워드 임베딩

23.05.18 유하영

통계적 언어 모델

대규모의 텍스트 데이터를 학습하여
단어 간의 확률 관계를 파악

단어 간의 관계, 문장 구조, 문법 규칙 등을 학습

'예측'

주어진 문맥에서 다음 단어가 무엇인지
추론

나는

학교에

밥을

공부를

통계적 언어 모델

대규모의 텍스트 데이터를 학습하여
단어 간의 확률 관계를 파악

단어 간의 관계, 문장 구조, 문법 규칙 등을 학습

'예측'

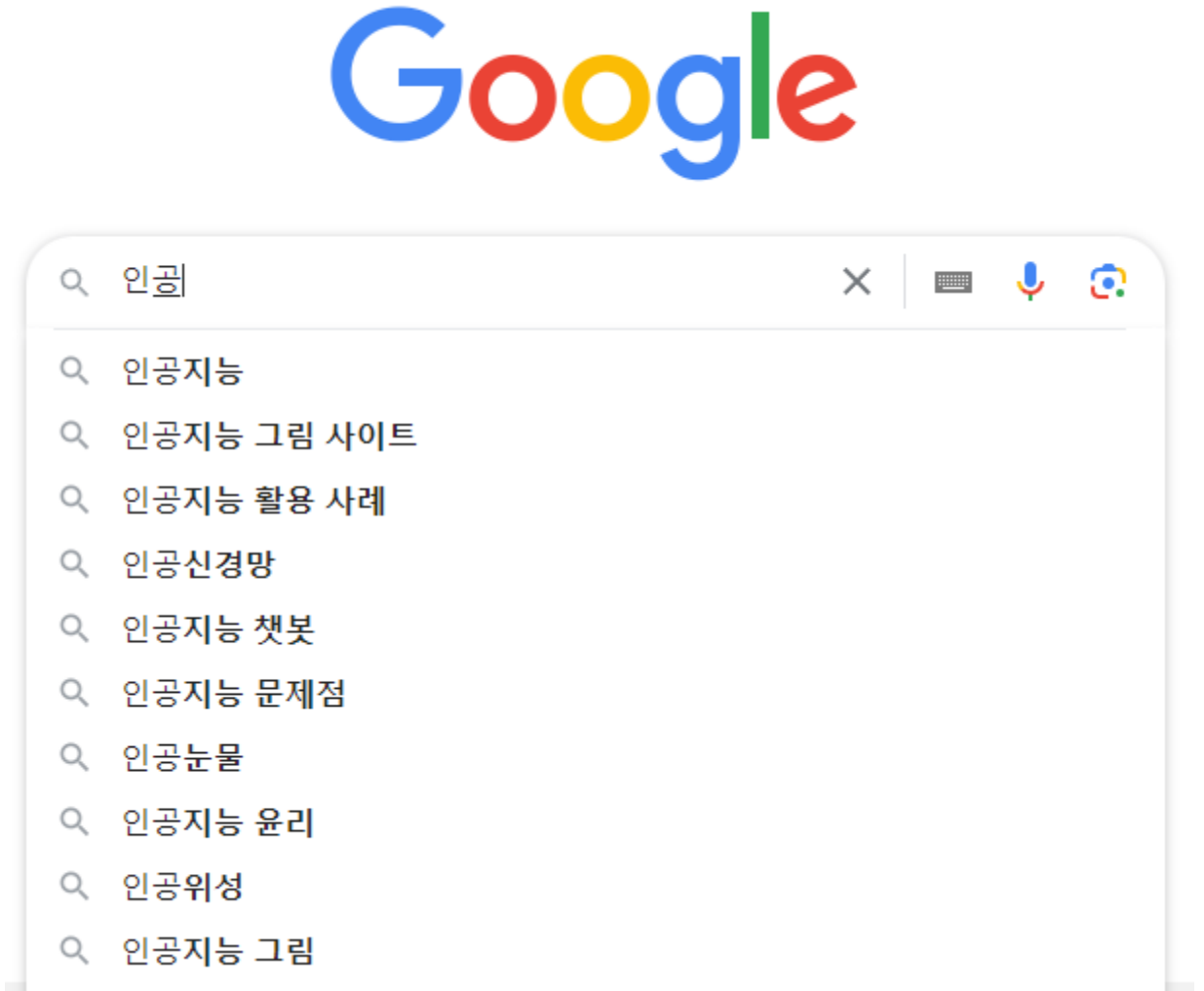
