Spring AI: Beginner to Guru

Generative AI with Spring

Overview of Retrieval Augmented Generation - RAG

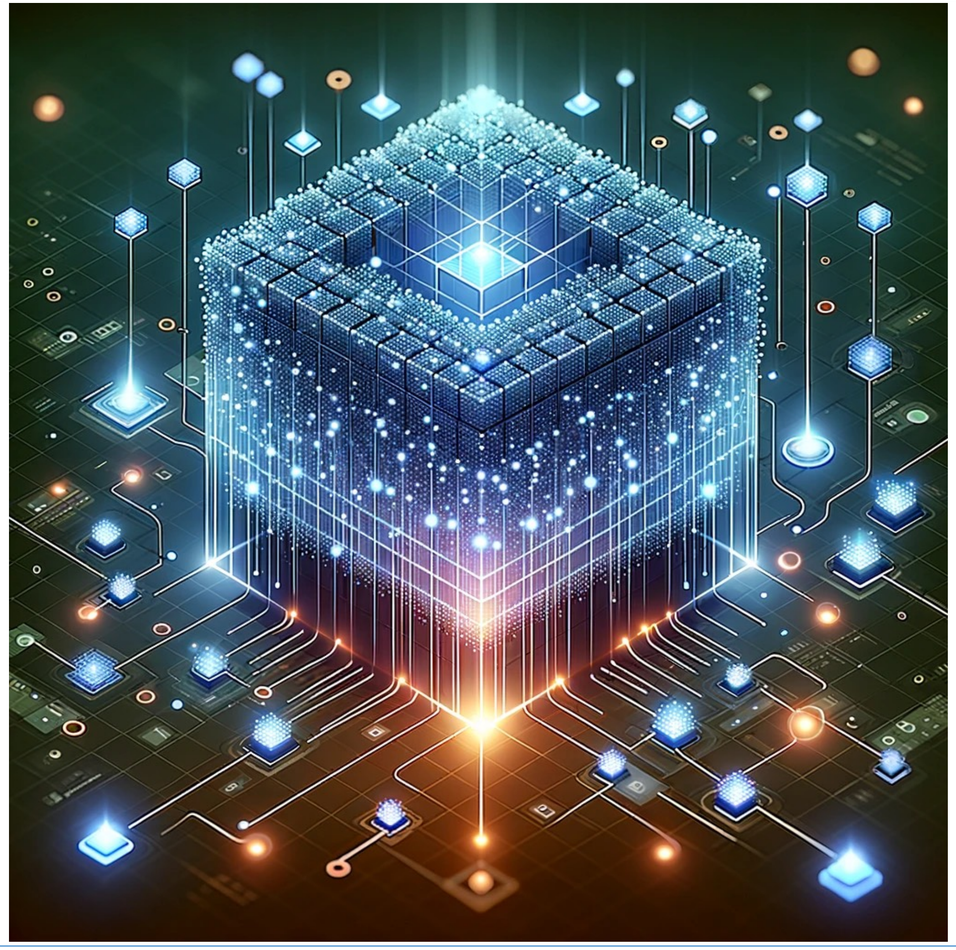
Overview of Retrieval Augmented Generation - RAG

- RAG is a powerful technique used in LLMs to improve their performance on tasks
- It combines the retrieval from databases to produce more accurate and relevant outputs
- The retrieval component searches through a large amount of data to find relevant information to the query
- The generation component (LLM) generates the response based on the query and retrieved information

Vector Databases and Embeddings

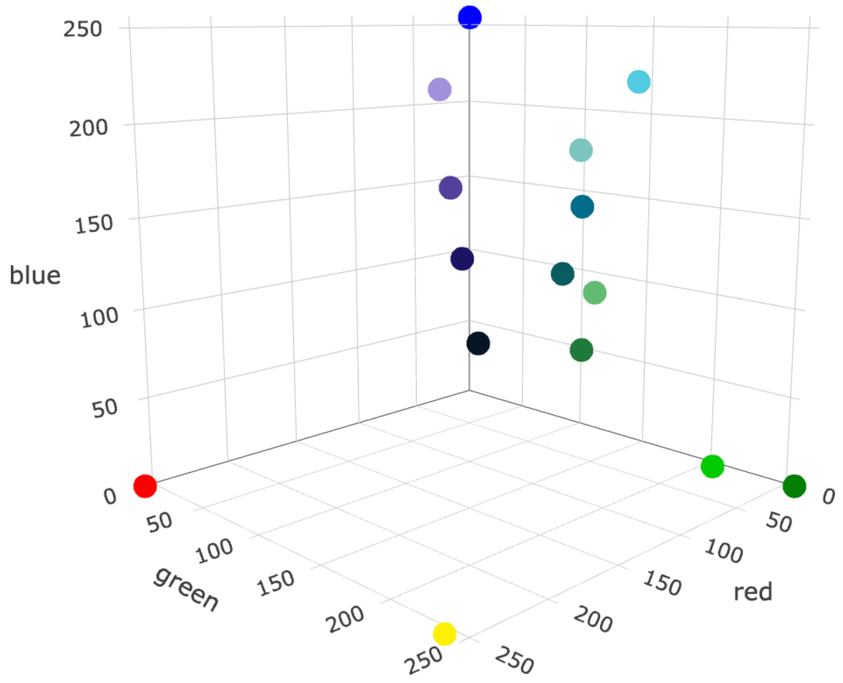
- RAG typically uses vector databases over a large set of data
- Text from the data is converted into vector representations called embeddings
- Embeddings are dense numerical representations of textual data in a high-dimensional space
- The embeddings are persisted into the vector database
- The input query is used to query the vector database for related data
- The vector database provides results ‘near’ the terms in the query
- The response of the LLM is generated from the users query and the vector database query results

