

Modern Document Data Modelling beyond the AI Era

Piti Champeethong (Fyi/พี)

PyLanna (Python FB User Group)
MongoDB Thailand FB User Group
<https://github.com/ninefyi>

National Coding Day 2026 (24-Jan-2026)



Introduction

- The Shift in Data Ecosystem
 - Common (Text, Number, Date)
 - Vector
 - Metadata
- Challenges of Integrating AI with Legacy Systems

MongoDB Schema Design Patterns for AI

- Data that is used together should be stored together.
- The Attribute Pattern & Metadata Management
- The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern
- Chat History with Bucket Pattern

MongoDB Schema Design Patterns for AI

➤ Data that is used together should be stored together.

- The Attribute Pattern & Metadata Management
- The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern
- Chat History with Bucket Pattern

Embedding Data Model

```
{  
  "topic": "ปี 2026 มือถือยี่ห้อไหน มีน้ำหนักรเบา",  
  "topic_vector": [0.123, -0.456, 0.789, 0.012, ...]  
  "top_comments": [  
    { "user": "JoJoBar", "comment": "EyePhone Ah 9" },  
    { "user": "Kaken", "comment": "Somesongs Slim 10"}  
  ]  
}
```

MongoDB Schema Design Patterns for AI

- Data that is used together should be stored together.

➤ The Attribute Pattern & Metadata Management

- The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern
- Chat History with Bucket Pattern

Simple Pattern

```
{  
  "name": "Modern Data Science Laptop",  
  "description": "High-performance laptop for AI"  
  "description_vector": [0.12, -0.05, 0.44, ...],  
  "brand" : "TechCorp",  
  "ram_gb" : 64,  
  "gpu", "v": "RTX 4090"  
}
```

Simple Pattern

```
{  
  "name": "Modern Data Science Laptop",  
  "description": "High-performance laptop for AI"  
  "description_vector": [0.12, -0.05, 0.44, ...],  
  "brand" : "TechCorp",  
  "ram_gb" : 64,  
  "gpu", "v": "RTX 4090"  
}
```


Attribute Pattern (Key and Value)

```
{  
  "name": "Modern Data Science Laptop",  
  "description": "High-performance laptop for AI",  
  "description_vector": [0.12, -0.05, 0.44, ...],  
  "attributes": [  
    {"k": "brand", "v": "TechCorp"},  
    {"k": "ram_gb", "v": 64},  
    {"k": "gpu", "v": "RTX 4090"}  
  ],  
}
```

MongoDB Schema Design Patterns for AI

- Data that is used together should be stored together.
- The Attribute Pattern & Metadata Management
 - The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern
- Chat History with Bucket Pattern

Extended Reference Pattern



Extended Reference Pattern

```
{  
  "_id": "auth123",  
  "name": "Fyi",  
  "expertise": "MDB"  
}
```

authors

```
{  
  "title": "Modern Document Data Modelling",  
  "body": "In the era of AI, schema design is  
still crucial...",  
  "title_vector": [0.01, 0.22, ...],  
  "author": {  
    "author_id": "auth123",  
    "display_name": "Fyi"  
  },  
}
```

articles

MongoDB Schema Design Patterns for AI

- Data that is used together should be stored together.
- The Attribute Pattern & Metadata Management
- The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern
- Chat History with Bucket Pattern

The Outlier Pattern



The Outlier Pattern

```
{  
  books  
  "_id": 888,  
  "title": "Data Modelling for AI 2026",  
  "author": "Fyi",  
  "is_outlier": true, // Flag บอกว่าเป็นเอกสารขนาดใหญ่  
  "summary": "This is a 1,000-page book about AI...",  
  "total_chunks": 250,  
}
```

book_outliers

```
[ {  
  "book_id": 888,  
  "chunk_index": 0,  
  "content": "Introduction to Referencing Model...",  
  "vector": [0.11, 0.22, ...], // Embedding ของเฉพาะส่วนนี้  
  "page": { from: 1, to: 4 }  
},  
{  
  "book_id": 888,  
  "chunk_index": 1,  
  "content": "Advanced Referencing Model..",  
  "vector": [0.33, 0.44, ...],  
  "page": { from: 5, to: 10 }  
}]
```



MongoDB Schema Design Patterns for AI

- Data that is used together should be stored together.
- The Attribute Pattern & Metadata Management
- The Extended Reference Pattern
- Handling Large Content: The Outlier Pattern

➤ Chat History with Bucket Pattern

Bucket Pattern

```
{
  "_id": "session_123_bucket_1",
  "session_id": "session_123",
  "user_id": "user_456",
  "bucket_index": 1,
  "message_count": 50,
  "messages": [ ],
  "summary": "User ถามเรื่องสรุปโปรเจค
MongoDB",
  "ts": "2026-01-21"
}
```



```
{
  "role": "user",
  "content": "สวัสดีครับ ช่วยสรุปโปรเจค MongoDB
ให้หน่อย",
  "ts": "2026-01-21"
},
{
  "role": "assistant",
  "content": "ได้เลยค่ะ โปรเจคนี้เน้นไปที่การใช้
Hybrid Search...",
  "ts": "2026-01-21",
  "tokens": 150
}
```

