



ChatGPT 와 word embedding

23.05.03 유하영

ChatGPT



Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →



Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests



Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021

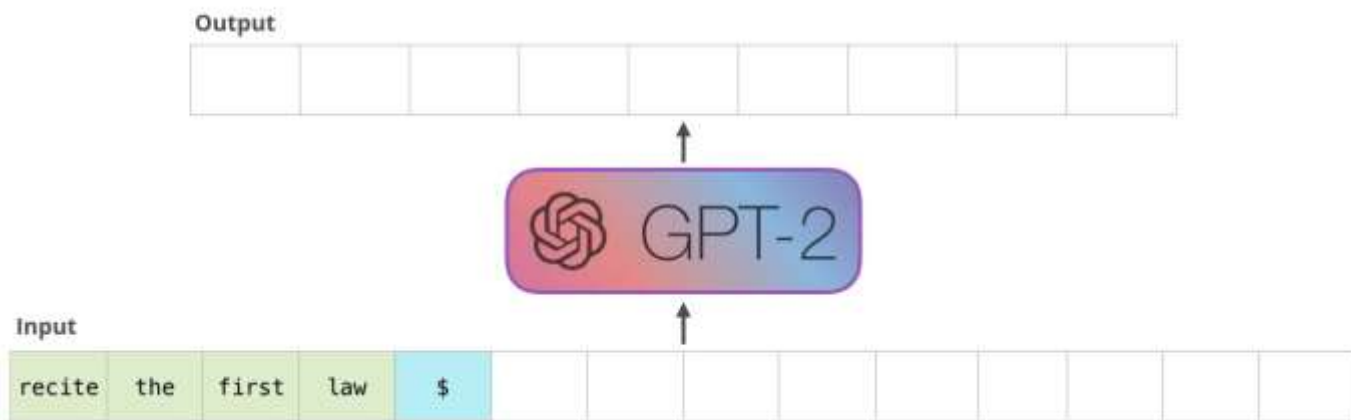
GPT

Generative Pre-trained Transformer

생성하는 사전학습된 변환기

GPT

Generative Pre-trained Transformer



학습 데이터를 학습해서
학습 데이터의 분포를 따르는
유사한 데이터를 생성

GPT

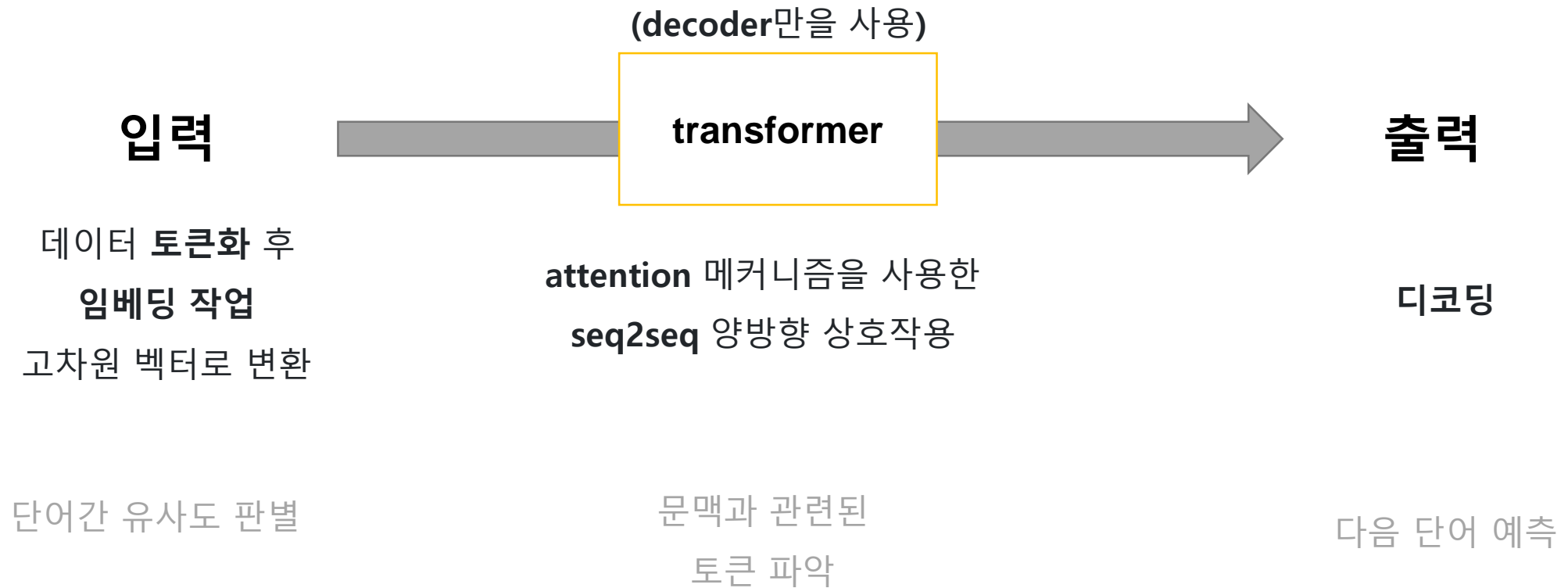
Generative **Pre-trained** Transformer

pre-training

말뭉치를 가지고 사전에 학습

GPT

Generative Pre-trained Transformer



GPT

GPT-1

GPT-2

GPT-3

GPT-3

GPT-3.5

ChatGPT



fine-tuning

특정 task에 맞게 **추가적으로 학습**시키는 것

GPT

GPT-1

- **unsupervised pre-training**
+ supervised fine-tuning
-> narrow expert

GPT-2

- 모델 크기 ↑
- WebText를 훈련 데이터로 이용
- fine-tuning X
- zero-shot, few-shot의 시초

GPT-3

- GPT-2에서 발전시킨 모델
(모델 크기, 데이터셋 크기, 학습횟수, 다양성 등)

