Natural Language Processing

자연어 처리: 단어의 표현

강사 양석환

단어의 의미와 유사성, 모호성

단어의 의미와 유사성, 모호성

- 자연어 처리 분야에서 가장 기초이면서 가장 어려운 문제
- 단어와 단어의 의미와의 관계
  - 단어는 글자로 적을 때는 하나의 형태를 가지지만
    상황에 따라 다른 의미로 사용됨
  - 주변 정보에 따라 숨겨진 의미를 파악, 이해함
  - 주변 정보의 부족, 또는 다른 해석에 따라 모호성 증가
    (사람도 제대로 이해하지 못하는 경우가 있음)

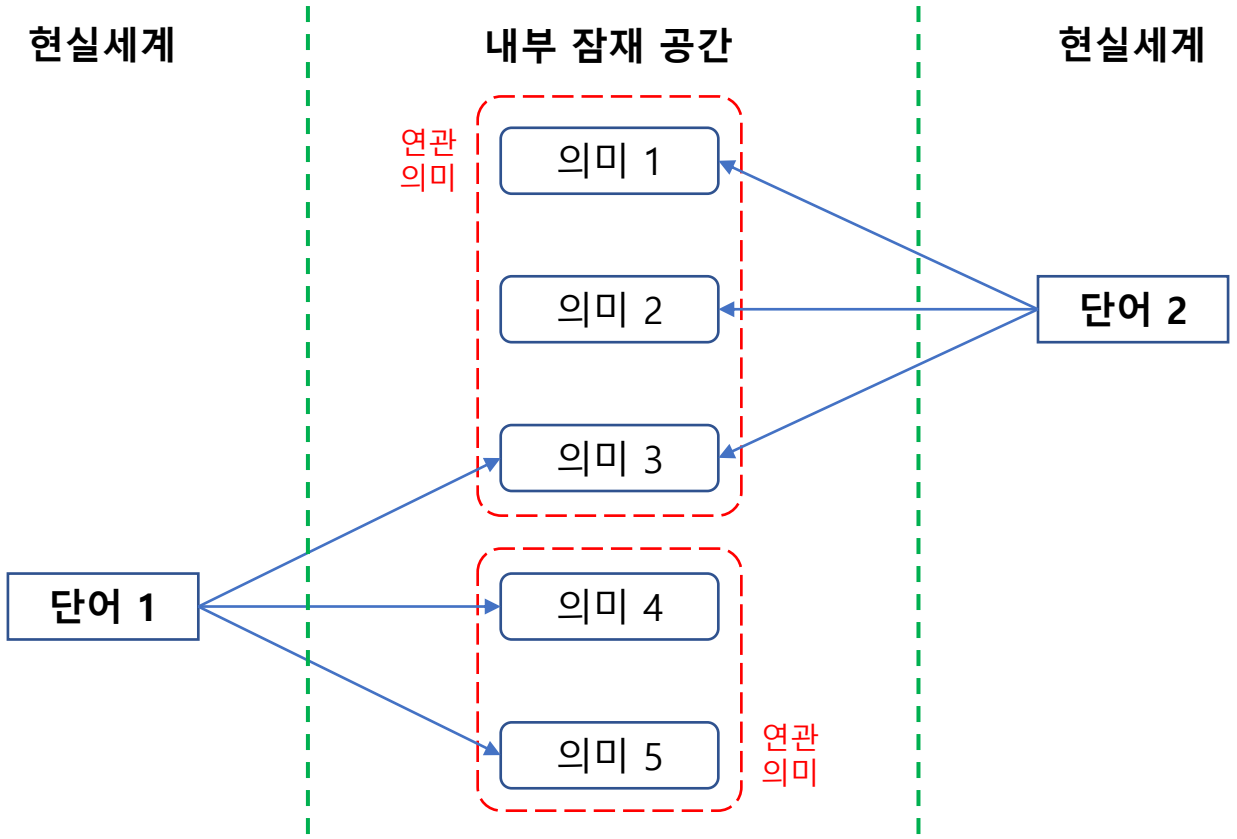
단어의 의미와 유사성, 모호성

- **단어의 중의성에 대한 예**

연관 의미	세부 의미	풀이
차 #1	차 1-1	좋은 향기나 맛이 있는 식물의 잎이나 뿌리, 열매 등을 달이거나 우려서 만든 마실 것
차 #2	차 2-1	바퀴가 달려 있어 사람이나 짐을 실어 나르는 기관
	차 2-2	사람이나 물건을 차에 실어 그 분량을 세는 단위
	차 2-3	장기의 말 중에서 '車' 자를 새긴 말
차 #2	차 3-1	둘 이상을 비교했을 때 서로 다르게 나타나는 수준이나 정도
	차 3-2	어떤 수나 식에서 다른 수나 식을 뺀 나머지
차 #4	차 4-1	어떤 일의 차례나 횟수를 나타내는 말
	차 4-2	어떠한 일을 하던 기회나 순간
	차 4-3	일정한 주기나 기간이 지난 해당 시기를 나타내는 말

단어의 의미와 유사성, 모호성

- **단어의 형태들과 내부 의미들이 갖는 관계**



단어는 그 형태를 공유하더라도
서로 다른 뜻을 가진 의미로 구성될 수 있다.

사람은 머릿속으로는 내부 잠재 공간의 의미를 받아들이지만
실제로 사용할 때는
현실세계의 단어를 매개체로 사용하여 의미를 전달한다.

한 가지 형태의 단어에 여러 의미가 포함되어 생기는
"중의성 문제"는 자연어 처리에 있어서 매우 큰 비중을 차지하며

특히 기계번역에서는 단어의 의미에 따라서
해당 번역 단어의 형태가 완전히 바뀌기 때문에 매우 중요하다.

단어의 의미와 유사성, 모호성

- 동형어, 다의어, 동의어
  - 동형어: 형태는 같으나 뜻이 서로 다른 단어
    - 어원이 서로 다른 의미들이 같은 형태를 띠는 단어
  - 다의어: 한 형태의 단어가 여러 의미를 지니는 단어
    - 각 의미들이 서로 관련된 뜻을 가진다는 점에서 동형어와 차이가 있음
  - 동의어: 같은 의미를 가지는 다른 형태의 단어

종류	단어 형태	의미 1	의미 2
동형어	차	마시는 차(茶, tea)	달리는 차(車, car)
다의어	다리	사람 다리(脚, leg)	책상 다리(跏, desk leg)

책상다리할 가

- 자연어 처리를 위하여 단어의 중의성을 제거하는 작업이 필요함

단어의 의미와 유사성, 모호성

- **상위어와 하위어**
  - **사람이 사용하는 단어는 하나의 추상적 개념을 나타냄**
  - **특정 개념을 하위 개념이라고 하면 그 하위 개념들을 포함하는 상위 개념이 있음**
    - 상위 개념을 가리키는 단어: 상위어
    - 하위 개념을 가리키는 단어: 하위어
  - **단어들의 어휘 분류에 따라 단어 간 관계구조의 계층화 가능 → 자연어 처리에 유용**

상위어	하위어
동물	포유류
포유류	코끼리
코끼리	아프리카 코끼리
사물	전화기
전화기	핸드폰
사물	컴퓨터
컴퓨터	노트북

단어의 의미와 유사성, 모호성

- 지금까지의 내용을 기반으로 보면
  - 자연어 처리를 위하여 단어의 중의성을 제거하는 작업이 필요하다

    → 중의성의 제거를 위해 각 단어 별로 명확한 값을 부여할 필요가 있다
  - 종종 각 단어의 관계구조에 따른 계층화 작업이 요구된다
  - 단어들은 의미, 단어 간의 관계, 분류체계 등에 기반한 유사도를 가진다

- 이상의 내용에 따라...
  - 자연어 처리를 위해서는 각 단어에 정확한 값을 부여하는 작업이 요구된다

임베딩

임베딩

- 임베딩이란?
  - 단어나 문장을 수치화 하여 벡터공간으로 표현하는 과정
  - 왜 임베딩 과정을 필요로 하는가?
    - 컴퓨터는 자연어를 직접적으로 처리할 수 없다
    - 컴퓨터는 수치 연산만 가능하다
    - 따라서 자연어를 숫자나 벡터 형태로 변환할 필요가 있다
    - 이때 사용되는 일련의 과정을 “임베딩”이라고 한다
  - 임베딩된 결과는 딥러닝 모델의 입력값으로 사용된다

임베딩

- 임베딩 기법의 종류
  - 문장 임베딩
    - 문장 전체를 벡터로 표현하는 방법
    - 전체 문장의 흐름을 파악해 벡터로 변환 → 문맥적 의미를 지니는 장점을 가짐
    - 단어 임베딩에 비해 품질이 좋으며 상용 시스템에 많이 사용됨
    - 학습을 위해서 수 많은 문장 데이터가 필요하며 학습 비용이 매우 높다

단어 임베딩

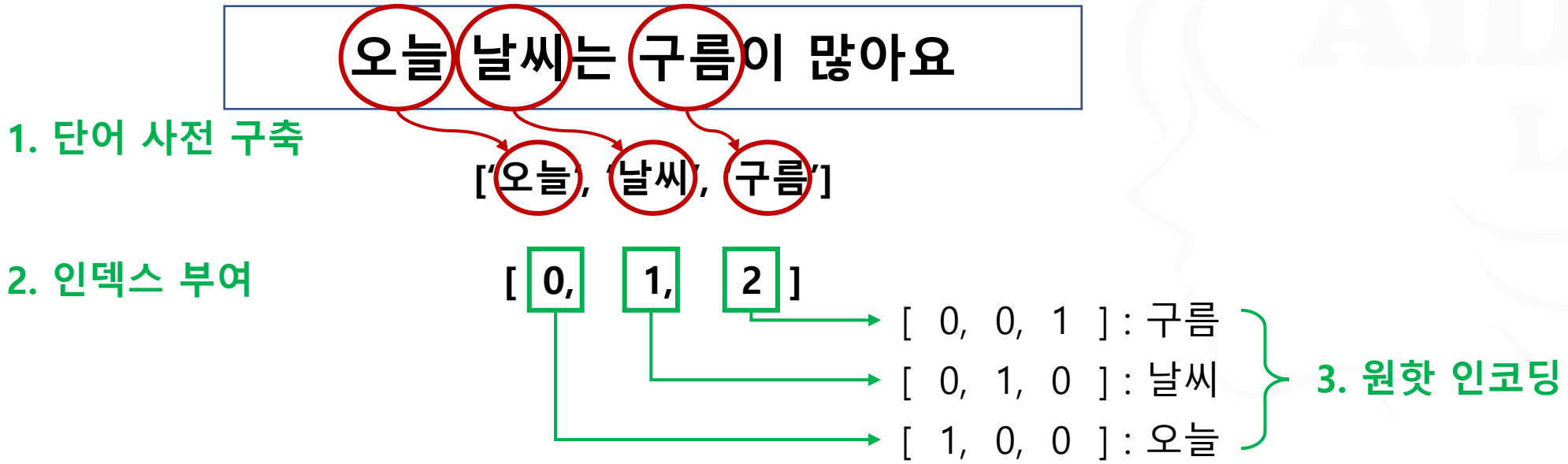
- **단어 임베딩이란?**
  - 개별 단어를 벡터로 표현하는 방법
  - 동음어 구분을 하지 않음 → 의미가 달라도 단어의 형태가 같으면 동일한 벡터값을 가짐 (단점)
  - 문장 임베딩에 비해 학습 방법이 간단함 → 성능은 떨어지지만 그래도 실무에 많이 사용

단어 임베딩

- 단어 임베딩
  - 말뭉치에서 각각의 단어를 벡터로 변환
  - 의미와 문법적 정보를 지님
  - 단어를 표현하는 방법에 따라 다양한 모델 존재
  - 토크나이징을 통해 문장에서 토큰 단위를 추출하는 경우, 추출된 토큰은 형태소 기반이므로 단어 임베딩이 효과적

단어 임베딩

- 원핫 인코딩(One-Hot Encoding)
  - 단어를 숫자 벡터로 변환하는 가장 기본적인 방법
  - 요소들 중 단 하나의 값만이 1이고 나머지 요소들의 값은 0인 인코딩 방식



단어 임베딩

- **원핫 인코딩의 특징과 단점**
  - **원핫 인코딩 벡터의 차원은 전체 어휘의 개수 → 매우 큰 차원이 됨**
    - 예: 총 20,567 종의 동물 이름의 경우

