

# KGs and LLMs

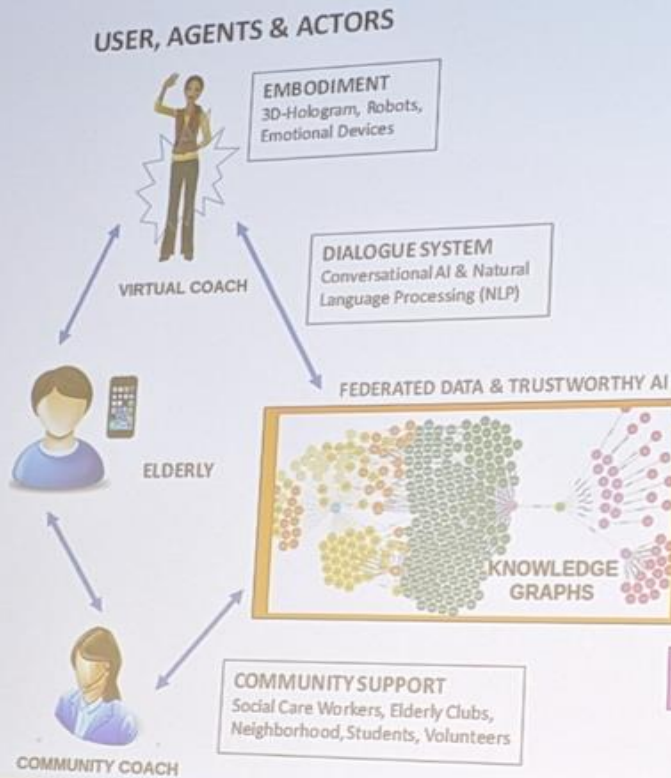
How do they complement each other ?

**Why Knowledge Graphs are the Future of AI Systems ?**

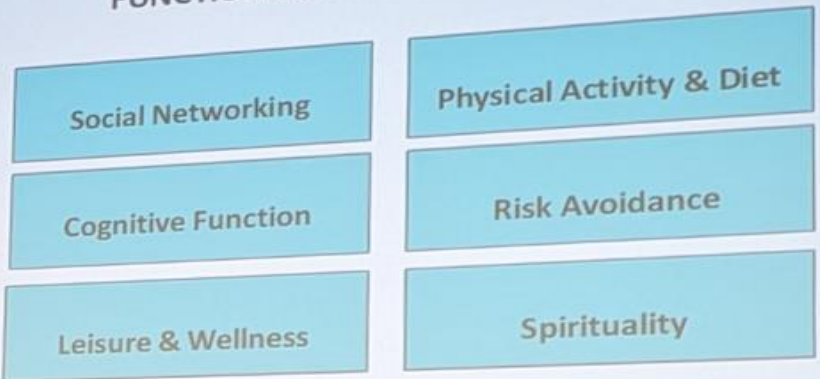
G rard Chollet, Haoyi Xiong, Graham Wilcock,  
Eric de la Clergerie, Kristiina Jokinen, Anthony Alcaraz,  
Christian Dugast, Hugues Sansen,  
Michael McTear, Maria In s Torres, Hermann Ney

# System Overview

## Socio-Informatics System



### FUNCTIONALITY (AREAS OF SUPPORT)



# Some of the recent blogs of Anthony Alcaraz

- KGLM-Loop: A Bi-Directional Data Flywheel for Knowledge Graph Refinement and Hallucination Detection in Large Language Models
- Logical Retrieval with KGs: The Key to Contextual and Intelligent AI
- Why Large Language Models Alone Are Not Enough
- Leveraging Structured Knowledge to Automatically Detect Hallucination in Large Language Models
- **Enriching Language Models with Knowledge Graphs for Powerful Question Answering**
- **Unlocking Whole Dataset Reasoning — Why Knowledge Graphs are the Future of AI Systems**
- **Integrating Large Language Models and Knowledge Graphs: A Neuro-Symbolic Perspective**
- **Embeddings + KGs: The Ultimate Tools for RAG Systems**

# KGs & LLMs : a State of the Art

- At least 26 papers published since January 1st, 2024
- More than 178 papers published last year
- 63 papers in 2022, 25 in 2021, 21 in 2020, 10 in 2019,...

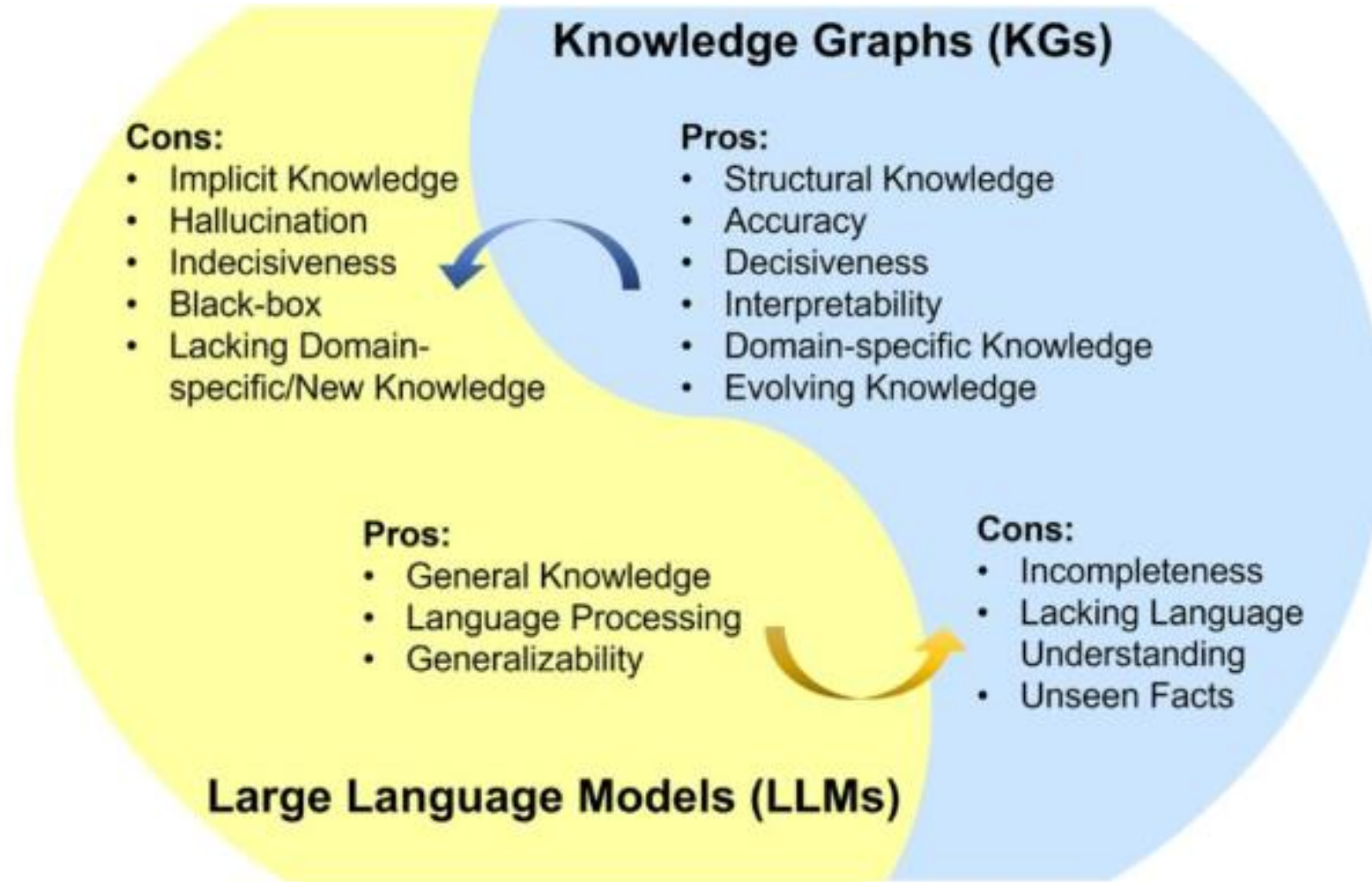
## Who is publishing ?

- Mostly academics from China, US, Australia, Singapore, Germany, UK, Switzerland, Brazil, Canada, France,...
- Some companies : Meta, Tencent, Baidu,...



# Knowledge Graphs and LLMs

---



• <https://arxiv.org/pdf/2306.08302.pdf>

• <https://www.youtube.com/watch?v=1RZ5ylyz31c>

• Unifying Large Language Models and Knowledge Graphs: A Roadmap

# What is a Large Language Model (LLM) ?

- A **language model** is a probabilistic model of a natural language

## **Weaknesses of Large Language Models**

**Hallucination**

**Black-box Nature**

**Indecisiveness**

**Implicit Knowledge**

**Lacking Domain-Specific/New Knowledge**

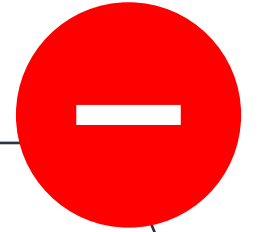
<https://www.linkedin.com/pulse/combining-large-language-models-knowledge-graphs-wisecube/>

