

Large Language Models in Domino Applications

Serdar Basegmez

Developi Information Systems, London, UK



Serdar Basegmez

- Developer/Half-blooded Admin
- New(ish) Londoner - Ex-Istanbulite
- Freelance Consultant at Developi UK
- Member Director at OpenNTF Board

- Notes/Domino since 1999
- IBM Champion Alumni (2011-2018)
- HCL Ambassador (2020-2024)

- Blog: LotusNotus.com / Twitter: @serdar_basegmez
- Also tweets/writes/speaks/podcasts on scientific skepticism

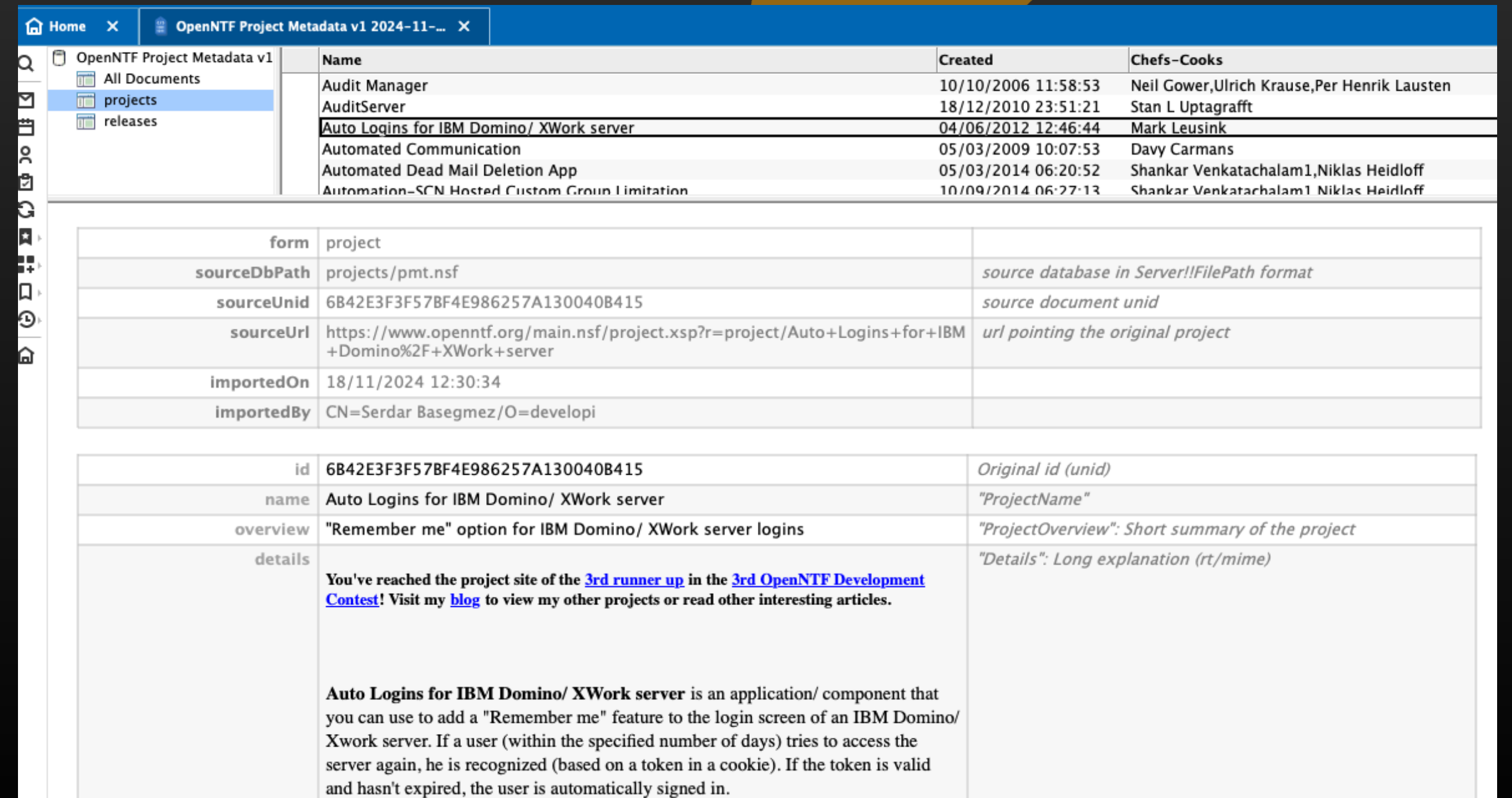


Everything Open Source!

Demos, source codes, libraries, integrations, datasets...



 <https://github.com/sbasegmez>

A screenshot of the OpenNTF Project Metadata v1 web application. The interface shows a table of project metadata and a detailed view for a specific project. The table lists various projects with their names, creation dates, and authors. The detailed view for the "Auto Logins for IBM Domino/ XWork server" project shows its source database path, source URL, and a detailed description of the project's functionality.

Name	Created	Chefs-Cooks
Audit Manager	10/10/2006 11:58:53	Neil Gower,Ulrich Krause,Per Henrik Lausten
AuditServer	18/12/2010 23:51:21	Stan L Uptagrafft
Auto Logins for IBM Domino/ XWork server	04/06/2012 12:46:44	Mark Leusink
Automated Communication	05/03/2009 10:07:53	Davy Carmans
Automated Dead Mail Deletion App	05/03/2014 06:20:52	Shankar Venkatchalam1,Niklas Heidloff
Automation-SCN Hosted Custom Group Limitation	10/09/2014 06:27:13	Shankar Venkatchalam1 Niklas Heidloff

form	project	
sourceDbPath	projects/pmt.nsf	source database in Server!FilePath format
sourceUnid	6B42E3F3F57BF4E986257A130040B415	source document unid
sourceUrl	https://www.openntf.org/main.nsf/project.xsp?r=project/Auto+Logins+for+IBM+Domino%2F+XWork+server	url pointing the original project
importedOn	18/11/2024 12:30:34	
importedBy	CN=Serdar Basegmez/O=developi	

id	6B42E3F3F57BF4E986257A130040B415	Original id (unid)
name	Auto Logins for IBM Domino/ XWork server	"ProjectName"
overview	"Remember me" option for IBM Domino/ XWork server logins	"ProjectOverview": Short summary of the project
details	<p>You've reached the project site of the 3rd runner up in the 3rd OpenNTF Development Contest! Visit my blog to view my other projects or read other interesting articles.</p> <p>Auto Logins for IBM Domino/ XWork server is an application/ component that you can use to add a "Remember me" feature to the login screen of an IBM Domino/ Xwork server. If a user (within the specified number of days) tries to access the server again, he is recognized (based on a token in a cookie). If the token is valid and hasn't expired, the user is automatically signed in.</p>	"Details": Long explanation (rt/mime)

Today...

Large Language Models

Glossary of Terms

Potential Applications

LLM Integration Methods

Assessing Our Toolbox

Conclusion



Understanding the Impact

Large Language Models: Transformative or Overhyped?



Game Changer?

- A new paradigm in programming?
 - Programming with prompts...
- New ways to interact
 - Conversation - Chat or audio
 - Accessibility
- Ability to use “unusable” data
 - Extract value from documents, audio, images
 - Multilingual content
 - Cultural context, specialized knowledge...



Or, Yet Another Big Hype?

- Safety, security, privacy, compliance
 - Ethical issues
 - Bias and Fairness
- “Glorified auto-complete”?
 - Lack of creativity and critical thinking
- Indeterministic behaviour
 - “Temperature” trade-off
 - Hallucinations
- Scalability and Efficiency



Insanity Check...

- ◉ Nearly 80% of AI projects fail!
 - ◉ Double rate of other IT projects.
- ◉ Why?
 - ◉ Misunderstood problem definition
 - ◉ Complex problem
 - ◉ Data Quality and Availability
 - ◉ Technology-driven rather than solution-focused
 - ◉ Infrastructure is not sufficient



