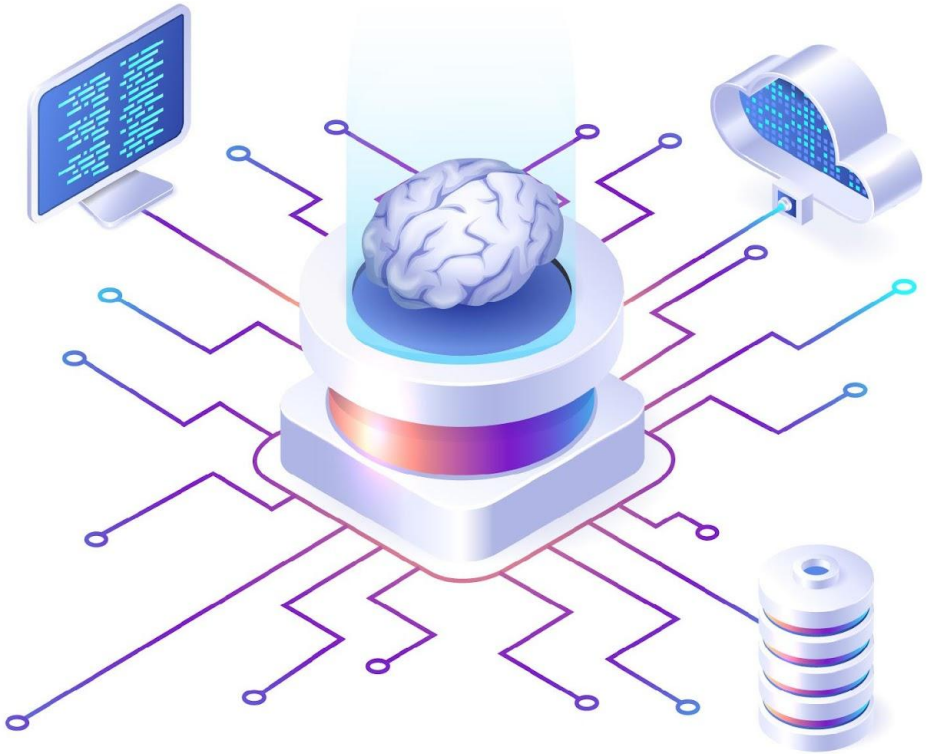
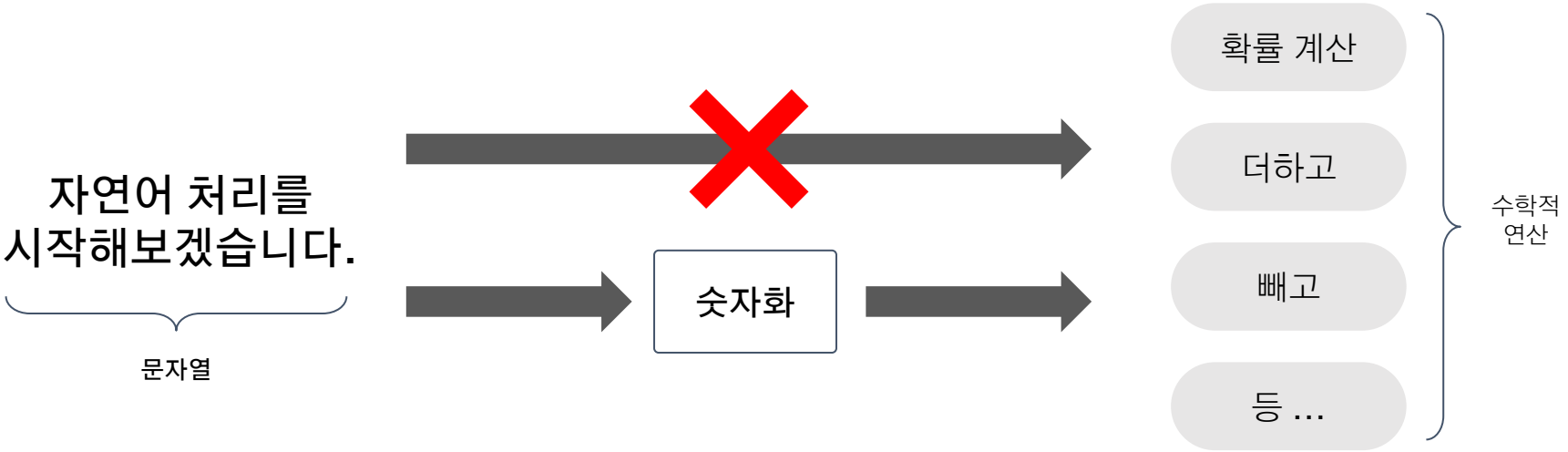
Insight campus SeSAC