# LLMs: Pros and Cons



## PROS

- ✓ Based on data
- ✓ Automatic
- ✓ Task/Domain independent

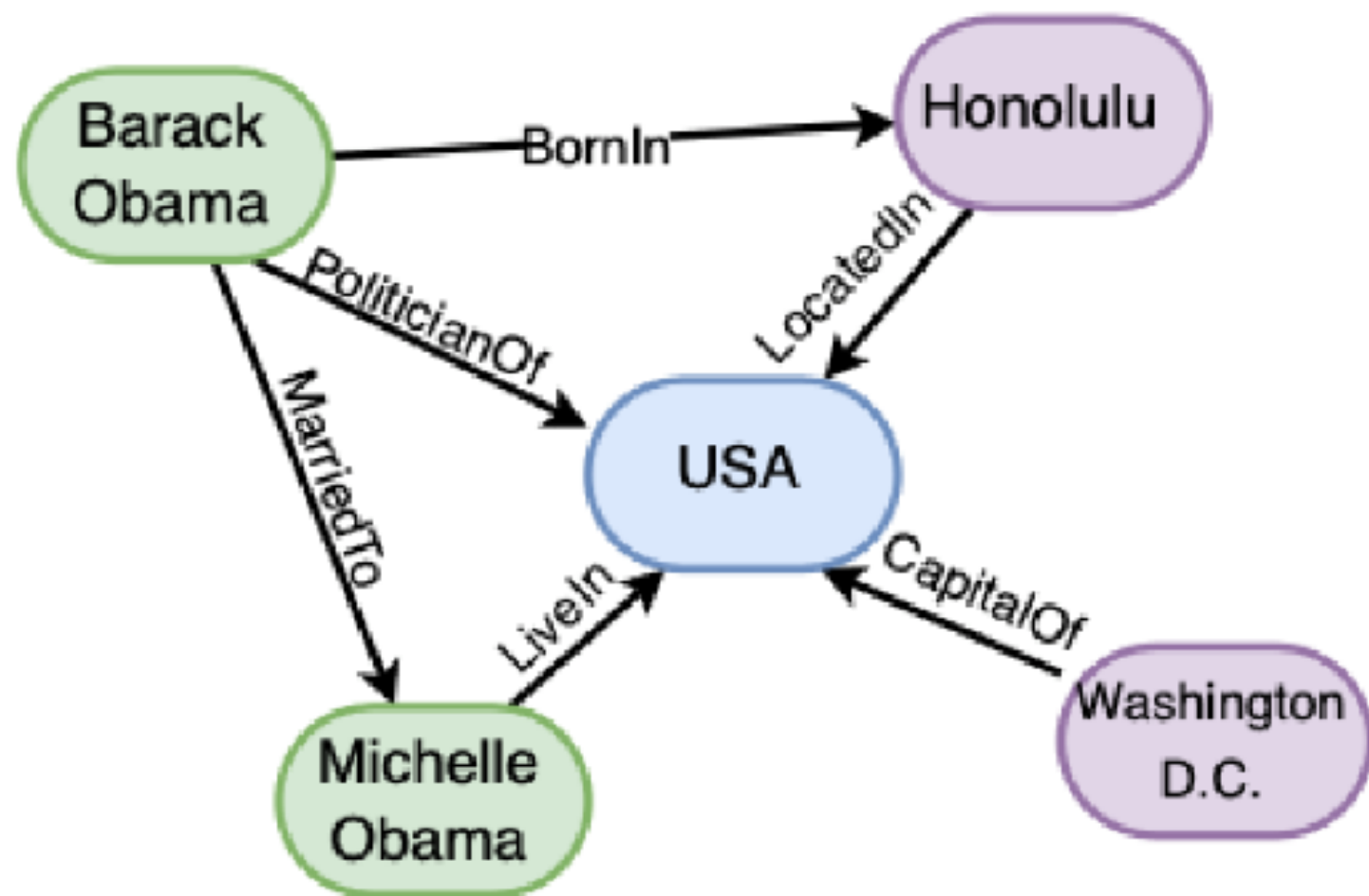


## CONS

- ✗ Hallucinates
- ✗ One answer per perspective
- ✗ No abstraction: No reasoning structure
- ✗ Always has an answer

# What is a Knowledge Graph?

- Triplets:
  - {Source, Destination, Relation}
- Typically a Directed Graph



# Strengths of Knowledge Graphs

- **Structural Knowledge Representation**
- **Decisiveness**
- **Interpretability and Explainability**
- **Accuracy and Consistency**
- **Domain-Specific Knowledge Capture**
- **Evolving Knowledge**

# Weaknesses of Knowledge Graphs

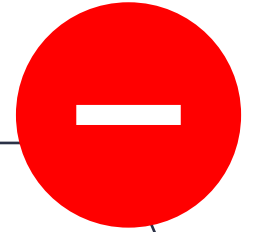
- **Incompleteness**
- **Unseen Facts and Updates**
- **Lacking Language Understanding**

# KGs: Pros and Cons



## PROS

- ✓ Manual KGs are factual
- ✓ Contains explicit alternatives / complementarity / inconsistencies
- ✓ Allows reasoning
- ✓ Does not always have an answer



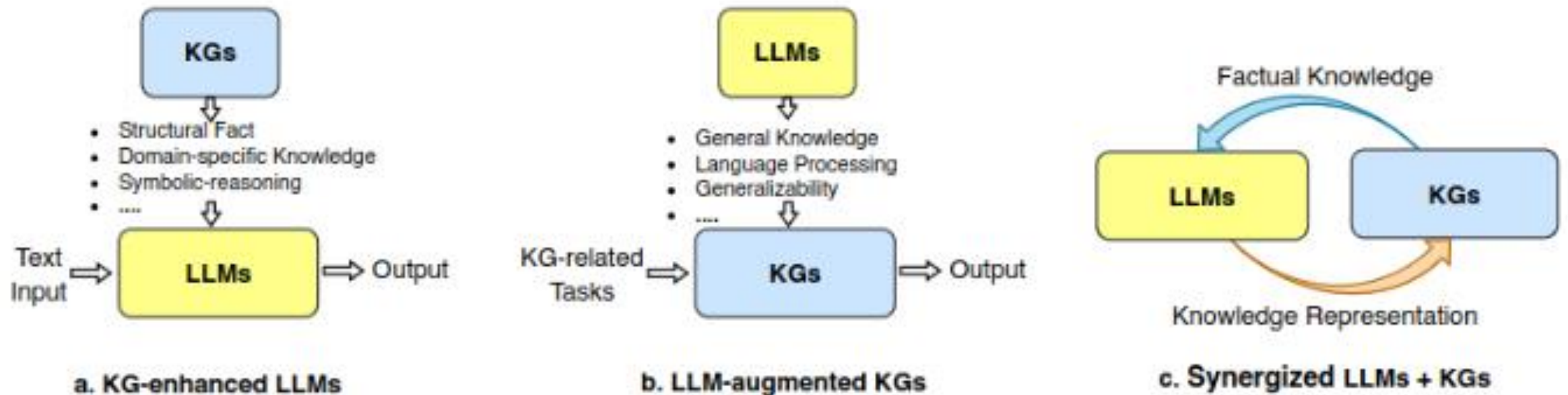
## CONS

- ✗ Relations are based on hard-coded ontologies
- ✗ Intensive manual work for high quality
- ✗ To be efficient, KG expansion is task dependent
- ✗ Precision impacts flexibility



# Unifying Large Language Models & Knowledge Graphs

## Large Language Model-Augmented Knowledge Graphs



<https://arxiv.org/pdf/2306.08302.pdf>

<https://www.youtube.com/watch?v=1RZ5ylyz31c>

Unifying Large Language Models and Knowledge Graphs: A Roadmap

# The challenge: Automatic creation of KGs using LLMs

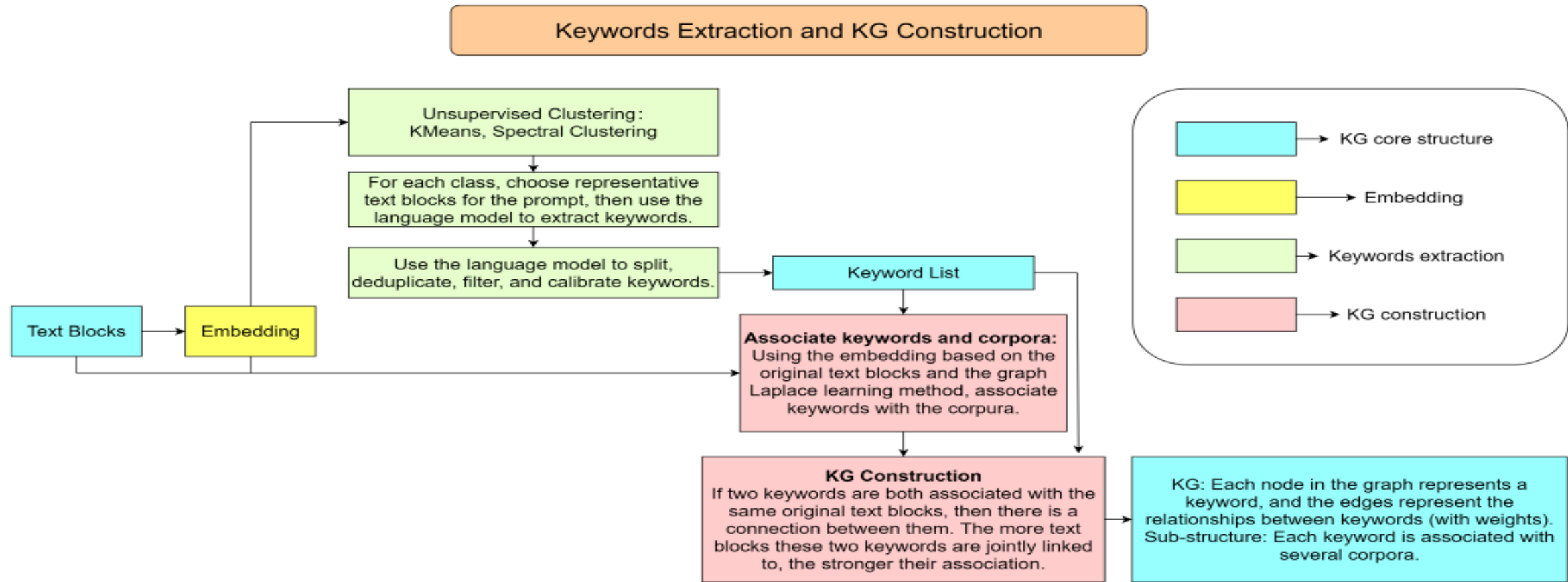


Fig. 1. Flowchart of the KG Construction Process. This figure illustrates the different steps involved in the construction of the KG. The blue blocks represent the core components of the KG, yellow blocks indicate the embedding process, green blocks focus on keyword extraction, and the red blocks correspond to the establishment of relationships between keywords and the corpus as well as among the keywords themselves.

# Programme of the afternoon

- 14h20 : Haoyi Xiong Natural Language based Context Modeling and Reasoning for Ubiquitous Computing with Large Language Models
- 14h40 : Graham Wilcock New technologies for spoken dialogue systems: LLMs, RAG and the GenAI Stack
- 15h00 : Eric de la Cergerie Coupling KG and LLM: a few directions
- 15h20 : Kristiina Jokinen Conversational Grounding, Trustworthy AI and Generative AI - Exploring LLMs for Active Healthy Aging
- 15h40 : Anthony Alcaraz **Towards Hybrid Reasoning: Assimilating Structure into Subsymbolic Systems**
- 16h00 : Christian Dugast AppTek's experience in building ClimateGPT, a factual domain specific LLM
- 16h20 : Hugues Sansen LifeLine
- 16h35 : Discussions starting with comments from Michael McTear and Maria Inès Torres
- 17h00 : Further discussions with a drink,...

# Natural Language based Context Modeling and Reasoning for Ubiquitous Computing with Large Language Models: A Tutorial

Haoyi Xiong (Ph.D from TSP 2015) & Daqing Zhang

Email: [haoyi.xiong.fr@ieee.org](mailto:haoyi.xiong.fr@ieee.org) [daqing.zhang@telecom-sudparis.eu](mailto:daqing.zhang@telecom-sudparis.eu)

# About Haoyi Xiong

- **Working Experience**

- Baidu Research, Big Data Lab
  - Principal Architect (2020.05—present); Staff Engineer (2018.05—2020.04);
- Missouri University of Science and Technology, Dept. CS, Rolla Mo, USA
  - Tenure-track Assistant Professor/Ph.D Advisor (2016.08—2018.08)
- University of Virginia, Dept. CS, Charlottesville VA, USA
  - Postdoctoral Research Associate (2015.07—2016.08)
- Télécom SudParis – CNRS UMR 5157, Evry, France
  - Postdoc (2015.02—2016.06), mentored by Vincent Gauthier