단어	사전 내 순서(index)	원핫 벡터
…	…	
강아지	8	0,0,0,0,0,0,0,1,0,0,0, … ,0
고양이	9	0,0,0,0,0,0,0,0,1,0,0, … ,0
개	10	0,0,0,0,0,0,0,0,0,1,0, … ,0
…	…	
기린	20,567	0, …,0,0,0,0,0,0,0,0,0,0,1

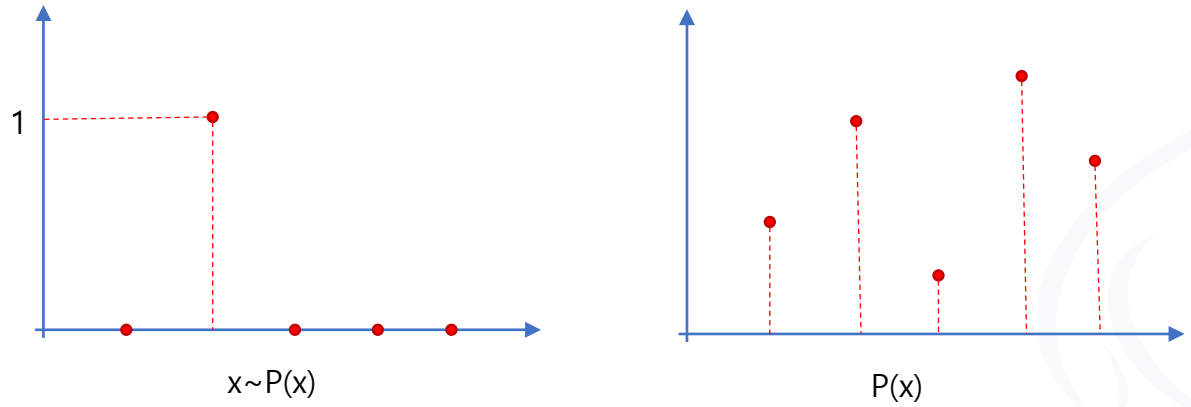
엄청난 메모리 낭비를 초래

강아지와 개는 유사한 단어이지만 처리할 방법이 없음

유사도 계산 시 결과가 0

단어 임베딩

- **단어는 불연속적인 심볼이며 이산 확률 변수로 나타남 → 원핫 벡터는 이산 확률 분포에서 추출한 샘플 → 불연속적인 값을 가짐**



이산 확률 분포로부터 샘플링하여 얻어지는 원핫 벡터

유사도를 계산하기 위하여 벡터간 연산을 할 때, 결과가 0 이 됨

단어 임베딩

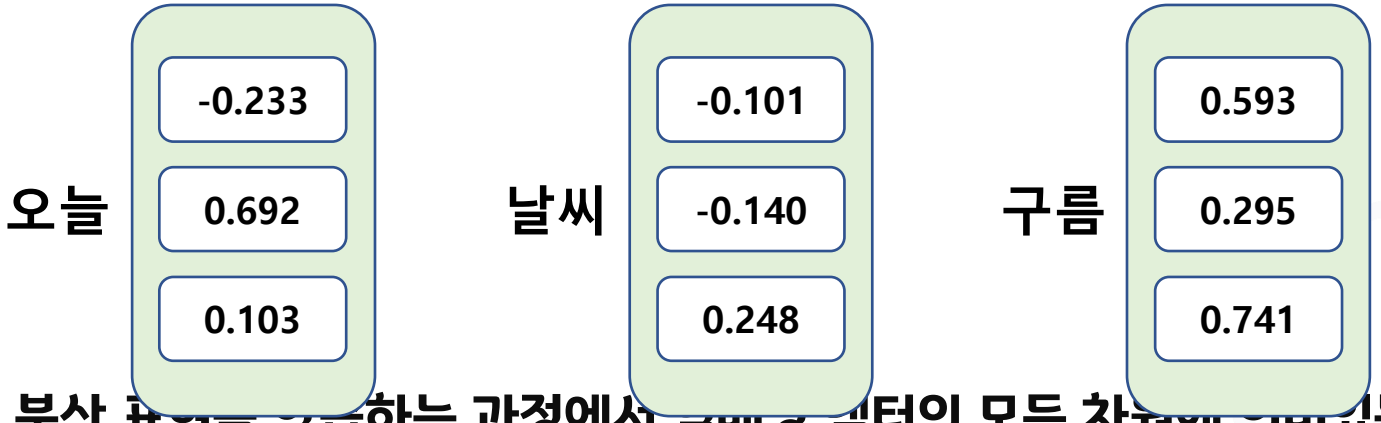
- **희소 표현**
  - **원-핫 인코딩: 표현하고자 하는 단어의 인덱스 요소만 1이고 나머지 요소는 모두 0으로 표현되는 희소 벡터(또는 희소 행렬) → 단어가 희소 벡터로 표현되는 방식: 희소 표현**
  - **희소 표현**
    - **각각의 차원이 독립적인 정보를 지님**
      - 장점: 사람이 이해하기에 직관적이다
      - 단점
        - 단어 사전의 크기가 커질수록 메모리 낭비와 계산 복잡도가 커진다
        - 단어 간의 연관성이 전혀 없어 의미를 담을 수 없다
    - **자연어 처리를 잘 하려면 기본 토큰이 되는 단어의 의미와 주변 단어 간의 관계가 단어 임베딩에 표현되어야 함 → 희소 표현은 이런 조건을 만족하지 못함**

단어 임베딩

- **분산 표현**
  - **희소 표현의 단점 해결을 위해 고안됨**
  - **각 단어 간의 유사성을 잘 표현하면서도 벡터 공간을 절약할 수 있는 방법**
  - **한 단어의 정보가 특정 차원에 표현되지 않고 여러 차원에 분산되어 표현됨**
  - **비유를 통한 예시**
    - **색상을 표시하는 RGB 모델 → 3차원 벡터, 분산 표현 방식**
    - **연두색**
      - 희소 표현으로 나타낸다면 → 매우 큰 벡터 차원. 다른 색상과의 유사성 파악 불가능
      - 분산 표현으로 나타낸다면 → RGB(204, 255, 204)
  - **희소 표현에 비해 많은 장점을 가지므로 단어 임베딩 기법에서 많이 사용됨**

단어 임베딩

- 분산 표현 방식을 그림으로 이해하면



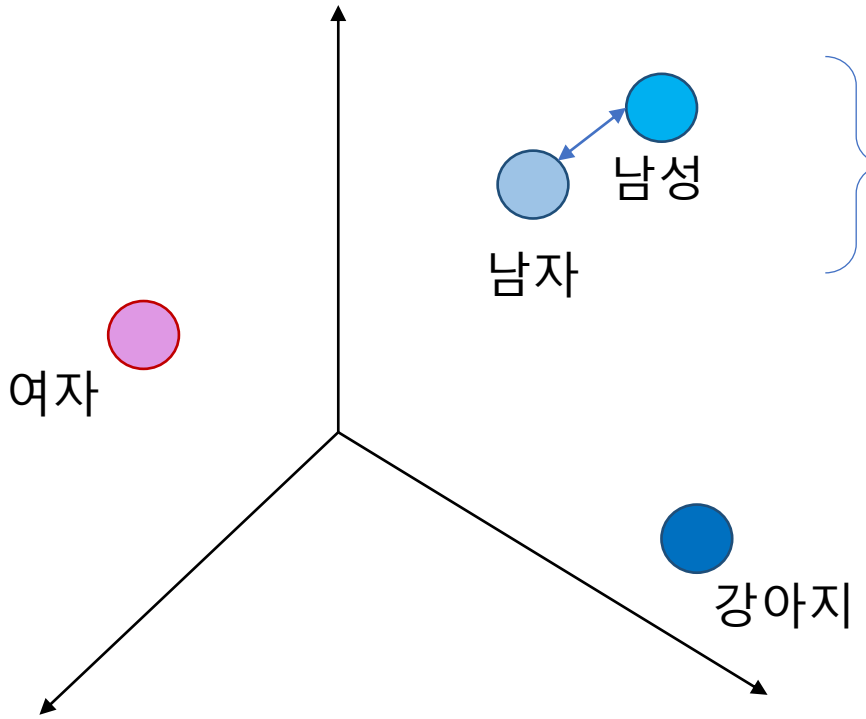
- 신경망에서는 분산 표현을 학습하는 과정에서 임베딩 벡터의 모든 차원에 의미있는 데이터를 고르게 밀집시킴 → 희소표현과 반대로 데이터 손실을 최소화 하면서 벡터 차원이 압축되는 효과를 가짐

단어 임베딩

- **희소 표현 대신 분산 표현을 사용한다면**
  - 임베딩 벡터의 차원을 데이터 손실을 최소화하면서 압축할 수 있다
  - 임베딩 벡터에 단어의 의미, 주변 단어와의 관계 등 많은 정보가 내포되어 있다 → 일반화 능력이 뛰어남

단어 임베딩

- **분산 표현 방식의 벡터 공간**



- 희소 표현 방식에서는 단 하나의 요소 값에 불과

- 분산 표현 방식에서는 둘 사이가 매우 가까움
  → 두 단어 사이의 거리를 계산하여 "남자", "남성"을 같은 의미로 해석할 수 있음

단어의 의미 파악

단어의 의미 파악

- 단어의 의미를 파악하기 위한 기법들
  - 시소러스 활용 기법
  - 통계 기반 기법
  - 추론 기반 기법

단어의 의미 파악

시소러스 활용 기법

단어의 의미 파악

- **시소러스 활용 기법**
  - **사람의 경우**
    - 단어의 이해를 위하여 "**사전**" 활용 → **사전에 단어의 의미를 정의**

      → 컴퓨터도 이렇게 할 수 있지 않을까? 라는 것이 아이디어
  - **시소러스: 유의어 사전, 어휘 분류 사전**
    - 뜻이 같은 단어(동의어), 뜻이 비슷한 단어(유의어)가 한 그룹으로 분류되어 있는 사전

단어의 의미 파악

- **단어들을 의미의 상/하위 관계에 따라 그래프로 표현, 처리**
  - 단어들을 의미의 상위, 하위 관계에 기초하여 그래프로 표현
  - 모든 단어에 대한 유의어 집합을 만들고

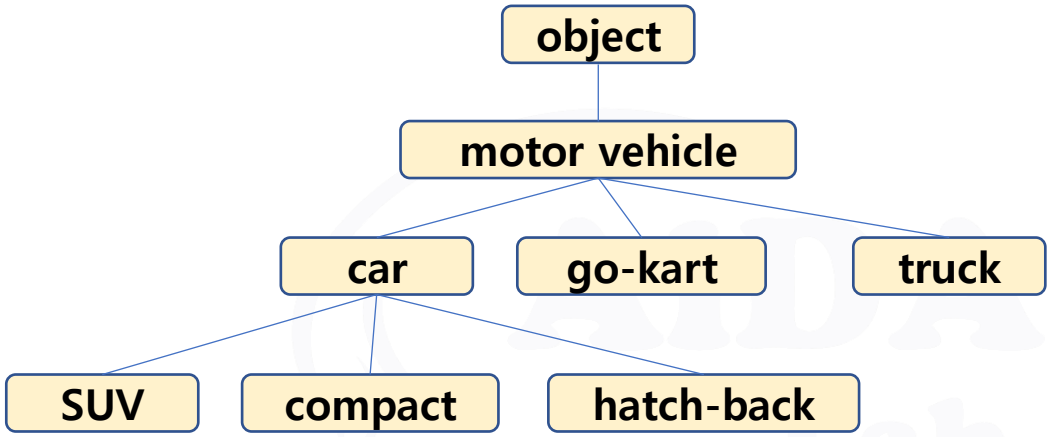
    → 단어들의 관계를 그래프로 표현하여

    → 단어 사이의 연결을 정의한 후

    → 단어의 네트워크를 이용하여

    → 컴퓨터에게 단어 사이의 관계를 주입

    → 컴퓨터에게 단어의 의미를 이해시킨 것으로 간주함



단어의 의미 파악

- **시소러스의 문제점**
  - 시대 변화에 대응하기 어려움 : 언어는 변한다 → 사람이 꾸준히 갱신, 관리해야 함
    - 신조어, 사멸어, 의미의 변화, 활용의 변화 등
  - 높은 비용 : 시소러스를 만들고 관리하기 위한 고가의 인적비용 발생
    - 현존 영어단어의 수: 1,000만개 이상 → 지금까지 WordNet에 등록된 단어: 20만개 이상
  - 단어의 미묘한 차이를 표현할 수 없다
    - 뜻은 비슷하지만 느낌, 활용 형태, 활용 상황이 다른 경우가 매우 많음 → 대응이 어려움

