미를 강조해서 반영한다.

셀프어텐션 계산

쿼리 단어 각각을 대상으로 모든 키 단어와 얼마나 유기적인 관계를 맺는 지 그 합이 1인 확률값으로 나타낸다.

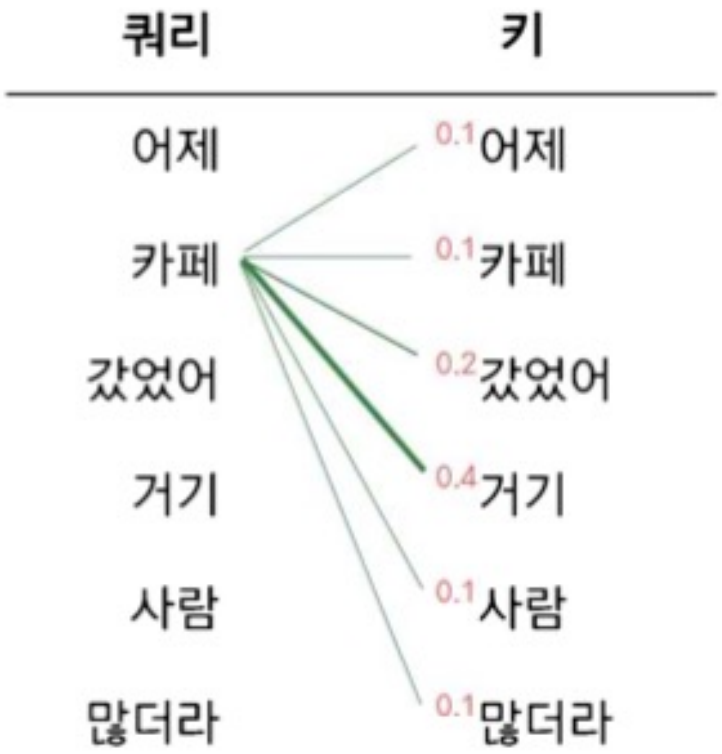


그림 3-22 셀프 어텐션 계산 예시

$$\mathbf{Z}_{\text{카페}} = 0.1\times\mathbf{V}_{\text{어제}} + 0.1\times\mathbf{V}_{\text{카페}} + 0.2\times\mathbf{V}_{\text{갔었어}} + 0.4\times\mathbf{V}_{\text{거기}} + 0.1\times\mathbf{V}_{\text{사람}} + 0.1\times\mathbf{V}_{\text{많더라}}$$

수식 3-6 셀프 어텐션 계산 예시

단어 임베딩
차원수(d) = 4 단어 개수 3개

- X는 4차원짜리 단어 임베딩이 3개 모인 행렬
- 셀프 어텐션은 쿼리, 키, 밸류 3개 요소사이의 문맥적 관계성을 추출하는 과정
- Wq,Wk,Wv 세 가지 행렬은 훈련을 통해 업데이트 된다.

$$X = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

수식 3-7 입력 벡터 시퀀스 X

$$Q = X \times W_Q$$
$$K = X \times W_K$$
$$V = X \times W_V$$

수식 3-8 쿼리, 키, 밸류 만들기

Q

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

수식 3-12 쿼리 만들기 (4)

K

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 4 & 4 & 0 \\ 2 & 3 & 1 \end{bmatrix}$$

수식 3-13 키 만들기

V

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & 0 \\ 0 & 3 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 0 \\ 2 & 6 & 3 \end{bmatrix}$$

수식 3-14 밸류 만들기

첫 번째 쿼리의 셀프 어텐션 출력값 계산하기

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_{\text{K}}}})\mathbf{V}$$

수식 3-15 셀프 어텐션 정의

$$\begin{bmatrix}1 & 0 & 2\end{bmatrix} \times \begin{bmatrix}0 & 4 & 2\\ 1 & 4 & 3\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}2 & 4 & 4\end{bmatrix}$$

수식 3-17 첫 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (1)

$$\text{softmax}\left(\begin{bmatrix}\frac{2}{\sqrt{3}} & \frac{4}{\sqrt{3}} & \frac{4}{\sqrt{3}}\end{bmatrix}\right) = \begin{bmatrix}0.13613 & 0.43194 & 0.43194\end{bmatrix}$$

수식 3-18 첫 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (2)

$$\begin{bmatrix}0.13613 & 0.43194 & 0.43194\end{bmatrix} \times \begin{bmatrix}1 & 2 & 3\\ 2 & 8 & 0\\ 2 & 6 & 3\end{bmatrix} = \begin{bmatrix}1.8639 & 6.3194 & 1.7042\end{bmatrix}$$

수식 3-19 첫 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (3)

