Building Block Foundation of RAG&RAGAS

MUST LEARN before deep diving to RAG and RAGAS Development

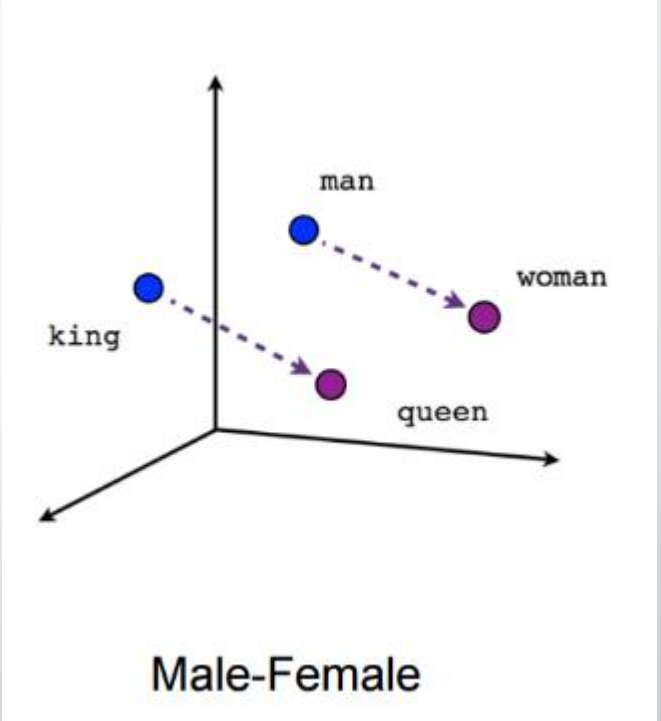
Prepared by Ponce, Bernard C.
DEP, Volunteer

Topic

- What is Embedding?
- Explain Tokens and Chunking
- What is Vector Database?

What is Embedding?

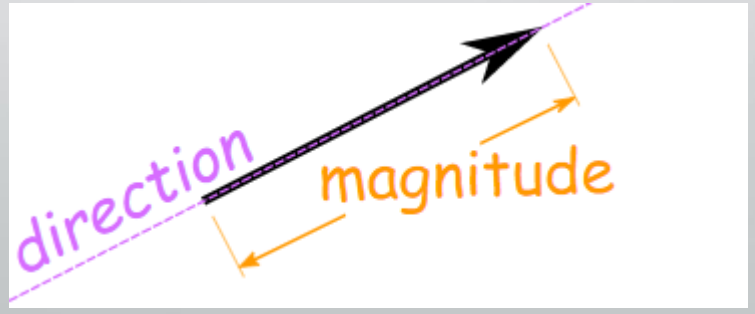
- Embedding is a means of *representing objects like text, images and audio* as points in a continuous vector space where the locations of those points in space are semantically meaningful to machine learning (ML) algorithms.

