SGNS

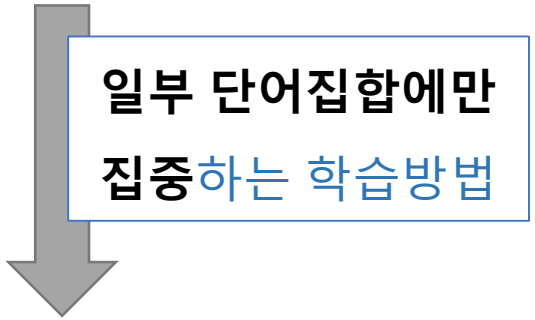
&

RNN LM

23.06.29 유하영

Word2Vec

단어 집합의 크기가 클 경우
모든 단어의 임베딩 벡터값을 업데이트 하는 것은
비효율적



SGNS

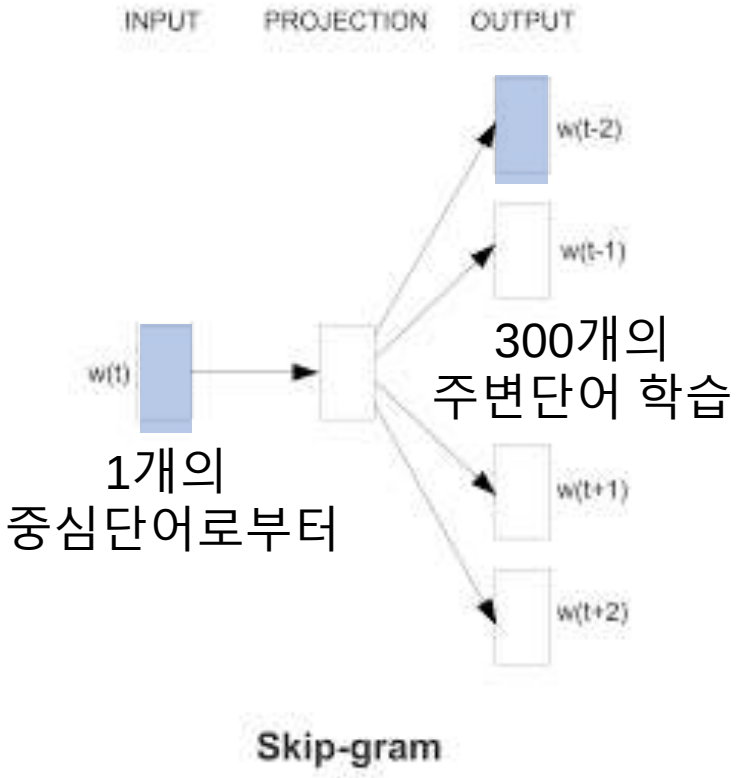
skip-gram negative sampling

skip-gram

모든 단어

총 **10,000**개의 단어로 이루어진 문서.

