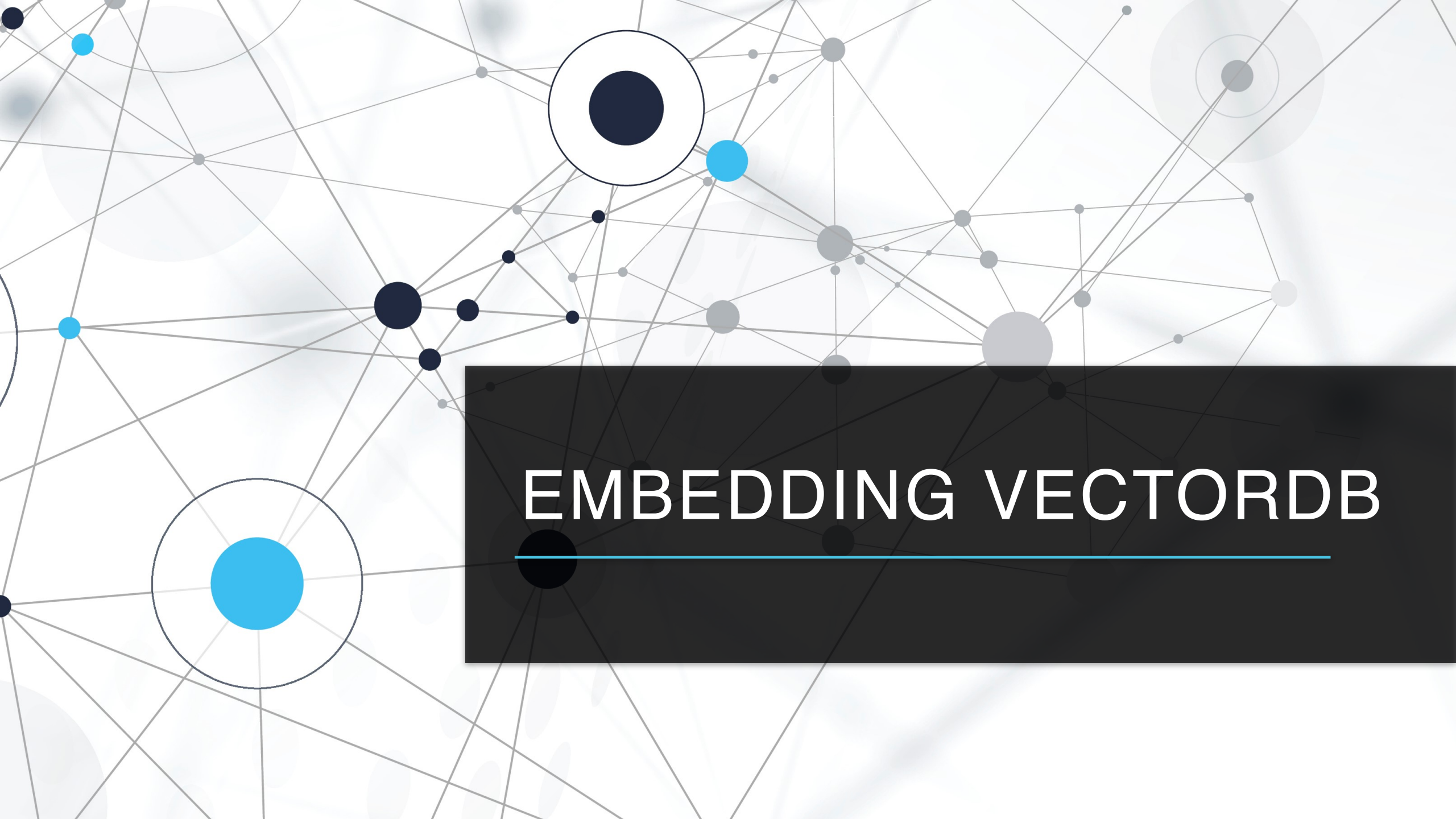

EMBEDDING VECTORDB

EMBEDDING

- Bag of Word
  - Bag of Words란 단어들의 순서는 전혀 고려하지 않고, 단어들의 출현 빈도(frequency)에만 집중하는 텍스트 데이터의 수치화 표현 방법.

  - 순서가 중요한 것이 아니라 특정 단어의 개수가 중요함.

  - 생성 순서

    각 단어에 고유한 정수 인덱스를 부여함.

    각 인덱스의 위치에 단어 토큰의 등장 횟수를 기록한 벡터를 만듬.