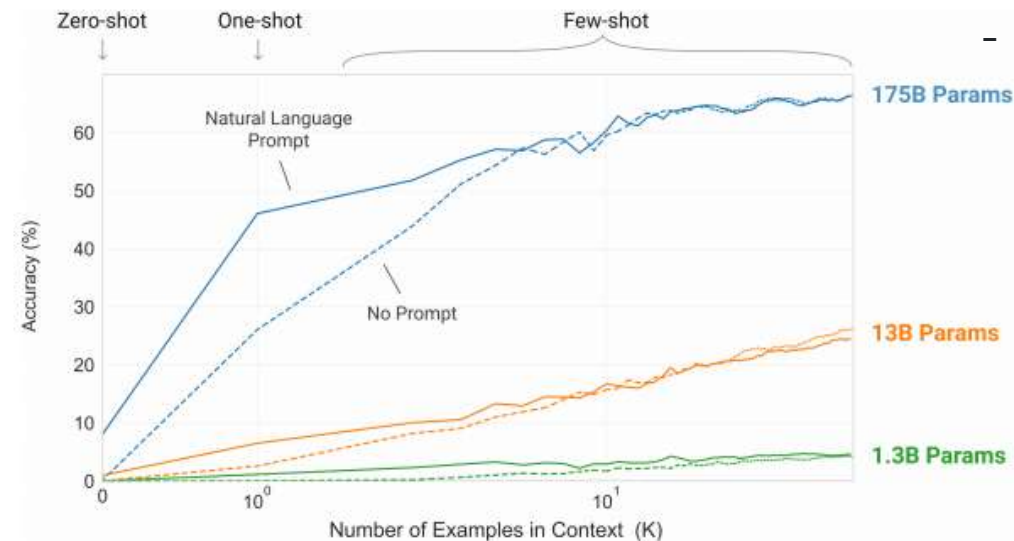
GPT-3

강화학습

GPT-3

- Few-Shot learning

(fine-tuning 없이도 task 수행 가능)



GPT-3.5

- 입력에 맞는 대답 도출
(인간이 방향성 제시)
- 파라미터 줄임

ChatGPT

(GPT-3.5의 Fine-Tuned 버전)

RL

너는 GPT-3 기반으로 fine-tuning된거니?



네, 저는 OpenAI에서 개발한 GPT-3 기반으로 fine-tuning된 모델입니다. 저는 고정된 큰 말
뭉치를 통해 훈련되었으며, 이후 특정 작업을 위해 fine-tuning되었습니다. 저는 사용자의
입력에 기반하여 텍스트 생성 및 답변을 제공하는 데 사용됩니다.

GPT

1. 큰 말뭉치에서 대용량의 언어모델을 학습

2. 분류 데이터를 써서

특정 과제에 맞춰 모델을 미세조정

Semi-supervised Learning(준지도 학습)

Transformers(트랜스포머)

Language Models(언어 모델)

: 단어가 주어지면 시퀀스에서 다음 단어를 예측할 수 있는 확률적 모델

Generative Models(생성 모델)

: 학습 후 기존 데이터와 유사한 새로운 데이터 생성

Multitask Learning(다중 작업 학습)

: 주어진 입력에 대해 다른 작업 해결

zero-shot(제로샷): 훈련 데이터 없이 학습

one-shot(원샷) : 단 한 개의 데이터로 학습

few-shot learning(퓨샷러닝): 소량의 데이터로 학습

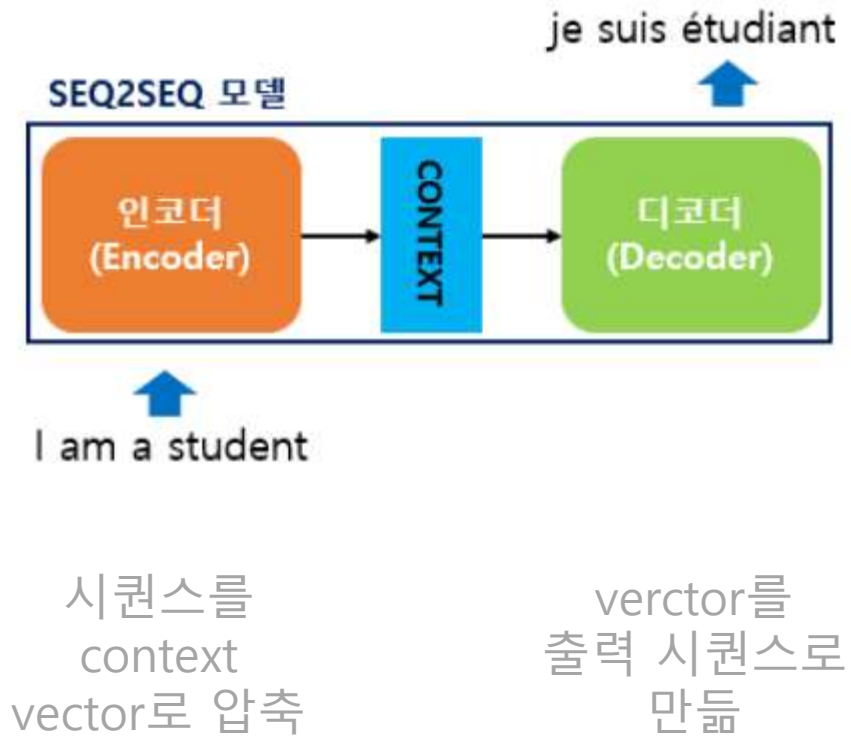
[GPT-3]