300개의 임베딩 벡터 학습



10,000 x 300 = **3,000,000번의 학습 필요**

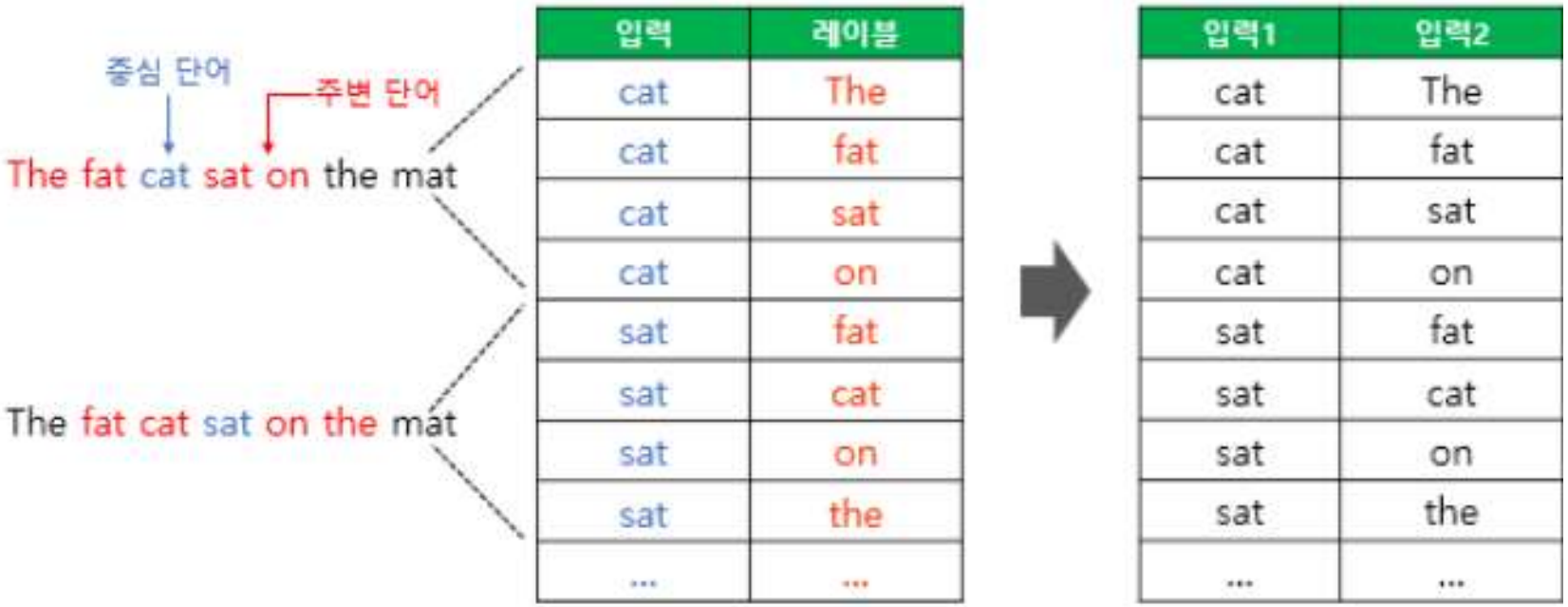
Negative sampling skip-gram

하나의 문서에서 **k+1**개의 단어에 대해서만

300개의 임베딩 벡터 학습

(k+1) x 300

1) SGNS에서는 Skip-Gram과 달리 (**중심단어**, **주변단어 or 주변단어가 아닌 단어**)**의 '쌍'**을 이용한다.