EMBEDDING

- TF-IDF Vectorizer
  - TF-IDF Vectorizer는 TF-IDF라는 특정한 값을 사용해서 텍스트 데이터의 특징을 추출하는 방법.
  - TF-IDF는 TF와 IDF를 곱한 값을 한다. 문서를 d, 단어를 t, 문서의 총 개수를 n이라고 표현함.
  - TF(Term Frequency)란 특정 단어가 하나의 데이터 안에서 등장하는 횟 수를 의미함.
  - DF(Document Frequency)는 문서 빈도 값으로, 여기서 특정 단어가 각 문서, 또는 문서들에서 몇 번 등장했는지는 관심가지지 않으며 오직 특정 단어 t가 등장한 문서의 수에만 관심을 가짐.

EMBEDDING

- TF-IDF Vectorizer
  - IDF(Inverse Document Frequency)는 DF 값에 역수를 취해서 구할 수 있으며, 특정 단어가 다른 데이터에 등장하지 않을수록 값이 커진다는 것을 의미함.

$$idf(d,t) = log(\frac{n}{1+df(t)})$$

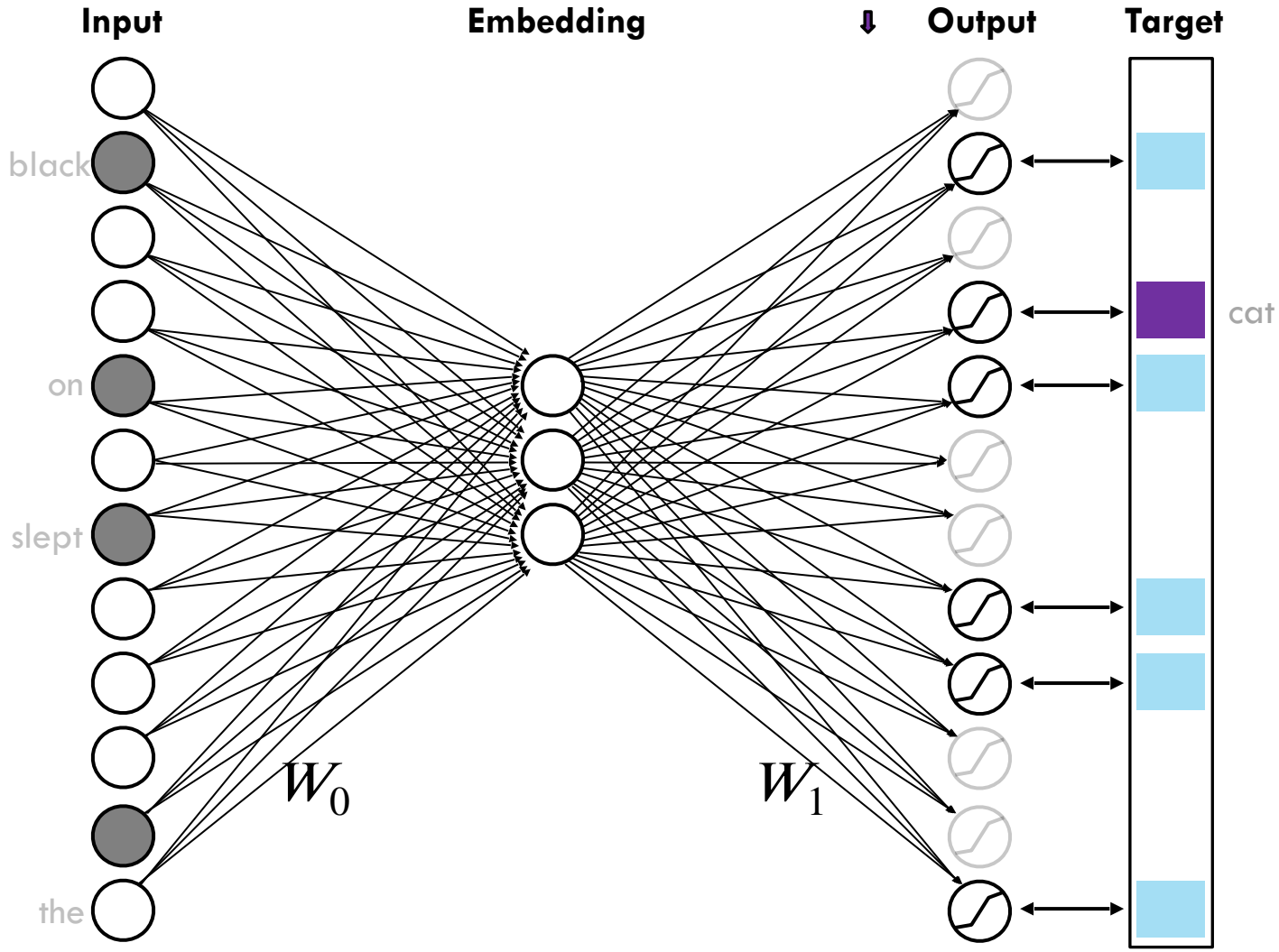
  - log를 사용하지 않았을 때, IDF를 DF의 역수로 사용한다면 총 문서의 수 n이 커질 수록, IDF의 값은 기하급수적으로 커지게 된다. 그래서 log를 사용한다.