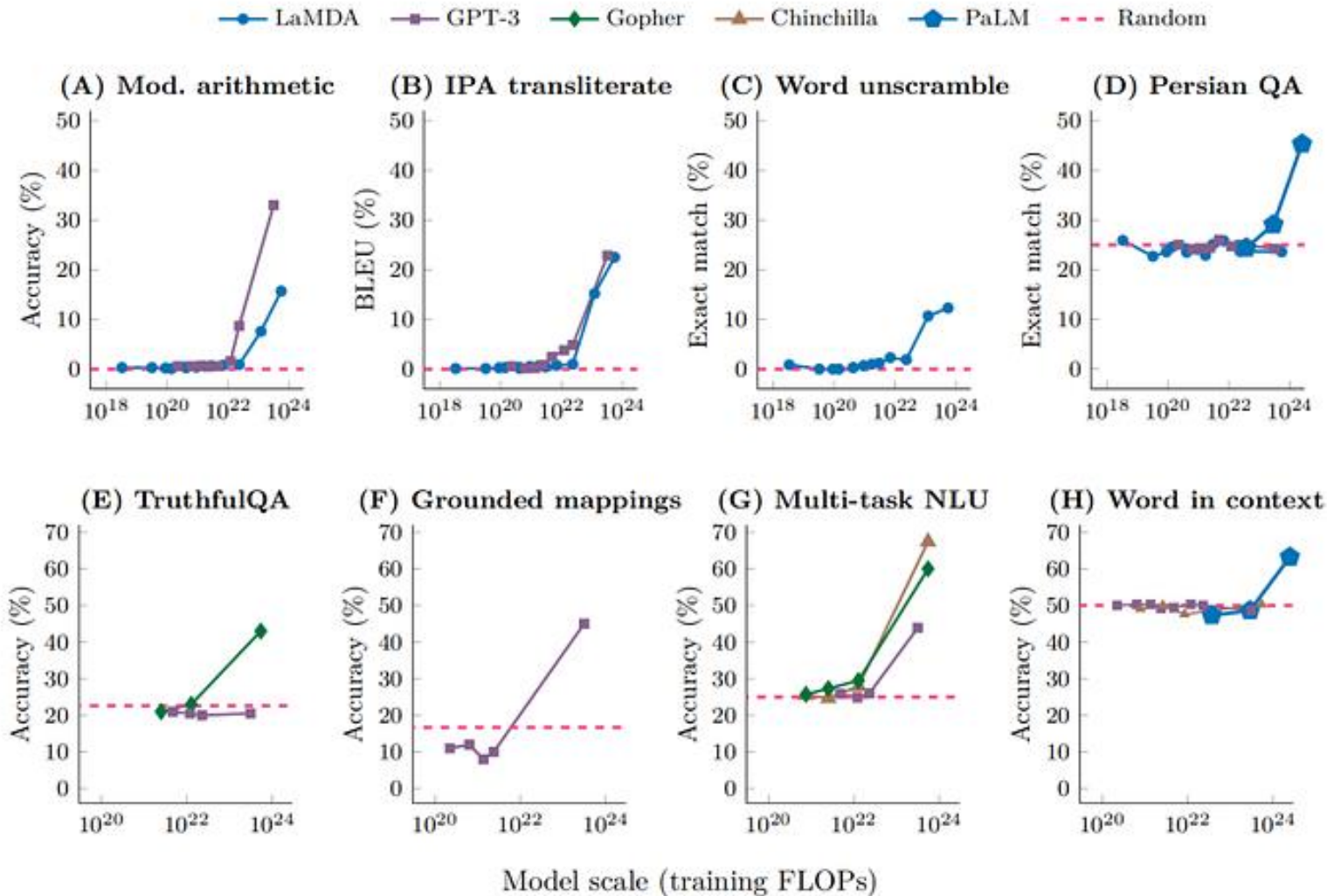
- **Education Backgrounds**

- Ph.D (Computer Science), Télécom SudParis & UPMC Paris VI, 2015
  - Advised by Profs. Monique Becker, Daqing Zhang, and Vincent Gauthier
- M.Sc (Information Technology), Hong Kong University of Science and Technology, 2010
- B.Eng (Electrical Engineering), Huazhong University of Science and Technology, 2009

# Outlines

- Foundation Models and LLMs: Trends and fundamentals
- Autonomous Agent: Old Concept but New Implementation
- LLM-driven Context-awareness: enabling pervasive computing with Agents
- Some examples of LLM-driven context-awareness: contexts and prompts

# The larger, the stronger (Language Models)

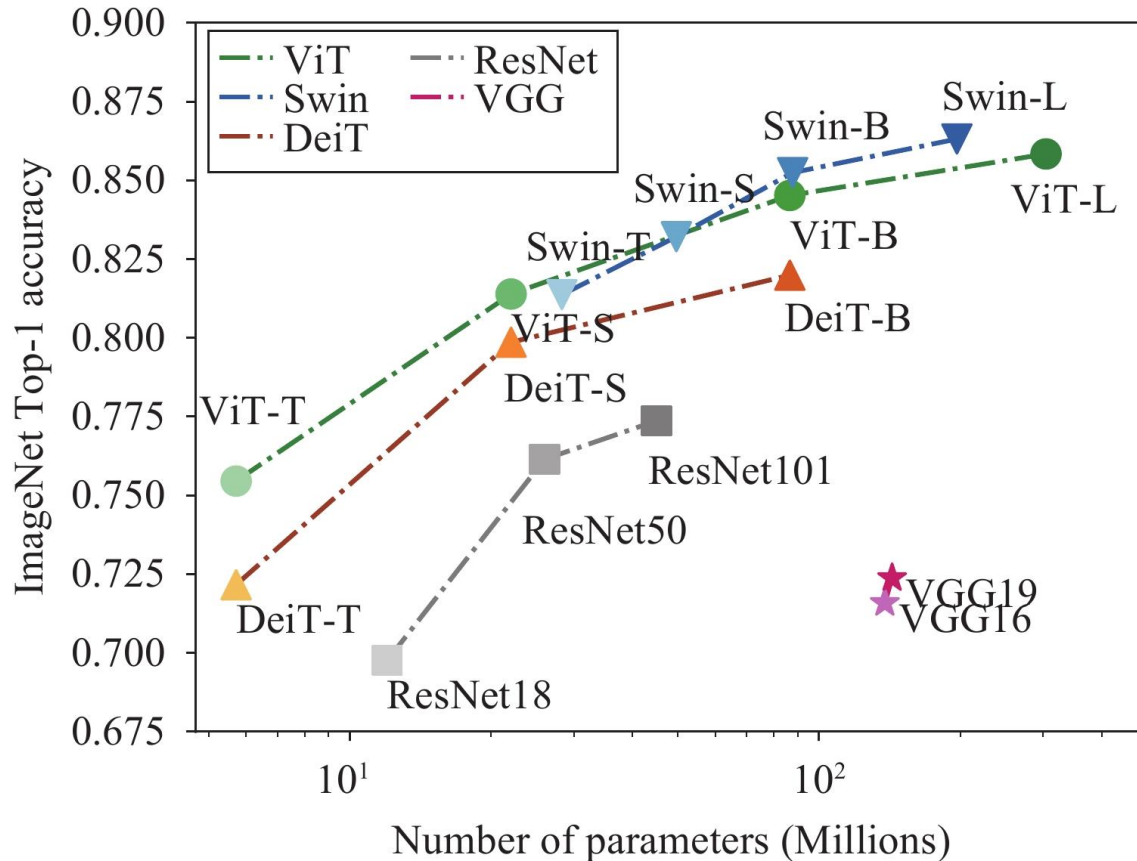


We are incredibly fortunate to be living in a "**monotonic**" era, where AI capabilities grow almost monotonically with model size, training data, and computational power.

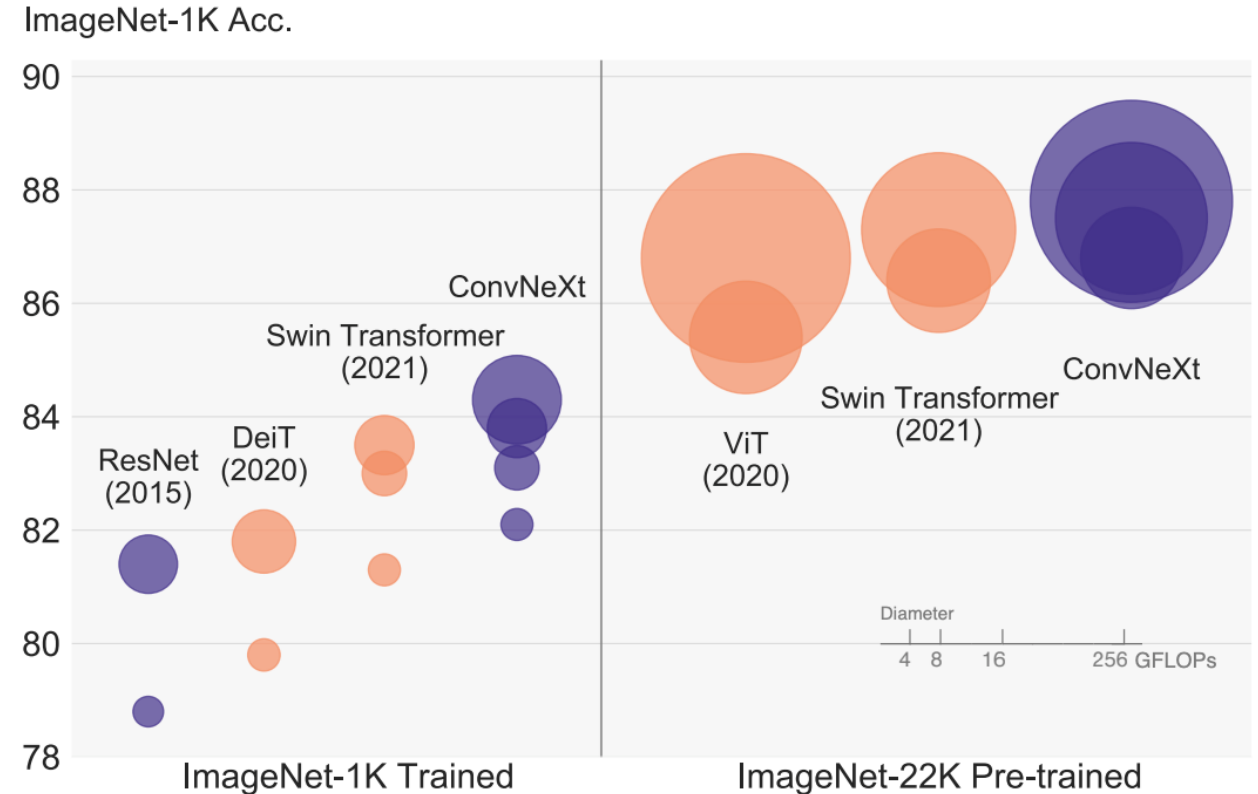
The unified development paradigm enables the creation of more effective and efficient AI systems, from language to vision and to multimodals, by leveraging the growth in model parameters, training samples, and other resources utilized.



# The larger, the stronger (Vision Models)

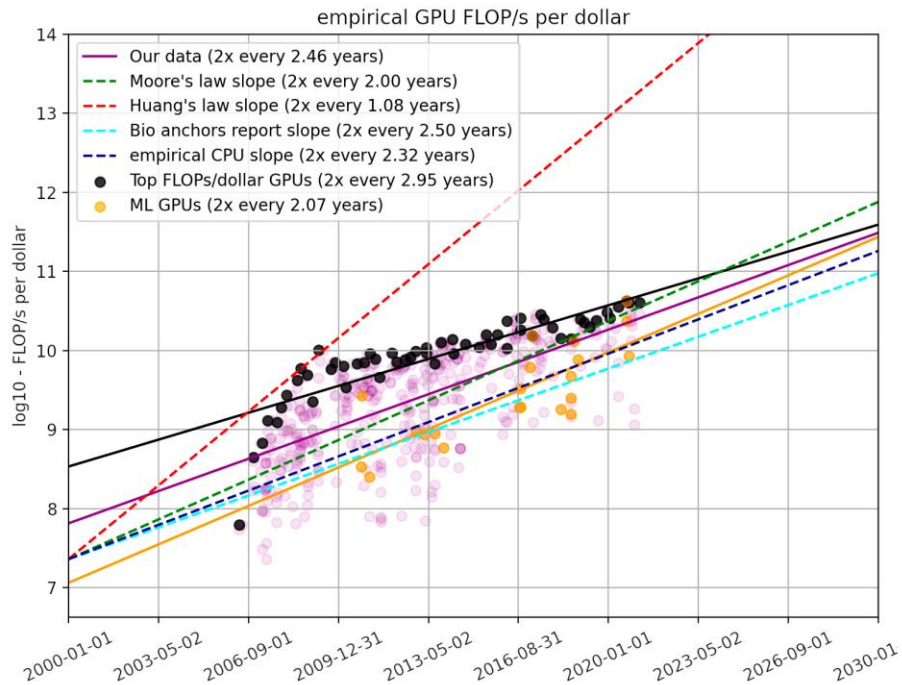


Zhou, Qiongyi, Changde Du, and Huiguang He. "Exploring the Brain-like Properties of Deep Neural Networks: A Neural Encoding Perspective." Machine Intelligence Research (2022): 1-17.



