단어표현

딥러닝 / 머신러닝

임 경 태

1. 단어표현의 개요

단어 표현?

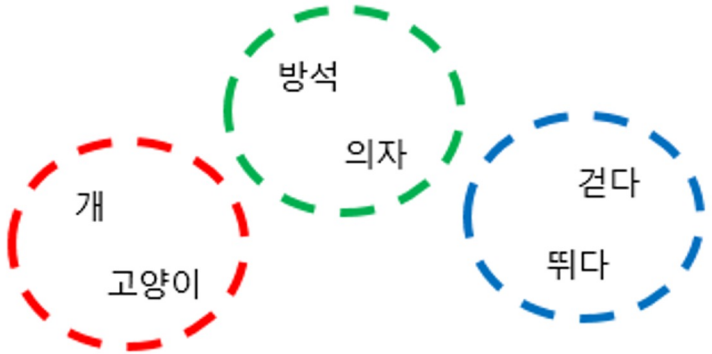
자연어를 컴퓨터가 이해하고, 효율적으로 처리하게 하기 위해서는
컴퓨터가 이해할 수 있도록 자연어를 적절히 변환할 필요가 있다.

```
9976970 아 더빙.. 진짜 짜증나네요 목소리        0
3819312 흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나        1
10265843        너무재밋었다그래서보는것을추천한다       0
9045019 교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정        0
6483659 사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 던스트가 너무나도 이뻐보였다  1
5403919 막 걸음마 뗀 3세부터 초등학교 1학년생인 8살용영화.ㅋㅋㅋ...별반개도 아까움.      0
```

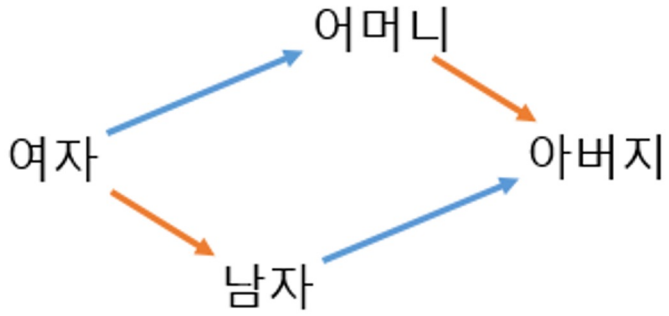
문자를 숫자로 변환하여 컴퓨터가 이해하도록 하는 것
-> Word Embedding

단어의 벡터화 방식은
1) 희소표현(Sparse Representation) 방식
2) 밀집표현(Dense Representation) 방식이 존재함

"유사 의미의 벡터는 가까운 공간에 존재"