주어진 문맥에서 다음 단어가 무엇인지
추론

나는 학교에 왔다

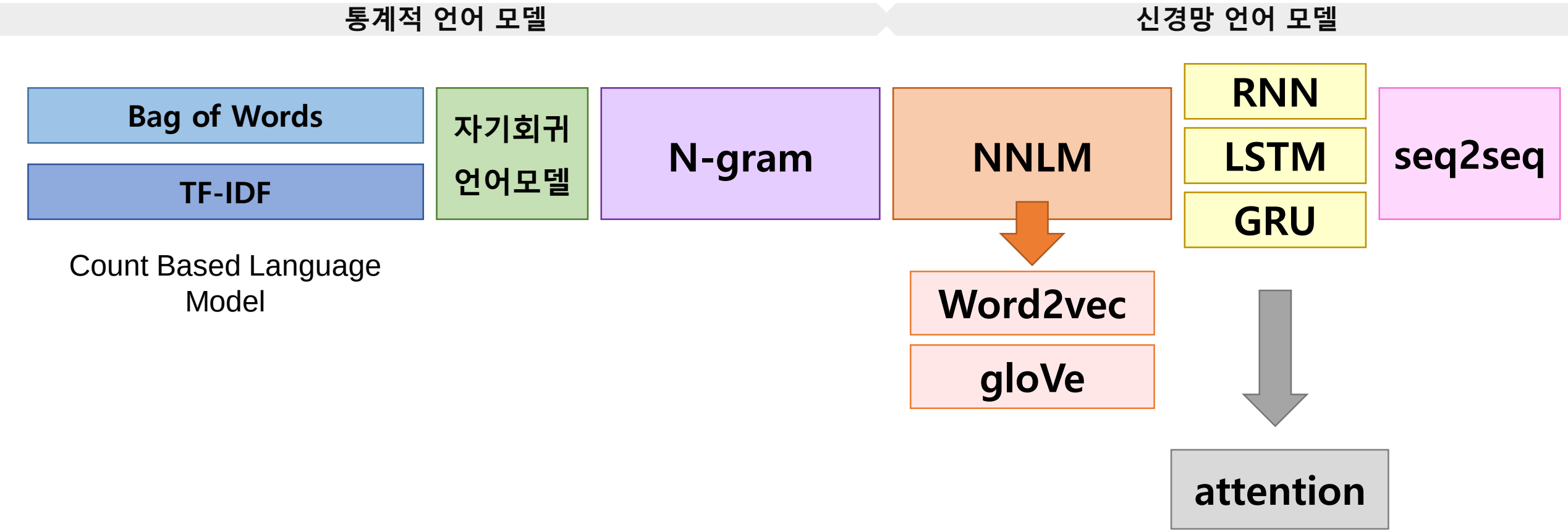
한국의 수도는 **서울이다.**

한국의 수도는 **파리이다.**

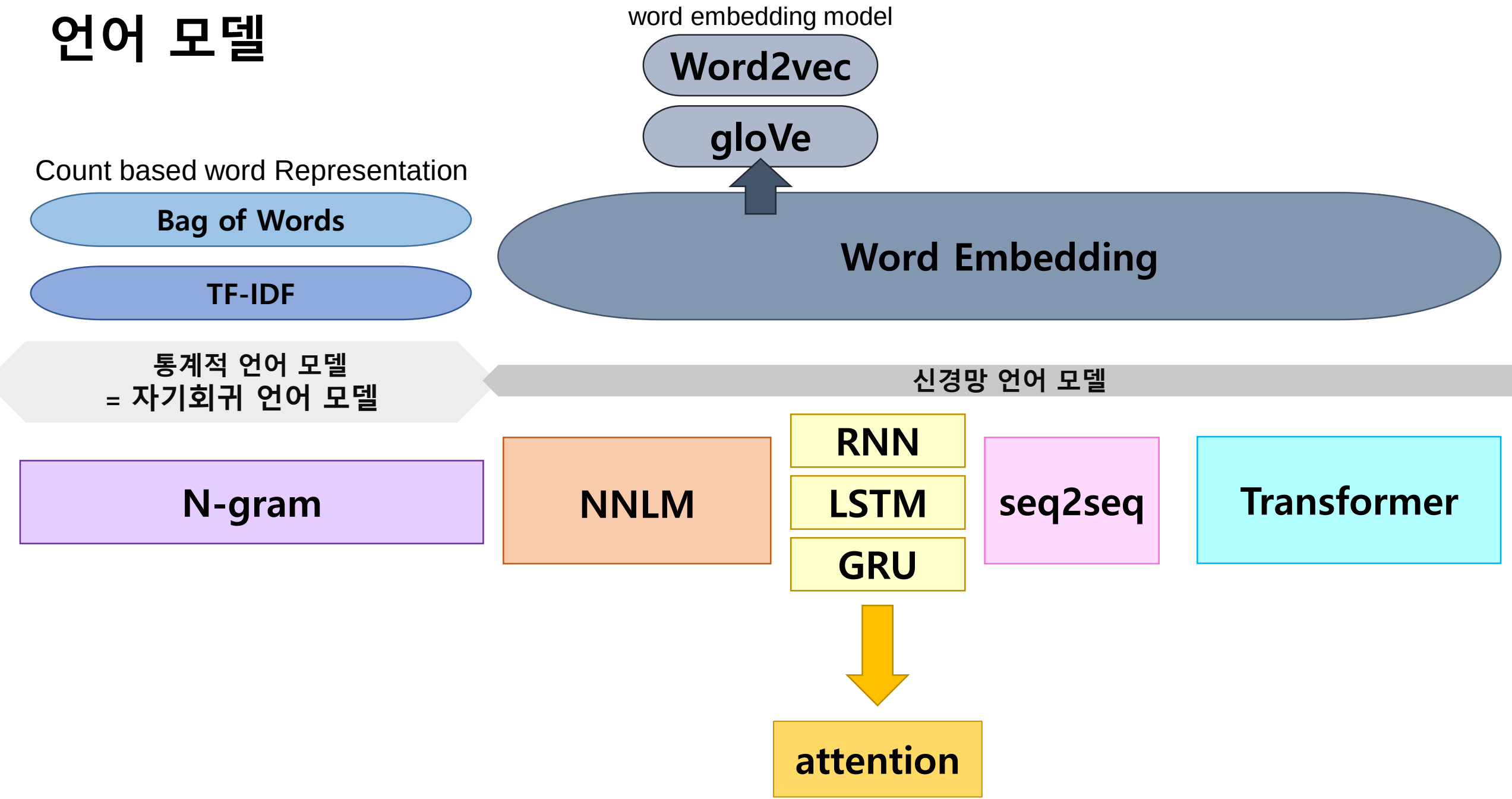
통계적 언어 모델


Google
인공
인공지능
인공지능 그림 사이트
인공지능 활용 사례
인공신경망
인공지능 챗봇
인공지능 문제점
인공눈물
인공지능 윤리
인공위성
인공지능 그림

