

A dark background featuring a complex, abstract network visualization composed of numerous small, glowing red, green, and blue dots connected by thin lines, resembling a neural network or a molecular structure.

Summer NLP

Lec 04. Word Embedding

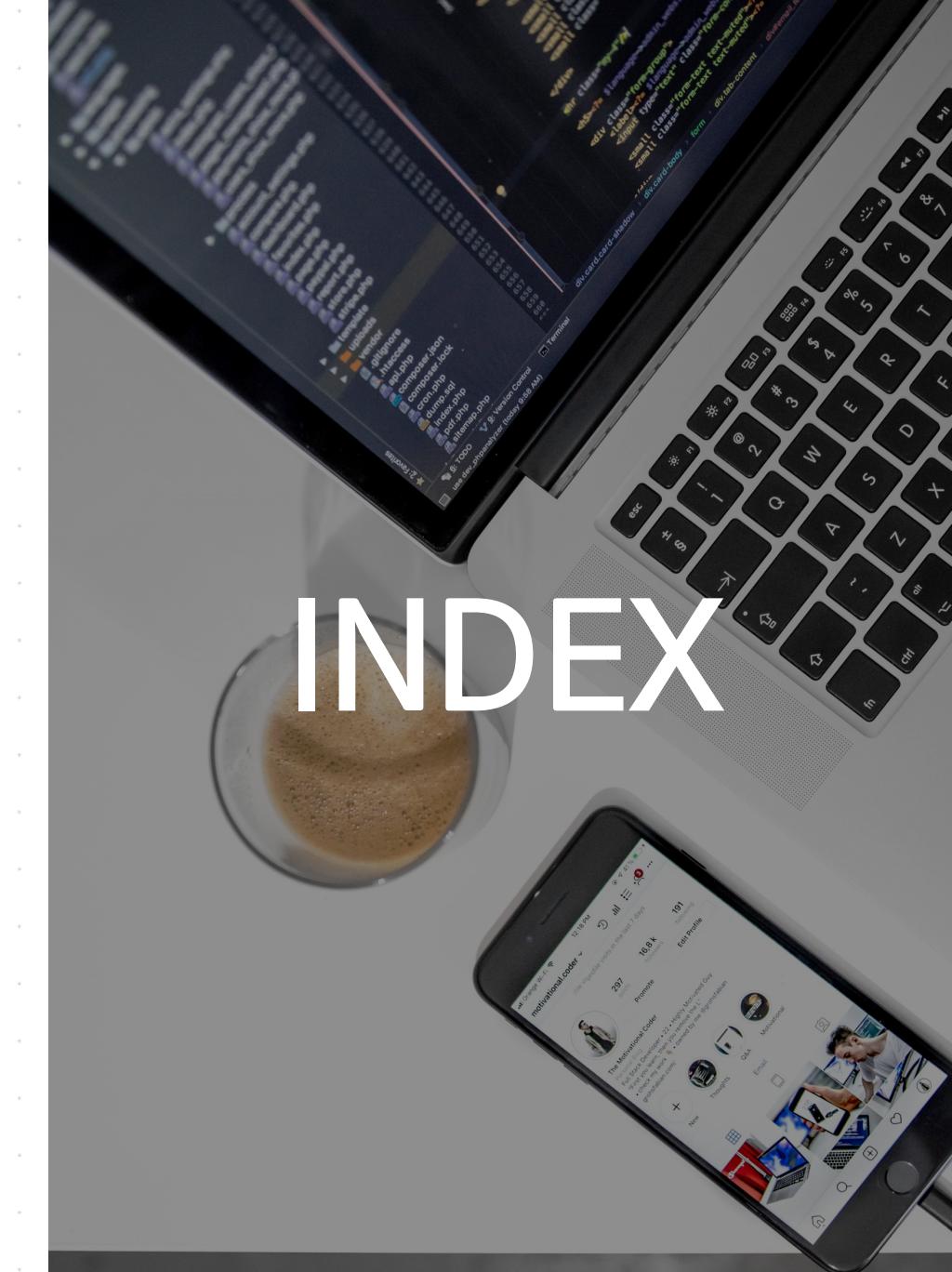
파이토치로 배우는 자연어 처리 CHAPTER 5

| 임베딩의 역할

| 임베딩의 원리

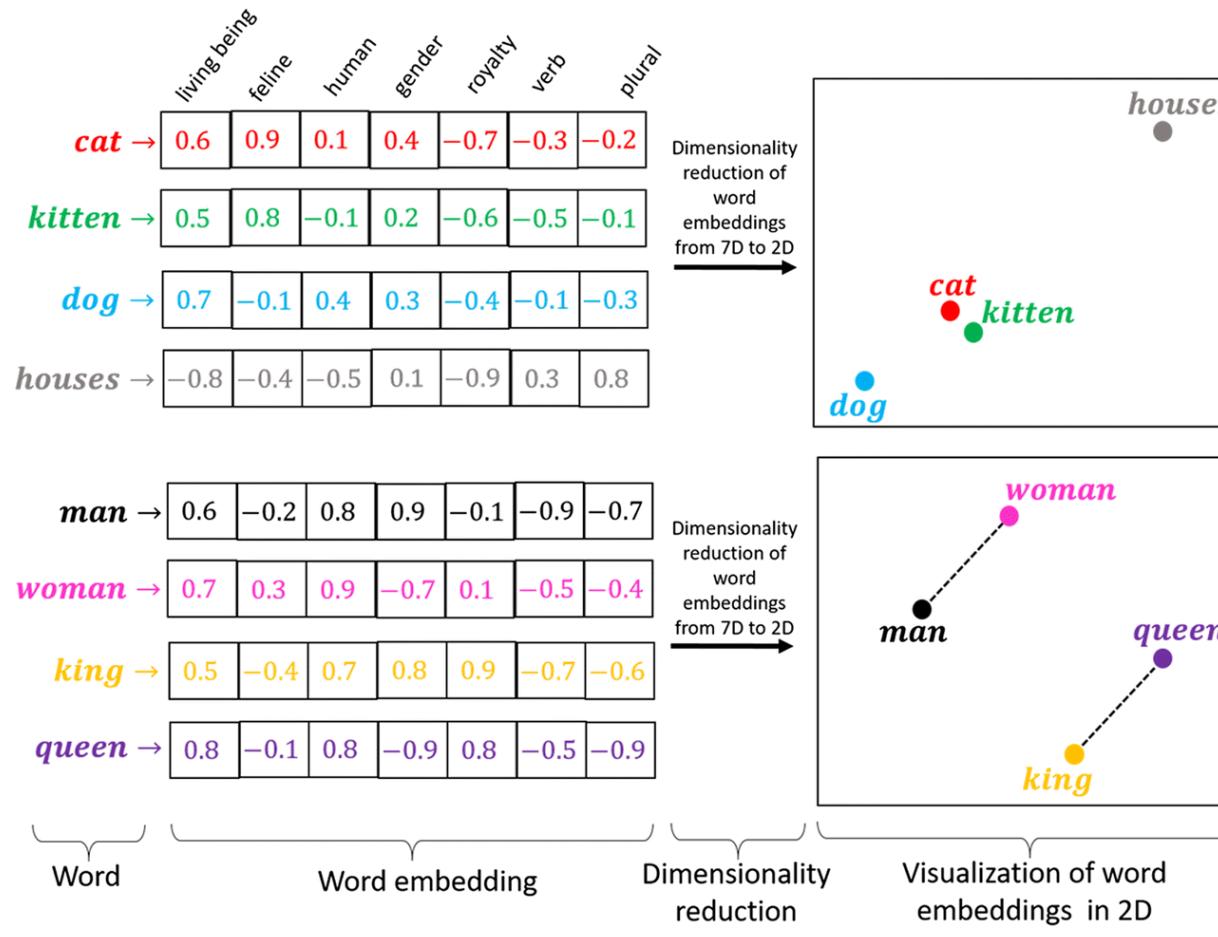
| 실습 : CBOW 학습하기

| 실습 : CBOW 사용하기



INDEX

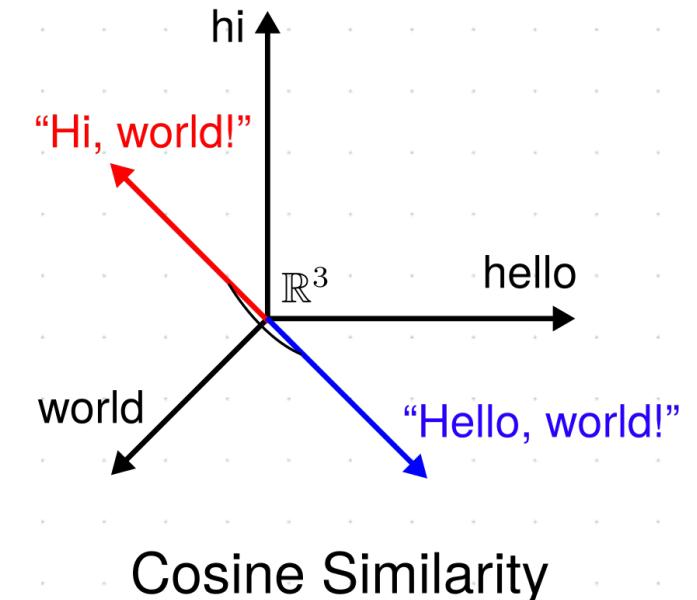
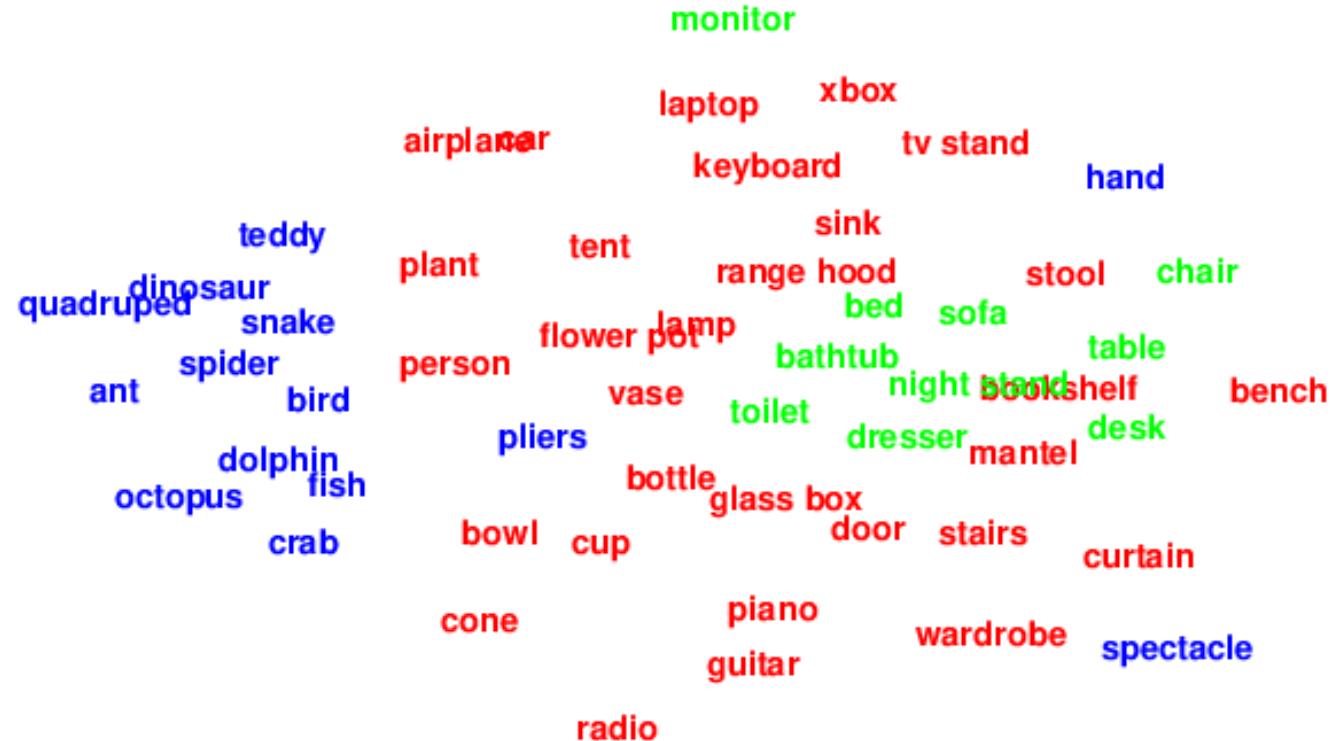
임베딩의 역할