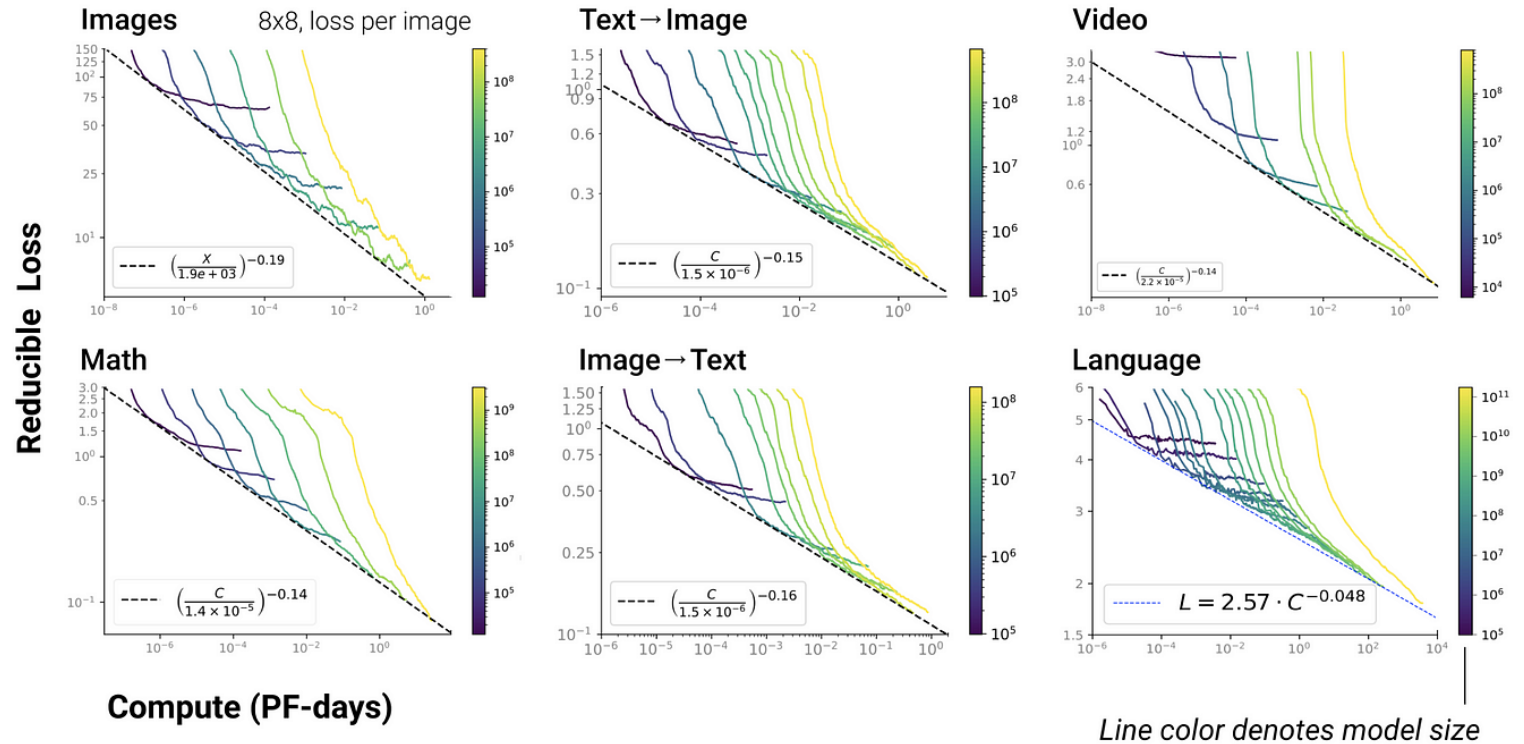
Liu, Zhuang, et al. "A convnet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

# Scaling laws (model abilities vs computation capacities)



The computational power per dollar increases **exponentially** over time. (Y-axis: FLOP/s in log-scale)

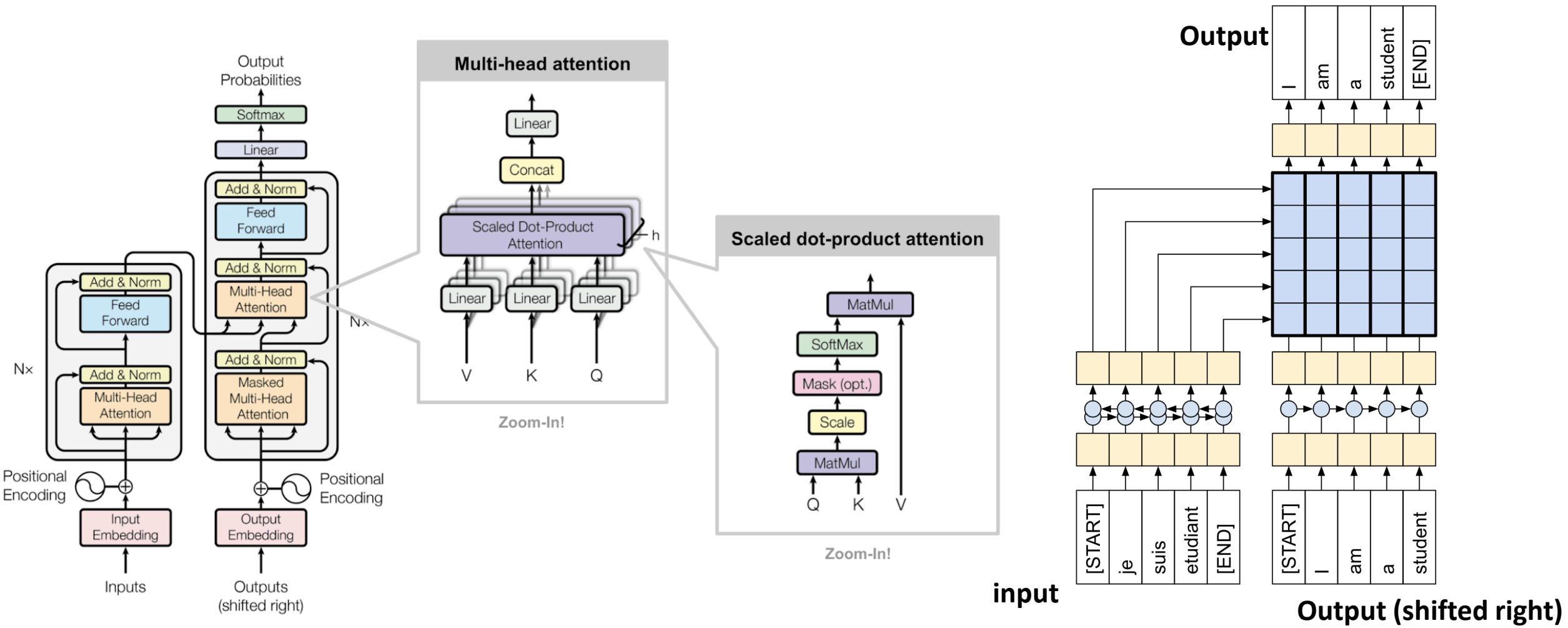
Marius Hobbhahn and Tamay Besiroglu . Trends in GPU price-performance. Epoch 2022.



The **log-log plots** of *testing losses* versus *computational costs* for training models in different sizes and for different tasks

<https://medium.com/@sharadjoshi/everything-you-need-to-know-about-scaling-laws-in-deep-learning-f4e1e559208e>

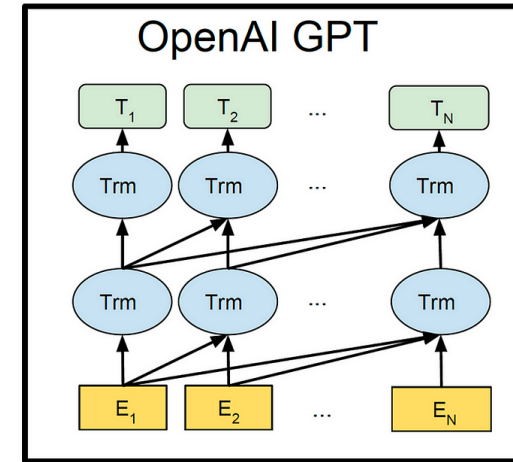
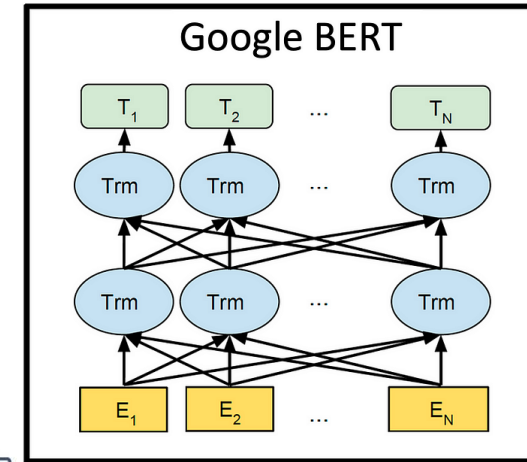
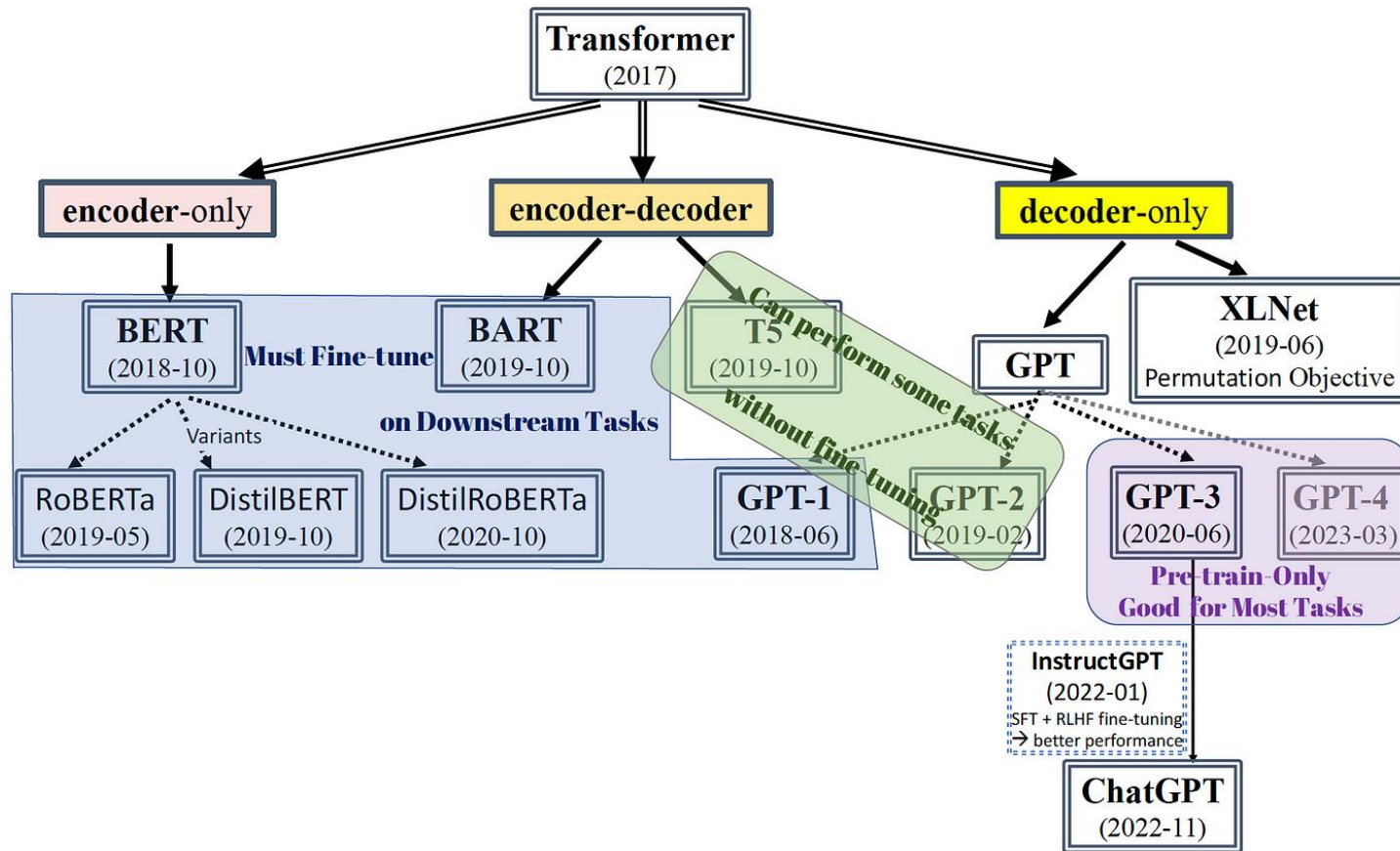
# Foundation bricks—transformer & attention mechanism