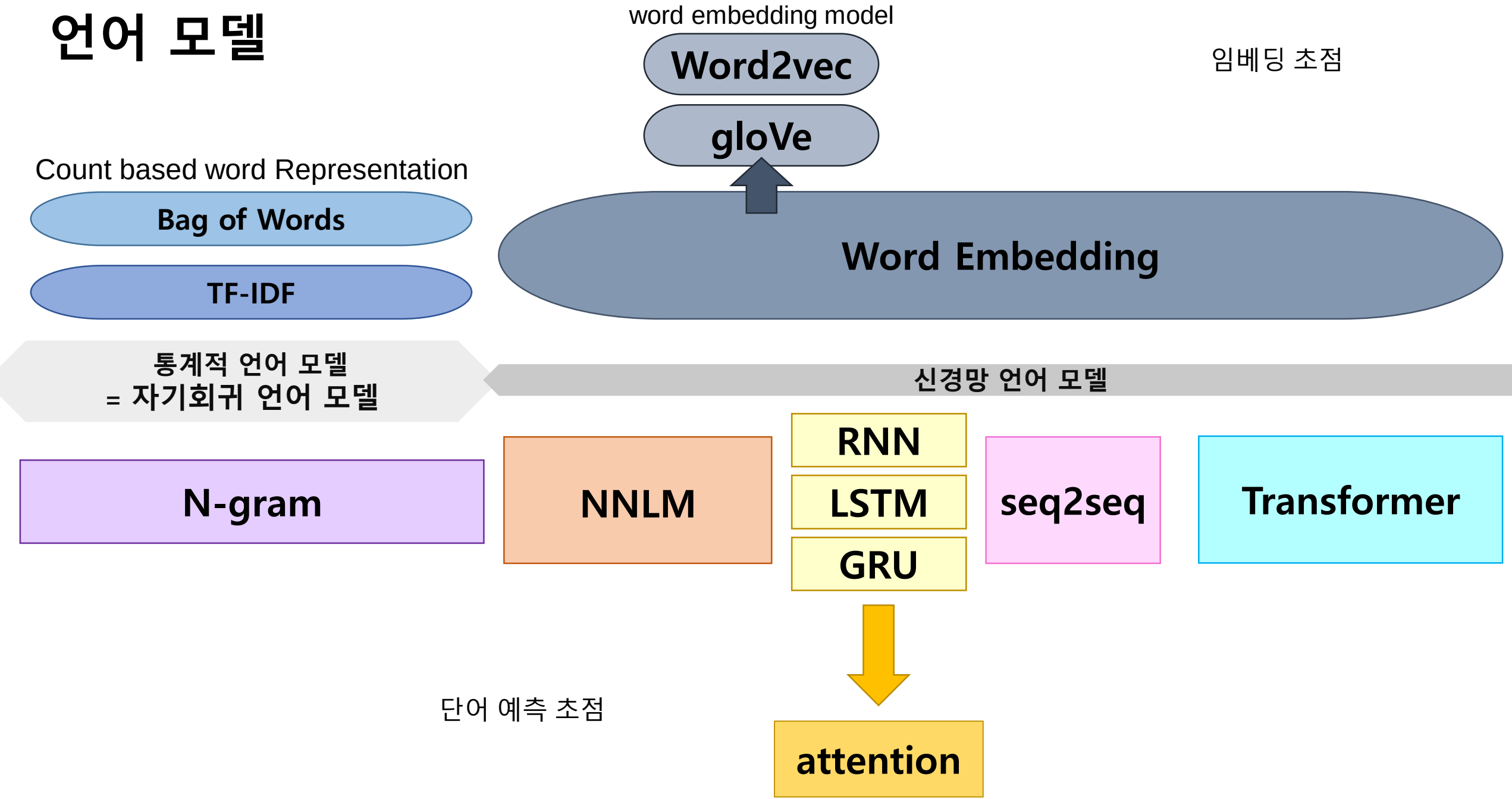
언어 모델


통계적 언어 모델
신경망 언어 모델
Bag of Words
TF-IDF
Count Based Language Model
자기회귀 언어모델
N-gram
NNLM
Word2vec
gloVe
RNN
LSTM
GRU
seq2seq
attention

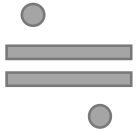
언어 모델


word embedding model
Word2vec
gloVe
Count based word Representation
Bag of Words
TF-IDF
Word Embedding
통계적 언어 모델
= 자기회귀 언어 모델
신경망 언어 모델
N-gram
NNLM
RNN
LSTM
GRU
seq2seq
Transformer
attention

언어 모델


word embedding model
Word2vec
gloVe
임베딩 초점
Count based word Representation
Bag of Words
TF-IDF
Word Embedding
통계적 언어 모델
= 자기회귀 언어 모델
신경망 언어 모델
N-gram
NNLM
RNN
LSTM
GRU
seq2seq
Transformer
단어 예측 초점
attention

