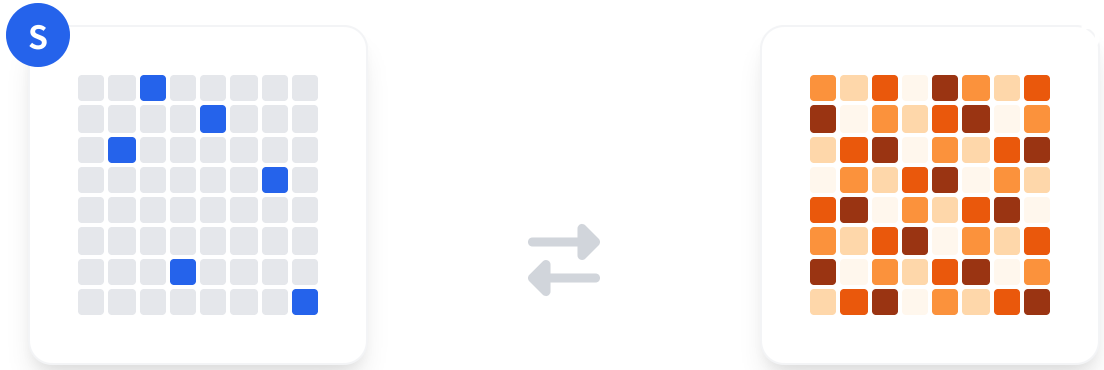
NLP 추천시스템 벡터 표현

밀집벡터 vs 희소벡터

데이터 표현의 두 가지 핵심 방식 이해와 활용 전략



희소벡터 (Sparse)
대부분이 0인 고차원

밀집벡터 (Dense)
실수값으로 채워진 저차원

OBJECTIVE 1
벡터 표현의 기본 개념

OBJECTIVE 2
두 벡터의 특징 비교

OBJECTIVE 3
최적의 활용 전략 도출

CONCEPT | 02. Sparse Vector

희소벡터 (Sparse Vector)

벡터의 **차원은 매우 크지만**(High-dimensional),
대부분의 요소가 **0**으로 채워져 있는 벡터 표현 방식

주요 특징

직관적 해석 (Interpretability)
각 차원이 특정 단어나 속성과 1:1 매칭되어 의미 파악 용이

차원의 저주
불필요한 0 값이 많아 공간 낭비 및 연산 비효율 발생 가능

One-hot Encoding 예시

Vocabulary Size: 10,000

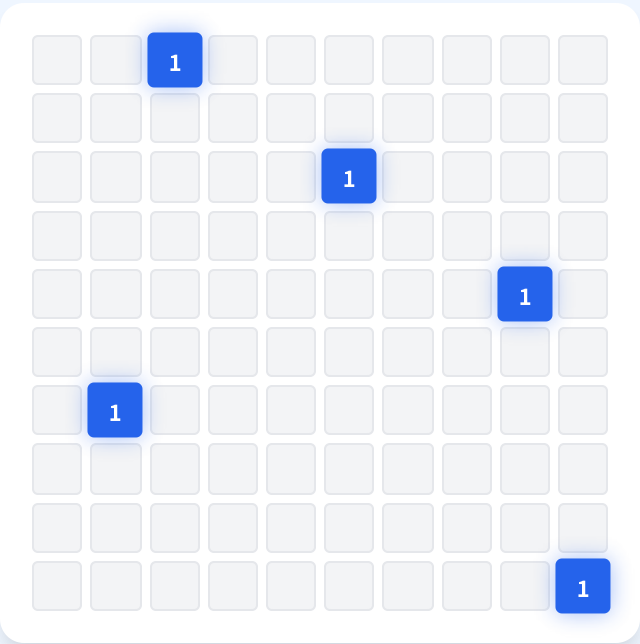
단어: **"사과(Apple)"**

```
[0, 0, ..., 1, 0, ..., 0]
```

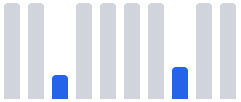
Index: 342

TF-IDF # Bag-of-Words # 검색엔진 역색인

희소 행렬 시각화



값의 분포 (Value Distribution)



CONCEPT | 03. Dense Vector

밀집벡터 (Dense Vector)

> 벡터의 차원을 줄이고, **모든 요소를 실수값**으로 채워
> 단어의 의미를 응축하여 표현한 임베딩(Embedding) 방식

주요 특징 및 장점

의미 일반화 (Generalization)
비슷한 의미의 단어는 벡터 공간에서 가깝게 위치

유사도 연산 용이
코사인 유사도 등을 통해 단어 간 관계 추론 가능

Word Embedding 예시

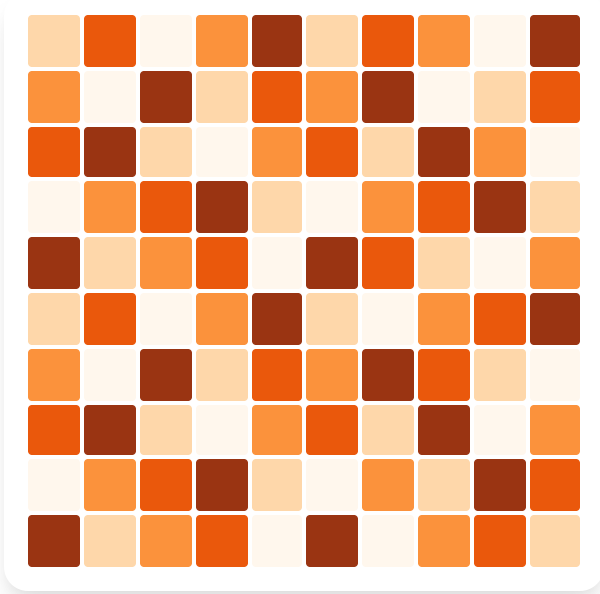
Dimension Size: 50~1024 (Low)

단어: **"사과(Apple)"**

```
[0.12, -0.45, 0.88, 0.03, -0.19, 0.67, ..., -0.32]
```

Word2Vec # GloVe # BERT/GPT

밀집 행렬 시각화 (Heatmap)



임베딩 공간 군집화

COMPARISON | 04. Summary & Application

희소 vs 밀집: 비교와 활용 전략

목적에 따른 적절한 표현 방식 선택이 성능의 핵심

상세 비교 분석

○ 희소 ○ 밀집

비교 항목	희소벡터 (SPARSE)	밀집벡터 (DENSE)
차원 (Dimension)	고차원 (수만~수백만)	저차원 (수십~수천)
메모리 효율	0값 압축 시 효율적 비압축 시 낭비 심함	고정 크기로 예측 가능 모든 값이 실수 (Float)
해석 가능성	직관적 (단어 매칭)	블랙박스 (난해함)

희소벡터 선택 (Best for...)

- 정확한 키워드 매칭이 필수적인 검색
- 학습 데이터가 부족하거나 리소스 제한

밀집벡터 선택 (Best for...)

- 문맥과 의미 파악이 중요한 챗봇/AI
- 이미지, 음성 등 멀티모달 데이터 처리

DATA TRANSFORMATION PIPELINE