Embedding

자연어를 기계가 이해할 수 있는 형태인 벡터로 바꾼 결과 혹은 일련의 과정

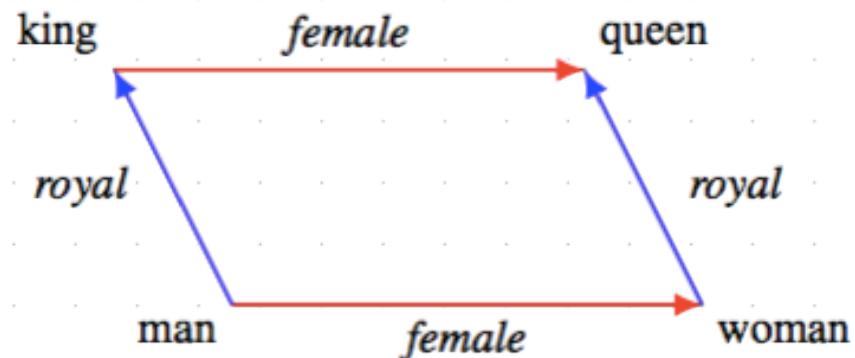
임베딩의 역할



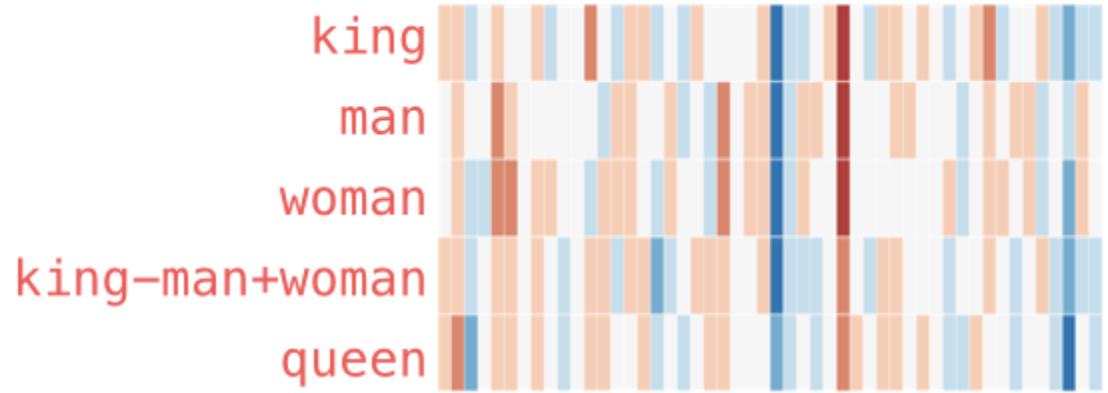
임베딩의 역할 1 : 단어/문장 간 관련도 계산

단어를 벡터공간에 적절히 위치시켜 각 단어의 관계를 수학적으로 표현한다.

임베딩의 역할



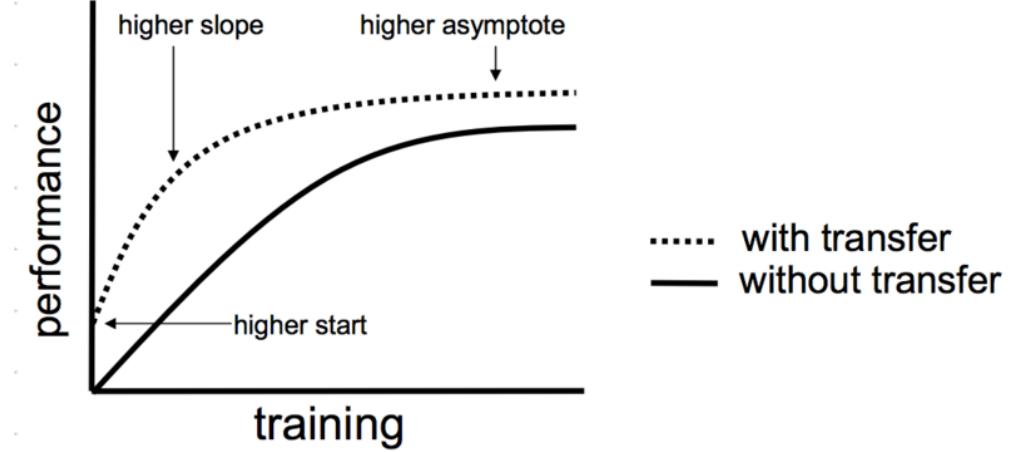
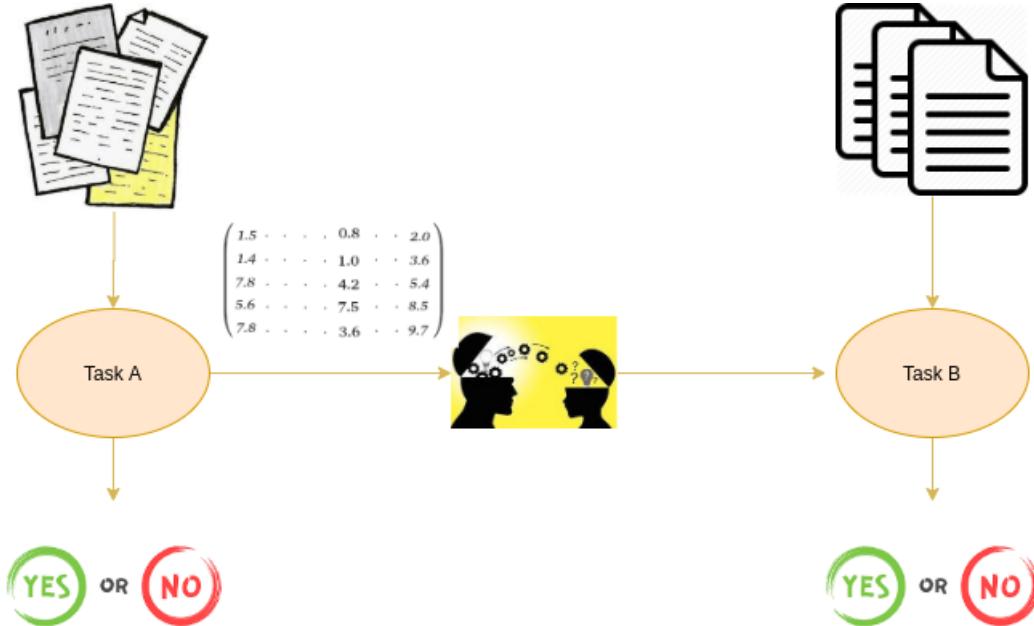
king - man + woman ~ queen