통계적 언어 모델



자기회귀 LM

문장이나 문서의 확률을 추정

이전에 생성된 단어들을 참고하여
다음 단어를 예측하는 언어 모델

= 통계적인 방식으로 문장의 확률을 모델링

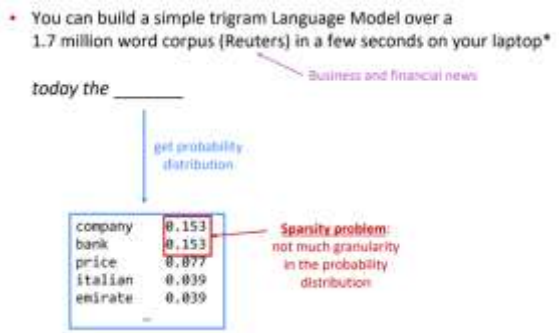
모든 언어 모델이 자기회귀인건 아니다
BERT

자기회귀 LM

$$P(w_1, w_2, w_3, w_4, w_5, \dots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \dots, w_{n-1})$$

$$P(x_1, x_2, x_3 \dots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_n|x_1 \dots x_{n-1})$$

확률을 차례로 곱해나감



N-gram LM

연속된 일부 단어만 고려
~~n : 일부 단어를 몇 개 보느냐?~~

확률을 계산하는 것이 아닌
n-1개의 단어에 영향을 받아 COUNT

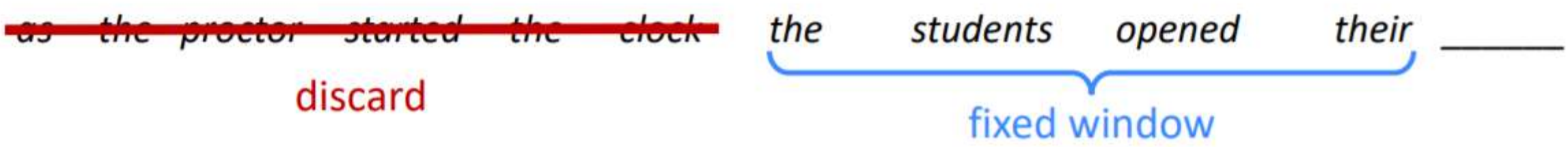
$$P(\boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)}, \dots, \boldsymbol{x}^{(1)}) = \frac{\text{count}(\boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(t)}, \dots, \boldsymbol{x}^{(t-n+2)})}{\text{count}(\boldsymbol{x}^{(t)}, \dots, \boldsymbol{x}^{(t-n+2)})}$$

희소 문제

단어 간 유사도

NNLM(Neural Network LM)

정해진 n개의 단어만을 참고하여 다음 단어를 예측



- input : 단어들의 시퀀스
- output : 다음 단어에 대한 확률 분포

NNLM(Neural Network LM)

Input

단어 시퀀스를 벡터 형태로 표현

피드 포워드 신경망

선형변환
+
활성화 함수

~~what~~ will the fat cat sit … (window size = 4)

sit = [0, 0, 0, 0, 1, 0]

Input layer

피드 포워드 신경망

Projection layer

단어 임베딩

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

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

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

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

x_{will}

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

$\times$

$W_{V \times M}$

2.1	1.8	1.5	1.7	2.7
0.1	0.8	1.3	2.7	1.1
		.		
		.		
		.		

$=$

e_{will}

2.1	1.8	1.5	1.7	2.7

희소 표현 → 밀집 표현

~~what~~ will the fat cat sit …

(window size(N) = 4)

sit = [0, 0, 0, 0, 1, 0]

Input layer

피드 포워드 신경망

Projection layer

단어 벡터 표현

Hidden layer

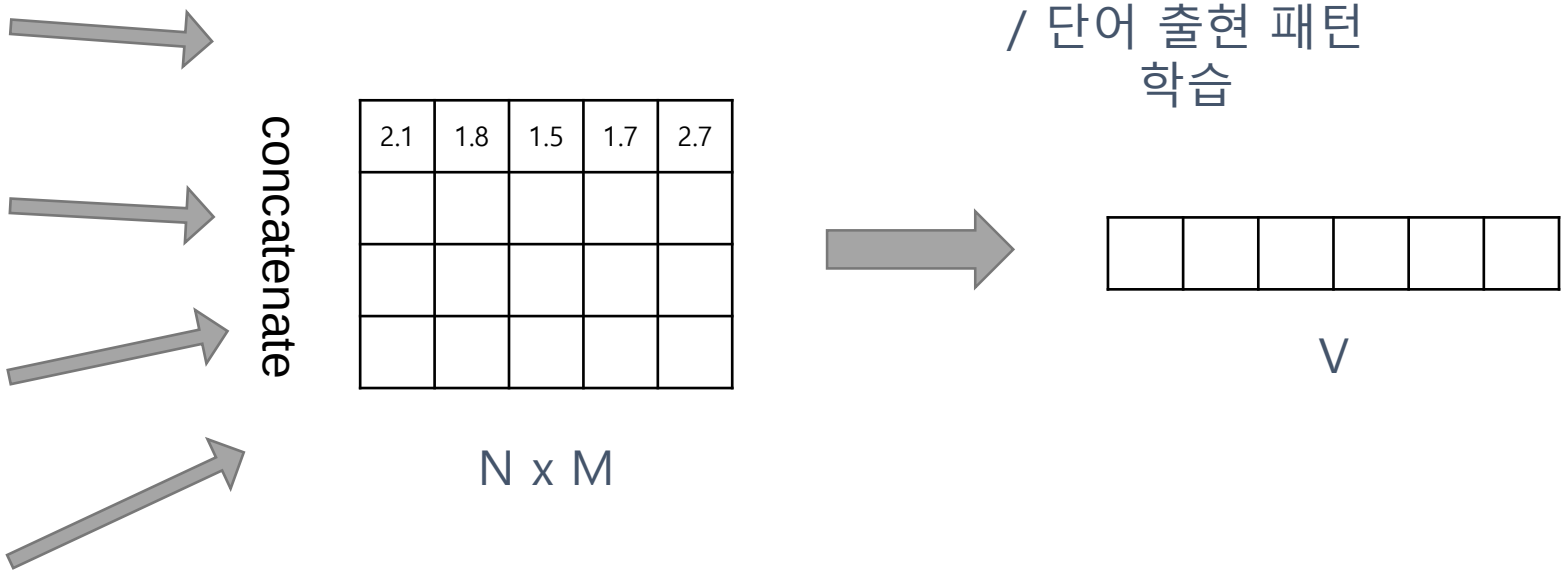
다음 단어를 예측하는 역할 / 단어 출현 패턴 학습

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

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

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

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



신경망 언어 모델

~~what~~ **will the fat cat sit** … (window size(N) = 4)

sit = [0, 0, 0, 0, 1, 0]

