Graphs, GraphRAG and Generative AI: An Introduction

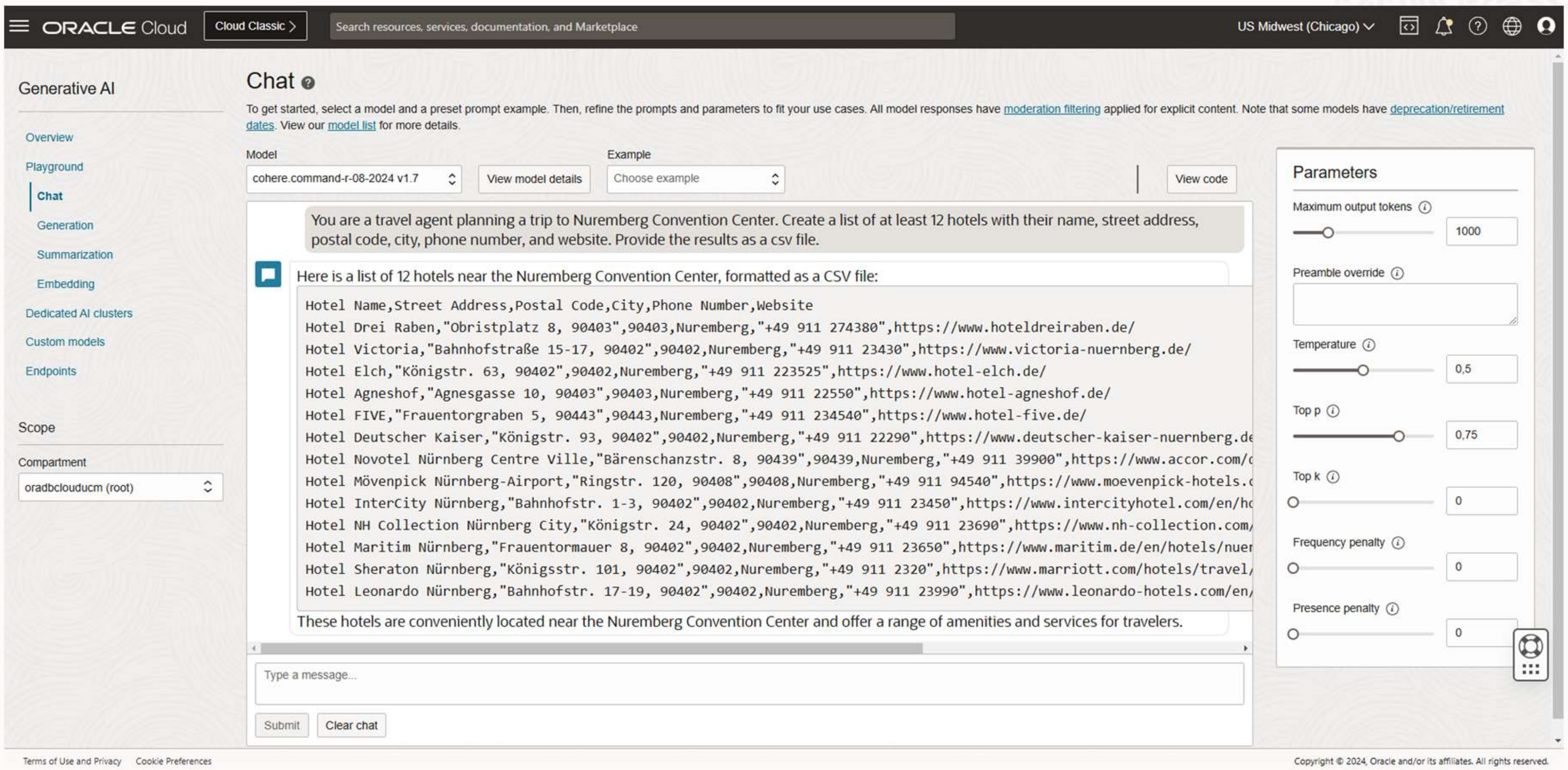
UKOUG Conference Discover 2024

Hans Viehmann
Director Product Management
Oracle Spatial and Graph
December 2nd, 2024

"You are a travel agent planning a trip to Nuremberg Convention Center. Create a list of at least 15 hotels with their name, street address, postal code, city, phone number, and website. Provide the results as a csv file."