Language Models are Few-Shot Learners

[Transformer]

Attention Is All You Need

seq2seq 모델

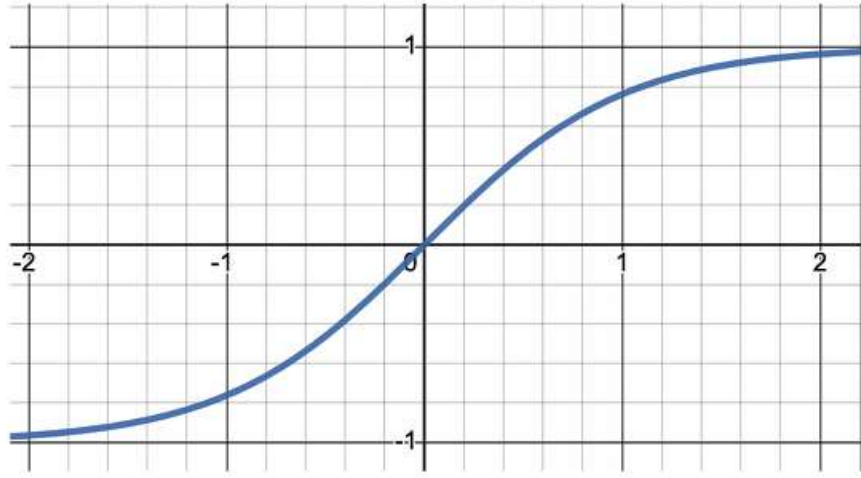


1. 정보 손실 발생

2. RNN의 기울기 소실 문제

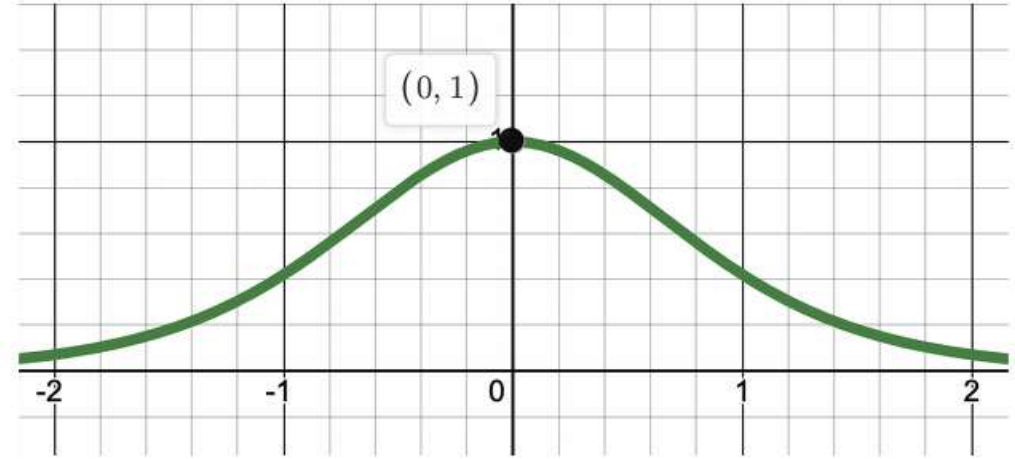
layer가 많아질수록 Gradient 값이 매우 작아지는 현상

긴 입력 시퀀스를 하나의 고정된 크기의 벡터로 압축하면서 **정보손실 발생**

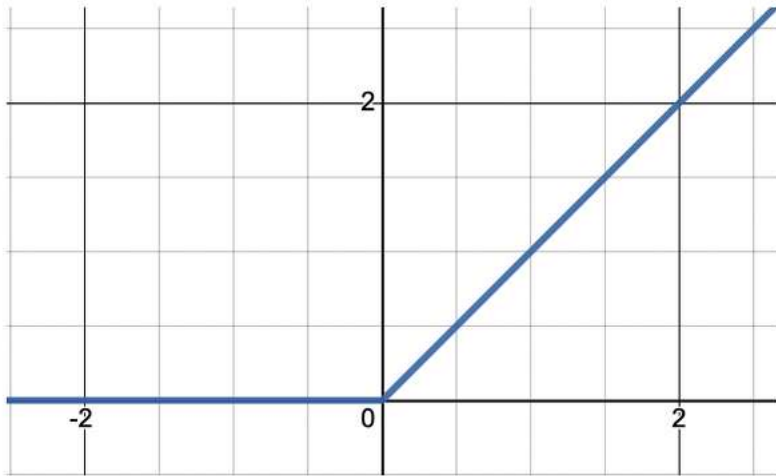


\tanh

<https://heytech.tistory.com/>