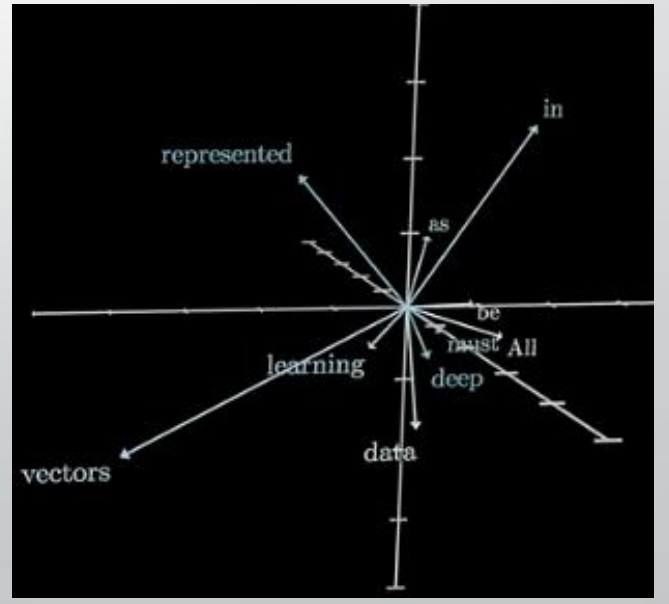


- https://www.ibm.com/think/topics/embedding

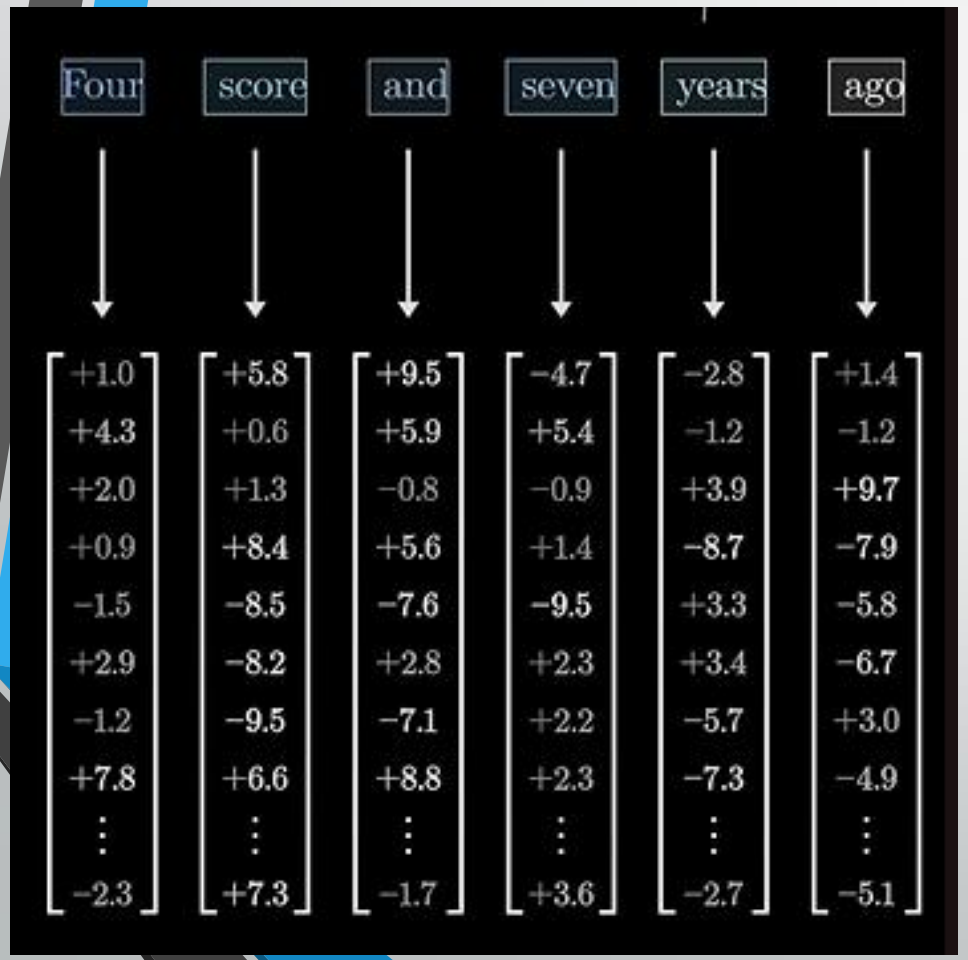
What is Embedding?