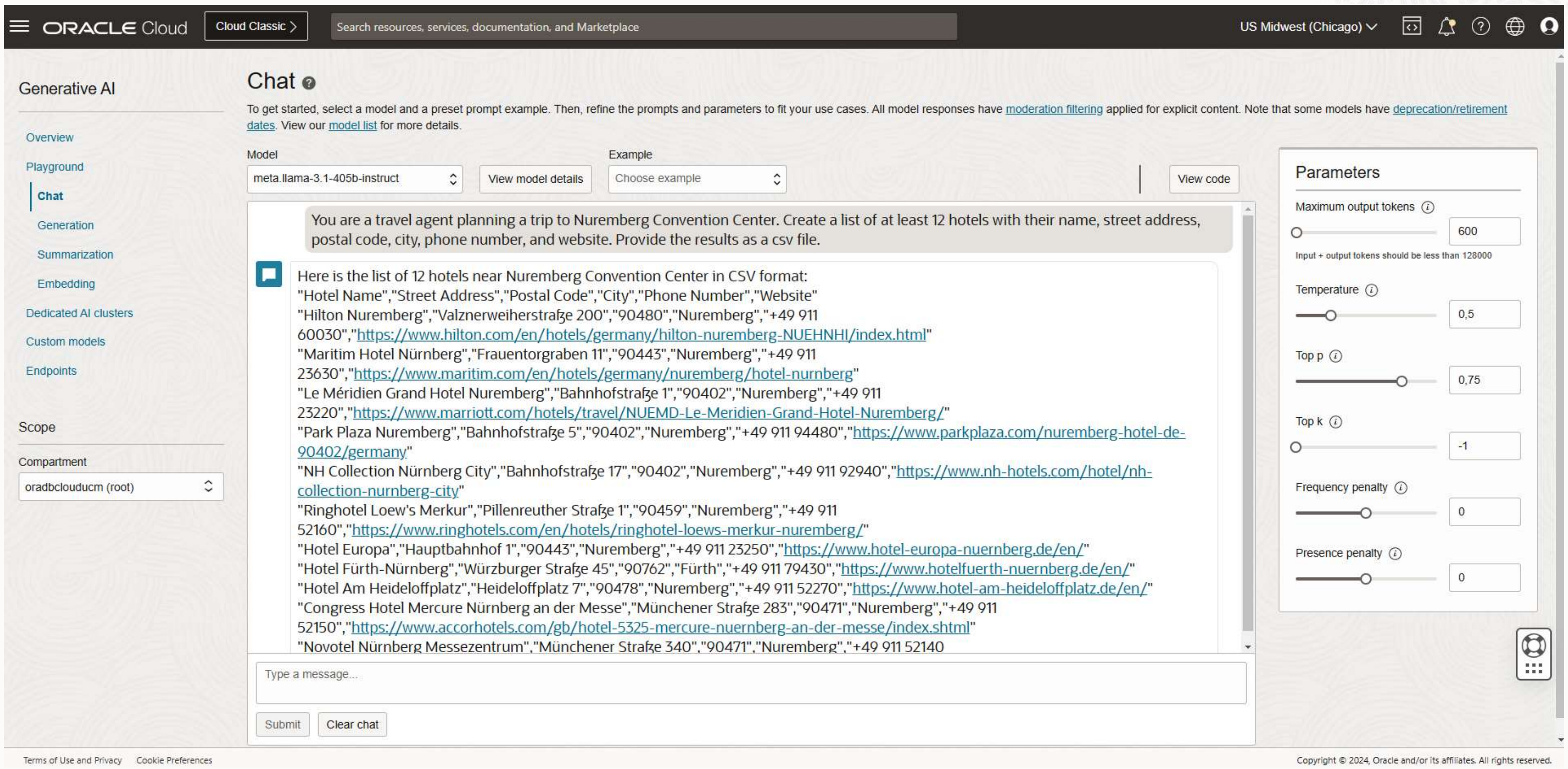
Using Chat interface in OCI GenAI

Model: cohere.command-r-08-2024 v1.7



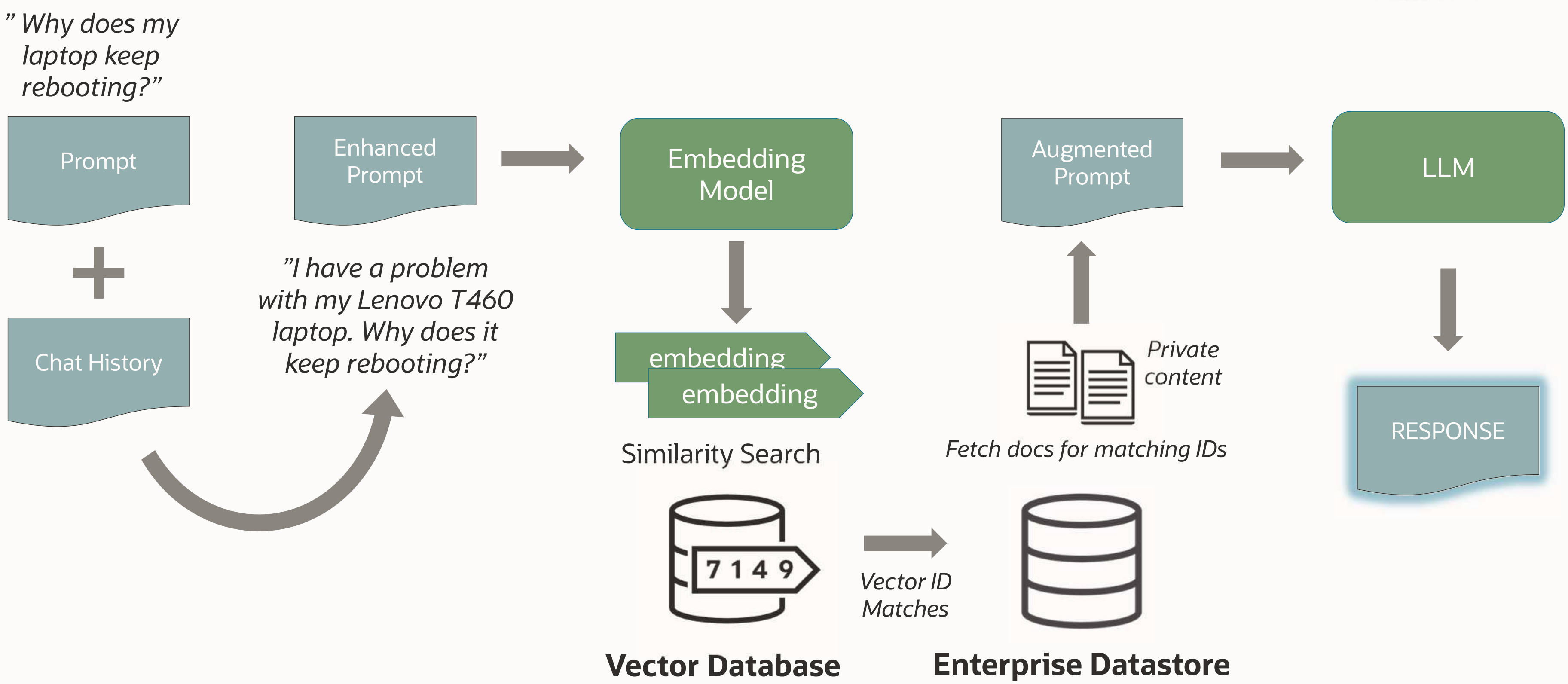
Using Chat interface in OCI GenAI

Model: meta.llama-3.1-405b-instruct



Retrieval Augmented Generation workflow

Supplying LLM-based chatbot with “enterprise knowledge”



RAG based on vector search

Development

Frameworks and tools evolving rapidly

- Langchain, LlamaIndex, …

Comprehensive support in Oracle 23ai

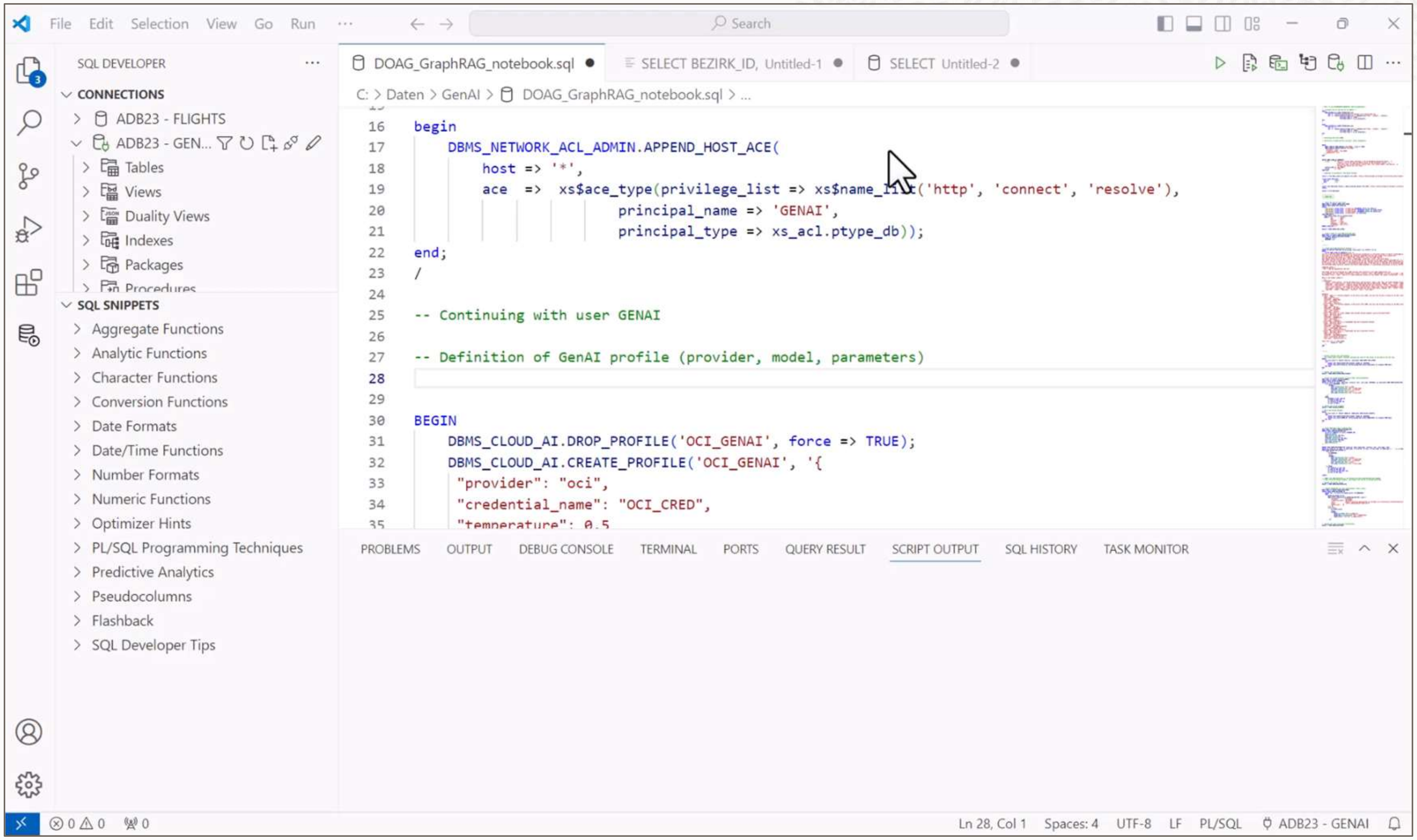
- Document storage
- Vector similarity search
- OML with ONNX support
- APIs for external GenAI services
- … All available through SQL and PL/SQL

Accessing GenAI APIs from Oracle 23ai

- Allow REST callouts
- Set up credentials
- Define GenAI profile

```
SELECT DBMS_CLOUD_AI.GENERATE(
    prompt       => 'You are a travel agent
<…> .',
    profile_name => 'OCI_GENAI',
    action       => 'chat')
FROM dual;
```