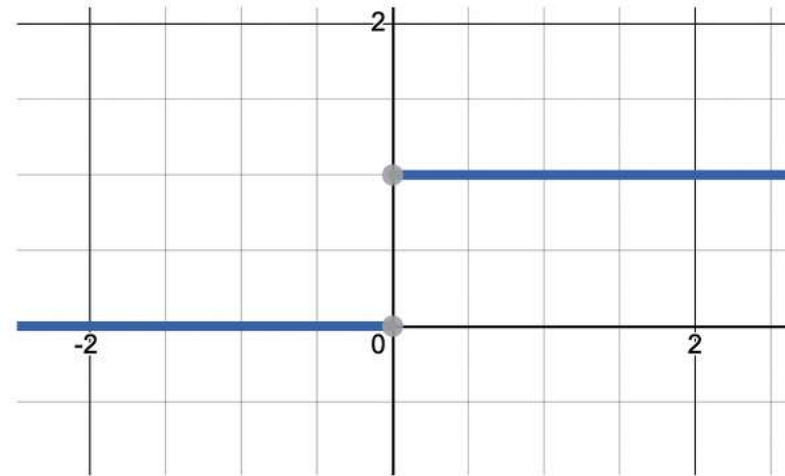


$\frac{d}{dx} \tanh$



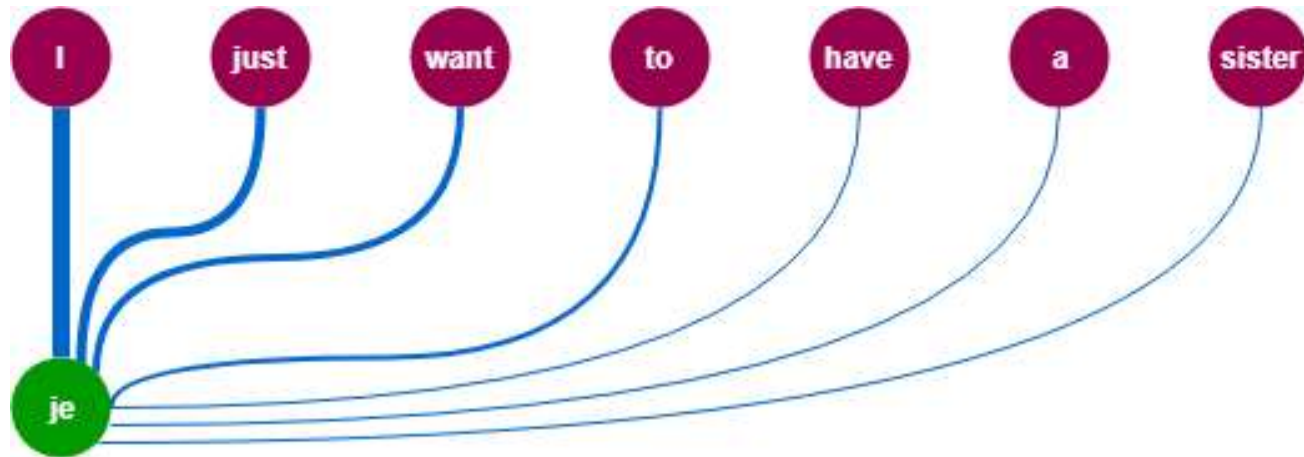
$ReLU$

<https://heytech.tistory.com/>



$\frac{d}{dx} ReLU$

Attention Mechanism

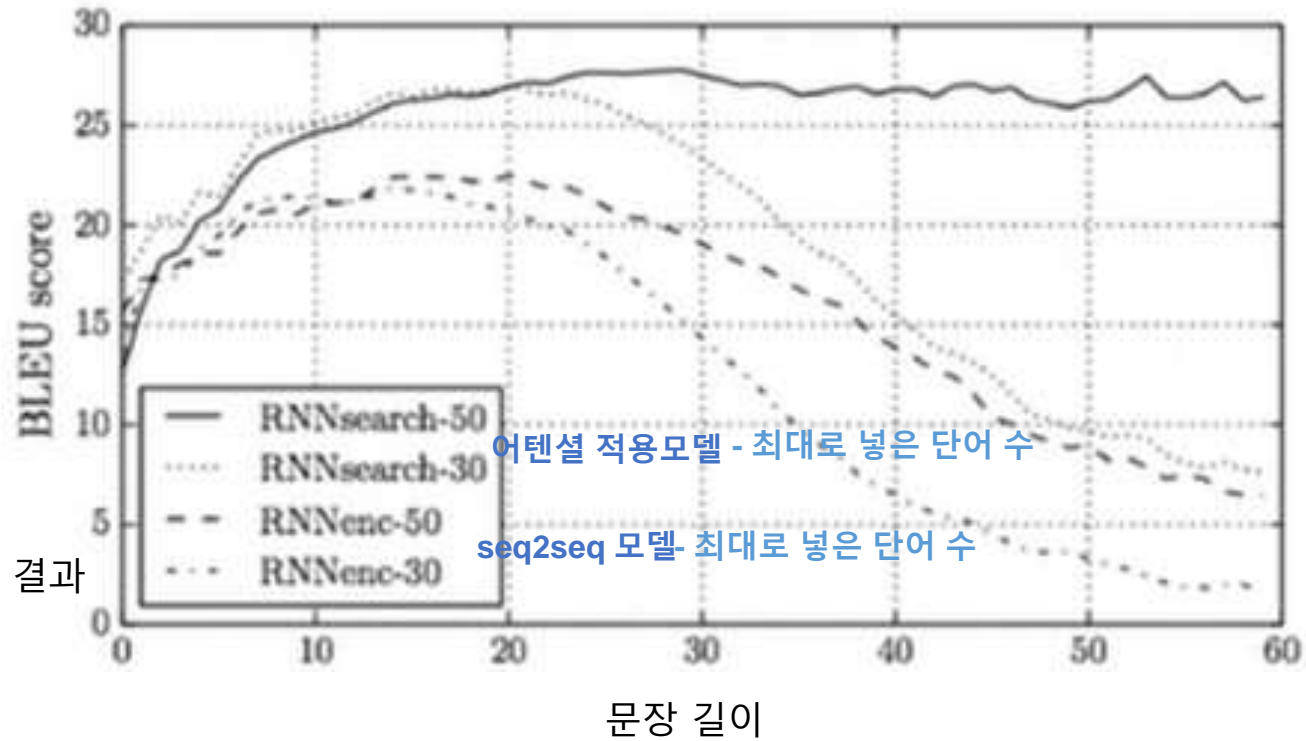


매 시점(time step)마다 인코더에서의
전체 **입력 문장**을 다시 한 번 참고한다

-> 예측할 단어와 연관이 큰 단어에 **더 집중**

이전 모델보다 더 다양하고 방대한 정보 처리 가능

["Neural Machine Translation by Jointly Learning to Align and Translate" \(2014\)](#)



