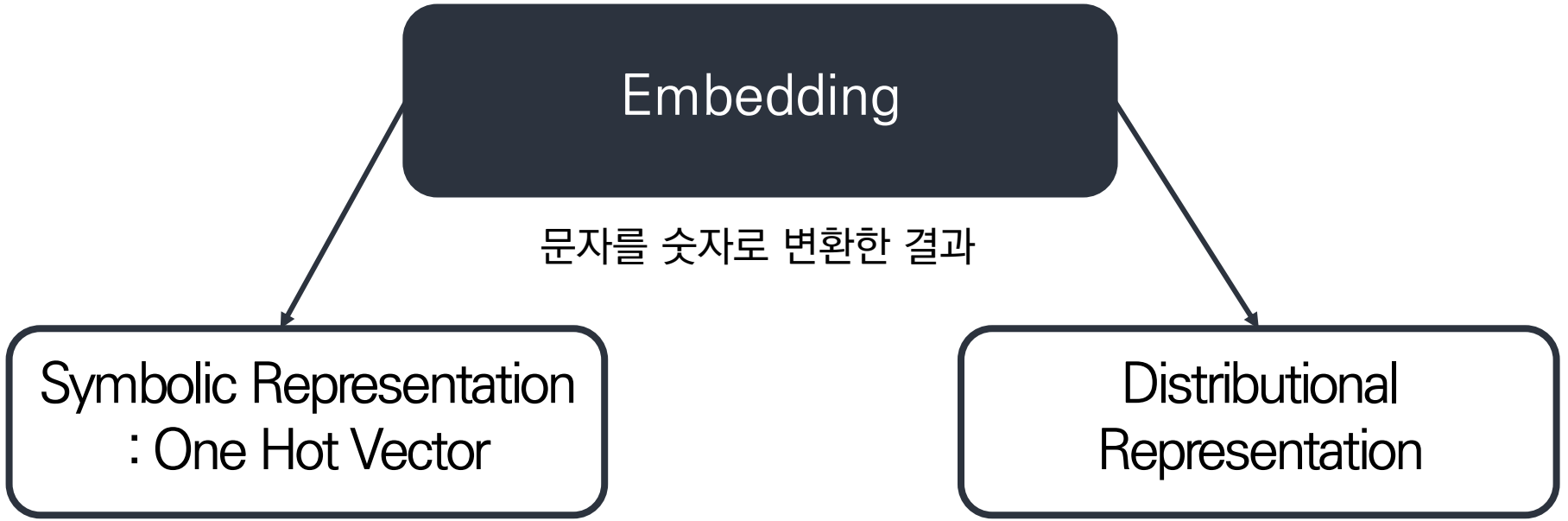
301: 자연어처리의 흐름

Topic 3: Natural Language Process

단어나 문서의 의미를
나타낼 수 있는
"표현"을 찾는다.

1. 기계가 이해할 수 있는 단어 표현
2. 통계기반의 모델
3. 언어모델의 종류
4. 전처리 방식
5. (실습) Pre-trained Model

기계가 이해할 수 있는 단어 표현방법은 무엇일까?