Accessing GenAI APIs from Oracle 23ai



Demo

Graph RAG

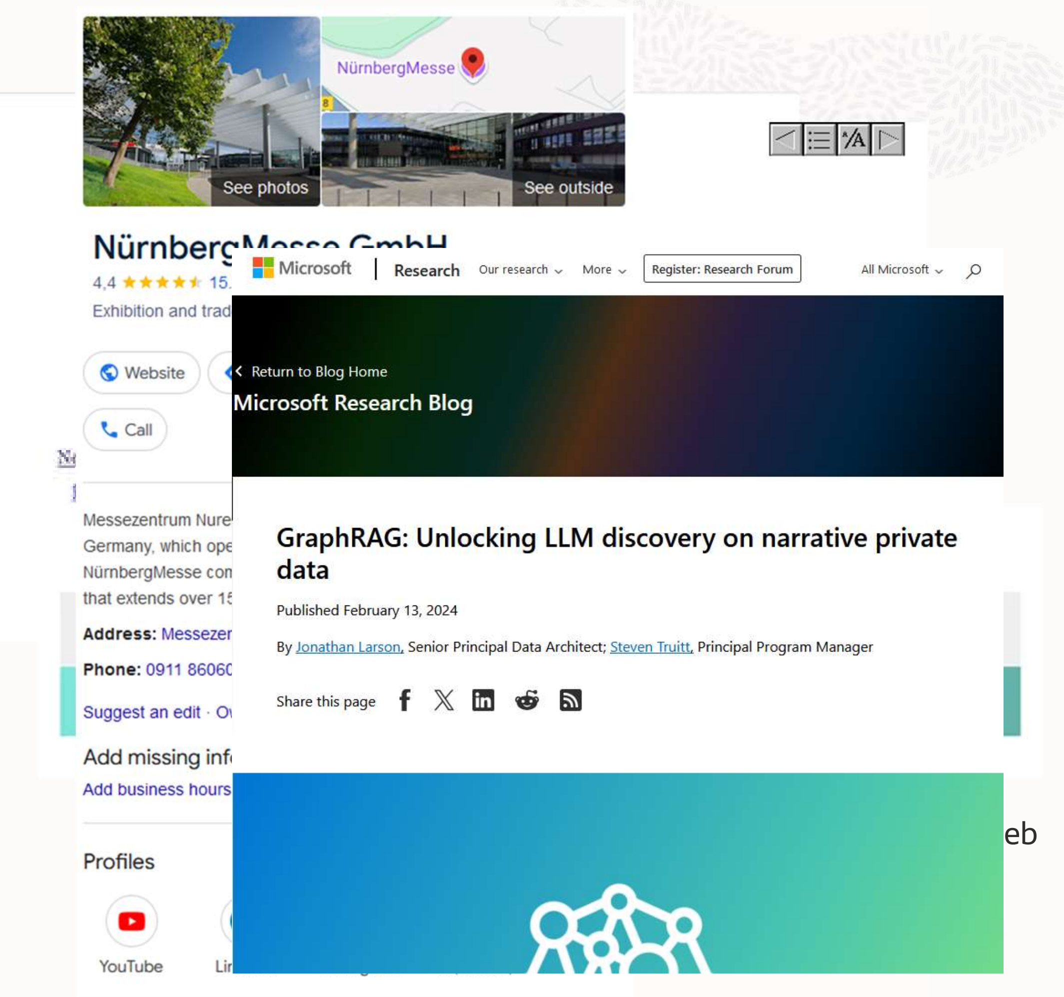
Improving baseline RAG with graphs

Evolution of search

- Keyword search (~1994)
- Pagerank (~1999)
- Semantic web (~1999)
- Google Knowledge Graph (2012)
- Graphs with LLMs (GraphRAG, 2024)

GraphRAG

- Includes graph retrieval step as part of RAG workflow
- Not necessarily graph retrieval only



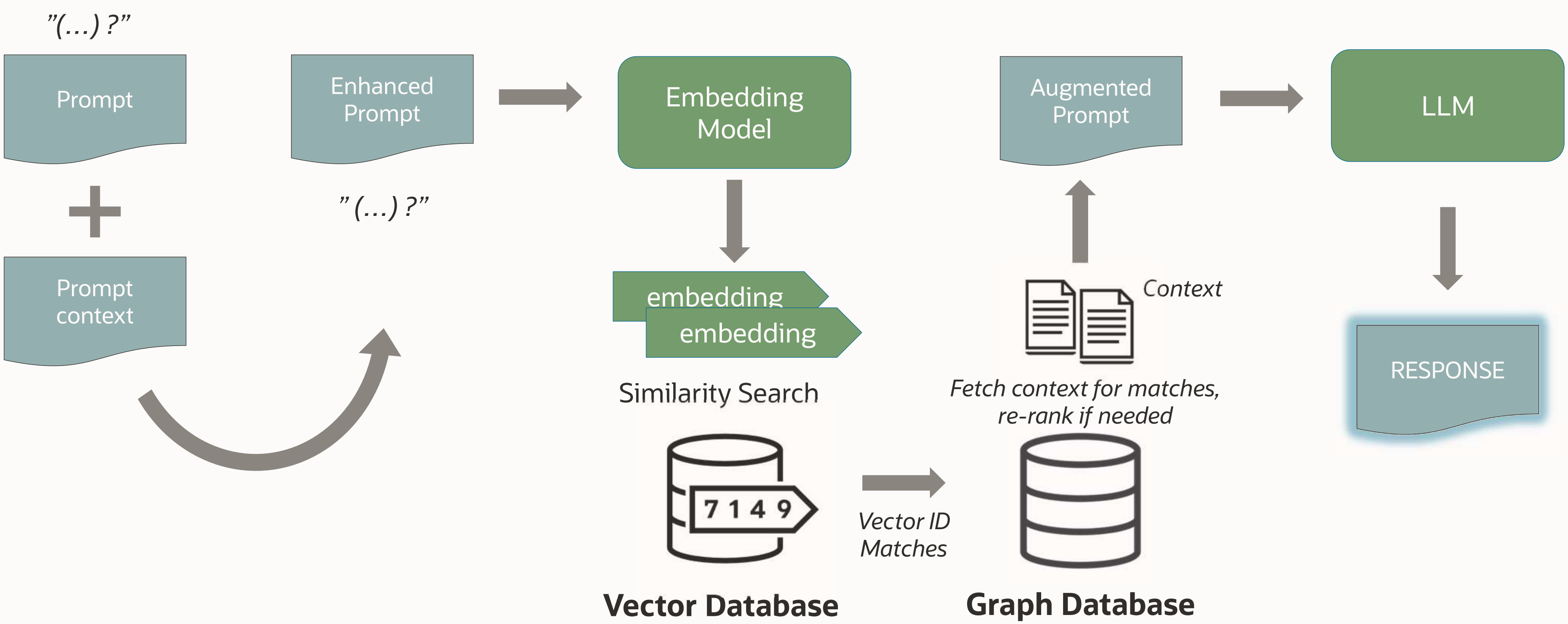
What are Knowledge Graphs?

Knowledge graphs use a graph structure to represent information about entities and their relations

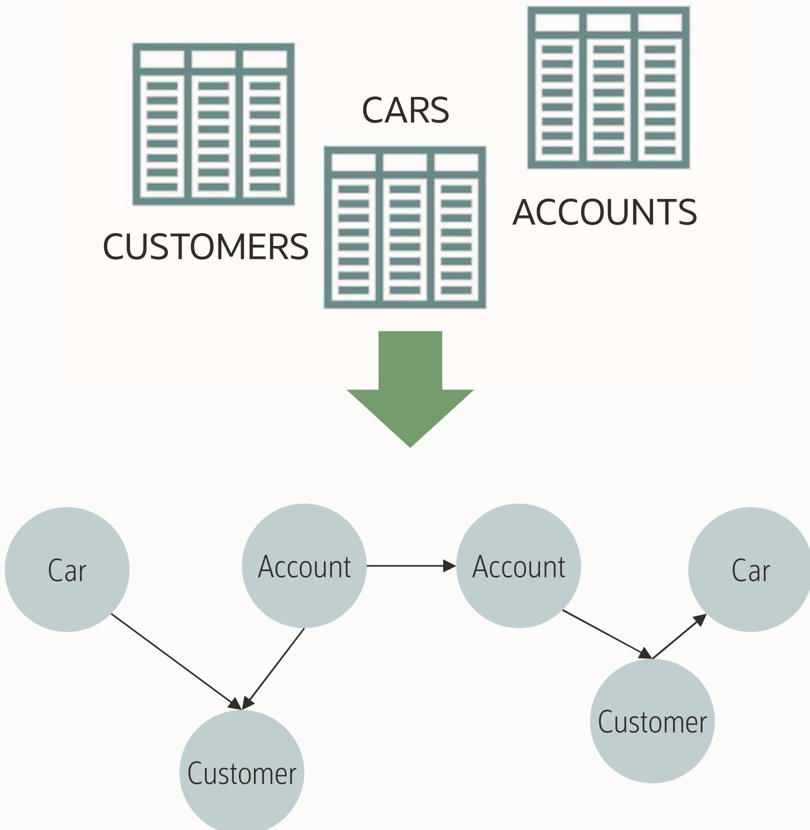
- Schema-flexible model

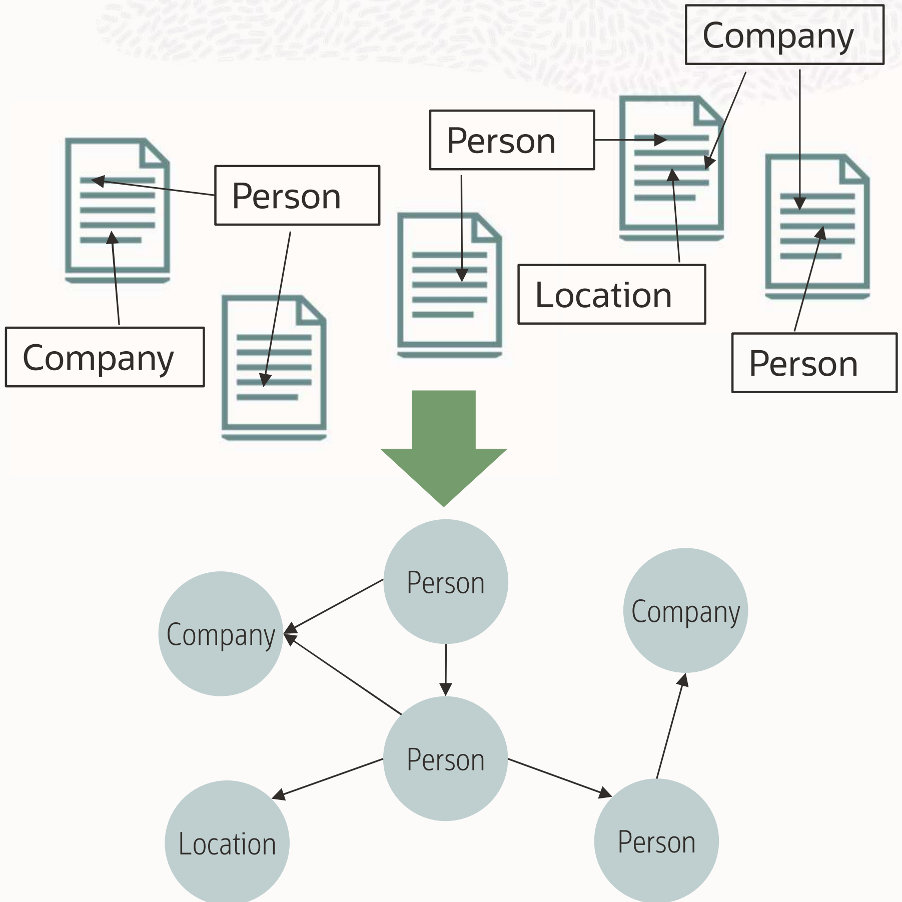
- Typical use cases include improving search engines, smart assistants, detecting fraud and analyzing financial crime, predicting diagnoses in healthcare.
- Expressive data model, powerful tool for data integration, unification, and analysis.