임베딩의 역할 2 : 의미/문법 정보 함축

단어 벡터간 덧셈/뺄셈을 통해 새로운 단어를 유추하도록 할 수 있다.

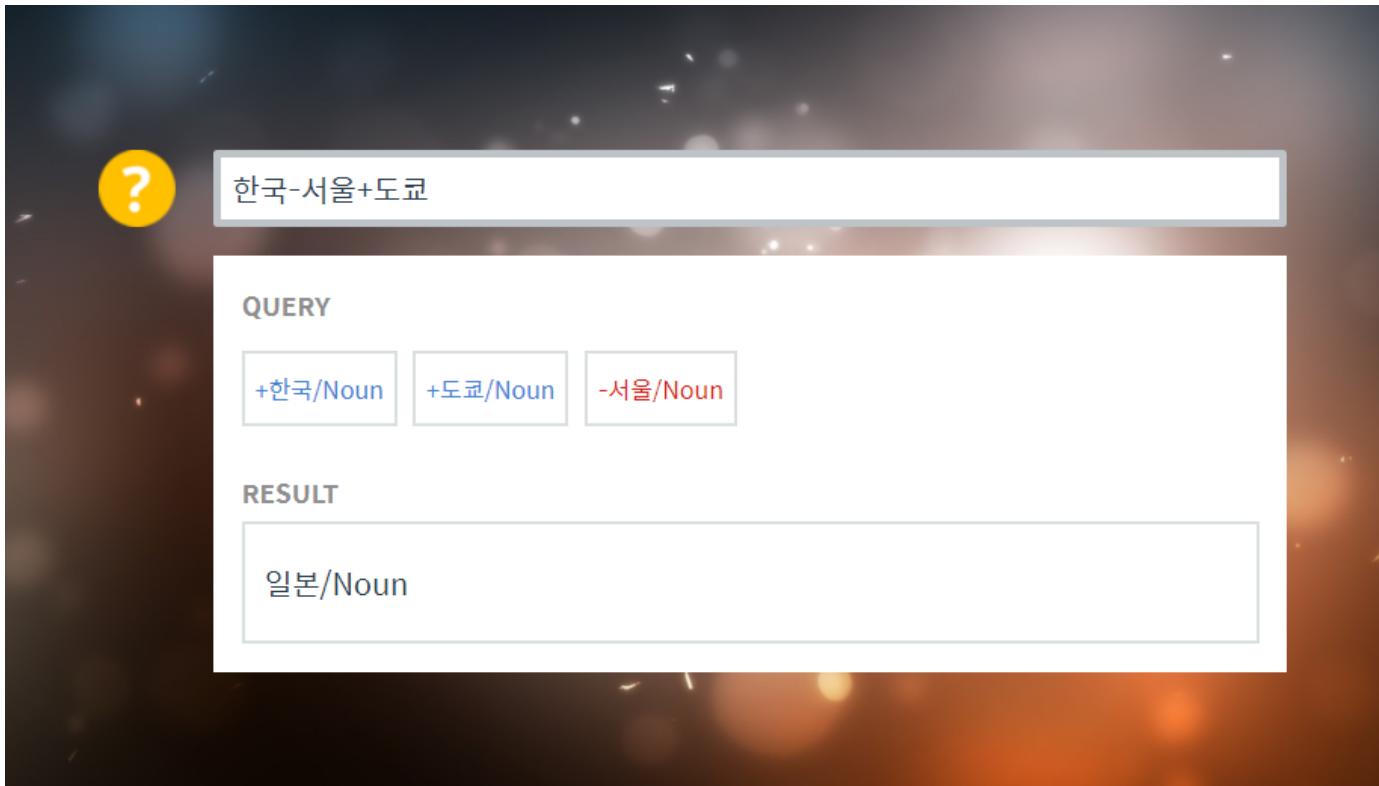
임베딩의 역할



임베딩의 역할 3 : 전이학습

사전 학습된 임베딩으로 텍스트 분류/번역 등 다양한 Task의 성능을 향상시킬 수 있다.