표현(Representation)

실무 프로젝트형 인공지능 자연어처리



단어의 표현이 필요한 이유



단어의 표현

표현(Representation)

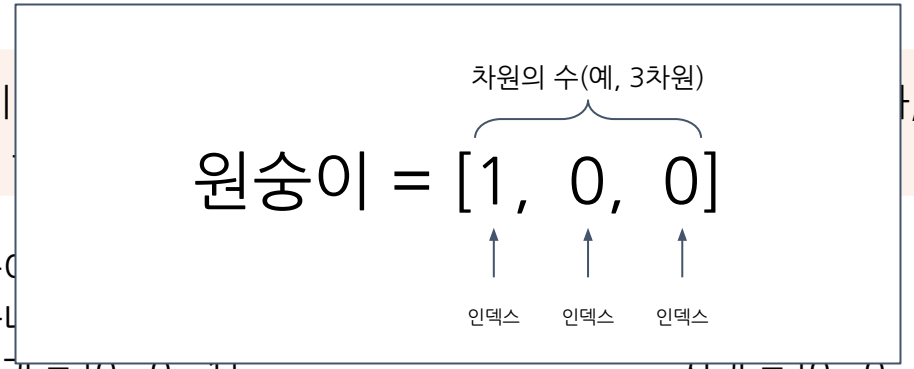
1 원핫-인코딩 (One-Hot-Encoding)

Insight campus SeSAC

원핫-인코딩(One-Hot-Encoding)

원핫-인코딩은 단어(word)를 숫자로 표현하고자 할 때 적용할 수 있는 간단한 방법론

원숭이



, 코끼리를

원숭이
바나나
사과 = [0, 0, 1]

0, 0]
0, 0]
사과 = [0, 0, 1, 0]
코끼리 = [0, 0, 0, 1]

원핫-인코딩(One-Hot-Encoding) 한계점

차원 크기의 문제

원숭이, 바나나, 사과를
표현할 때

원숭이 = [1, 0, 0]
바나나 = [0, 1, 0]
사과 = [0, 0, 1]

단어의 수만큼 차원이 필요함

단어수가 많아진다면?

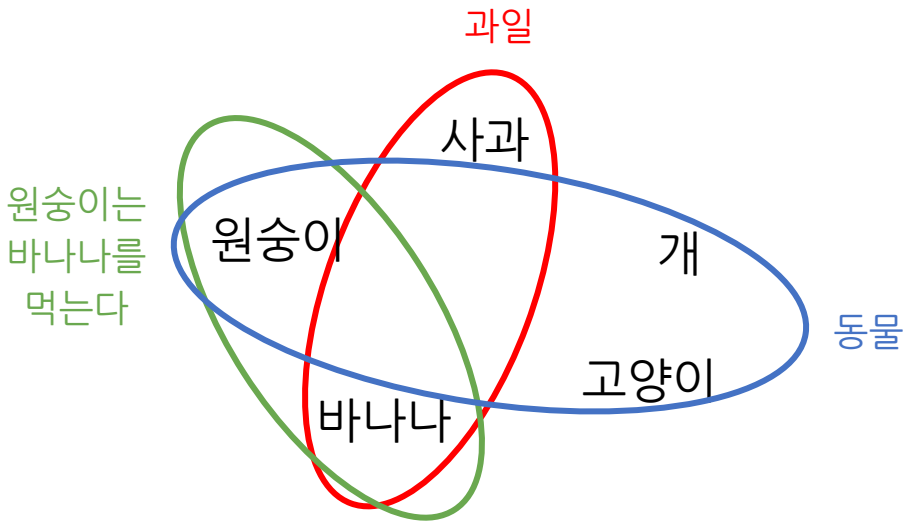
2017년 표준국어대사전에 등재된 단어 수 약 50만개
=> 50만개의 차원이 필요

