10장. 자연어 처리를 위한 임베딩

10.1 임베딩

임베딩: 사람이 사용하는 언어를 컴퓨터가 이해할 수 있는 언어 형태인 벡터로 변환한 결과 혹은 일련의 과정

- 단어 및 문장 간 관련성 계산
- 의미적 혹은 문법적 정보의 함축

1. 희소 표현 기반 임베딩

   희소 표현은 대부분의 값이 0으로 채워져 있는 경우로, 대표적으로 원-핫 인 코딩이 있음

   원-핫 인코딩: 주어진 텍스트를 숫자(벡터)로 변환해 주는 것

   단점: 단어끼리 서로 독립적 관계가 됨, 차원의 저주 문제가 발생

2. 횟수 기반 임베딩

   횟수 기반은 단어가 출현한 빈도를 고려하여 임베딩하는 방법

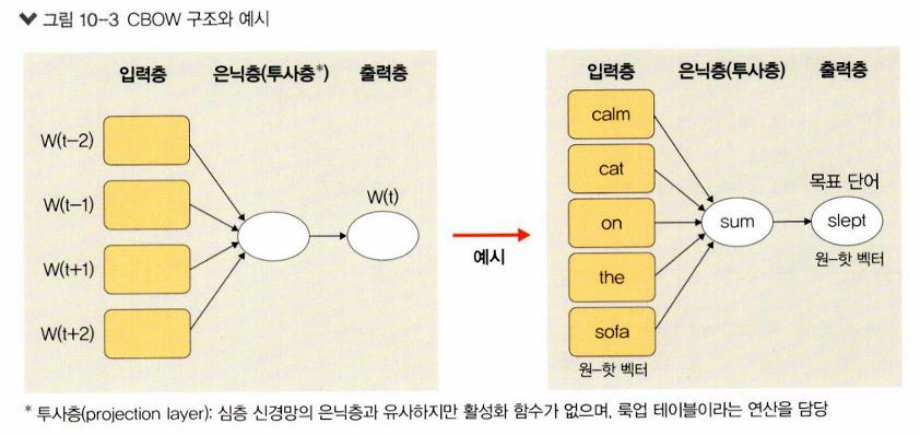
   1) 카운터 벡터
      문서 집합에서 단어를 토큰으로 생성하고 각 단어의 출현 빈도수를 이용하여 인코딩해서 벡터를 만드는 방법

   2) TF-IDF
      정보 검색론에서 가중치 구할 때 사용되는 알고리즘

3. 예측 기반 임베딩
   신경망 구조 혹은 모델을 이용하여 특정 문맥에서 어떤 단어가 나올지 예측하면서 단어를 벡터로 만드는 방식
   Ex) 워드투벡터
   신경망 알고리즘으로, 주어진 텍스트에서 텍스트의 각 단어마다 하나씩 일련의 벡터를 출력함