Input layer

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

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

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

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

피드 포워드 신경망

Projection layer

단어 벡터 표현

가중치 업데이트 how?
-> 역전파 사용

concatenate

2.1	1.8	1.5	1.7	2.7

N x M

윈도우크기X투사층

- 차원축소
- 밀집 표현(의미

Hidden layer

다음 단어를 예측하는 역할 / 단어 출현 패턴 학습

은닉층에서 또 다른 가중치와 곱해지고 편향이 더해지면

입력이었던 원-핫 벡터들과 동일하게 V차원의 벡터를 얻는다.

- 비선형성을 추가(표현 능력 향상시킴)

설명

~~what~~ will the fat cat sit … (window size = 4)

sit = [0, 0, 0, 0, 0, 1, 0]

Input layer

Projection layer

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

피드 포워드 신경망

단어 임베딩

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

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

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

x_{will}

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

$W_{V \times M}$

0.1	0.8	1.3	2.7	1.1
2.1	1.8	1.5	1.7	2.7
		.		
		.		
		.		

e_{will}

2.1	1.8	1.5	1.7	2.7

룩업테이블 해서 5차원으로 표현

초기에는 랜덤값
학습과정 중에는 값이 계속 변경됨

희소 표현 → 밀집 표현

설명

~~what~~ **will the fat cat sit** … (window size = 4)

sit = [0, 0, 0, 0, 0, 1, 0]

단어를 고차원의 희소 표현에서 저차원의 밀집 벡터 표현으로 변환

Input layer

Projection layer

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

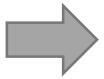
the [0, 0, 1, 0, 0, 0, 0]

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

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

인코딩은 각 단어를 고유한 인덱스로 표현

단어 임베딩 학습
+ 단어의 의미를 저차원의 벡터로 표현(모델 성능 향상)
단어 간의 의미적 관계를 잡아내고 유사한 단어들끼리 가까이 모여있도록 구성



M 투사층의 크기-> 사용자가 임의로 정하기

보통 어휘에 비해 상대적으로 작게 설정
어휘가 10000이라면 투사층의 크기는 몇십개~몇백개로 설정
brcause 투사층의 작은 차원으로 단어의 특징을 압축하여 다음 계층인 은닉층으로 전달하여 계산 비용을 줄이고 모델의 일반화 능력을 향상시키기 위함
(즉, 투사층의 크기는 모델의 복잡성과 표현 능력에 영향을 줌)

설명

~~what~~ will the fat cat sit …

(window size = 4)

sit = [0, 0, 0, 0, 0, 1, 0]

단어를 고차원의 희소 표현에서 저차원의 밀집 벡터 표현으로 변환

Input layer

Projection layer

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

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

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

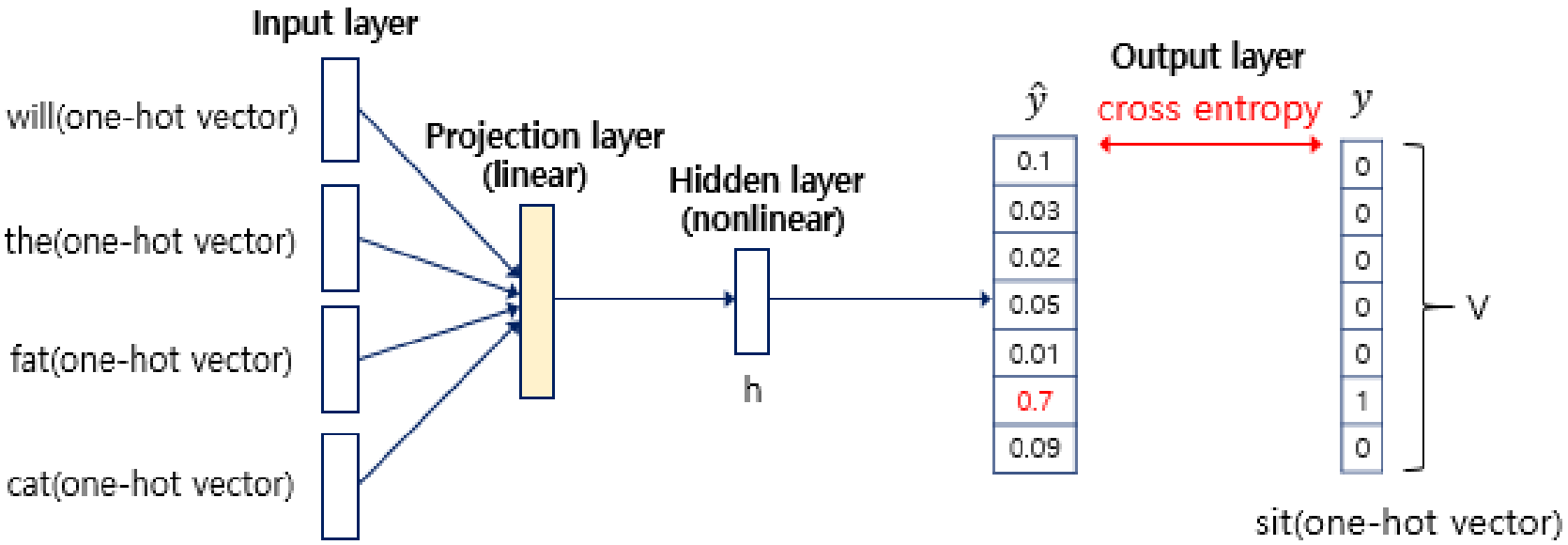
cat [0, 0, 0, 0, 1, 0, 0]