GraphRAG Workflow: LLM-based chatbot using Knowledge Graph



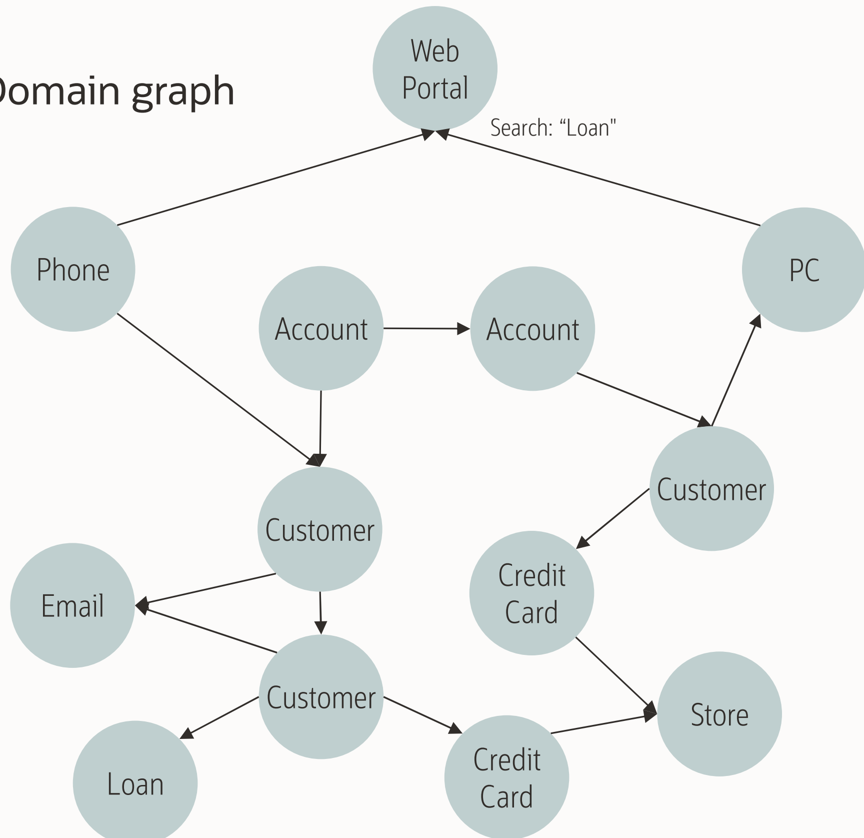
Where do graphs come from?