어텐셜 적용모델 - 최대로 넣은 단어 수

seq2seq 모델 - 최대로 넣은 단어 수

BLEU score
: 기계 번역 결과와 사람이 번역한 결과
의
유사도를 비교하여
번역에 대한 성능 측정 방법

1. 병렬화 문제로 인해 대규모 학습이 불가능함
2. long distance dependency(장거리 의존성) 문제

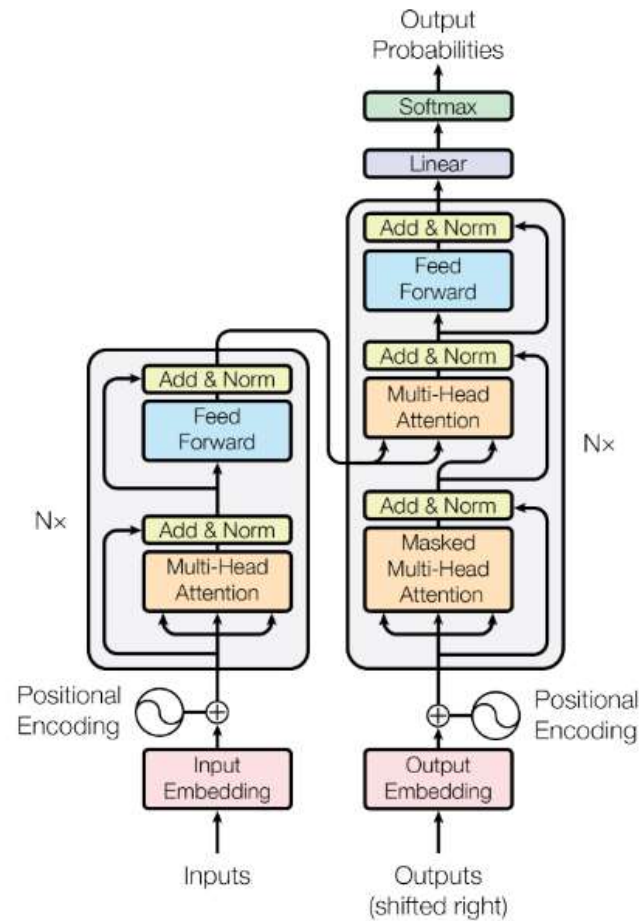
seq2seq 모델 + Attention - RNN 구조

=

Transformer

Transformer

인코더
소스 언어의 시퀀스 압축



디코더
타겟 언어의 시퀀스 생성

Figure 1: The Transformer - model architecture.

