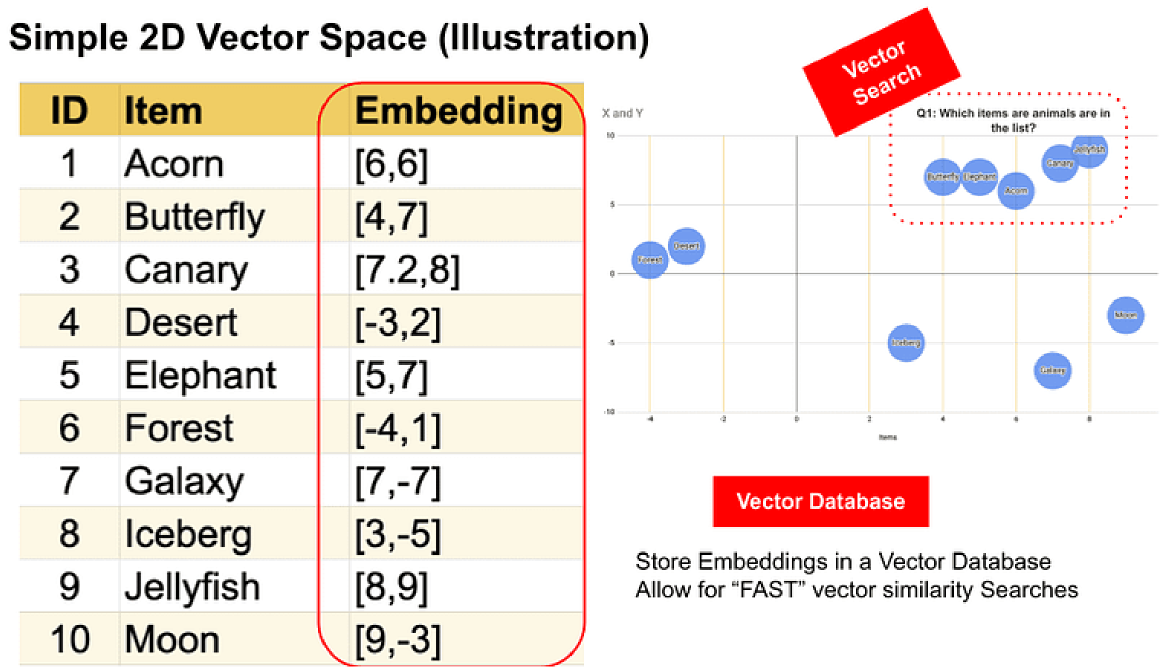
Snowflake Embeddings and Vector Search

Snowflake's advanced capabilities extend beyond traditional data warehousing and analytics. With the advent of Snowflake Cortex, powerful tools like embedding generation and vector search have become accessible for unlocking deeper insights from unstructured text data.

Understanding Embeddings

Embeddings are numerical representations of words or phrases in a high-dimensional space.



Simple 2D Vector Space (Illustration)

ID	Item	Embedding
1	Acorn	[6,6]
2	Butterfly	[4,7]
3	Canary	[7.2,8]
4	Desert	[-3,2]
5	Elephant	[5,7]
6	Forest	[-4,1]
7	Galaxy	[7,-7]
8	Iceberg	[3,-5]
9	Jellyfish	[8,9]
10	Moon	[9,-3]

Store Embeddings in a Vector Database
Allow for "FAST" vector similarity Searches

Take a list of Items for example. The corresponding embeddings captures different aspects of meaning in a higher dimensional space, allowing us to compare the semantic similarity between pieces of text. Snowflake's **EMBED_TEXT_768 function**, part of the Cortex suite, generates these embeddings using a pre-trained language model.

Hands-On Example: Everyday Objects

Let's explore embeddings with a simple example:

```
CREATE OR REPLACE DATABASE EmbeddingsDB;


USE EmbeddingsDB;

CREATE TABLE EverydayObjects (
  object_id INT PRIMARY KEY,
  object_name VARCHAR(255) NOT NULL
);

INSERT INTO EverydayObjects (object_id, object_name)
VALUES
  (1, 'Chair'),
  (2, 'Table'),
  (3, 'Book'),
  (4, 'Pen'),
  (5, 'Phone'),
  (6, 'Computer'),
  (7, 'Car'),
  (8, 'House'),
  (9, 'Tree'),
  (10, 'Flower'),
  (11, 'Food'),
  (12, 'Water'),
  (13, 'Clothes'),
  (14, 'Shoes'),
  (15, 'Key'),
  (16, 'Lamp'),
  (17, 'Bed'),
  (18, 'Door'),
  (19, 'Window'),
  (20, 'Clock'),
  (21, 'Beaver'),
  (22, 'Hunt'),
  (23, 'Galaxy');
```

This script creates a table EverydayObjects and populates it with 23 common objects.

