텍스트 마이닝과 데이터 마이닝

Part 03. 단어 임베딩과 문장 임베딩

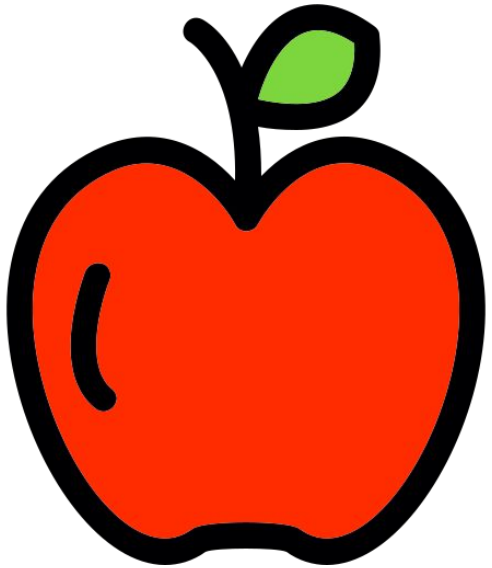
정 정 민

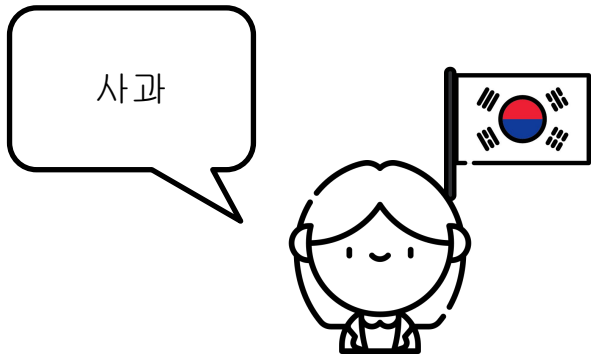
Chapter 06. 임베딩, Embedding

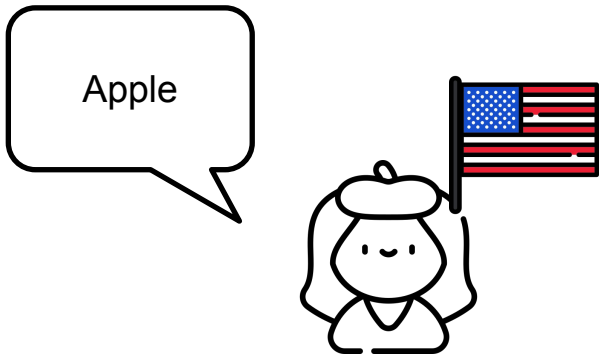
1. 컴퓨터는 '사과'를 알까?
2. 임베딩이란?
3. 임베딩의 발전 과정

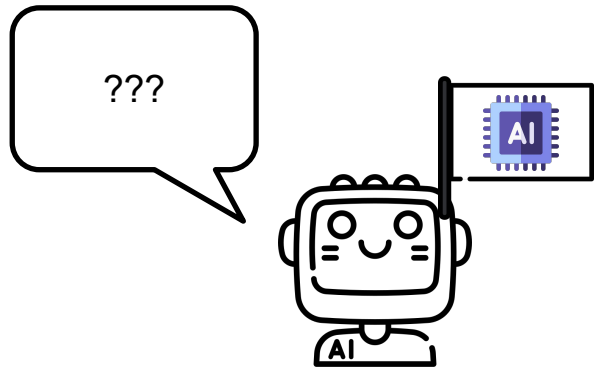
컴퓨터는 '사과'를 알까?

이걸 뭐라고 불러야 할까요?









왜 사과가 '사과'일까?