Foundations and Evolution

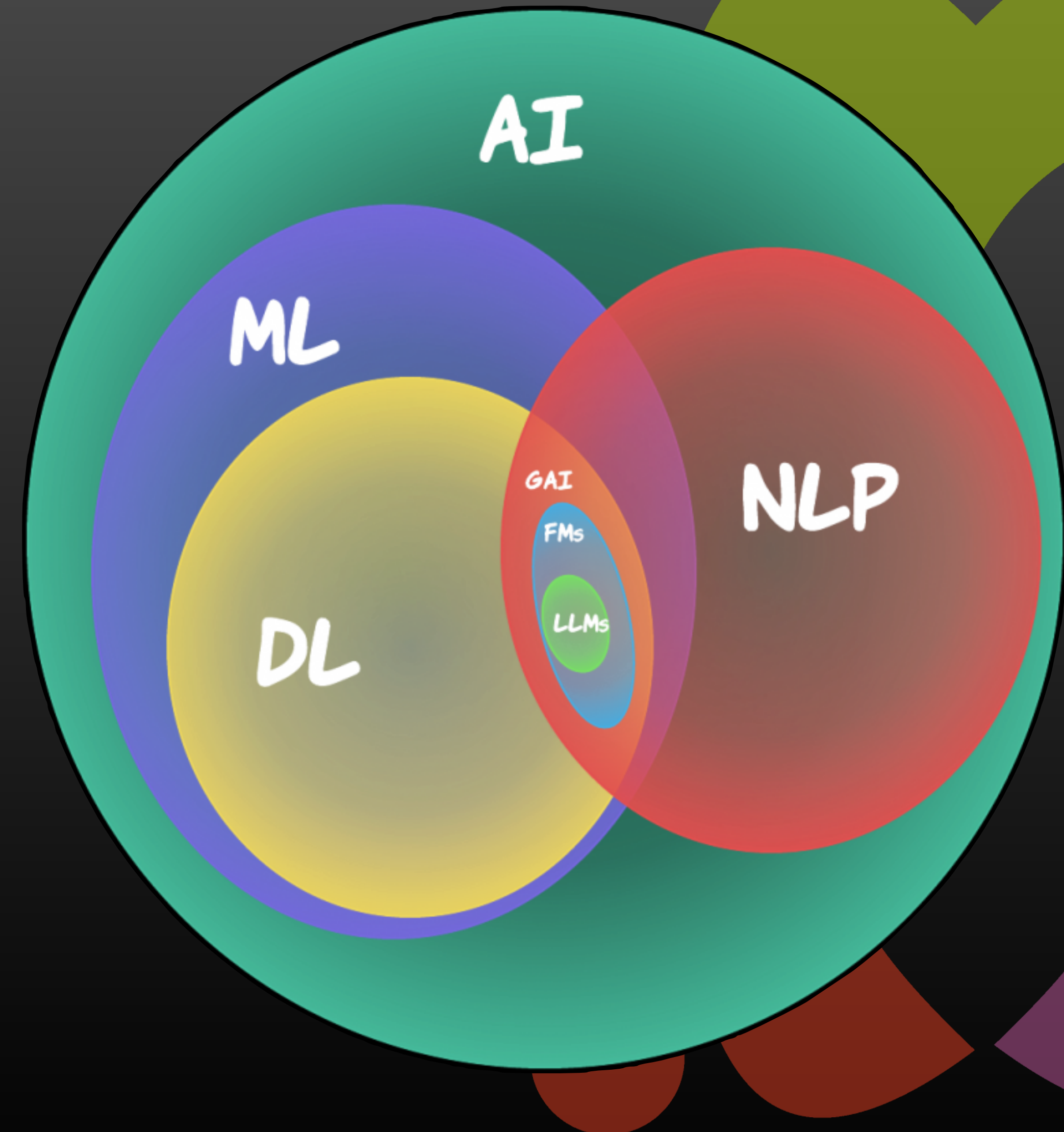
Key concepts and their progression



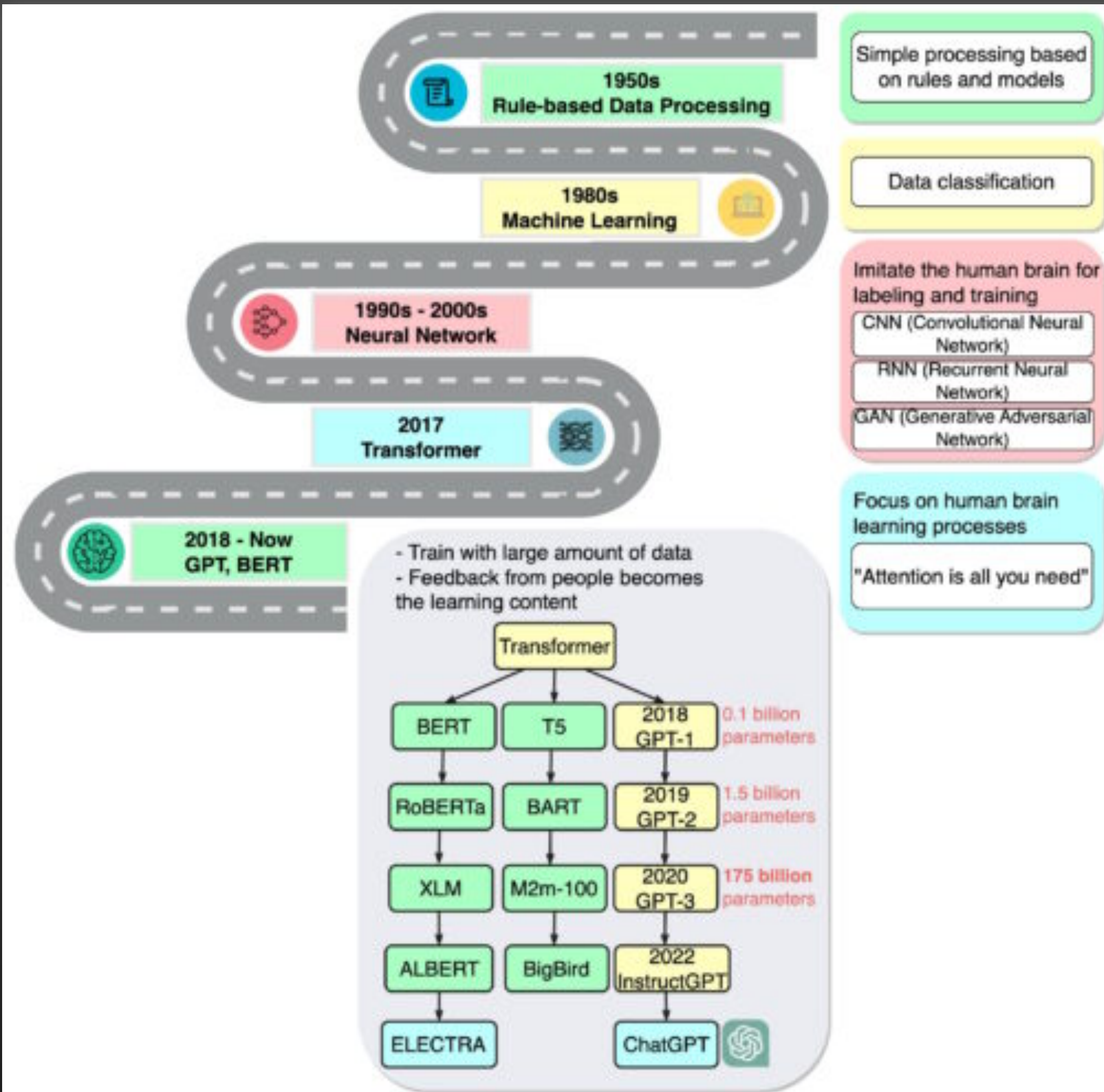
Glossary of Terms

- Artificial Intelligence
- Machine Learning
- Deep Learning
- Natural Language Processing

- Generative AI
- Foundation Models
- Large Language Models



Short History



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Lukasz Kaiser* Google Brain lukaszkaizer@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

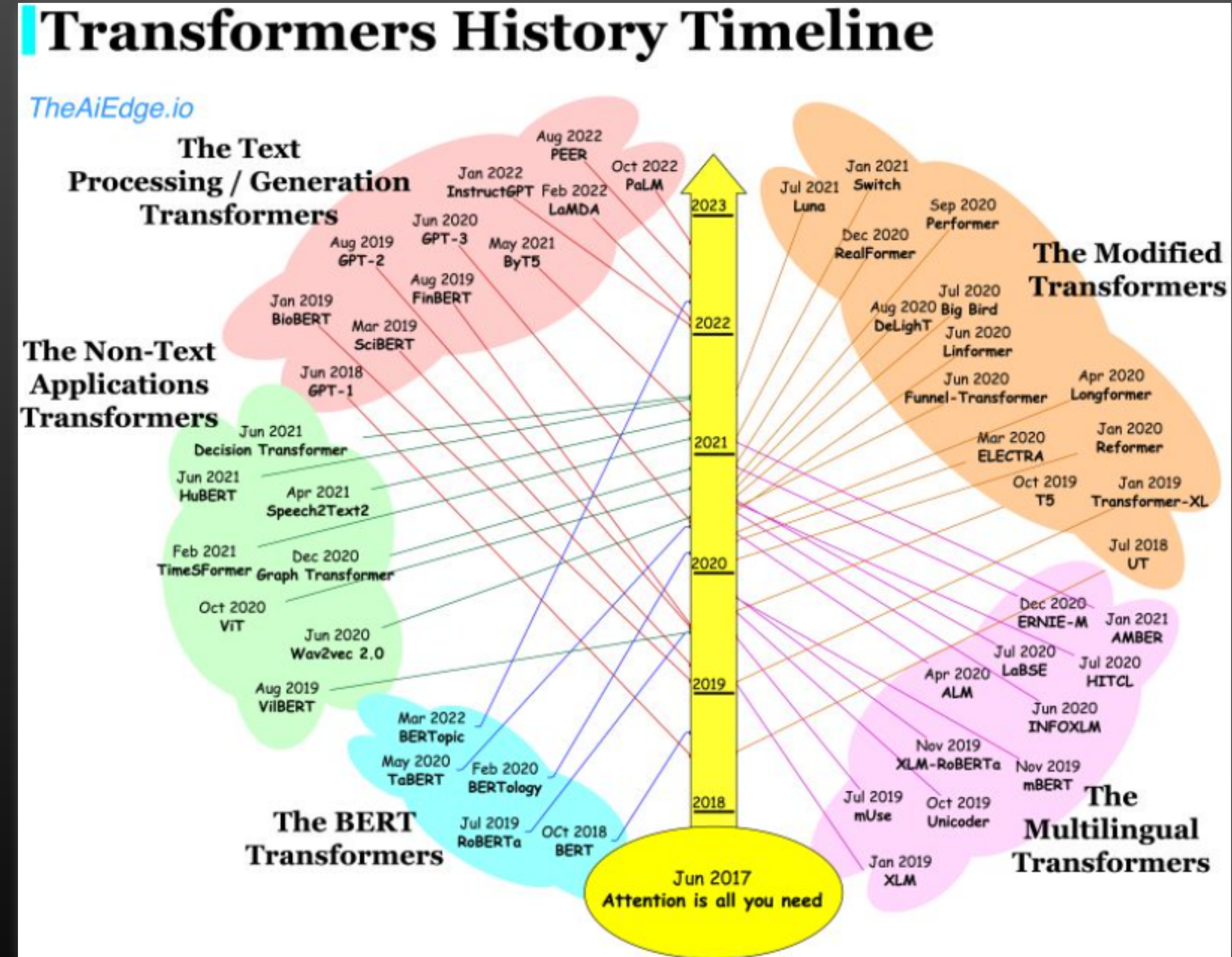
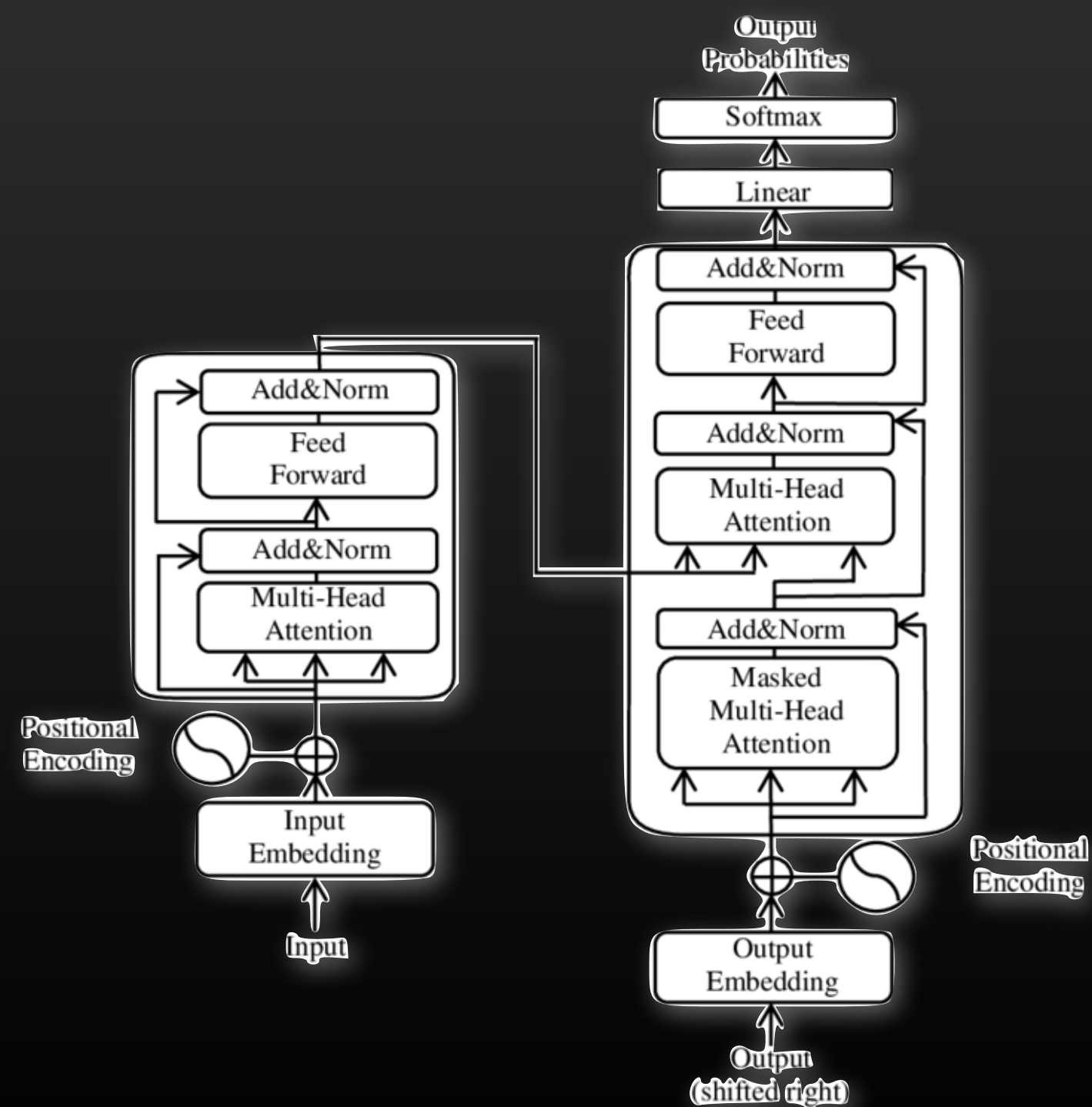
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-