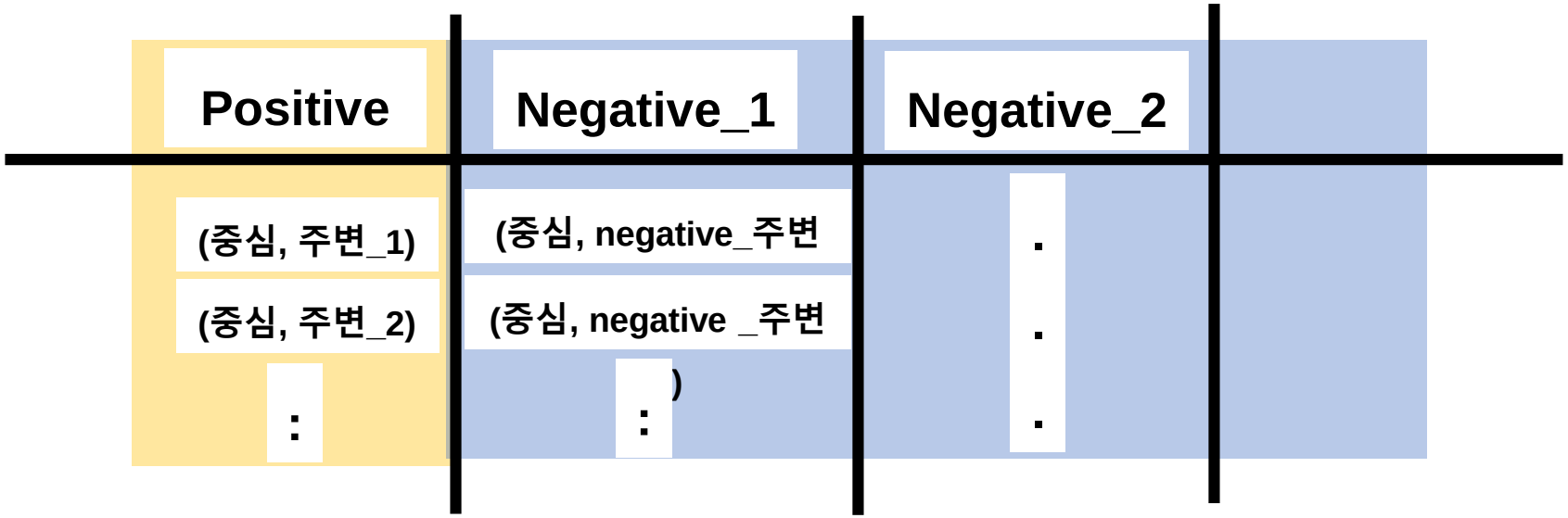


입력	레이블
cat	The
cat	fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the
...	...

입력1	입력2
cat	The
cat	fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the
...	...

2) (중심단어-주변단어)에 대해서는 positive를,

(중심단어-주변단어가 아닌 단어)에 대해서는 negative를 취해준다.

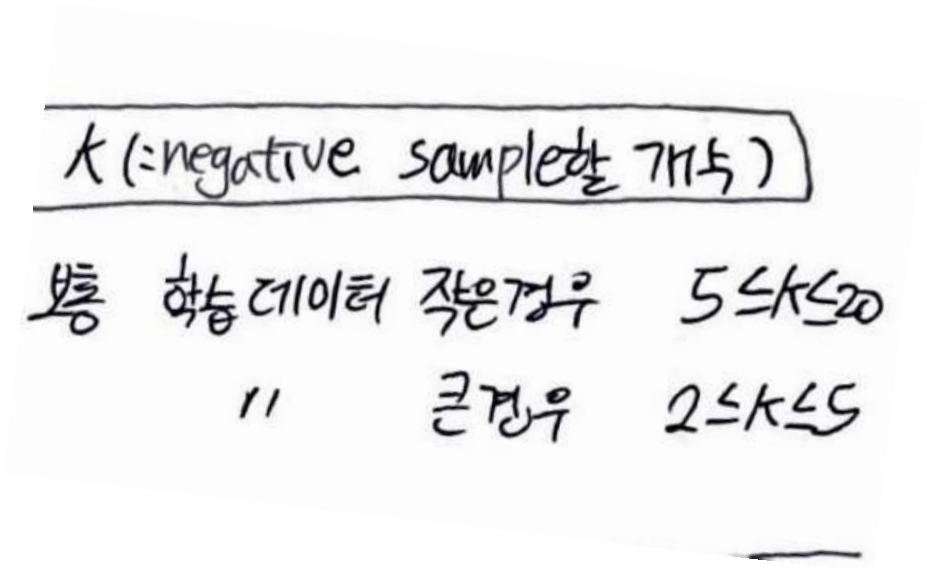


1 k

(Positive sample 1개 ⊕ Negative sample K개) → K+1 개의 단어만 보게됨!

전체 단어 10,000 개 대신

Negative sampling할 단어의 개수 선정 기준



학습 데이터의 양에 따라
negative sampling할 개수(k)를 정해준다.