원숭이 = [1, 0, 0, 0, 0, 0, 0, 0, 0, ... 0, 0, 0, 0, 0]

50만 차원 벡터

원핫-인코딩(One-Hot-Encoding) 한계점

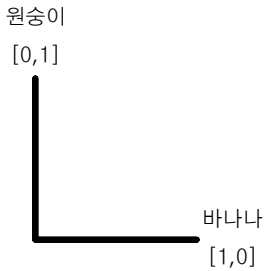
의미를 담지 못하는 문제



원핫-인코딩(One-Hot-Encoding) 한계점

의미를 담지 못하는 문제

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$



$$\text{similarity} = \frac{(1 \times 0) + (0 \times 1)}{(1^2 + 0^2) \times (0^2 + 1^2)} = 0$$

원핫-인코딩(One-Hot-Encoding) 한계점

의미를 담지 못하는 문제

원숭이, 바나나, 사과, 개, 고양이를 표현할 때

원숭이 = [1, 0, 0, 0, 0]
바나나 = [0, 1, 0, 0, 0]
사과 = [0, 0, 1, 0, 0]
개 = [0, 0, 0, 1, 0]
고양이 = [0, 0, 0, 0, 1]

- “원숭이, 사과” 코사인 유사도 : 0
- “원숭이, 바나나" 코사인 유사도 : 0
- “개, 고양이" 코사인 유사도 : 0

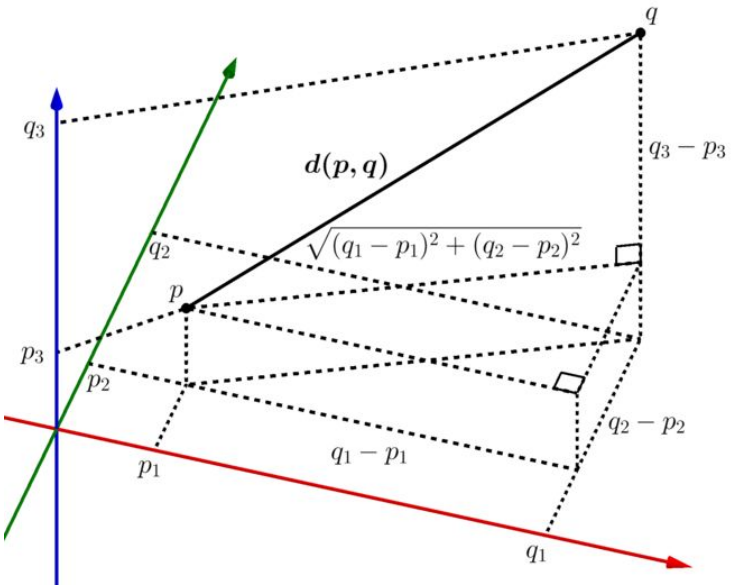
=> 원핫 벡터간 코사인 유사도는 모두 0

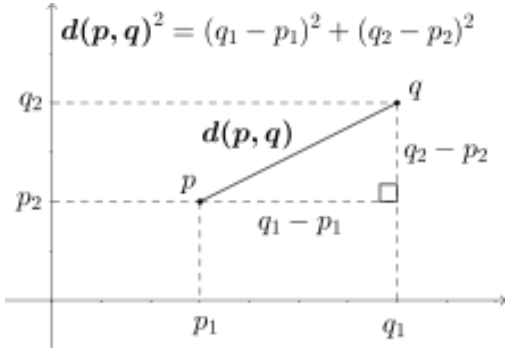
=> 따라서 의미를 분간 하기 어려움

2 유사도 계산(Similarity)

