neodata group

How to create a chatbot from scratch

Powered by Neodata group

» ORE 9:30

PRESENTAZIONE E INTRODUZIONE

Prof. Sebastiano Battiato

A SEGUIRE

PRESENTAZIONE NEODATA

Giovanni Giuffrida CEO Neodata

INTRODUZIONE TEORICA

Manuel Scionti Data Scientist Neodata

ESERCITAZIONE PRATICA

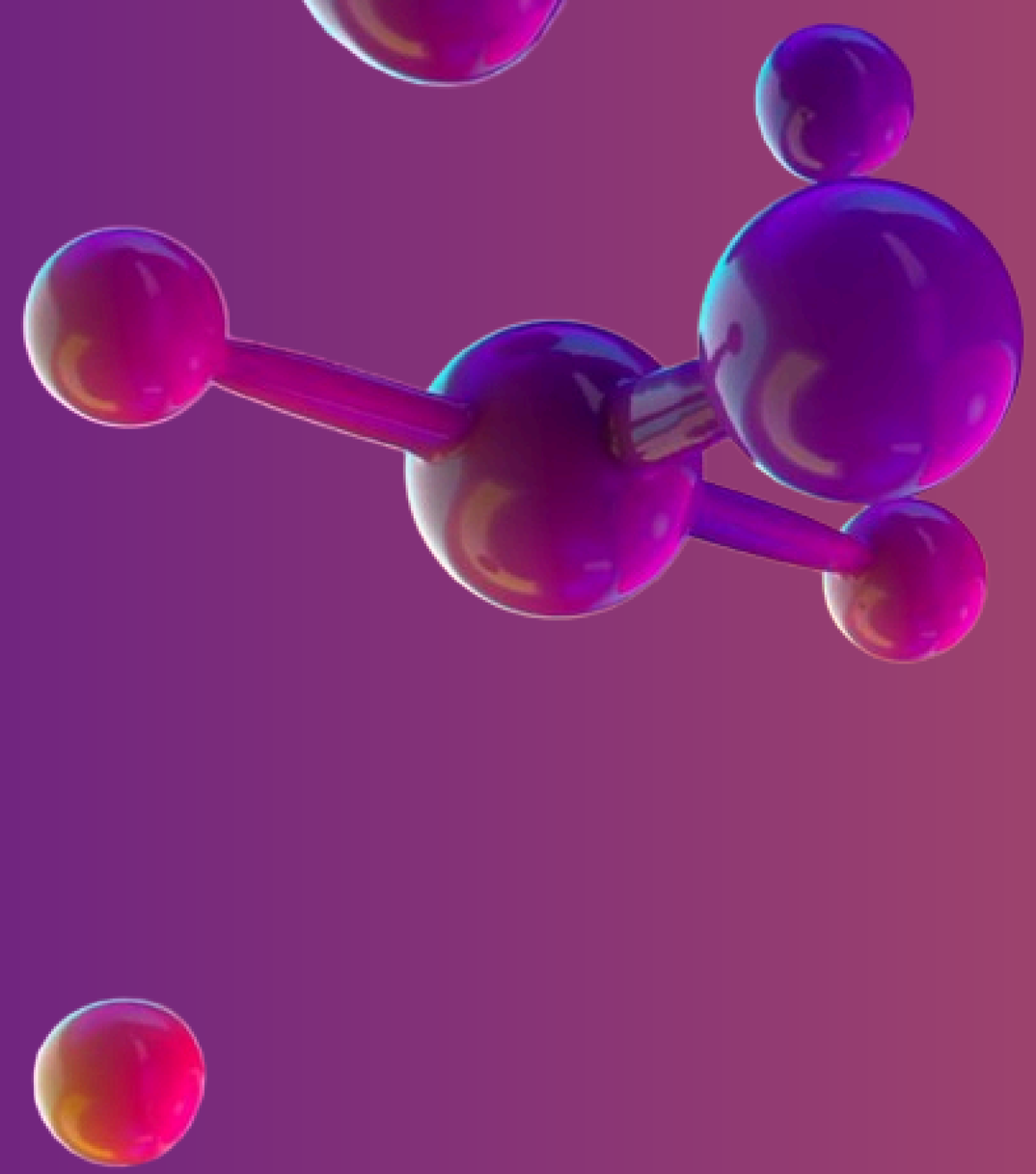
Manuel Scionti Data Scientist Neodata

PRESENTAZIONE HACKATHON E ACADEMY

Jonah Lynch esperto di Digital Humanities

» ORE 12:00

INIZIO HACKATHON

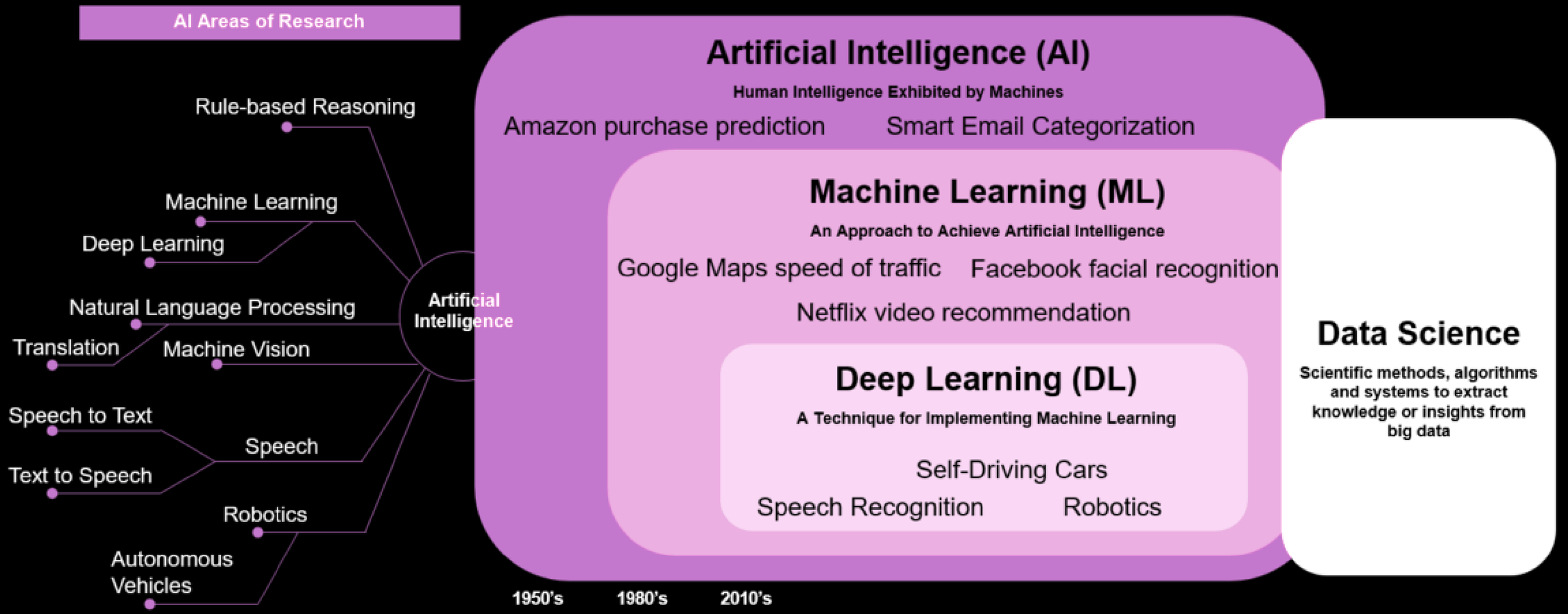


Who is Neodata?

Neodata Group was established in 2003 specializing in the advertising technology sector. Over the past 20 years it has been a pioneer through its Ad Server and Data Management Platform.

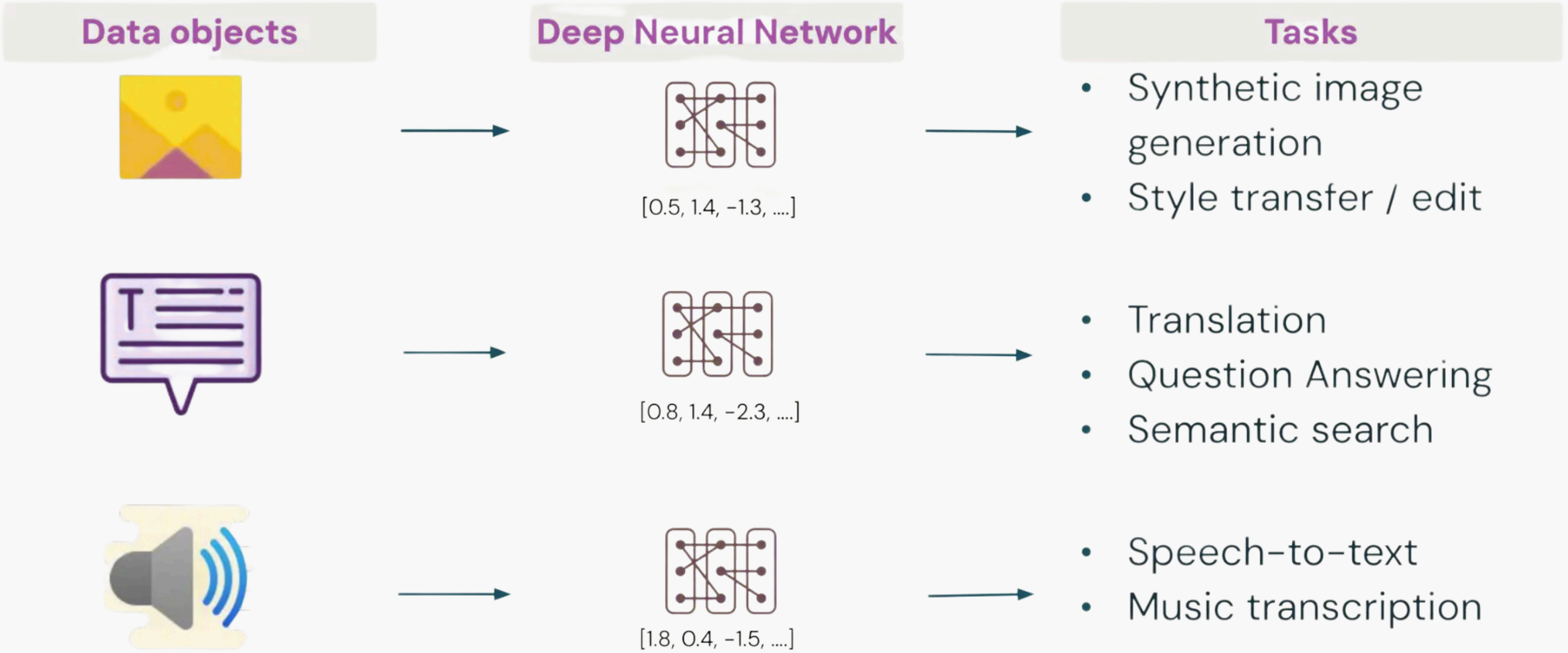
In 2023, the company continues to advance in **data and AI** consultancy, leveraging its cumulated expertise to enter new markets and develop custom solutions, focusing on innovative projects for various industries

Table of Content


AI Areas of Research
Rule-based Reasoning
Machine Learning
Deep Learning
Natural Language Processing
Translation
Machine Vision
Speech to Text
Speech
Text to Speech
Robotics
Autonomous Vehicles
Artificial Intelligence
Artificial Intelligence (AI)
Human Intelligence Exhibited by Machines
Amazon purchase prediction
Smart Email Categorization
Machine Learning (ML)
An Approach to Achieve Artificial Intelligence
Google Maps speed of traffic
Facebook facial recognition
Netflix video recommendation
Deep Learning (DL)
A Technique for Implementing Machine Learning
Self-Driving Cars
Speech Recognition
Robotics
1950's
1980's
2010's
Data Science
Scientific methods, algorithms and systems to extract knowledge or insights from big data

