# 텍스트 마이닝과 데이터 마이닝

# Part 03. 단어 임베딩과 문장 임베딩

정 정 민

# Chapter 08. 문장 임베딩, Sentence Embedding

1. 단어와 문장 임베딩의 차이점
2. 원핫 인코딩
3. 학습 기반 임베딩
4. 문장 임베딩 실습

# 문장과 단어 임베딩의 차이점

# 단어 임베딩과 문장 임베딩

- **단어 임베딩**
  - 앞선 수업에서 다룸
  - 단어를 숫자의 형태로 변환하는 과정

- **문장 임베딩**
  - 단어 임베딩과 마찬가지로
  - 문장 자체를 숫자의 형태로 변환
  - 단어를 넘어 문장 자체가 갖고 있는 의미를 벡터로 표현
  - 이를 이용해 전반적인 글의 이해, 문맥 파악, 글 생성 등 다양한 자연어 처리 작업 진행

# 왜 서로 다른 임베딩이 있을까요??

- 문장 임베딩과 단어 임베딩은 서로 다른 목적과 사용 사례를 기반으로 개발됨
- 풀어야 하는 문제를 해결하는 서로 다른 도구임

- **단어 임베딩**은
  - 단어의 의미, 문맥적 유사성, 동의어 등과 같이
  - 단어 수준에서 의미를 활용하는 경우에 사용
  - 전체적인 문장의 의미를 한번에 포착하기는 어려움

- **문장 임베딩**은
  - 전반적인 글의 이해, 문맥 파악, 글 생성 등과 같이
  - 문장 혹은 그 이상의 단위에서 정보를 포착하는 경우에 사용
  - 글의 전반적인 이해가 쉽게 가능하지만
  - 보다 많은 자원과 계산이 필요하며, 단어 수준의 미묘한 변화를 잡아내기가 어려운 단점이 있음

# 원핫 인코딩

# 문장에 원핫 인코딩 적용하기

- 단어 원핫 인코딩과 마찬가지로
  - 사용하는 전체 단어의 고유 집합 확보
  - 각 단어에 독립된 인덱싱 진행

- 문장에 소속된 각 단어를 해당 단어의 인덱스 위치에 1을 부여
- 나머지 부분을 0으로 채움

- 예를 들어,
  - 사과는 = 0, 바나나는 = 1, 맛있다 = 2
  - 사과는 맛있다 → [1, 0, 1]
  - 바나나는 맛있다 → [0, 1, 1]

# 다양한 문장 임베딩 기법

# 단어 임베딩을 활용한 문장 임베딩

- 문장은 단어를 기반으로 만들어지는 개념
- 따라서 각 단어들의 임베딩을 이용해 문장의 임베딩을 생성

- 각 단어의 임베딩의 평균을 활용
  - 가장 직관적인 방법
  - 하지만 다른 의미의 문장이 서로 비슷한(혹은 같은) 임베딩 값을 갖을 수 있음
    - “고양이가 강아지를 쫓는다” 와 “강아지가 고양이를 쫓는다”는 안전 다른 의미 But 평균 임베딩은 동일

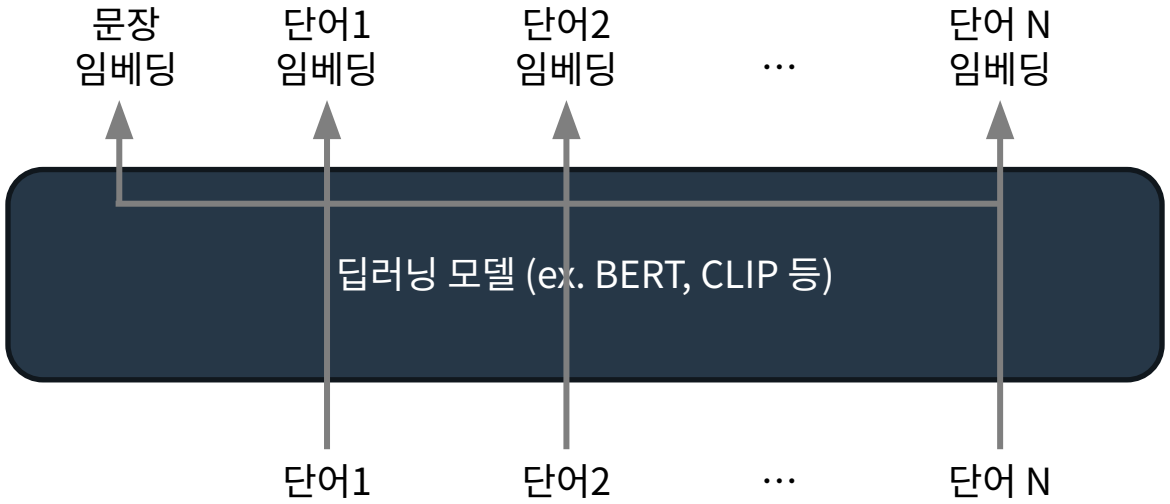
- TF-IDF 를 활용한 단어 가중치를 적용해 문장 임베딩을 생성
  - TF-IDF란 문장 내 단어의 중요도를 나타내는 척도
  - 이 값을 이용해 각 단어 임베딩에 가중
  - 가중된 값들을 활용해 평균 값 활용

# 딥러닝을 활용한 학습 기반 문장 임베딩

- 문장 임베딩은 단어 임베딩보다 고차원적인 연산이 필요함
- 따라서 딥러닝 기법을 활용하는 과정에서 많이 발전이 됨

- 딥러닝 모델이 단어를 임베딩하는 과정에서
- 전체 문장의 의미를 담는 벡터를 생성

- 문장을 구성하는 각 단어에서 정보를 공급받아 임베딩 벡터를 생성



# 문장 임베딩 실습



강의 실습 코드 : 링크

# 원핫 인코딩 문장 적용 실습

- CountVectorizer 활용
  - 각 문장에 나온 단어를 독립적인 인덱스로 바꿔주는 과정과
  - 또한, 해당 인덱스에 1의 값을 넣어주는 과정 지원

- binary=True 라는 값을 넣어주면 중복된 단어가 나와도 1로 표현
  - 만약 Flase라면 BoW의 형태의 코드가 됨

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(binary=True)
vectorizer.fit(sentences)

vectorizer.vocabulary_
# 바나나 : 0, 사과는 : 1, 맛있다 : 2

vectorizer.transform("사과는 맛있다").toarray()
# [0, 1, 1]
```

# E.O.D