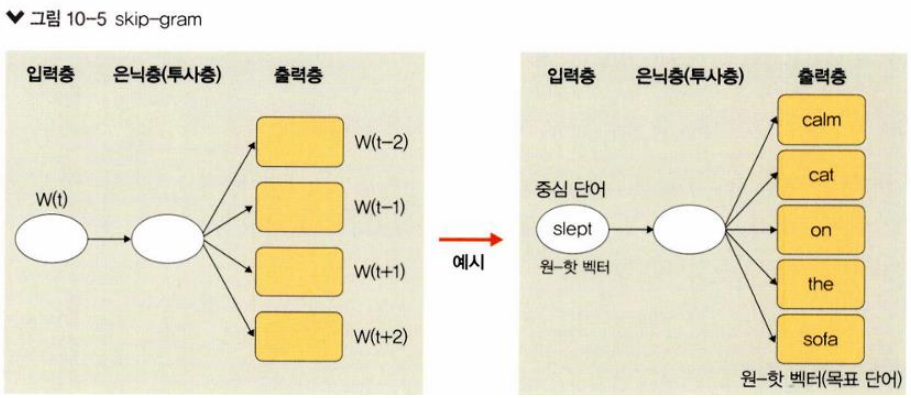
1) CBOW

단어를 여러 개 나열 후 이와 관련된 단어를 추정하는 방식



2) Skip-gram

반대로 특정한 단어에서 문맥이 될 수 있는 단어를 예측함



3) 패스트텍스트

워드투벡터의 단점을 보완하고자 페이스북에서 개발한 임베딩 알고리즘

자주 사용되지 않는 단어에 대해서는 학습이 불안정

4. 횟수/예측 기반 임베딩

1) 글로브

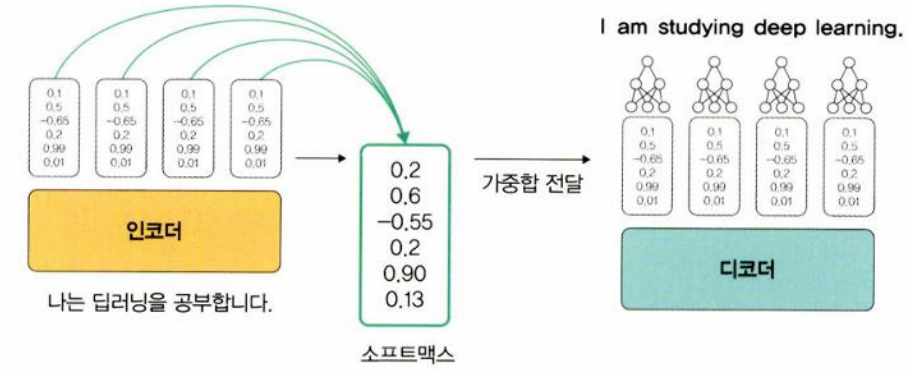
횟수 기반의 LSA와 예측 기반의 워드투벡터 단점을 보완하기 위한 모델

10.2 트랜스포머 어텐션

어텐션은 주로 언어 번역에서 사용되기 때문에 인코더와 디코더 네트워크를 사용함

- 초기 정보를 잃어버리는 기울기 소멸 문제 해결을 위함
- But 모든 벡터가 전달되므로 행렬 크기가 굉장히 커지는 단점

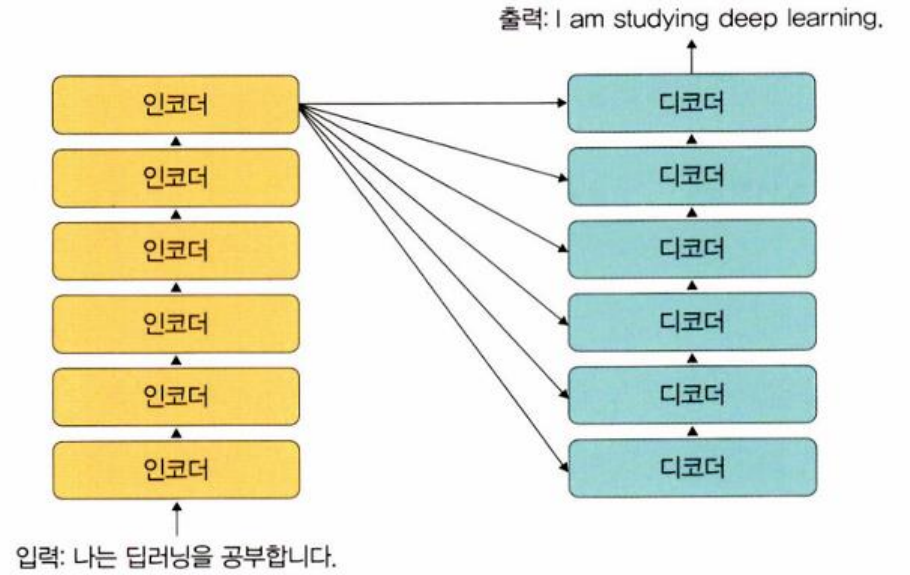
