INF Informatik
Hochschule Reutlingen

Intelligente Informationssysteme
RAG Part 2 - Knowledge Extraction

Dominik Neumann

Bock 3 – Retrieval Augmented Generation

Design Patterns

Informatik
Hochschule Reutlingen

Conversional AI

Simulate a conversation with the feeling of having a conversation with a human.

- conversational memory
- dialogue generation

- Examples: ChatGPT, Anthropic Claude, Perplexity.AI

Retrieval Augmented Generation

Knowledge Retrieval and understanding is key

- Access to contextual data
- Retrieval and augmentation strategies

CoPilot

Assists a human in his work.
Key differentiator for becoming a CoPilot is understanding of the environment in which the user works.

- access to tools and data,
- reasoning and planning capabilities,
- and specialized profiles
- Examples: GitHub CoPilot

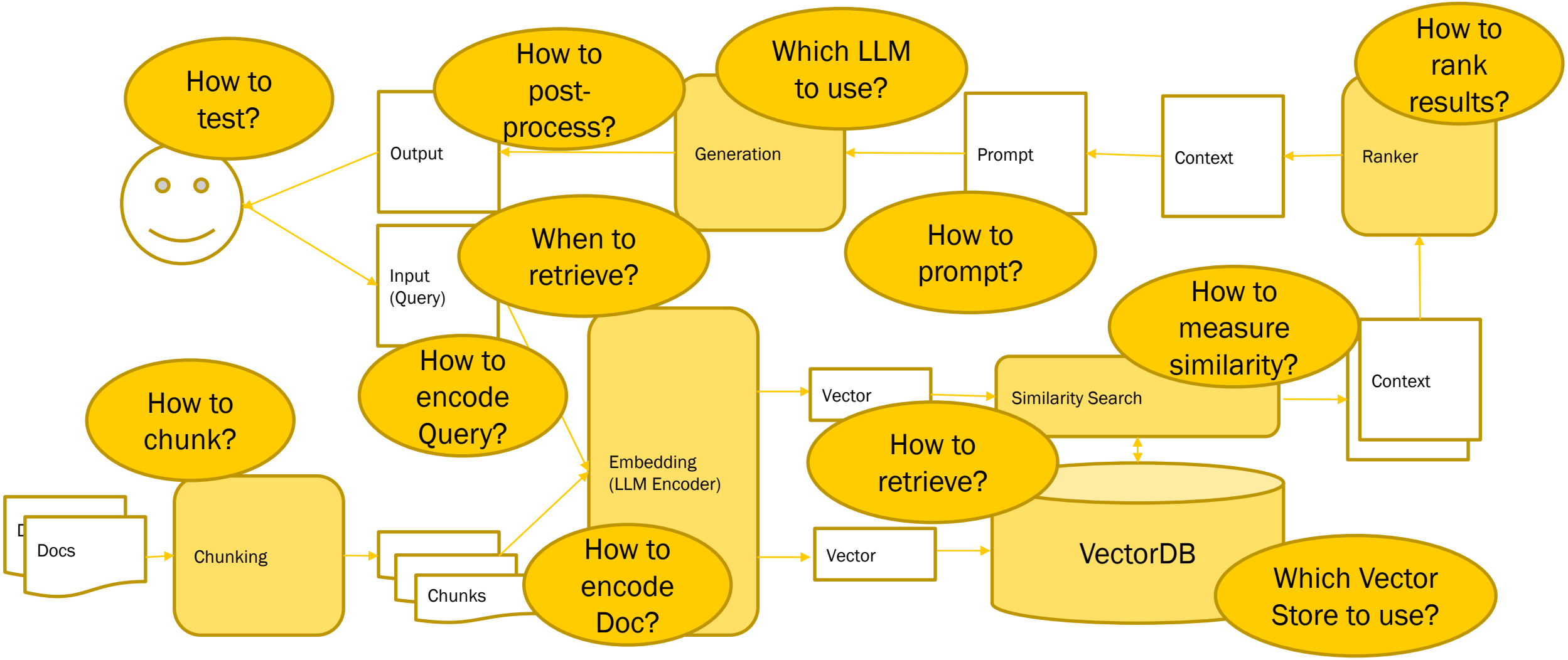
Multi Agent Problem Solver

Multiple agents collaborate to solve a problem. Each agent

- has access to its own set of tools and
- can assume a very specific role while reasoning
- is planning (and executing) its actions.