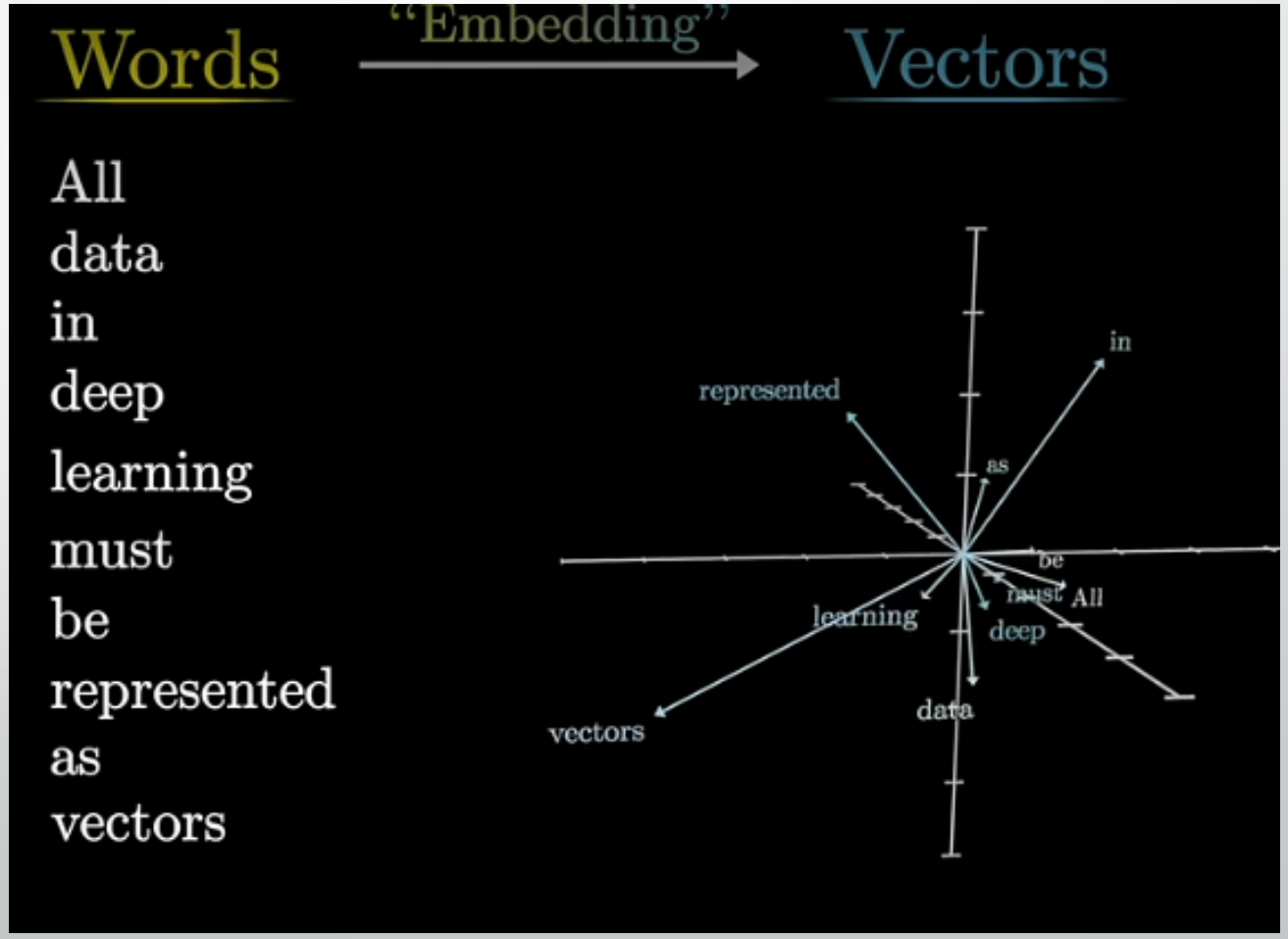
- A **Vector** is like giving your friend two important pieces of information:
  1. **How far to walk (Magnitude):** This is the length of your steps or the distance they need to cover. Think of it as saying, "Walk 10 steps."
  2. **Which way to go (Direction):** This tells them which way to point their feet. You might say, "Walk towards that big tree."





What is Embedding?





Explain Tokens and Chunking

- What is Token?
  - Tokens are words, character sets, or combinations of words and punctuation that are generated by large language models (LLMs) when they decompose text.
  - Small unit a sentence/text data can breakdown.

- https://learn.microsoft.com/en-us/dotnet/ai/conceptual/understanding-tokens

Explain Tokens and Chunking

```python
from sentence_transformers import SentenceTransformer
from typing import Any
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

sentences = ["This is an example sentence", "Each sentence is converted"]

model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
print('Max Length:', model.max_seq_length)

sentence = sentences[0]
tokenizer = model.tokenizer
tokens = tokenizer.tokenize(sentence)
token_ids = tokenizer.encode(sentence)
decoded_sentence = tokenizer.decode(token_ids)

print("Original Sentence:", sentence)
print("Tokens:", tokens)
print("Token IDs:", token_ids)
print("Decoded Sentence:", decoded_sentence)
```

```
Max Length:256
Original Sentence: This is an example sentence
Tokens: ['this', 'is', 'an', 'example', 'sentence']
Token IDs: [101, 2023, 2003, 2019, 2742, 6251, 102]
Decoded Sentence: [CLS] this is an example sentence [SEP]
```

Embedding

Chunk of Text

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Embedding Tokenizer

List of Tokens

[-0.01, 1.321, 0.51, -0.01, 1.321, 0.51, ..., 0.23, 0.153, 0.213, 0.23, 0.153, 0.213]

Embedding

Chunk of Text

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Embedding Tokenizer

List of Tokens

[-0.01, 1.321, 0.51, -0.01, 1.321, 0.51, ..., 0.23, 0.153, 0.213, 0.23, 0.153, 0.213]

Problem when considering the Chunk

Max Sequence Length is 12 Token

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

If Sequence of Embedding model only generated 7 Token, there have a loss of data.