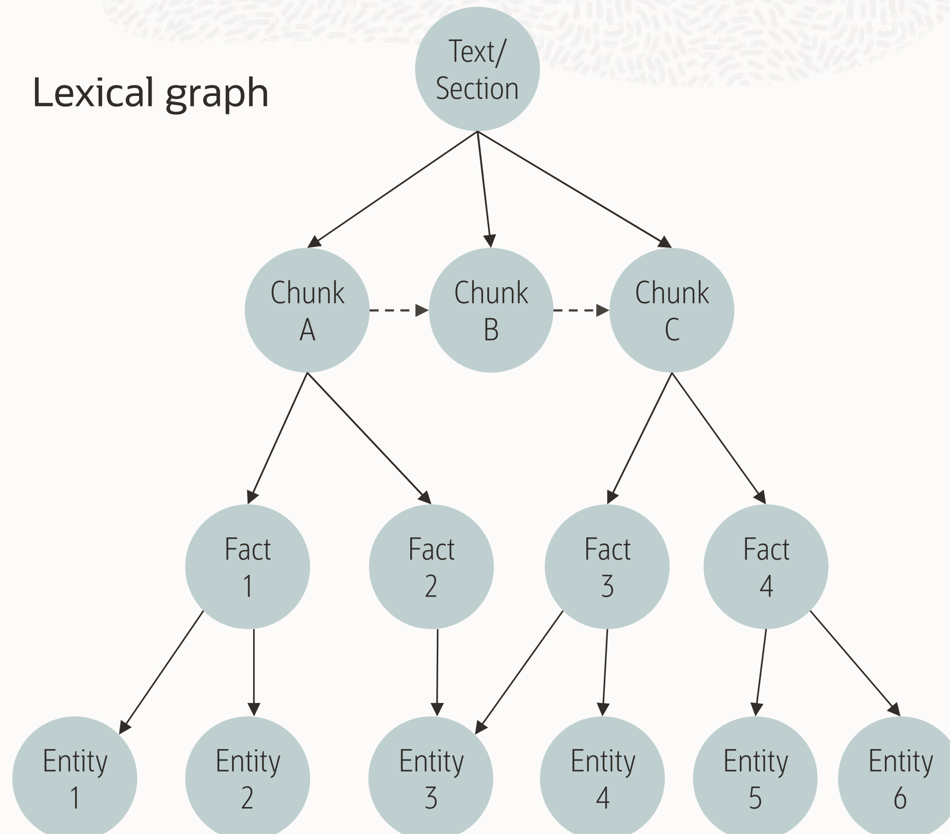


Exatracting graphs from documents

Domain graph

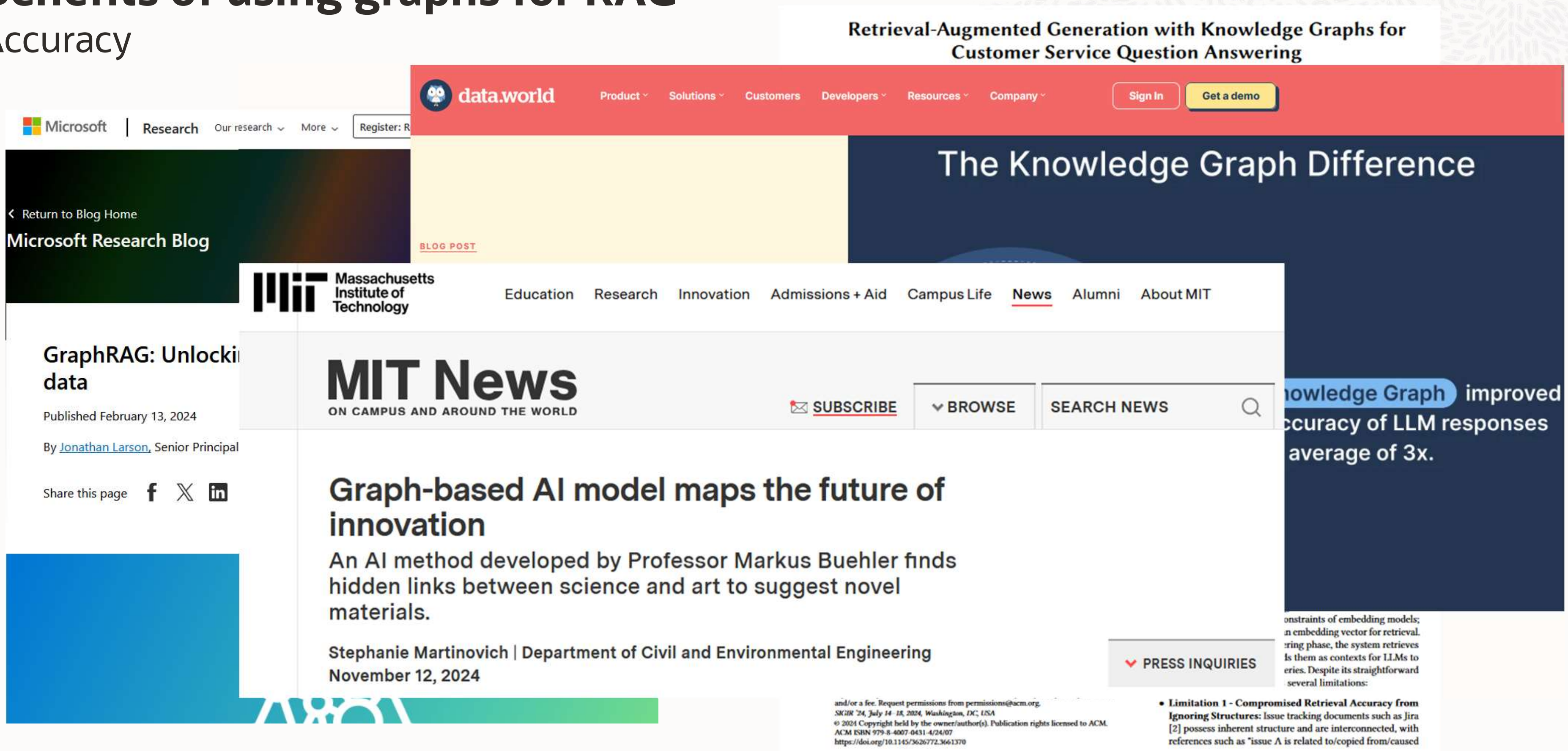


Lexical graph



Benefits of using graphs for RAG

Accuracy

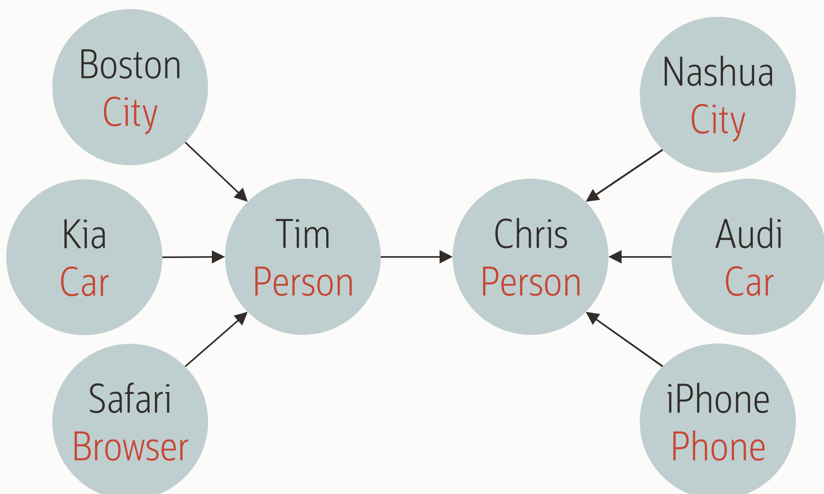


Benefits of using graphs for RAG

Explainability

Graph query

- Explicit representation of relationships
- Takes semantics into account
- Search based on graph traversal



Vector similarity search

- Opaque data structure
- Different distance metrics
- Search based on approximate nearest neighbor analysis

Tim: `[0.247, 0.889, 0.144, 0.183,…]`

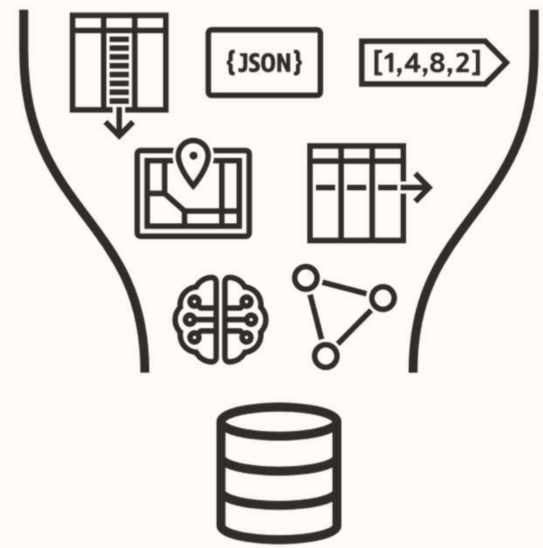
Chris: `[0.551, 0.795, 0.201, 0.229,…]`

Why use Oracle for GraphRAG?

Converged database
- Graph support natively in the database kernel
  - Graph visualization for validation, debugging
- Vector similarity search built in
- Document store (Text, JSON, XML, …)

Integration of external services
- Secure REST calls from database
- PL/SQL APIs

OCI GenAI Services
- Various LLM services and options
- Comfortable tooling

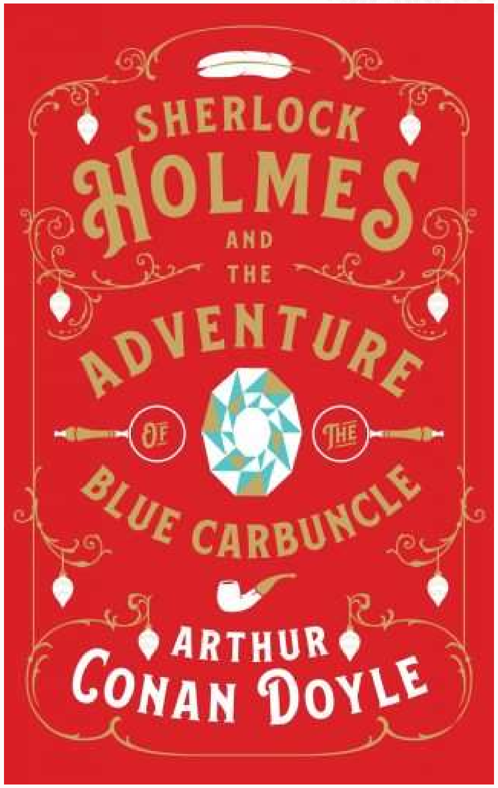


Demo scenario

Analysis of Sherlock Holmes story

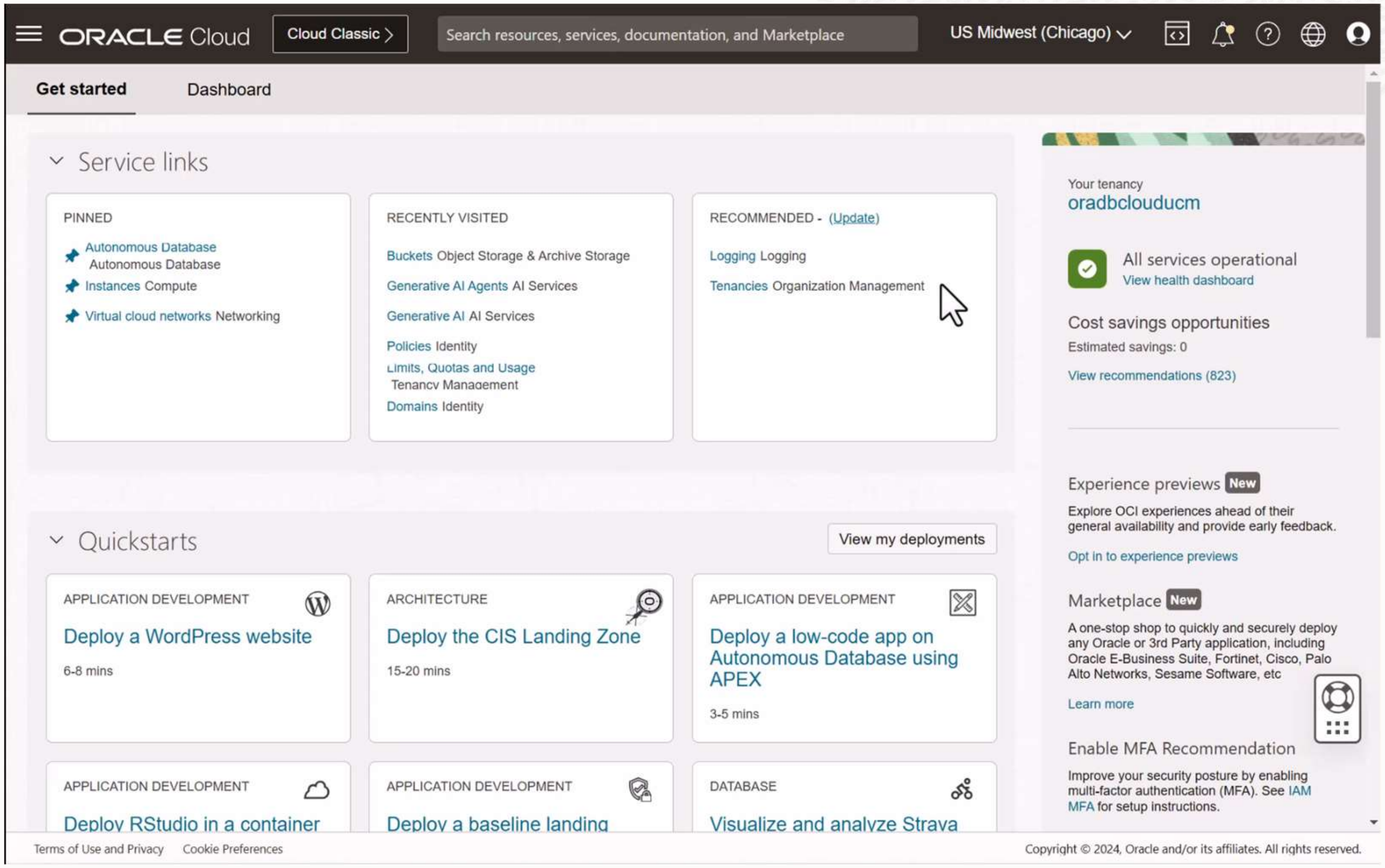
Demo flow:

- Load text from object storage
- Split into chunks
- Extract relationships from each chunk in JSON format using LLM with self-defined prompt
- Extract entities, relationships and respective types from each JSON string
- Compute embedding for JSON string using embedding model, create vector index
- Create property graph from entities and relationships for graph queries
- Compute query embedding and select most relevant chunks
- Use LLM to generate answer