Glossary of Terms

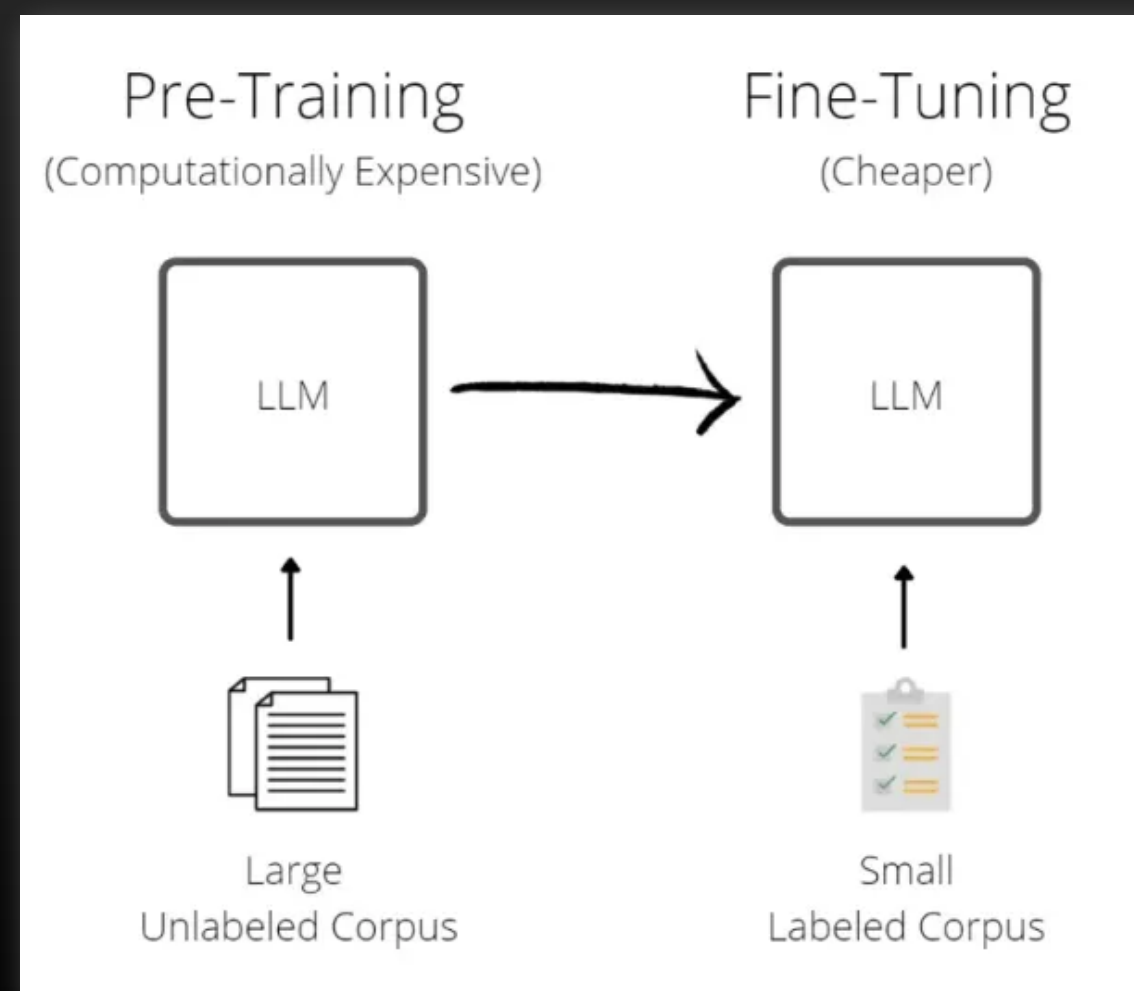
Transformers

- BERT: Bidirectional Encoder Representations from Transformers
- GPT: Generative pre-trained transformer

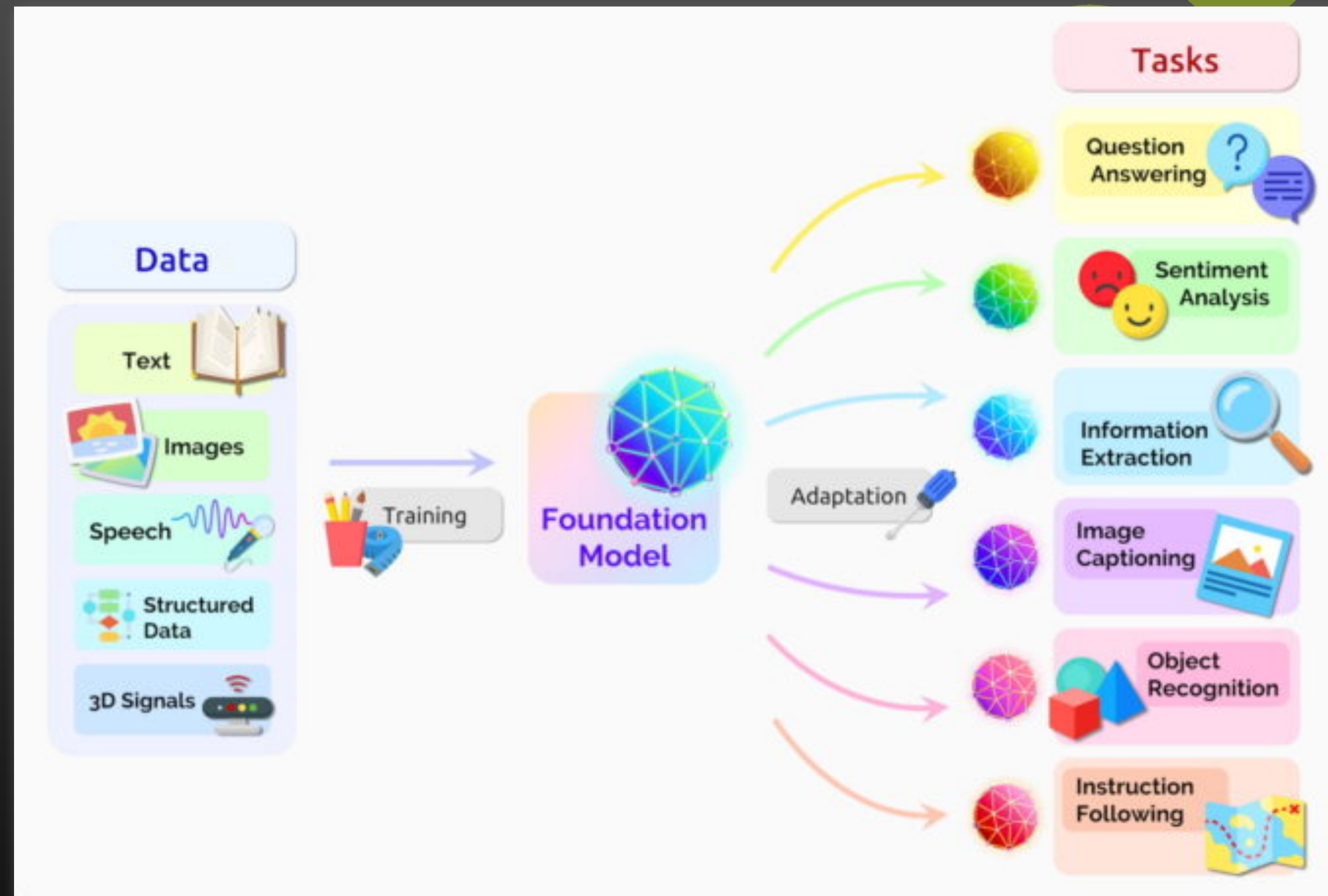


Glossary of Terms - Models

- Large Language Models
- Base / Foundation Models
- Modalities
- Tasks
- Fine tuning



<https://research.aimultiple.com/large-language-models/>



<https://blogs.nvidia.com/blog/what-are-foundation-models/>

Major Large Language Models (LLMs)

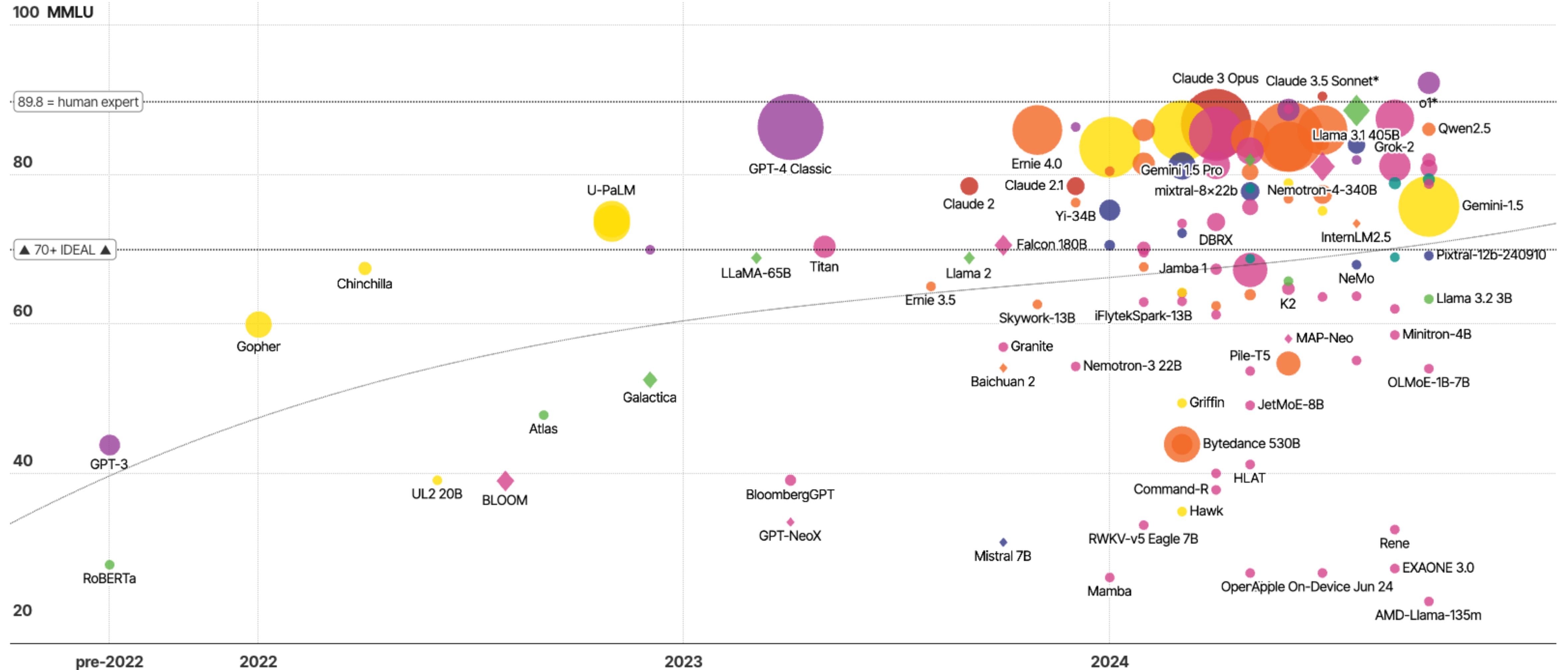
ranked by capabilities, sized by billion parameters used for training

Parameters (Bn) open access

CLICK LEGEND ITEMS TO FILTER

anthropic chinese google meta microsoft mistral openAI other

show only: all



David McCandless, Tom Evans, Paul Barton
Informationisbeautiful // Nov 2024

MMLU = benchmark for measuring LLM capabilities
* = parameters undisclosed // source: [LifeArchitect](#) // [data](#)

MADE WITH VIZsweet

What is it Good for: Large Language Model Tasks

- Text summarisation / Simplification
- Sentiment analysis
- Chatbots / Conversational AI
- Classification / Entity recognition
- Semantic Search
- Speech recognition
- Recommendation
- Text/Image/Audio/Video Generation
- Text-to-speech synthesis
- Spell/Grammar correction
- Translation
- Fraud detection
- Code generation
- AI Agents

Demo

Suggest an OpenNTF app for logging

What ???