두 번째 쿼리의 셀프 어텐션 출력값 계산하기

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\text{K}}})\mathbf{V}$$

수식 3-15 셀프 어텐션 정의

$$\begin{bmatrix}2 & 2 & 2\end{bmatrix} \times \begin{bmatrix}0 & 4 & 2\\ 1 & 4 & 3\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}4 & 16 & 12\end{bmatrix}$$

수식 3-20 두 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (1)

$$\text{softmax}\left(\begin{bmatrix}\frac{4}{\sqrt{3}} & \frac{16}{\sqrt{3}} & \frac{12}{\sqrt{3}}\end{bmatrix}\right) = \begin{bmatrix}0.00089 & 0.90884 & 0.09027\end{bmatrix}$$

수식 3-21 두 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (2)

$$\begin{bmatrix}0.00089 & 0.90884 & 0.09027\end{bmatrix} \times \begin{bmatrix}1 & 2 & 3\\ 2 & 8 & 0\\ 2 & 6 & 3\end{bmatrix} = \begin{bmatrix}1.9991 & 7.8141 & 0.2735\end{bmatrix}$$

수식 3-22 두 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (3)

세 번째 쿼리의 셀프 어텐션 출력값 계산하기

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\text{K}}})\mathbf{V}$$

수식 3-15 셀프 어텐션 정의

$$\begin{bmatrix}2 & 1 & 3\end{bmatrix} \times \begin{bmatrix}0 & 4 & 2\\ 1 & 4 & 3\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}4 & 12 & 10\end{bmatrix}$$

수식 3-23 세 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (1)

$$\text{softmax}\left(\begin{bmatrix}\frac{4}{\sqrt{3}} & \frac{12}{\sqrt{3}} & \frac{10}{\sqrt{3}}\end{bmatrix}\right) = \begin{bmatrix}0.00744 & 0.75471 & 0.23785\end{bmatrix}$$

수식 3-24 세 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (2)

$$\begin{bmatrix}0.00744 & 0.75471 & 0.23785\end{bmatrix} \times \begin{bmatrix}1 & 2 & 3\\ 2 & 8 & 0\\ 2 & 6 & 3\end{bmatrix} = \begin{bmatrix}1.9926 & 7.4796 & 0.7359\end{bmatrix}$$

수식 3-25 세 번째 쿼리 벡터에 관한 셀프 어텐션 계산 (3)

코드로 확인하기

- https://github.com/kimjeonghyon/NLP/blob/main/transformer/selfattention_example.ipynb

멀티헤드 어텐션

- 셀프어텐션을 동시에 여러번 수행하는 것을 말한다.
- 오른쪽 그림은 입력 단어 수는 2개, 밸류의 차원은 3, 헤드는 8개인 멀티헤드 어텐션을 나타낸다.
- W0의 크기는 셀프어텐션 결과 행렬의 열수 * 목표차원수 이다.
- 목표차원수를 3차원으로 하고 싶으면 W0은 24 x 3의 행렬이다.

