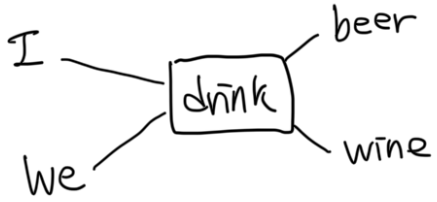
You/say/goodbye/and/I/say/hello.
0 1 2 3 4 1 5 6

말뭉치 preprocess

- word2id 각 딕셔너리에 보관
- id2word
- np.array(idx) → 순서대로 key 값저장
  (0,1,2,3,4,1,5,6) ⇒ corpus

색도 RGB ex) (110, 33, 22) 라하면 빨간색 계열임을 알수 있는 것처럼
단어도 벡터를 사용하여 분산 표현을 구축

분포가설 - 단어의 의미는 주변단어에 의해 형성된다.
단어자체의 의미보다 맥락이 의미를형성함



you say goodbye and I say hello. : 윈도우 크기가 2인 맥락의 예

동시발생 행렬 - 어떤 단어에 주목했을때 그 주변에 어떤 단어가 몇번이나 등장하는지 세어 집계

window size: 1

you say goodbye and I say hello.

	you	say	goodbye	and	I	hello	.
you	0	1	0	0	0	0	0
say	1	0	1	0	1	1	0

→ C[word2id['say']]

say의 백터표현

goodbye	0	1	0	1	0	0	0
and	0	0	1	0	1	0	0
I	0	0	0	1	0	1	0
hello	0	1	0	0	0	0	1
.	0	0	0	0	0	1	0

벡터간 유사도 - 단어 벡터의 유사도를 나타낼 때는 코사인 유사도를 자주 이용
→ 두벡터가 가리키는 방향이 얼마나 비슷한가 (같 1, 반대 -1)

$x = \langle x_1, x_2, \cdots, x_n \rangle \quad y = \langle y_1, y_2, \cdots, y_n \rangle$

$$similarity(x, y) = \frac{x \cdot y}{|x||y|} = \frac{x_1y_1 + x_2y_2 + \cdots + x_ny_n}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}\sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}}$$

유사 단어의 랭킹 표시

most_similar(query, word2id, id2word, word_matrix, top=5)
- query: 검색어 (단어)
- word_matrix: 단어 벡터들을 모은 행렬
- top=5: 상위 몇개까지 출력?

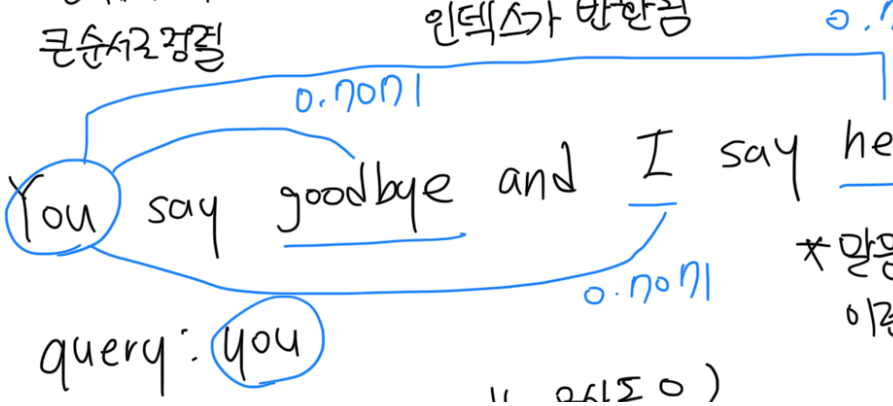
↓

① 검색어의 단어 벡터를 꺼낸다

② 코사인유사도계산 (검색어의 단어벡터와 다른 모든 단어벡터)

③ 코사인 유사도를 기준으로 내림차순 출력

- similarity.argsort()
  - similarity 배열을 정렬
  - 인덱스가 반환됨
- 단어유사도가 큰순서로정렬



query: you

* 말뭉치가 작아서 이런 결과가 나옴

(say와 and는 ~~~

통계 기반 기법 개선하기

기존
고빈도 단어

the car

car는 사실 drive 와 관련이 깊으나 the 와 같이 발생횟수가 높아 강한 관련성을 가진다고 나옴

→ 점별 상호정보량 (PMI)
확률변수 x, y에 대해 정의됨

$$PMI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

P(x) : x가 일어날 확률

P(y) : y가 일어날 확률

P(x,y) : x와 y가 동시에 일어날 확률

PMI 값이 높을수록 관련성이 높다.