OpenNTF (Open Network for Transformation and Freedom) is a collaborative platform primarily focused on IBM and HCL technologies, offering community-driven applications and tools. If you are looking for an app to handle logging, here are a few OpenNTF solutions that might suit your needs:

1. XPages Log File Reader

- **Purpose:** This application allows developers and administrators to read and analyse Domino server log files through an XPages interface.
- **Features:**
 - Easy access to log data from Domino servers.
 - Filter and search logs to identify issues quickly.

XPages Log File Reader

Suggest an OpenNTF app for logging

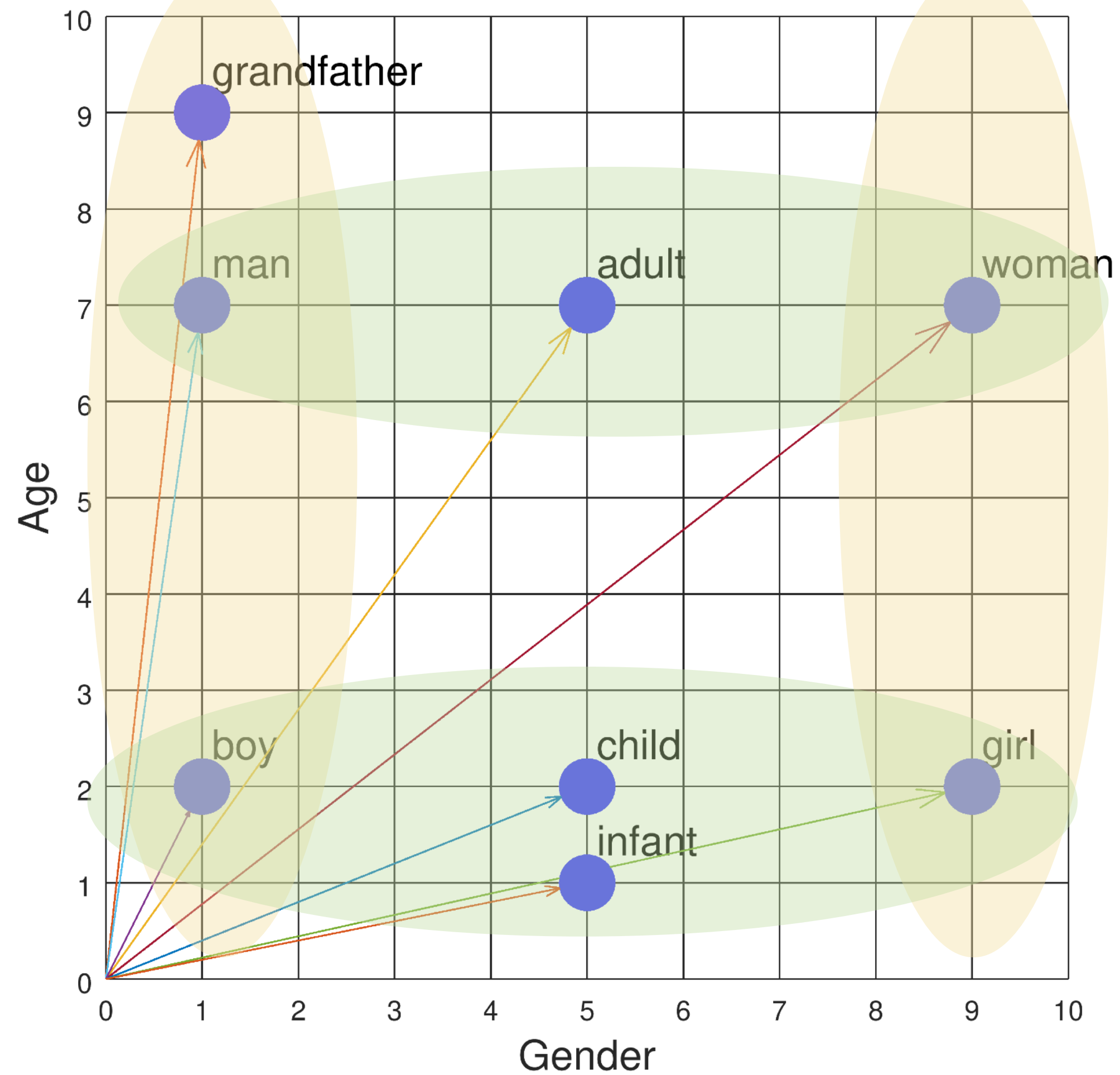
Word Embeddings

Vectors and Vector Search



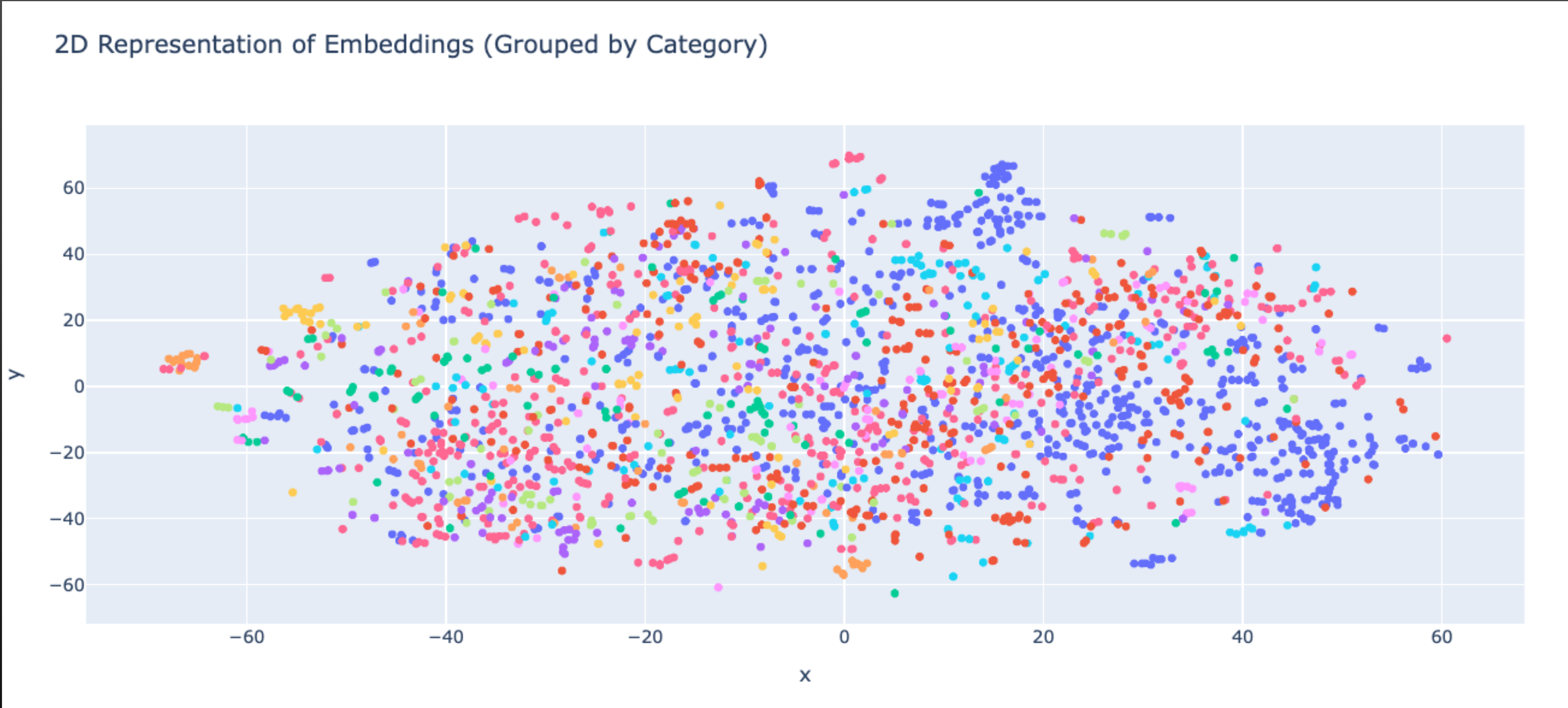
Word Embeddings

Words As Vectors



- Vector representation for words in multi-dimensional space

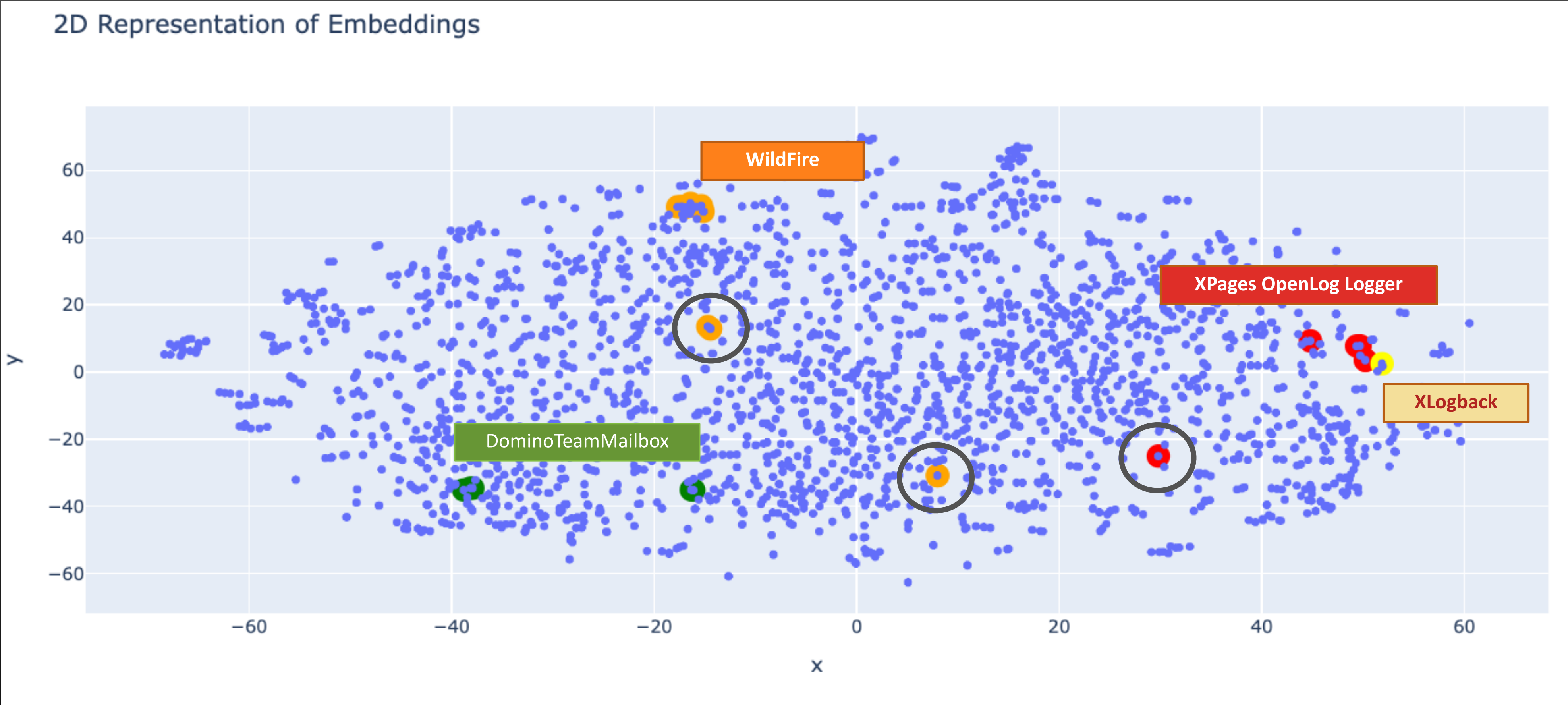
Word Embeddings - Real Life



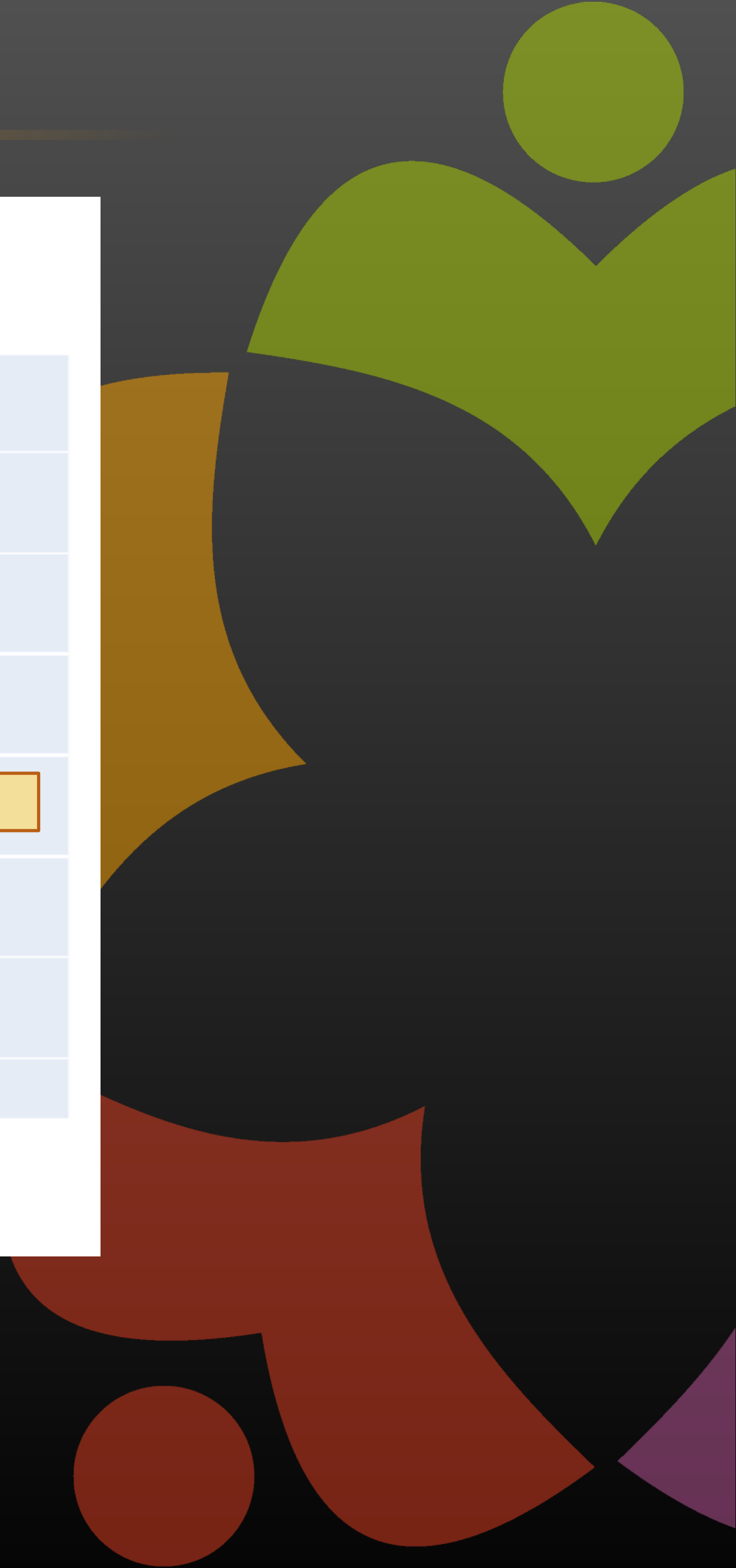
Vector space representation of project embeddings