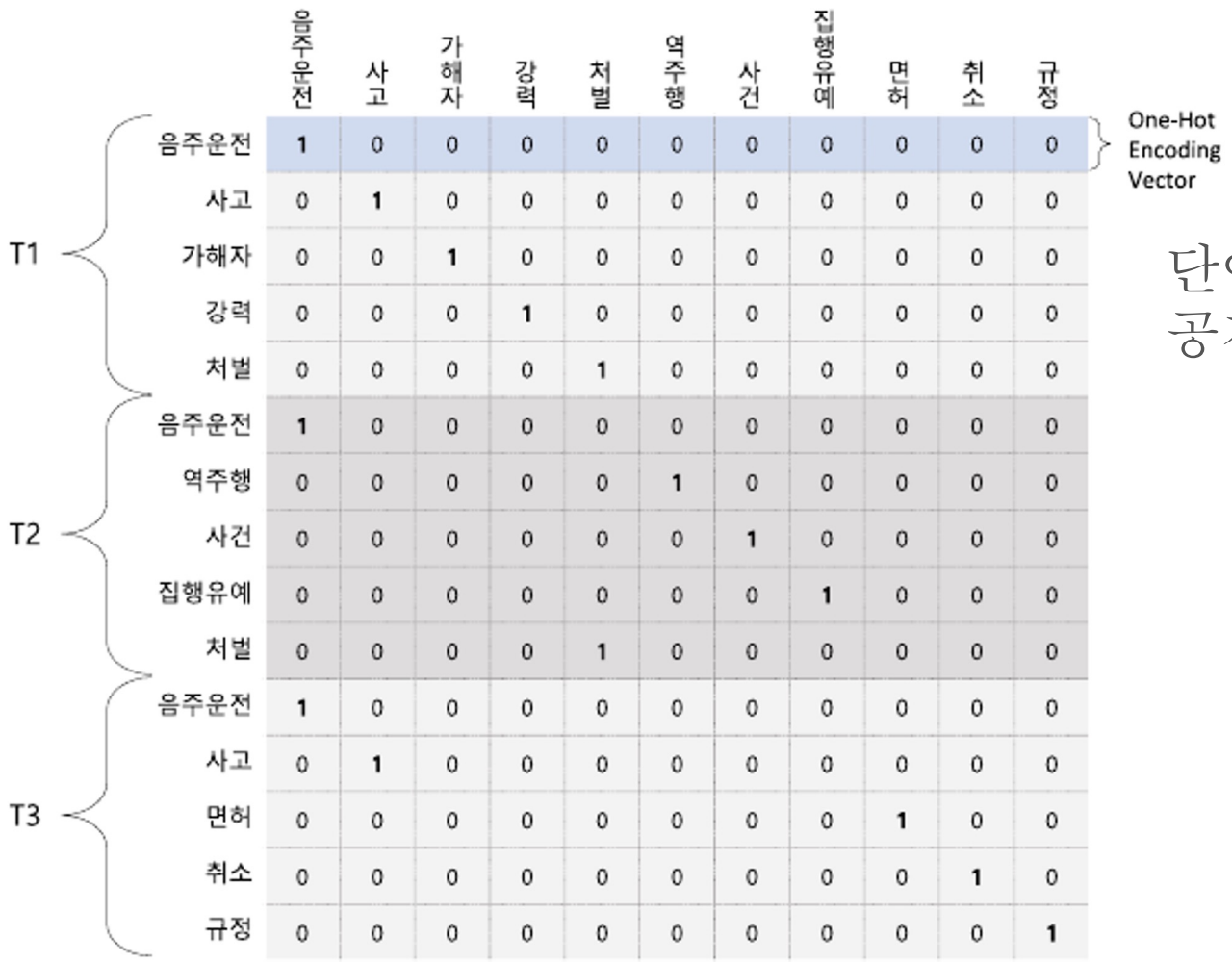


"단어를 벡터화 하여 산술 연산"



실습 http://w.elnn.kr/search/

희소 표현(Sparse Representation)



		음주운전	사고	가해자	강력	처벌	역주행	사건	집행유예	면허	취소	규정
T1	음주운전	1	0	0	0	0	0	0	0	0	0	0
	사고	0	1	0	0	0	0	0	0	0	0	0
	가해자	0	0	1	0	0	0	0	0	0	0	0
	강력	0	0	0	1	0	0	0	0	0	0	0
	처벌	0	0	0	0	1	0	0	0	0	0	0
T2	음주운전	1	0	0	0	0	0	0	0	0	0	0
	역주행	0	0	0	0	0	1	0	0	0	0	0
	사건	0	0	0	0	0	0	1	0	0	0	0
	집행유예	0	0	0	0	0	0	0	1	0	0	0
	처벌	0	0	0	0	1	0	0	0	0	0	0
T3	음주운전	1	0	0	0	0	0	0	0	0	0	0
	사고	0	1	0	0	0	0	0	0	0	0	0
	면허	0	0	0	0	0	0	0	0	1	0	0
	취소	0	0	0	0	0	0	0	0	0	1	0
	규정	0	0	0	0	0	0	0	0	0	0	1

One-Hot Encoding Vector

단어간 유사도 반영 X
공간적 비용 최악

사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원을 맞춤

11차원

	음주운전	사고	가해자	강력	처벌	역주행	사건	집행유예	면허	취소	규정
음주운전	1	0	0	0	0	0	0	0	0	0	0

↓ 음주운전 = [0.2 1.8 1.1 -2.1 1.1 2.8 .]