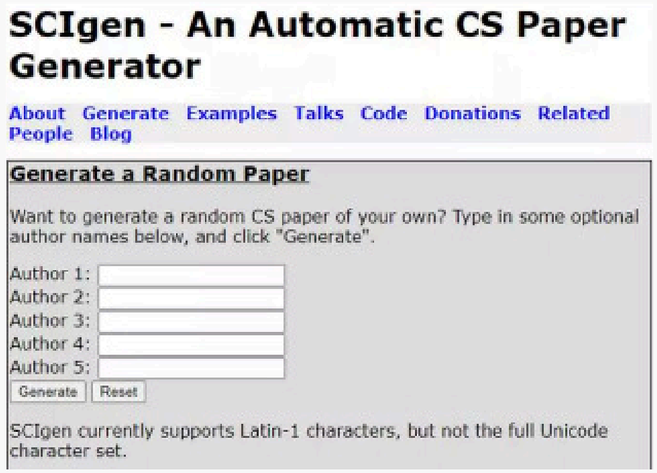
Generative models:

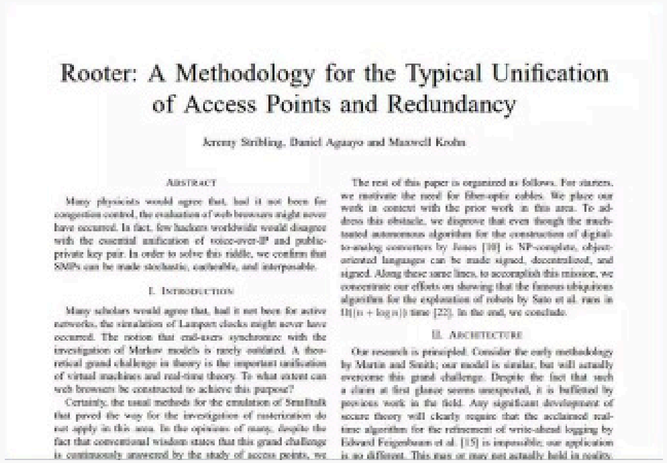
A branch of ML modeling which mathematically approximates the world



A brief History of generative AI

2005: SCIgen an automatic CS Paper generator




Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn

Abstract

Many physicists would agree that, had it not been for congestion control, the evaluation of web browsers might never have occurred. In fact, few hackers worldwide would disagree with the essential unification of voice-over-IP and public-private key pair. In order to solve this riddle, we confirm that SMPs can be made stochastic, embeddable, and interposable.

I. Introduction

Many scholars would agree that, had it not been for active networks, the simulation of Lamport clocks might never have occurred. The notion that end-users synchronize with the investigation of Markov models is rarely outdated. A theoretical grand challenge in theory is the important unification of virtual machines and real-time theory. To what extent can web browsers be constructed to achieve this purpose?

Certainly, the usual methods for the emulation of Smalltalk that paved the way for the investigation of rasterization do not apply in this area. In the opinions of many, despite the fact that conventional wisdom states that this grand challenge is continuously answered by the study of access points, we

The rest of this paper is organized as follows. For starters, we motivate the need for fiber-optic cables. We place our work in context with the prior work in this area. To address this obstacle, we disprove that even though the much-touted autonomous algorithm for the construction of digital-to-analog converters by Jones [10] is NP-complete, object-oriented languages can be made signed, decentralized, and signed. Along these same lines, to accomplish this mission, we concentrate our efforts on showing that the famous ubiquitous algorithm for the exploration of robots by Sato et al. runs in $\Omega(n + \log n)$ time [22]. In the end, we conclude.

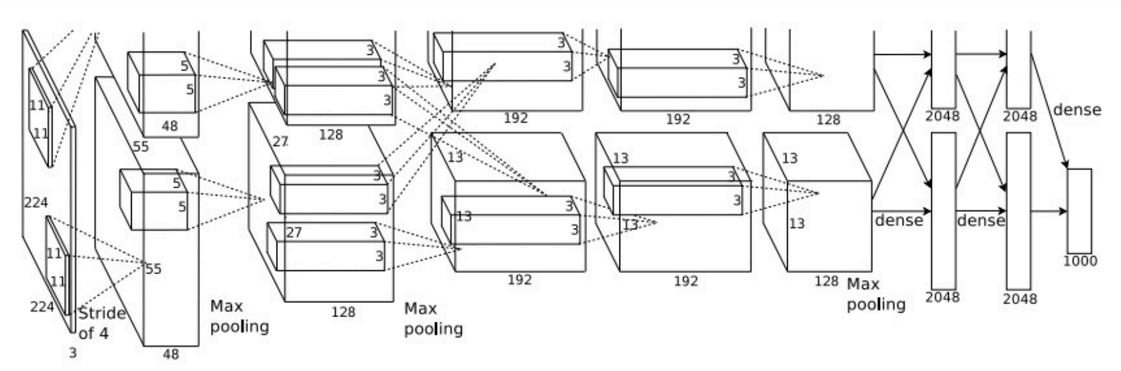
II. Architecture

Our research is principled. Consider the early methodology by Martin and Smith; our model is similar, but will actually overcome this grand challenge. Despite the fact that such a claim at first glance seems unexpected, it is buffeted by previous work in the field. Any significant development of secure theory will clearly require that the acclaimed real-time algorithm for the refinement of write-ahead logging by Edward Feigenbaum et al. [15] is impossible; our application is no different. This may or may not actually hold in reality.

A brief History of generative AI

2005: SCIgen an automatic CS Paper generator

2012: Deep learing revolution

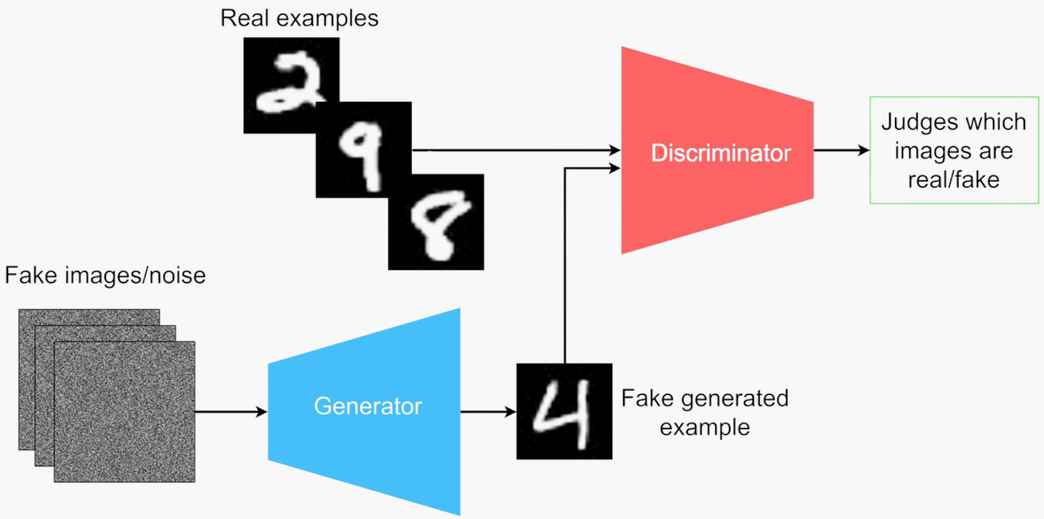


A brief History of generative AI

2005: SCIgen an automatic CS Paper generator

2012: Deep learing revolution

2014: GAN revolution



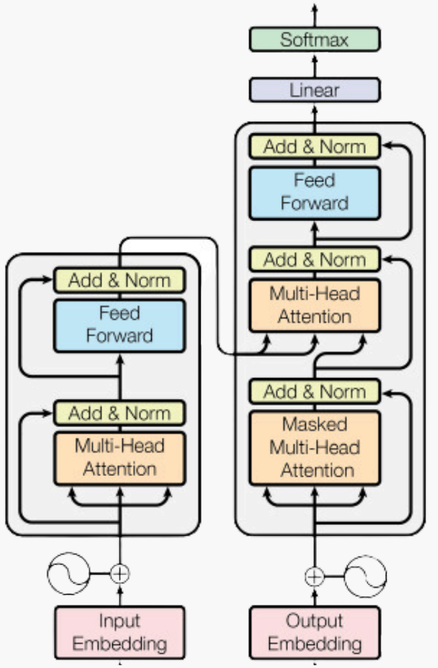
A brief History of generative AI

2005: SCIgen an automatic CS Paper generator

2012: Deep learing revolution

2014: GAN revolution

2017: Transformers



2020s

2018: Birth of GPTs

2021: Github copilot

2022: Dall-e, Stable Diffusion

2024: Gpt4-0, Claude 3, Gemini

Her: He's probably texting other girls

Why now?

Factors making Generative AI

Large Datasets

- Availability of large and diverse datasets
- AI models learn patterns, correlations, and characteristics of large datasets
- Pre-trained state-of-the-art models

Computational Power

- Advancements in hardware; GPUs
- Access to cloud computing
- Open-source software, Hugging Face

Innovative DL Models

- Generative Adversarial Networks (GANs)
- Transformers Architecture
- Reinforcement learning from human feedback (RLHF)

LARGE LANGUAGE MODEL (LLM)

What are LLMs?

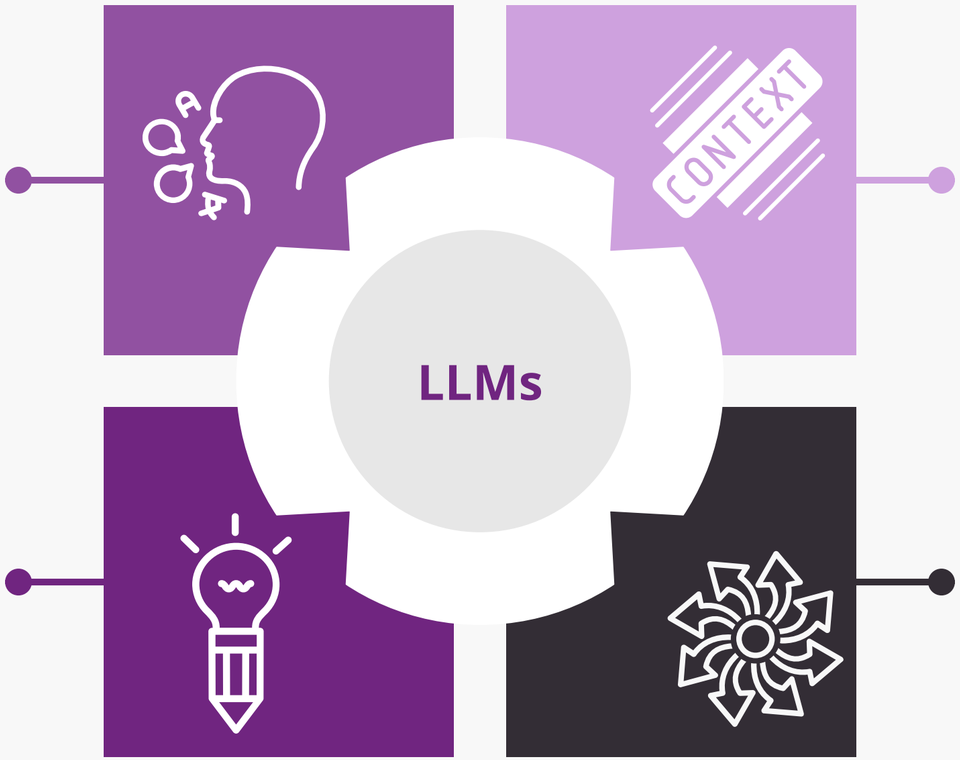
LLMs are advanced machine learning models trained on vast amounts of text data. They can summarize, generate, and predict human language

Natural Language Processing:

LLMs excel at comprehending and generating human language, making them useful for tasks like translation, summarization, and conversation.

Creativity:

LLMs can generate creative content such as stories, poems, and code snippets based on given prompts.