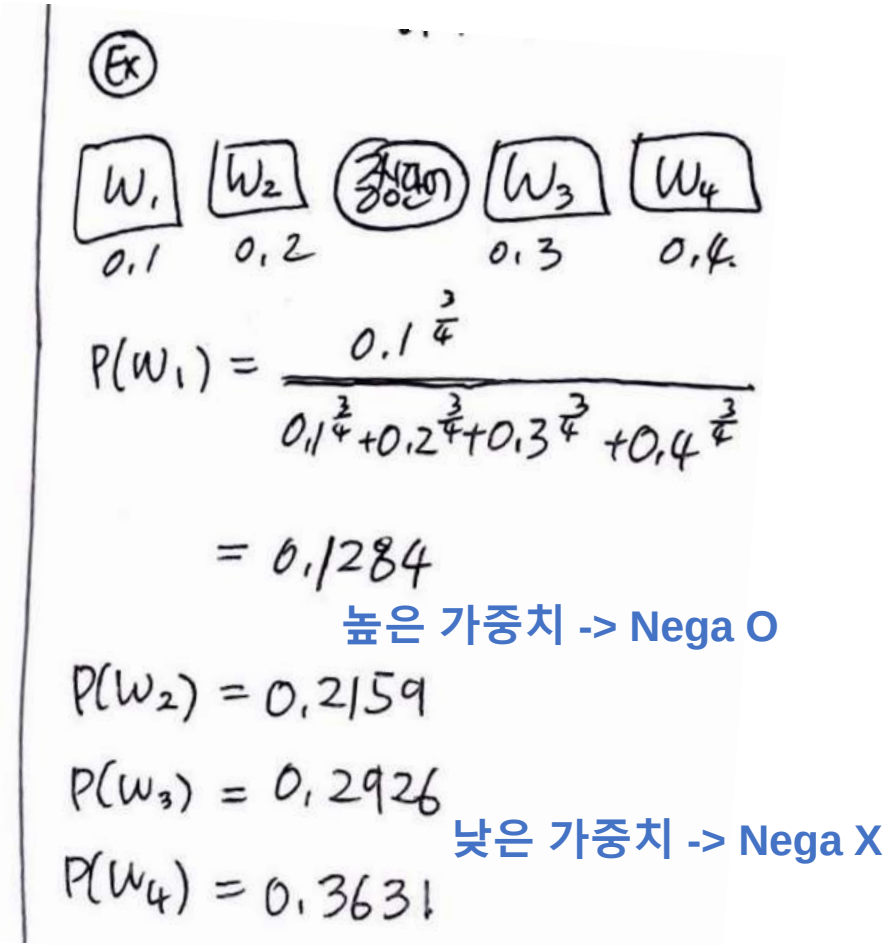
Negative sampling 단어의 선정 기준

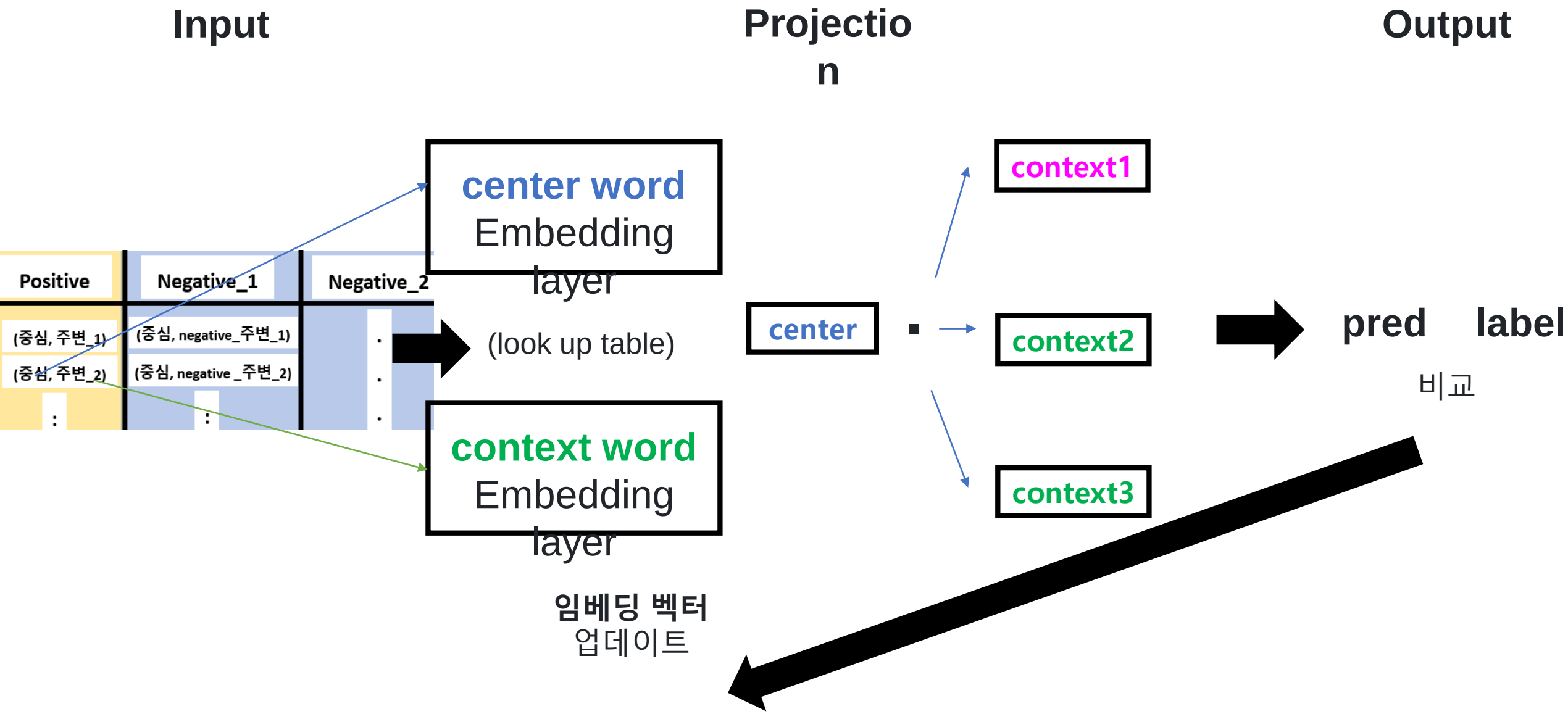
등장빈도가 낮은 단어에 **높은 확률**을 부여하여,
이 단어들이 negative sampling으로 뽑힐 수 있게끔 한다.