6차원

밀집 벡터(dense vector)의 형태로 표현하는 방법
-> 워드 임베딩(word embedding)

	원-핫 벡터	임베딩 벡터
차원	고차원	저차원
다른 표현	희소 벡터의 일종	밀집 벡터의 일종
표현 방법	수동	훈련 데이터로부터 학습
값의 타입	1과 0	실수

워드 임베딩 방법론으로는 **LSA, Word2Vec, FastText, Glove**가 있다

희소 표현으로는 유사성 표현 X -> **분산 표현(Distributed Representation)**

분포 가설(distributional hypothesis)
'비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다'

희소 표현 – one-hot-encoding

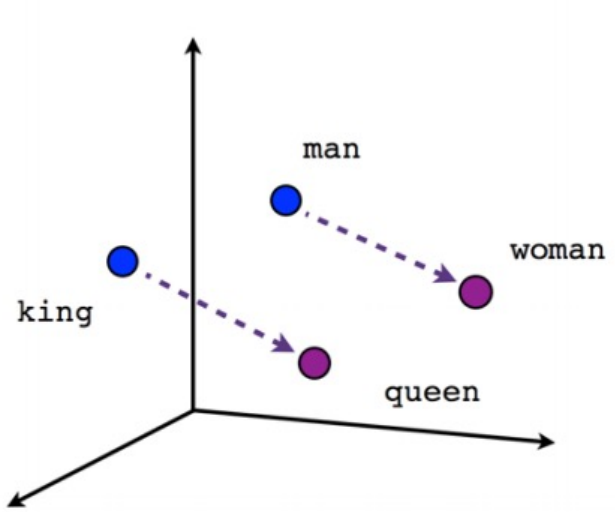
강아지 = [0 0 0 0 1 0 0 0 0 0 0 ... 중략 ... 0]

분산 표현 –

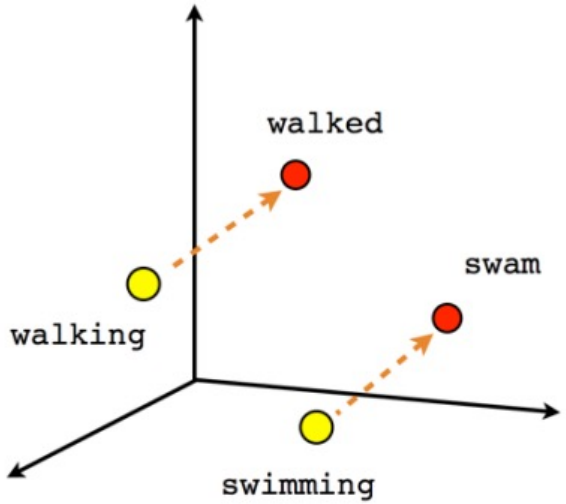
강아지 = [0.2 0.3 0.5 0.7 0.2 ... 중략 ... 0.2]

단어의 의미를 여러 차원에 분산하여 표현
이런 표현 방법을 사용하면 **단어 간 유사도**를 계산할 수 있음

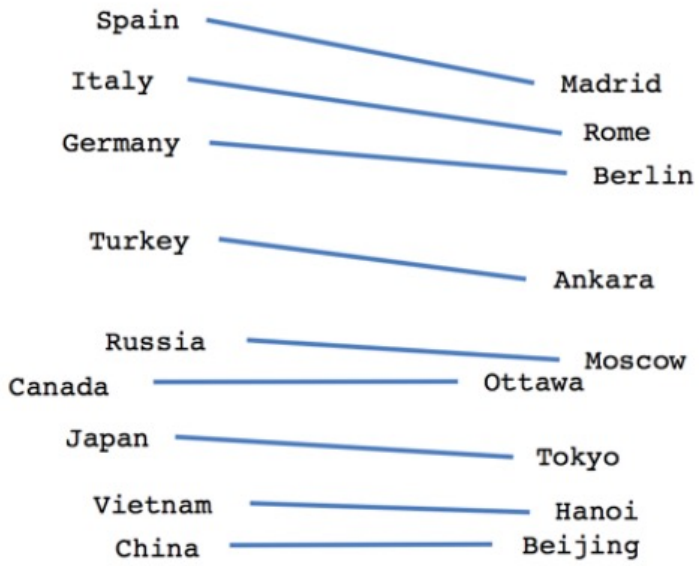
분포 가설(distributional hypothesis)
'비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다'