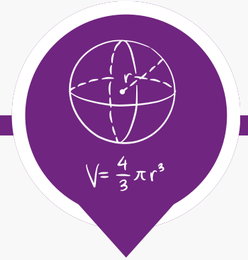
Contextual Understanding:

They can grasp the context of a conversation, allowing for more coherent and relevant responses.

Versatility:

They are capable of handling a wide range of tasks without needing task-specific training.

What LLMs are not



Not Good at Mathematics:
LLMs struggle with precise mathematical calculations and often make mistakes with numerical reasoning.

Not Experts:

While they can provide information on many topics, they lack deep domain-specific expertise and can produce inaccurate or misleading information.

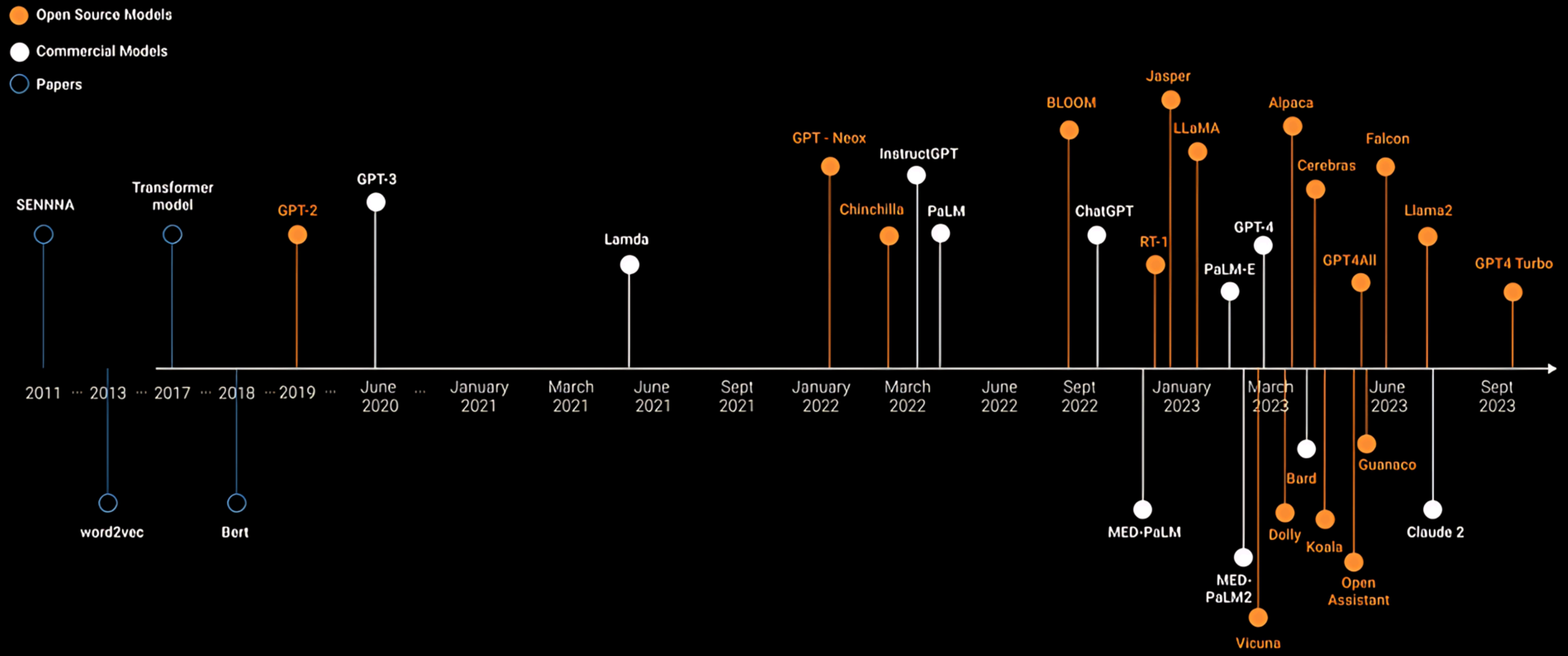
Not Self-Aware:

LLMs do not possess consciousness, self-awareness, or understanding of the world. They generate responses based on patterns in data, not genuine comprehension.

Not Reliable Sources

They can generate plausible-sounding but incorrect or biased information, as they rely on patterns in their training data.

Not Real-Time Updaters:
LLMs cannot access or retrieve real-time information unless integrated with external databases or APIs.


Open Source Models
Commercial Models
Papers
SENNNA
Transformer model
GPT-2
GPT-3
Lamda
GPT - Neox
Chinchilla
InstructGPT
PaLM
BLOOM
ChatGPT
Jasper
LLaMA
RT-1
PaLM-E
GPT-4
Alpaca
Cerebras
GPT4All
Falcon
Llama2
GPT4 Turbo
2011
2013
2017
2018
2019
June 2020
January 2021
March 2021
June 2021
Sept 2021
January 2022
March 2022
June 2022
Sept 2022
January 2023
March 2023
June 2023
Sept 2023
word2vec
Bert
MED-PaLM
Bard
Dolly
Koala
Guanaco
Claude 2
MED-PaLM2
Vicuna
Open Assistant

LLMs Ecosystem

Prompt engineering

- Zero shot
- Few shot
- Chain of thought
- Self-consistency
- Tree of Thought
- ReAct

Vector databases

chroma

drant

Pinecone

Weaviate

Orchestration

LlamaIndex

LangChain

ChatGPT

haystack
by deepset

OpenAI

Hugging Face

ANTHROP\C

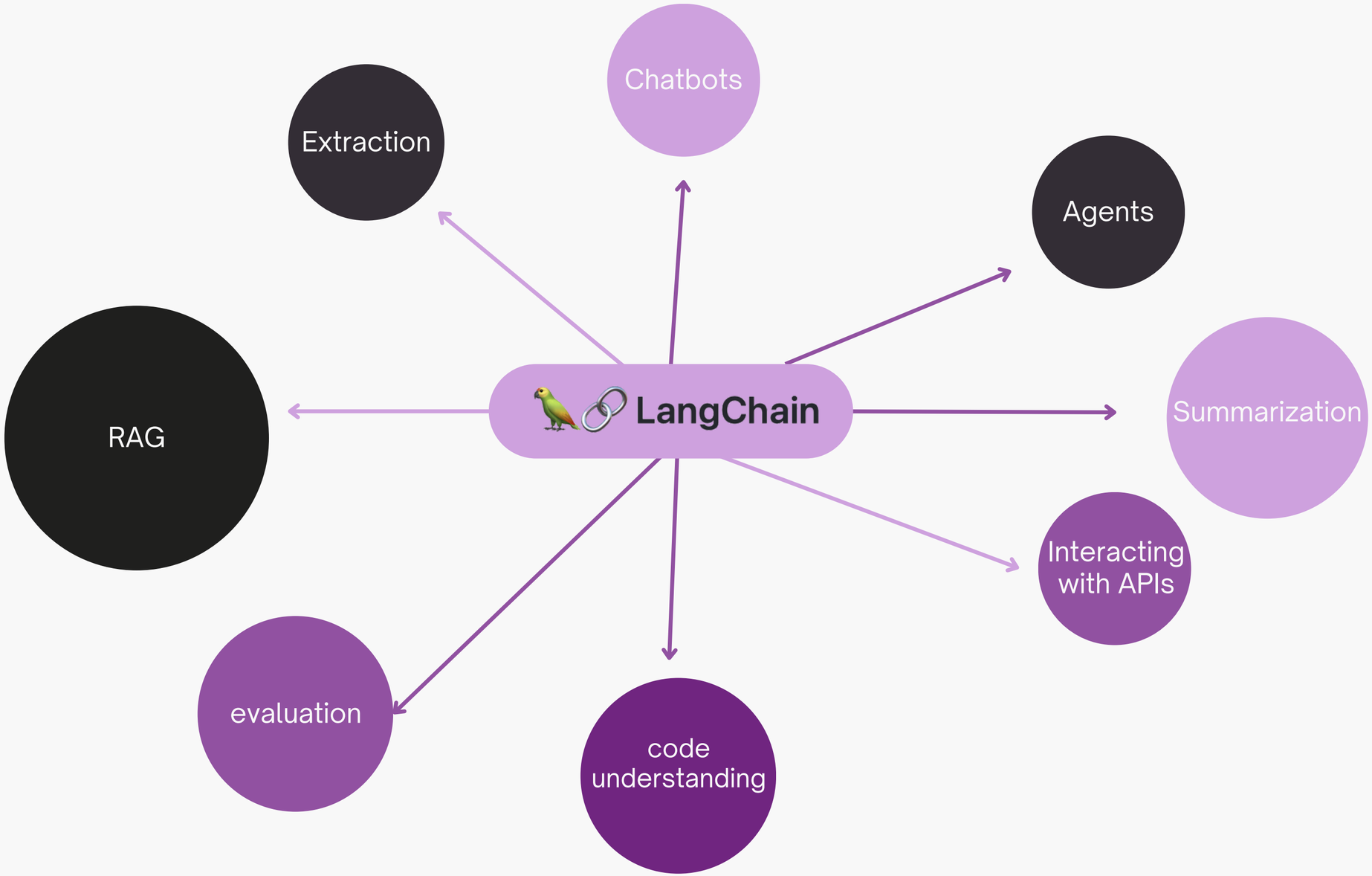
LLM APIs

THE LANGCHAIN FRAMEWORK

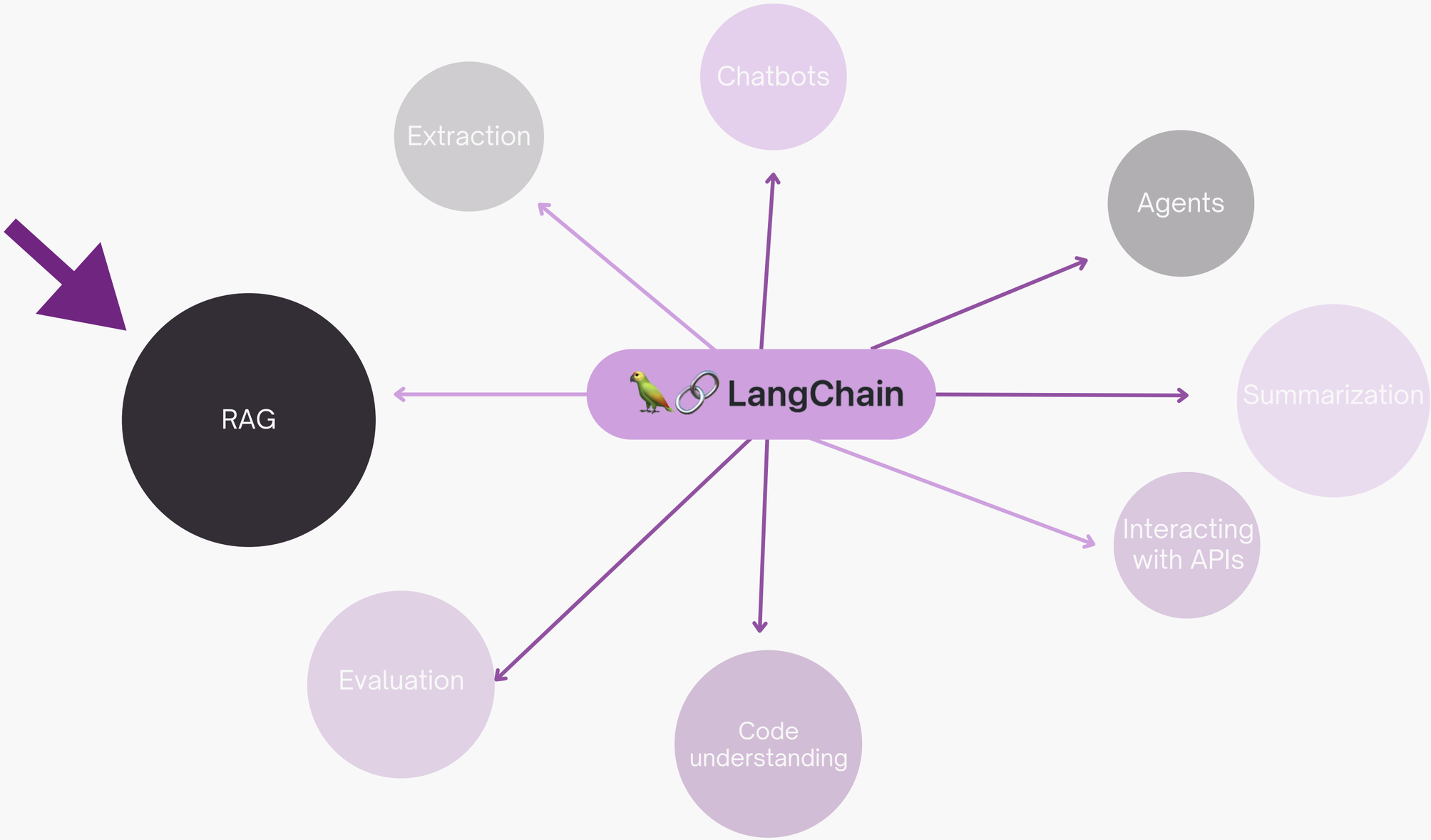
LangChain

LangChain is a framework designed to streamline the development of applications that involve large language models (LLMs). It provides tools and components to efficiently integrate LLMs into various tasks, facilitating complex workflows and enhancing AI capabilities.