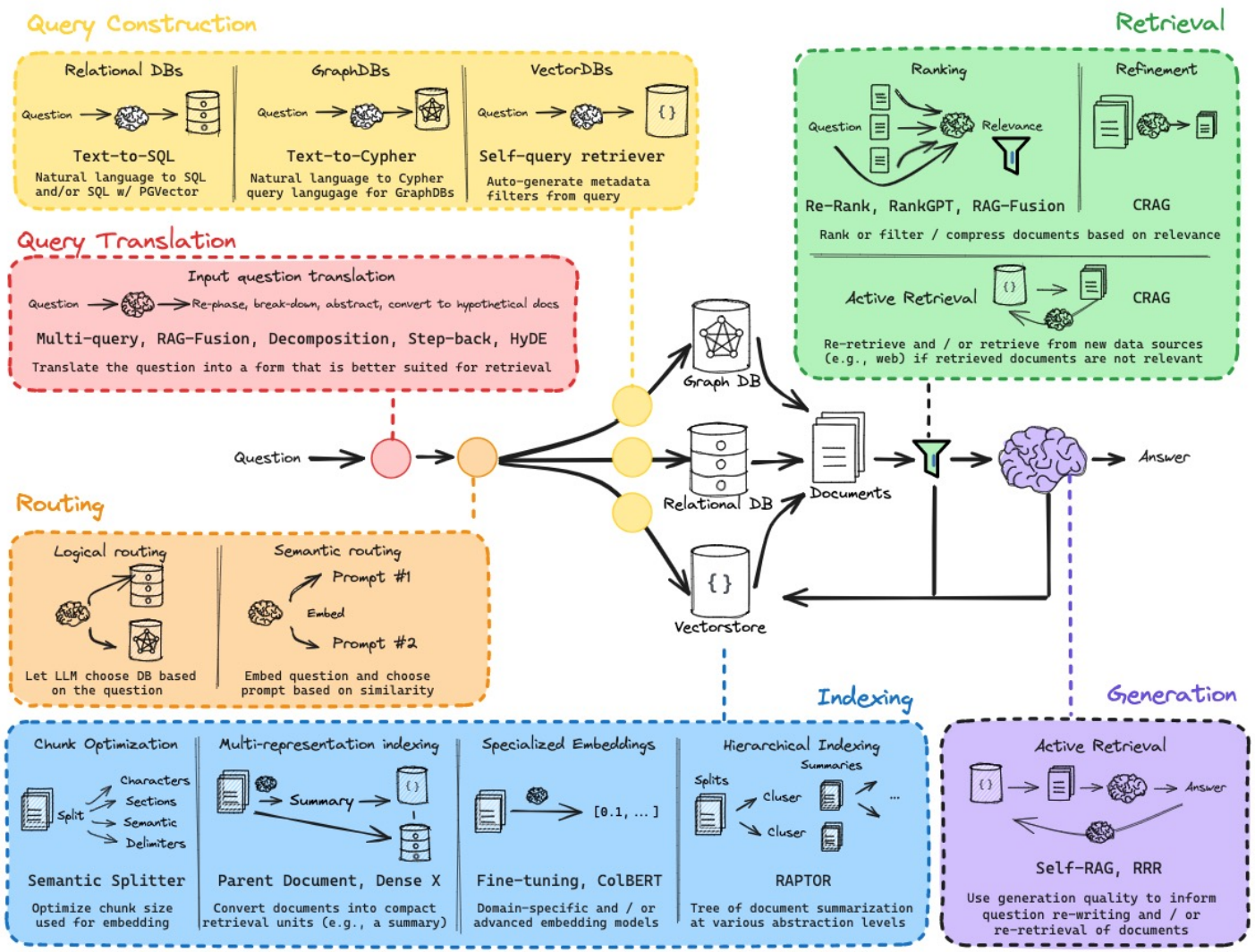
- Example: Devin (https://preview.devin.ai)

Recap: Retrieval Augmented Generation



Recap: Retrieval Augmented Generation

1. Query Transformation
2. Routing
3. Query construction
4. Retrieval
5. Generation



Source: Langchain https://docs.google.com/presentation/d/1C9IaAwHoWcc4RSTqo-pCoN3h0nCgqV2JEYZUJunv_9Q