차원 조정(고->저)
단어 의미 수학적으로 계산 가능한 형태로 만들기

희소 표현

밀집 표현

신경망 언어 모델


Input layer
will(one-hot vector)
the(one-hot vector)
fat(one-hot vector)
cat(one-hot vector)
Projection layer
(linear)
Hidden layer
(nonlinear)
h
Output layer
$\hat{y}$
cross entropy
y
0.1
0.03
0.02
0.05
0.01
0.7
0.09
0
0
0
0
0
1
0
V
sit(one-hot vector)

신경망 언어 모델

하영이가 밥을 아주 맛있고 배부르게 먹다

하지만, !!!!!!!!!!!!!!!!!
학습 데이터에 없는 단어는 예측 ㄴㄴㄴㄴㄴㄴㄴㄴ
양 학습 데이터 필요하긴 함.

단어의 표현이 학습 데이터에서 학습되기 때문
아님 사전에 학습된 단어 임베딩 사용하던가~~

배부르게 먹다
라는 코퍼스가 있기 때문에 다른 모델로도 저렇게 예측할 수 있음

기존의 모델은 '먹다'란 단어만 한다면 '먹다'를 예측하지 못함.

하지만 NNLM은 학습 데이터에 먹다 라는 단어만 있으면
배부르게 뒤에 먹다 라는 단어를 예측할 수 있음
왜냐하면 단어간의 의미적 유사성을 벡터로 다 표현해놓았기 때문
에
예측가능함

RNNLM

- input : 단어들의 시퀀스
- output : 다음 단어에 대한 확률 분포

word2vec

언어모델의 일종이기는 함
그냥 단어를 임베딩 하는 모델로 단어 간의 의미적 유사성을 학습하는 역할에 더 초점(단어 간의 관계 이해)

예측이 주 목적이라면 다른 언어 모델 사용고려해야

N-gram은 이러한 빈도 기반의 통계적 접근 방식으로 작동하며, 이를 위해 학습 데이터가 필요합니다. 학습 데이터에 등장하는 N-gram의 빈도를 기반으로 모델이 구축되기 때문에 학습 데이터가 없으면 N-gram 모델을 사용할 수 없습니다.