EMBEDDING

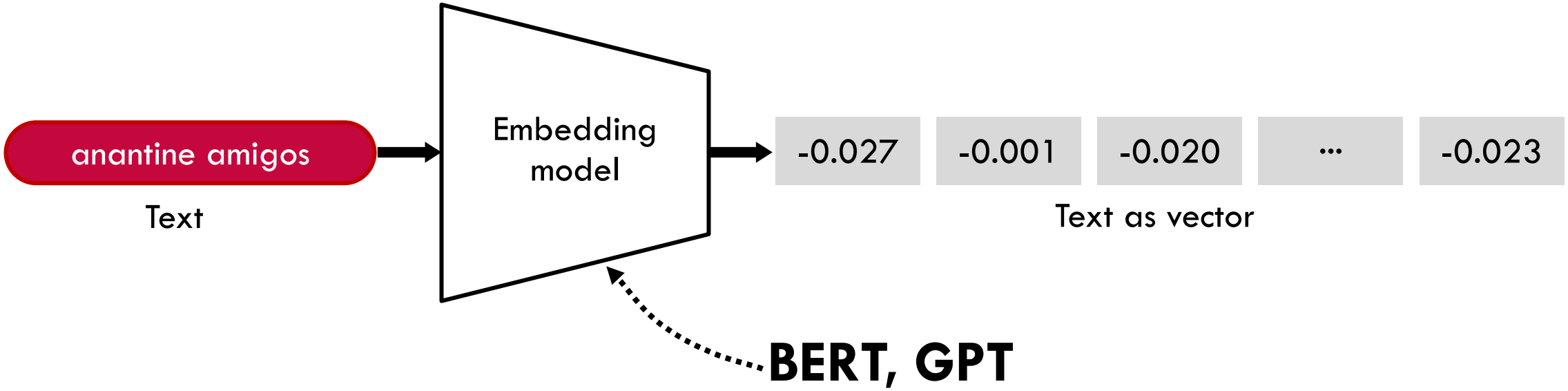
- word2vec
  - 원-핫 인코딩을 통해서 얻은 원-핫 벡터는 표현하고자 하는 단어의 인덱스의 값만 1이고, 나머지 인덱스에는 전부 0으로 됨.
  - 하지만 이러한 표현 방법은 각 단어 벡터간 유의미한 유사성을 표현할 수 없다는 단점이 있음.
  - 단어의 의미를 다차원 공간에 벡터화하는 방법을 사용함