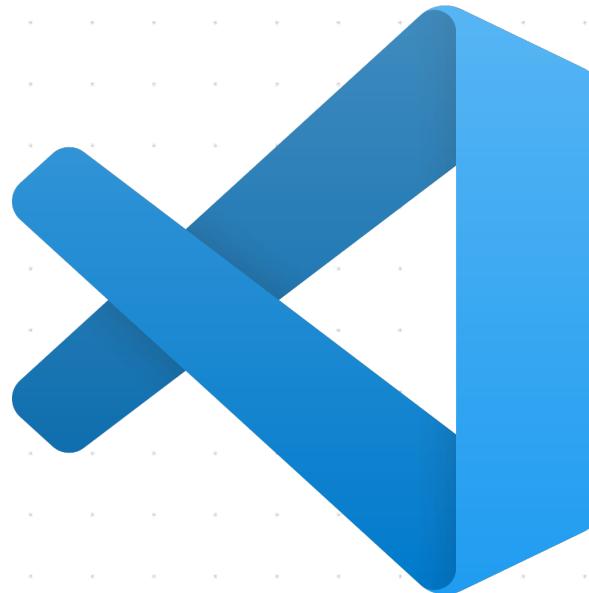
임베딩의 역할



<https://word2vec.kr/>

Word2Vec
구글이 만들었던 Embedding 기법

| 임베딩의 역할



Pretrained Embedding 사용하기 With Colab

| 임베딩의 원리

$$W(\text{"cat"}) = (0.2, -0.4, 0.7, \dots)$$

$$W(\text{"mat"}) = (0.0, 0.6, -0.1, \dots)$$

How Natural Language can Vectorized?
자연어 의미를 어떻게 벡터에 함축할 수 있을까?

| 임베딩의 원리

How Natural Language can Vectorized?
자연어의 통계적 패턴 정보를 통째로 임베딩에 반영한다!

Method 1. Bag of Words 가정 “어떤 단어가 많이 쓰였는가?”

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

임베딩의 원리

Method 1. Bag of Words 가정
“어떤 단어가 많이 쓰였는가?”

-	과일이	길고	노란	먹고	바나나	사과	싫은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

Term–Document Matrix
각 문서 내 단어별 빈도표

임베딩의 원리

Method 1. Bag of Words 가정
“어떤 단어가 많이 쓰였는가?”

$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

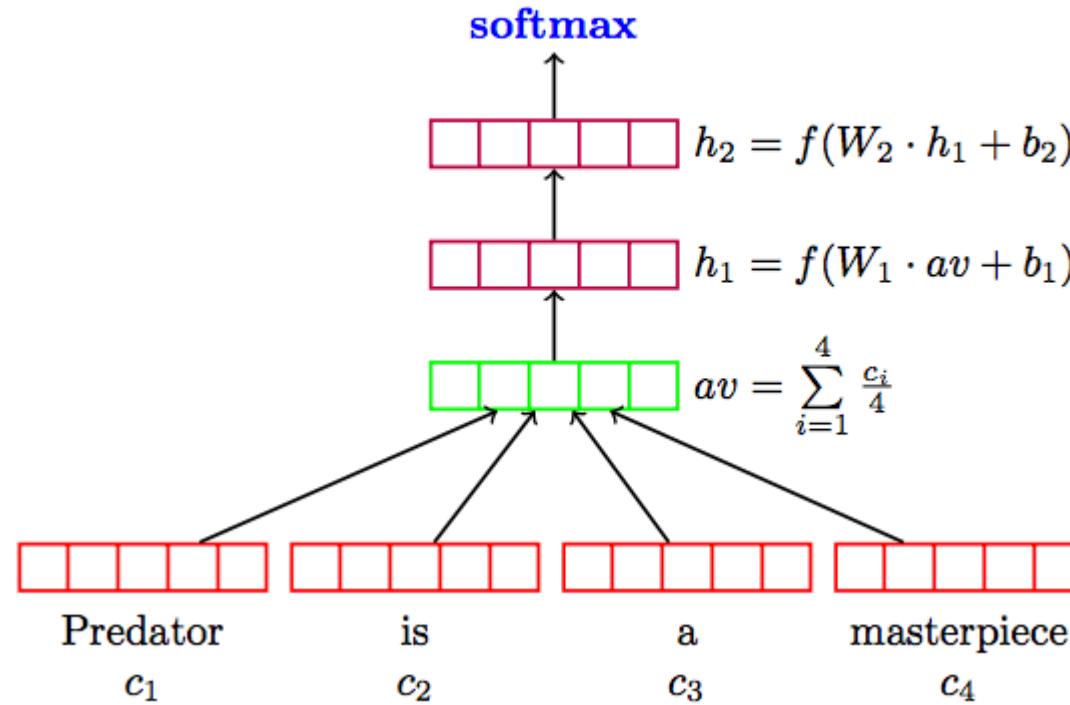
TF-IDF

“문서간 자주 등장한 단어는 그 문서의 특징을 잘 표현하지 못함”

임베딩의 원리

Method 1. Bag of Words 가정
“어떤 단어가 많이 쓰였는가?”

DAN



Deep Average Network
각 단어의 임베딩을 평균으로 사용

Method 2. 언어 모델 “단어가 어떤 순서로 쓰였는가?”

$$P(\text{Daniel ate apple.}) = 96\%$$

Example	Probability
The cat sat on the mat	0.95
The cat sad on the mat	0.20
<hr/>	
High wind tonight	0.97
Large wind tonight	0.31

언어 모델 (Language Model)

단어 시퀀스에 확률을 부여하는 모델로, 단어의 순서와 연관이 있음.

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

$P(\text{An adorable little boy is spreading smiles}) =$

$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \times P(\text{boy}|\text{An adorable little}) \times P(\text{is}|\text{An adorable little boy})$
 $\times P(\text{spreading}|\text{An adorable little boy is}) \times P(\text{smiles}|\text{An adorable little boy is spreading})$

언어 모델 (Language Model)

문장의 확률은 이전 단어가 주어졌을 때 다음 단어의 확률에 대한 곱으로 이루어짐.

$$P(\text{is}|\text{An adorable little boy}) = \frac{\text{count}(\text{An adorable little boy is})}{\text{count}(\text{An adorable little boy })}$$