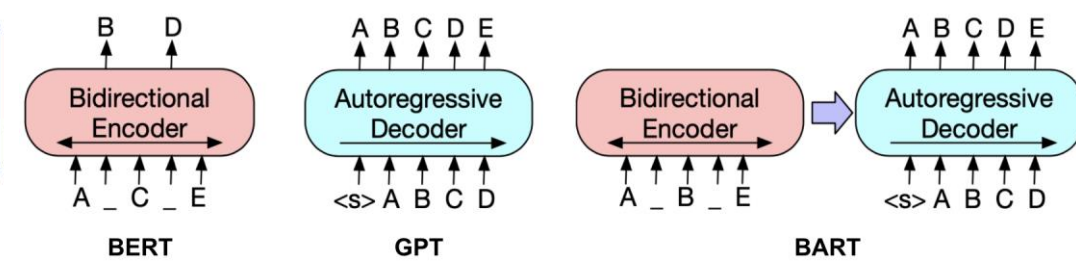
An example of transformer-based translation

- <https://neptune.ai/blog/bert-and-the-transformer-architecture>
- <https://www.tensorflow.org/text/tutorials/transformer>

# Transformer: BERT vs GPT



Bi-directional vs unidirectional attention flows



- Three types of self-supervised learning tasks:
1. Masked autoencoding for encoder training
  2. Autoregressive prediction for decoder training
  3. Doing both in one encoder-decoder structure

- <https://www.youtube.com/watch?v=iFhYwEi03Ew>
- <https://medium.com/the-modern-scientist/an-in-depth-look-at-the-transformer-based-models-22e5f5d17b6b>
- <https://lilianweng.github.io/posts/2019-01-31-lm/>

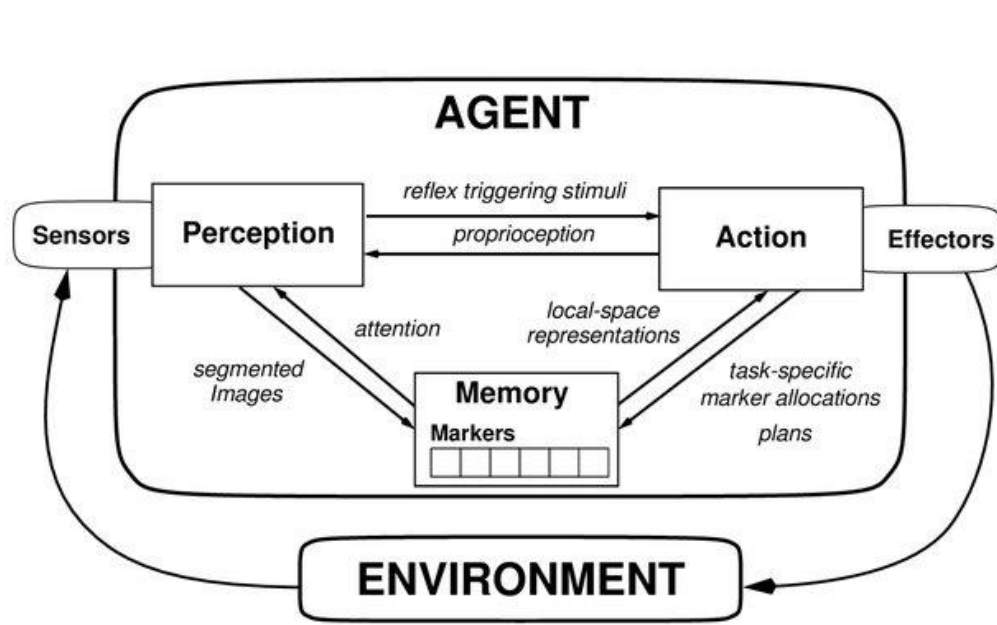
# A most recent benchmark on Some LLMs

	Average ▼	Multi-choice Qs ⬆	Reasoning ⬆	Python coding ⬆	Future Capabilities ⬆	Grade school math ⬆	Math Problems ⬆
Claude 3 Opus	84.83%	86.80%	95.40%	84.90%	86.80%	95.00%	60.10%
Gemini 1.5 Pro	80.08%	81.90%	92.50%	71.90%	84%	91.70%	58.50%
Gemini Ultra	79.52%	83.70%	87.80%	74.40%	83.60%	94.40%	53.20%
GPT-4	79.45%	86.40%	95.30%	67%	83.10%	92%	52.90%
Claude 3 Sonnet	76.55%	79.00%	89.00%	73.00%	82.90%	92.30%	43.10%
Claude 3 Haiku	73.08%	75.20%	85.90%	75.90%	73.70%	88.90%	38.90%
Gemini Pro	68.28%	71.80%	84.70%	67.70%	75%	77.90%	32.60%
Palm 2-L	65.82%	78.40%	86.80%	37.60%	77.70%	80%	34.40%
GPT-3.5	65.46%	70%	85.50%	48.10%	66.60%	57.10%	34.1%
Mixtral 8×7B	59.79%	70.60%	84.40%	40.20%	60.76%	74.40%	28.40%

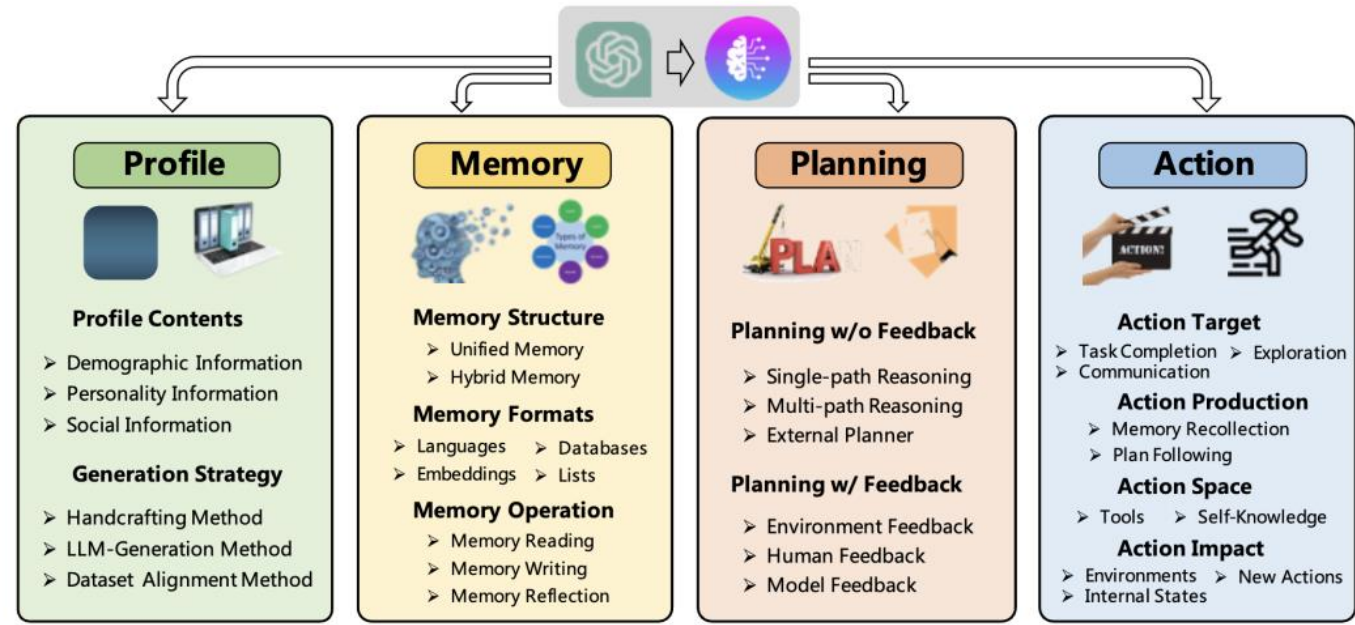
<https://www.vellum.ai/blog/llm-benchmarks-overview-limits-and-model-comparison>