Hybrid, Vector Search Indexes & Re-ranking

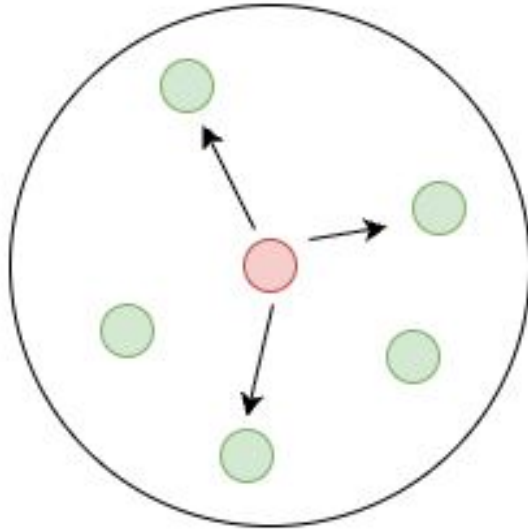
- Vector Search: (k-NN, ANN, ENN)
- Hybrid Search: Full Text (BM25) + Vector Search
- Boosting Accuracy with Re-ranking (RRF)

Hybrid, Vector Search Indexes & Re-ranking

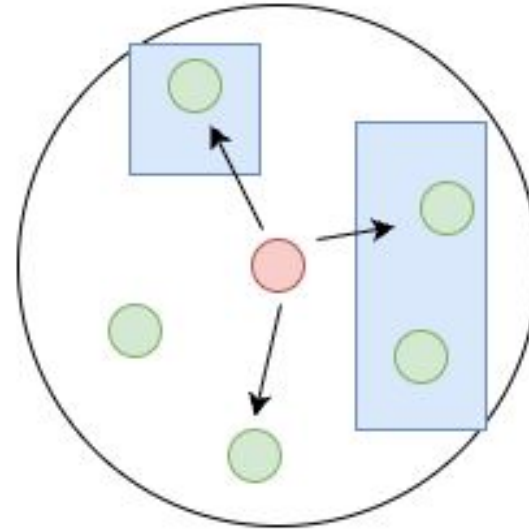
➤ Vector Search (k-NN, ANN, ENN)

- Hybrid Search: Full Text (BM25) + Vector Search
- Boosting Accuracy with Re-ranking (RRF)

k-NN vs ENN vs ANN : Nearest Neighbor



k-NN/ENN



ANN

k-NN vs ENN vs ANN

คุณสมบัติ	k-NN / Exact-NN	Approximate-NN
ความแม่นยำ	100% (ได้จุดที่ใกล้เคียงที่สุดจริงๆ)	~90-99% (มีโอกาสพลาดเล็กน้อย)
ความเร็ว	ช้ามาก (เมื่อข้อมูลเยอะ)	เร็วมาก (คงที่แม้ข้อมูลมหาศาล)
การใช้งาน	งานวิจัย, ข้อมูลขนาดเล็ก	ระบบ Search, ChatGPT, YouTube
เปรียบเทียบ	เดินวัดสายวัดทีละจุด	มองด้วยตาเปล่าแล้วกะระยะเอา

Hybrid, Vector Search Indexes & Re-ranking

- Vector Search: (k-NN, ANN, ENN)

➤ Hybrid Search: Full Text (BM25) + Vector Search

- Boosting Accuracy with Re-ranking (RRF)

Hybrid Search : Full Text + Vector Search

```
{  
  "title": "Modern Document Data Modelling",  
  "body": "In the era of AI, schema design is still  
crucial...", // Full Text Search  
  "title_vector": [0.01, 0.22, ...], // Vector Search  
  "author": {  
    "author_id": "auth123", // Full Text Search  
    "display_name": "Fyi" // Full Text Search  
  },  
}
```


Hybrid, Vector Search Indexes & Re-ranking

- Vector Search: (k-NN, ANN, ENN)
- Hybrid Search: Full Text (BM25) + Vector Search
- Boosting Accuracy with Re-ranking (RRF)

Boosting Accuracy with Re-ranking (RRF)

$$RRFscore(d) = \sum_{r \in R} \frac{1}{k + r(d)}$$

d คือ document

$r(d)$ คือ rank of document ที่มาจากการค้นหา (เช่น 1, 2, 3)

k คือ ค่าคงที่ (นิยมใช้ 60)

Boosting Accuracy with Re-ranking (RRF)

ระบบการค้นหา	อันดับของเอกสาร A	อันดับของเอกสาร B
BM25 (Text)	อันดับ 1	อันดับ 5
Vector Search	อันดับ 10	อันดับ 2