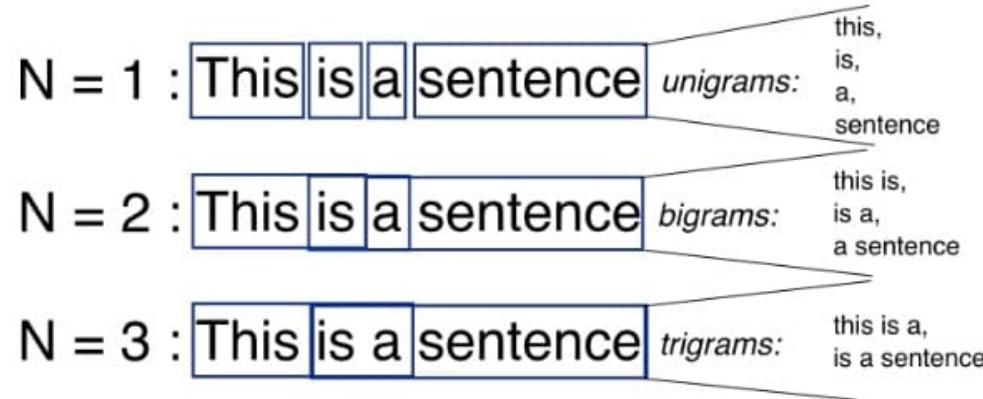
데이터가 없으면? 문자 또는 분모가 0이됨 : ‘희소 문제(Sparsity Problem)’ 발생

카운트 기반 접근

해당 문장이 등장한 횟수로 확률 계산, 하지만 데이터가 적을 경우 ‘희소 문제’ 발생

임베딩의 원리

N Gram



uni-gram (1-gram)

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{boy})$$

bi-gram (2-gram)

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{little boy})$$

마르코프 가정을 사용, “한 상태는 그 직전 상태에만 의존한다”.
즉 uni-gram은 전 단계의 단어 ‘boy’에만 의존하고,
bi-gram은 전 단계의 두 단어 ‘little boy’에 의존하도록 한다.

An adorable little boy is spreading ?
무시됨!
n-1개의 단어

$$P(w|\text{boy is spreading}) = \frac{\text{count}(\text{boy is spreading } w)}{\text{count}(\text{boy is spreading})}$$

N-gram

n개 단어를 묶어서 그 빈도를 바탕으로 학습한 언어모델

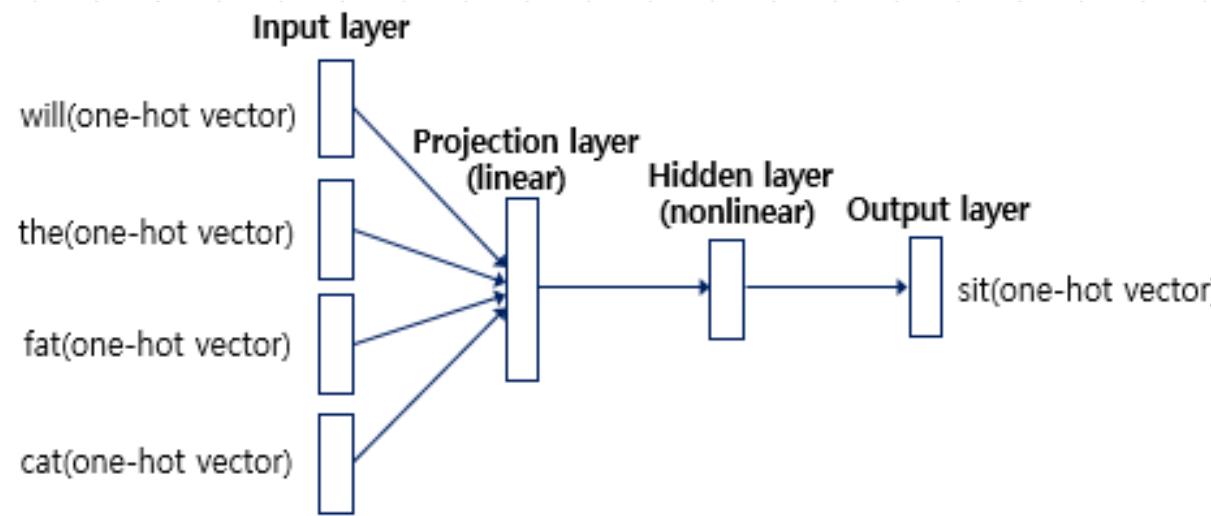
Method 2. 언어 모델

“단어가 어떤 순서로 쓰였는가?”

임베딩의 원리

Method 2. 언어 모델

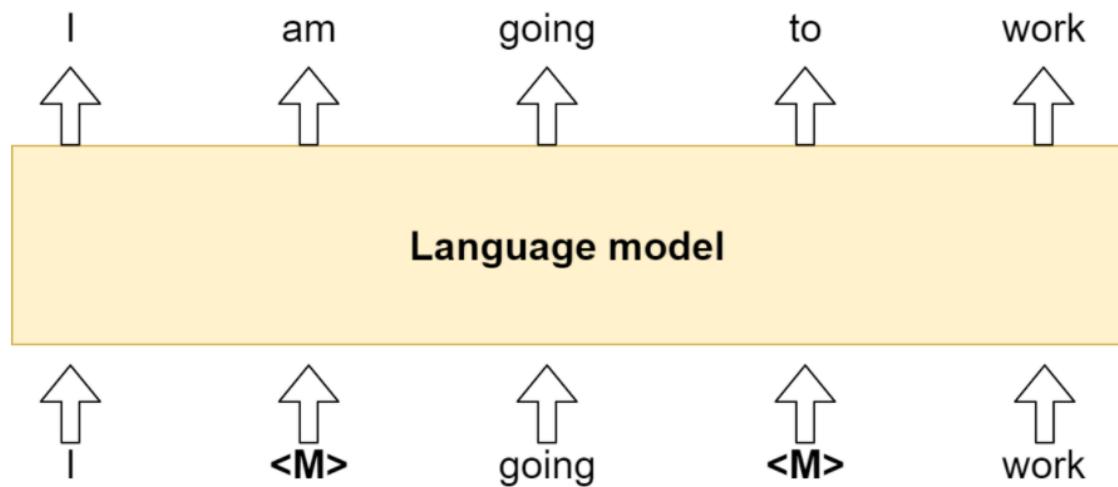
“단어가 어떤 순서로 쓰였는가?”