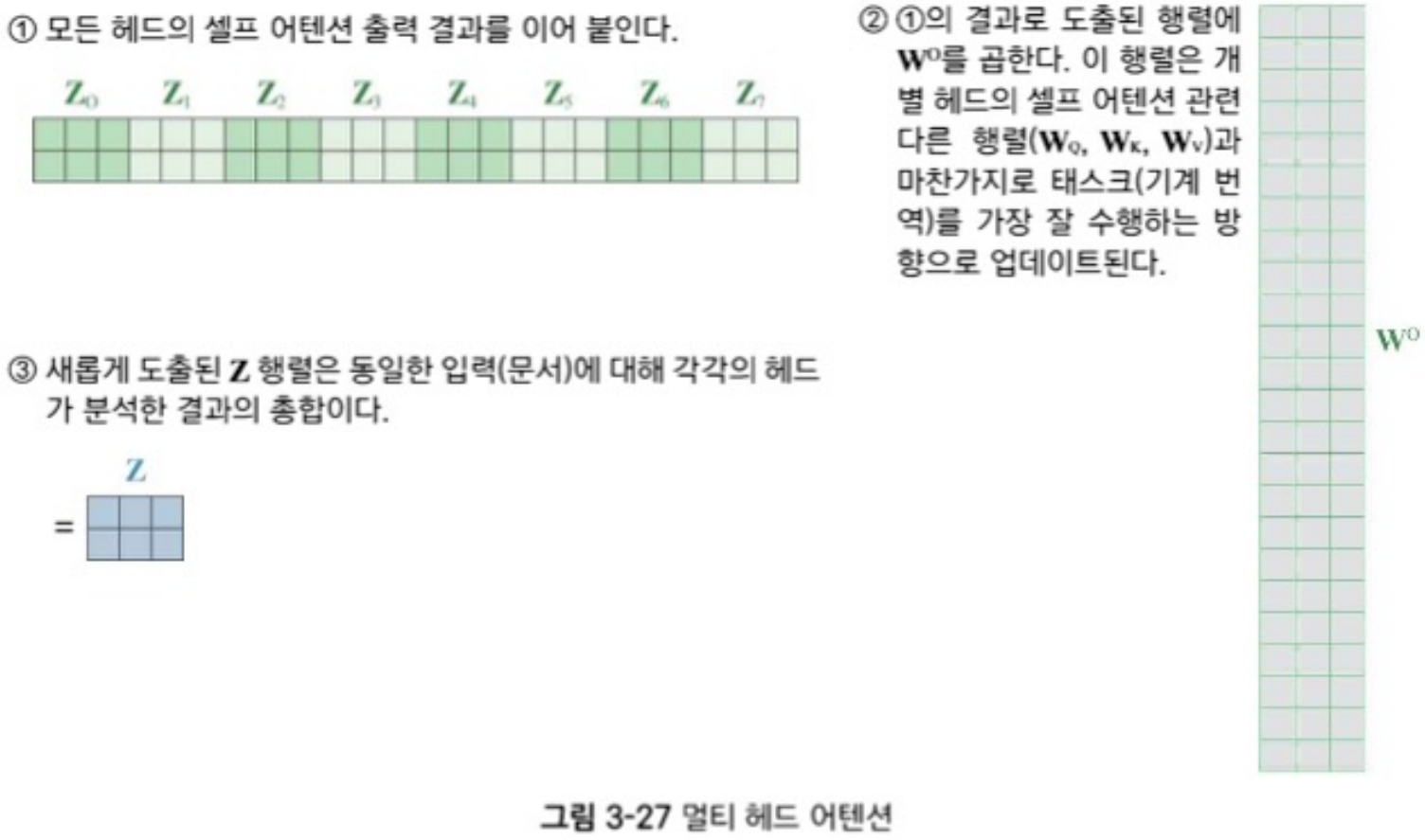


그림 3-27 멀티 헤드 어텐션

* 이후로 셀프어텐션은 멀티헤드 어텐션을 뜻한다.

인코더의 셀프 어텐션

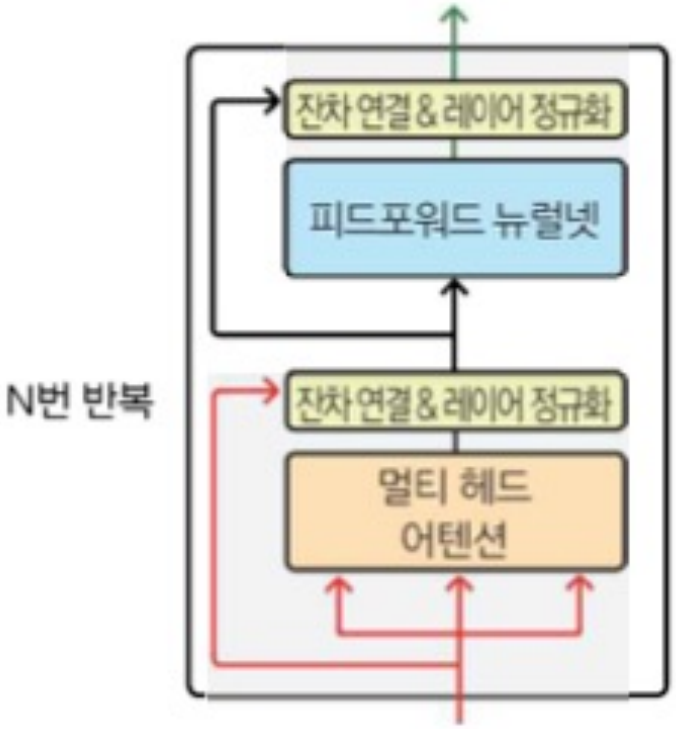


그림 3-28 트랜스포머 인코더 블록

그림 3-30 쿼리가 '카페'일 때 셀프 어텐션

인코더에서 수행하는 셀프어텐션은 소스 시퀀스 내의 모든 단어 쌍 사이의 관계를 나타낸다.