Insight campus SeSAC

유클리디언 거리(Euclidean distance)





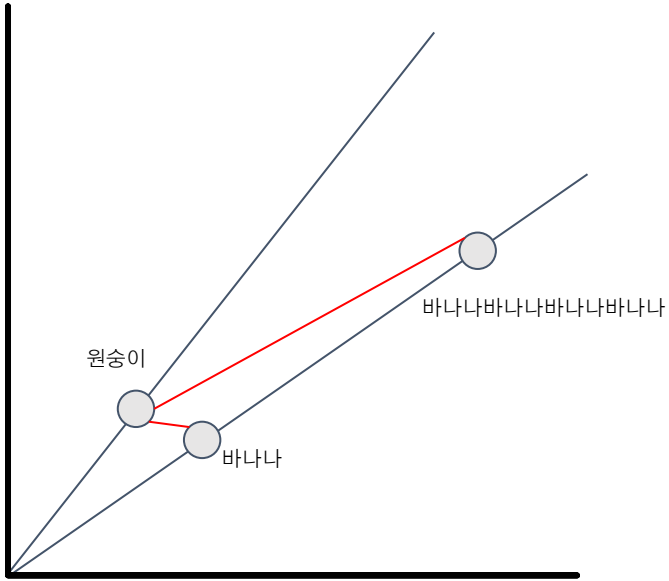
$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

https://en.wikipedia.org/wiki/Euclidean_distance

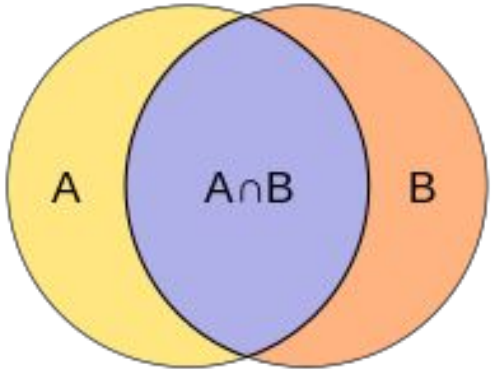
벡터간 직진거리 계산

유클리디안 거리의 한계점



Insight campus SeSAC

자카드 유사도(Jaccard index)



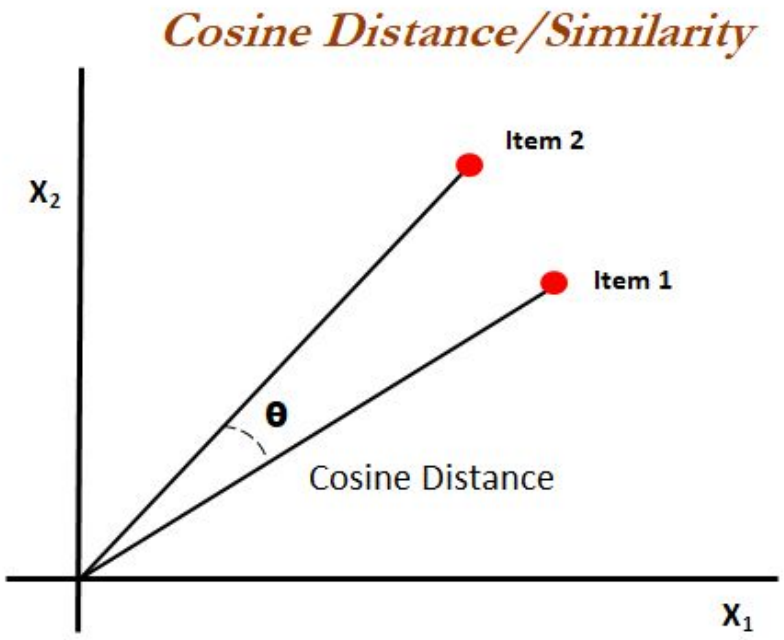
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

https://en.wikipedia.org/wiki/Jaccard_index

문서 혹은 문장간 유사도 측정 (겹치는 토큰의 비율)