- 언어는 특정 개념을 표현하기 위한 약속의 집합

- 사과가 '사과'인 이유는 한국인들이 그렇게 정했기 때문!
- 또한 영어권자들이 'apple' 이라고 하는 이유는 그들이 그렇게 하자고 했기 때문

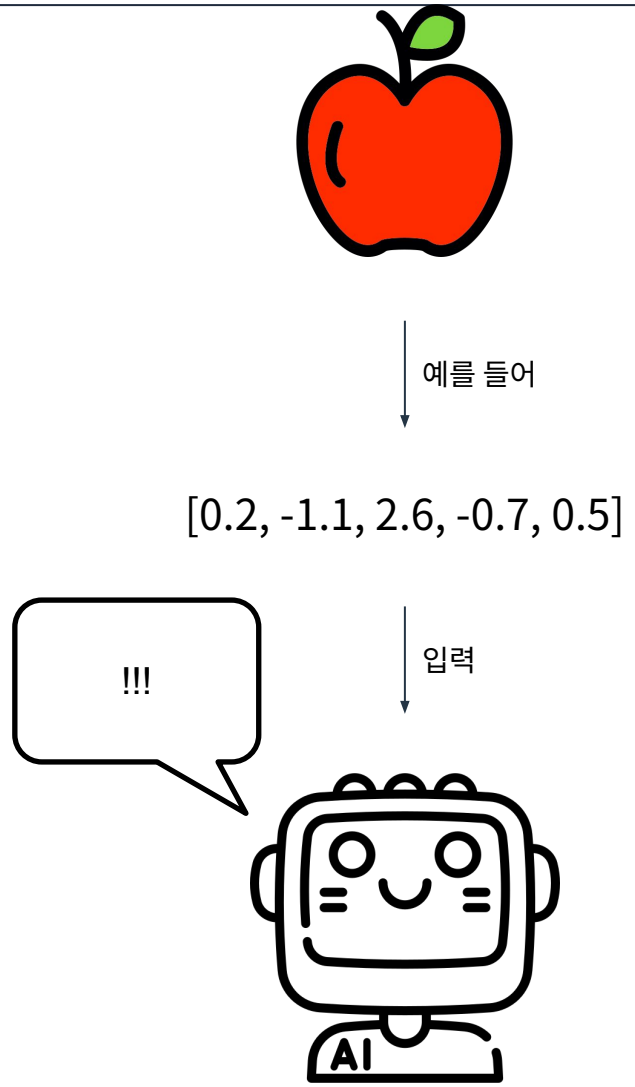
- 그럼 우리는 컴퓨터에게 사과를 어떻게 알려줄까요?
  - 편하게 '사과'라고 알려줄까요?
  - 아니면 'apple'??
  - 그것도 아니면 또 다른 나라 언어로?

단어를 숫자로!

- 글자를 컴퓨터에 입력으로 넣어주기 위해서는
- 컴퓨터가 이해할 수 있는 형태로 변경해야 함

- 컴퓨터는 특정 단어를 숫자의 형태로 받아들임
- 따라서 단어를 숫자의 형태로 변형하는 과정이 필요
  - 이 과정을 임베딩이라고 함

- 특정한 단어는 정해진 숫자들의 집합(벡터)으로 대체
- 이 숫자 집합은 컴퓨터로부터 특정 단어로 인식됨



임베딩이란?

임베딩, Embedding

- 정의
  - 텍스트 데이터를 벡터로 변환하는 기술
  - 이는 텍스트 마이닝을 비롯해 자연어 처리에서 매우 기본적인 과정
  - 글에서 유용한 정보를 추출해 분석하는 과정에서 컴퓨터가 이를 처리할 수 있도록 함

- 과정
  - 데이터 준비 : 원문 데이터 혹은 다른 소스로부터 데이터를 수집
  - 전처리 : 불용어, 오타 등의 데이터를 제외
  - 임베딩 : 목적에 맞는 임베딩 알고리즘을 적용
  - 시각화 : 필요시 임베딩 결과를 그려보고 이를 확인