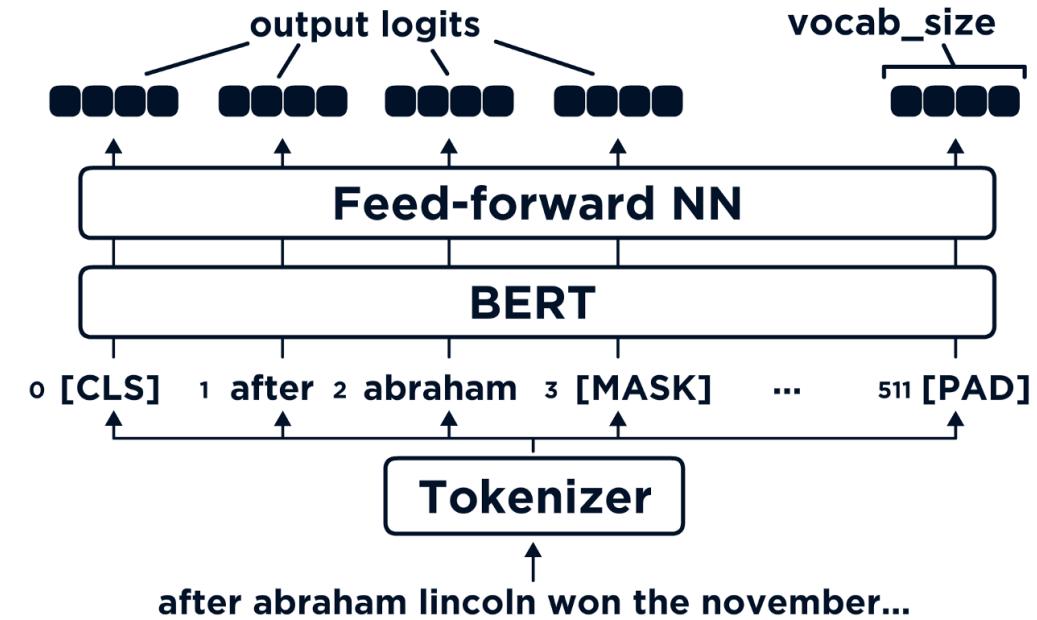
Neural Network 기반 언어 모델

Neural Network로 단어 시퀀스를 입력받으면 다음 단어를 맞추는 모델을 만든다.

임베딩의 원리



Method 2. 언어 모델
“단어가 어떤 순서로 쓰였는가?”



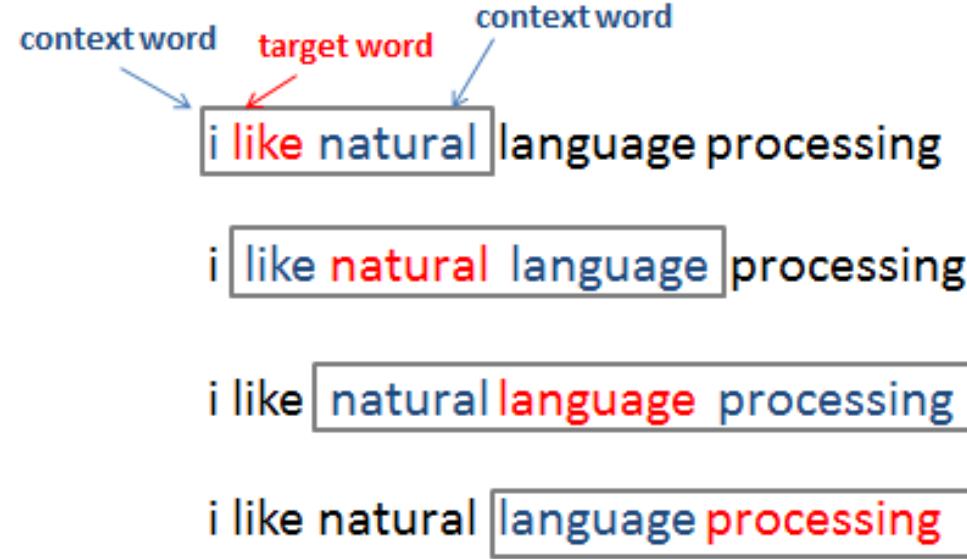
Masked Language Model

문장 중간에 마스크를 씌우고, 해당 마스크의 단어를 추측. 양방향 연산 가능하여 성능 향상!

Method 3. 분포 가정 “어떤 단어가 같이 쓰였는가?”

임베딩의 원리

Method 3. 분포 가정
“어떤 단어가 같이 쓰였는가?”



위와 같이 Sliding Window를 돌며, Target 단어와 그 주변부 단어인 Context 단어가 이웃하여 자주 등장하면, 그 두 단어의 의미는 비슷한 의미를 가질 것이라는 것이 분포 가정!

NLP에서의 분포

윈도우 내에 동시에 등장하는 이웃 단어 또는 문맥의 집합

임베딩의 원리

$$PMI(a, b) = \log\left(\frac{P(a, b)}{P(a)P(b)}\right)$$

Note (직접 계산해 보세요)

I enjoy flying

I like NLP

I like deep learning

-	I	like	enjoy	deep	learing	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	0
deep	0	1	0	0	1	0	1
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	1	0