디코더의 셀프 어텐션 (구조도는 4페이지 참조)

[마스크 멀티 헤드 어텐션]

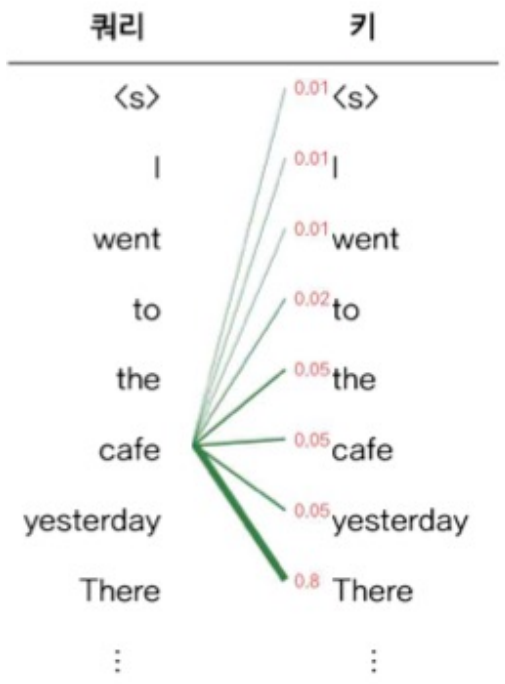


그림 3-32 타깃 문장의 셀프 어텐션

한국어를 영어로 번역하는 기계번역 태스크라고 하면, 영어 단어에 대해 인코더 쪽 셀프 어텐션과 동일한 작업을 한다.

[멀티 헤드 어텐션]

그림 3-33 소스-타깃 문장 간 셀프 어텐션

인코더에서 넘어온 단어 벡터 시퀀스를 키로, 디코더 직전 블록에서 넘어온 단어 벡터 시퀀스를 쿼리로 삼아 셀프 어텐션 수행

디코더의 학습시 수행되는 셀프어텐션

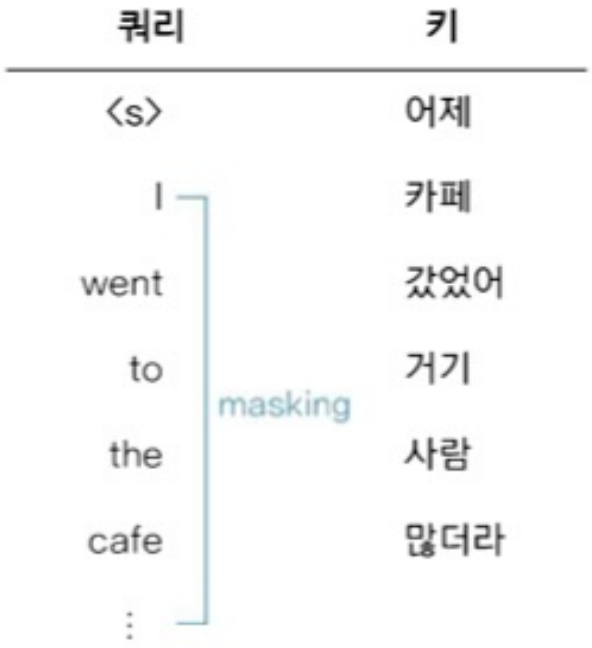


그림 3-34 학습 시 디코더에서 수행되는 셀프 어텐션(1)

마스킹은 소프트맥스 확률이 0이 되도록 하여, 해당 단어 정보가 무시되게끔 한다.
<s> 벡터를 가지고 I를 맞히도록 학습한다.

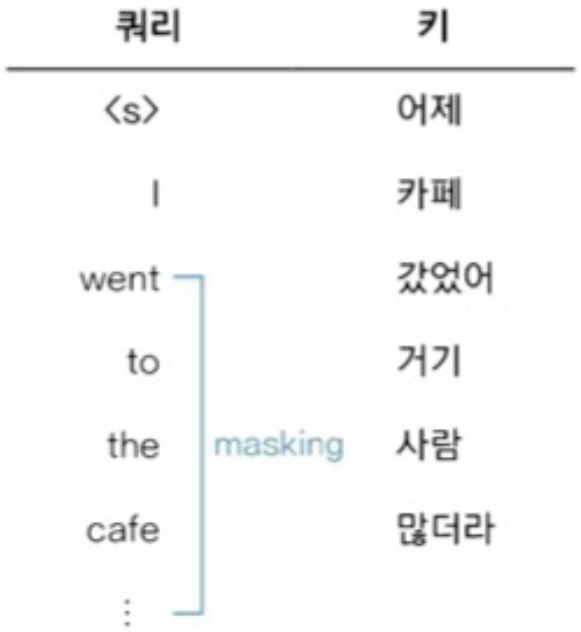


그림 3-36 학습 시 디코더에서 수행되는 셀프 어텐션(2)

다음은, 디코더에 <s> I 가 입력된 상황.

I 벡터를 가지고 went를 맞히도록 학습한다.