→ 문제의 해결을 위하여: 통계 기반 기법, 신경망을 이용한 추론 기반 기법 등이 제안됨

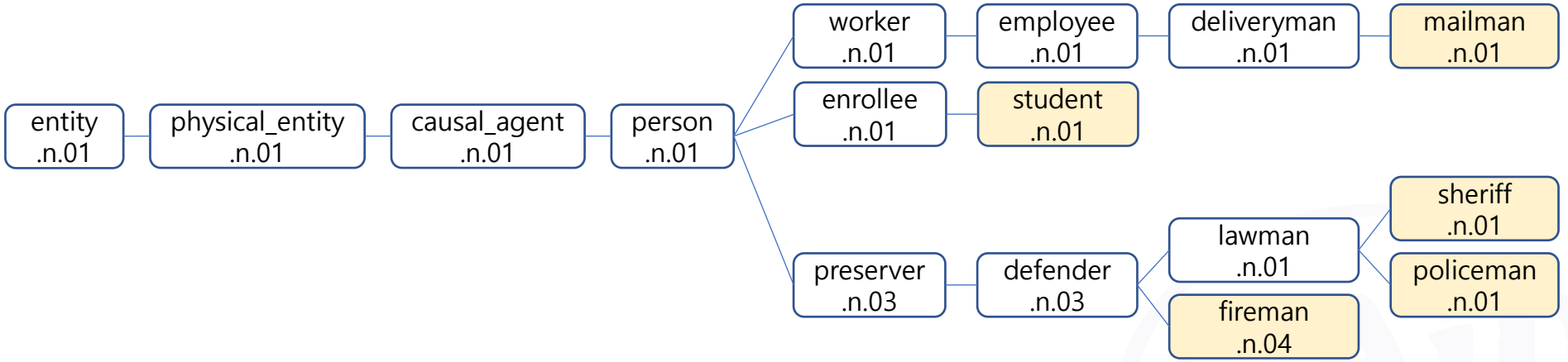
- **시소러스의 대표적인 모델: WordNet**

WordNet

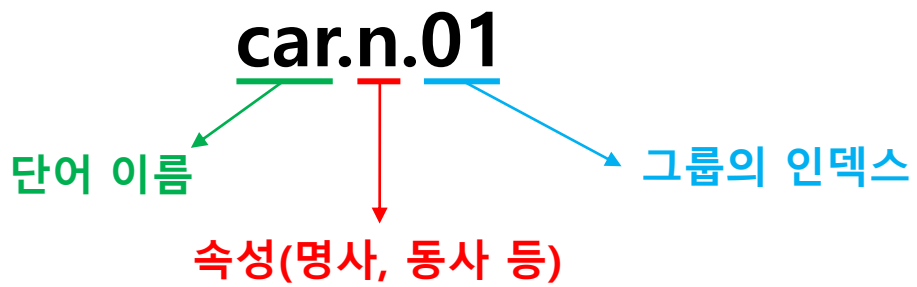
- WordNet이란?
  - 프린스턴 대학교 심리학 교수 “조지 아미티지 밀러”의 지도 하에 1985년부터 개발
  - 기계번역을 돕기 위한 목적으로 개발됨
  - 동의어 집합, 상위어, 하위어에 대한 정보가 잘 구축되어 있음
  - 유향 비순환 그래프(Directed Acyclic Graph, DAG)로 구성됨
    (트리구조가 아닌 이유: 하나의 노드가 여러 개의 상위 노드를 가질 수 있음)

WordNet

- **WordNet 사용 예시**



- **표제어 읽는 방법** WordNet 내의 단어 별 top-1 의미의 top-1 상위어만 선택하여 트리 구조로 나타낸 경우



WordNet

- **한국어 WordNet**
  - **표준이라고 할 만한 것은 없음**
  - **몇 개의 워드넷이 존재하는 정도**

이름	기관	웹사이트
KorLex	부산대학교	http://korlex.pusan.ac.kr
Korean WordNet(KWN)	KAIST	http://wordnet.kaist.ac.kr

WordNet

- 특징
  - 단어들은 그 내용을 설명할 수 있는 다양한 특징을 가짐
  - 특징벡터: 이러한 특징 별 수치를 모아 벡터로 표현한 것

  - 단어의 특징 벡터를 구성하기 위한 가정
    - 의미가 비슷한 단어라면 쓰임새가 비슷할 것
    - 쓰임새가 비슷하므로 비슷한 문장 안에서 비슷한 역할로 사용될 것
    - 따라서 함께 나타나는 단어들이 유사할 것

- 특징 추출하기
  - TF-IDF(Term Frequency-Inverse Document Frequency)
    - 어떤 단어가 출현한 문서의 숫자의 역수
      - TF: 단어가 문서에 출현한 횟수 → 값이 클 수록 문서에서 중요한 단어일 확률 높음
      - DF: 해당 단어가 출현한 문서의 수 →값이 클 수록 중요하다고 하기보다는 일반적으로 많이 쓰이는 단어일 가능성이 높음 (‘the’와 같은 것)
      - IDF를 구해 TF에 곱해줌으로써 ‘the’와 같은 단어에 대한 페널티를 적용함
    - 어떤 단어 w가 문서 d 내에서 얼마나 중요한지 나타내는 수치
    - 이 수치가 높을 수록 w는 d를 대표하는 성질을 띠게 된다고 볼 수 있음
    - $TF - IDF(w, d) = \frac{TF(w,d)}{DF(w)}$

단어의 의미 파악

통계 기반 기법

통계 기반 기법

- 통계 기반 기법은
  - 통계를 기반으로
    - 어떤 데이터가 많이 사용되는지
    - 데이터가 어떻게 분포되어 있는지
    - 데이터의 분포에 따라 어떤 연관성과 의미를 갖는지
  - 등의 정보를 추출하여 자연어를 처리하는 기법
  - 통계를 사용하므로 통계 결과를 계산하기 위한 충분한 크기를 가진 대규모의 텍스트 데이터가 요구됨

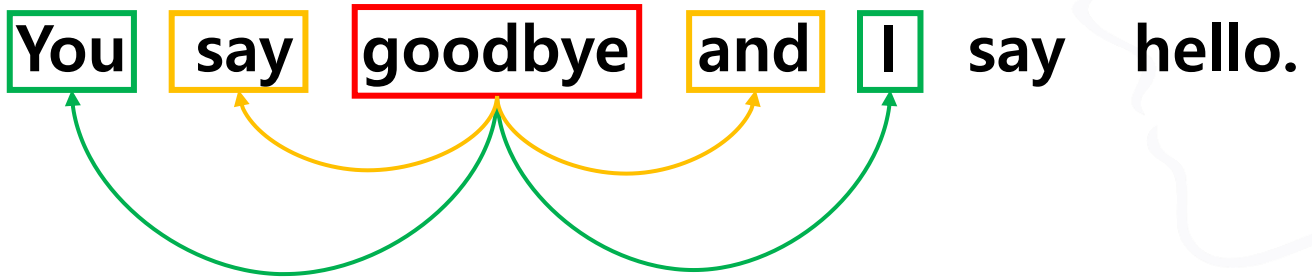
통계 기반 기법

- 말뭉치(Corpus)를 이용한다
  - 말뭉치: 자연어 처리에 대한 연구, 애플리케이션의 개발 등을 염두에 두고 수집된 대량의 텍스트 데이터
  - 말뭉치 자체는 단순한 텍스트 데이터이지만 텍스트에 담긴 문장은 사람이 쓴 글이다
    - 말뭉치에는 자연어에 대한 사람의 지식이 충분히 담겨있다.
      - 문장을 쓰는 방법, 단어를 선택하는 방법, 단어의 의미 등
  - **말뭉치에서 자동으로, 그리고 효율적으로 핵심을 추출하는 것**

    → 통계 기반의 자연어 처리 기법

통계 기반 기법

- 분포 가설(Distributional Hypothesis)
  - 단어의 의미는 주변 단어에 의해 형성된다.
  - 단어 자체에는 별 의미가 없고 그 단어가 사용된 맥락(Context)이 의미를 형성한다.
  - 자연어 처리에 관한 중요한 기법은 대부분 분포가설을 기반으로 한다.



통계 기반 기법

이런 방식으로
단어 사이의 연관성과
맥락을 이용함

I drink beer

We drink wine

Drink 의 주변에는
음료가 등장할 확률이 높다

I guzzle beer

We guzzle wine

Guzzle는 drink와 같은 맥락으로 사용될 확률이 높다
Guzzle는 drink와 가까운 의미의 단어일 것이다

* Guzzle : 폭음하다

통계 기반 기법

- 분포 가설을 기초로 통계적 분석을 하려면?
  - 일단 어떤 단어가 몇 번이나 나오는지 세어보자! → 핵심
  - 동시발생 행렬: 주어진 단어의 맥락으로써 동시에 발생하는 단어의 출현빈도
    - 맥락의 크기를 윈도우라고 하고, 윈도우가 1인 경우(주어진 단어에서 한 단어까지 처리)

You say goodbye and I say hello.

- You의 맥락으로 발생하는 단어는 say 뿐 → 동시발생 행렬은 [0, 1, 0, 0, 0, 0, 0]

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

You say goodbye and I say hello.

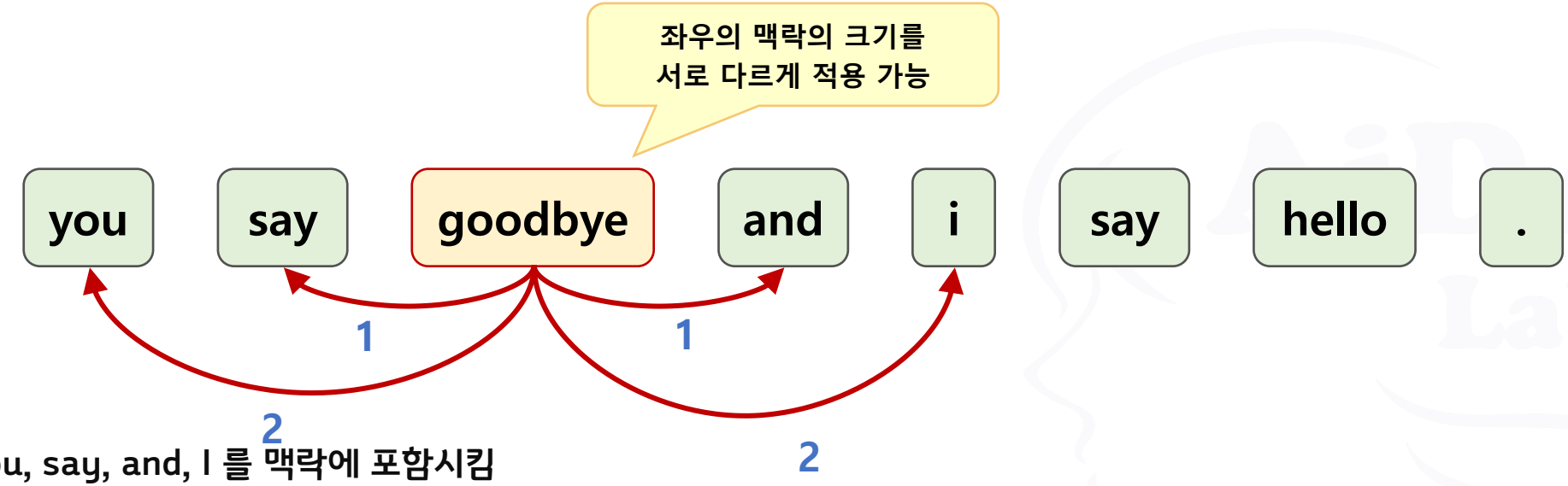
- say의 맥락으로 발생하는 단어는 You, goodbye, I, hello → [1, 0, 1, 0, 1, 1, 0]

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

통계 기반 기법

- **맥락이란 특정 단어를 중심에 둔 주변 단어를 말함**
  - 맥락의 크기: “윈도우”라고 부름
    - 윈도우 = 2 라면

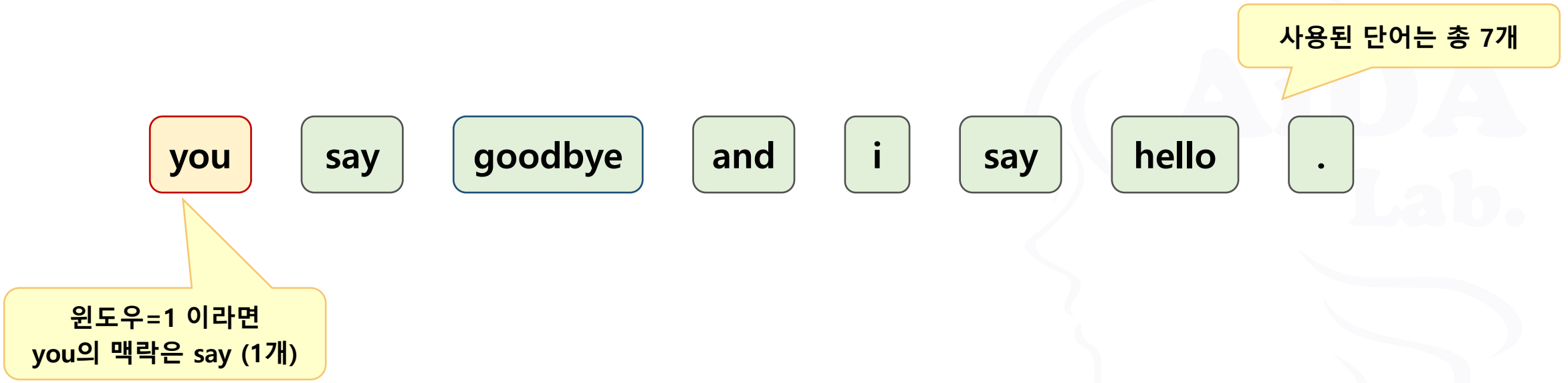
맥락이란 사물, 대상 또는 어떤 의미의 흐름이 서로 이어지는 관계성을 의미함. 그러나 컴퓨터에서 처리하기 위해서는 수치적으로 명확한 정의가 필요하므로..