EMBEDDING

- word2vec



EMBEDDING

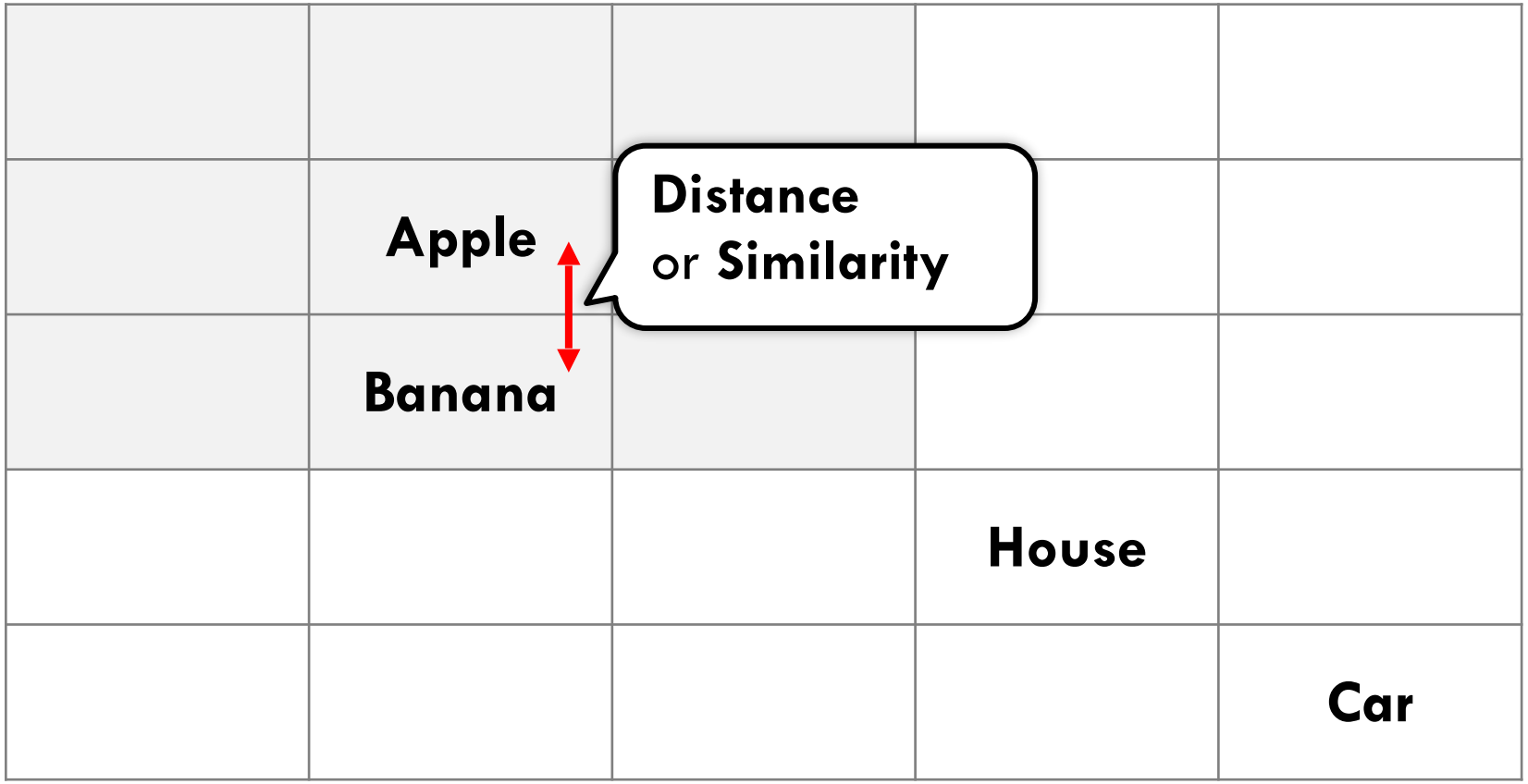

anantine amigos
Text
Embedding model
-0.027
-0.001
-0.020
…
-0.023
Text as vector
BERT, GPT

EMBEDDING

Word	Embedding Vector			
Apple	-0.82	-0.32	…	-0.23
House	0.419	1.28	…	-0.06
Car	…	…	…	…
Orange	-0.74	-1.02	…	1.35

EMBEDDING

- Vector Search



EMBEDDING

- Embedding Model

Overall | Bitext Mining | Classification | Clustering | Pair Classification | **Retrieval** | Reranking | STS | Summarization

Retrieval Leaderboard

- **Metric:** Normalized Discounted Cumulative Gain @ k (ndcg_at_10)
- **Languages:** English

Rank	Model	Average	ArguAna	ClimateFEVER	CQADupstackRetrieval	DBPedia	FEVER
1	multilingual-e5-large	51.43	54.38	25.73	39.68	41.29	82.81
2	e5-large-v2	50.56	46.42	22.21	37.89	44.02	82.83
3	e5-base-v2	50.29	44.49	26.56	38.54	42.23	84.99
4	SGPT-5.8B-weightedmean-msmarco-specb-bitfit	50.25	51.38	30.46	39.4	39.87	78.24
5	e5-large	49.99	49.35	22.4	39.44	42.39	65.03
6	instructor-xl	49.26	55.65	26.54	43.09	40.24	70.03
7	text-embedding-ada-002	49.25	57.44	21.64	41.69	39.39	74.99