Male-Female



Verb tense

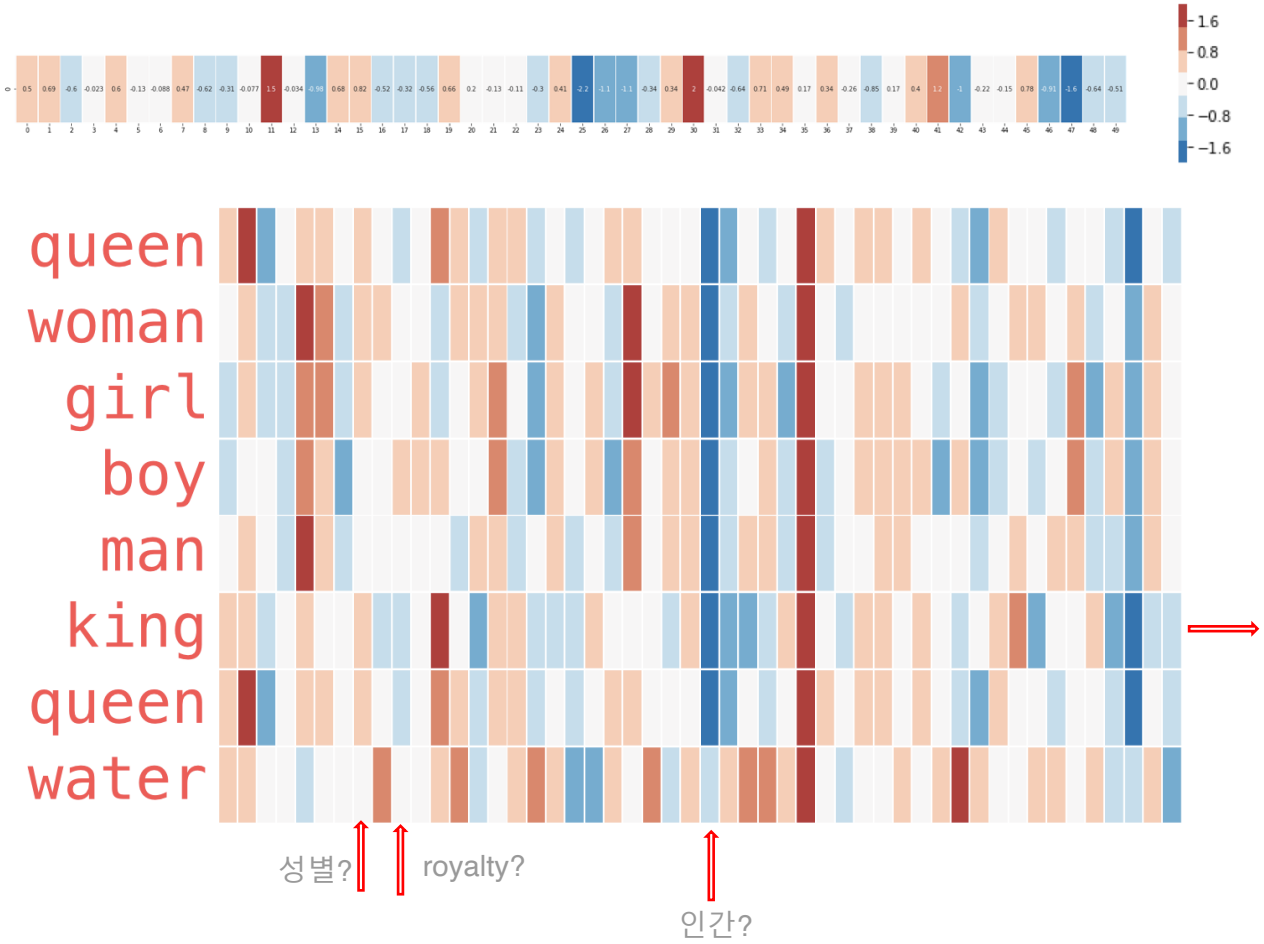


Country-Capital

실습 http://w.elnn.kr/search/

분포 가설(distributional hypothesis)