- You, say, and, I 를 맥락에 포함시킴

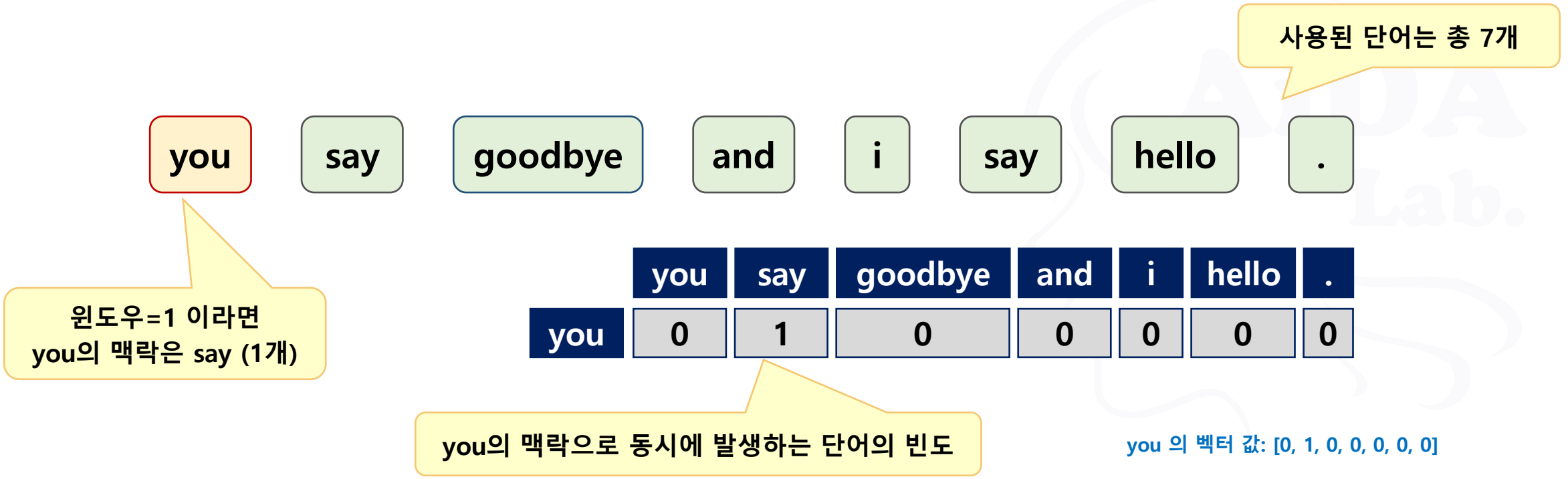
통계 기반 기법

- 동시발생 행렬
  - 분포 가설에 기초하여 단어를 벡터로 바꾸는 방법
  - 단어가 몇 개나 나오는지 세어본다. (통계 기반 기법)



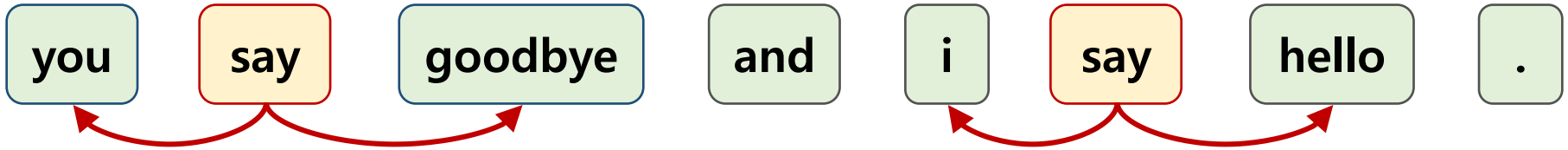
통계 기반 기법

- 동시발생 행렬
  - 분포 가설에 기초하여 단어를 벡터로 바꾸는 방법
  - 단어가 몇 개나 나오는지 세어본다. (통계 기반 기법)



통계 기반 기법

- 동시발생 행렬



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

[0, 1, 0, 0, 0, 0, 0]
[1, 0, 1, 0, 1, 1, 0]
[0, 1, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 1, 0, 0]
[0, 1, 0, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1, 0]

동시 발생 행렬

you 의 벡터 값: [0, 1, 0, 0, 0, 0, 0]

ID=0 인 단어의 벡터표현
[0 1 0 0 0 0 0]

통계 기반 기법

- 벡터 간 유사도
  - 동시발생 행렬을 이용하여 단어를 벡터로 표현하였다면
  - 벡터 사이의 유사도를 계산하여 해당 단어 간의 유사도, 관련성을 확인 가능
    - 벡터 간 유사도 계산 방법
      - 벡터의 내적
      - 유클리드 거리
      - **코사인 유사도**: 단어 벡터 간의 유사도를 나타
        - $Similarity(x, y) = \frac{x \cdot y}{||x||||y||} = \frac{x_1 y_1 + \cdots + x_n y_n}{\sqrt{x_1^2 + \cdots + x_n^2}\sqrt{y_1^2 + \cdots + y_n^2}}$
      - 기타...

코사인 유사도 → 두 벡터가 가리키는 방향이 얼마나 비슷한가?를 확인
벡터의 방향이 완전히 같다면 코사인 유사도 1, 완전히 반대라면 -1

통계 기반 기법

- 유사 단어의 랭킹 표시
  - 유사도가 높은 순서로 순위를 매겨서 활용
  - 입력 데이터인 말뭉치(텍스트 데이터)가 커질수록 랭킹은 정확해짐
  - 말뭉치가 작을수록 납득하기 어려운 순위를 보여줌(당연히)

이런 과정을 거쳐 도출된 각 단어의 유사도 랭킹(순위)에 따라
단어, 문맥(맥락), 문장에 해당하는 의미를 선택하고
그 결과를 활용하는 것이 통계 기반의 기법

통계 기반 기법

- 개선 사항
  - 통계 기반 기법에서는 단어의 동시발생 행렬을 이용함
  - 발생횟수, 단어의 빈도수는 많이 활용되는 특성이지만 좋은 특징은 아님
  - 전문적인 내용의 말뭉치일 경우, 해당 주제에 관련된 말이 비정상적으로 많이 나올 수 있음
  - 영어권 언어에서는 관사 등 실제 의미와 무관한 단어가 최고 빈도수를 가짐

통계 기반 기법

- **단어의 빈도수를 이용할 경우 발생가능한 문제의 사례**
  - **the, car, drive 등의 단어가 포함된 말뭉치의 경우**
    - car, drive는 관련성이 깊은 단어임
    - 그러나 처리를 해 보면 the와 car의 관련성이 더 높은 것으로 계산됨
    - the와 같은 관사는 내용과 관계없이 가장 많이 사용되는 단어의 하나이므로 최다빈도수를 가짐 → 올바르지 않은 결과 초래

통계 기반 기법

- **점별 상호 정보량(Pointwise Mutual Information, PMI)**
  - $\boldsymbol{PMI(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)}}$
    - $P(x)$: x가 일어날 확률, $P(y)$: y가 일어날 확률, $P(x,y)$: x와 y가 동시에 일어날 확률
    - PMI 값이 높을수록 관련성이 높음
  - 말뭉치에 적용하면...
    - $P(x)$: 단어 x가 말뭉치에 등장할 확률
    - $P(x,y)$: 단어 x와 y가 말뭉치에 동시에 등장할 확률

10,000개의 단어로 구성된 말뭉치에서

"the"가 100번 등장한다면, $\boldsymbol{P(\text{“}the\text{”}) = \frac{100}{10000} = 0.01}$

"the"와 "car"가 10번 동시에 발생했다면, $\boldsymbol{P(\text{“}the\text{”},\text{“}car\text{”}) = \frac{10}{10000} = 0.001}$

통계 기반 기법

- **동시발생 행렬을 사용하여 다시 정리하면**
  - $\boldsymbol{PMI(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)} = log_2 \frac{\frac{C(x,y)}{N}}{\frac{C(x)}{N}\frac{C(y)}{N}} = log_2 \frac{C(x,y)\cdot N}{C(x)C(y)}}$
    - C: 동시발생 행렬, $C(x)$: 단어 x의 등장횟수, $C(y)$: 단어 y의 등장횟수
    - $C(x,y)$: 단어 x와 y가 동시 발생하는 횟수
    - N: 말뭉치에 포함된 단어 수
  - **예) 말뭉치 단어 수가 10,000개, “the” 1,000번, “car” 20번, “drive” 10번 등장했고, “the”, ”car”의 동시발생 수는 10회, “car”, “drive”의 동시발생 수는 5회라고 가정**
    - $\boldsymbol{PMI(“the”,”car”) = log_2 \frac{10\cdot 10000}{1000\cdot 20} \approx 2.32}$, $\boldsymbol{PMI(“car”,”drive”) = log_2 \frac{5\cdot 10000}{20\cdot 10} \approx 7.97}$
    - “car”, “drive”의 관련성이 더 높은 것을 확인할 수 있음

통계 기반 기법

- PMI의 문제점
  - 두 단어의 동시발생 횟수가 0 이면 $log_2 0 = -\infty$가 됨
  - 문제 해결을 위하여 실제 구현 시에는 양의 상호정보량(Positive PMI, PPMI) 사용
    - $PPMI(x, y) = \max(0, PMI(x, y))$ → PMI가 음수일때는 0으로 취급
  - PMI / $PPMI$ 모두 말뭉치의 어휘 수 증가에 따라 단어벡터의 차원 수가 증가
    → 차원 감소를 위한 방안이 요구됨

통계 기반 기법

- **차원 감소**
  - 벡터의 차원을 줄이는 기법
  - 단순한 차원 감소가 아니라 “중요한 정보”는 최대한 유지하면서 줄여야 함
  - 단어 벡터는 주로 원핫 인코딩과 같은 희소벡터(=희소행렬)를 사용하므로

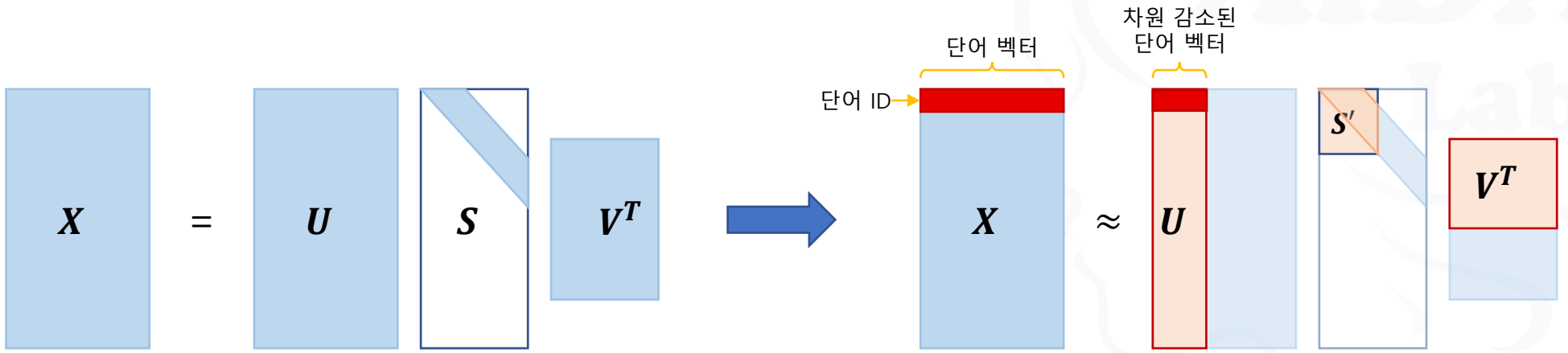
    → 희소벡터에서 중요한 축을 찾아내어 더 적은 차원으로 다시 표현하는 것이 핵심
  - 차원 감소 결과는 원소 대부분이 0이 아닌 값으로 구성된 밀집벡터가 됨