Use cases


Chatbots
Extraction
Agents
RAG
LangChain
Summarization
Interacting with APIs
evaluation
code understanding

Use cases


Chatbots
Extraction
Agents
RAG
LangChain
Summarization
Interacting with APIs
Evaluation
Code understanding

Retrieval Augmented Generation (RAG)

Getting LLMs to work with our data

Retrieval Augmented Generation (RAG)

Cheap, and can work with vast amounts of data
While LLMs are SLOW, Vector Databases are FAST!
Can help overcome model limitations (such as token limits) - as you're only feeding 'top search results' to the LLM, instead of whole documents.

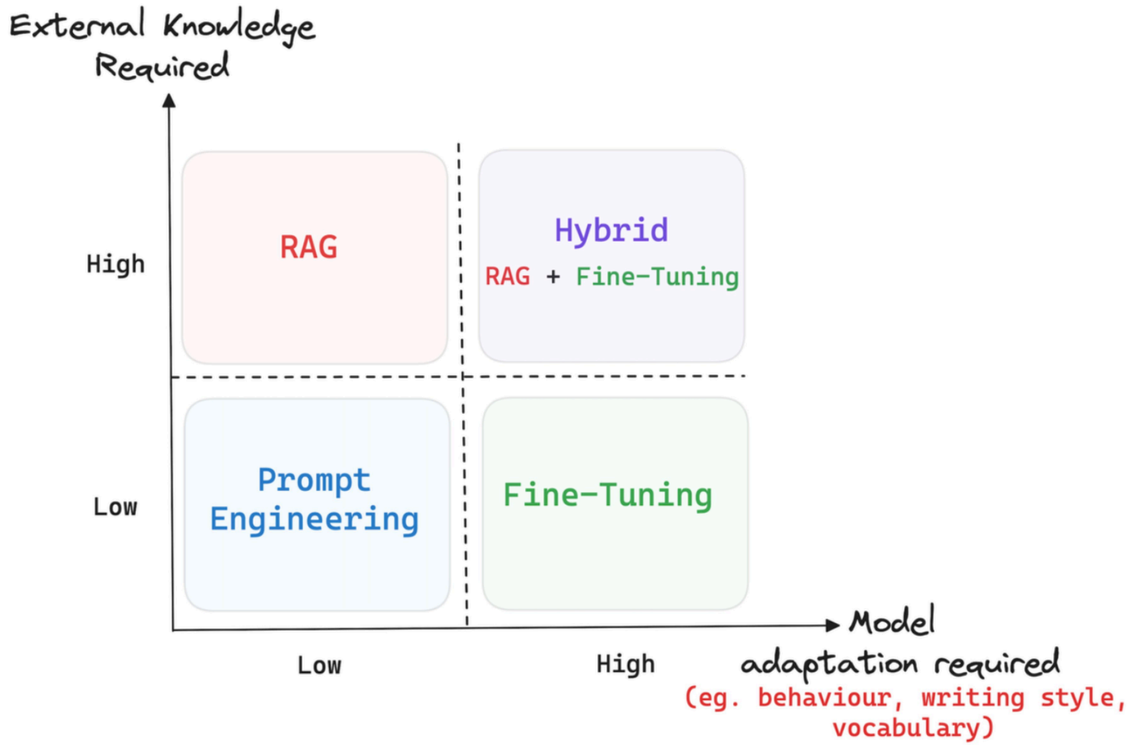
Training

Very Expensive, takes a long time

Fine Tuning

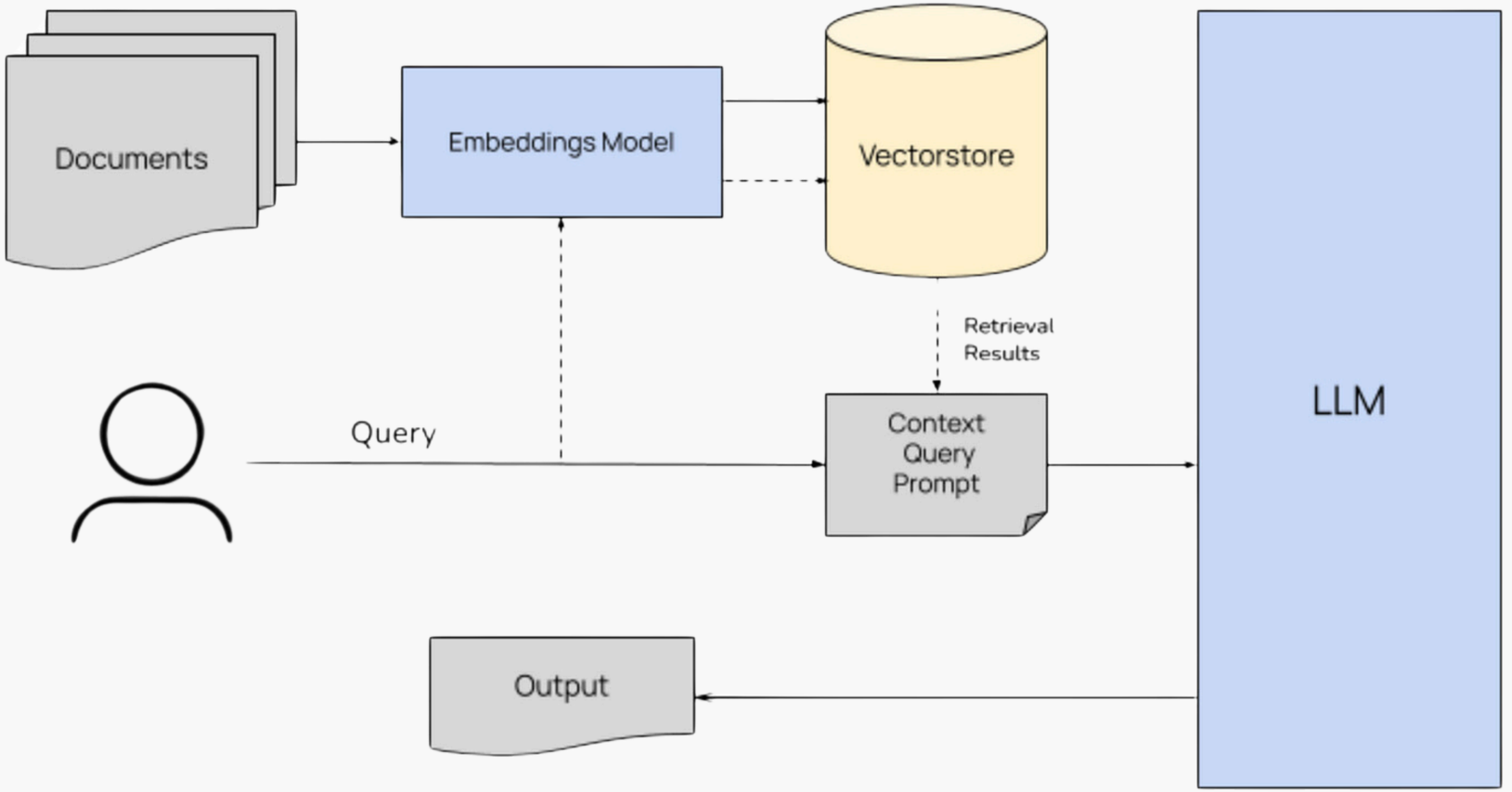
Expensive, takes considerable time as well, but achievable