Next, let's see how traditional string matching falls short in capturing semantic relationships:

```
select * from everydayobjects where object_name like 'P%'; -- Returns objects starting with 'P'

select * from everydayobjects where object_name like 'Writing'; -- Doesn't return 'Pen'
select * from everydayobjects where object_name like 'animal'; -- Doesn't return 'Beaver'
```

String matching only works for exact or partial keyword matches. It cannot understand that "Pen" is related to "Writing" or that "Beaver" is an "animal."

Generating Embeddings

To address this, we'll use SNOWFLAKE.CORTEX.EMBED_TEXT_768 to create embeddings:

```
CREATE OR REPLACE TABLE everydayobjects_embeddings AS
SELECT *, SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', object_name)
AS object_embedding
FROM EverydayObjects;
```

This new table everydayobjects_embeddings contains the original objects along with their vector embeddings (a new data type in Snowflake).

Vector Similarity Search

Now, the magic happens. We can find semantically similar objects using VECTOR_COSINE_SIMILARITY:

```
SELECT
 object_name,
 VECTOR_COSINE_SIMILARITY(
   object_embedding,
   SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'tree')
 ) AS similarity
FROM everydayobjects_embeddings
ORDER BY similarity DESC
LIMIT 5;
```

This will return the top 5 objects most closely related to the word ‘tree’ semantically.

This query returns objects most similar to “tree,” considering their meaning rather than just their names.

Example – 2 :Putting It Together: Document Chunking and Summarization

This concept extends to larger text documents. Let’s create a DocumentChunks table and embed the chunks:

```
-- Table Creation
CREATE OR REPLACE TABLE DocumentChunks (
   chunk_id INT PRIMARY KEY,
   document_name VARCHAR(255),
   page_number INT,
   chunk_text TEXT
);


-- Sample Data Insertion
INSERT INTO DocumentChunks (chunk_id, document_name, page_number, chunk_text)
VALUES
   (1, 'Sales_Report_Q1_2024.pdf', 1, 'This report summarizes the sales performance for the first quarter of 2024. The company achieved a 15% increase in revenue compared to the same period last year.'),
   (2, 'Sales_Report_Q1_2024.pdf', 2, 'Key factors contributing to the growth include the launch of new products and expansion into new markets.'),
   (3, 'Product_Brochure.pdf', 1, 'Our innovative product X is designed to revolutionize the way you work. It offers a seamless user experience and powerful features to enhance productivity.'),
   (4, 'Technical_Manual.pdf', 5, 'Troubleshooting Guide: If you encounter an error message E101, please check the connection cables and restart the device.'),
  (5, 'Marketing_Strategy_2024.pdf', 3, 'The marketing team will focus on social media
campaigns and influencer partnerships to increase brand awareness.');

Select * from DocumentChunks;


create or replace table documentembeddings as select *,
SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', CHUNK_TEXT) AS
object_embedding from documentchunks;

select * from documentembeddings;

SELECT
 CHUNK_TEXT,
 VECTOR_COSINE_SIMILARITY(
  object_embedding,
  SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'sales growth in Q1')
 ) AS similarity
FROM DocumentEmbeddings
ORDER BY similarity DESC
LIMIT 1; --Change to from Limit 5 to limit 1
```

We can then find the chunk most similar to a query and summarize it:

```
SELECT
 chunk_text AS Original_Chunk,
 SNOWFLAKE.CORTEX.SUMMARIZE(chunk_text) AS Cortex_Summary
FROM (
 SELECT
  CHUNK_TEXT,
  VECTOR_COSINE_SIMILARITY(
   object_embedding,
   SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'sales growth in Q1')
  ) AS similarity
 FROM DocumentEmbeddings
 ORDER BY similarity DESC
 LIMIT 1
) AS MostSimilarChunk;
```

This query will search the document chunks for text related to “sales growth in Q1”, then return the original chunk along with a summary generated by Snowflake Cortex.

Key Takeaways:

- Embeddings unlock the ability to search and analyze text based on meaning.
- Snowflake Cortex makes it easy to generate and work with embeddings.

- Vector similarity search enables finding semantically related information.
- Applications range from simple object comparisons to complex document analysis and RAG based applications.

RESET Environment

To reset the environment for further experimentation or to clean up resources, you can drop the EmbeddingsDB database using the provided script.

```
-- Drop the database
DROP DATABASE IF EXISTS EmbeddingsDB CASCADE;
```

This will permanently delete all tables, data, and other objects created within this database. Remember, this action is irreversible, so use it with caution!

Reference :

Snowflake Embeddings and Vector Search | by Fru | DemoHub Tutorials | Medium

Snowflake Documentation & Resources:

- Snowflake Cortex Overview: Large Language Model (LLM) Functions (Snowflake Cortex)
- Snowflake Cortex Embeddings: Vector Embeddings | Snowflake Documentation
- EMBED_TEXT_768 Function: EMBED_TEXT_768 (SNOWFLAKE.CORTEX)
- VECTOR_COSINE_SIMILARITY Function: https://docs.snowflake.com/en/sql-reference/functions/vector_cosine_similarity