GPT

오늘 나는 치킨을 먹었다

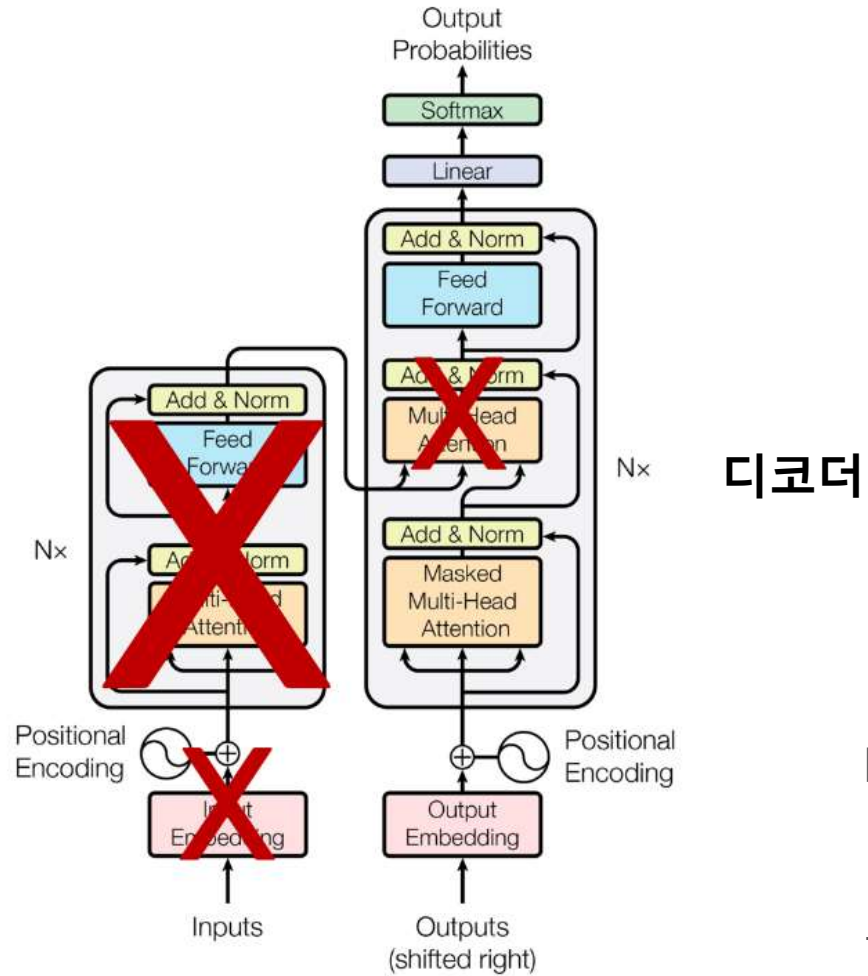


입력

오늘 나는
예측



mask



디코더

Bert

입력의 모두를 참조할 수 있기 때문에
문장의 의미 추출에 강점

GPT

Generative Pre-trained Transformer

(decoder만을 사용)

입력

transformer

출력

데이터 토큰화 후

임베딩 작업

고차원 벡터로 변환

attention 메커니즘을 사용한
seq2seq 양방향 상호작용

디코딩

단어간 유사도 판별

문맥과 관련된
토큰 파악

다음 단어 예측

word embedding

워드 임베딩

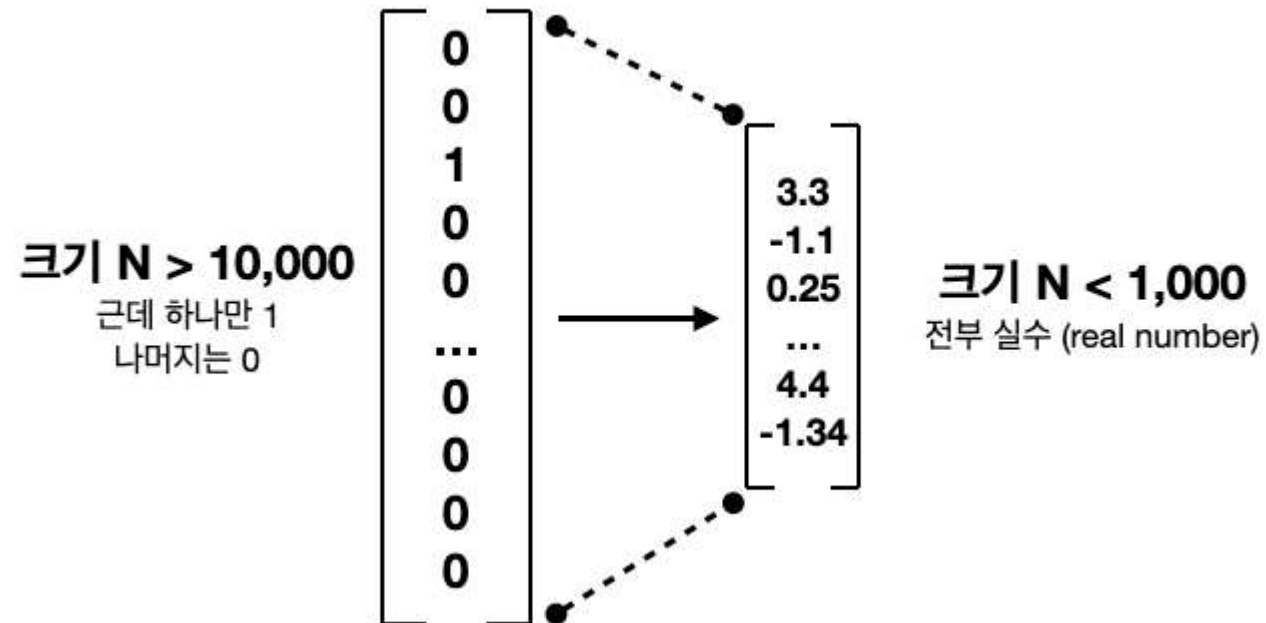
: 단어를 벡터로 표현하는 방법

강아지 = [0 0 0 0 1 0 0 0 0 0 0 ... 중략 ... 0] 희소 표현

차원은 10,000, 0은 9,999
(공간낭비, 유사도 반영 X)

강아지 = [0.2 1.8 1.1 -2.1 1.1 2.8 ... 중략 ... 0.2] 밀집 표현

사용자가 설정한 값이 벡터의 차원



얕은 신경망을 사용하여 단어 임베딩을 학습하는 방법



Google의 **word2vec** 모델

Stanford의 **GloVe** 모델

word2vec

주변 단어를 통해 중심단어를 추측하는 방법

- 신경망 기반 방법(활성화함수x)
- 단어의 의미와 맥락을 포착하는데 효과적

한국 - 서울 + 파리 = 프랑스

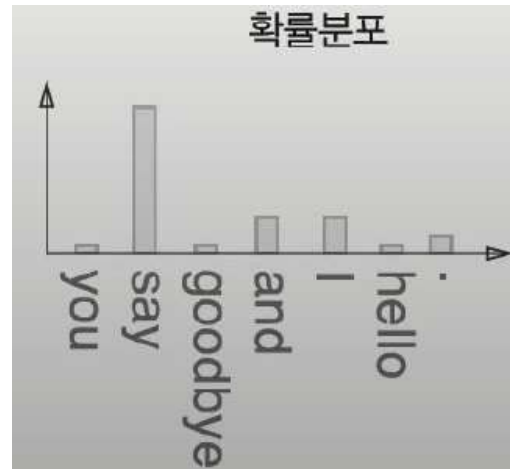
어머니 - 아버지 + 여자 = 남자

CBOW

you [?] goodbye and I say hello.