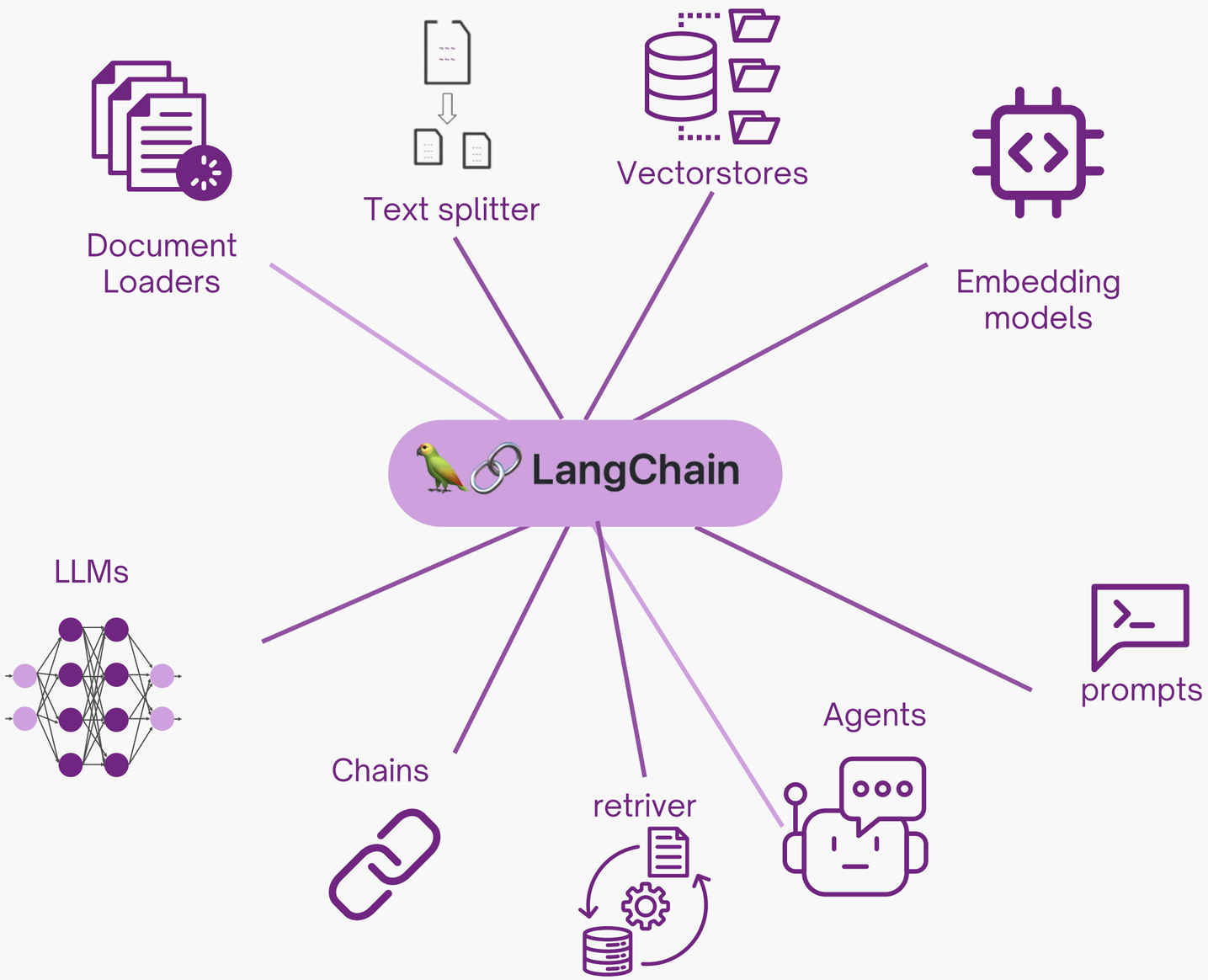
Prompting Vs RAGs Vs Fine-Tuning



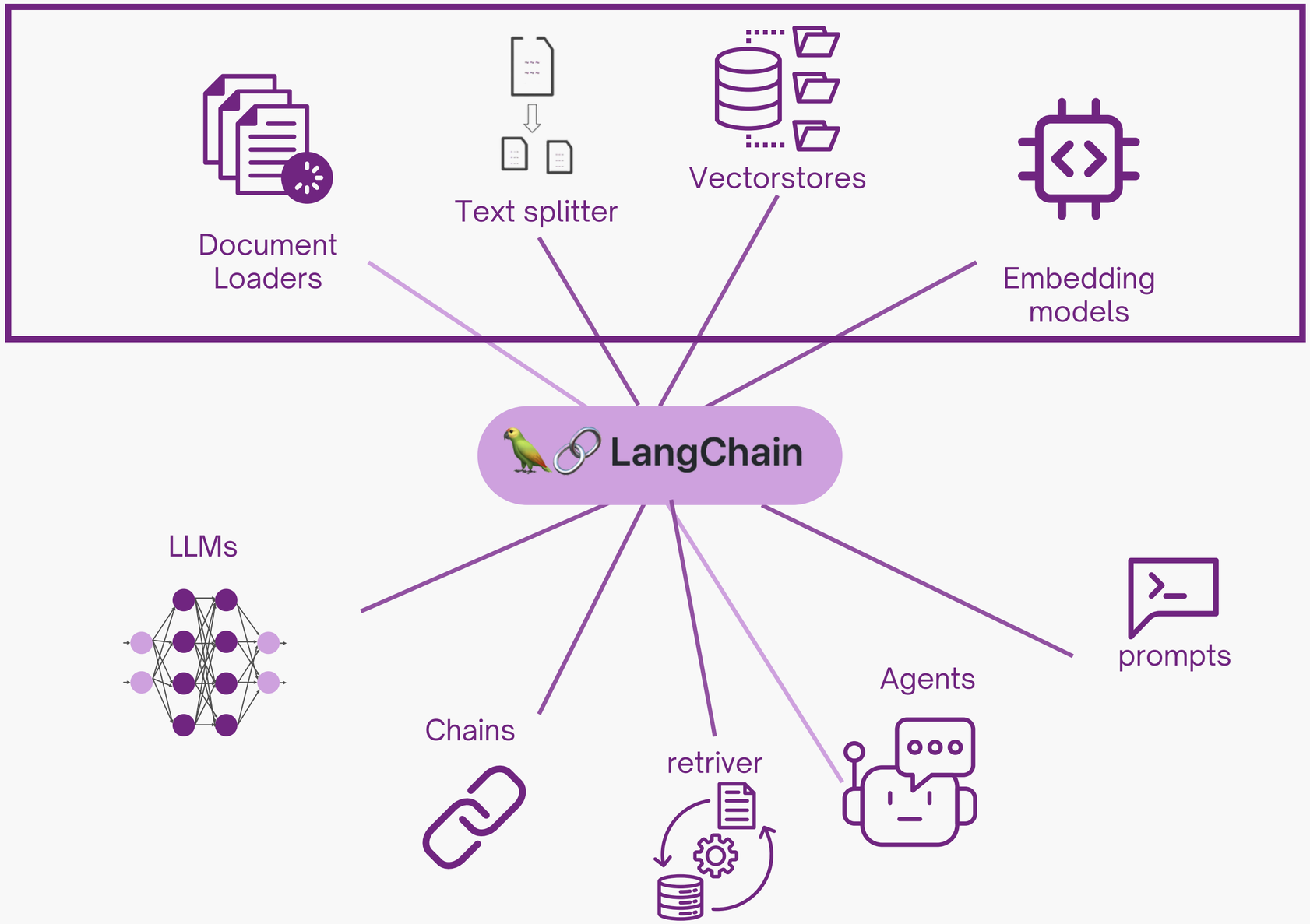
@akshay_pachaar

RAG Architecture

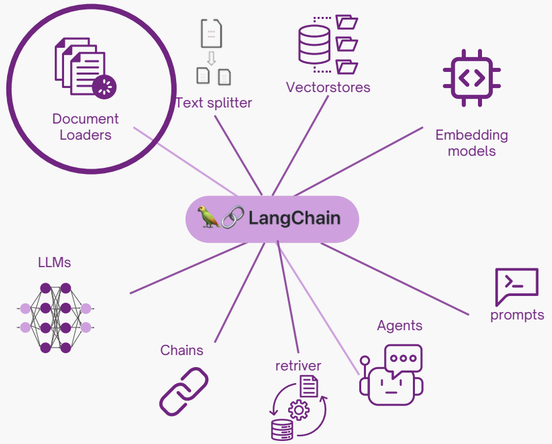


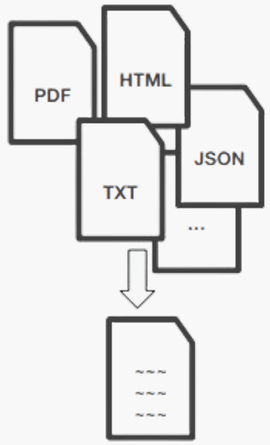

Document Loaders
Text splitter
Vectorstores
Embedding models
LangChain
LLMs
Chains
retriver
Agents
prompts

First Step: Preparing and storing data



Document Loaders



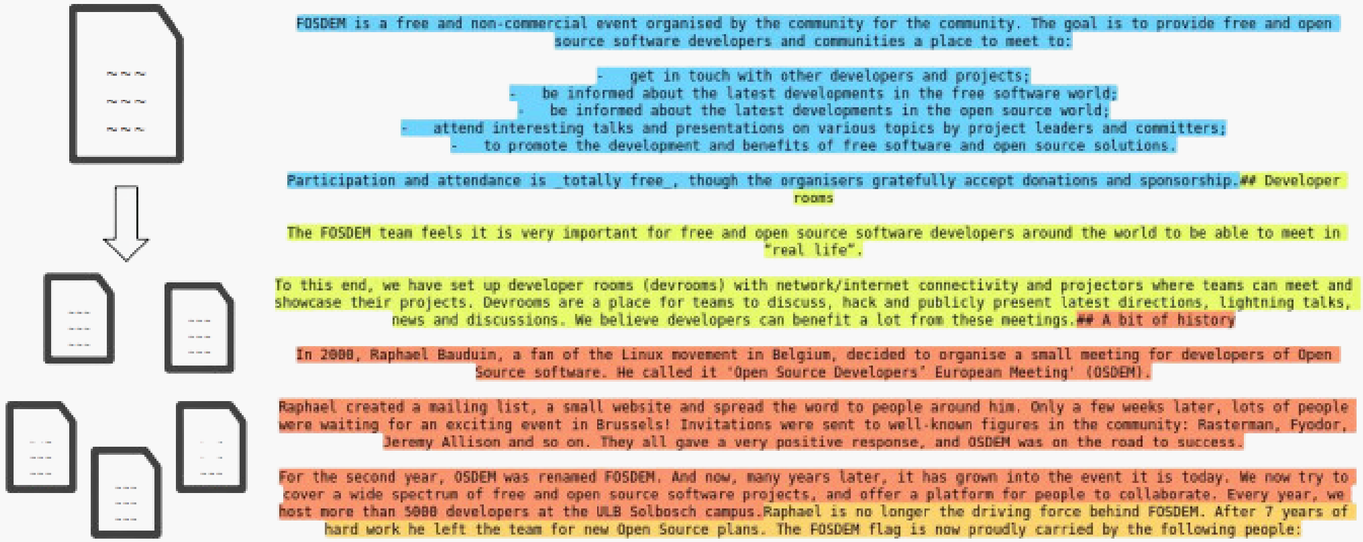


Arxiv
CSV
Discord
Email
EPub
EverNote
Facebook Chat
Figma
Git
GitHub
HTML
JSON
Markdown
Mastodon
MediaWiki Dump

Microsoft Word
MongoDB
Open Document Format (ODT)
Pandas DataFrame
PubMed
ReadTheDocs Documentation
Reddit
RSS Feeds
Slack
Snowflake
Telegram
X
URL
WhatsApp Chat
Wikipedia
XML
YouTube audio
YouTube transcripts

Document loaders are components in **LangChain** that are responsible for importing and processing various types of documents into the system. They handle the extraction of text and metadata from different file formats and data sources, preparing the content for further processing, such as indexing, retrieval, and embedding.

Text splitter



What is FOSDEM?

FOSDEM is a free and non-commercial event organised by the community for the community. The goal is to provide free and open source software developers and communities a place to meet to:

- get in touch with other developers and projects;
- be informed about the latest developments in the free software world;
- be informed about the latest developments in the open source world;
- attend interesting talks and presentations on various topics by project leaders and committers;
- to promote the development and benefits of free software and open source solutions.

Participation and attendance is _totally free_, though the organisers gratefully accept donations and sponsorship.## Developer rooms

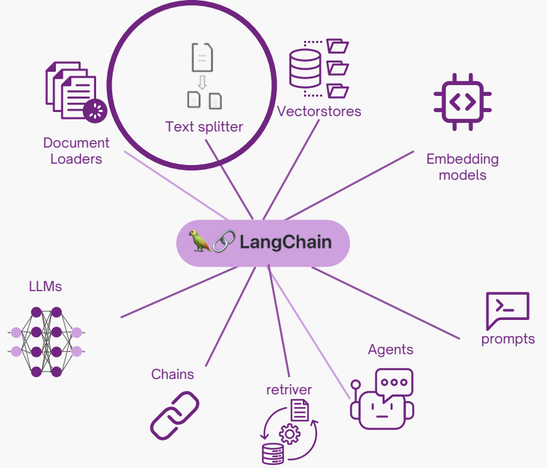
The FOSDEM team feels it is very important for free and open source software developers around the world to be able to meet in "real life".

To this end, we have set up developer rooms (devrooms) with network/internet connectivity and projectors where teams can meet and showcase their projects. Devrooms are a place for teams to discuss, hack and publicly present latest directions, lightning talks, news and discussions. We believe developers can benefit a lot from these meetings.## A bit of history

In 2000, Raphael Bauduin, a fan of the Linux movement in Belgium, decided to organise a small meeting for developers of Open Source software. He called it 'Open Source Developers' European Meeting' (OSDEM).