'비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다'



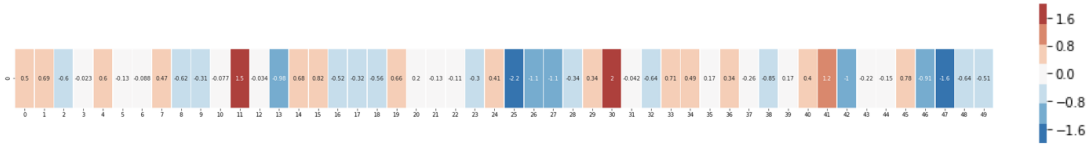
[관측포인트]

1. 각각의 컬럼이 어떤 의미를 함축하고 있나 알 수 없음
2. Man과 boy의 유사성과 woman과 girl의 유사성
3. Water와 비교 시 사람을 함축하는 컬럼
4. Royalty로 가정할 수 있는 컬럼이 있음

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 ,
-0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961
, -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 ,
-0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 ,
-1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

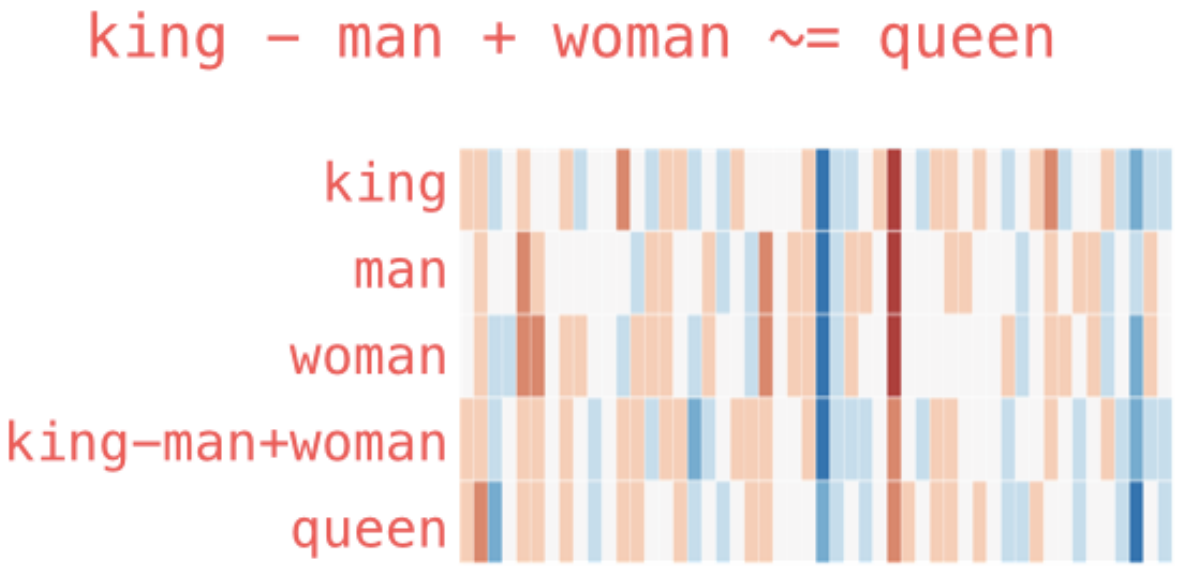
참조 https://jalammar.github.io/illustrated-word2vec/

분포 가설(distributional hypothesis)
'비슷한 위치에서 등장하는 단어들은 비슷한 의미를 가진다'




```
model.most_similar(positive=["king","woman"], negative=["man"])
```