Skip-gram

[?] say [?] and I say hello.



말뭉치 후보군 중에서 하나를 예측

- CBOW

중심 단어 주변 단어

↓ ↓

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

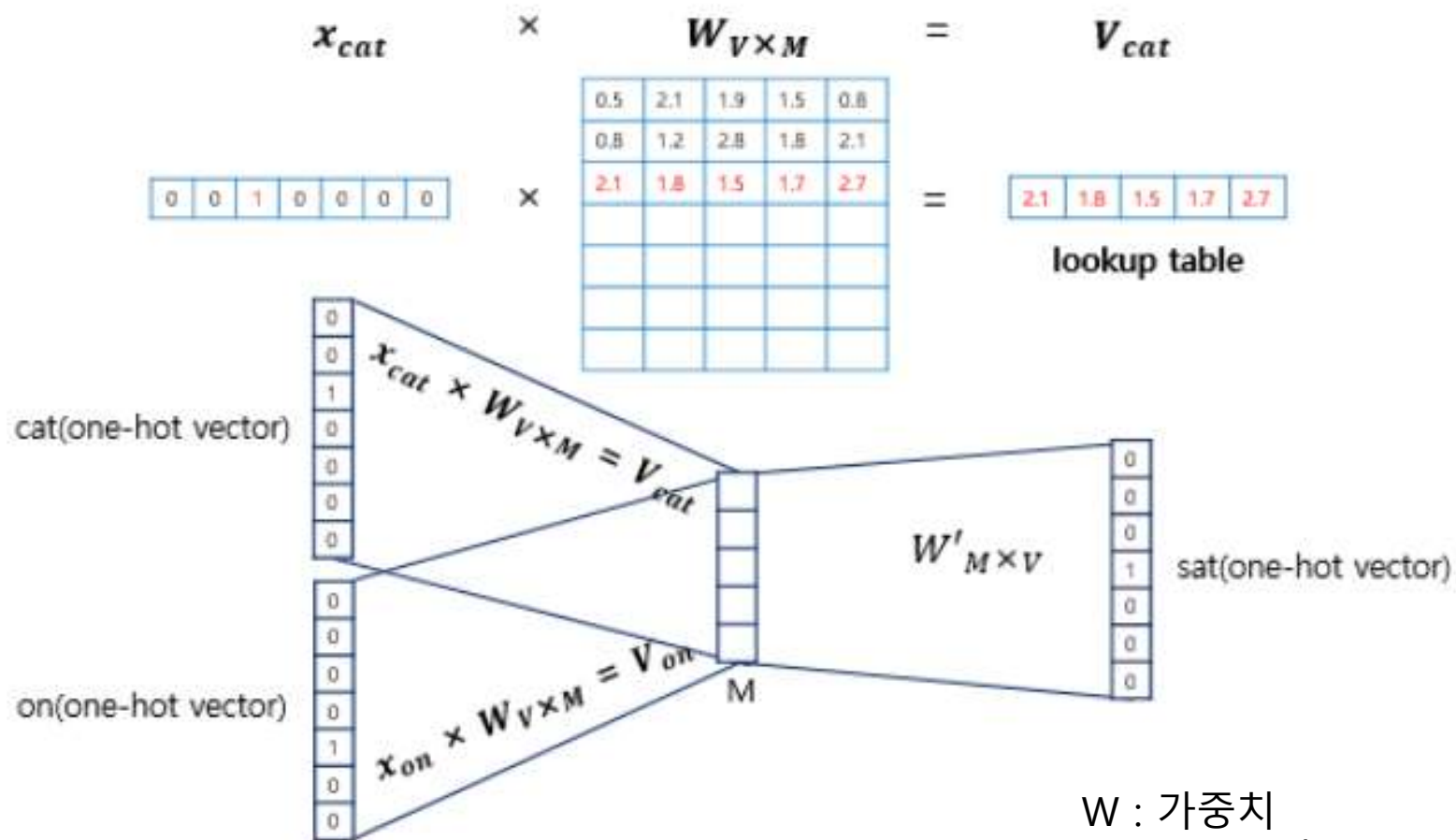
The fat cat sat on the mat

The fat cat sat on the mat

중심 단어	주변 단어
[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]