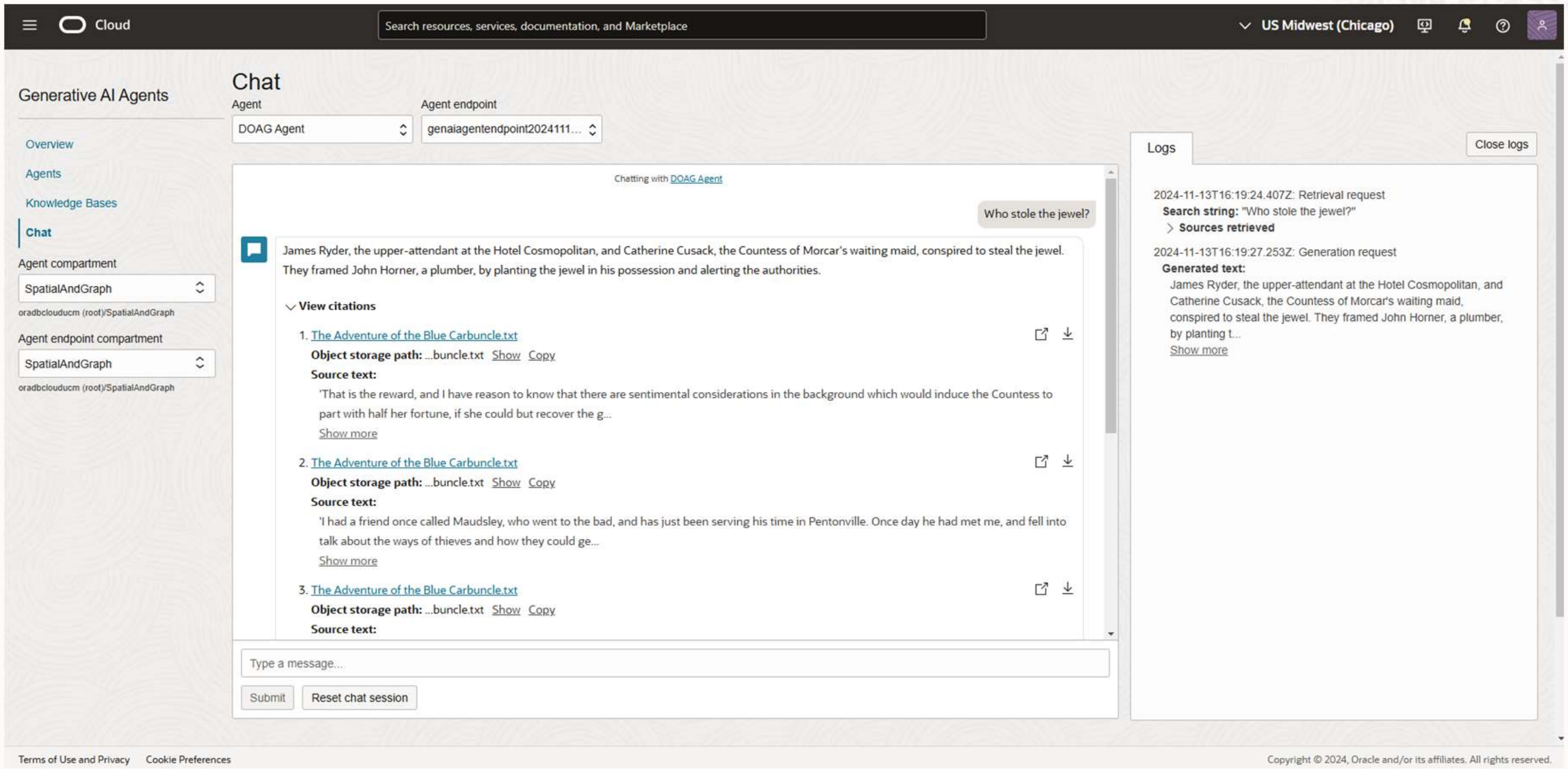


Demo

Using the new Generative AI Agents in OCI GenAI



Using the new Generative AI Agents in OCI GenAI



Wrap-up

Summary

GraphRAG produces more accurate, explainable results than baseline RAG

Using Oracle 23ai and Oracle Graph simplifies development of GraphRAG workflows

GraphRAG as a technique offers huge potential and is evolving rapidly

Further information on Oracle Graph

Oracle Graph technologies landing page on oracle.com

- www.oracle.com/database/graph

Videos on Youtube

- www.youtube.com/c/OracleSpatialandGraph

Oracle LiveLabs: Search for „graph“ on

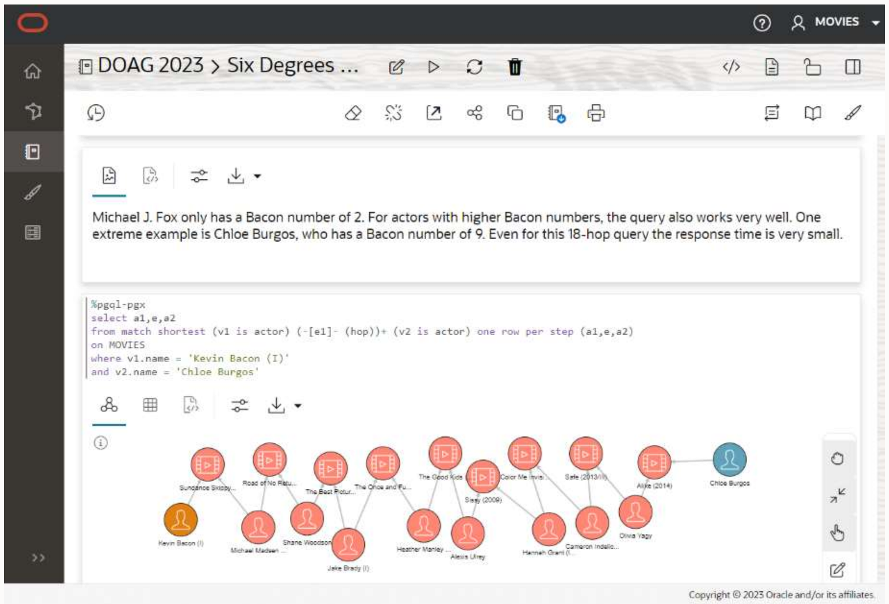
- livelabs.oracle.com/

Blogs

- Database Insider (category: Graph)
- medium.com (tag: Oracle Graph)

Documentation

- Oracle Property Graph documentation



Appendix

GraphRAG use case in Financial Services

Use Case

Case Graph Construction from Suspicious Activity Report (SAR)

Introduction:

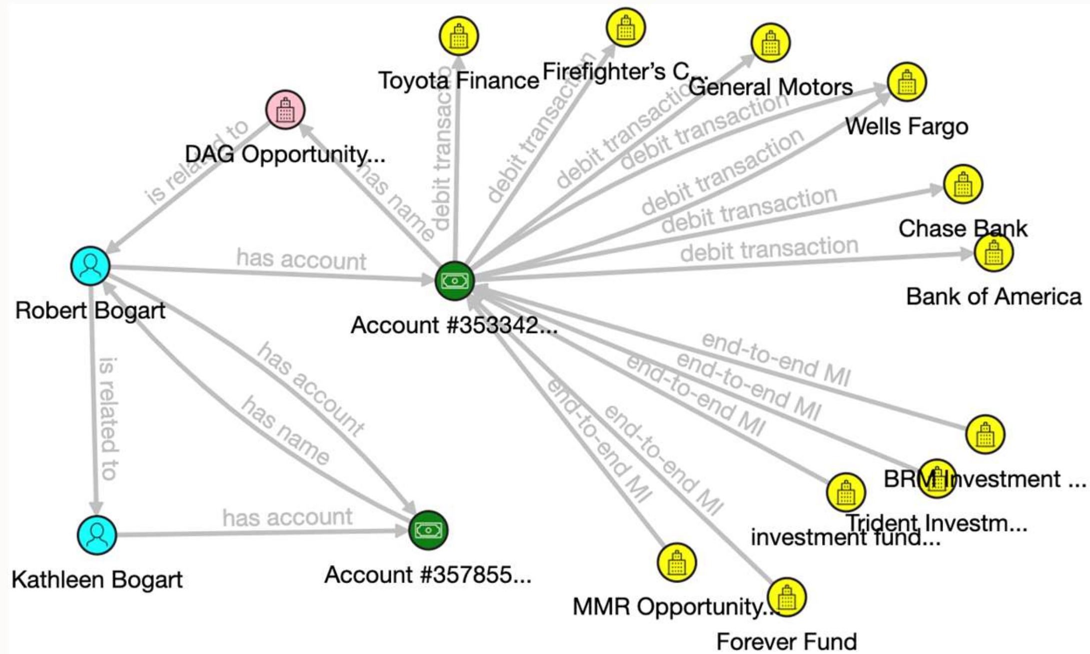
This case was referred for investigation by the MyBank AML Detection unit. The referral identifies potential cash deposit structuring activity in account number 353342287 in the name of DAG Opportunity Fund, LLC.

This investigation, which covers the time period of 1/1/20 through7/6/20 (account closure), revealed suspicious activity totaling $1,399,185.00, occurring between 1/8/20 and 6/18/20.