- Asking ChatGpt-4 to create an image visualizing how data is stored in a vector database created the image to the left
  - Interesting image, but does a poor job of illustrating how data is stored in a vector database
- Data is optimized for semantic searches allowing relevant data to be retrieved even if it is not an exact match



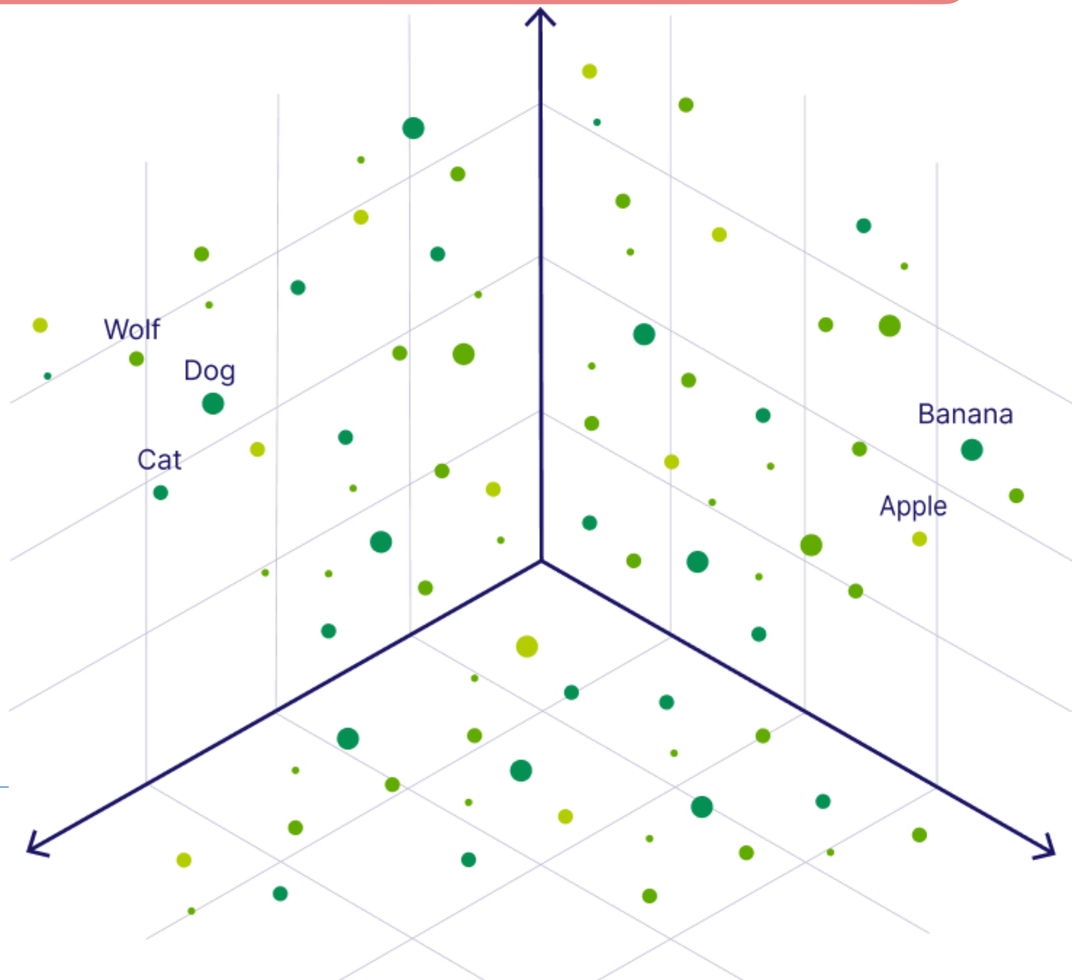
Vector Visualization - Source Weaviate

- RGB System - Red, Green, Blue
- Vectors map the relationships between the various color combinations