Projektaufgabe

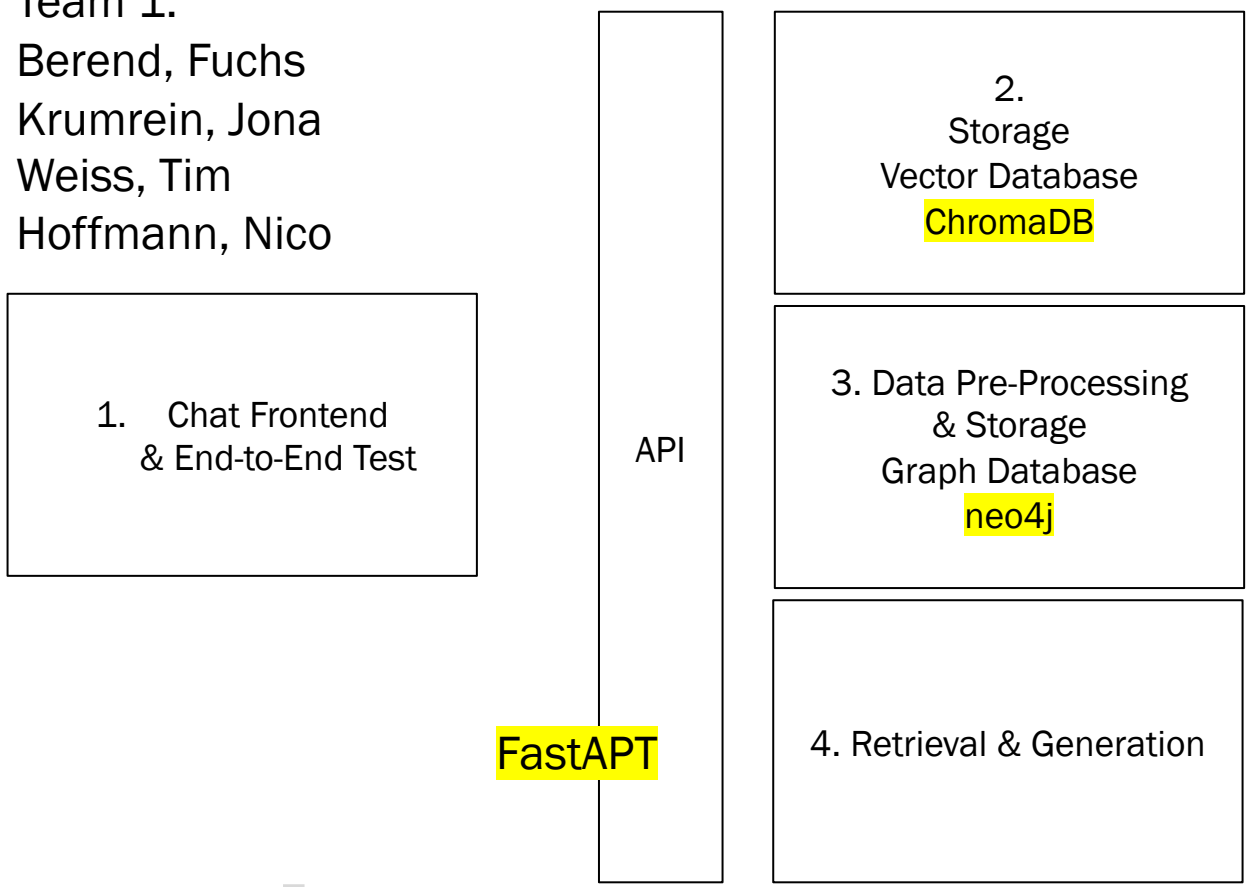
INF
Informatik
Hochschule Reutlingen

Projektaufgabe

Hochschule Reutlingen

- Idee: Wir erstellen als gemeinsames Projekt ein Retrieval Augmented Generation System, das Vorlesungen aus Youtube verarbeitet.

Team 1:
Berend, Fuchs
Krumrein, Jona
Weiss, Tim
Hoffmann, Nico



Team 2:
Buehler, Philipp
Ressler, Malte

Team 3:
Froehner, Thimo
Laib, Lukas
Kraemer, Dominic
Berndt, Nick

Team 4:
Fink, Robin
Doebele, Nico
Zink, Michael

Knowledge Graph

INF
Informatik
Hochschule Reutlingen

Knowledge Graph

Informatik
Hochschule Reutlingen

In Vector Databases we focus on semantic similarity of „text chunks“ - documents with similar concepts will be positioned closer together.