Word Embeddings - Real Life

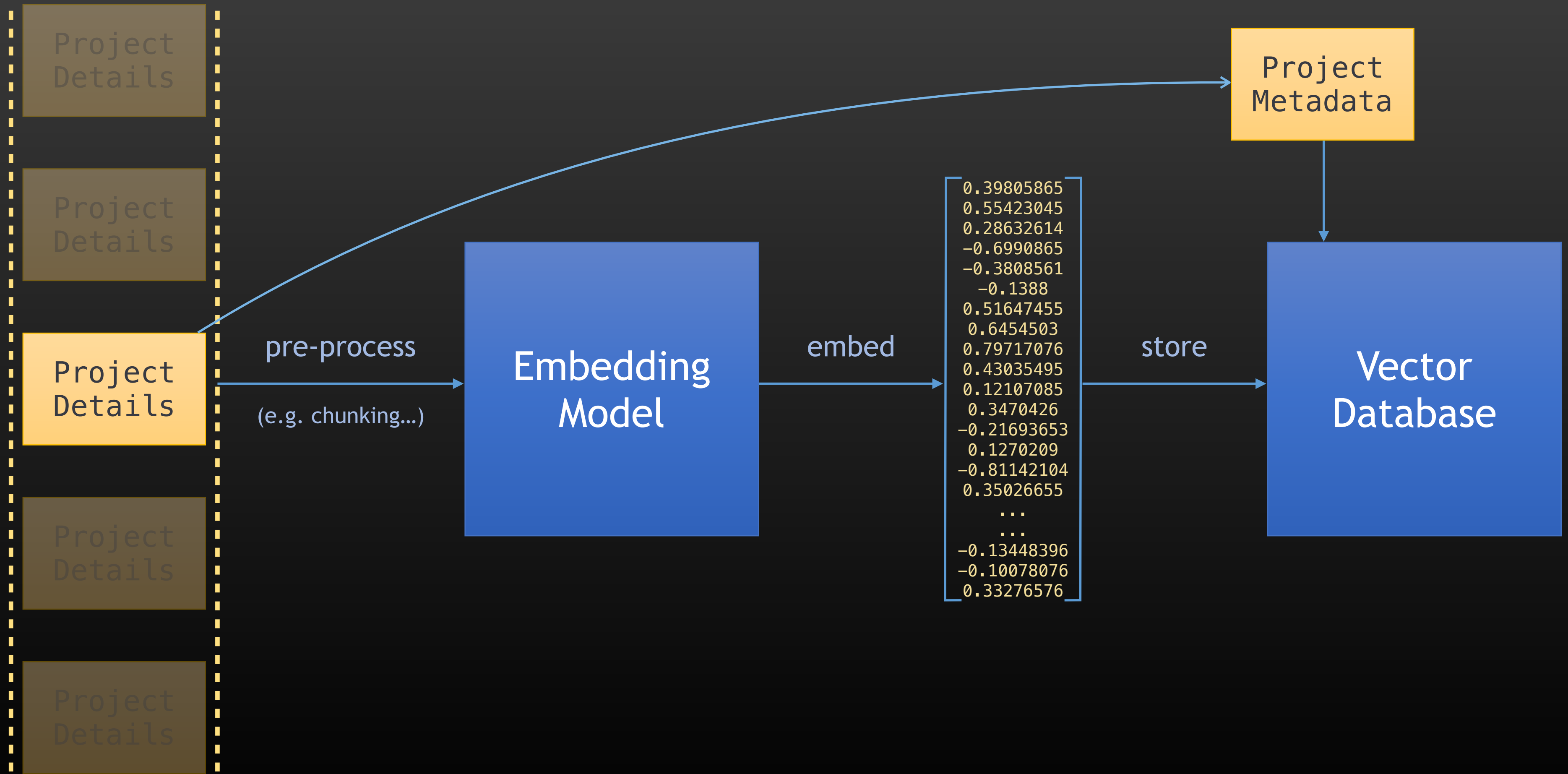


Vector space representation of project embeddings

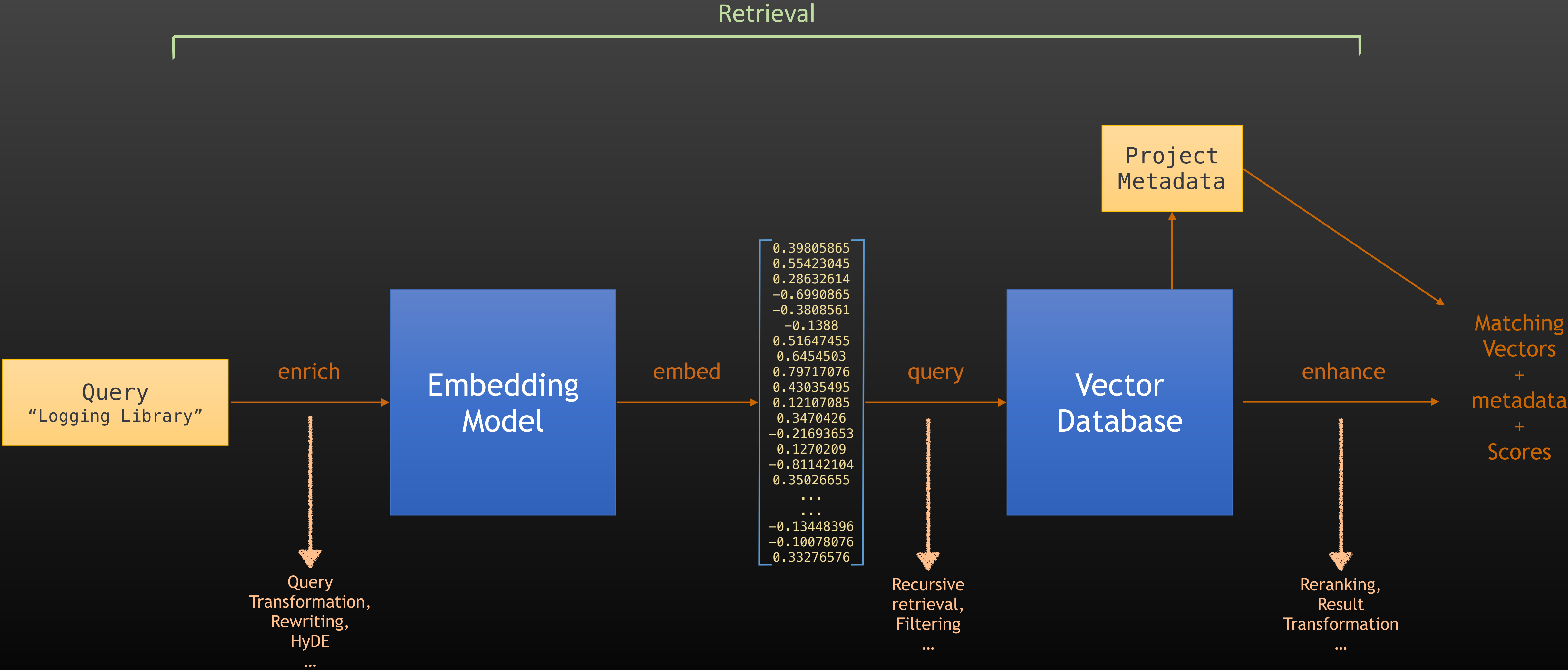


Building a Vector Store

Ingesting



Query a Vector Store



Vector Database

Pure vector databases



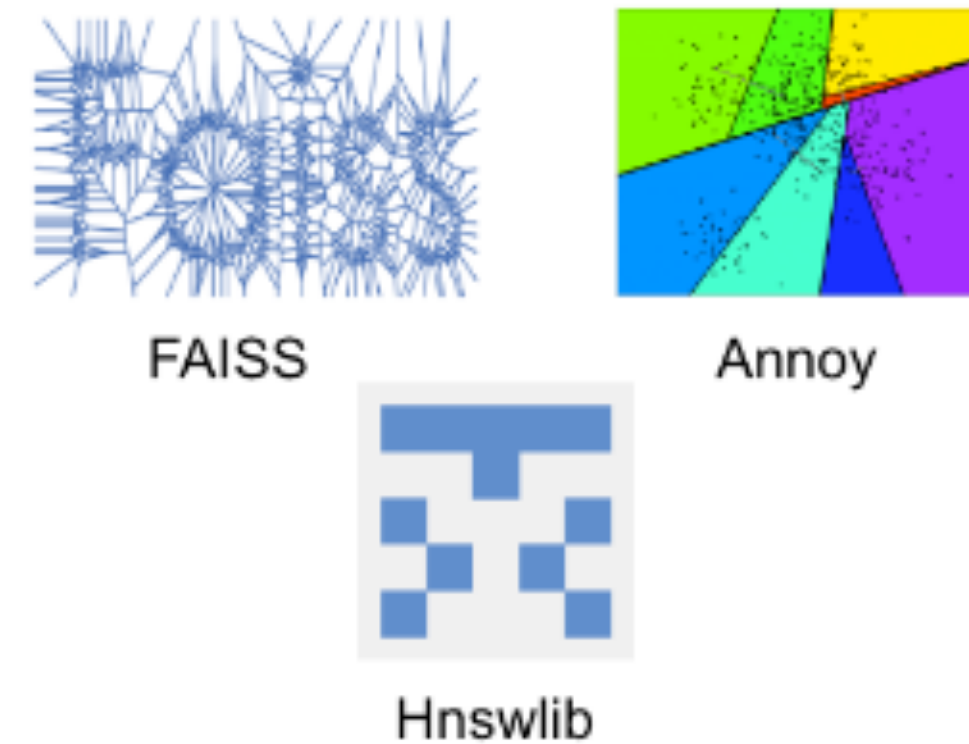
Text search databases



Vector-capable NoSQL databases



Vector libraries



Vector-capable SQL databases



Picking the Right Model

Finding the right fit for the task



Model Cards

● DATA FOCUSED

- Data Sheets
- Data Statements
- Data Nutrition Labels
- Data Cards for NLP
- Dataset Development Lifecycle Documentation Framework
- Data Cards

● MODELS & METHODS FOCUSED

- Model Cards
- Value Cards
- Method Cards
- Consumer Labels for Models

● SYSTEMS FOCUSED

- System Cards
- FactSheets
- ABOUT ML

SAMPLE OF POTENTIAL AUDIENCES

- ML Engineers
- Model Developers/Reviewers
- Students
- Policymakers
- Ethicists
- Data Scientists/Business Analysts
- Impacted Individuals

Word Embeddings - Models

Large
Language
Model

Local Models (e.g. Ollama, Onnx files...)

- ✓ Your data won't leave the server
- ✓ Most are free with permissive licenses
- ✓ No vendor lock-in
- ✓ No cost per operation
- ✗ Model files are huge.
- ✗ LLM tasks are resource-intensive
- ✗ Less capable models
- ✗ Programmability restrictions



Word Embeddings - Models

Large
Language
Model

Cloud Models (e.g. OpenAI, Vertex AI, etc.)

- ✓ Managed services
- ✓ Pay-per-use model
- ✓ Easy to use - RESTful API and native SDKs
- ✓ Scalable / Available
- ✓ High performance / High quality
- ✓ Much better in complicated tasks
- ✗ Privacy and security concerns
- ✗ Network latency
- ✗ High costs for very busy systems
- ✗ Vendor lock-in



Decide and Test the Model

```
from qdrant_client import QdrantClient
from qdrant_client.models import Distance, VectorParams

client = QdrantClient(url="http://localhost:6333")

def newCollection(collection_name, model, items):
    points_data = []

    for item in items:
        points_data.append({
            "id": points_data.__len__() + 1,