$$P(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{i=0}^{n} (f(w_i)^{\frac{3}{4}})}$$

($f(w_i)$은 (3/4 보정)

(주변)단어의 출현 빈도
출현 확률이 낮은 단어를
버리지 않기 위해




Input
Projection
Output
Positive
Negative_1
Negative_2
(중심, 주변_1)
(중심, 주변_2)
(중심, negative_주변_1)
(중심, negative _주변_2)
center word
Embedding
layer
(look up table)
context word
Embedding
layer
center
context1
context2
context3
pred
label
비교
임베딩 벡터
업데이트

(총 10개 단어)

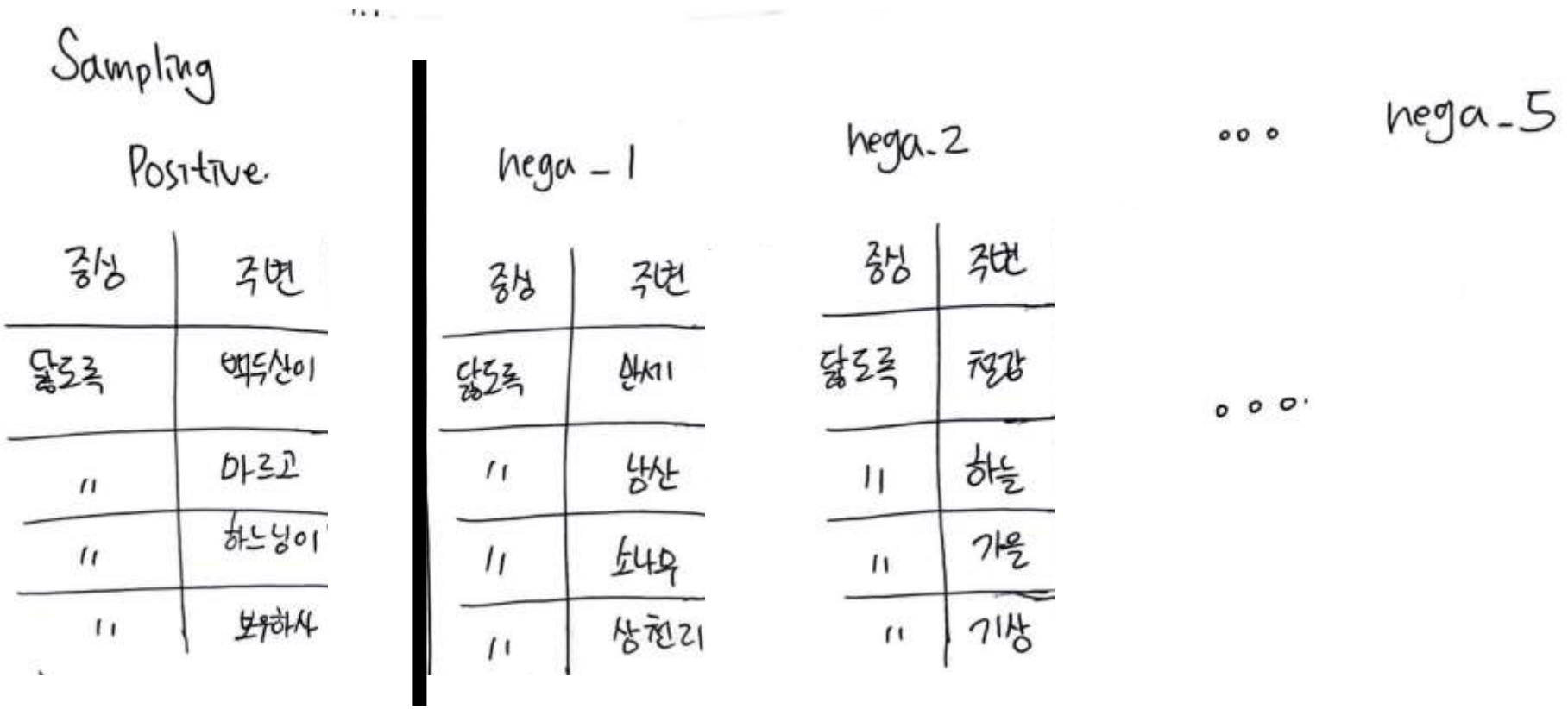
동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세 무궁화 삼천리