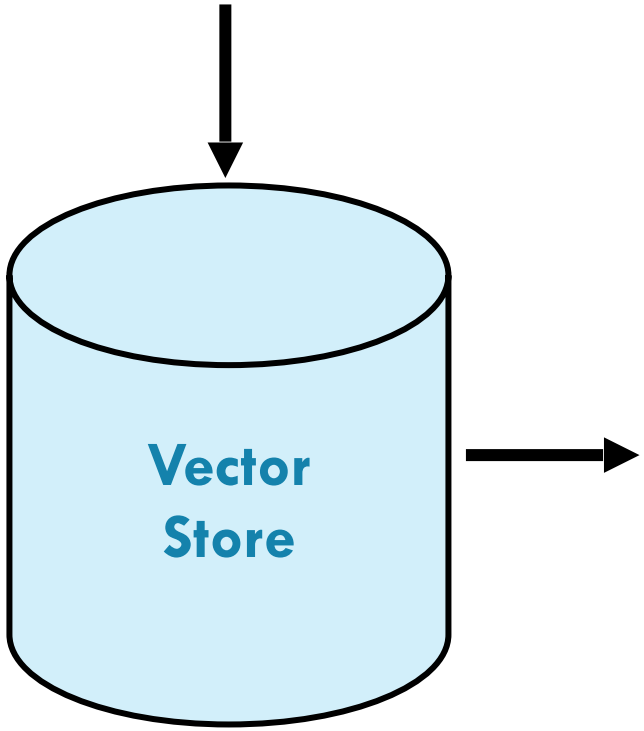
https://huggingface.co/spaces/mteb/leaderboard

EMBEDDING

- Embedding Model 선택
  - 모델의 성능은 데이터셋마다 다르므로, 사용할 데이터에 테스트 해보고 가장 적절한 모델을 선택해야 함
  - 모델의 정확도가 높아야 하는 경우와 민감도 있는 경우에는 파인튜닝한 모델을 고려해야 함
  - OpenAI의 text-embedding-ada-002 모델이 대부분의 경우 일정 수준 이상의 결과를 보여주고, 한국어에도 대체로 성능이 우수한 편임

VECTOR SEARCH

[0.13123, 0.41923, … -0.23112, 0.72153]



[0.13123, 0.41923, … -0.23112, 0.72153]

[0.13123, 0.41923, … -0.23112, 0.72153]

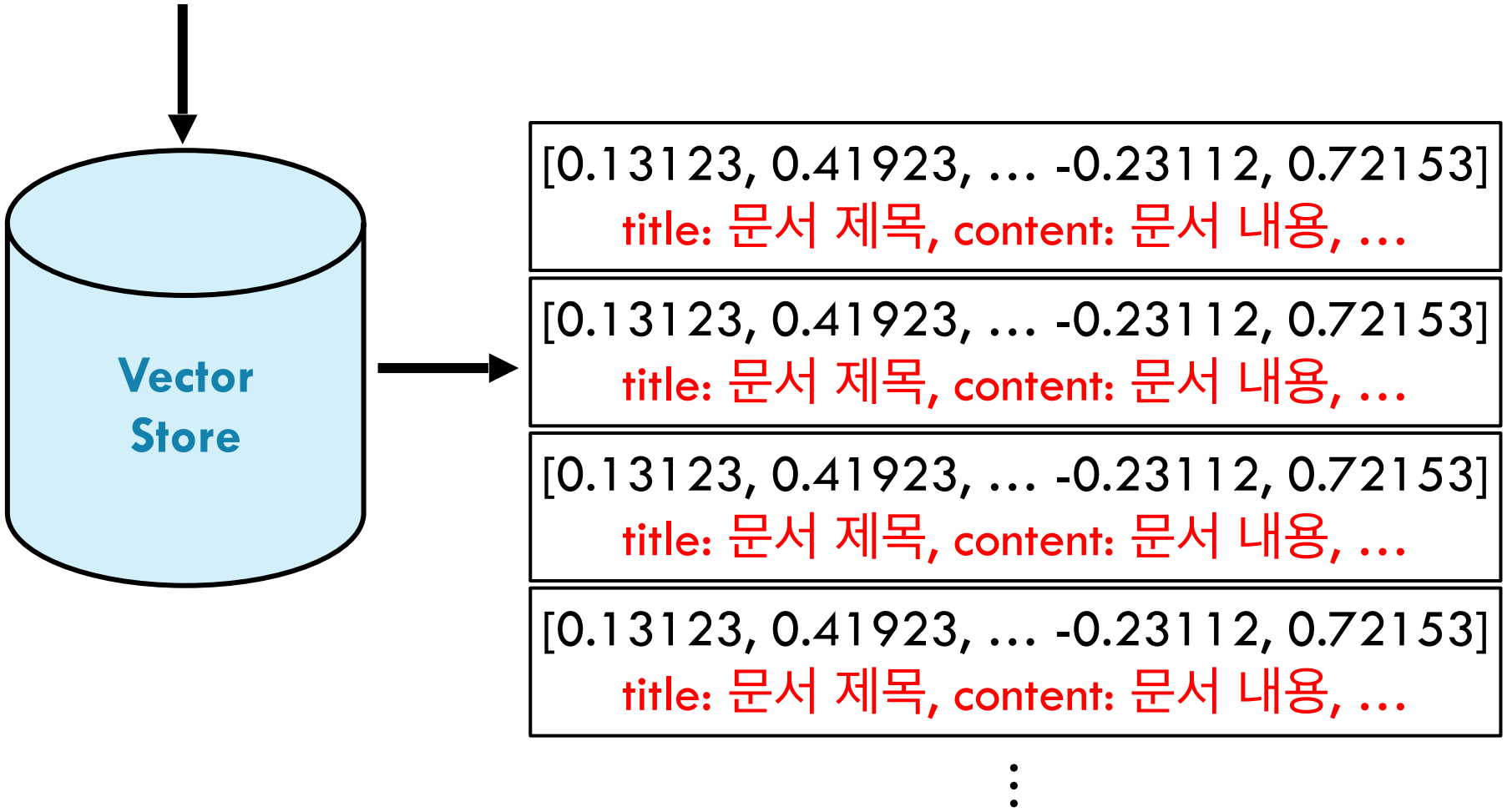
[0.13123, 0.41923, … -0.23112, 0.72153]