word embedding

NNLM은 신경망 기반의 언어 모델로, 이전에 학습된 데이터가 없어도 학습이 가능

워드 임베딩

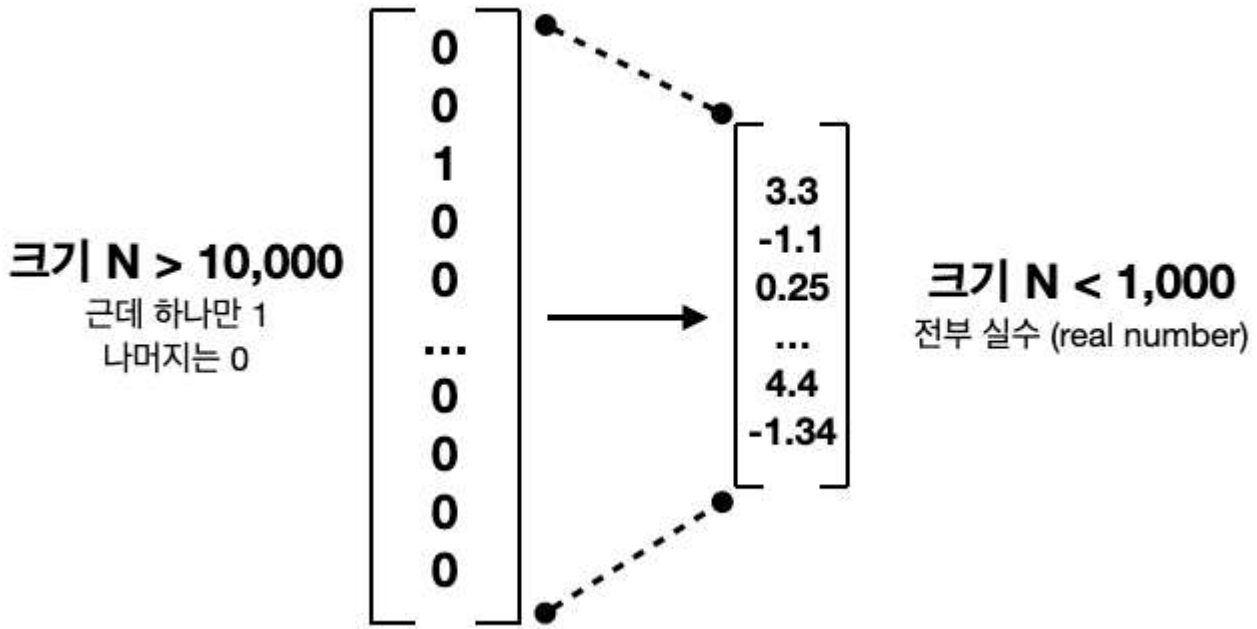
: 단어를 벡터로 표현하는 방법

강아지 = [0 0 0 0 1 0 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

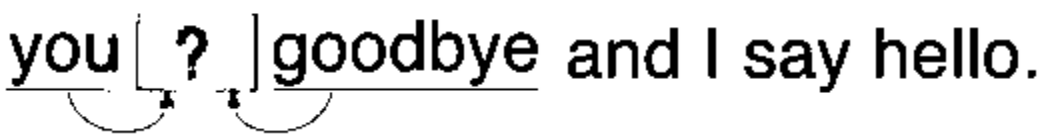
주변 단어를 통해 중심단어를 추측하는 방법

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

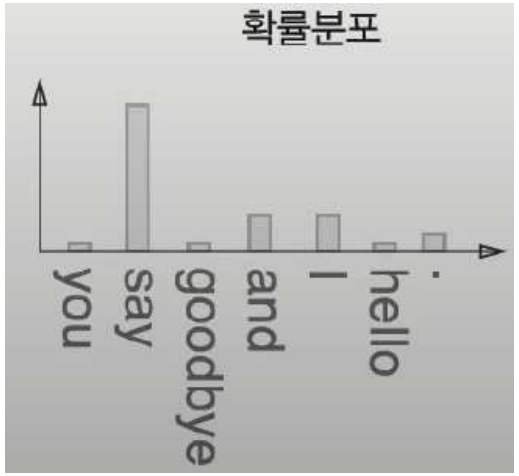
한국 - 서울 + 파리 = 프랑스

어머니 - 아버지 + 여자 = 남자

CBOW



Skip-gram