First Idea Similarity Graph: Make a graph with documents as nodes and edges between two documents with a similarity > threshold.

What if the system needs to understand the nature of relationships between different pieces of information?

Knowledge Graph

Hochschule Reutlingen

Example 1:

“Imagine a RAG system used by a market research analyst. The task is to generate a comprehensive report on a competitor, drawing information from news articles, earnings reports, and industry analyses. A vector database approach might retrieve relevant documents, but it would struggle to piece together the timeline of acquisitions, leadership changes, and evolving product lines — the analyst would be left doing most of the time-consuming connective work.”

Example 2:

“Consider a healthcare setting. A RAG system assisting with differential diagnosis needs more than just finding documents containing symptoms and diseases. It needs to understand the hierarchical relationships of medical conditions, causal links between test results and diagnoses, and potential drug interactions. A vector database cannot inherently provide this structural understanding.”

Vector similarity tell you that two entities (text chunks) are conceptually related, but it won’t tell you how they are related.

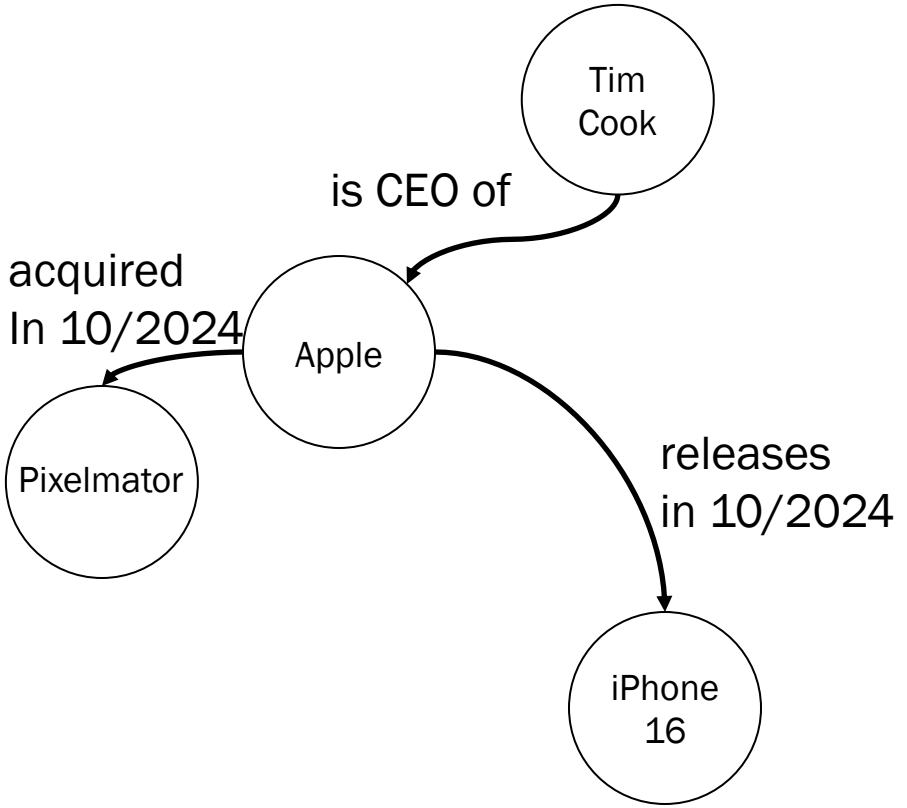
https://medium.com/@learn-simplified/llm-apps-why-knowledge-graphs-are-super-critical-to-know-if-you-care-about-rag-1-9357b69ceb32

Knowledge Graph

A knowledge graphs models knowledge as a network of entities (nodes) and relationships (edges).

→ the way we naturally think about the world.

For example, a knowledge graph for a market research analyst might include entities like companies, products, executives, and market trends, with relationships like “competes with,” “acquired by,” “CEO of,” and “influences.”



Knowledge Graph

Informatik
Hochschule Reutlingen

A knowldge graph is a kind of world model:

→ relations could be causal, temporal or just by similarity

- Mimics Real-World Connections: the way we intuitively understand the world is often through interconnections.
- Semantic Representation: the labels on edges encode meaning. This adds a layer of understanding that computers can process.
- Flexibility, Explainability and Transparency

Knowledge Graph

Hochschule Reutlingen

Based on a knowledge graph, the RAG system can follow the chain of connections to understand the full story!