P(x) : 단어 x가 말뭉치에 등장할 확률

ex) 10_000개 단어로 이루어진 말뭉치에서 the가 100번 등장

$$P("the") = \frac{100}{10000} = 0.01$$

the 와 car 가 10번 동시발생했다면

$$P("the", "car") = \frac{10}{10000} = 0.001$$

$$PMI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} = \log_2 \frac{\frac{C(x,y)}{N}}{\frac{C(x)}{N}\frac{C(y)}{N}} = \log_2 \frac{C(x,y)\cdot N}{C(x)C(y)}$$

C는 동시발생행렬

C(x,y) : x와 y가 동시발생하는 횟수

N : 말뭉치에 포함된 단어수

$$PMI("the", "car") = \log_2 \frac{10 \cdot 10000}{100 \cdot 20} \fallingdotseq 2.32$$

N = 10000

the 와 car의 동시발생횟수 = 10

car와 drive의 동시발생횟수 = 5

the 횟수 = 1000

car 횟수 = 20

$PMI("car", "drive") = \log_2 \frac{5 \cdot 10000}{20 \cdot 10} \fallingdotseq 7.97$ 관련이 더큼

하지만, 두단어의 동시발생횟수가 0이면 $\log_2 0 = -\infty$ 가됨

그래서! 실제 구현시 양의 상호정보량 (PPMI) 사용

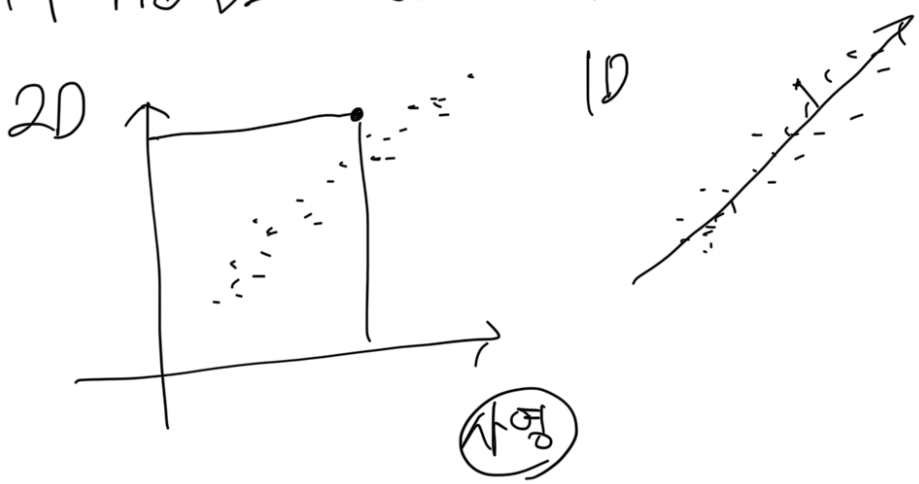
$$PPMI(x,y) = \max(0, PMI(x,y))$$

PMI가 음수일때 0으로 취급 → 단어사이의 관련성을 0이상으로 표현

그러나, 말뭉치의 어휘수가 10만개라면 그벡터의 차원도 10만

↳ 너무 크다. 그리고 대부분이 0임 (대부분이 안중요)

벡터의 차원 감소 → 중요정보 유지



특이값분해 (SVD)

$$X = USV^T$$

U : 직교행렬 → 단어공간 $= AA^T$ ↑ 해당 축의 중요도

↳ 대각성분이 특이값이 큰순서대로 나열

S : 대각행렬 [illegible]

V : 직교행렬 $= A^TA$

$$\underset{U}{m\times m}\cdot\underset{S}{m\times n}\cdot\underset{V^T}{n\times n}$$

중요도가 높다

ex)

$$A=\begin{pmatrix}4&0\\3&-5\\0&-3\end{pmatrix}=USV^T=\begin{pmatrix}-0.64&0.49&0.59\\-0.64&-0.85&0.07\\-0.42&0.19&-0.89\end{pmatrix}\begin{pmatrix}7.07&0\\0&3\\0&0\end{pmatrix}\begin{pmatrix}-0.85&-0.55\\0.53&-0.85\end{pmatrix}$$

⇓ 차원축소

$$U'S'V'^T=\begin{pmatrix}-0.64\\-0.64\\-0.42\end{pmatrix}\begin{pmatrix}7.07&0\\0&0\\0&0\end{pmatrix}\begin{pmatrix}-0.85&-0.53\end{pmatrix}$$