negative positive 중심 positive negative

window size = 2

k (negative sample 개수) = 5

Input

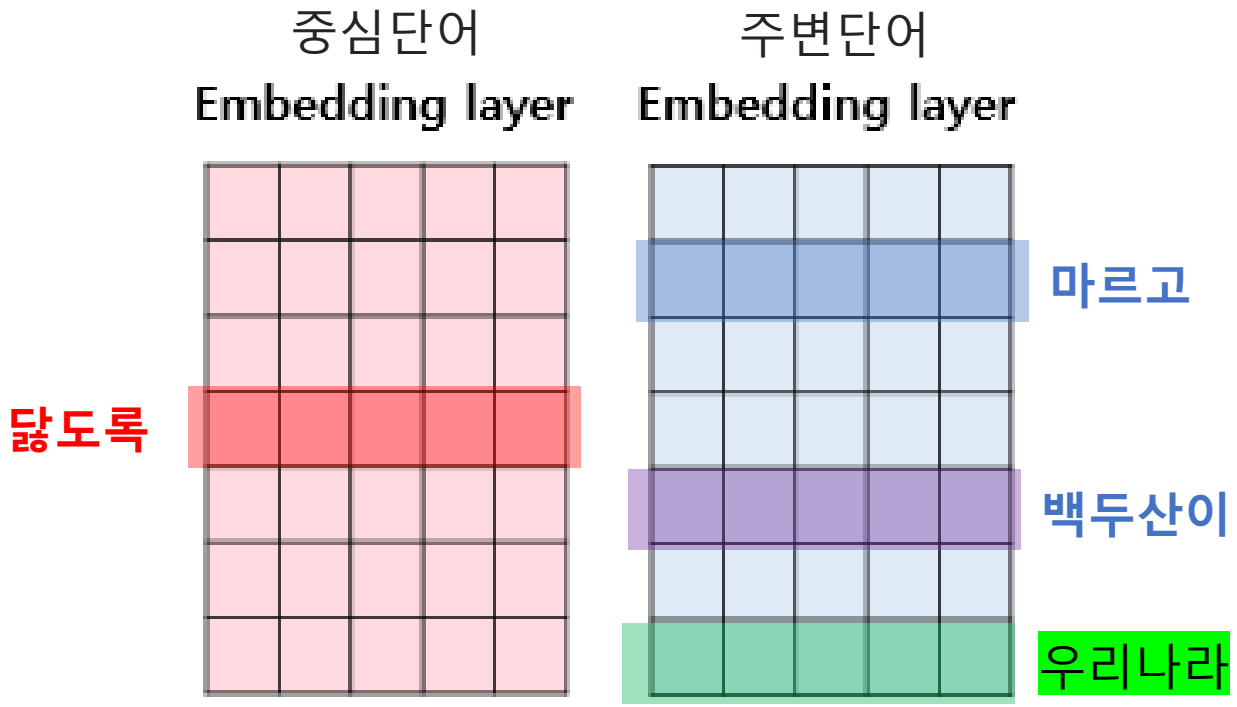


(총 10개 단어)

동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라 만세 무궁화 삼천리

negative positive 중심 positive negative

입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	pizza	0
cat	computer	0
cat	sat	1
cat	on	1
cat	cute	1
cat	mighty	0
...	...	...