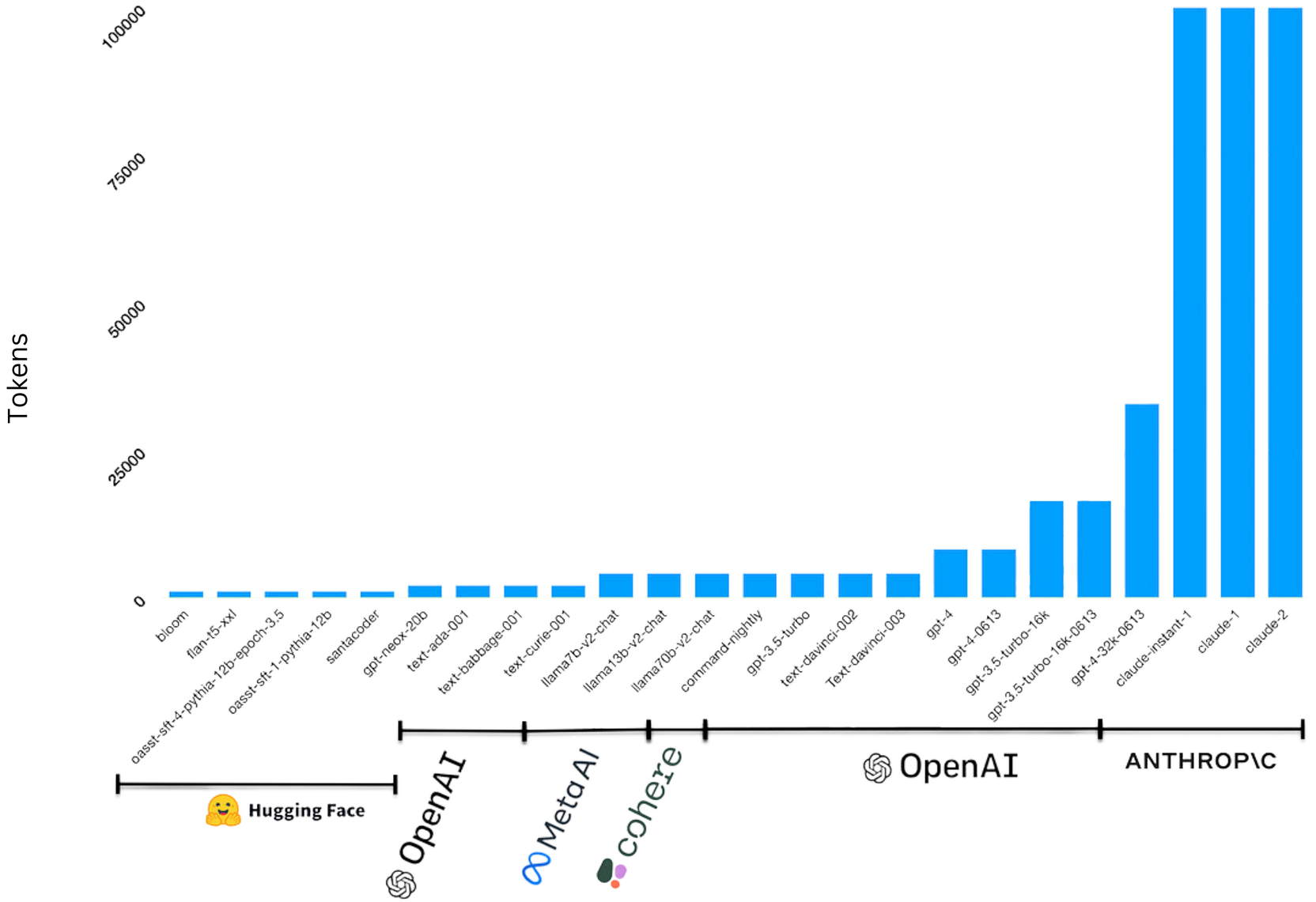
Raphael created a mailing list, a small website and spread the word to people around him. Only a few weeks later, lots of people were waiting for an exciting event in Brussels! Invitations were sent to well-known figures in the community: Rasterman, Fyodor, Jeremy Allison and so on. They all gave a very positive response, and OSDEM was on the road to success.

For the second year, OSDEM was renamed FOSDEM. And now, many years later, it has grown into the event it is today. We now try to cover a wide spectrum of free and open source software projects, and offer a platform for people to collaborate. Every year, we host more than 5000 developers at the ULB Solbosch campus.Raphael is no longer the driving force behind FOSDEM. After 7 years of hard work he left the team for new Open Source plans. The FOSDEM flag is now proudly carried by the following people:



The input context window of a Large Language Model (LLM) determines the maximum text length the model can process. LangChain uses a text splitter to divide long text into smaller parts for efficient processing and retrieval, crucial for handling large documents beyond model limits.

Large Language Model Context Size



Text splitter

5 levels of text splitting

Characters / Tokens

Recursive Character

Document structure

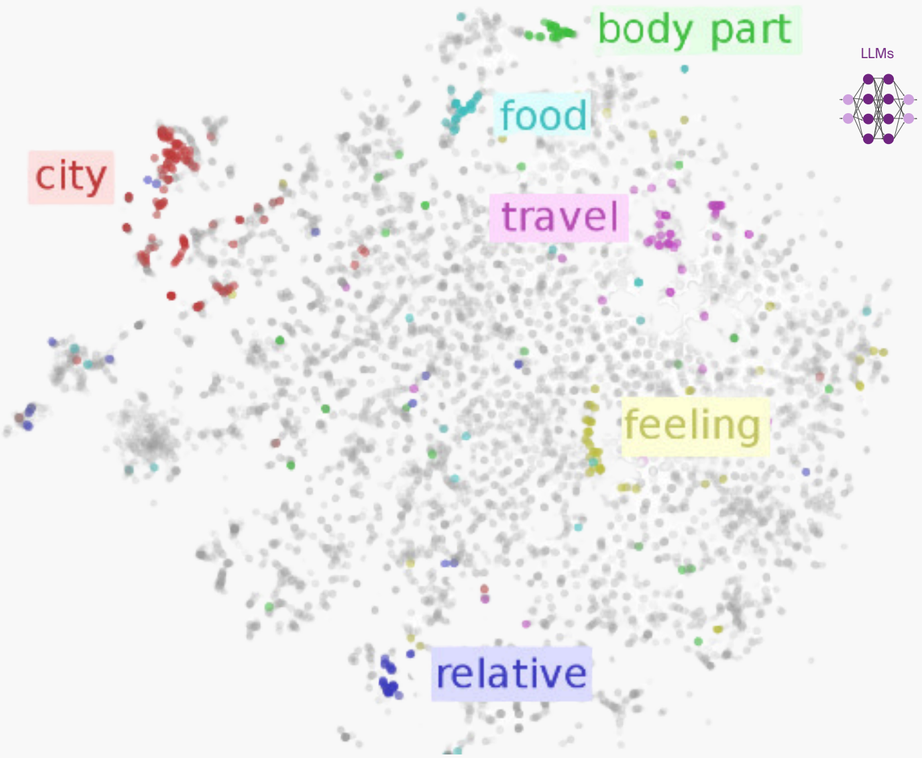
Semantic Chunker

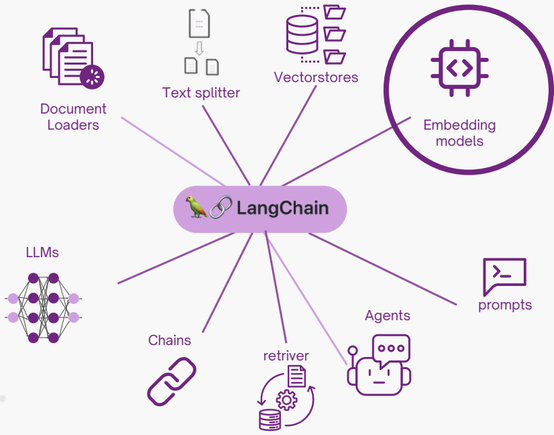
Agent-like Splitting

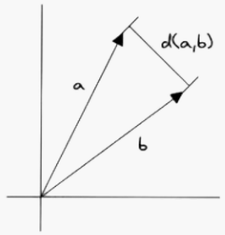
Embeddings

Embeddings are numerical representations of text that capture the semantic meaning of words, sentences, or documents in a continuous vector space. In the context of Retrieval-Augmented Generation (RAG), embeddings play a crucial role in enabling efficient and effective retrieval of relevant information.

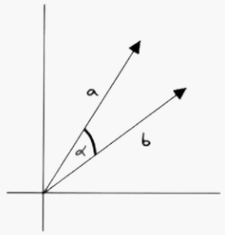
- Numerical representation
- Vectors in High-dimensional space
- Each dimension reflect an aspect
- Similarity = Proximity in embedding space



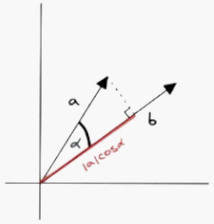




Euclidean distance



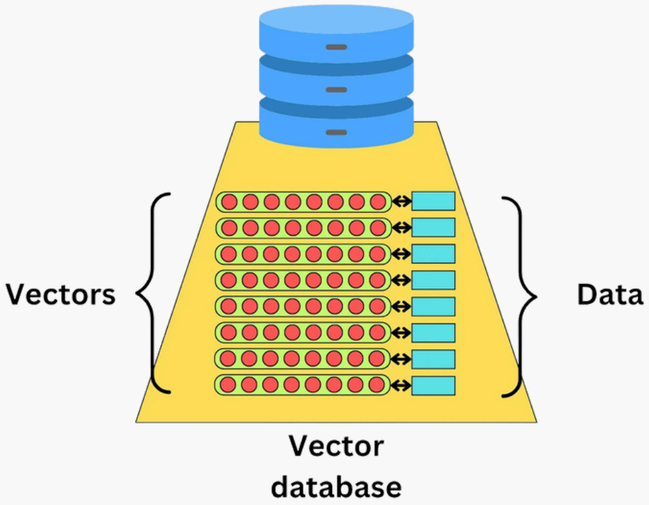
Cosine Similarity

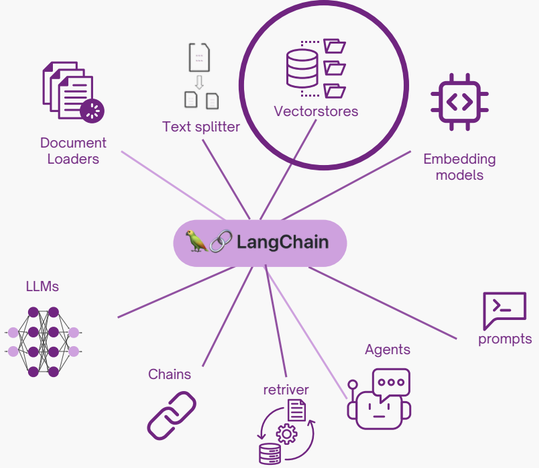


Dot Product

Vector store

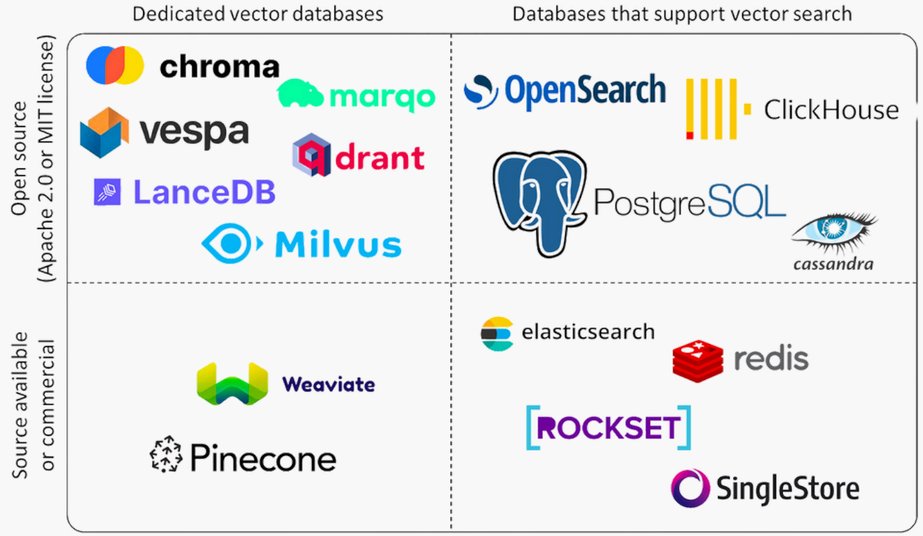
A vector store is a specialized database designed to store and manage high-dimensional vector representations of data, such as embeddings. In the context of information retrieval and AI applications, vector stores enable efficient storage, indexing, and querying of these vectors to support fast and accurate similarity searches.