Insight campus SeSAC

코사인 유사도(Cosine Similarity)



$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- 두 벡터간의 유사도를 측정하는 방법 중 하나
- 두 벡터 사이의 코사인을 측정
- 0도 = 1, 90도 = 0, 180도 = -1
  => 1에 가까울수록 유사도가 높음
  => 유사도가 높다는 것은 유사한 의미를 가짐을 의미

https://en.wikipedia.org/wiki/Cosine_similarity

두 벡터간 각(코사인 유사도)을 이용한 유사도 측정

3 단어 임베딩 (Word Embedding)

원핫-인코딩(One-Hot-Encoding) 한계점

벡터로 표현한 단어 차원이 너무 큼 → 연산이 낭비되어 모델 학습에 불리하게 적용

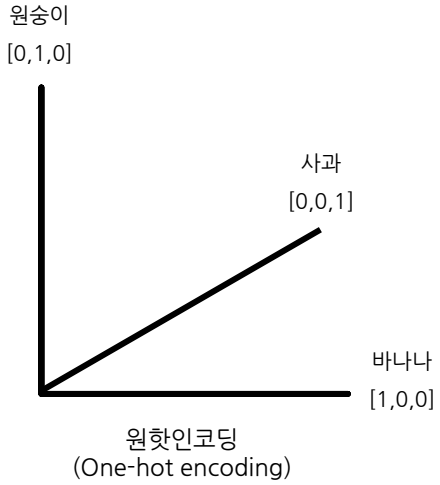
단어 의미를 담지 못함 → 분석을 효과적으로 수행할 수 없음

Insight campus SeSAC

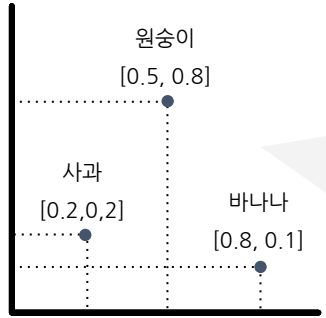
단어 임베딩(Word embedding)

단어 임베딩은 단어의 의미를 간직하는 밀집 벡터(Dense Vector)로 표현하는 방법

원숭이, 바나나, 사과를 표현할 때



원핫인코딩
(One-hot encoding)



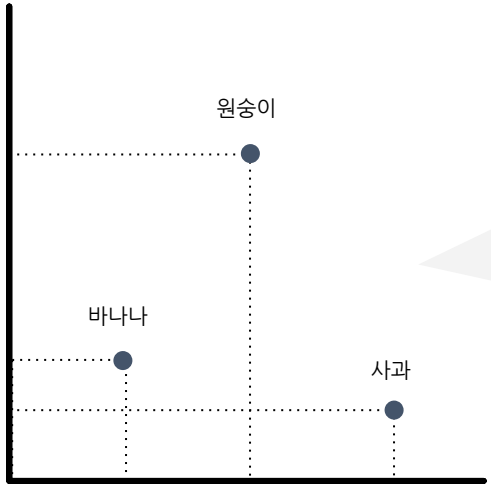
밀집 벡터
(Dense Vector)

- 벡터가 공간에 꽉차 있음
- 새로운 단어 추가시 차원을 추가할 필요가 없음
  => 차원을 줄일 수 있음
  => 추후 분류나 예측 모델을 학습할 때 연산을 줄일 수 있는 이점을 갖음

Insight campus SeSAC

단어 임베딩 (Word Embedding)의 한계

벡터로 표현한 단어 차원이 너무 큼 → 밀집 벡터(Dense vector)로 해결



사과 벡터는 어디에
표현되는 것이 맞을까요?