skip-gram

총 10개의 단어로 이루어진 문서를 바탕으로
4개의 임베딩 벡터 학습

Negative sampling skip-gram

하나의 문서에서 k+1개의 단어에 대해서만
4개의 임베딩 벡터 학습

결과

skip-gram

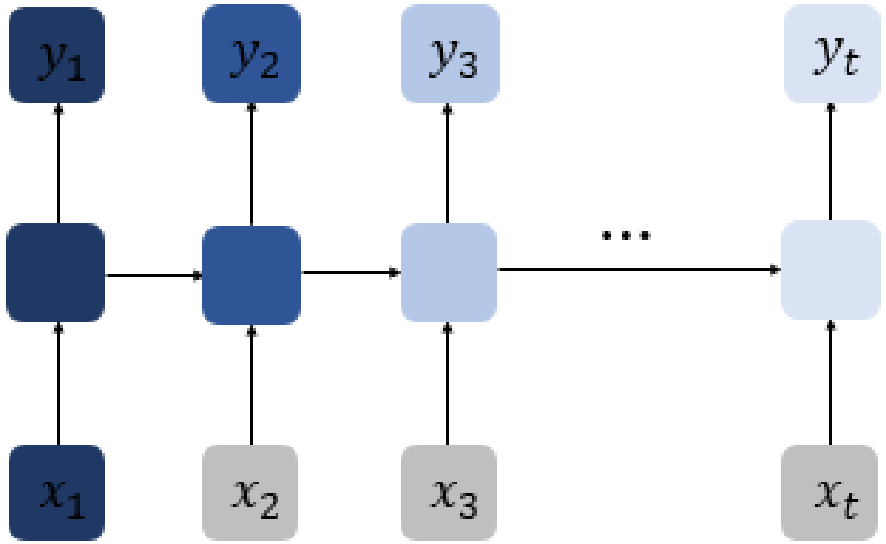
10 x 4 = 40

SGNS

(1+5) x 4 = 24

negative sample할 개수

바닐라 RNN

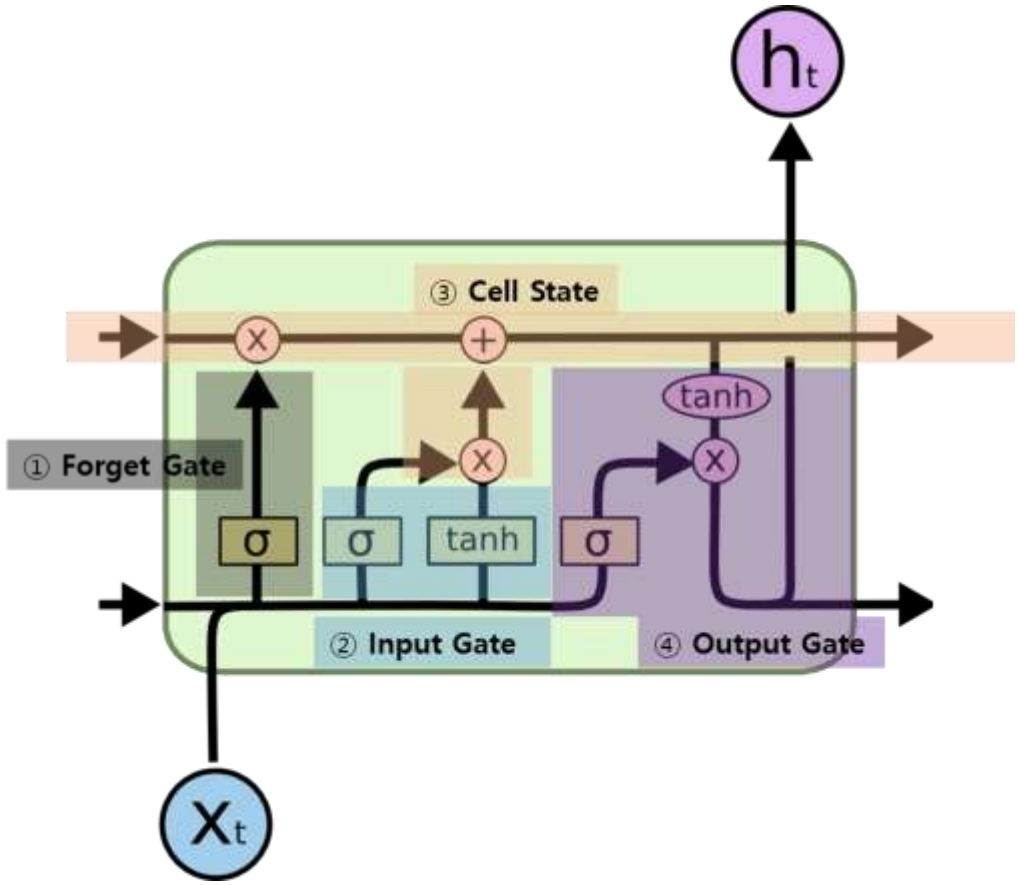


현시점의 상태가 다음 번 시점의 상태에 영향을 줌

시점이 길어질수록 + 시퀀스가 길어질수록
정보가 충분히 전달 X

LSTM

(Longh short-term memory)



① **Forget Gate** layer — 어떤 정보를 **버릴지**(잊을지) 결정 = f_t

② **Input Gate** layer — 어떤 정보를 **사용할지**(업데이트 할지) 결정
i_t (앞으로 어떤 값을 업데이트할지) × $\bar{C}_t$(새로운 후보값들 vector)

③ **Cell State** update — f_t (잊을 정보 **잊기**) + $i_t \times \bar{C}_t$ (**업데이트** 하기로 한 값)

④ **Output Gate** layer — sigmoid(input$[h_{t-1}, x_t]$)
: **Cell State**의 어느부분을 output으로 내보낼지 정하기 = O_t

X

tanh(cell state)

= h_t (: 내보내고자 하는 부분만 내보냄)

gate : 장기 의존성 정보를 유지
cell state : 장기 의존성 정보를 업데이트+전달

GRU

(Simplification of LSTM)

: LSTM 모델을 간소화시킨 버전

- Cell state가 없음

Tagging Task : 각 단어가 어떤 유형에 속해 있는지를 알아내는 작업

(개체명 인식 - 정보 추출 / 품사 태깅 - 문장 구조 분석)

양방향 LSTM
(앞뒤 시점의 입력을 모두 참고)

seq2seq : 입력 시퀀스와 다른 길이의 출력 시퀀스로 변환할 때 사용

ex) 번역, 질문응답, 텍스트 요약

두 개의 RNN 연결
(인코더-디코더 구조)