            "vector": model.encode(f"{item['name']}: {item['overview']} {item['content']}"),
            "payload": {
                "unid": item["unid"],
                "name": item["name"],
                "lastUpdated": item["lastUpdated"]
            }
        })

    client.delete_collection(collection_name)

    client.create_collection(
        collection_name=collection_name,
        vectors_config=VectorParams(size=model.get_sentence_embedding_dimension(), distance=Distance.COSINE)
    )

    client.upsert(collection_name=collection_name, points=points_data)

newCollection("coll_mxbai", model_mxbai, data)
newCollection("coll_msmarco", model_msmarco, data)
```

```
from qdrant_client import models

def searchCollection(collection_name, model, search_term):
    result = client.search(
        collection_name=collection_name,
        search_params=models.SearchParams(hnsw_ef=128, exact=False),
        query_vector=model.encode(search_term),
        limit=10,
    )

    print(f"\nModel {collection_name} search for '{search_term}':")
    print(30*"~")

    # Incoming result:
    # ScoredPoint(id=412, version=0, score=0.73846227, payload={ [name/unid/lastupdated] }, vec

    for hit in result:
        print(f"{hit.payload['name']}: {hit.score:.3f}")

    print(30*"~")

    return result

def test_models(search_term):
    searchCollection("coll_mxbai", model_mxbai, search_term)
    searchCollection("coll_msmarco", model_msmarco, search_term)

test_models("logging library")
```

Suggestion: Learn Python!

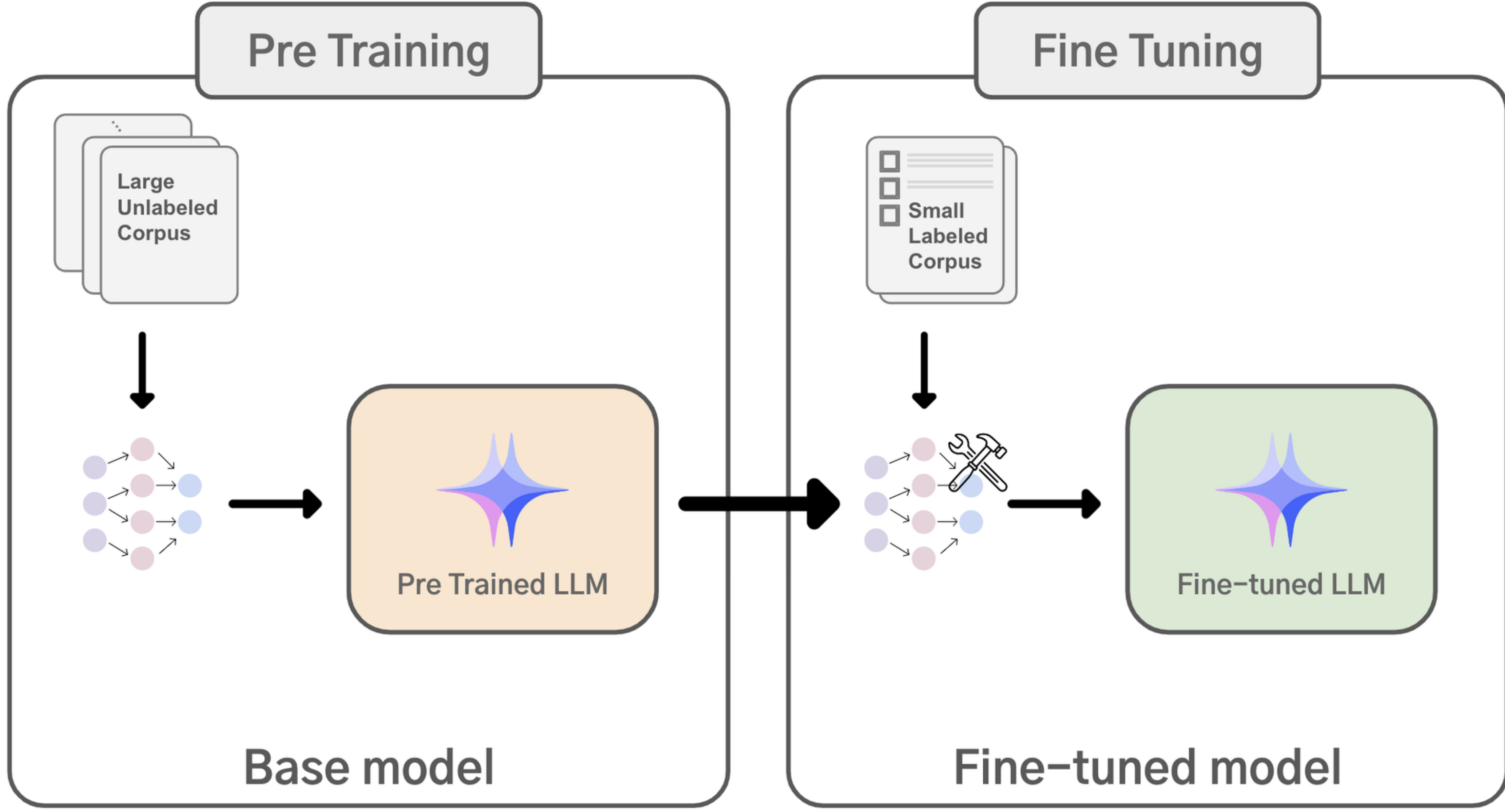
```
Model coll_mxbai search for 'logging library':
-----
xLog - logger framework: 0.738
Debug Logger: 0.710
Simple Log - Logging Lotusscript simply and flexibly: 0.668
Flow: 0.662
XPages OpenLog Logger: 0.640
Enhanced Log: 0.639
XLogback: 0.614
DOTS Chatstore: 0.609
Sametime Chat Logging API: 0.608
Advanced Domino-Web-Server-Log Template: 0.592
-----
```

Improving Models

Tweaking the Brain

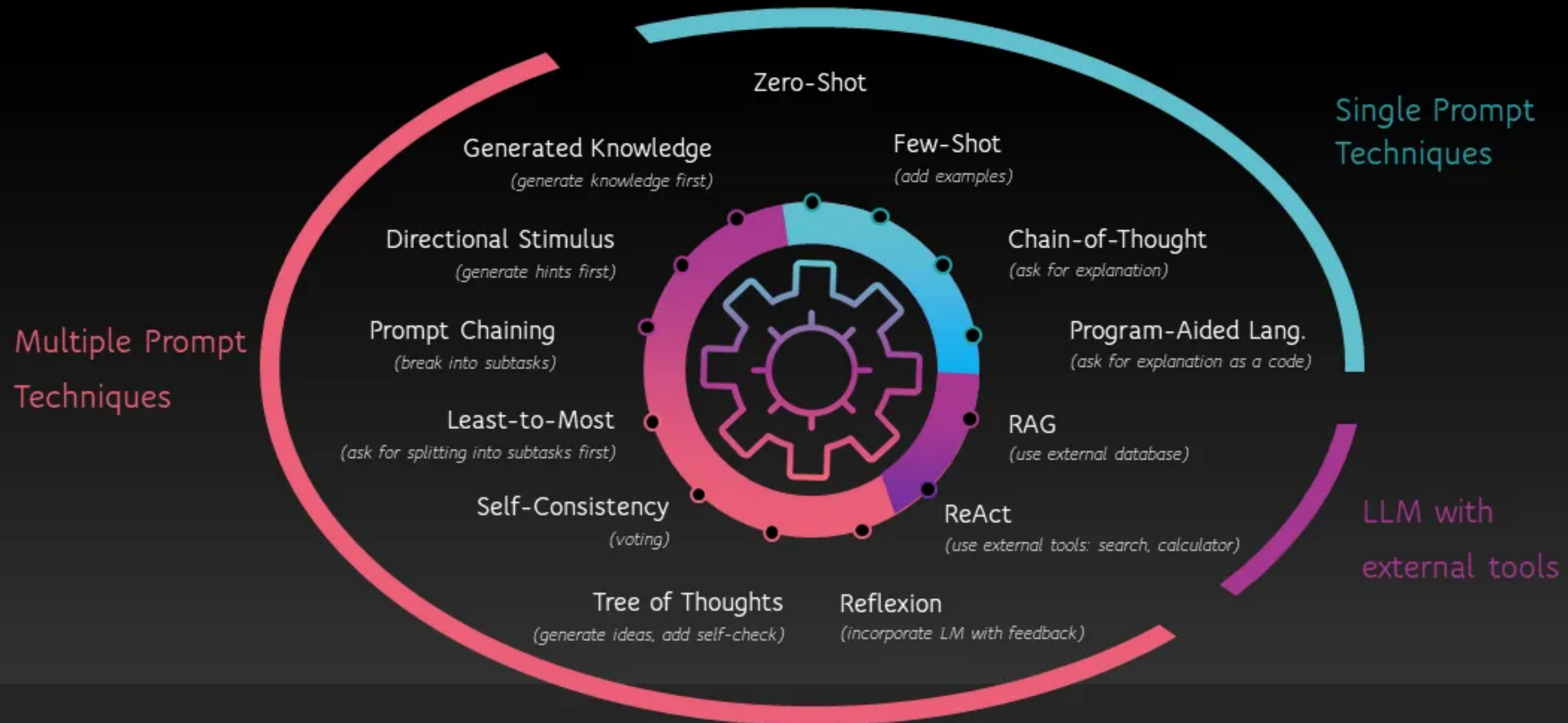


Increase Knowledge: Fine Tune (Transfer Learning)