[0.13123, 0.41923, … -0.23112, 0.72153]

⋮

VECTOR SEARCH

[0.13123, 0.41923, … -0.23112, 0.72153]



VECTOR DB

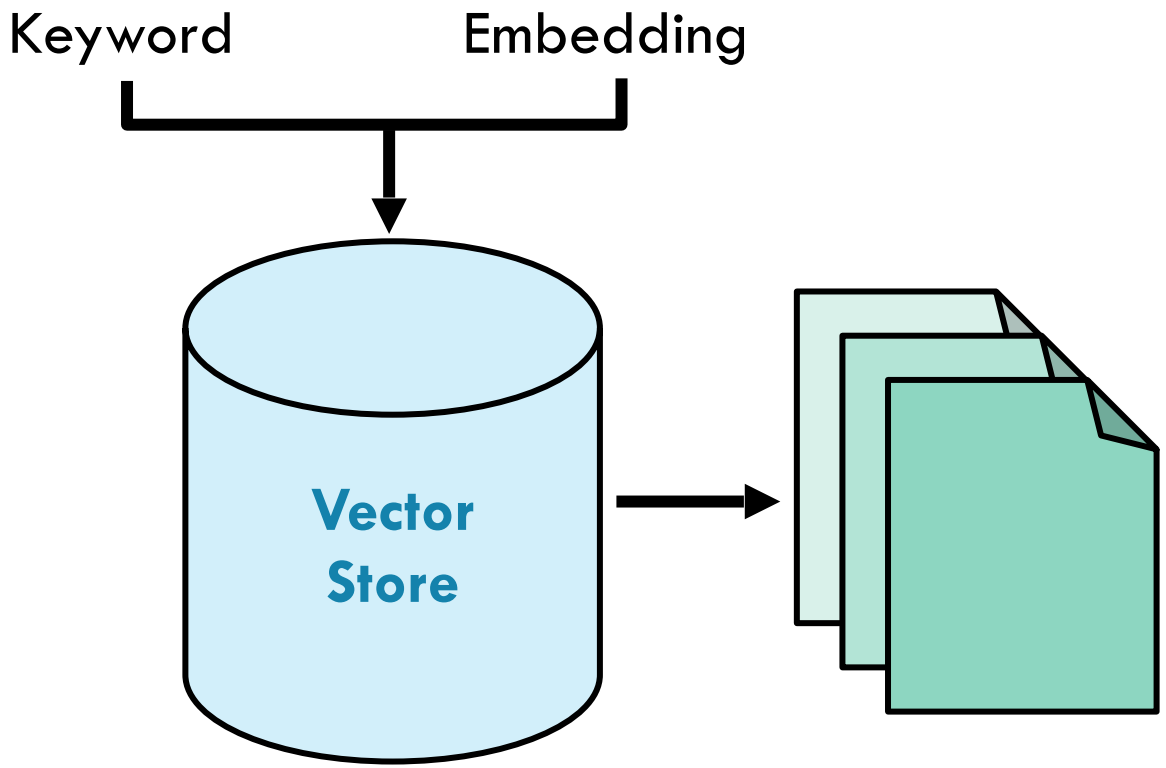
- 메타 데이터와 함께 결과 반환
- 필터링 등을 이용한 하이브리드 검색
- 실시간 인덱싱
- 다양한 인덱싱 및 검색 알고리즘 제공
- 높은 확장성 및 편의 기능 제공