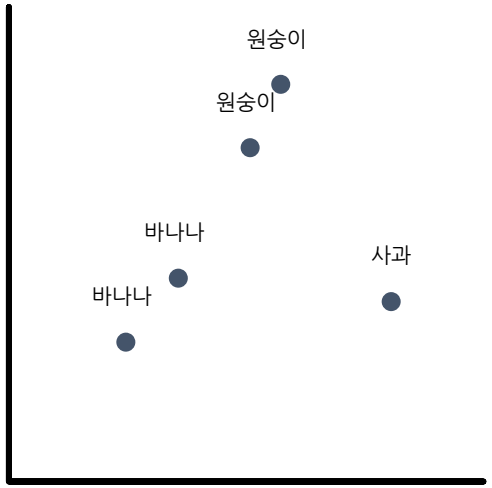
=> 단어를 벡터로
표현하는 명확한 방법이
존재하지 않음

단어 의미를 담지 못함 → ?

밀집 벡터를 만드는 방법

분포 가설이란,

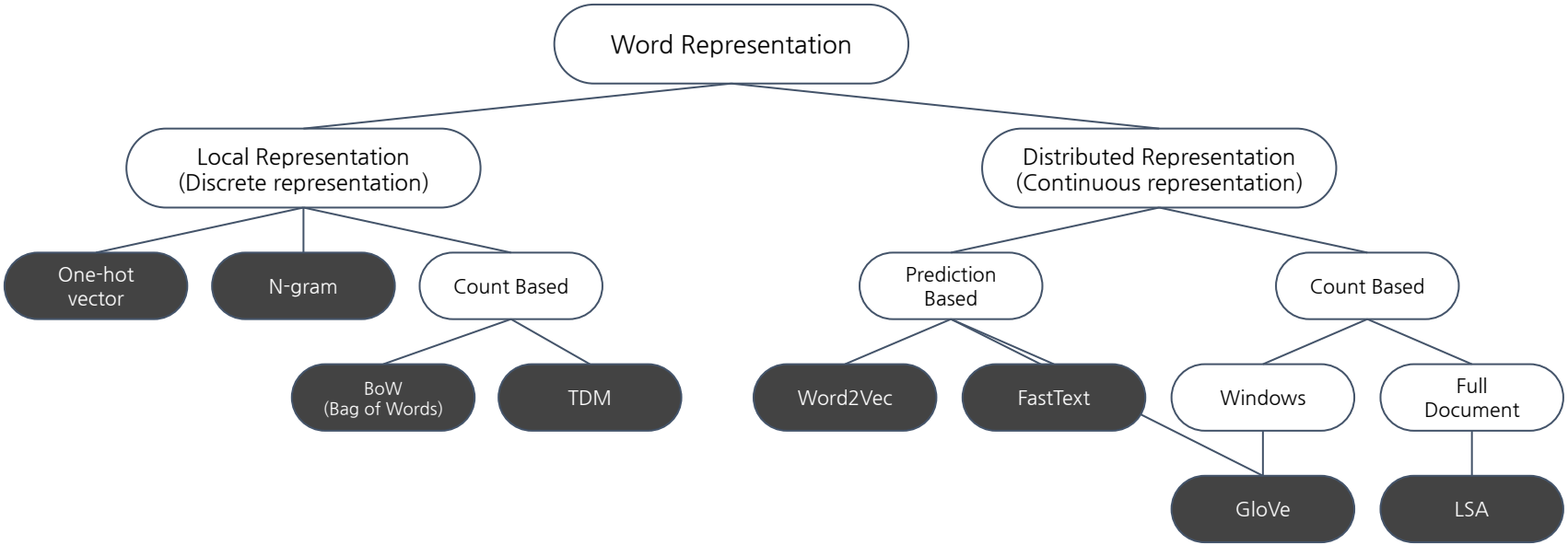
'같은 문맥에서 등장하는 단어는 유사한 의미를 지닌다'



1) 임의의 위치에 벡터 생성

2) 같은 문맥이 등장하는 단어를 더 가까이 표현

Word Representation



- Local representation (Discrete representation) : 해당 단어 그 자체만 보고 값을 매핑하여 표현
- Distributed representation (Continuous representation) : 단어를 표현하기 위해 주변을 참고

감사합니다.

Insight campus SeSAC

daniel@fins.ai