# Autonomous Agent: Old Concept but New Implementation



Definition of Agent by 1996



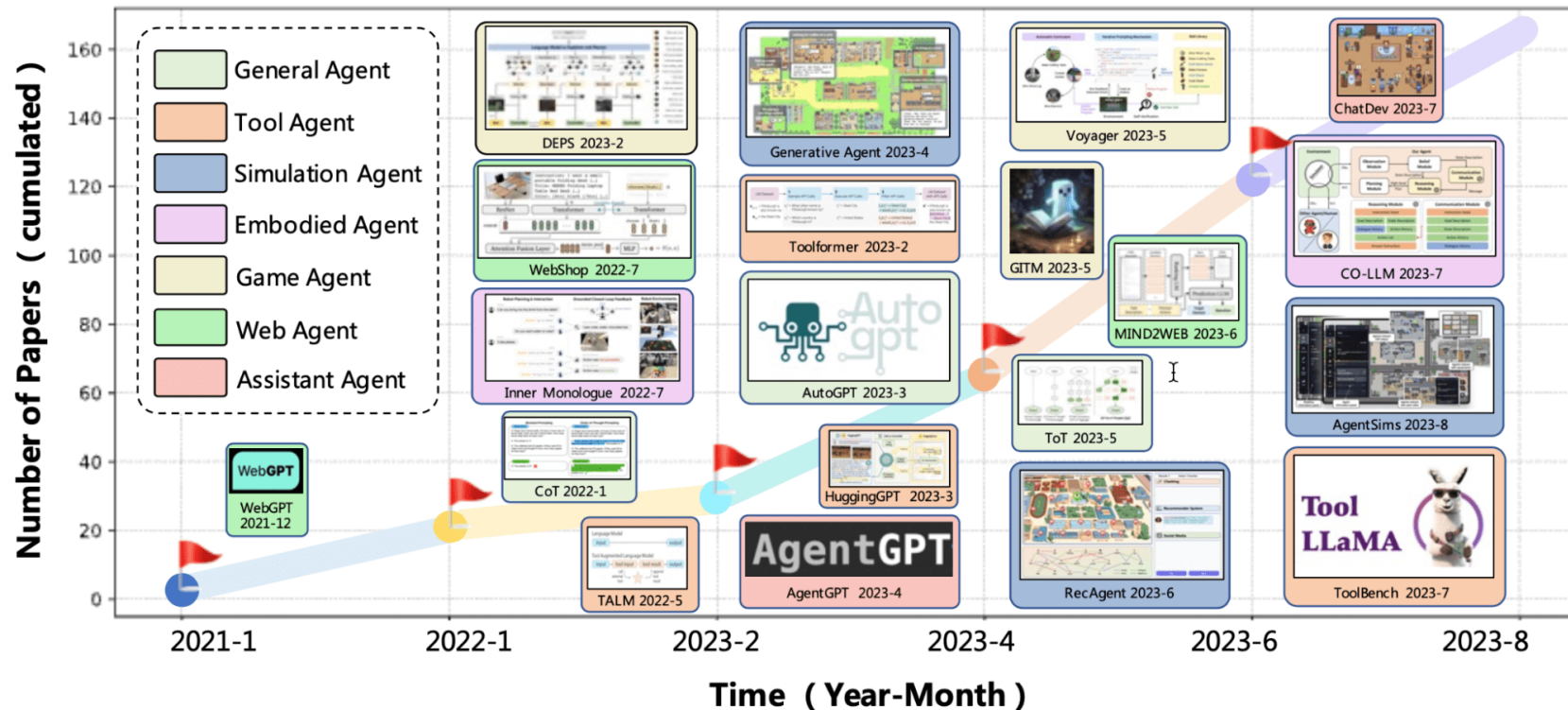
LLM-Driven Agents: Memory, Planning and Actions (2023)

## Three takeaways

- (Almost) the same definitions,
- The use of LLM for decision-making in planning, and
- The use of external tools for action.

- Brill III FZ. Representation of Local Space in Perception/Action Systems: Behaving Appropriately in Difficult Situations. University of Virginia; 1996.
- <https://www.kdnuggets.com/the-growth-behind-llmbased-autonomous-agents>

# Autonomous Agent: mind behind the trends



**Yann LeCun**  
@ylecun

My position/vision/proposal paper is finally available: "A Path Towards Autonomous Machine Intelligence"

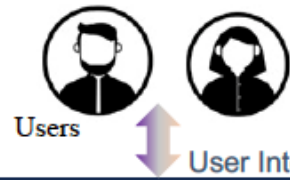
It is available on [OpenReview.net](https://openreview.net/forum?id=BZ5a1...) (not arXiv for now) so that people can post reviews, comments, and critiques:  
[openreview.net/forum?id=BZ5a1...](https://openreview.net/forum?id=BZ5a1...)  
1/N

- Agents enable the LLM a “world model”, which
1. Needs configuration to structure the brain,
  2. Interacts with the world with perception & action,
  3. Leverages short-long term memory to improves decision making,
  4. Makes decision for action through modeling and reasoning...

- <https://www.kdnuggets.com/the-growth-behind-llmbased-autonomous-agents>
- <https://twitter.com/ylecun>



# LLM-driven context-awareness: Anything new?

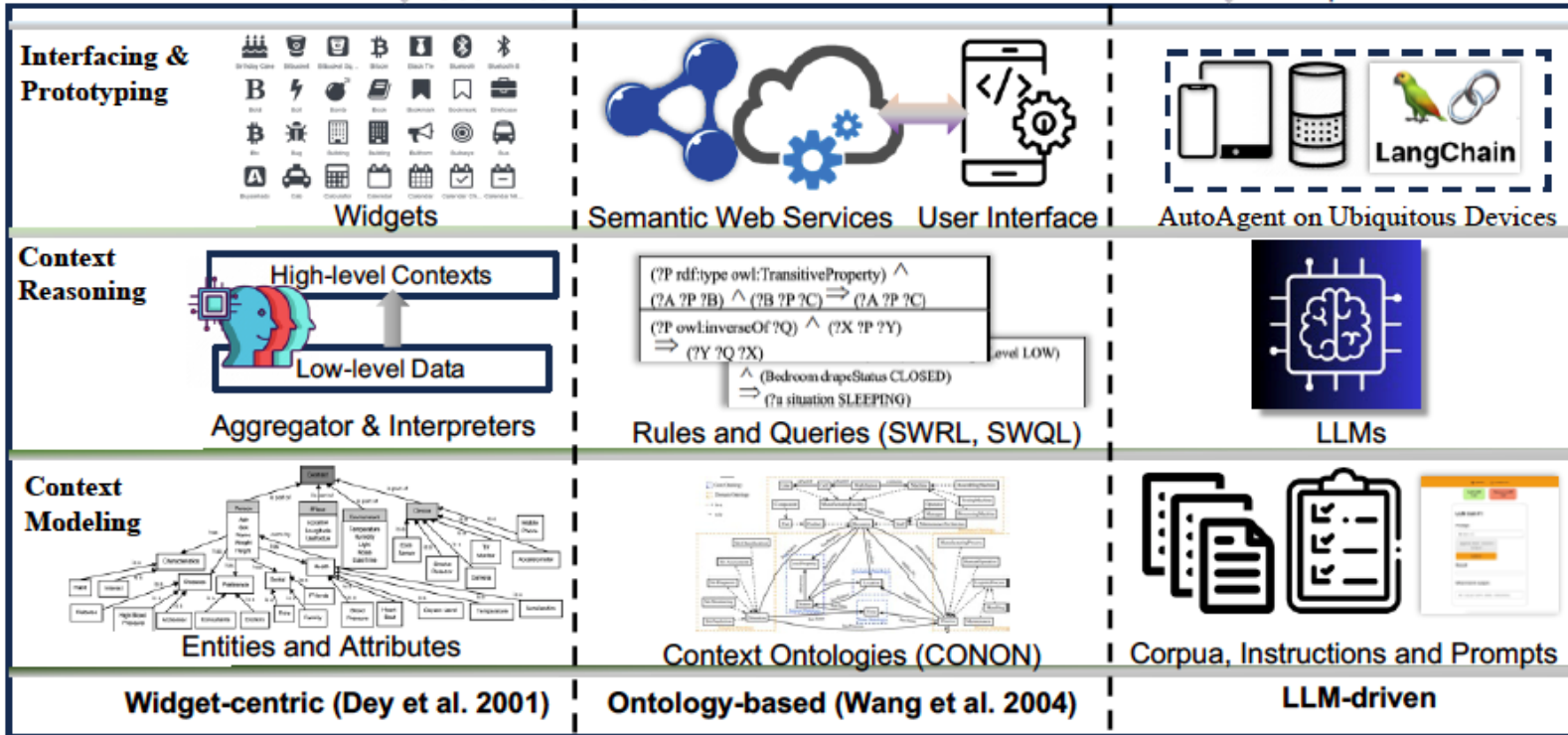


User Interactions

Sensors & Actuators



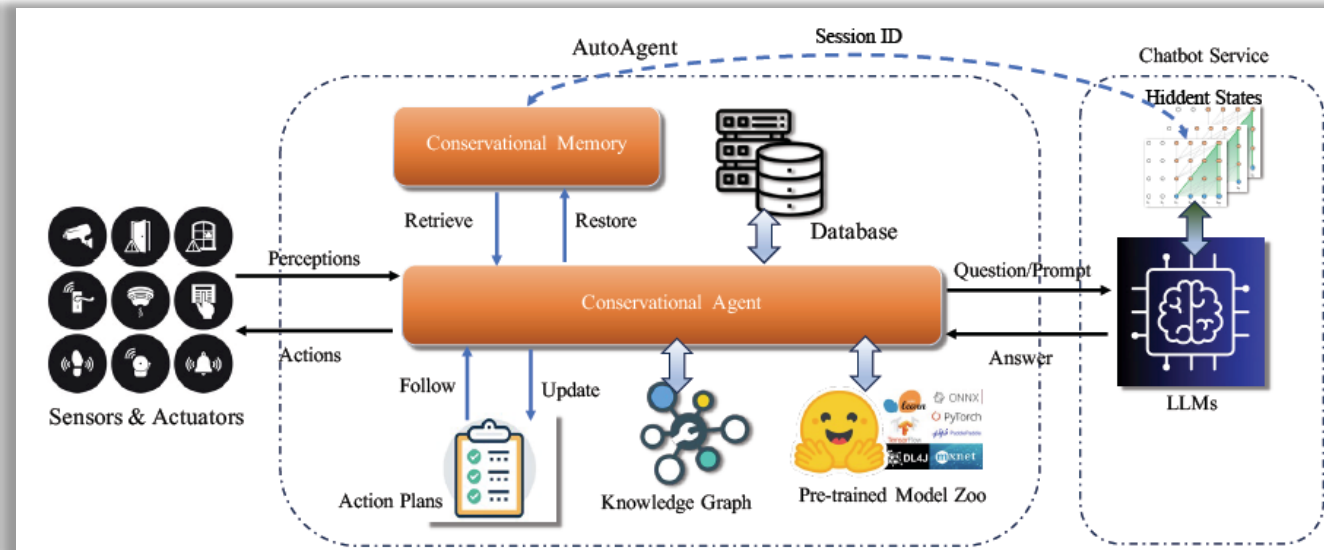
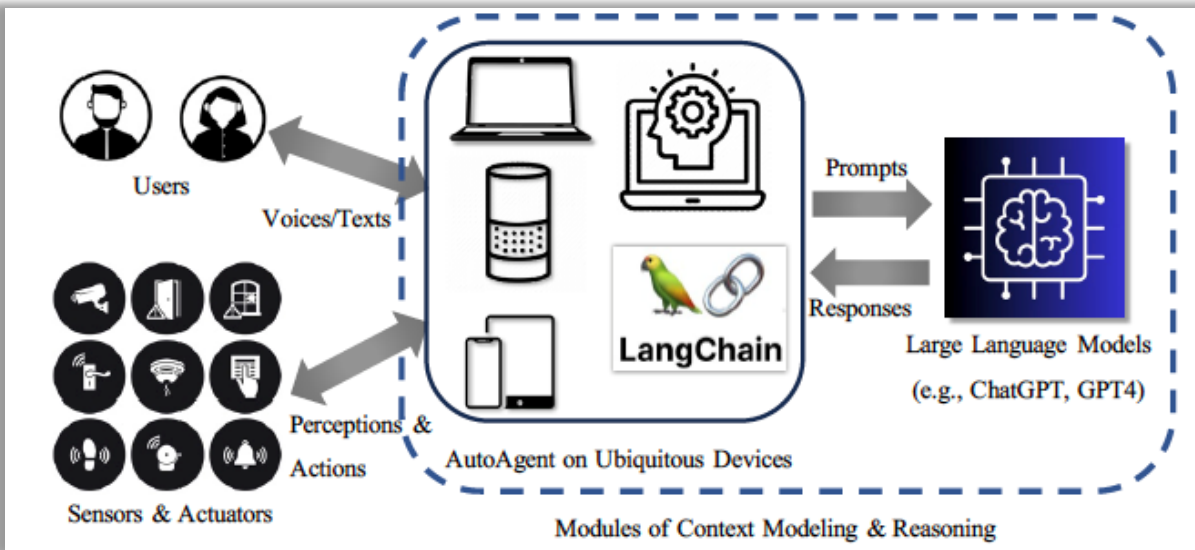
Perceptions & Actions



- 1. Easy-to-Use/Implement:** Adopting LUI (Language User Interfaces)+LangChain to interact with users and devices;
- 2. Intelligence:** Using LLMs to perform context reasoning;
- 3. Pervasiveness:** Using prompts and texts to model contexts.