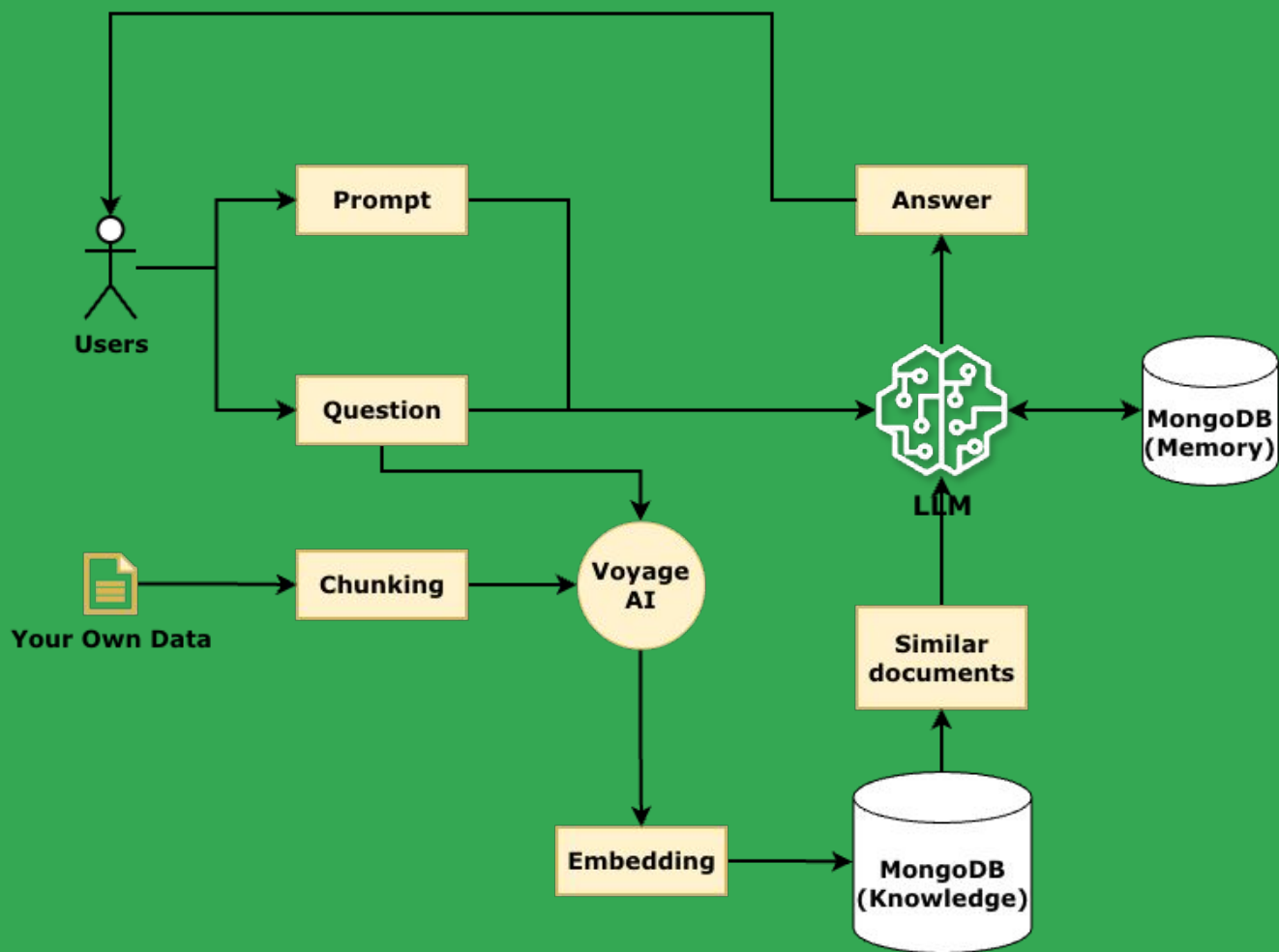
Boosting Accuracy with Re-ranking (RRF)

คำนวณคะแนนของเอกสาร A:

- จาก BM25: $\frac{1}{60+1} = \frac{1}{61} \approx 0.01639$
- จาก Vector: $\frac{1}{60+10} = \frac{1}{70} \approx 0.01428$
- คะแนนรวม A: $0.01639 + 0.01428 = \mathbf{0.03067}$

คำนวณคะแนนของเอกสาร B:

- จาก BM25: $\frac{1}{60+5} = \frac{1}{65} \approx 0.01538$
- จาก Vector: $\frac{1}{60+2} = \frac{1}{62} \approx 0.01612$
- คะแนนรวม B: $0.01538 + 0.01612 = \mathbf{0.03150}$



References

- <https://github.com/ninefyi/ncd2026>
- <https://www.mongodb.com/docs/manual/data-modeling>
- <https://www.mongodb.com/docs/atlas/atlas-vector-search/vector-search-overview/>

ขอบคุณครับ!!!!