→ Multi-Hop Reasoning: traverse relationships across multiple “hops”:

Question: “What are the potential environmental implications of Company X acquiring Company Y??”

A knowledge graph-powered RAG system might follow a path like this:

- Company X -> [acquires] -> Company Y
- Company Y -> [manufactures] -> Product Z
- Product Z -> [uses] -> Chemical A
- Chemical A -> [classified as] -> Environmental Pollutant

This chain of reasoning, impossible for a standard vector store, enables the system to surface nontrivial insights essential for a thorough analysis.

Knowledge Graph

Hochschule Reutlingen

Definition:

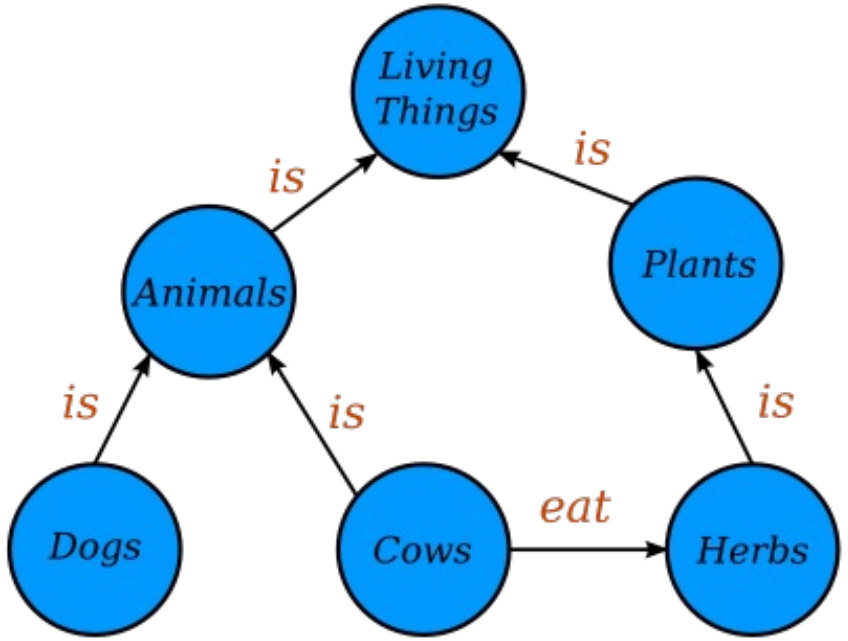
In knowledge representation and reasoning, a knowledge graph is a knowledge base that uses a graph-structured data model or topology to represent and operate on data. Knowledge graphs are often used to store interlinked descriptions of entities – objects, events, situations or abstract concepts – while also encoding the free-form semantics or relationships underlying these entities. (https://en.wikipedia.org/wiki/Knowledge_graph)

- Nodes are the building blocks.

  They represent entities like people, places, things, concepts — anything you want to model

- Edges are connections between nodes.

  These are labeled relationships giving context and meaning to how things are related (e.g., “is a friend of,” “lives in,” “is a type of”).