Improve Behavior: Prompt Engineering

Prompt Engineering Techniques

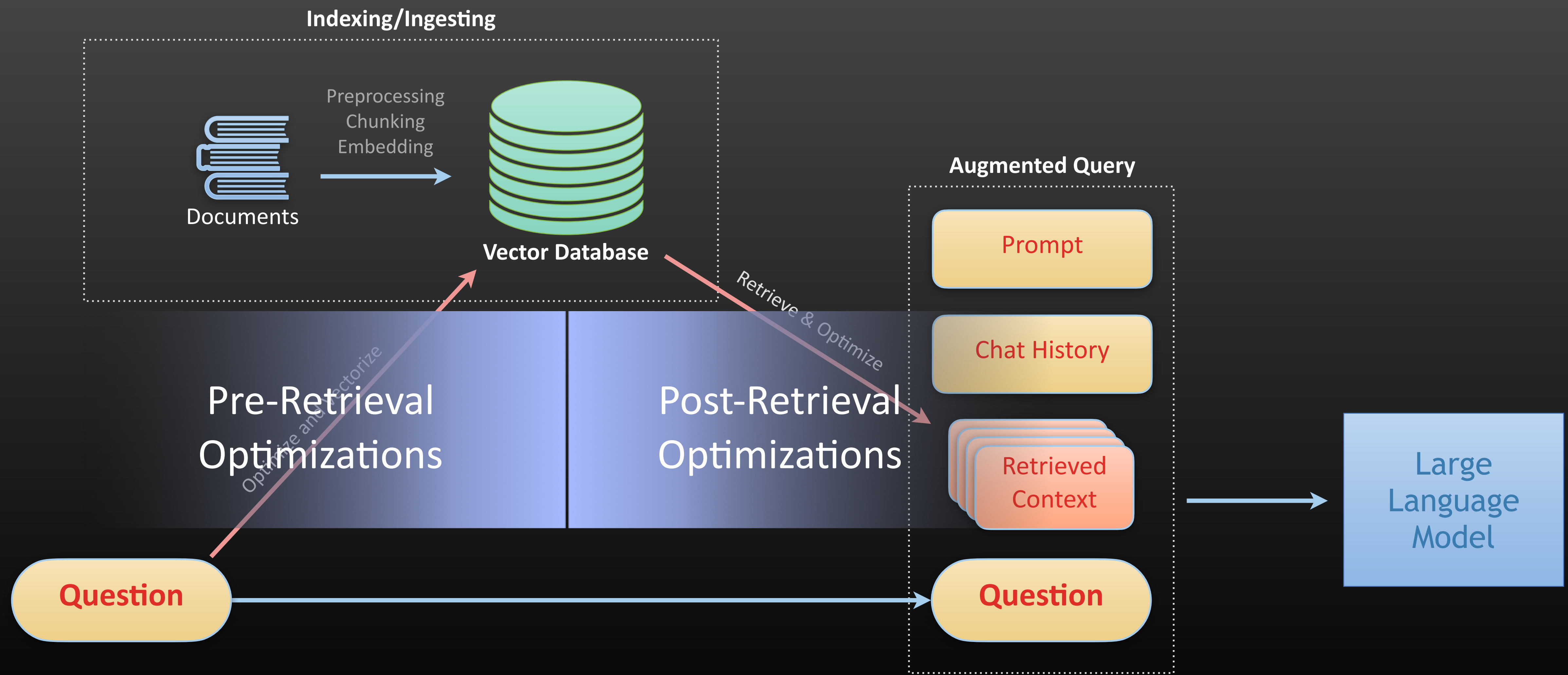


Improve Prompts: Retrieval-augmented generation

- ◉ Scenario

- Domain Knowledge in documents, databases, etc.
- LLM to respond questions aligned with domain knowledge

Improve Prompts: Retrieval-augmented generation



Demo

Prompts and Chat



Working with LLMs for Domino Apps

LLM Integration is a simple REST API integration



Access LLMs using Java in Domino

- XPages
- OSGi Plugins
- RESTful API (OpenNTF JakartaEE project by Jesse)
- Java Agents (Notes Client or Server side)
- DOTS
- Java Addin



.xsp



For Java Developers

Current LangChain4j Integrations



LLM Integrations	
Amazon Bedrock	Google Vertex AI PaLM 2
Azure OpenAI	HuggingFace
ChatGLM	LocalAI
DashScope	Ollama
Google Vertex AI Gemini	OpenAI
	Mistral

Image Model Integrations
Azure OpenAI
OpenAI DALL-E
Vertex AI Gemini
Ollama
Qwen

Current LangChain4j Integrations

Embedding Stores	
Chroma	Astra DB
Elasticsearch	Cassandra
Milvus	Neo4j
Pinecone	OpenSearch
Vespa	PGVector
Weaviate	MongoDB
Redis	Qdrant

Code Execution Engines
GraalVM Polyglot/Truffle
Judge0

Document Loaders	
txt	ppt
html	url
doc	github loader
pdf	S3
xls	Azure blob loader
Tencent COS	