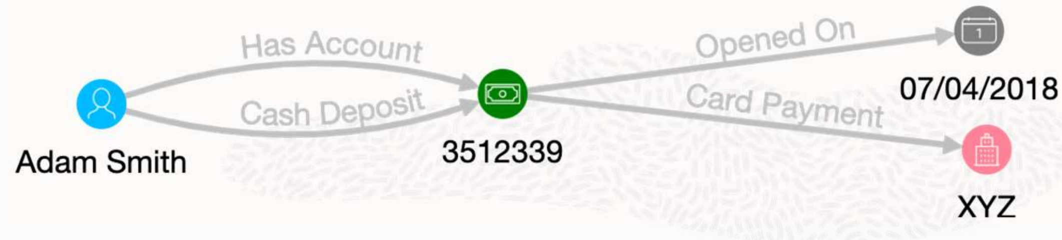
Details of Investigation:

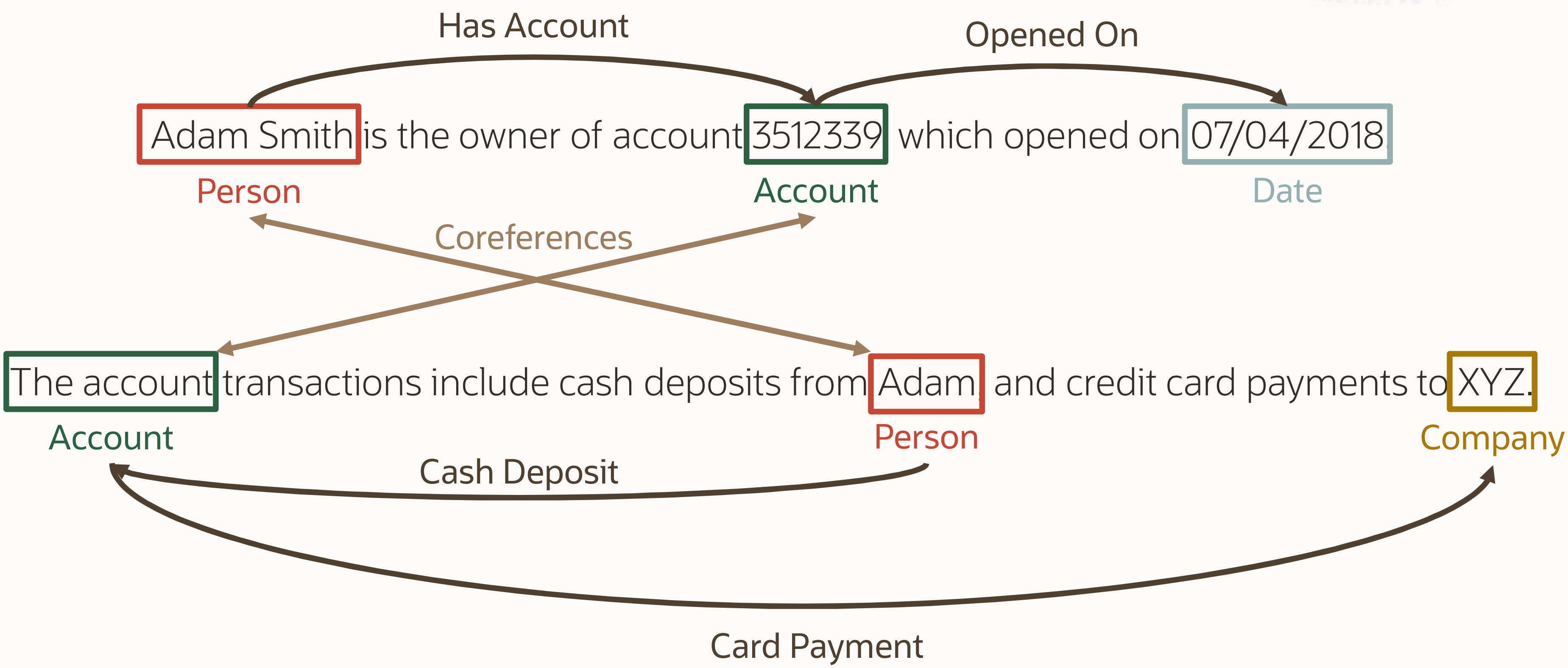
According to internal bank records, DAG Opportunity Fund, LLC is a pooled | investment account. The account signer, Robert Bogart, age 55, along with his spouse, Kathleen, age 48, are listed as an investment broker and schoolteacher respectively, and maintain the following account relationship with MyBank:

Account #353342287 – DAG Opportunity Fund, LLC - opened 4/16/15 – closed 7/6/20 - signer is Robert Bogart. Credits to this MyBank Business Premier Checking account are cash deposits, and checks drawn against the accounts of various individuals and other investment fund accounts. A considerable number of cash deposits, totaling $96,400.00, appear to be structured to avoid the filing of a Currency Transaction Report (CTR) and are therefore deemed suspicious. A sample of the structuring activity is as follows:



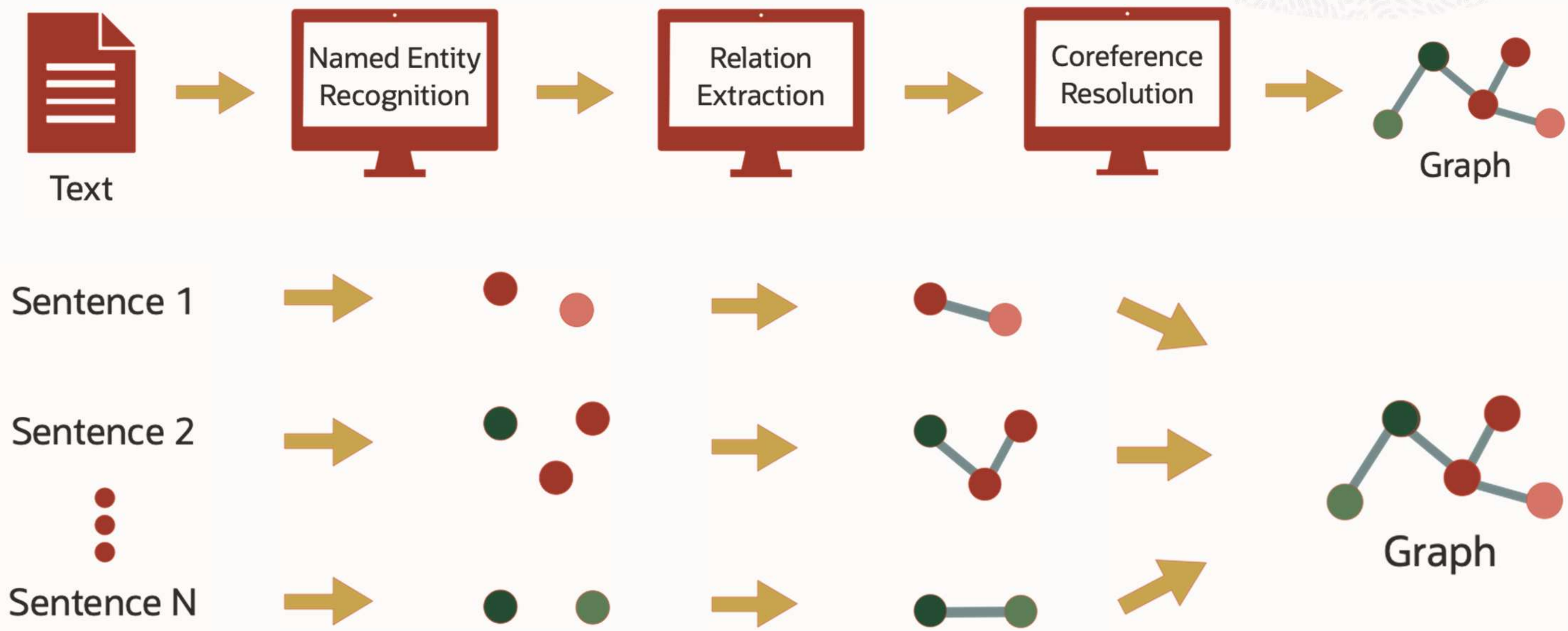
Tasks to Solve
Example





Pipeline Approach

Overview

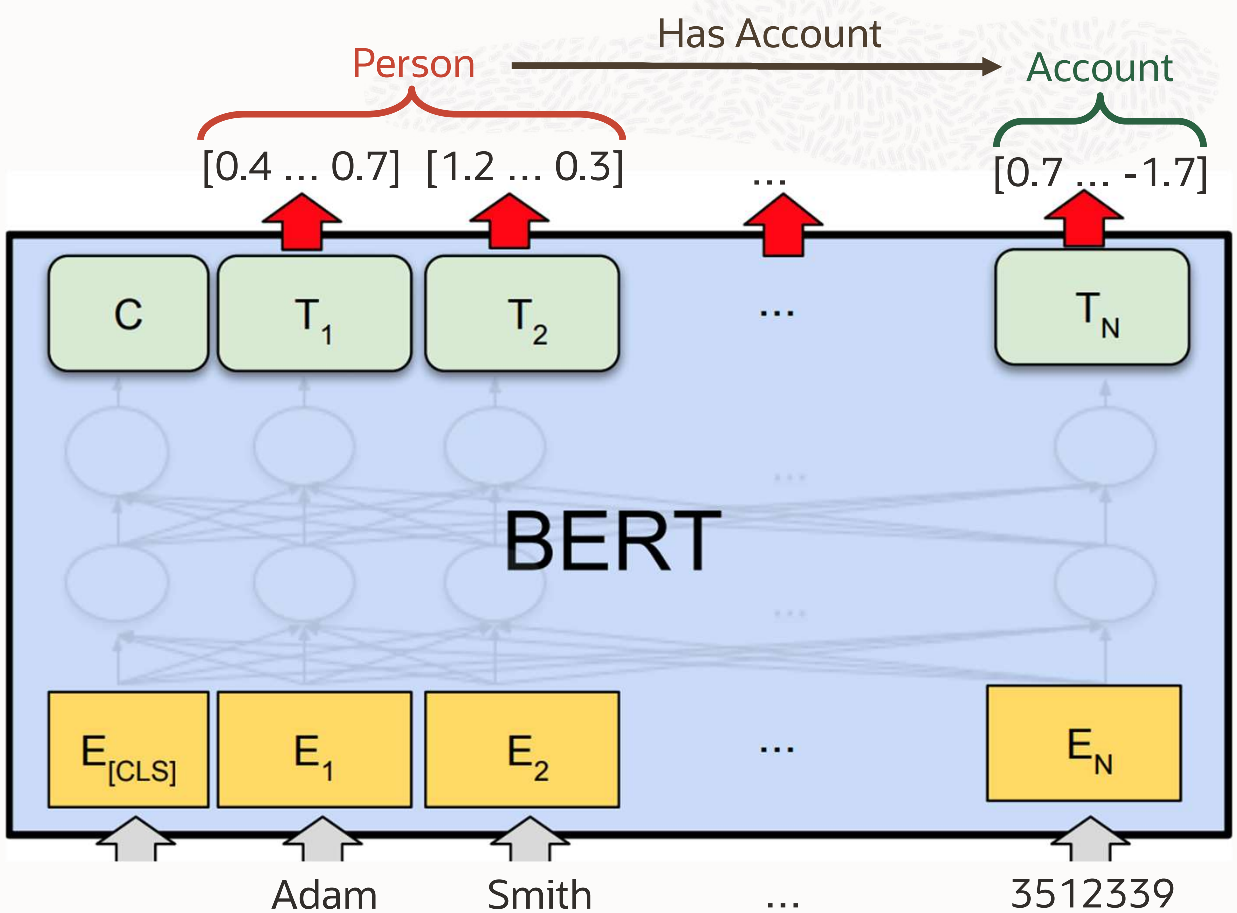


Pipeline Approach
Models

- We built three models, one for each of the tasks in the pipeline.
- All models are based on pre-trained transformer models.

- Transformer models have 100s of millions of parameters, pre-trained on a language modeling task using large datasets.
- They capture information about syntax and semantics of natural language.

- Simple classifiers on generated embeddings are enough to solve all three tasks **after fine-tuning.**



Source: BERT

Pipeline Approach

Data Annotation

Introduction:

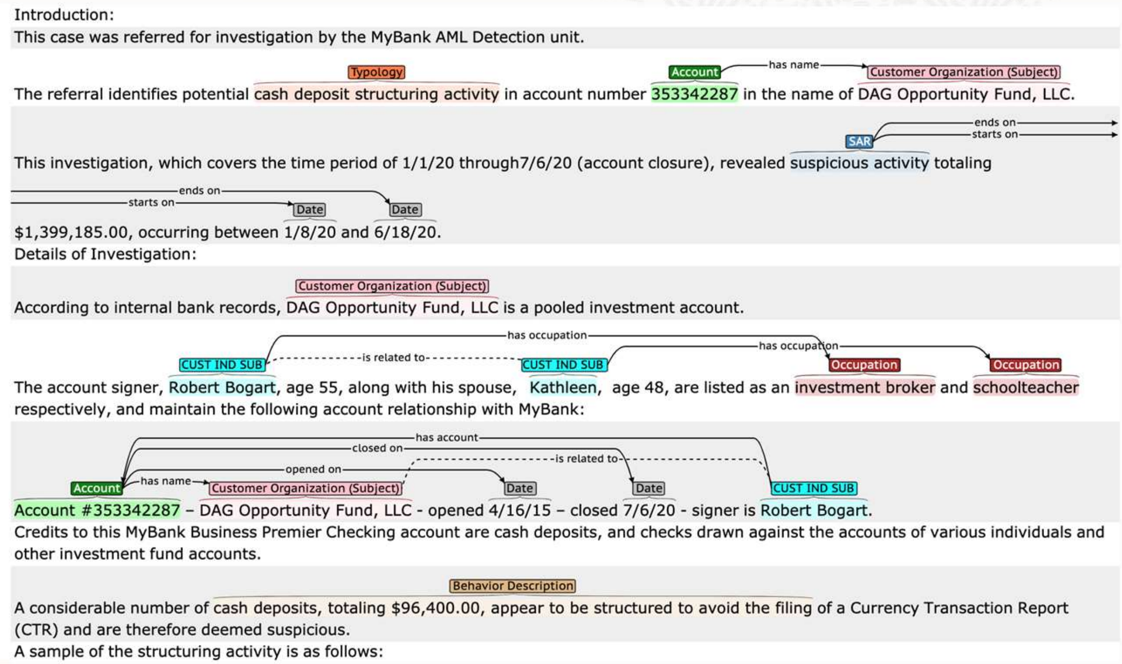
This case was referred for investigation by the MyBank AML Detection unit. The referral identifies potential cash deposit structuring activity in account number 353342287 in the name of DAG Opportunity Fund, LLC.

This investigation, which covers the time period of 1/1/20 through7/6/20 (account closure), revealed suspicious activity totaling $1,399,185.00, occurring between 1/8/20 and 6/18/20.

Details of Investigation:

According to internal bank records, DAG Opportunity Fund, LLC is a pooled | investment account. The account signer, Robert Bogart, age 55, along with his spouse, Kathleen, age 48, are listed as an investment broker and schoolteacher respectively, and maintain the following account relationship with MyBank:

Account #353342287 – DAG Opportunity Fund, LLC - opened 4/16/15 – closed 7/6/20 - signer is Robert Bogart. Credits to this MyBank Business Premier Checking account are cash deposits, and checks drawn against the accounts of various individuals and other investment fund accounts. A considerable number of cash deposits, totaling $96,400.00, appear to be structured to avoid the filing of a Currency Transaction Report (CTR) and are therefore deemed suspicious. A sample of the structuring activity is as follows:



To train the pipeline, entity + relation + coreference annotations are needed to be done manually on a number of documents using an annotation tool, such as: brat

Pipeline Approach

Results

On benchmark dataset SciERC (**Sci**entific **E**ntity **R**elation **C**oreference):

NAMED ENTITY RECOGNITION	RELATION EXTRACTION	COREFERENCE RESOLUTION
70.1	50.5	60.2

F1 Scores

There are now better scores published for individual tasks on this dataset, e.g.:

- **NER:** 71.1 (Jeong and Kim, 2022)
- **RE:** 51.3 (Santosh et al., 2021)

But the state-of-the-art has not progressed too far.

On our financial dataset SAR (more structured documents):

NAMED ENTITY RECOGNITION	RELATION EXTRACTION	COREFERENCE RESOLUTION
88.82	81.26	87.06

F1 Scores

SAR Graph Quality Metrics:

NODES	EDGES
83.12	82.13

F1 Scores

Generalizability & Annotation Requirements

While the pipeline approach works well, it requires **fine-tuning three models on many manually-annotated documents.** SAR dataset has 105 annotated documents and SciERC has 500.

Fine-tuned models end up being specific to the domain and schema they were trained with. This means for each new use case we need to retrain on a set of annotated documents for that use case.

Most stakeholders don't have such a dataset and would prefer not to go through the annotation process. So we have been focusing our research on reducing these annotation requirements.

Appendix

Other Ways of Using Graphs with GenAI

Generate Graph Queries

- Use LLMs to translate human language queries into graph query language
- If the LLM is not trained to formulate such queries, they can have the capacity to learn with a few examples (‘few-shot’ learning)
- Eliminates the need for developers to learn new syntax

Steps

- Embed as many queries as possible in a vector database
- From a user query identify the top k matches and retrieve the corresponding graph queries
- Use these related pairs as prompt examples to instruct our LLM to generate SPARQL queries

Semantic Knowledge Graphs based on RDF in RAG

Generating SPARQL queries using OCI GenAI Service

- Graphs can be complex to navigate, often requiring the use of query languages like SPARQL
- Typically, LLMs are not directly trained to formulate these queries. However, their capacity for learning from a few examples, or 'few-shot learning,' especially through prompt engineering, is a game-changer.
- By adopting prompt-based few-shot learning, we can teach LLMs to write SPARQL queries with a high level of proficiency.

Strategy:

1. **Indexing**: Cataloging all the queries in our dataset into a vector store to streamline the retrieval process.
2. **Match User Queries:** When a user submits a query, we identify the most closely related queries in our index.
3. **Retrieval:** We then extract the corresponding SPARQL queries for these matches.
4. **Educate the LLM:** Finally, we use these related pairs as prompt examples to instruct our LLM in generating accurate SPARQL queries.
5. **Final Answer Generation:** After obtaining the SPARQL query, we run it against the graph to gather the necessary results. These results, combined with the initial user query are then fed to the LLM to craft a well-rounded final response.

Thank you

oracle.com/database/graph/

Our mission is to help people see data in new ways, discover insights, unlock endless possibilities.

ORACLE