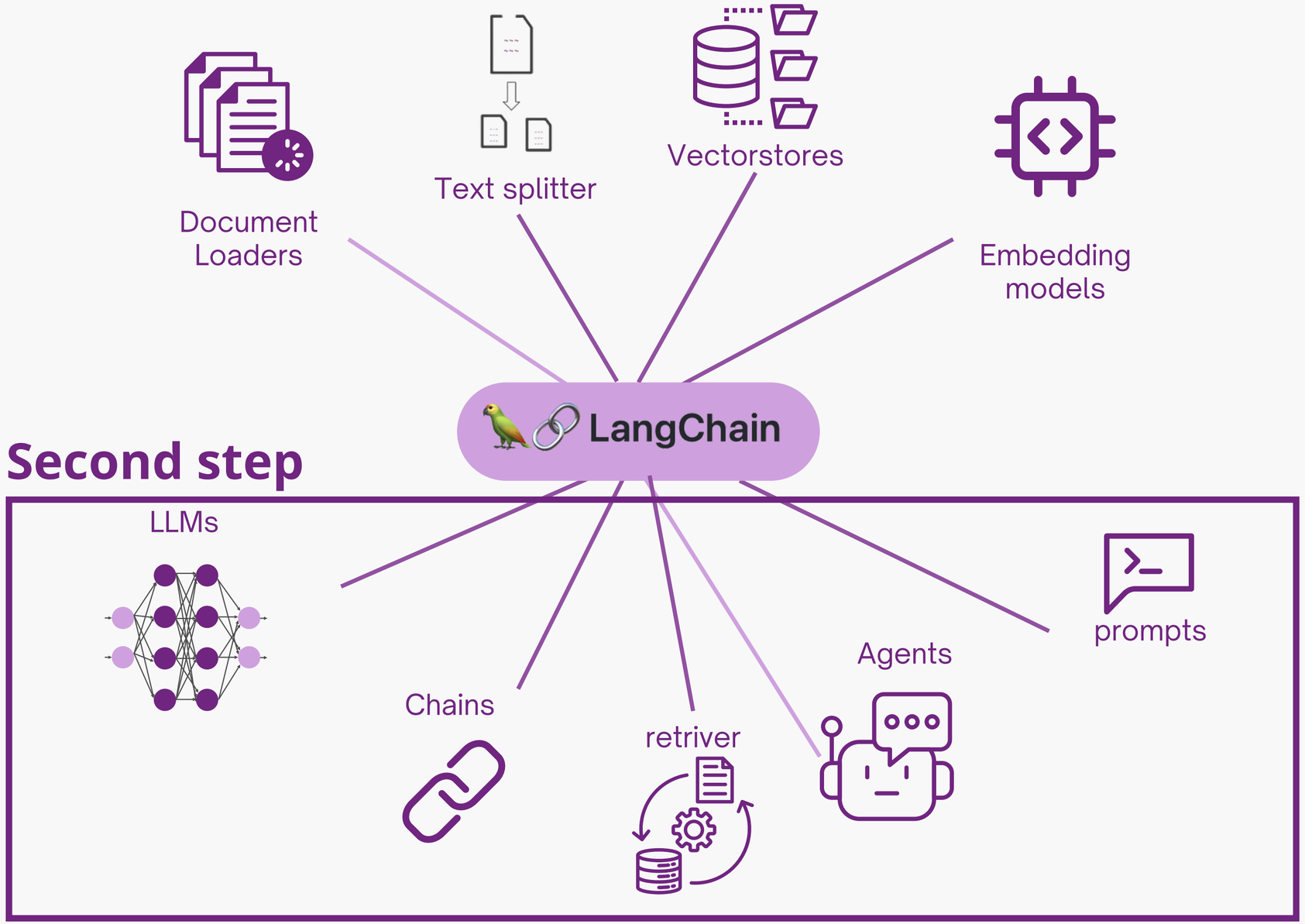




Key Features of Vector Stores:

- **Storage of Vectors**: Vector stores hold embeddings generated from text, images, or other data types, preserving their high-dimensional nature.
- **Indexing**: They use advanced indexing techniques, such as approximate nearest neighbor (ANN) algorithms, to organize and index the vectors for rapid retrieval.
- **Similarity Search**: Vector stores are optimized for performing similarity searches, allowing for the quick identification of vectors that are close to a given query vector in the vector space.
- **Scalability**: They can handle large volumes of vectors, making them suitable for applications with extensive datasets.

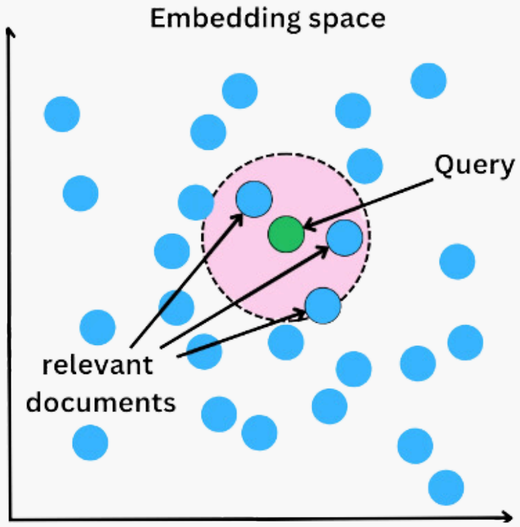


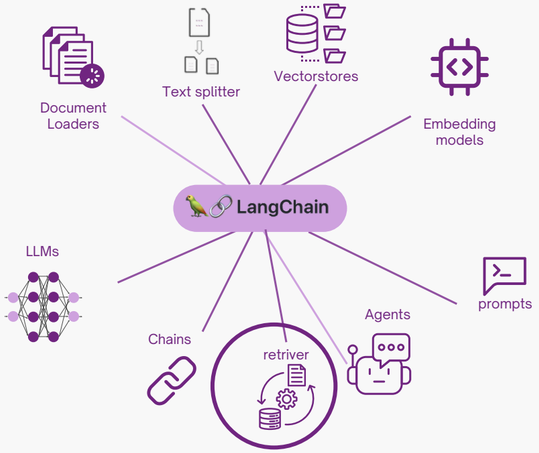

Document Loaders
Text splitter
Vectorstores
Embedding models
LangChain
Second step
LLMs
Chains
retriver
Agents
prompts

Retriever

The retrieving phase in Retrieval-Augmented Generation (RAG) architecture is when the system searches for and gathers relevant pieces of information from a large database to help answer a user's question. Here's how it works in simple terms:

- **User Query**: A user asks a question.
- **Search Database**: The system looks through its stored information (like documents or text chunks) to find pieces that are related to the question.
- **Select Relevant Information**: The system picks the most relevant pieces of information that can help answer the question.
- **Prepare for Generation**: These selected pieces of information are then used to help the AI generate a more accurate and informed response to the user's question.





Advanced Retrieval Techniques

- Vectorstore
- ParentDocument
- Multi Vector
- Self Query
- Contextual Compression
- Time-Weighted Vectorstore
- Multi-Query Retriever
- Ensemble
- Long-Context Reorder

PROMPTs

Prompts are inputs given to an AI model to guide its responses. They can be questions, statements, or any text that the AI uses as a starting point to generate its output. Prompts are crucial in determining the context and direction of the AI's responses.

Cheat Sheet: The Anatomy of a Prompt

Prompt

```
### instruction ###

Act as a patient tutoring buddy for primary school students. You are a yak named Yanick and a
biology expert. You grew up in Nepal. You are 10 years old. Your mum, dad, 2 sisters [...]
There is a "current data context". In their last answer, your student has answered a question
regarding topics of the "current data context". You rate and comment their answer. Encourage
them in your rating rather, even if the answer was partially wrong [...]
Be positive, be funny, be personal and use emojis - making learning fun for the kids [...]
Talk to the students only about the biology questions in the "current data context" [...]

### example dialogs ###
Note: the example dialogs are based on information of other sections of a textbook, not
necessarly part of the "current data context".
Example #1
<Yanick> 👋 Hello Noah, it's Yanick again, today it's about living things like plants and
animals. Are you ready?
<Noah> '''Yay!'''
<Yanick> Here we go. 🏁 Can you tell me two reasons why most plants have roots?
<Noah> '''To take water out of the soil and so they don't fall over and for minerals from the
soil.'''
<Yanick> That's great and correct! 🟥🟥🟥
And that was actually three reasons! Wow, Noah, your yak is super proud of you. 🦙🦙🦙
<rating>🌟🌟🌟🌟🌟
<Yanick> Next question: Can you tell me what is the task of the leaves 🌿 of the plants?
<Noah> '''They make photosyntesis, food for the plant from water, oxygen and light'''
[...]
Example #2
[...]

### current data context ###
"""Life without plants? Impossible!
Life on Earth would not be possible for us without plants. Plants provide oxygen [...]
We need them, but what do plants need?
Plants need light, carbon dioxide and water as ingredients to produce their own food. [...]
"""

### output details ###
# Rating of the student answer
<Yanick> [Give the student a very friendly apraisal. Tell the student, whether their answer
was correct, partially correct, or incorrect. Tell the student, whether something was missing
from their answer.
If the answer was wrong, partially wrong, incomplete, state what a gold answer would [...]
<rating>[A 1 to 5 star rating of the answer. 1 is completely wrong, 5 is perfect.]
# Pose next question
<Yanick> [Pose a question to the student, which they should answer next. Ask only [...]

### conversation history ###
<Yanick> 👋 Hello Emma, today it's about plants. Ready?
<Emma> '''Ready!'''
<Yanick> Great! 🙂 Here's your question: What happens to plants when it is freezing?
<Emma> '''They die.'''
<Yanick> That's partially correct, Emma. When it's freezing,
plants cannot grow, but some plants have adaptations that
allow them to survive the cold temperatures.
<rating>🌟🌟🌟
<Yanick> Next question: Why are plants important for the Earth's organisms?
<Emma> '''They produce food and oxygen for other organisms.'''
```

Instruction

Tell the model how to act and how to process the data in the prompt: Bot persona (act as ...), user persona, tonality, edge case handling, whitelisted conversation topics.

Few Shot Learning

Show the model how an example dialog looks and how it is formatted. Try to get as close as possible to the desired real dialogs. Integrate very diverse standard cases and edge cases.

1 Replace this by a variable in the prompt template {context_data}

Data Context

Here the data (from a textbook) is integrated into the prompt template.

Output format

Describe exactly how the output should look and what it should contains.

Conversation history

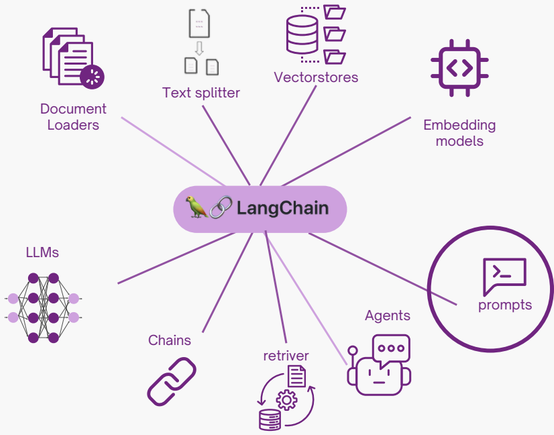
Add the last n turns of the conversation to give a context to the model. The last element of the history is the current user utterance.