Lorem Ipsum is simply dummy text of ~~the printing and typesetting industry.~~

Missing Data?

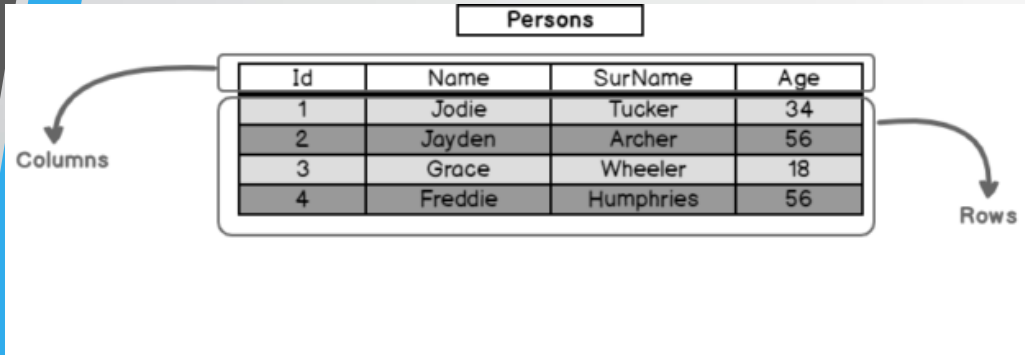
Problem when considering the Chunk

Vector Database

- Database - is an **organized collection of information** that is structured to make it easy to **store, manage, and retrieve** data efficiently.
- Vector(Embedding Vector) - is a **dense numerical representation** that captures the **semantic meaning** of data in a way that similar items have **nearby vectors** in a multi-dimensional space.

Vector Database - Type of Databases

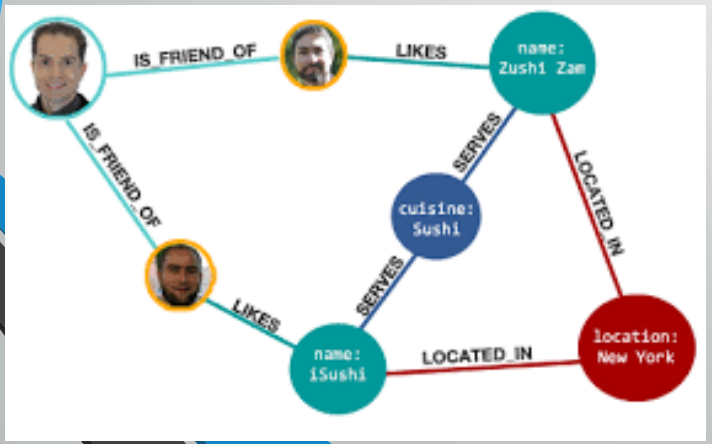
- SQL



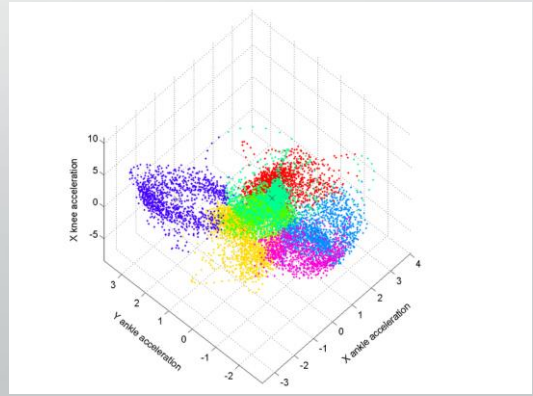
- NoSQL

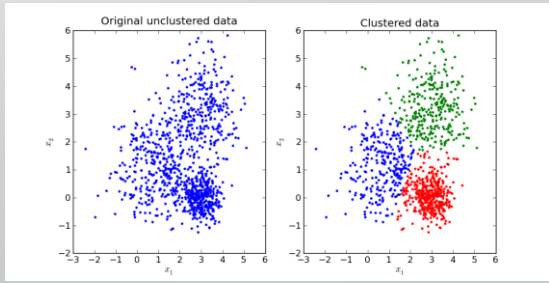
```
{
  "string": "Hi",
  "number": 2.5,
  "boolean": true,
  "null": null,
  "object": { "name": "Kyle", "age": 24 },
  "array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],
  "arrayOfObjects": [
    { "name": "Jerry", "age": 28 },
    { "name": "Sally", "age": 26 }
  ]
}
```