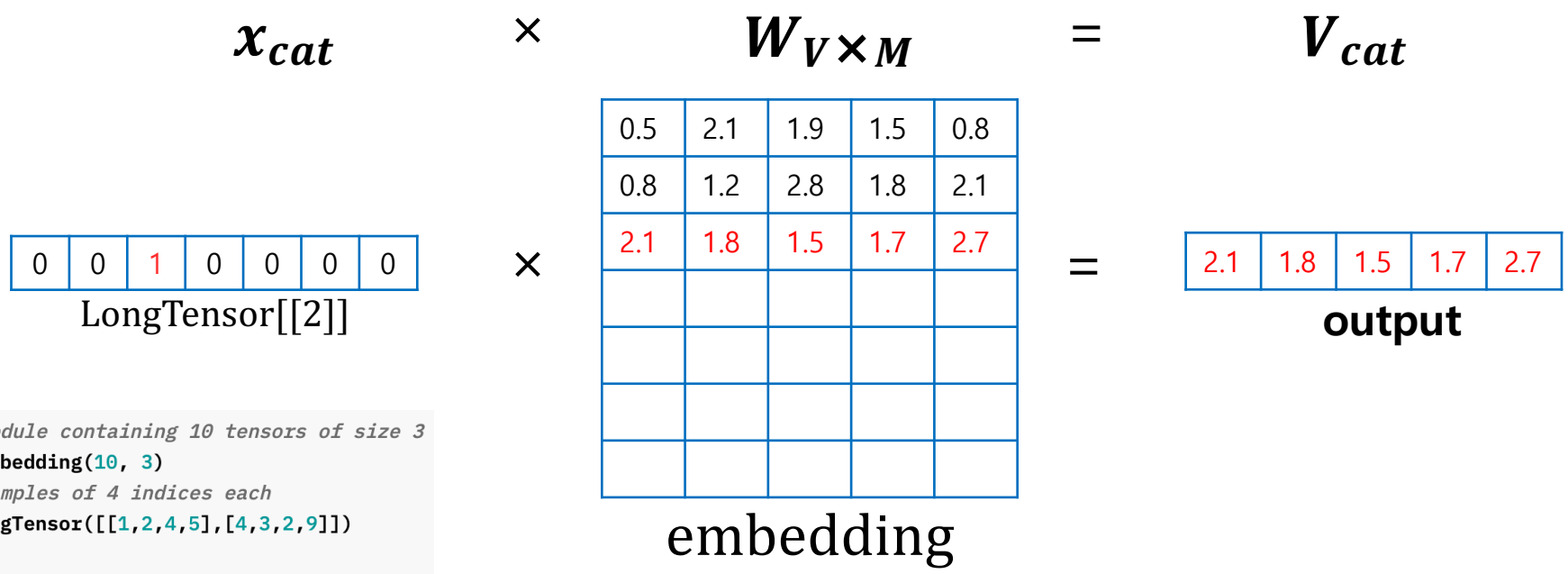
```
[('queen', 0.8523603677749634),
 ('throne', 0.7664333581924438),
 ('prince', 0.7592144012451172),
 ('daughter', 0.7473883032798767),
 ('elizabeth', 0.7460219860076904),
 ('princess', 0.7424570322036743),
 ('kingdom', 0.7337411642074585),
 ('monarch', 0.721449077129364),
 ('eldest', 0.7184862494468689),
 ('widow', 0.7099430561065674)]
```



The resulting vector from "king-man+woman" doesn't exactly equal "queen", but "queen" is the closest word to it from the 400,000 word embed dings we have in this collection.

참조 https://jalammar.github.io/illustrated-word2vec/

분산 표현(Distributed Representation)은 어떻게 구현할까?




```
>>> # an Embedding module containing 10 tensors of size 3
>>> embedding = nn.Embedding(10, 3)
>>> # a batch of 2 samples of 4 indices each
>>> input = torch.LongTensor([[1,2,4,5],[4,3,2,9]])
>>> embedding(input)
tensor([[[-0.0251, -1.6902,  0.7172],
         [-0.6431,  0.0748,  0.6969],
         [ 1.4970,  1.3448, -0.9685],
         [-0.3677, -2.7265, -0.1685]],

        [[ 1.4970,  1.3448, -0.9685],
         [ 0.4362, -0.4004,  0.9400],
         [-0.6431,  0.0748,  0.6969],
         [ 0.9124, -2.3616,  1.1151]]])
```

2. 단어표현의 학습 알고리즘


감사합니다.