VECTOR DB

- Pincone
- Milvus
- Weaviate
- Qdrant
- Chroma
- Redis
- Elasticsearch

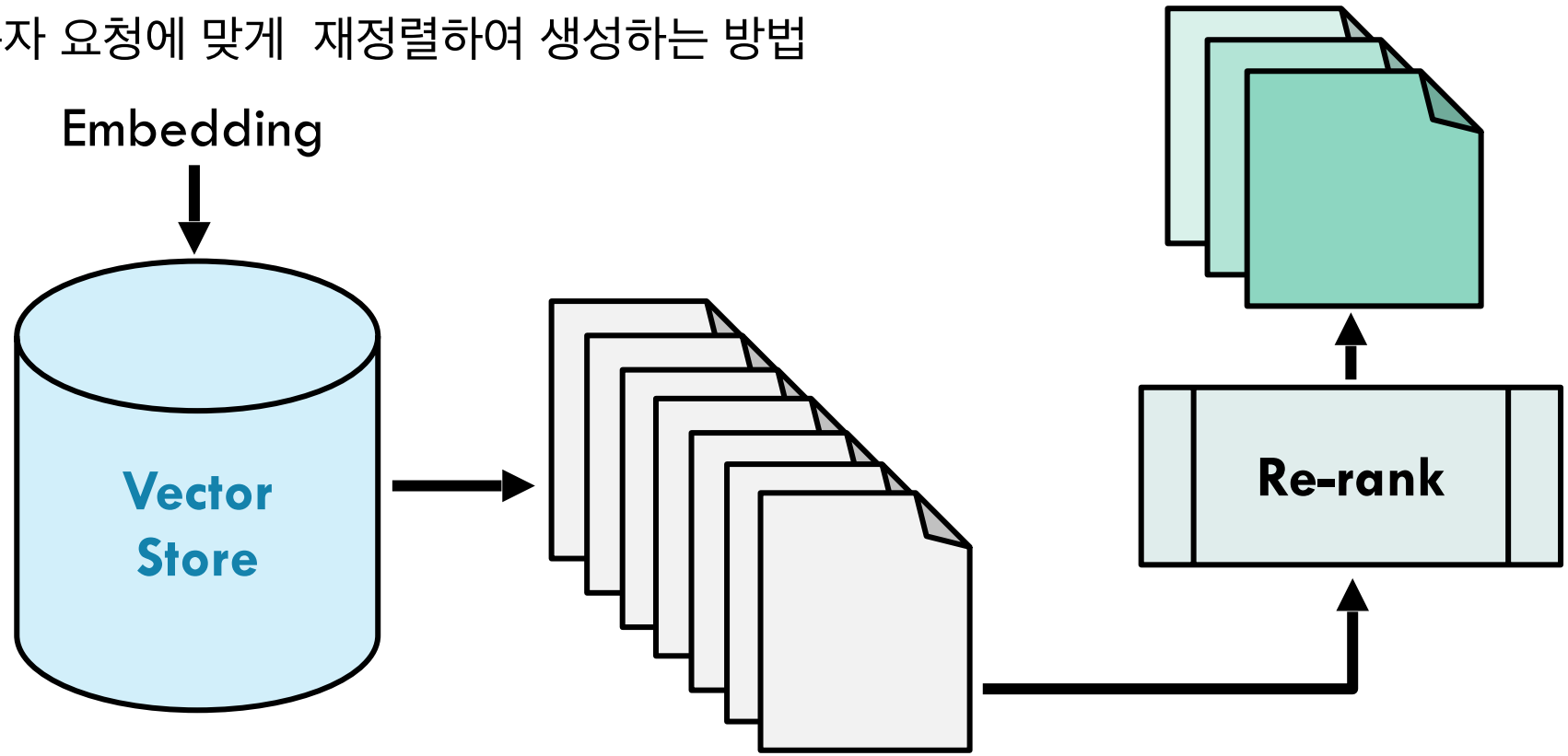
HYBRID SEARCH

- 검색 정확도를 높이기 위해, 키워드 필터링이나 Dense, Sparse 벡터 등을 조합해 검색하는 방식



RE-RANK

- 1차 벡터 서치에 사용한 것과 다른 경량의 임베딩 모델을 사용하거나 또는
  1차 결과를 LLM으로 사용자 요청에 맞게 재정렬하여 생성하는 방법