Xiong, H., Bian, J., Yang, S., Zhang, X., Kong, L. and Zhang, D., 2023. Natural Language based Context Modeling and Reasoning with LLMs: A Tutorial. arXiv preprint arXiv:2309.15074.

# LLM-driven context-awareness: How does it work?



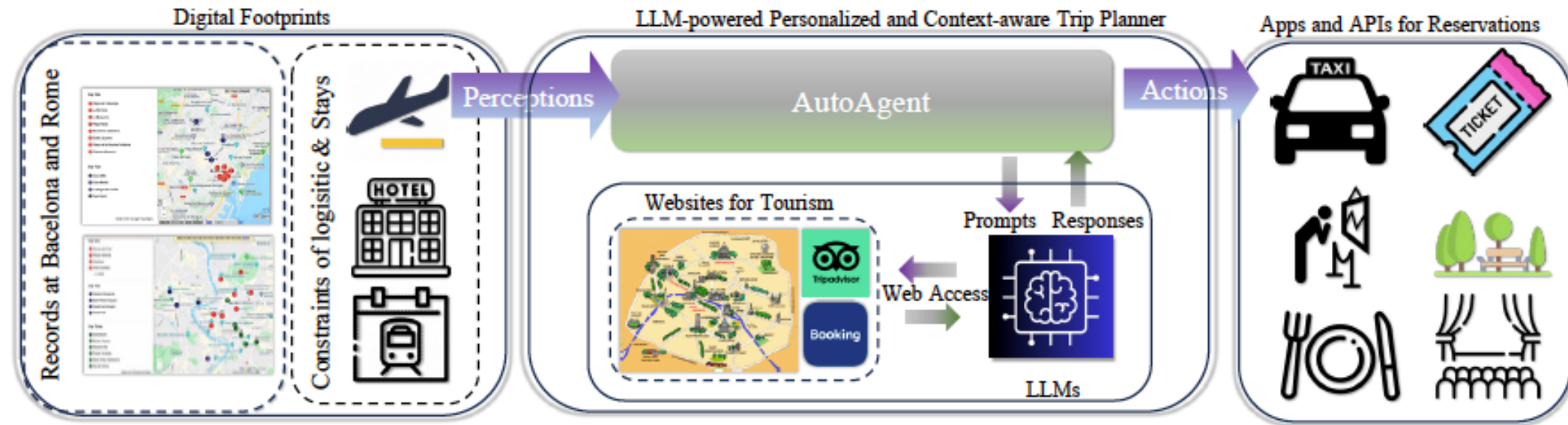
How every component work with each other:

- The user interacts with the Agent with instructions in the form of texts or voices;
- The agent percpts and reacts with physical words by external sensors, actuators and other tools;
- The Agent prompts LLMs and receives the responses to makes decisions for planning or action.

How we implement the Agent with LangChain

- A conversational agent that
  - handles the user's requests,
  - retrieves/restores data with long-term memory,
  - interacts with LLMs to adjust the action plan,
  - calls external tools to augmented LLMs for decision,
  - follows the generated plan for actioning,
  - engages with sensors & actutators for perception and action.

# Example: Trip Planner



## Requirements

- **Perception (personalisation):** learn to recommend Locations or Point-of-Interests from the user's past travel records.
- **Planning:** Be able to make a schedule of the trip based on the geospatial constraints of the user, e.g., arrivals/departures and locations of stay.
- **Actions:** Be able to convert the schedule to an actionable plan, booking or reserving necessities by incorporating external abilities.

# Example of Planning (w. personalisation)

## Prompt

Please answer the question by considering descriptions and examples below. \n  
Description: Suppose you are playing a role as a trip planner, which recommends attractions and schedules itinerary for the user, by considering following issues:\n  
1. Access internet contents for recommendation and scheduling.\n  
2. Learn the user's interests from the past travel records. \n  
3. Make the schedule satisfy the itinerary constraints. \n  
4. Consider the time spent and transportation to transit from one location to the next one. \n  
5. Please respond in the format [time stamp] action. location. est. time. \n

Past travel records: Rome, Italy: S.S. Lazio (Stadio Olimpico), Trionfale, and the Pantheon;\n  
Barcelona, Catalonia: La Boqueria, FC Barcelona (Spotify Camp Nou). \n  
Itinerary constraints: [17:10, 2023-07-23] Train Arriva: at Gare de SaintCloud.\n  
[10:00, 2023-07-24] Research talk, Weizmann France.\n  
[20:00, 2023-07-24] Banquet, Le Pavillon Des Princes.\n  
[13:00, 2023-07-25] Flight Departure from CDG Airport.\n

Examples:\n  
[17:00, 2011-01-01] Check out and Leave Waldorf Astoria New York, time est. 10mins.\n  
[17:10, 2011-01-01] Take an Uber to Per Se for dinner, time est. 20 mins. \n  
[17:30, 2011-01-01] Enjoy the dinner, 1 hours.\n  
[18:30, 2011-01-01] Check out Per Se, time est. 5 mins.\n  
[18:35, 2011-01-01] Take an Uber to LaGuardia airport, 40 mins.\n  
[19:15, 2011-01-01] Check in for the flight, 20 mins.\n

Question: Please make a trip plan to Paris and schedule the itinerary, recommend the hotels to stay, attractions to visit, restaurants for every meal, snakes and café. \n  
Answer:

replaceables

## GPT-4 Response

Based on your past travel records, it seems you appreciate historical sites, soccer, and local markets. So, I included these interests into this Paris trip plan. I have also taken into account your time and transportation constraints....

**Day 1: Arrival and Exploring Local Scene**  
[17:15, 2023-07-23] Take a cab to Le Bristol Paris, time est. 30 mins.  
[17:45, 2023-07-23] Check into Le Bristol Paris, time est. 15 mins.  
[19:00, 2023-07-23] Explore the local neighborhood, visit local patisseries and café, time est. 1 hour.  
[20:00, 2023-07-23] Dinner at Epicure, time est. 2 hours.

**Day 2: Research, Sightseeing, and Banquet**  
[09:00, 2023-07-24] Breakfast at café, time est. 30 mins.  
[10:00, 2023-07-24] Attend research talk at Weizmann France, time est. 2 hours.  
[12:30, 2023-07-24] Lunch at Le Jules Verne, time est. 1 hour 30 mins.  
[14:30, 2023-07-24] Visit the Eiffel Tower, time est. 2 hours.  
[17:00, 2023-07-24] Visit PSG (Parc des Princes), time est. 1 hour.  
[19:00, 2023-07-24] Prepare for Banquet at the hotel, time est. 1 hour.  
[20:00, 2023-07-24] Banquet at Le Pavillon Des Princes, time est. 3 hours.

**Day 3: Sightseeing and Departure**  
[09:00, 2023-07-25] Visit Louvre Museum, get a chance to see the Mona Lisa, time est. 2 hours.  
[11:30, 2023-07-25] Early lunch at L'Ambroisie, time est. 1 hour.  
[13:15, 2023-07-25] Transfer to CDG Airport, time est. 45 min.  
[14:00, 2023-07-25] Check-in for the flight, time est. 1 hour.

Please note that estimations are dependent on factors such as traffic and queues at the attractions. This plan incorporates a mix of local culture, football, and historical architecture ...

Without tuning the GPT-4 model, the Agent could make personalized trip planning according to his/her past trip records

- Profile the user from his/her past travel records,
- Understand his/her interests in traveling from the profiles,
- “Implicitly” match the potential point of interests and the user’s profile,
- Make personalized recommendation while satisfying the itinerary constraints, all in once!
- Context information (for personalisation and constraints) are replacable in a template...



# Example of Actioning (thru. Code generation)

**Prompt**

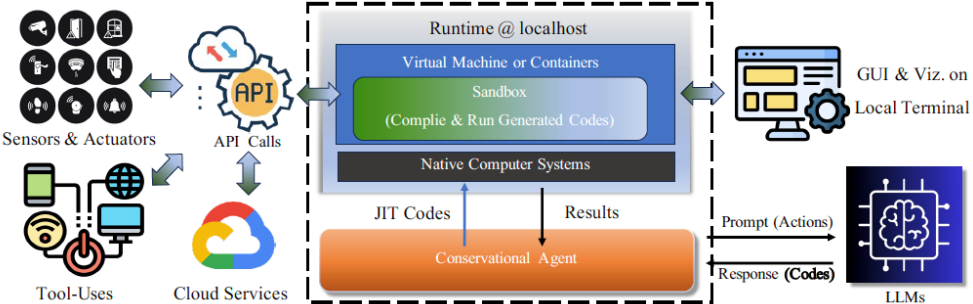
Please answer the question by considering descriptions and examples below.\n
 Description: Suppose you are playing a role as a program for trip reservations, which calls taxi, books hotel, reserves restaurants by calling APIs. The APIs are as follows.\n
 1. Bookings.bookHotel(UserID, HotelName, CheckinDate, CheckoutDate)\n
 2. Uber.reserveTaxi(UserID, DepartureLocation, DestinationLocation, DepatureTime)\n
 3. Yelp.reserveMeal(UserID, RestaurantName, ArrivialTime)\n
 4. Amadeus.ticketing(UserID, AttractionName, VisitTime)\n
 Example:\n
 Question: Please reserve meals, hotel rooms, taxis, and tickets of attractions for every schedule in the itinerary. \n
 Answer:

**GPT-4 Generated API Calls for Reservations**

```
Bookings.bookHotel(UserID, 'Le Bristol Paris', '2023-07-23', '2023-07-25')
Uber.reserveTaxi(UserID, 'Gare de SaintCloud', 'Le Bristol Paris', '2023-07-23 17:15')
Yelp.reserveMeal(UserID, 'Epicure', '2023-07-23 20:00')

Yelp.reserveMeal(UserID, 'Café near Le Bristol Paris', '2023-07-24 09:00')
Yelp.reserveMeal(UserID, 'Le Jules Verne', '2023-07-24 12:30')
Amadeus.ticketing(UserID, 'Eiffel Tower', '2023-07-24 14:30')
Amadeus.ticketing(UserID, 'Parc des Princes', '2023-07-24 17:00')

Amadeus.ticketing(UserID, 'Louvre Museum', '2023-07-25 09:00')
Yelp.reserveMeal(UserID, "L'Ambroisie", '2023-07-25 11:30')
Uber.reserveTaxi(UserID, 'Le Bristol Paris', 'CDG Airport', '2023-07-25 13:15')
```



Code generation and enviornment to run

By referencing the API definitions, the Agent generates codes to book tickets and reserve meals and taxis.