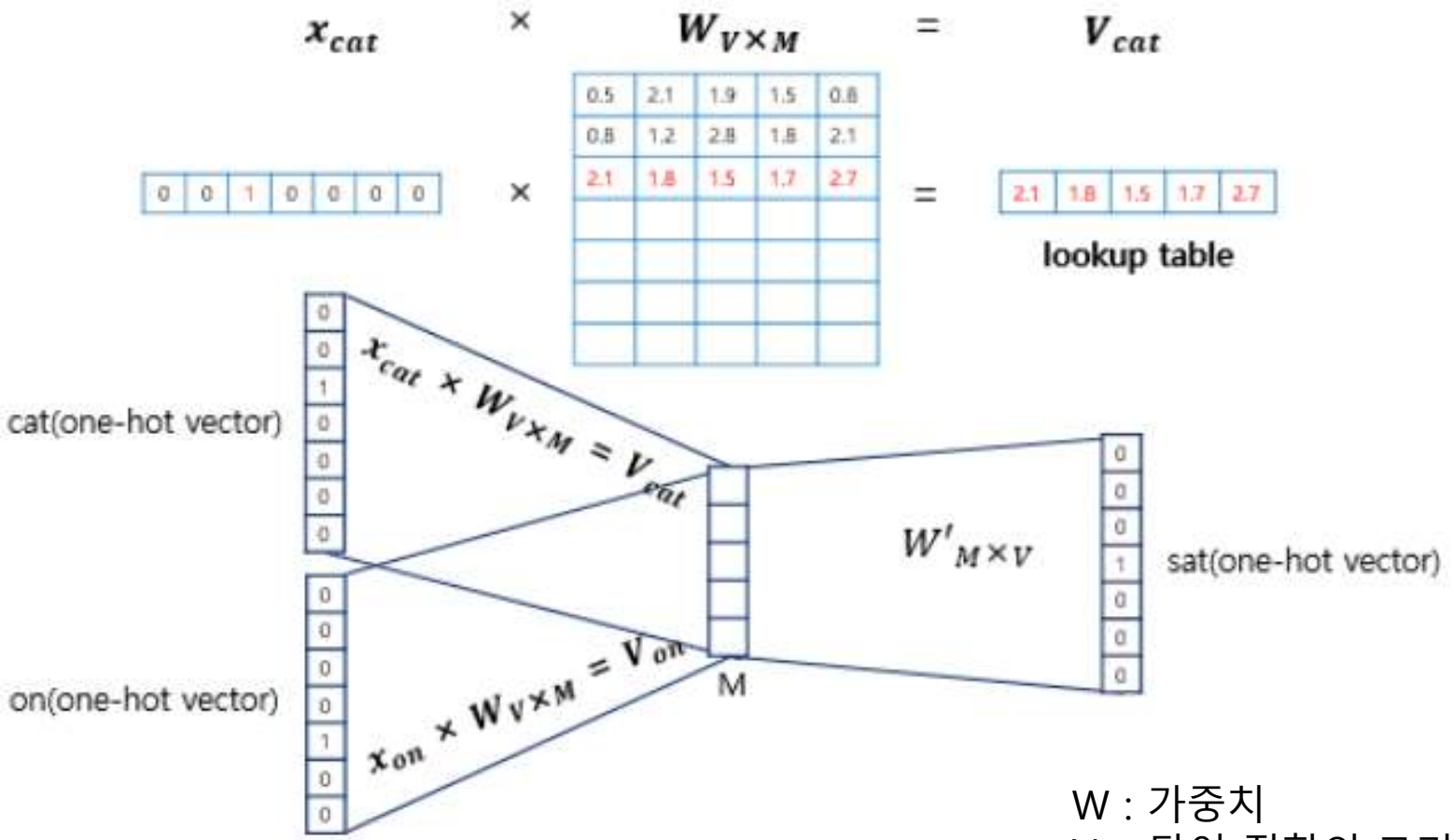
말뭉치 후보군 중에서 하나를 예측

- CBOW

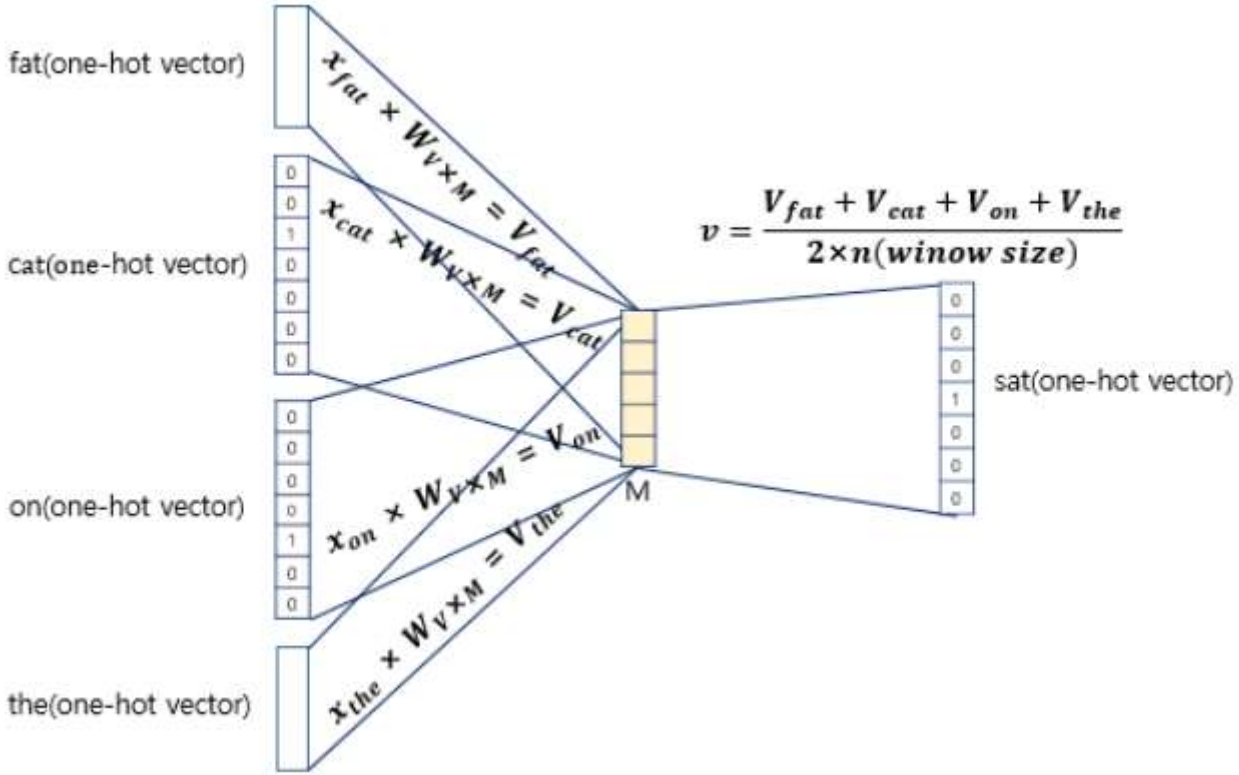
중심 단어	주변 단어
[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 1, 0, 0,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



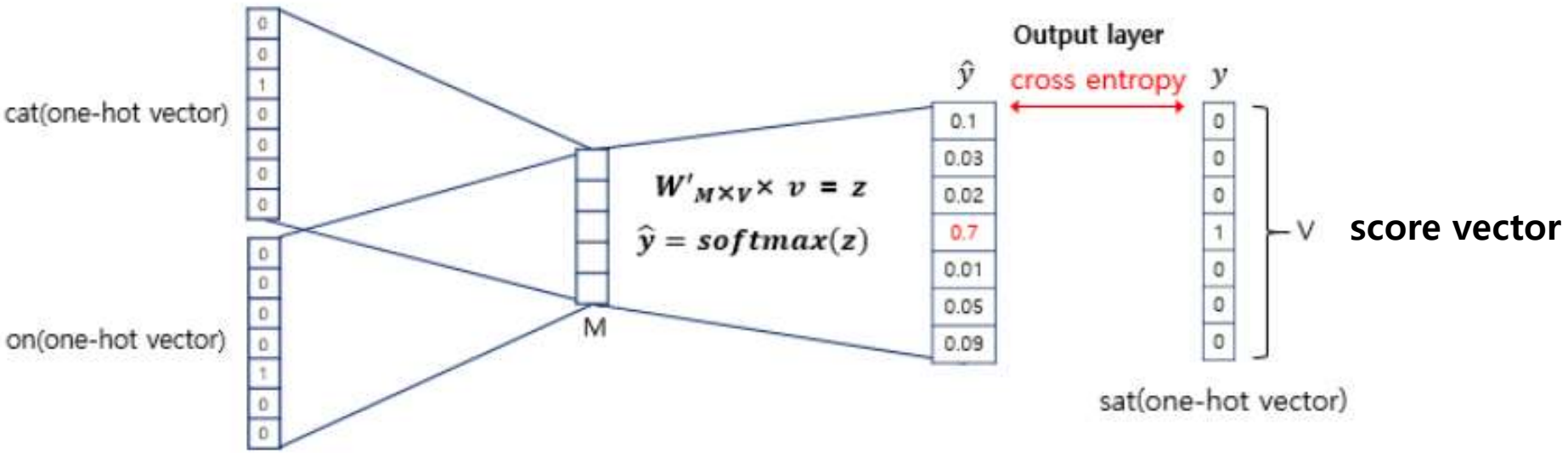
W : 가중치
V : 단어 집합의 크기
M : 투사층의 크기(임베딩 한 후의 벡터의 차원)



모델의

예측값(스코어 벡터) $\hat{y}$과 실젯값(중심 단어) y의 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