    → 단어의 분산 표현
  - 다양한 차원감소 기법 중 단어 처리에서 주로 사용되는 것은

    → 특잇값 분해(Singular Value Decomposition, SVD)

통계 기반 기법

- **특잇값 분해(SVD)**
  - 주어진 행렬(X)을 3개의 행렬(U,S,V)의 곱으로 분해 → $X = USV^T$
    - U, V: 직교행렬(열 벡터가 서로 직교함) → 어떤 공간의 축(기저)을 형성. U행렬은 단어 공간
    - S: 대각행렬(대각성분 외에는 모두 0) → 대각성분은 특잇값(=해당 축의 중요도)
      → 중요도가 낮은 원소(특잇값이 작은 원소)를 깎아내어 차원 축소에 반영



SVD에 의한 행렬의 변환(행렬의 흰 부분은 원소가 0임)

SVD에 의한 차원 감소

통계 기반 기법

- **PTB(Penn Treebank) 데이터 셋**
  - 펜실베이니아 대학교에서 관리하는 데이터 셋
  - NLP(자연어 처리) 연구를 위한 기계 학습에 널리 사용되는 데이터 셋
  - 대부분의 최신 데이터 집합에 비해 상대적으로 작아서 주어진 기법의 품질을 측정하는 벤치마크로 자주 이용됨
  - 대문자, 숫자 및 문장 부호가 포함되어 있지 않으며 어휘는 10k 고유 단어로 제한됨
  - 스피치 조각, 구문 및 의미론적 골격과 같은 다양한 종류의 주석으로 나뉨

통계 기반 기법

- 문제점
  - 통계 기반 기법에서는 주변 단어의 빈도를 기초로 단어를 표현함
  - 단어의 동시발생 행렬을 만들고 그 행렬에 SVD를 적용하여 밀집벡터(분산표현)를 획득 → 대규모의 말뭉치를 다룰 때 문제 발생

SVD를 $n \times n$ 행렬에 적용하는 비용은 $O(n^3)$

100만개의 어휘를 다루려면 → $10^6 \times 10^6 = 10^{12}$ 크기의 행렬 생성

$O((10^{12})^3)$만큼의 계산 시간이 소요됨 → 슈퍼컴퓨터로도 처리 불가능에 가까움

  - 특히 통계 [redacted] 현을 확보 → 시스템의 부하가 매우 큼

단어의 의미 파악

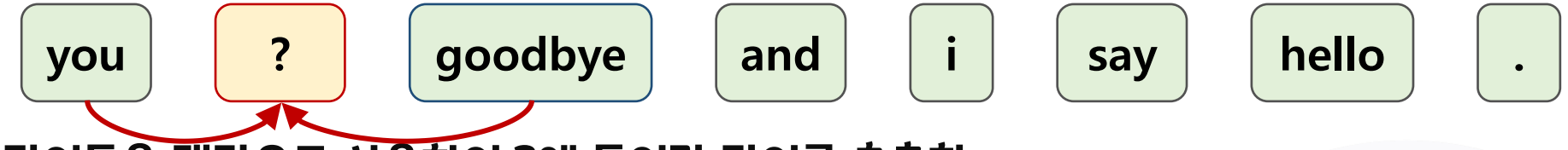
추론 기반 기법

추론 기반 기법

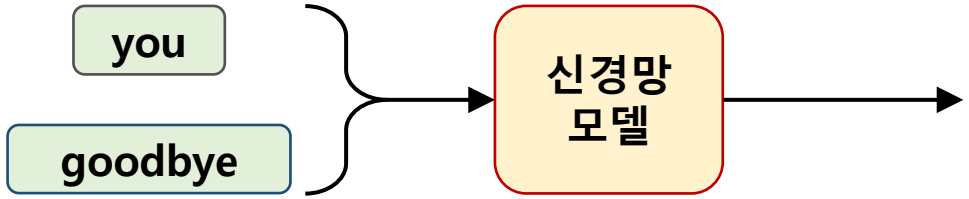
- 추론 기반 기법은
  - 신경망을 이용한 추론을 기반으로 단어의 분산 표현을 얻는 기법
  - 통계 기반 기법과 마찬가지로 분포 가설을 기초로 함
  - 통계 기반 기법과의 비교
    - 통계 기반 기법: 학습 데이터를 한꺼번에 처리함(배치 학습)
    - 추론 기반 기법: 학습 데이터의 일부를 사용하여 순차적으로 학습함(미니배치 학습)
      - 말뭉치의 어휘 수가 많아 SVD 등 계산량이 큰 작업을 처리하기 어려운 경우에도 학습 가능
      - GPU를 이용한 병렬 계산 가능 → 학습 속도 향상

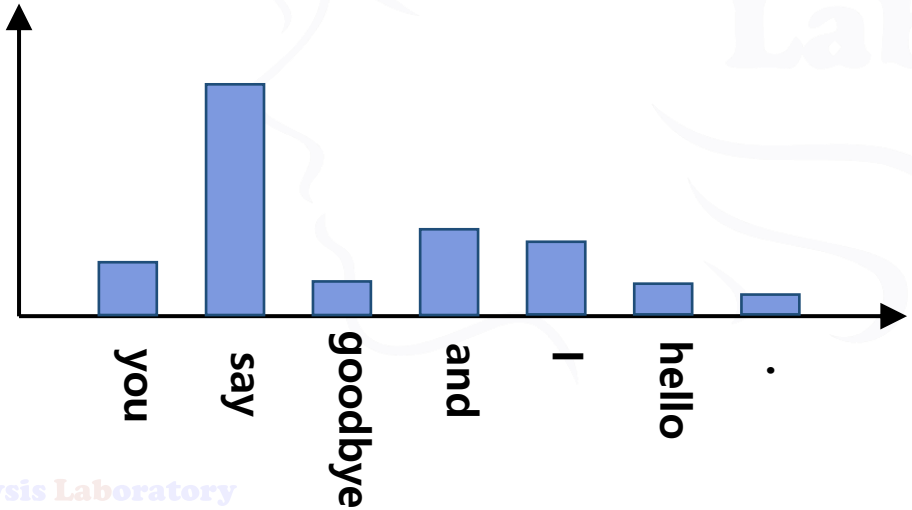
추론 기반 기법

- 추론 기반 기법에서의 “추론”이란?



- 주변 단어들을 맥락으로 사용하여 ?에 들어갈 단어를 추측함
- 이러한 추론 문제를 반복 수행하여 단어의 출현 패턴을 학습함

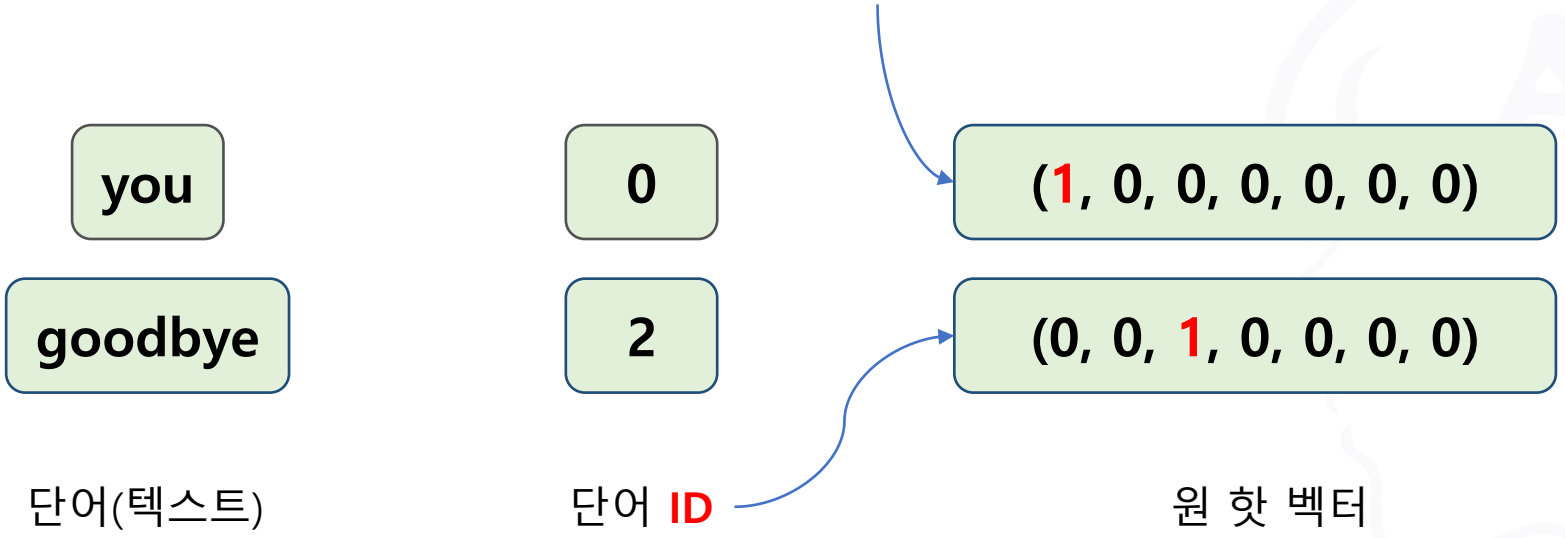




맥락을 입력하여 신경망 모델을 통해 각 단어의 출현확률을 계산

추론 기반 기법

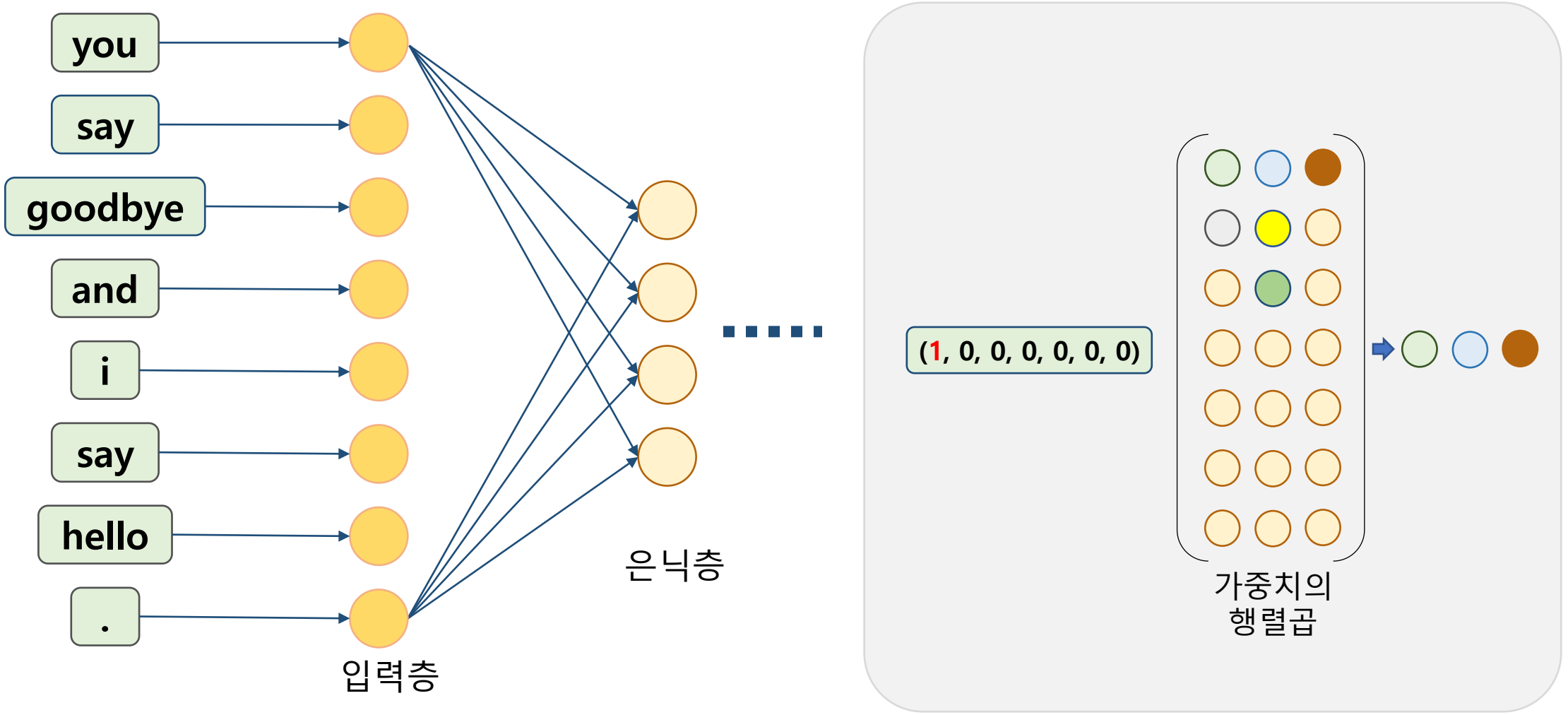
- 신경망에서의 단어 처리
  - 단어를 고정 길이의 벡터로 변환 (원 핫 벡터: One-hot Vector)
    - 원 핫 벡터: 벡터의 원소 중 1개만 1이고 나머지는 모두 0인 벡터
      - You say goodbye and I say hello : 총 7개의 단어