자연어의 표현: One-Hot Vector

위치 값을 기준으로 단어를 표현하는 방법

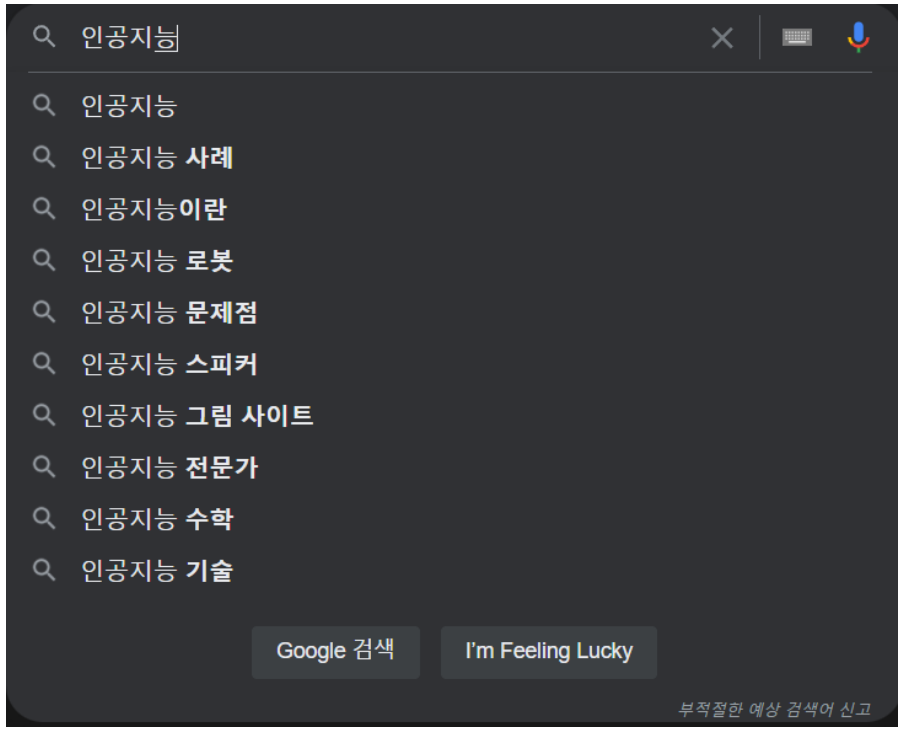
한국인	중국인	일본인
1	0	0
0	1	0
0	0	1

안	녕	하	세	요
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

‘한국인 의 One Hot Vector 표현 100

‘하’ 의 One Hot Vector 표현 00100

자연어의 표현: 유사도

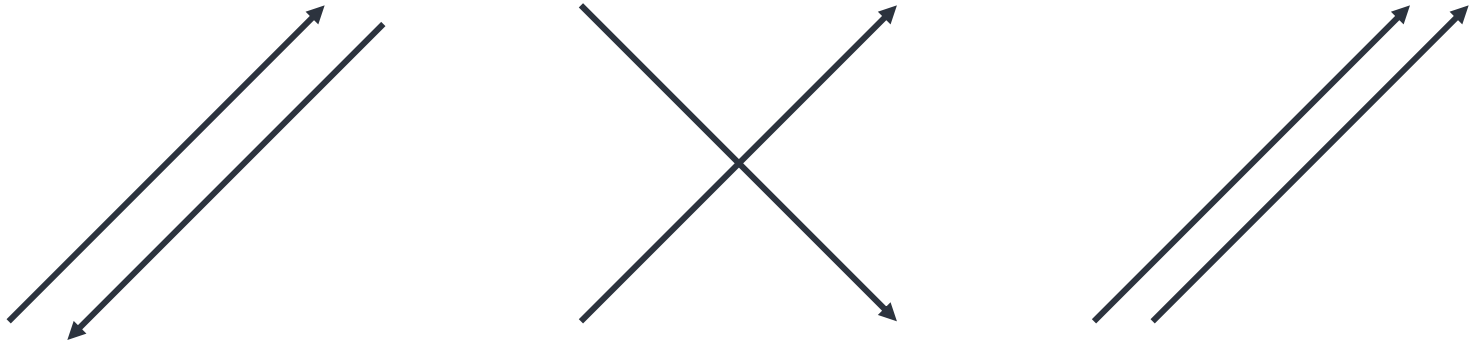


1. 유클리드거리 $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

2. 맨해튼거리 $d = \sum_{i=1}^{n}|x_i - y_i|$

3. 코사인 유사도

자연어의 표현: 유사도



코사인 유사도 -1　　코사인 유사도 0　　코사인 유사도 1

1. -1과 1 사이로 정규화 된다.
2. 벡터의 내적공식을 변형, 크기는 무시한다.
3. 벡터의 '각도 ' 만 고려한다.