CHUNKING

- 보통 토큰 수 단위나 단어 단위로 잘라도 되지만, 문장이나 문단 혹은 구조화된 문서라면 섹션 단위로 자르는 둥의 방법을 사용함
- 임베딩 모델과 문서에 따라 적절한 chunking 토큰 수를 찾아야함
- OpenAI의 text-embedding-ada-002 모델의 경우 일반적으로 200~500 토큰 수를 사용함

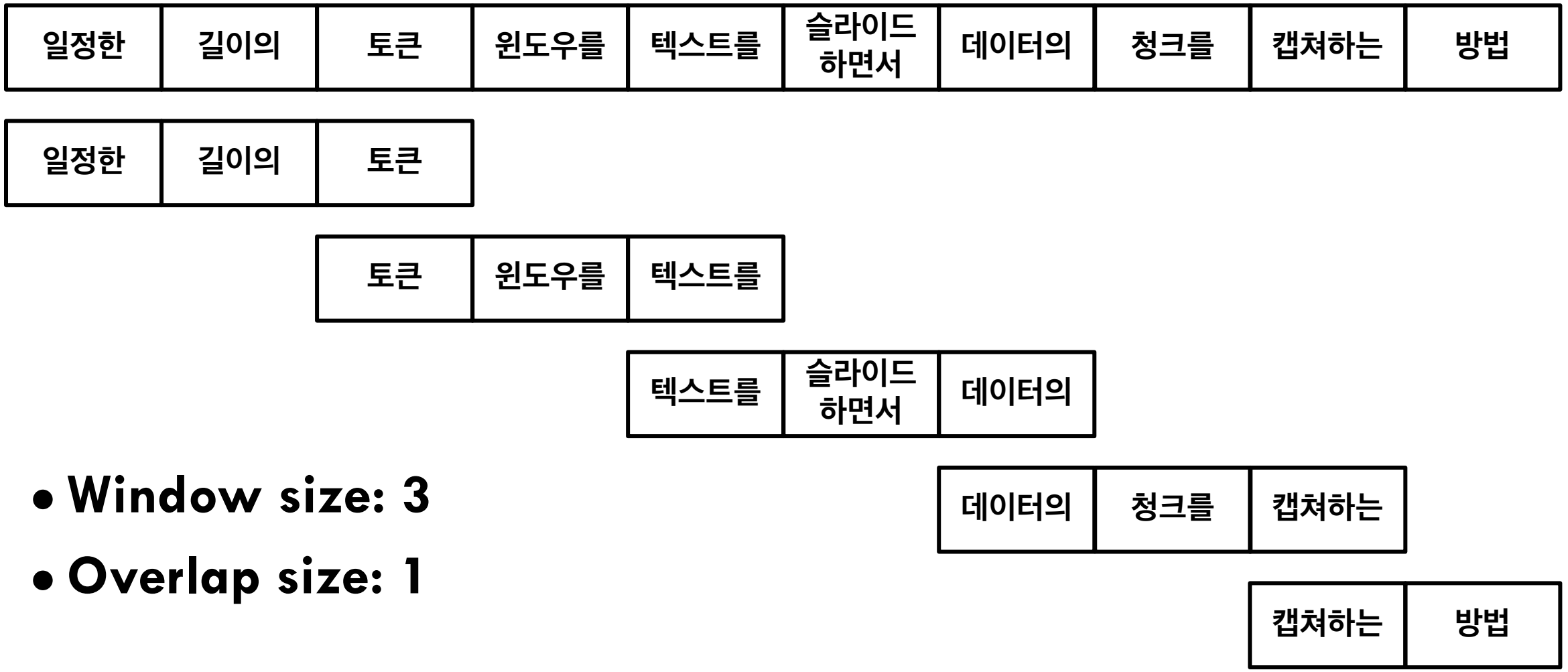
.

OVERLAP & SLIDING

- 텍스트를 분리했을 때 의미가 소실되거나 왜곡되는 것을 방지하고 문맥을 보존하기 위해 사용
- Overlap
  - 각 chunk가 일부의 공통된 데이터를 포함하도록 하는 기법
- Sliding Window
  - 일정한 길이의 토큰(단어) 윈도우로 텍스트를 슬라이드하면서 데이터의 chuck를 수집하는 방법

.

SLIDING



- **Window size: 3**
- **Overlap size: 1**