Knowledge Graph

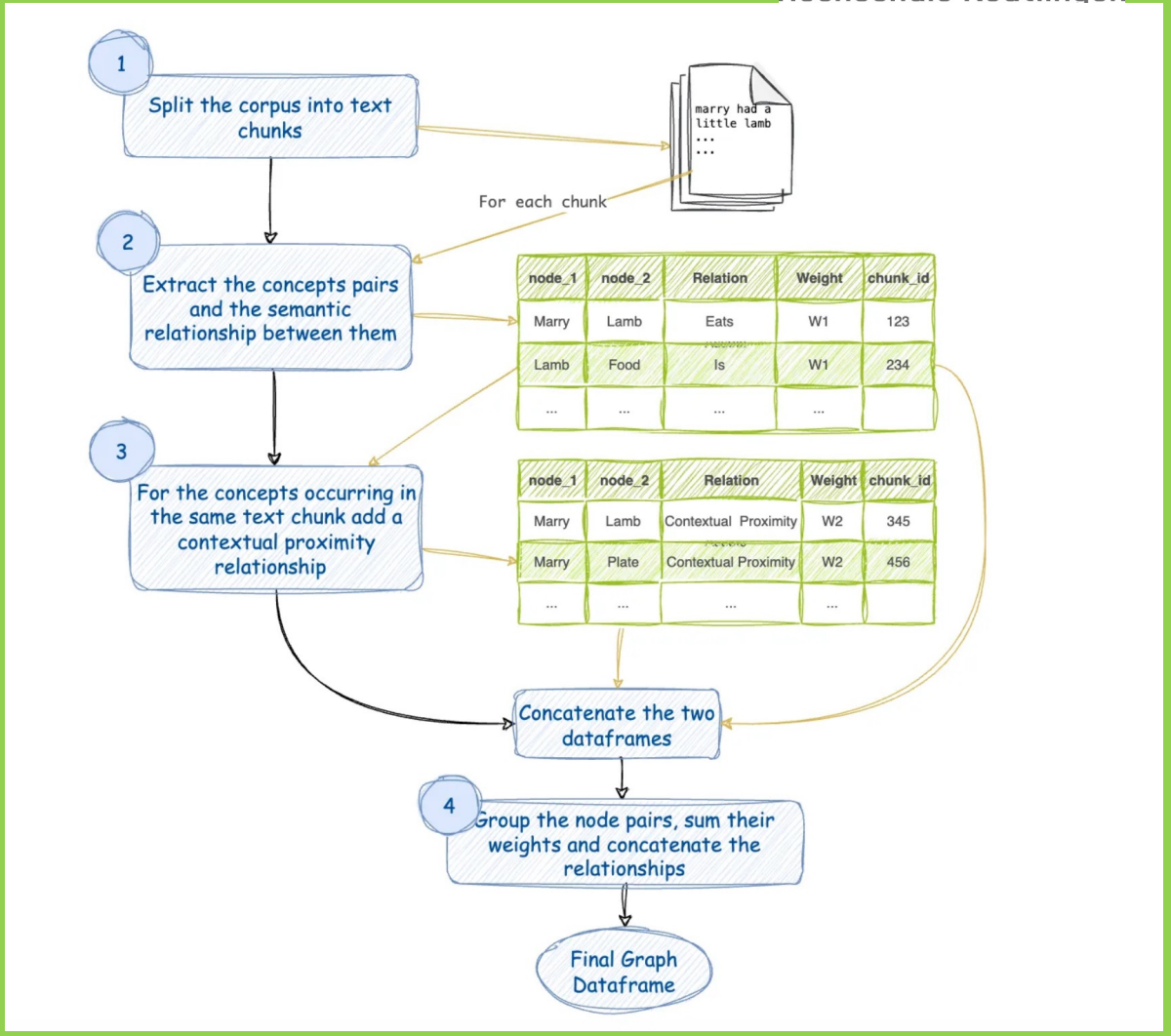
How can we extract entities and relationships out of text?

1. Entities
   - Entity extraction with classical NER
   - convert any text into a Knowledge Graph using LLM
   - NER with GliNER
   - Entity Alignment

2. Relations
   - Co-Mentioning / Contextual Proximity: If two entities are in the same text chunk then there might be a relationship between them
   - Use Prompting to identify relations between entities / nodes

Convert text into Knowledge Graph using LLM

- Extract concepts and entities from text. These are the nodes.
- Extract relations between the concepts. These are the edges.
- Identify contextual proximity between nodes as additional edges
- Populate nodes (concepts) and edges (relations) in a graph data structure



https://towardsdatascience.com/how-to-convert-any-text-into-a-graph-of-concepts-110844f22a1a

GraphRAG

- „From Local to Global: A Graph RAG Approach to Query-Focused Summarization“, 2024, https://arxiv.org/pdf/2404.16130v1
- „Why GraphRAG Is Such a Game Changer for RAG — Part 1“, https://ai.gopubby.com/why-graphrag-is-such-a-game-changer-for-rag-part-1-9edda6cf5487
- „Let’s Do GraphRAG : A Practical Hands-On Guide : Part 2 “, https://ai.gopubby.com/lets-do-graphrag-a-practical-hands-on-guide-part-2-87ae5d88f2b0