추론 기반 기법

- 원 핫 벡터(고정길이 벡터)를 입력층의 뉴런에 대응

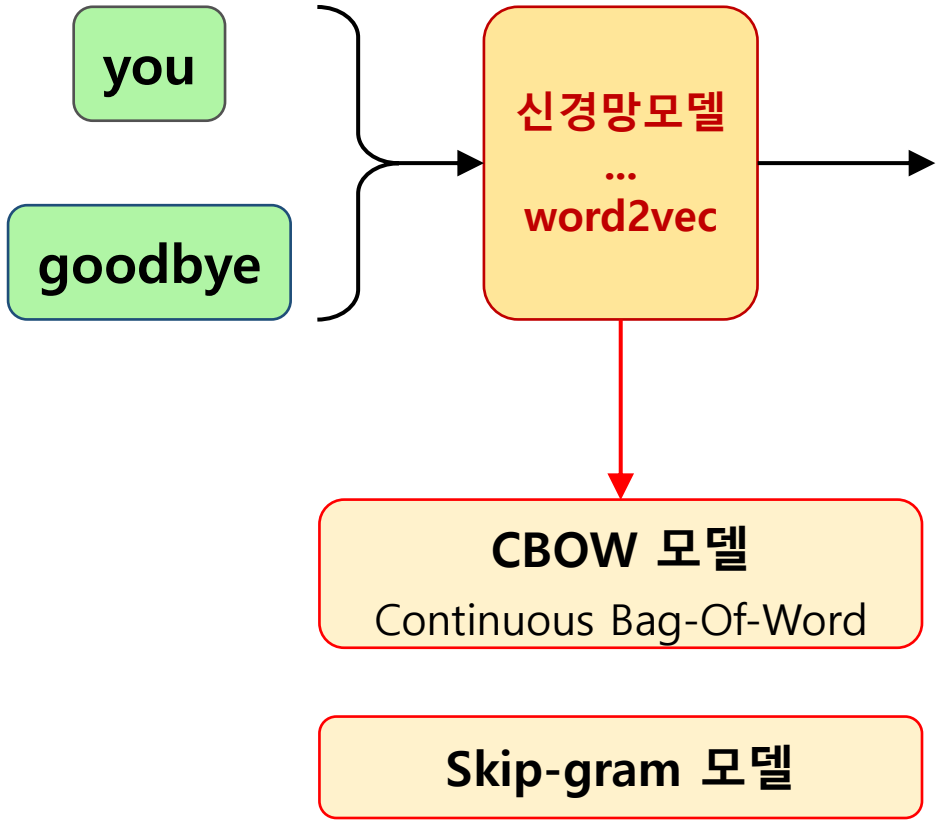
추론 기반 기법



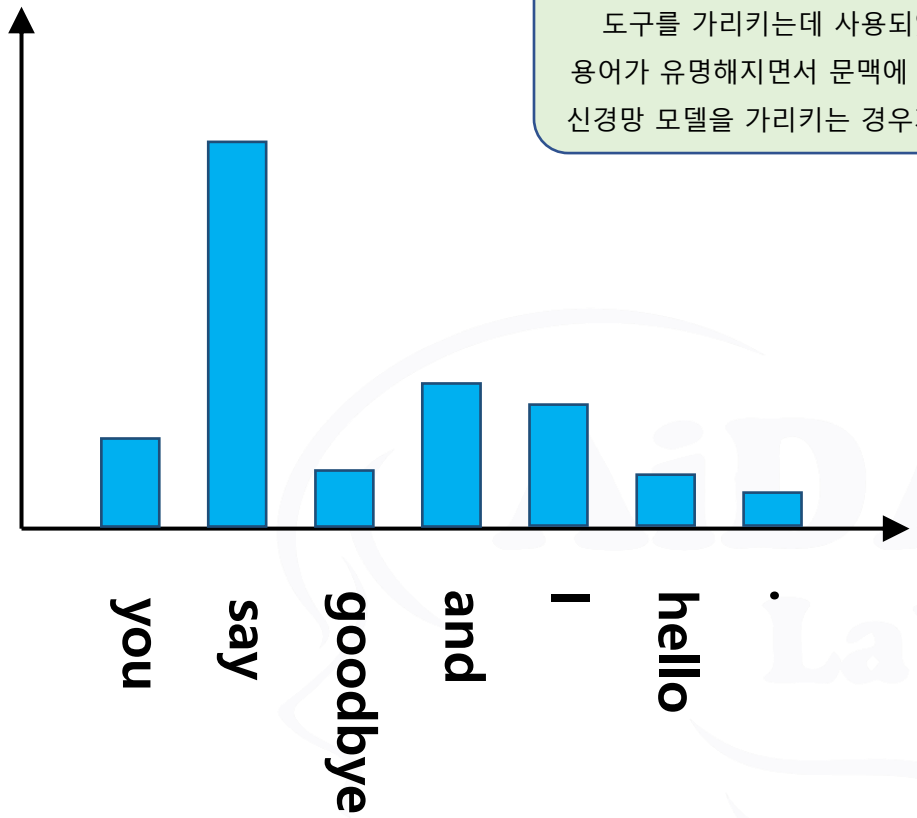
추론 기반 기법

- **Word2Vec**
  - **2013년 Google에서 발표, 현재 가장 많이 사용되고 있는 단어 임베딩 모델**
  - **기존 신경망 기반의 단어 임베딩과 구조의 차이는 크지 않으나 계산량을 획기적으로 줄여 빠른 학습을 가능하게 하였음**
  - **CBOW(Continuous Bag-of-Words) 모델과 skip-gram 모델이 있음**
    - CBOW: 신경망의 입력을 주변 단어들로 구성하고 출력을 타깃 단어로 설정하여 학습된 가중치 데이터를 임베딩 벡터로 활용
    - skip-gram: 하나의 타깃 단어를 이용하여 주변 단어들을 예측하는 모델(CBOW와 반대)