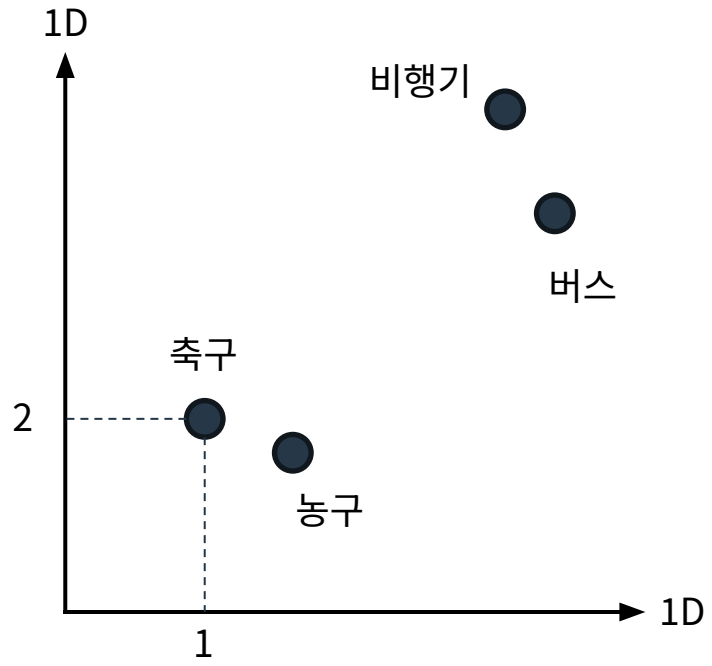
- 종류
  - 단어 임베딩 : 하나의 단어를 벡터로 변환
  - 문장 임베딩 : 문장 자체를 벡터로 변환

임베딩과 벡터 공간

- 임베딩의 결과는 벡터이므로,
- 벡터가 존재하는 벡터 공간에 표현 가능
  - 마치 [1, 2] 라는 벡터는 2차원 공간에서 표시될 수 있듯!

- 만약, 단어들을 2차원 벡터로 표현하게 되면
- 단어들을 평면 위에 표시할 수 있음
  - 좋은 임베딩이 이루어지면
  - 비슷한 의미의 단어는 비슷한 공간에 존재

- 하지만 일반적으로 단어는 고차원으로 표현됨
  - 그래야 다양한 종류의 단어를 포괄하고,
  - 단어의 의미를 담은 벡터를 만들 수 있음



임베딩의 발전 과정

원핫 인코딩 (One-hot encoding)

- 정의
  - 임베딩의 한 방법으로
  - 0과 1을 통해 단어를 정의
  - 특정 단어를 표현하는 위치만 1이고 나머지의 위치는 0으로 구성

- 예를 들어, 전체 단어가 ‘dog’, ‘cat’, ‘apple’ 이 있다고 할 때,
  - dog : [1, 0, 0]
  - cat : [0, 1, 0]
  - apple : [0, 0, 1]
- 과 같이 단어의 수 만큼의 크기를 갖는 벡터가 생성

- 각 단어의 위치(1이 표현되는 위치)는 설정하기에 따라 다름

- 직관적으로 쉽게 단어를 벡터로 변환 가능

인코딩(Encoding)과 임베딩(Embedding)

- 인코딩과 임베딩 모두 데이터를 새로운 형태로 변환하는 과정을 의미

- **인코딩**이란,
  - 데이터를 표준화된 형식으로 변환 목적
    - 이미 변환 과정은 정의가 되어있고, 그 정의에 맞춰 변환 진행
    - 그래서 원핫 인코딩에 인코딩이라는 이름이 붙었겠죠?
  - 본질적으로 의미가 변하지 않음
  - 압축과 비슷한 과정으로 저장과 데이터 전송에 용이
  - 데이터의 형식을 변환하는데 중점!

- **임베딩**이란,
  - 머신 러닝 모델이 처리하기 쉬운 형태로 변환 목적
  - 데이터의 의미적, 문맥적 특성을 모델이 이해할 수 있는 형태로 변환
  - 이를 이용해 머신 러닝 모델은 데이터를 더욱 쉽게 분석하고 처리

원핫 인코딩의 한계

- **차원의 저주**
  - 하나의 단어를 표현하는 벡터의 크기는 전체 단어 수와 같음
  - 즉, 매우 큰 차원을 갖을 수 있음
  - 차원이 크면 효율성과 계산 복잡도가 증가함