The fat cat sat on the mat

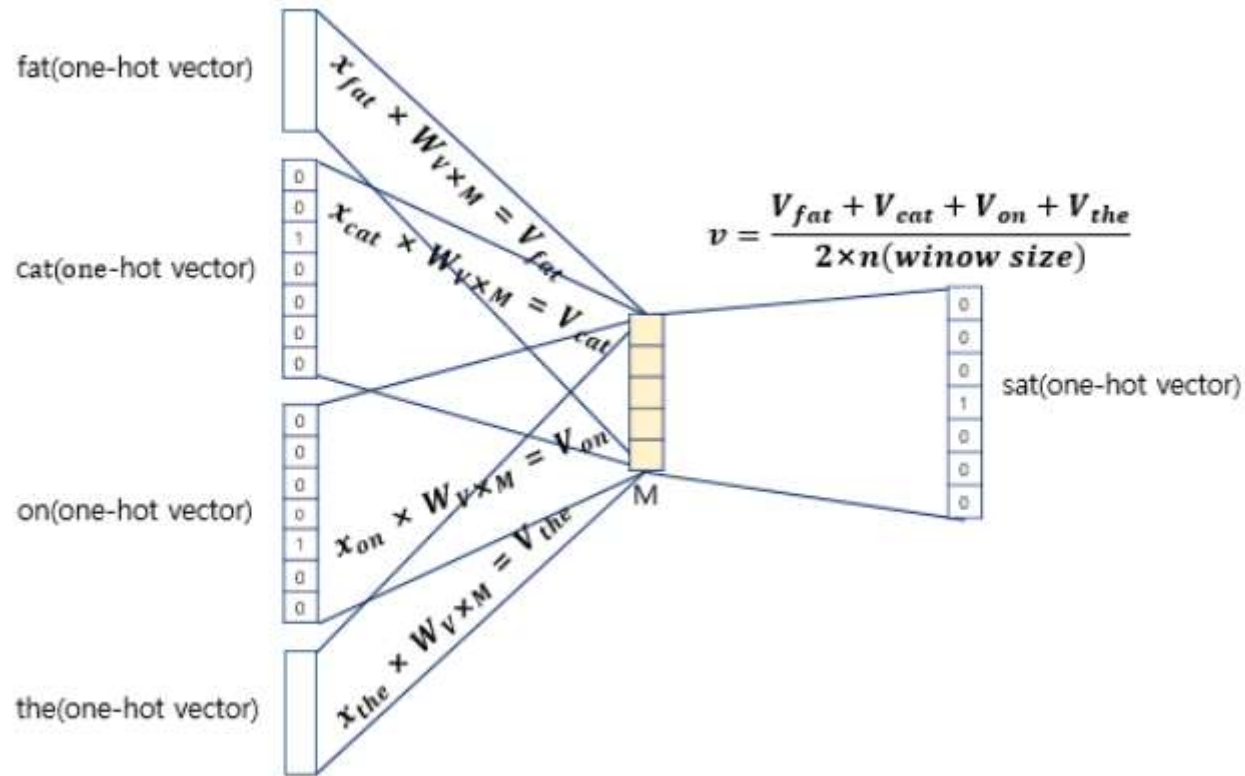
[0, 0, 0, 1, 0, 0, 0]



W : 가중치

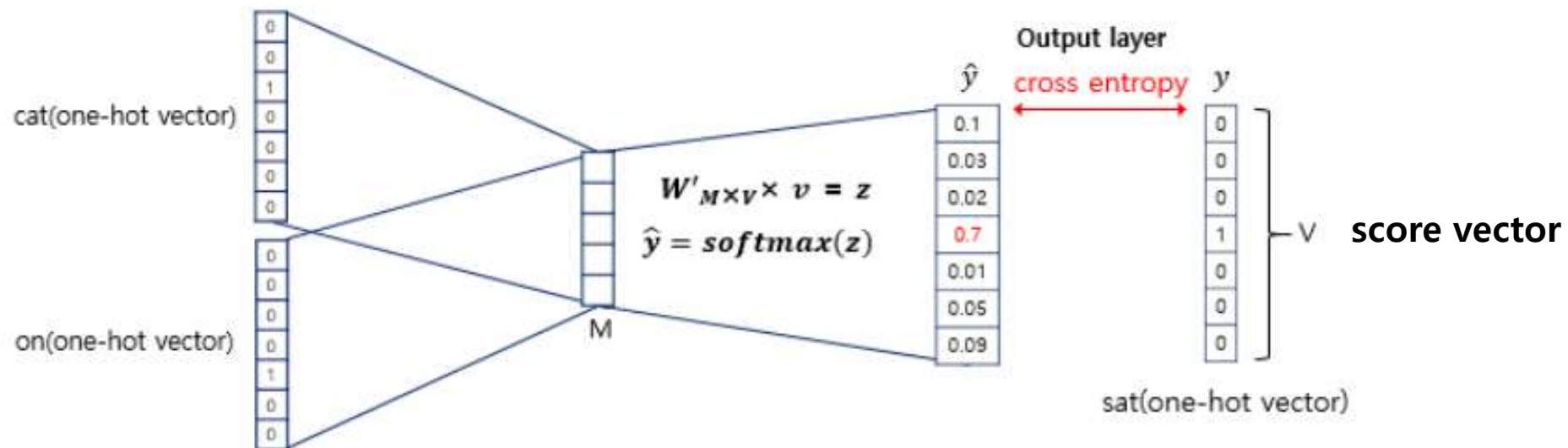
V : 단어 집합의 크기

M : 투사층의 크기(임베딩 한 후의 벡터의 차원)



모델의

예측값(스코어 벡터) \hat{y} 과 실젯값(중심 단어) y 의 차이를 최소화할 수 있는 **최적의 가중치 행렬**을 찾는 것



- Skip-gram

중심 단어 주변 단어

The fat **cat** sat on the mat

The fat cat sat on the mat

중심 단어	주변 단어
cat	The
cat	Fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the