추론 기반 기법

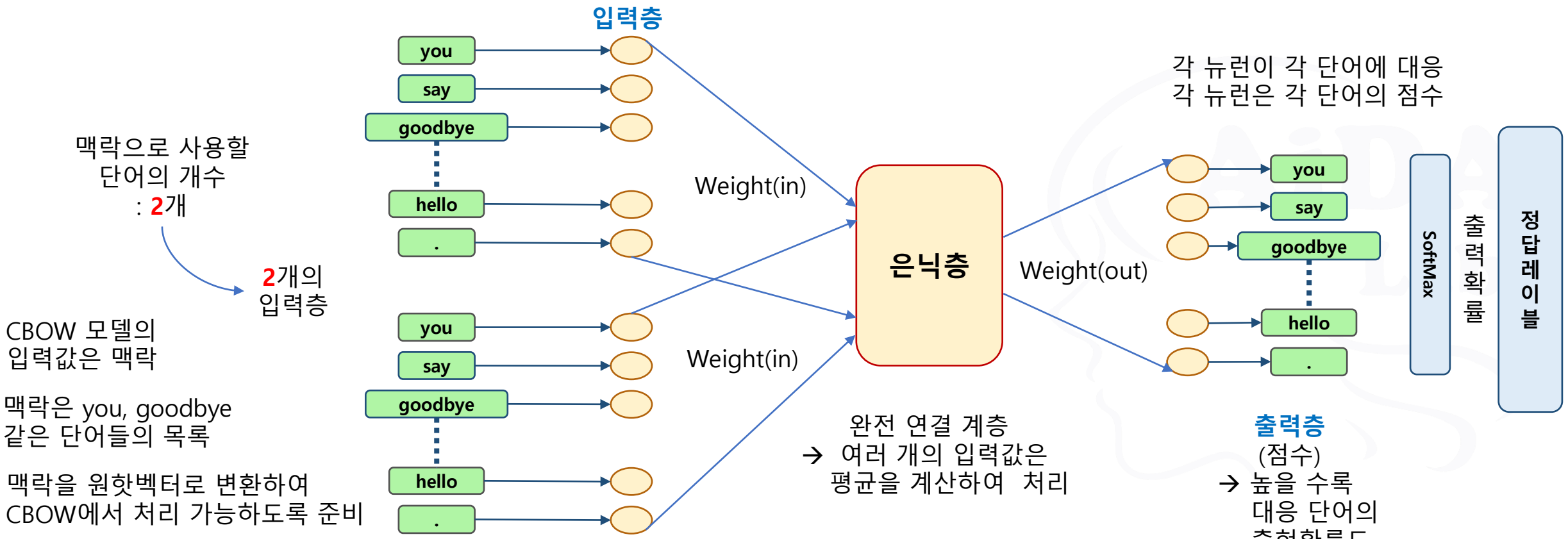


word2vec은 원래 프로그램이나
도구를 가리키는데 사용되었으나
용어가 유명해지면서 문맥에 따라서는
신경망 모델을 가리키는 경우가 늘어남

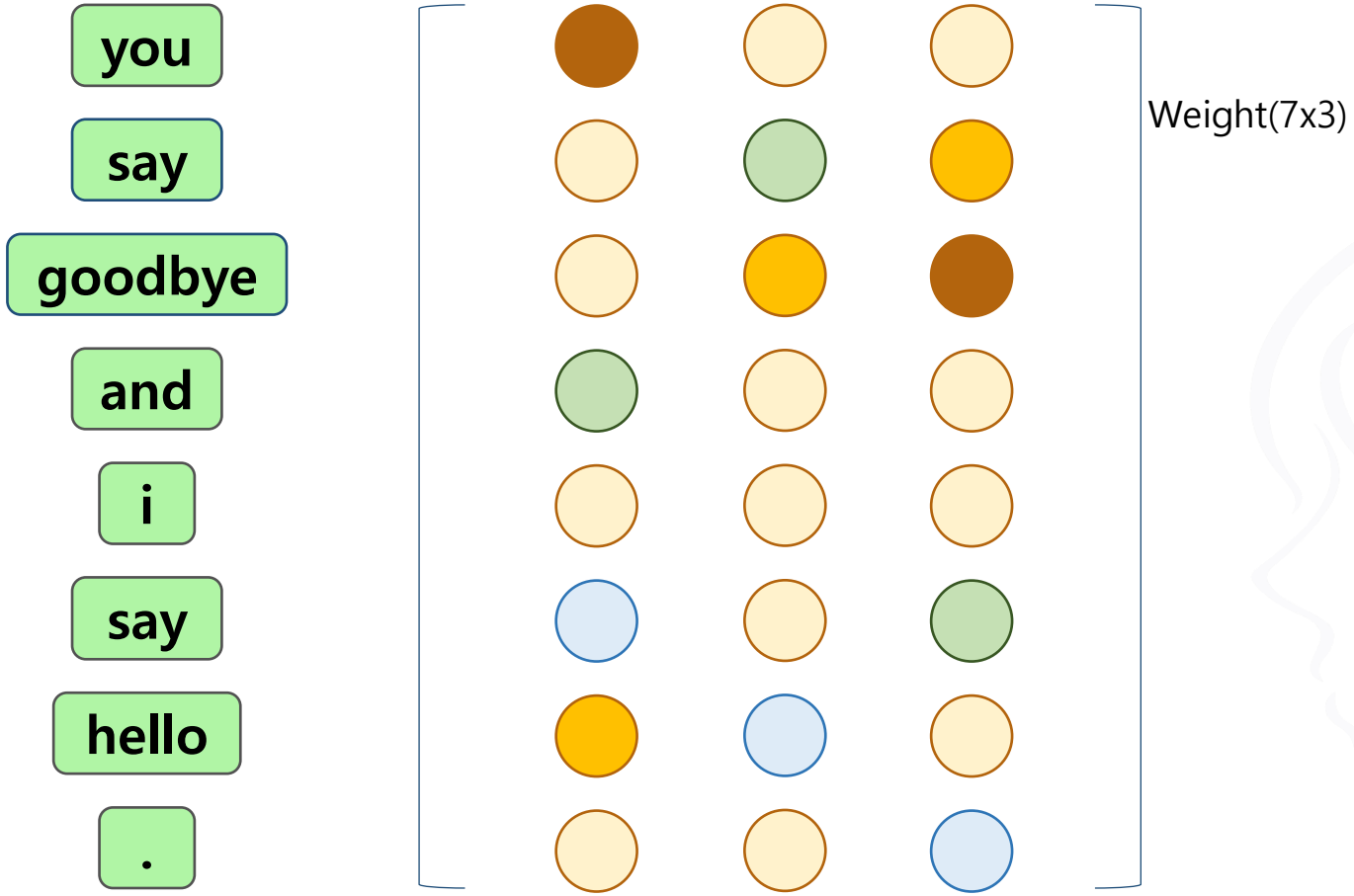


추론 기반 기법

- **CBOW 모델**
  - **맥락으로부터 타겟을 추측하기 위한 신경망 (타겟: 중앙 단어, 맥락: 주변 단어)**



추론 기반 기법

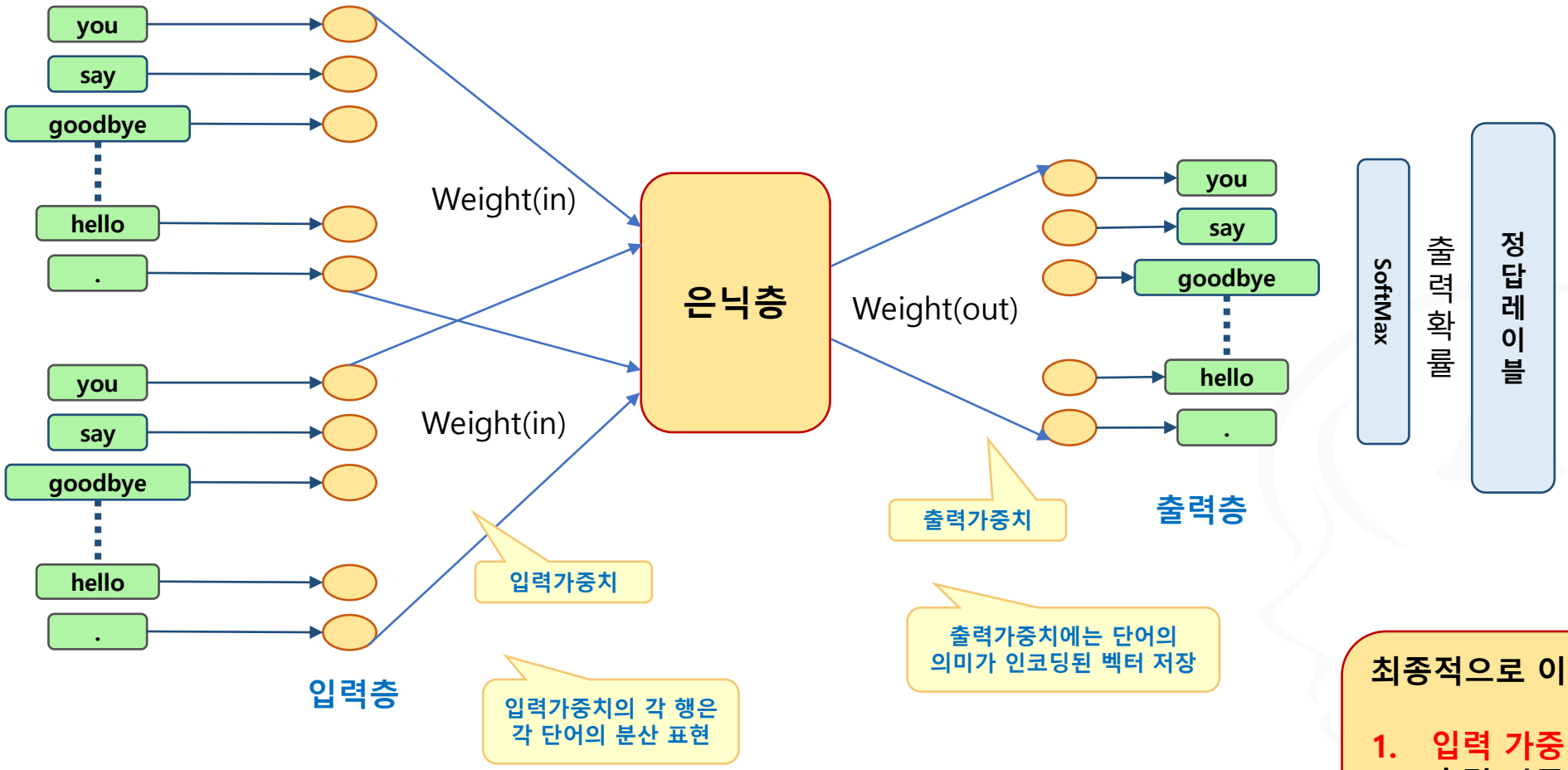


가중치의 각 행이
해당 단어의 분산 표현이다

학습이 진행될수록
맥락에서 출현하는 단어를
잘 추측하는 방향으로
분산표현이 갱신됨

신경망을 운영해 보니
단어의 의미도
벡터에 잘 녹아 들어 있었음

추론 기반 기법

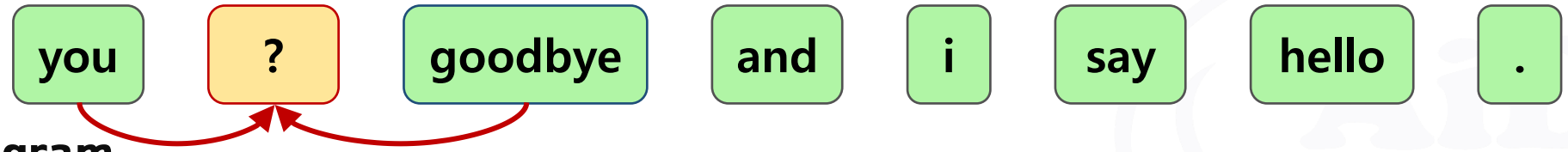


최종적으로 이용하는 단어의 분산표현

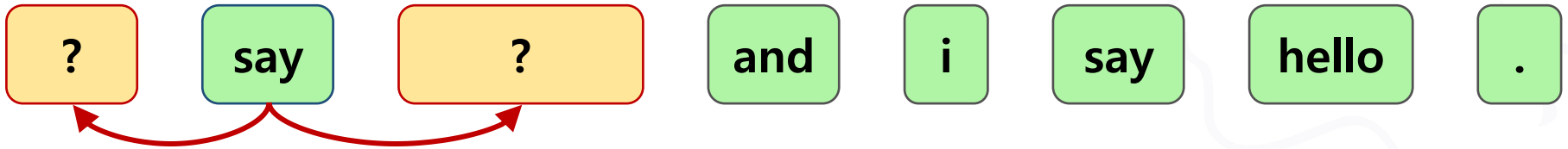
1. **입력 가중치만 활용 (대세)**
2. **출력 가중치만 활용**
3. **모든 가중치를 활용**
   →어떻게 조합할 것인가?

추론 기반 기법

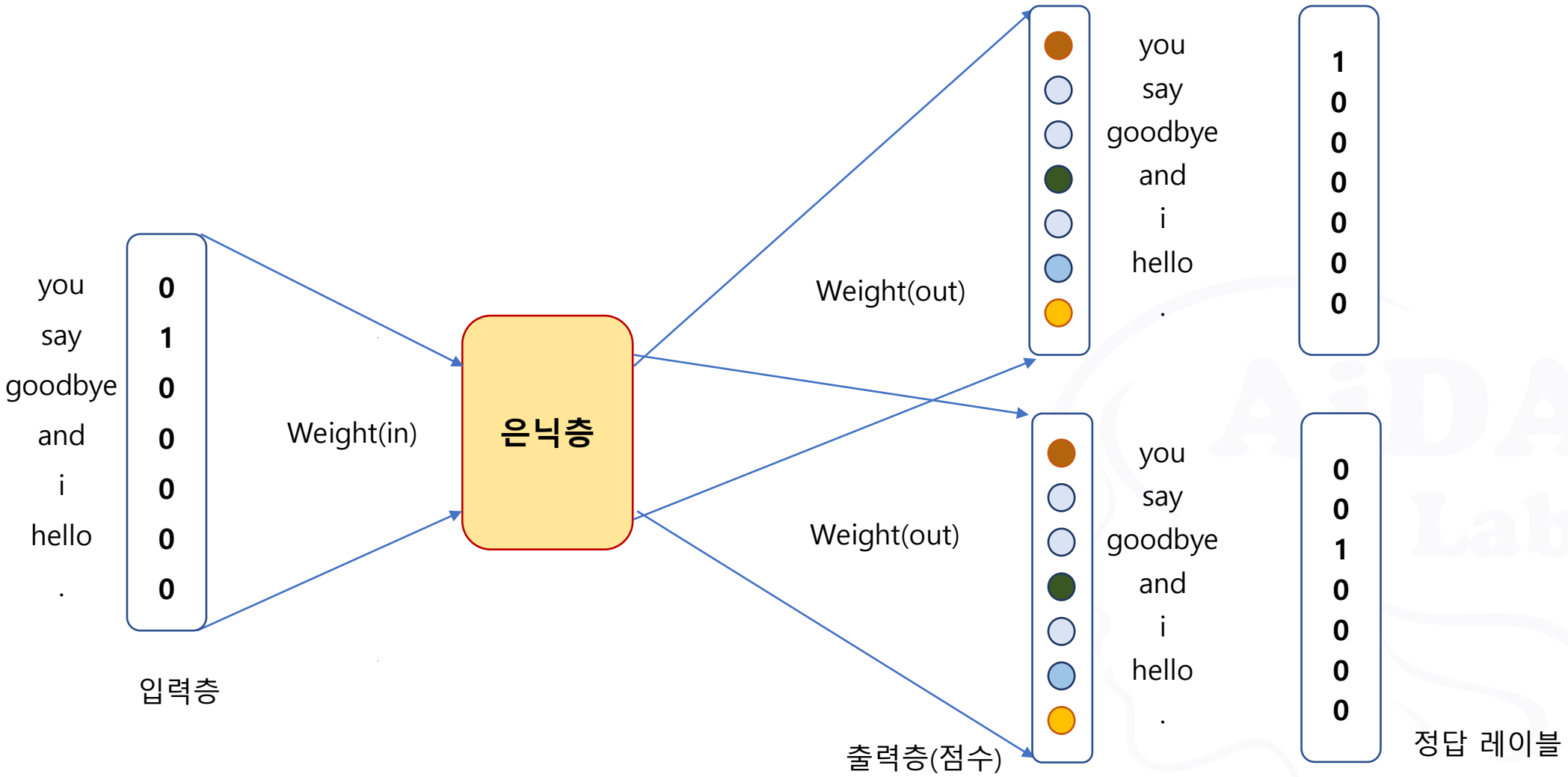
- skip-gram 모델
  - CBOW 모델에서 맥락과 타겟을 역전시킨 모델
  - CBOW