- **의미 부재**
  - 의미적으로 비슷한 단어끼리 비슷한 공간에 존재하지 않음
  - 축구 ↔ 농구의 관계와 축구 ↔ 비행기의 관계의 차이가 없음

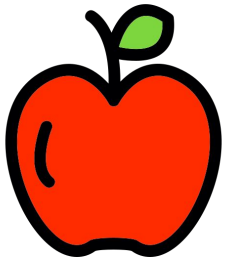
- **정보의 희소성**
  - 특정 단어의 위치만 1이므로
  - 중요한 정보가 매우 매우 희소함
  - 대부분 0이 많음!

분산 표현 (Distribution Representation)

- 원핫 인코딩의 한계를 극복하기 위해 제안된 개념

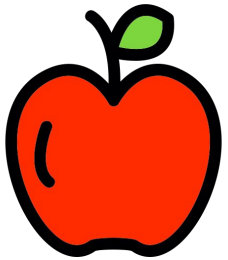
- 분산 표현이란, 정수가 아닌 실수(연속적인 값)로 이루어진 벡터로 임베딩 진행
  - 원핫 인코딩은 0과 1 두 정수 값으로 임베딩을 진행

- 연속적인 실수의 값으로 단어를 변경하면서
- 데이터의 의미를 여러 특성(feature)에 걸쳐 분산시켜 표현
  - 단어나 개체의 의미가 하나의 차원 혹은 공간에 집중되어 표현되는 것이 아니라, (마치 하나의 1 처럼)
  - 여러 요소에 걸쳐 분산되어 표현됨
  - 따라서 다양한 의미 & 문맥적 특성을 풍부하게 포착 가능



→ [0, 0, 1, 0, 0]

원핫 인코딩



→ [0.2, -1.1, 2.6, -0.7, 0.5]

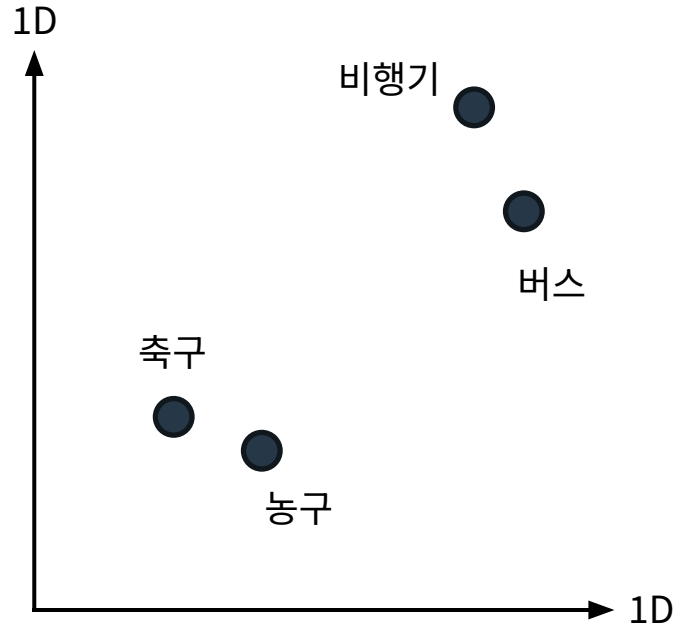
분산 표현

좋은 임베딩의 시각적 효과

- 분산 표현으로 특정 단어가 잘 임베딩이 된다면,
- 비슷한 의미를 갖는 단어들은 비슷한 분포를 갖게 됨

- 즉, 이를 벡터 공간에 표현하면 비슷한 공간에 표현될 수 있음
  - 농구는 의미적으로 축구와 비슷
  - 비행기는 농구와 축구와는 의미적으로 비슷하지 않음
  - 따라서 벡터 공간에 멀리 위치

- 또한, 임베딩 벡터도 수의 집합이므로 의미 차원에서 연산이 가능!
  - 예를 들어,
    - 왕 – 남자 + 여자 = 여왕
    - 서울 – 대한민국 + 일본 = 도쿄
    - 등등



E.O.D