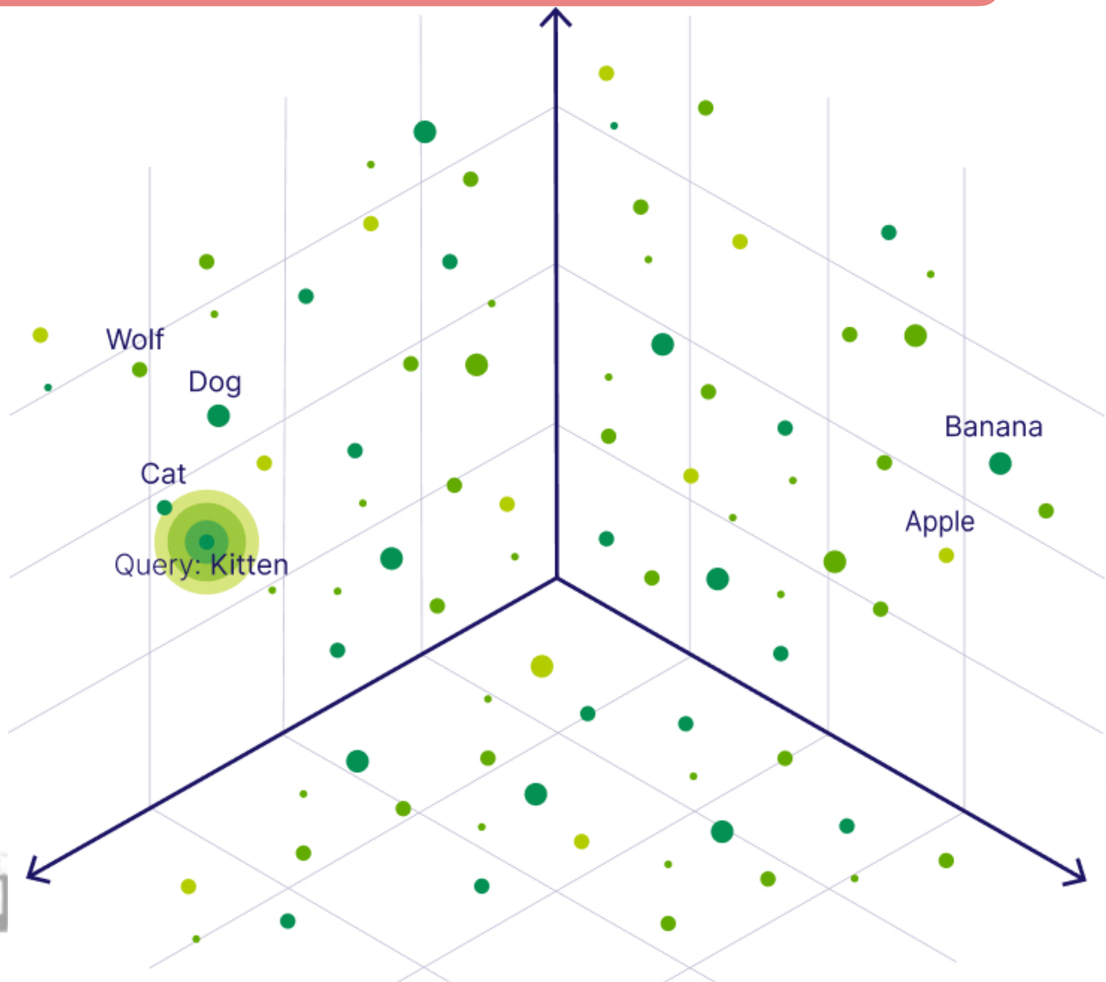
Vector Visualization - Source Weaviate

- Word Vectors
- Wolf, Dog, Cat - close to each other
- Apple, Banana - close to each other



Vector Visualization - Source Weaviate

- Word Vector Searches
- Query for Kitten shows it is close to Cat, Dog further away, Wolf further away
- Apple, Banana not close at all
- **NOTE:** Word searches still work with typos or misspellings



Traditional Search VS Semantic Search

```
SELECT question
FROM JeopardyQuestions
WHERE (question LIKE '%dog%' OR
       question LIKE '%cat%' OR
       question LIKE '%wolf%' OR
    (… and so on)
```

```
{
  Get {
    JeopardyQuestions (
      nearText: { concepts: ["animals"] }
    ) { question }
  }
}
```

Source: Weaviate

SPRING FRAMEWORK

GURU