자연어의 표현: One-Hot Vector의 유사도

한국인	중국인	일본인
1	0	0
0	1	0
0	0	1

```python
# import library
import numpy as np
from numpy.linalg import norm

# cosine similarity
def cosine_similarity(A, B):
        return np.dot(A, B)/(norm(A)*norm(B))

# one-hot vector
korean = np.array([1, 0, 0])
chinese = np.array([0, 1, 0,])
japanese = np.array([0, 0, 1])

# similarity
print(cosine_similarity(korean, chinese))  # 0.0
print(cosine_similarity(korean, japanese))  # 0.0
```

자연어의 표현: One-Hot Vector의 한계

One-Hot Vector	
장점	간단하게 단어를 표현할 수 있다.
단점	1. 의미를 담을 수 없다. 2. “희소행렬” Sparse Matrix이 생성된다.

희소행렬 Sparse Matrix	
특징	1. 값이 대부분 0으로 이루어진 행렬 2. 데이터가 밀집되지 않고 넓게 분산된 표현방식 (↔ 밀집행렬 Dense Matrix)
단점	1. 불필요한 0으로 인한 메모리 낭비 2. 행렬의 크기로 인한 연산 시간 증가

자연어의 표현: Distributional Semantics

단어의 의미는 주변 단어에 의해 형성된다.

분포가설 (distributional hypothesis)

Count-based Model	• 말뭉치 (corpus) 로부터 DTM을 구축하여 단어, 문서 벡터를 추출하는 과정
Prediction-based Model (Language Model)	• 단어의 의미를 스스로 학습하여 벡터에 담는 모델 • 밀집 벡터 (Dense Vector)를 만드는 것이 목표 • P(next|context)를 최대화 하는 방향으로 학습

DTM & TF-IDF

Document Term Matrix

Count-based Model

Tip
- 문장: 단어(Term)의 집합
- 문서: 문장의 집합
- 말뭉치(corpus): 문서의 집합

전제

- 분포가설 + 자주 쓰이는 단어는 문장 안에서 중요한 위치를 차지한다.

방법

1. 문서(Document)에서 쓰인 모든 단어(Term)를 구한다.
2. 각 문장을 그 단어가 쓰인 빈도를 나타내는 벡터로 변환한다.
3. 벡터들을 모아 하나의 문서를 행렬 (Matrix)으로 표현

DTM & TF-IDF

Document Term Matrix

Count-based Model

장점

1. 가설에 의하면 단어의 의미를 반영됨: 문장간 유사도를 알아낼 수 있다.
2. 자주 쓰는 단어가 기준이므로 희소행렬 문제 일부 해소

단점

- 단순 빈도만 따지는 문제
  1. 의미 없이 자주 쓰이는 단어를 중요한 단어로 인식 → 불용어 판단 불가
  2. 어순 반영 불가
  3. I love you의 DTM과 I love you I love you 의 cosine similarity는 동일(=1)

DTM & TF-IDF

Term Frequency-Inverse Document Frequency

Count-based Model

$$TFIDF(t,d) = TF(t,d) * IDF(t)$$

$$IDF(t) = \log(\frac{n}{1+DF(t)})$$

말뭉치 전체에서 차지하는 비중은 작으나
특정 문서에서 많이 사용되었다면,
해당 단어가 문서를 대표하는 단어일 것이다.

TF(term, document)

- 문서에서 쓰인 단어의 개수
- 문서에서 용어가 몇 번 사용되었는가
- DTM과 같다

IDF(Inversed DF)

- 용어가 등장한 문서는 몇 개인가
- DF : 말뭉치에서의 용어 사용 횟수
- IDF가 크면 말뭉치 전체에서 용어가 사용되는 빈도가 낮다.

DTM & TF-IDF

Term Frequency-Inverse Document Frequency

Count-based Model

장점

1. 확률적으로 문서를 대표하는 단어를 찾을 수 있다.
   (맥락을 반영하지는 않는다.)

단점

- 동음이의어를 구분할 수는 없다.
  - Rule based model은 문장과 맥락을 이해하지 못하므로 동음이의어 등을 구분할 수 없다.