  - skip-gram



추론 기반 기법



추론 기반 기법

- **skip-gram의 상세 훈련 방식**
  - **학습 과정**

최대우도법: 원래 모집단의 분포에 최대한 가깝게 파라미터를 추정하는 방법

    - **최대우도법(Maximum Likelihood Estimation, MLE)을 통해 수식을 최대로 만드는 θ를 찾는다.**

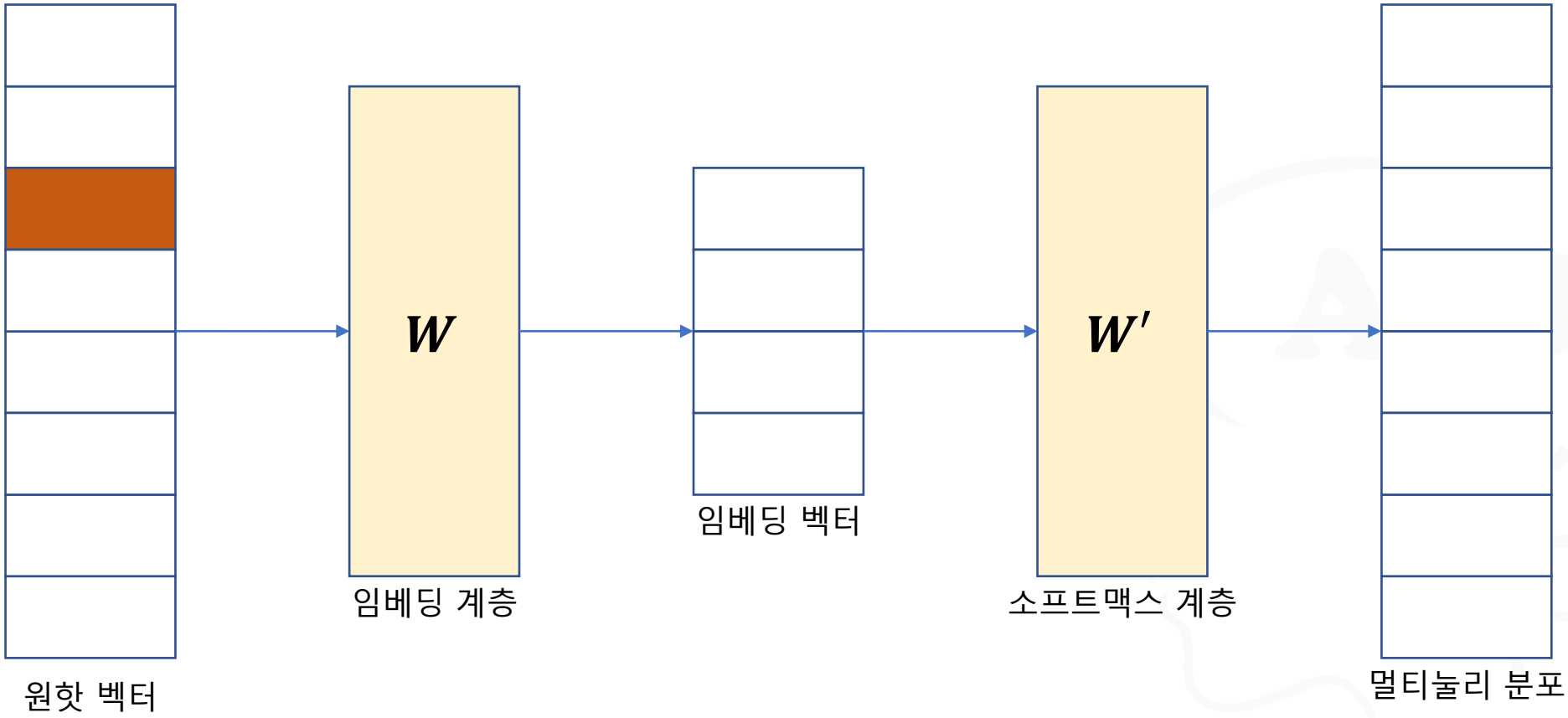
      - $\hat{\theta} = \underset{\theta}{\text{argmax}} \sum_{t=1}^{T} (\sum_{i=1}^{n} logP(w_{t-i}|w_t;\theta) + \sum_{i=1}^{n} logP(w_{t+i}|w_t;\theta))$

    - **이를 통해 w_t가 주어졌을 때, 앞뒤 n개의 단어 $(w_{t-n}, \dots, w_{t+n})$를 예측하도록 훈련한다.**
      이때 윈도우의 크기는 n이다.
    - **아래의 식을 통해 임베딩 벡터로 만든다.**

      - $\hat{y} = \underset{y \in Y}{\text{argmax}}\ softmax(W'W_x),\ where\ W' \in \mathbb{R}^{|V| \times d}, W \in \mathbb{R}^{d \times |V|}\ and\ x \in \{0,1\}^{|V|}$

추론 기반 기법

- **skip-gram의 구조**



멀티눌리 분포: 이산 확률 분포의 하나

추론 기반 기법

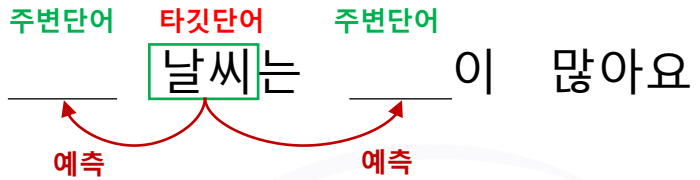
- CBOW 모델 vs. skip-gram 모델
  - Skip-gram 모델의 경우가 더 좋은 결과를 출력함
    - Skip-gram 모델: 맥락의 수 만큼 추측 → 손실함수는 각 맥락에서 구한 손실의 총합
    - CBOW 모델: 타겟 하나의 손실
  - 정밀도 면에서 skip-gram 모델이 우세
  - 말뭉치가 커질수록 저빈도 단어, 유추문제 성능에서 skip-gram 모델이 뛰어남
  - 단, 학습 속도는 CBOW 모델이 빠름
  - Skip-gram 모델은 맥락의 수 만큼 손실을 구해야 하므로 계산비용이 높아짐

추론 기반 기법

- **CBOW**

- 타깃 단어의 손실만 계산하면 된다
- 학습속도가 빠르다

- **skip-gram**



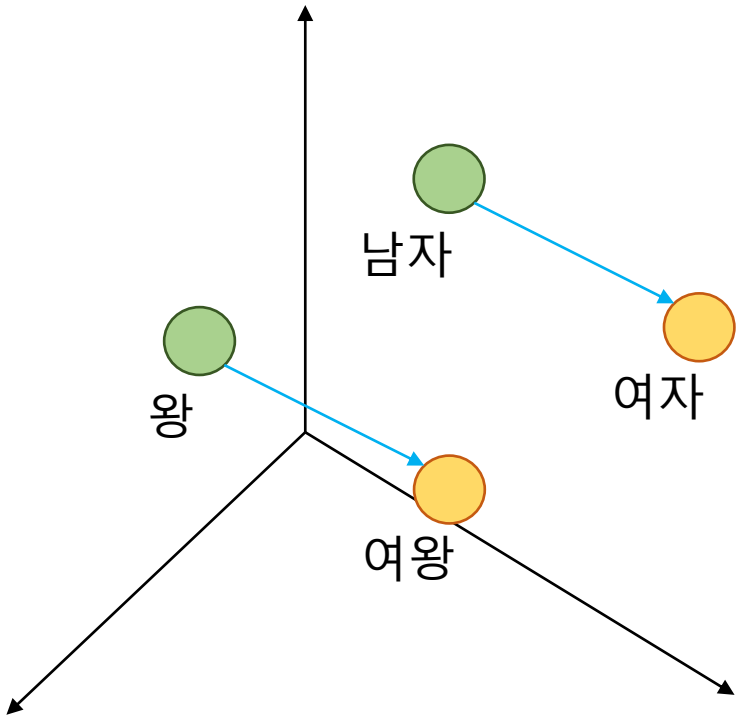
- CBOW 모델에 비해 예측해야 하는 맥락이 많다
- 단어 분산 표현력이 우수하므로 CBOW 모델보다 임베딩 품질이 우수하다

추론 기반 기법

- word2vec 모델의 독특한 성능
  - 언어(단어) 사이의 유추 문제를 연산을 통해서 풀 수 있다
    - King – man + woman = queen
    - 안경 낀 남자 – 남자 + 안경 끼지 않은 여자 = 안경 낀 여자
    - 등…

추론 기반 기법

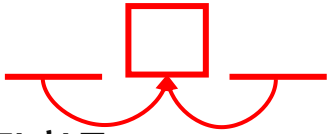
- **Word2Vec의 두 가지 모델이 다루는 문제**



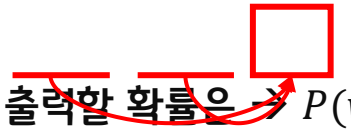
- 해당 단어를 밀집벡터로 표현하며
- 학습을 통해 의미상 비슷한 단어들을 비슷한 벡터공간에 위치시킴
- 또한 벡터의 특성 상 방향성을 갖게 됨
- 임베딩된 벡터 간의 연산이 가능함
- 왕과 여왕의 방향 차이만큼 남자와 여자의 방향 차이가 발생하는 특징이 있으며, 이를 이용해 자연어의 의미 파악이 가능함

확률의 관점에서 보는 word2vec

- **CBOW 모델**
  - **맥락의 단어로부터 타깃 단어를 추측함**
    - **윈도우 크기가 1일때:** $w_1 \quad w_2 \quad \dots \quad w_{t-1} \quad w_t \quad w_{t+1} \quad \dots \quad w_{T-1} \quad w_T$



      - w_{t-1}과 w_{t+1}이 주어졌을 때 타깃이 w_t가 될 확률: $P(w_t|w_{t-1}, w_{t+1})$
      - CBOW 모델은 위의 식에 대한 사후 확률을 모델링 함
      - 지금까지는 맥락을 좌우 대칭으로만 생각했으나, 왼쪽 윈도우 만으로 한정한다면...
      - $w_1 \quad w_2 \quad \dots \quad w_{t-2} \quad w_{t-1} \quad w_t \quad w_{t+1} \quad \dots \quad w_{T-1} \quad w_T$



      - CBOW 모델이 출력할 확률은 → $P(w_t|w_{t-2}, w_{t-1})$

확률의 관점에서 보는 word2vec

- $P(w_t|w_{t-2}, w_{t-1})$ → 교차 엔트로피 오차를 이용해 유도하면
  → $L = -logP(w_t|w_{t-2}, w_{t-1})$ 라는 손실함수를 유도할 수 있음
- CBOW 모델의 학습에서 수행하는 일은 손실함수(말뭉치 전체의 손실함수의 총합)를 최소화하는 가중치 매개변수를 찾는 것
- 최적의 가중치 매개변수를 이용하면 더 정확한 타깃 추측이 가능해짐
- 또한 학습의 부산물로 단어의 의미가 인코딩된 "단어의 분산표현"을 얻을 수 있음

- CBOW 모델의 목적인 "맥락으로부터 타깃을 추측하는 것"은 어디에 활용하나?
- $P(w_t|w_{t-2}, w_{t-1})$ 식은 어디에 쓸 수 있나?

➔ 이러한 활용을 위해 적용하는 것이 다양한 **언어모델**

언어모델은 단어의 나열에 확률을 부여함 → **기계 번역, 음성 인식, 문장 생성** 등에 응용할 수 있음

확률의 관점에서 보는 word2vec

- **CBOW 모델을 직접 언어 모델로 적용한다면?**
  - **적당한 규모의 데이터에서는 활용 가능함**
  - **그러나 CBOW 모델은 맥락 안의 단어의 순서가 무시된다는 한계를 가지고 있음**
  - **따라서 CBOW 모델의 결과물을 제대로 된 언어 모델에 적용하는 것이 더 좋음**

CBOW란 Continuous Bag-of-Words 의 약어
→ 가방 안의 단어들을 의미하므로...
→ 순서와는 관계없이 들어가 있음
→ 대신 "분포"의 개념이 사용됨

단어의 의미 파악

통계 기반 기법과 추론 기반 기법 비교

통계 기반 기법 vs 추론 기반 기법

- 새로운 단어가 생겨서 말뭉치에 추가해야 한다?
  - 통계 기반 기법: 처음부터 모두 다시 시작
  - 추론 기반 기법: 현재 상황에서 추가로 학습 가능

- 단어의 분산 표현의 성격이나 정밀도는?
  - 통계 기반 기법: 주로 단어의 유사성이 인코딩 됨
  - 추론 기반 기법: 단어의 유사성, 단어 사이의 패턴도 인코딩 됨

- 그러나 실제로 단어의 유사성을 정량평가 해 보면 → 둘 다 비슷

통계 기반 기법 vs 추론 기반 기법

- 통계 기반 기법과 추론 기반 기법은 내부적인 계산을 기준으로 비교해보면 서로 연관된 관계에 있음
  - 예:
    - 추론 기반 기법의 Skip-gram 모델과 네거티브 샘플링을 이용한 통계기반 기법의 모델은
    - 말뭉치 전체의 동시발생 행렬에 특수한 행렬분해를 적용한 것과 동일함

      → 서로 연관되어 있는 모델