- Graph



- Vector





Vector Database

- In-Memory (Custom)
- 3rd Party Vector Database
  - MongoDb
  - ChromeDb
  - FAISS
  - Weaviate
  - Pinecone

Vector Database

```python
documents : list[str] = [
    "The quick brown rabbit jumps over the lazy frogs.",
    "A fast tan hare leaps above sleepy toads.",
    "Artificial intelligence is transforming various industries.",
    "AI is having a significant impact on the future of work.",
    "The weather today is sunny and warm.",
    "It's a beautiful day with clear skies and high temperatures.",
    "The old wooden door creaked loudly in the wind.",
    "A fluffy white cat slept peacefully on the sunny windowsill.",
    "Freshly baked bread filled the kitchen with a warm aroma.",
    "The little girl giggled as she chased butterflies in the garden.",
    "Heavy rain poured down, creating puddles on the pavement.",
    "A tall green tree swayed gently in the light breeze.",
    "The scientist carefully mixed the colorful liquids in the lab.",
    "A delicious cup of coffee helped him start his busy day.",
    "Bright stars twinkled in the dark night sky.",
    "The new book quickly became a bestseller."
]

document_embeddings = model.encode(documents)
```

```python
document_embeddings = model.encode(documents)
vector_database = list(zip(documents, document_embeddings))

print(f"Shape of document embeddings: {document_embeddings.shape}")
# Output will be something like: (16, 384) - 16 documents, each with a 384-dimensional vector
```

```
Max Length: 256
Shape of document embeddings: (16, 384)
```

Vector Database

```python
from typing import Any
from sklearn.metrics.pairwise import cosine_similarity

def search_documents(query: str, vector_database: list[tuple[str, Any]], model, top_n=3):
    query_embedding = model.encode(query)
    similarity_scores = cosine_similarity([query_embedding], [embedding for doc, embedding in vector_database])[0]
    ranked_results = sorted(zip(vector_database, similarity_scores), key=lambda x: x[1], reverse=True)
    return [(doc, score) for (doc, _), score in ranked_results[:top_n]]

def print_result(results : list[tuple[str, Any]], query):
    print(f"\nTop relevant documents for query: '{query}'")
    for doc, score in results:
        print(f"- '{doc}' (Score: {score:.4f})")
```

Vector Database

```
query = "What are the impacts of AI?"
results = search_documents(query, vector_database, model)
print_result(results, query)

query_weather = "Tell me about the weather."
results = search_documents(query_weather, vector_database, model)
print_result(results, query_weather)

query_cat = "A cat is sleeping."
results = search_documents(query_cat, vector_database, model)
print_result(results, query_cat)
```

```
Top relevant documents for query: 'What are the impacts of AI?'
- 'AI is having a significant impact on the future of work.' (Score: 0.7218)
- 'Artificial intelligence is transforming various industries.' (Score: 0.5749)
- 'Bright stars twinkled in the dark night sky.' (Score: 0.0739)

Top relevant documents for query: 'Tell me about the weather.'
- 'The weather today is sunny and warm.' (Score: 0.6475)
- 'It's a beautiful day with clear skies and high temperatures.' (Score: 0.6388)
- 'Heavy rain poured down, creating puddles on the pavement.' (Score: 0.2741)

Top relevant documents for query: 'A cat is sleeping.'
- 'A fluffy white cat slept peacefully on the sunny windowsill.' (Score: 0.6308)
- 'A fast tan hare leaps above sleepy toads.' (Score: 0.1334)
- 'The quick brown rabbit jumps over the lazy frogs.' (Score: 0.1047)
```

Thank you

Prepared by:

Ponce, Bernard C.

DEP AI Group Study Volunteer