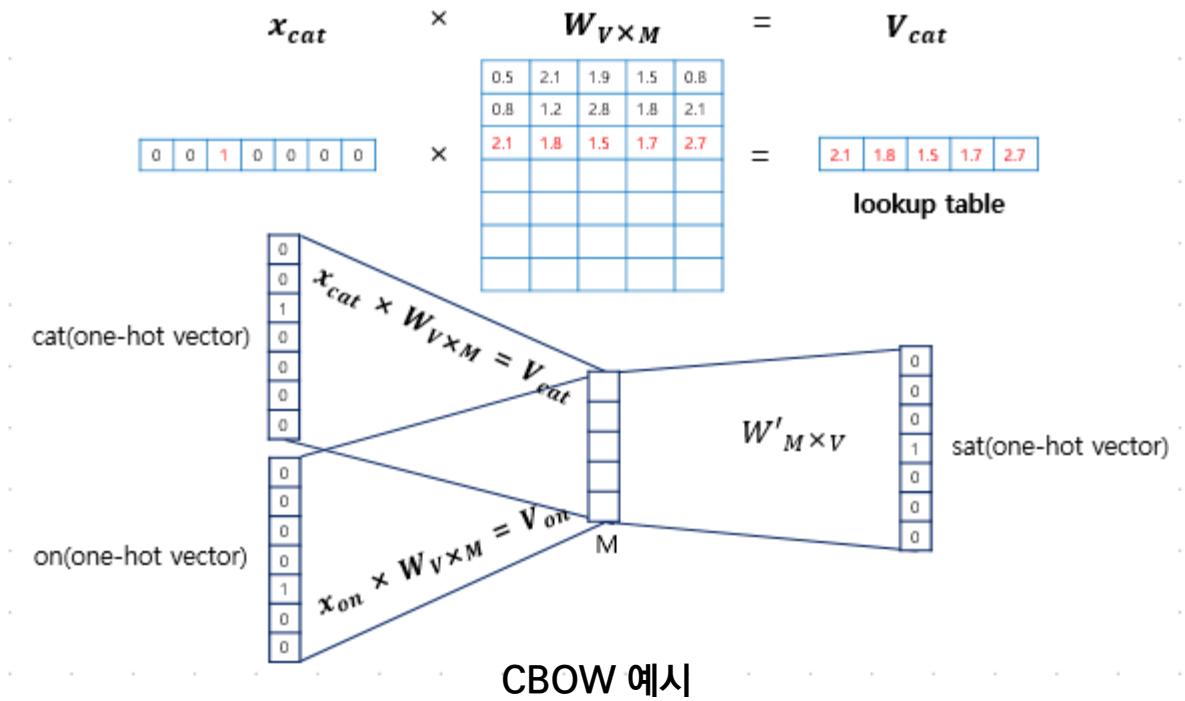
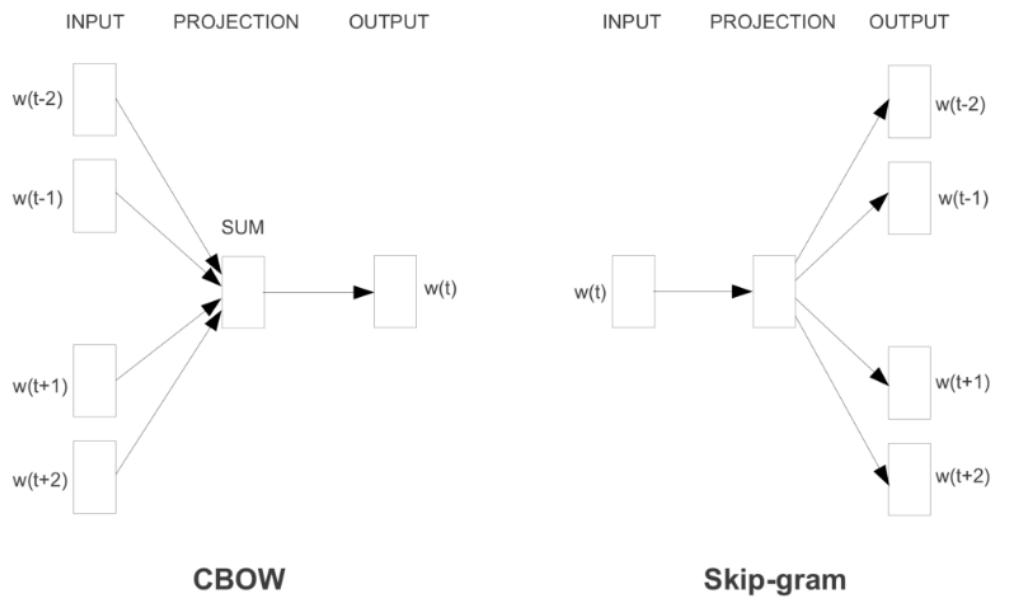
PMI (점별 상호 정보량)

두 확률변수 사이의 상관성을 계량화 : 독립일 경우 0, 상관 관계가 높으면 커짐.

Method 3. 분포 가정
“어떤 단어가 같이 쓰였는가?”

임베딩의 원리

Method 3. 분포 가정
“어떤 단어가 같이 쓰였는가?”



Word2Vec : CBOW and Skip-gram

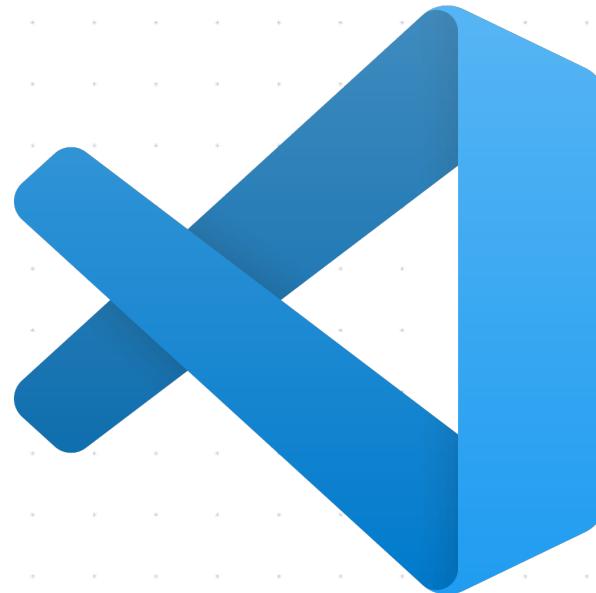
CBOW는 문맥단어로 타깃단어를, Skip-gram은 타깃단어로 문맥단어를 예측한다.

| 실습 : CBOW 학습하기



CBOW 학습하기
[With Colab](#)

| 실습 : CBOW 사용하기



CBOW 사용하기
[With Colab](#)



Thank You