❯ 그림 10-9 어텐션



-

- 트랜스포머는 어텐션을 극대화하는 방법

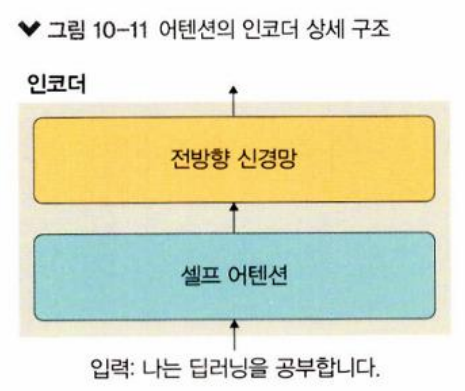
❯ 그림 10-10 어텐션에서 인코더와 디코더



-

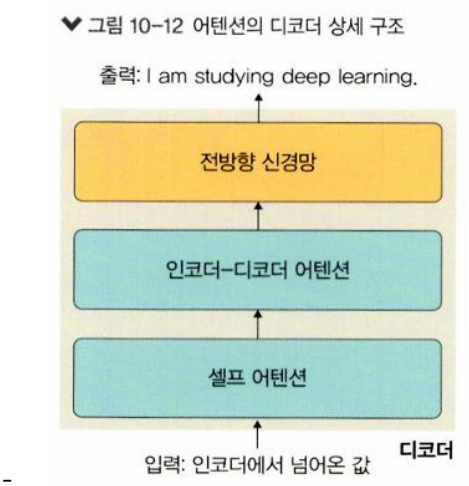
- 인코더 블록 구조는 단어를 벡터로 임베딩하며, 이를 셀프 어텐션과 전방향 신경망으로 전달

❯ 그림 10-11 어텐션의 인코더 상세 구조



-

- 디코더는 세 개의 층을 가지고 있음



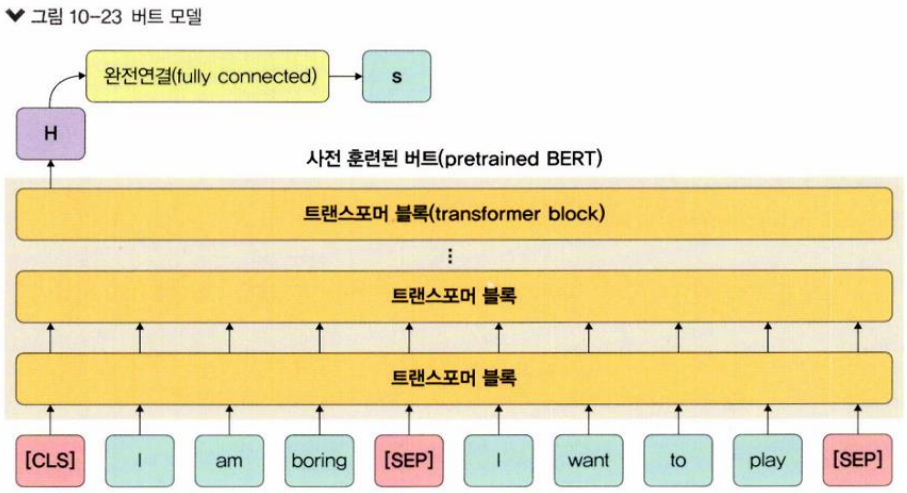
-

10.2.1 seq2seq

입력 시퀀스에 대한 출력 시퀀스를 만들기 위한 모델

10.2.2 bert

언어 모델 BERT는 기존의 단방향 자연어 처리 모델들의 단점을 보완한 양방향 자연어 처리 모델



버트의 기본 구조는 트랜스포머라는 인코더를 쌓아 올린 구조로, 주로 문장 예측을 할 때 사용함