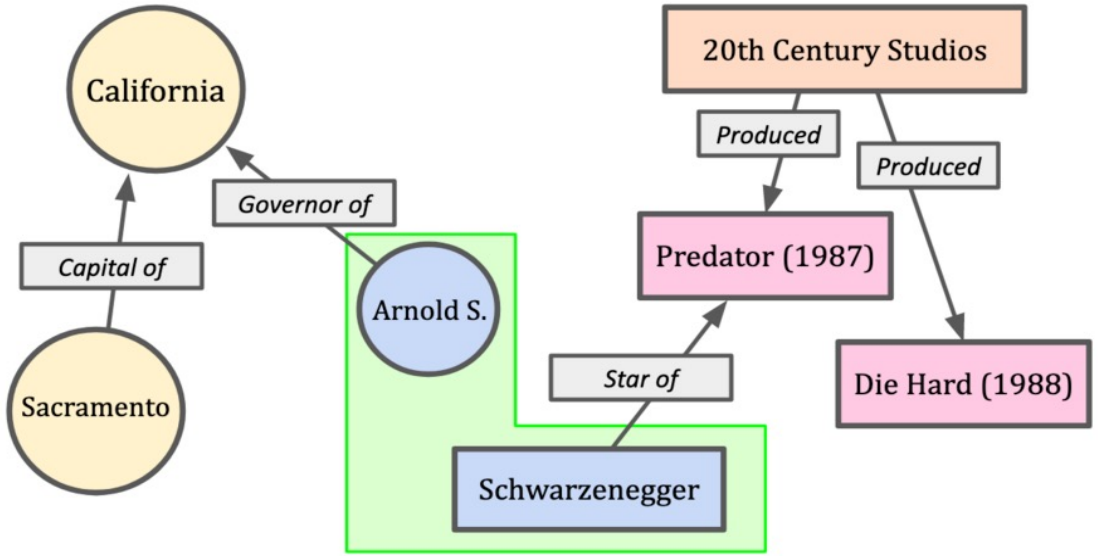
NER with GliNER

- https://github.com/urchade/GLiNER
- https://arxiv.org/pdf/2311.08526
- https://huggingface.co/urchade/gliner_medium-v2.1

How can we do “entity alignment”?

The recent successes of large language models (LLMs), in particular their effectiveness at producing syntactically meaningful embeddings, has spurred the use of LLMs in the task of entity alignment.

- https://aidanhogan.com/docs/art_wikidata_kgs_llms.pdf
- https://www.sciencedirect.com/science/article/pii/S266665102
- https://medium.com/@EleventhHourEnthusiast/knowledge-graph review-d3ec3964bcd4



Entity alignment

Source: https://en.wikipedia.org/wiki/Knowledge_graph#/media/File:Knowledge_graph_entity_alignment.png

Semantic Routing

INF Informatik
Hochschule Reutlingen

Semantic Routing

INF Informatik
Hochschule Reutlingen

Python Package semantic-router

The Semantic Router's potential extends far beyond basic routing and response modification. It's a versatile tool that can serve multiple purposes in enhancing AI interactions:

Protection Against Certain Queries: The Semantic Router can act as a safeguard, steering conversations away from unwanted or sensitive topics.

Efficient Function Calling: It enables function calling without the latency usually associated with agent processing. This makes the system faster and more responsive.

A New Approach to Retrieval-Augmented Generation (RAG): In the past, I've discussed different RAG methodologies, such as naive RAG and agent-based RAG. The Semantic Router introduces a new variant, which I like to call 'Semantic RAG.' This approach combines the power of agent-based RAG with the speed of naive RAG, offering the best of both worlds.

- https://medium.com/ai-insights-cobet/beyond-basic-chatbots-how-semantic-router-is-changing-the-game-783dd959a32d
- https://www.geeky-gadgets.com/semantic-router-superfast-decision-layer-for-llms-and-ai-agents/
- https://www.kaggle.com/code/adriensales/semantic-router-ollama-gemma2-hotline/notebook

DFA-RAG (Definite Finite Automaton - based Retrieval Augmented Generation)

The framework assumes that a specific workflow is embedded within the training data dialogues, which we model using a Definite Finite Automaton.

DFA-RAG acts as a Semantic Router like a decision-making layer.

DFA- RAG routes conversations through a predefined trajectory. This ensures that the LLM adheres to the workflow encoded in the DFA. Specifically, each conversational utterance cor- responds to a particular DFA state, where each state encapsu-lates responses from similar historical contexts.

https://arxiv.org/pdf/2402.04411

INF Informatik
Hochschule Reutlingen

Dominik Neumann
Hochschule Reutlingen, Alteburgstraße 150, 72762 Reutlingen
www.reutlingen-university.de
T. +49 172 9861157
dominik.neumann@reutlingen-university.de
dominik.neumann@exxeta.com