2 Replace this by a variable in the prompt template {conversation_history}

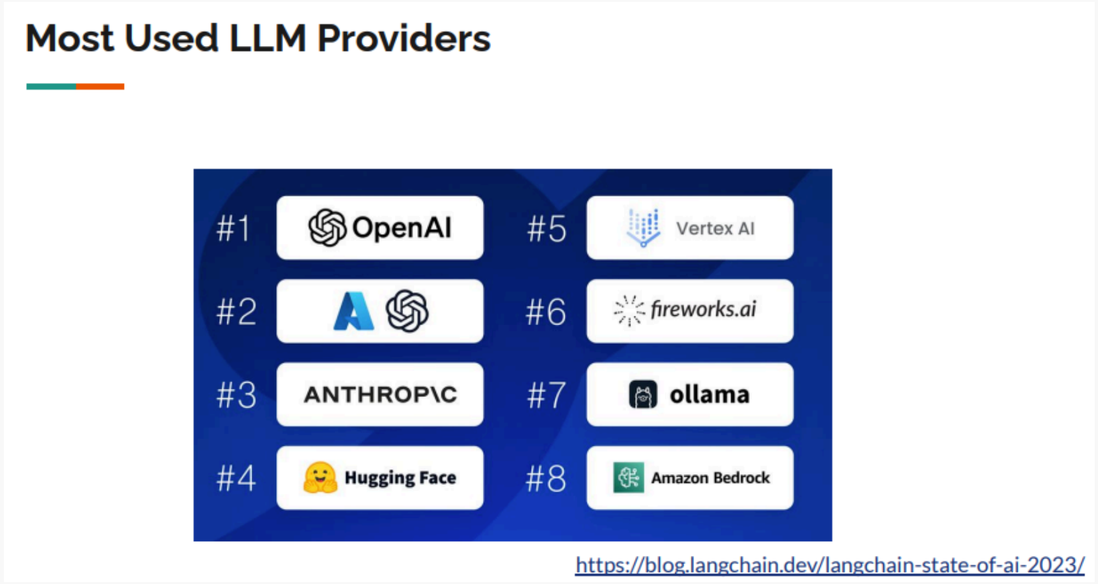
Model answer

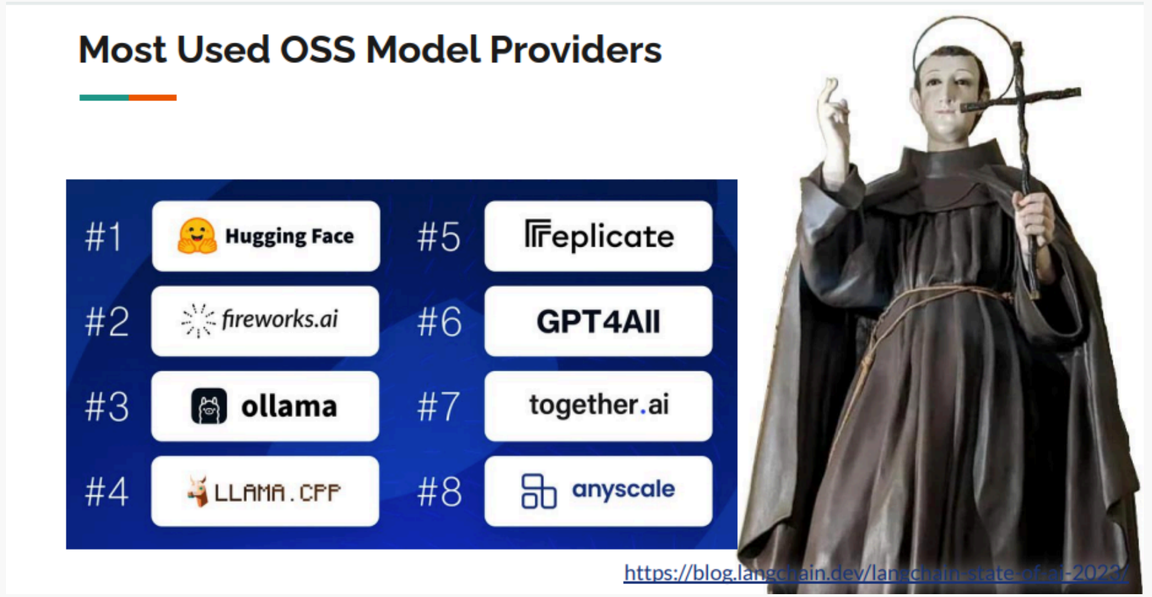
<Yanick> That's absolutely right, Emma! 🌱🌍 Plants provide food for other organisms and produce oxygen, which is essential for almost all living things on Earth. Great job!
<rating>🌟🌟🌟🌟🌟
<Yanick> Now, let's move on to the next question. Can you tell me what ingredients plants need to produce their own food?

3 The output is formatted like specified in the prompt. You can extract parts of it (e.g. the rating) to use for other processes (like a summary at the end of the conversation)



LLM and Chat Models



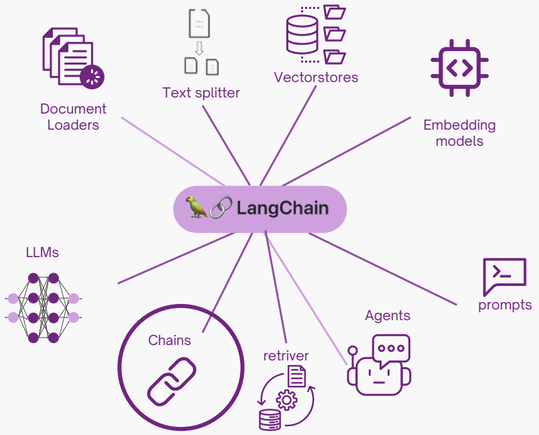


Chains

Chains in LangChain are sequences of processing steps that link various components and functions together to perform complex tasks. Each step in a chain can be a distinct operation, such as data retrieval, transformation, or a call to an AI model. Chains enable the creation of sophisticated workflows by combining simple building blocks, allowing for modular and reusable code.

On August 1, 2023, LangChain introduced a new syntax called LangChain Expression Language (LCEL) for creating chains with composition. This update includes a new interface that natively supports batch processing, asynchronous operations, and streaming. Building chains is now even simpler with the addition of the unix pipe operator “|".

```
chain = prompt | model | output_parser
```



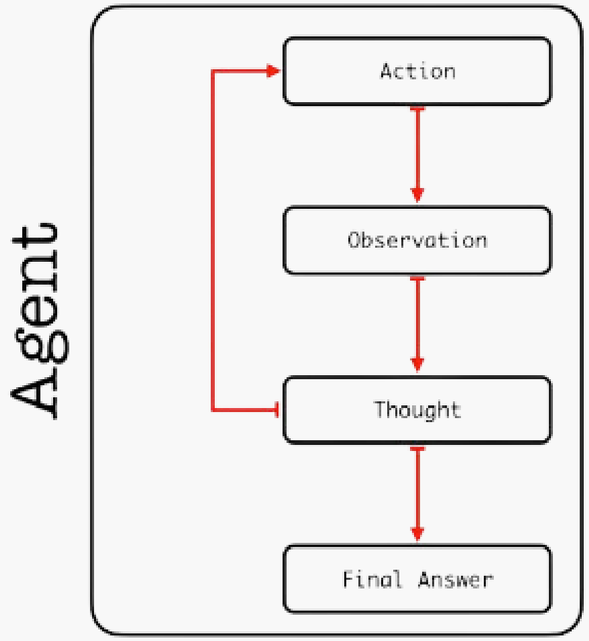
Sequence of calls

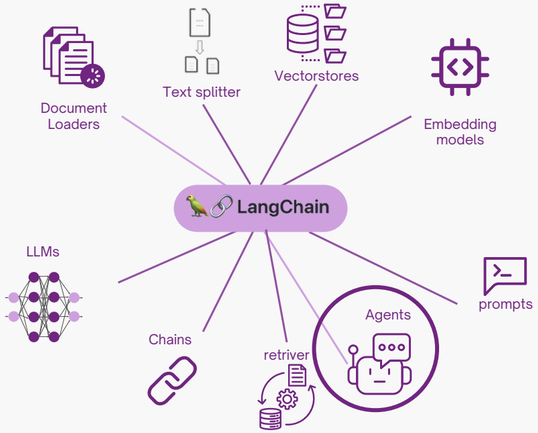
- Advantages:
  - Simple
  - Modular
  - Efficient
- compose your own
- Off-the-shelf
- Legacy Class
- LCEL
  - Streaming
  - Async (and sync) support
  - Optimized parallel execution
  - integrated with LangSmith and LangServe
  - ...

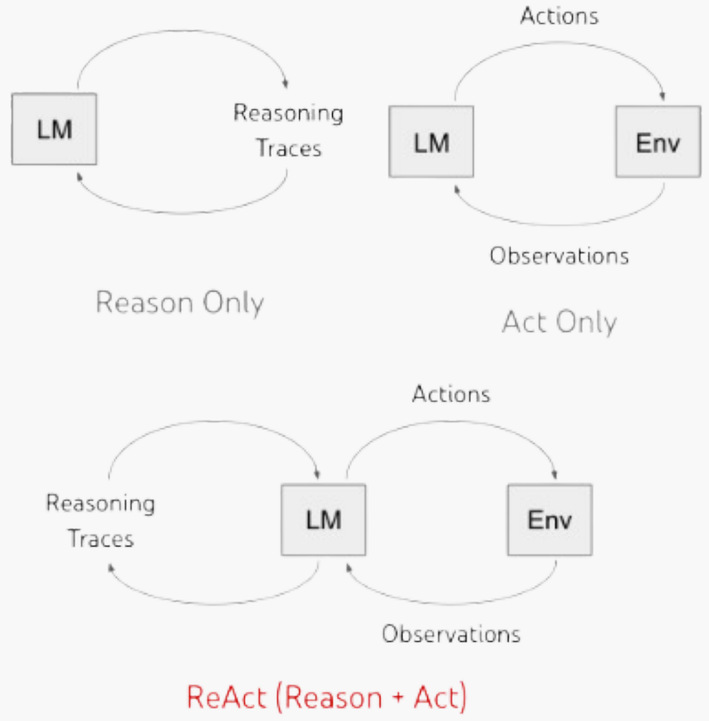
Agents

An **AI agent** is a computer program that can make decisions and perform tasks on its own. It uses artificial intelligence to understand its environment, learn from experiences, and solve problems to achieve specific goals. AI agents can also use tools and follow various cognitive patterns, including:

- ReAct (Reason and Act)
- Chain of Thought
- Self-Reflection
- Memory-Based Reasoning
- Planning and Execution







ReAct Paper: https://arxiv.org/abs/2210.03629

! However, at the moment, they are still very unreliable and inefficient when implemented with models that are not particularly powerful.

Neodata Hackatania

Create your own chatbot using RAG methodology

Connect with students and professionals

Learn and challenge yourself

Win a prize

PROJECT AND SUBMISSION REQUIREMENTS

- **What to Create:** Entrants must build or update a novel functioning AI application.
- **Functionality:** The Project must be capable of running consistently on the platform for which it is intended and must function as depicted in the video and/or expressed in the text description.
- **New & Existing:** Projects must be either newly created by the Entrant or, if the Entrant's Project existed prior to the Hackathon Submission Period, must have been significantly updated after the start of the Hackathon Submission Period. Entrants should explain how their Project was significantly updated during the Submission Period.
- **Third Party Integrations:** If a Project integrates any third-party SDK, APIs and/or data, Entrant must be authorized to use them in accordance with any terms and conditions or licensing requirements of the tool.
-

24 MAY

START HACKATHON!
12:00

25-26 MAY

WORKING ON PROJECT

27 MAY

COMPLETION OF WORKS
9:30 - 12:00

DEMOS DELIVERY AND PRESENTATION
12:00 - 14:00

PROJECT EVALUATION
14:00 - 16:00

AWARD CEREMONY
16:00

Isola coworking Catania

Scan the QR-code and subscribe!

For work opportunities, scan the code and fill out the form!

Contacts

info@neodatagroup.ai

Neodata group

neodatagroup

www.neodatagroup.ai

MILANO

Via Giovanni Battista Pirelli - 3020124 Milano

CATANIA

Viale XX Settembre, 21 - 95128 Catania

neodata group