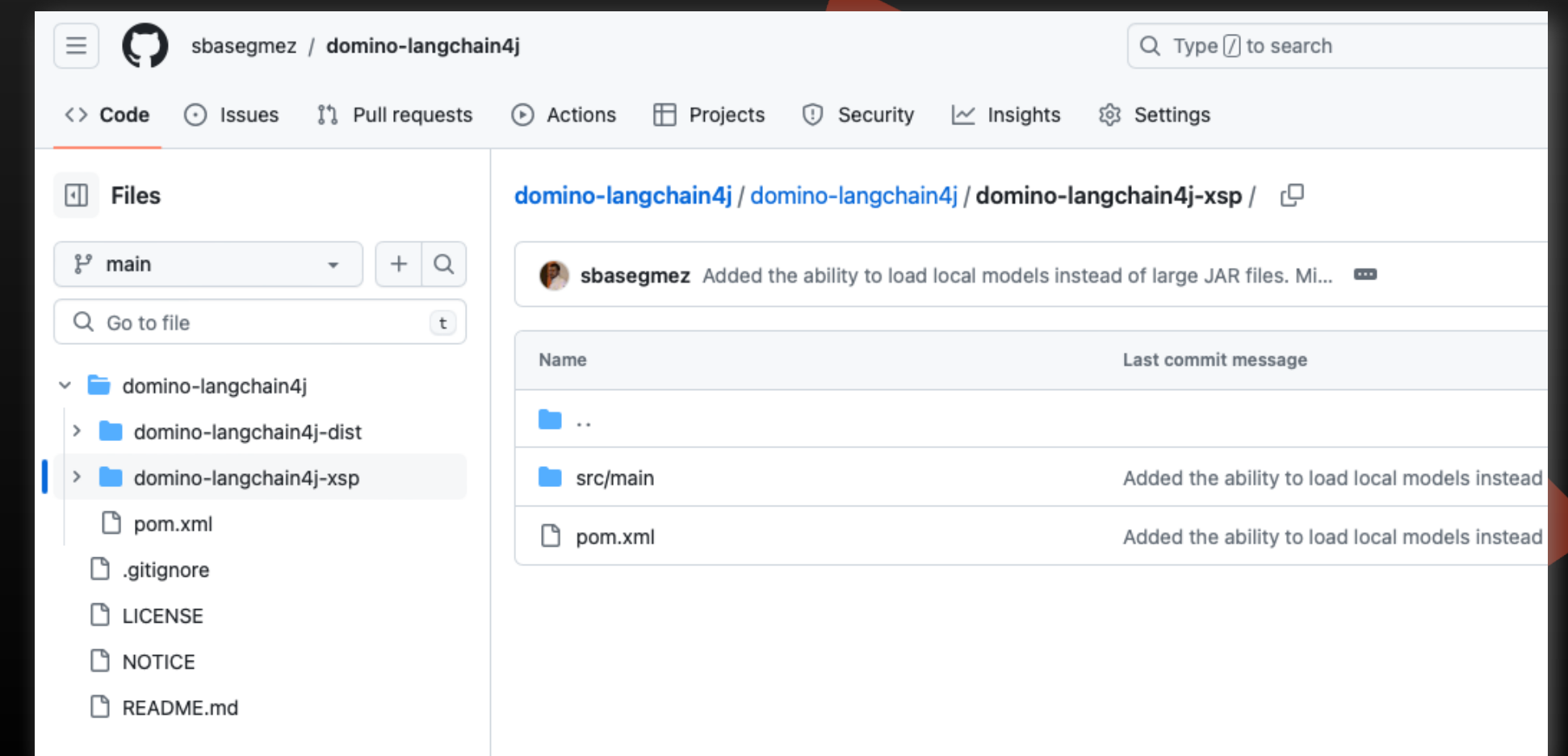
Frameworks
Quarkus
Spring Boot



LangChain4j is very promising

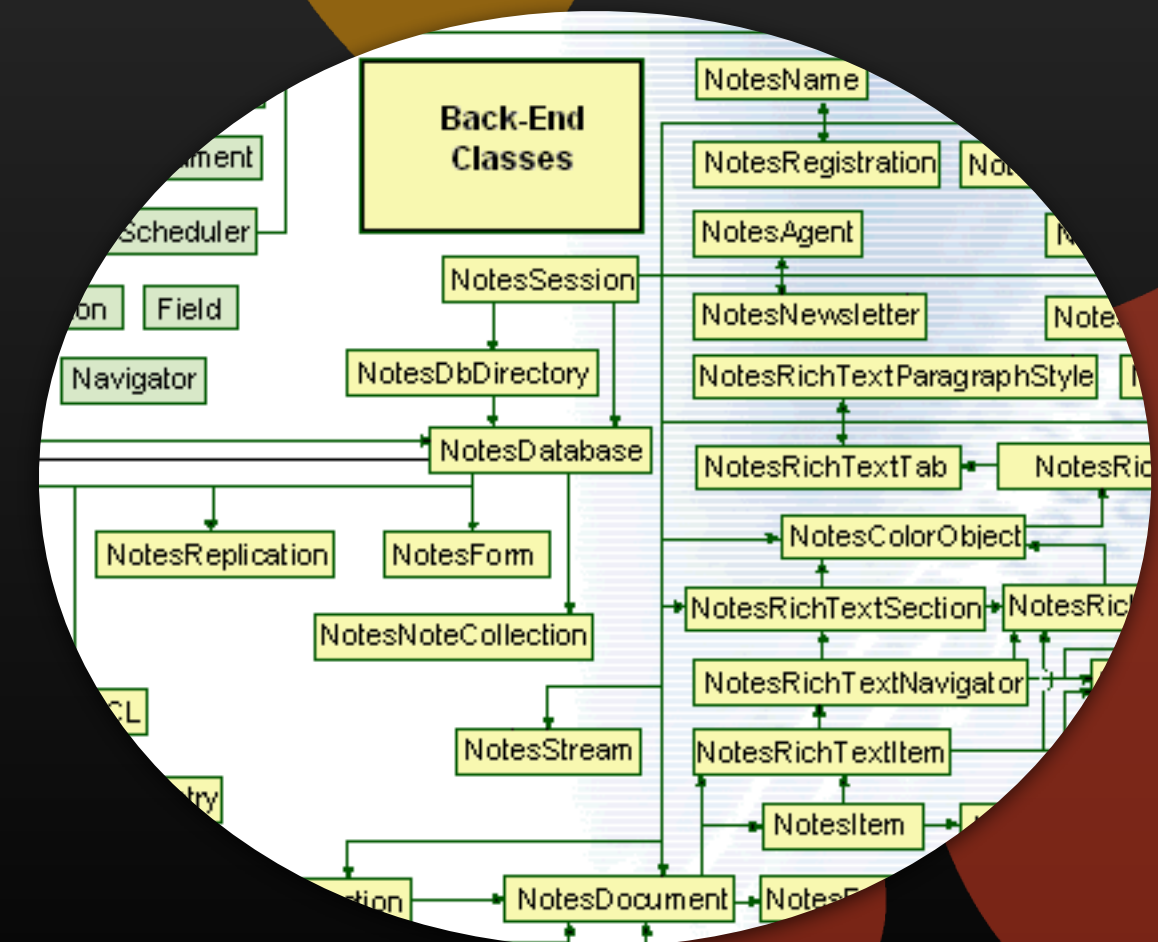
A New Project: Domino-LangChain4j

- Experimental phase
- Import langchain4j library into Domino
 - Utilise ChatModel w/ Local or Cloud LLM
 - Embedding
 - RAG
- Server and Designer plugins
- Add some utilities
 - Local Model Support
 - Managed beans
 - Configuration / Logging
 - RAG document loaders for Domino
- Looking into Java Agent and DOTS support
- Feedbacks are welcome!



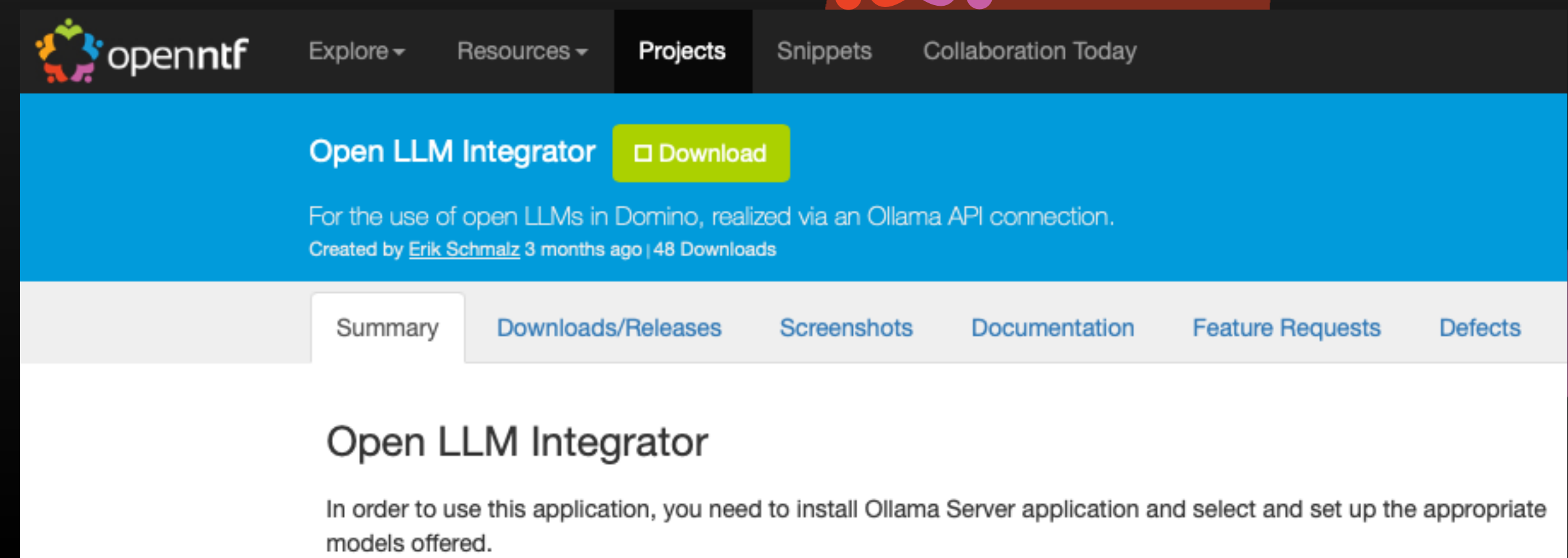
Access LLMs using LotusScript in Domino

- For LotusScript, there are still options.
 - Use RESTful access using LotusScript
- Use Java Agent
 - LLM integration might be done with Java agents.
LotusScript can call agents



Other LLM Projects

- HCL Domino IQ (Future Product)
 - Uses Llama.cpp
 - Integrated to the server
- Open LLM Integrator on OpenNTF
 - Ollama integration with RAG and QDrant support
 - By Erik Schmalz
- ChatGPT APIs for Domino on OpenNTF
 - Credits: Ayhan Sahin & Christian Sadeghi



Integration Outside of the Domino Server

- Implement LLM logic in your favorite platform
 - Volt MX
 - Python
 - Java
 - JavaScript
 - ...
- Access to Domino Data
 - Using Domino REST API
 - Implement your own services with the OpenNTF Jakarta EE Project



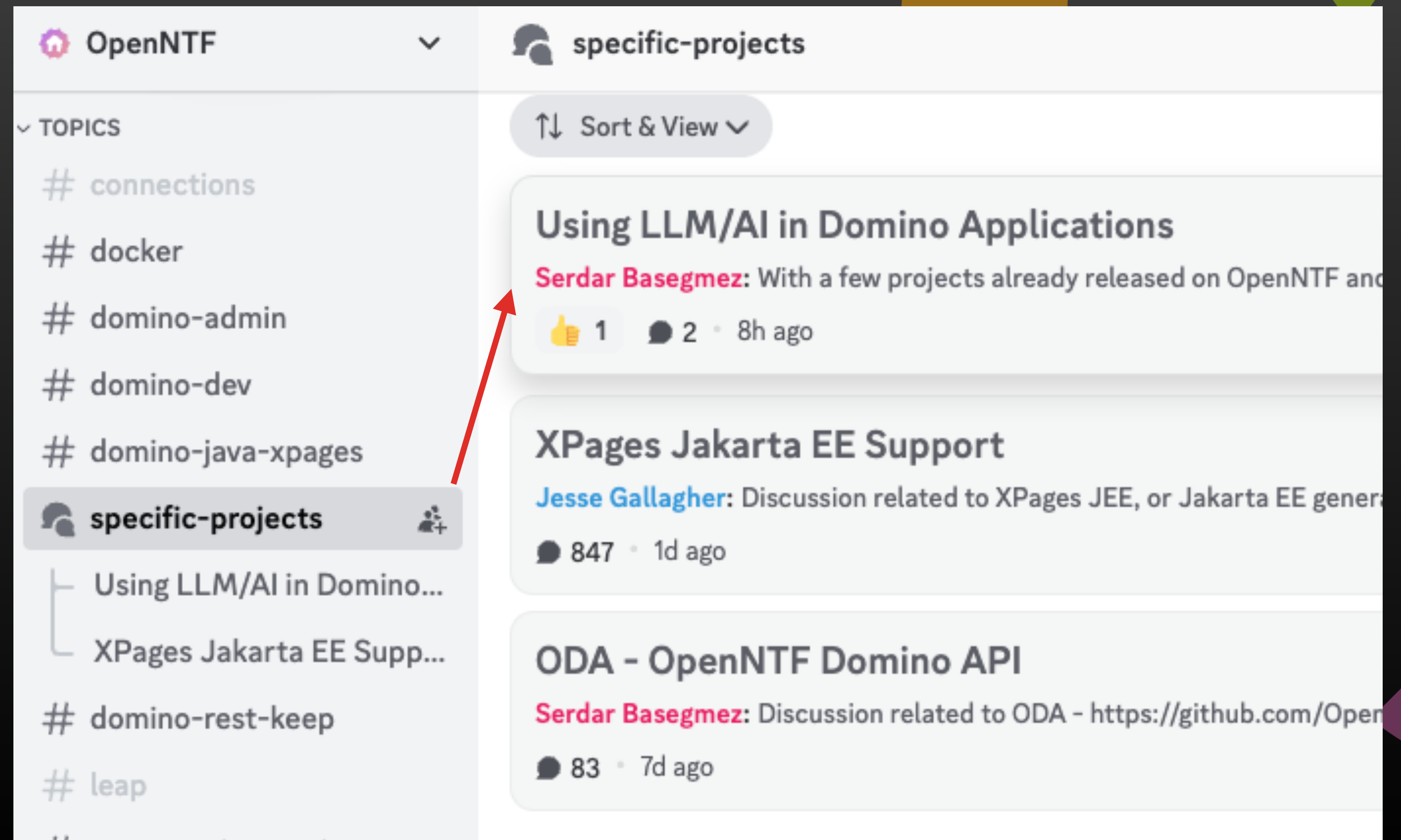
Topics for Another Day...

- Models deep-dive
- Prompt Engineering
- Development Methodology
 - Prototyping, validation, optimization, testing, lifecycle
- Safety and Security
 - Guardrails, moderation
 - Prompt Injections
 - Regular Compliance Audits
 - AI Accountability



Feedbacks and Discussions

- OpenNTF Discord Server
 - Specific Projects —> Using LLM/AI in Domino Applications



The screenshot shows the OpenNTF Discord server interface. On the left, a sidebar lists various channels under the heading 'TOPICS'. The 'specific-projects' channel is selected and highlighted. A red arrow points from this channel in the sidebar to a message in the main chat area. The message is titled 'Using LLM/AI in Domino Applications' and is from user 'Serdar Basegmez'. Below the message, it shows 1 thumbs up and 2 replies, posted 8 hours ago. Other visible messages include 'XPages Jakarta EE Support' by 'Jesse Gallagher' (847 replies, 1d ago) and 'ODA - OpenNTF Domino API' by 'Serdar Basegmez' (83 replies, 7d ago).

Resources

➔ All the demo materials:

- <https://github.com/sbasegmez/LLM-Demos>

➔ OpenNTF Projects Metadata:

- TBA

➔ Domino-Langchain4j experimental version:

- <https://github.com/sbasegmez/domino-langchain4j>



More Good Stuff: Odds and Ends

- Further reading...

- Huggingface blogs
- RAG - Retrieval Augmented Generation
- Multimodal approaches
- Prompt Engineering

- Courses, guides

- Quick Start Guide to Large Language Models (LLMs)
Course by Sinan Ozdemir
- Large Language Models: Application through Production
Databricks
- Large Language Model Ebooks
NVidia