- Use short-term memory to recall the plan (generated in the last round of conversation),
- Call right API to do right things,
- Automatically fill the attributes for API calls,
- Turn the plan to an actionable.

\*This example might be over-simplified, one more call to look-up the location ID by the name of every location could be used.

# Key takeaways

- LLMs

- Be able to respond your requests through completing the dialogue;
- Be able to follow the instructions from a user when “prompted”;
- Know some “ingredients” of the world by pre-training, understand some specific domains by supervised fine-tuning (SFT).

- Agents

- Encapsulate pre-trained/fine-tuned LLMs with pre-defined sets of workflows (control flows & procedures);
- Formate the instructions to LLMs with prompt templates, while filling the replacables inside the template with the user’s request,
- Be able to resolve complex tasks through step-by-step planning, be able to make action through leveraging external tools.

Thank you!

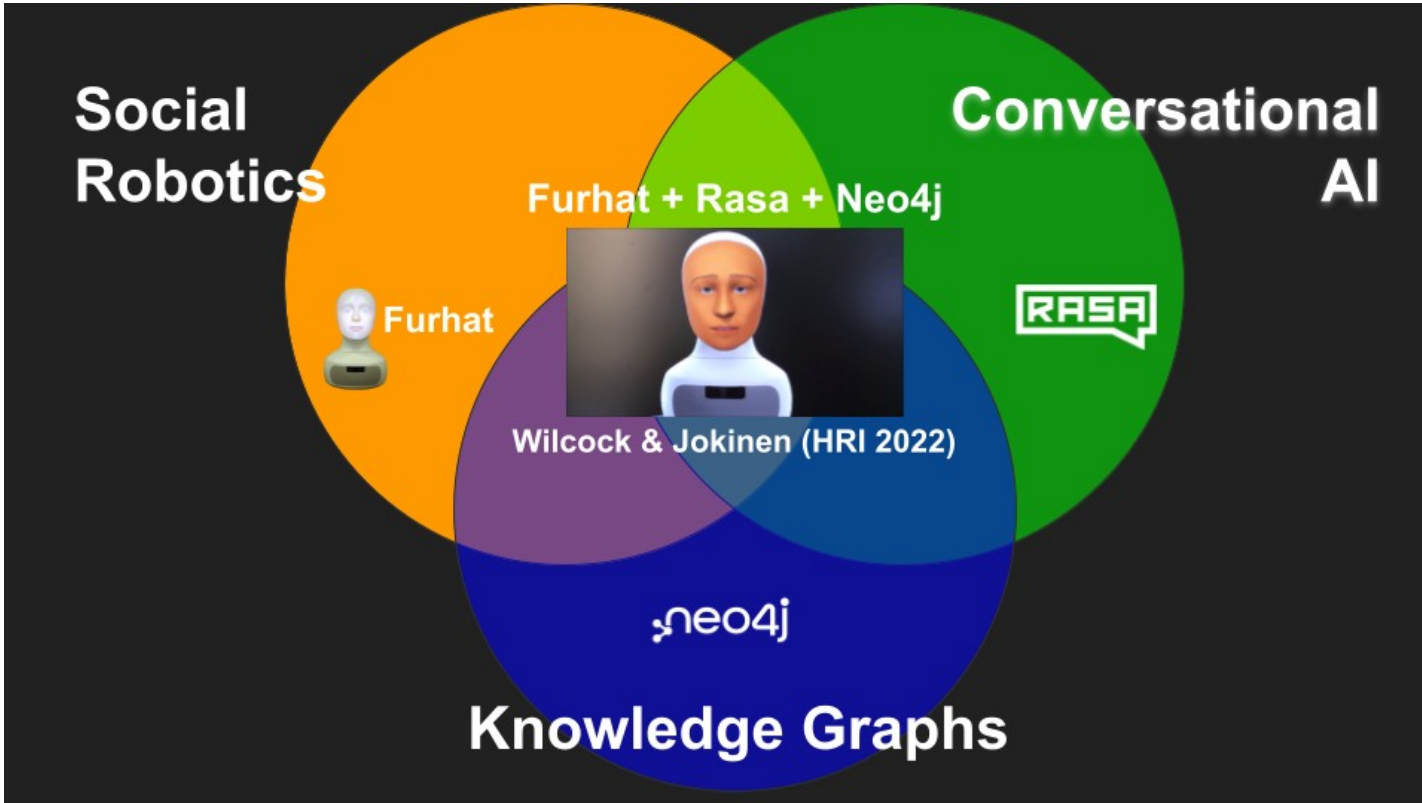


# **New Technologies for Spoken Dialogue Systems: LLMs, RAG and the GenAI Stack**

**Graham Wilcock**

University of Helsinki

# With CityTalk, Robots Search Knowledge Graphs



# From Conversational AI to Generative AI

- **Conversational AI**
  - Example open source tool: Rasa open source Conversational AI.
  - Successful for domain-specific dialogue systems, not open domain.
  - Transformers enabled successful domain-specific NLU.
  - NLG in Rasa has mainly been done by template-based generation.
- **Generative AI**
  - Example open source tools: Llama2, CodeLlama, LangChain.
  - Potential for success with open-domain dialogue systems.
  - LLMs can be successful for open-domain NLU.
  - LLMs can also be successful for open-domain NLG.

# Retrieval Augmented Generation (RAG) from Documents

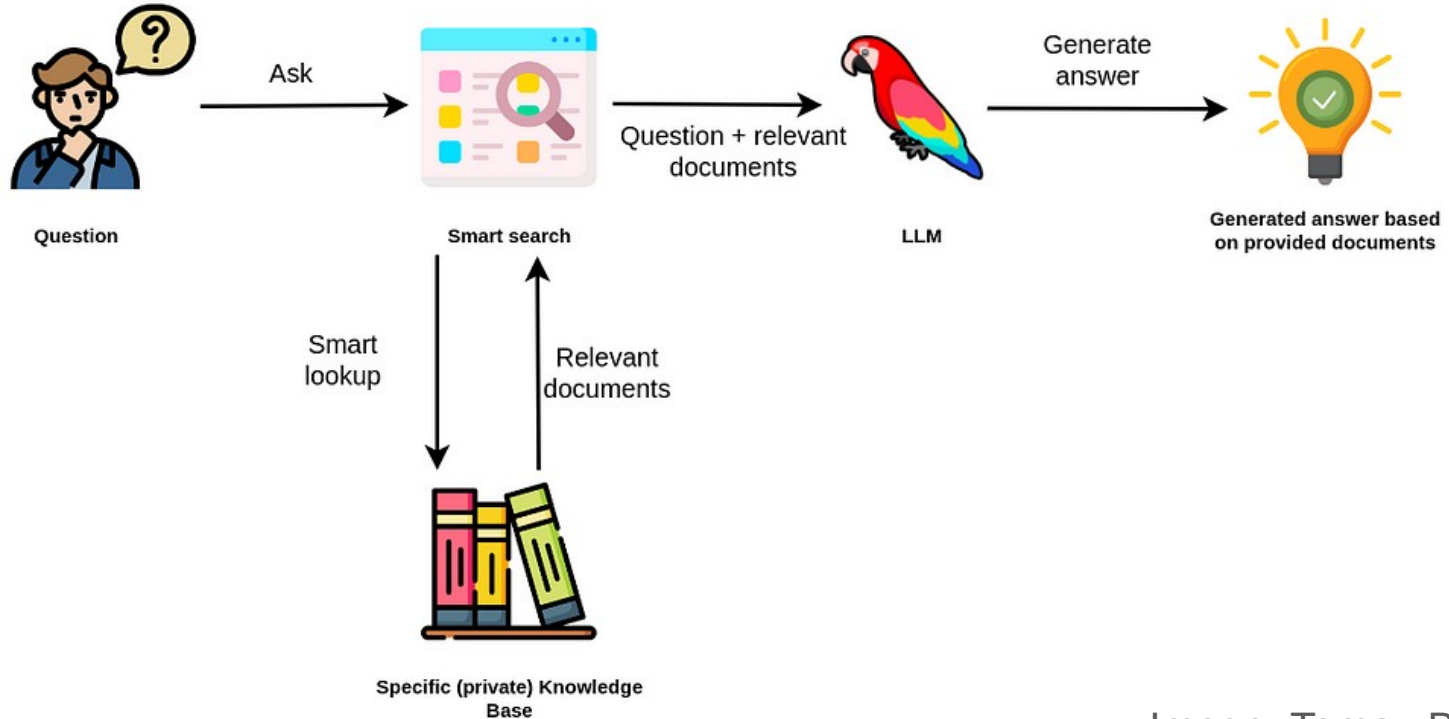
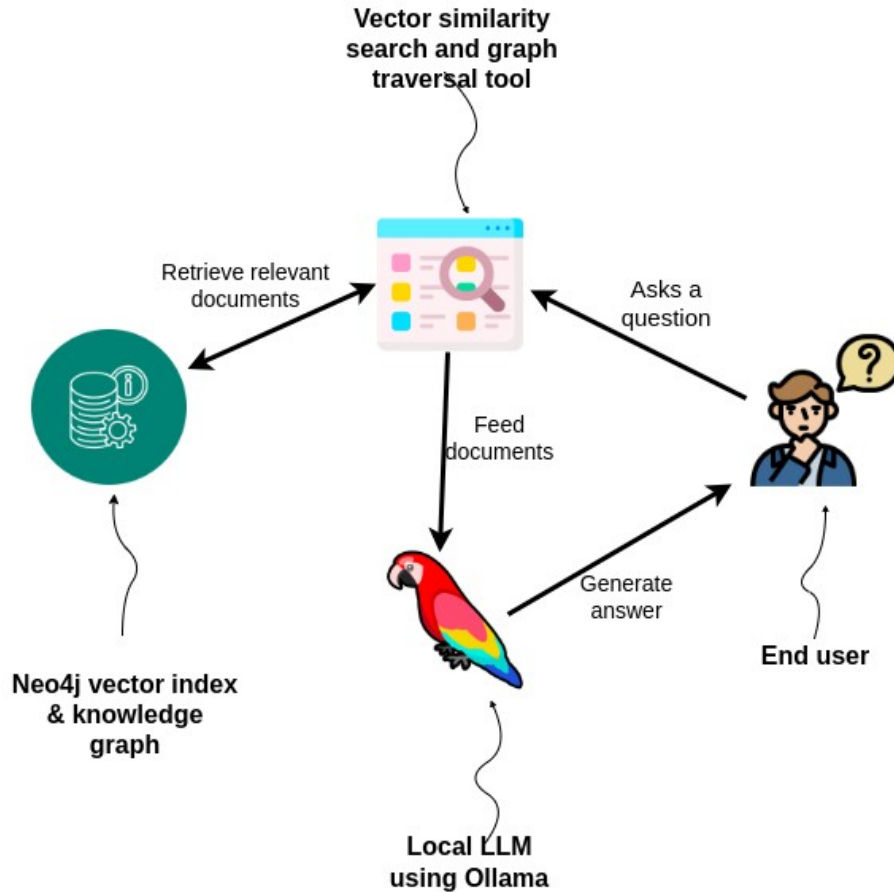


Image: Tomaz Bratanic

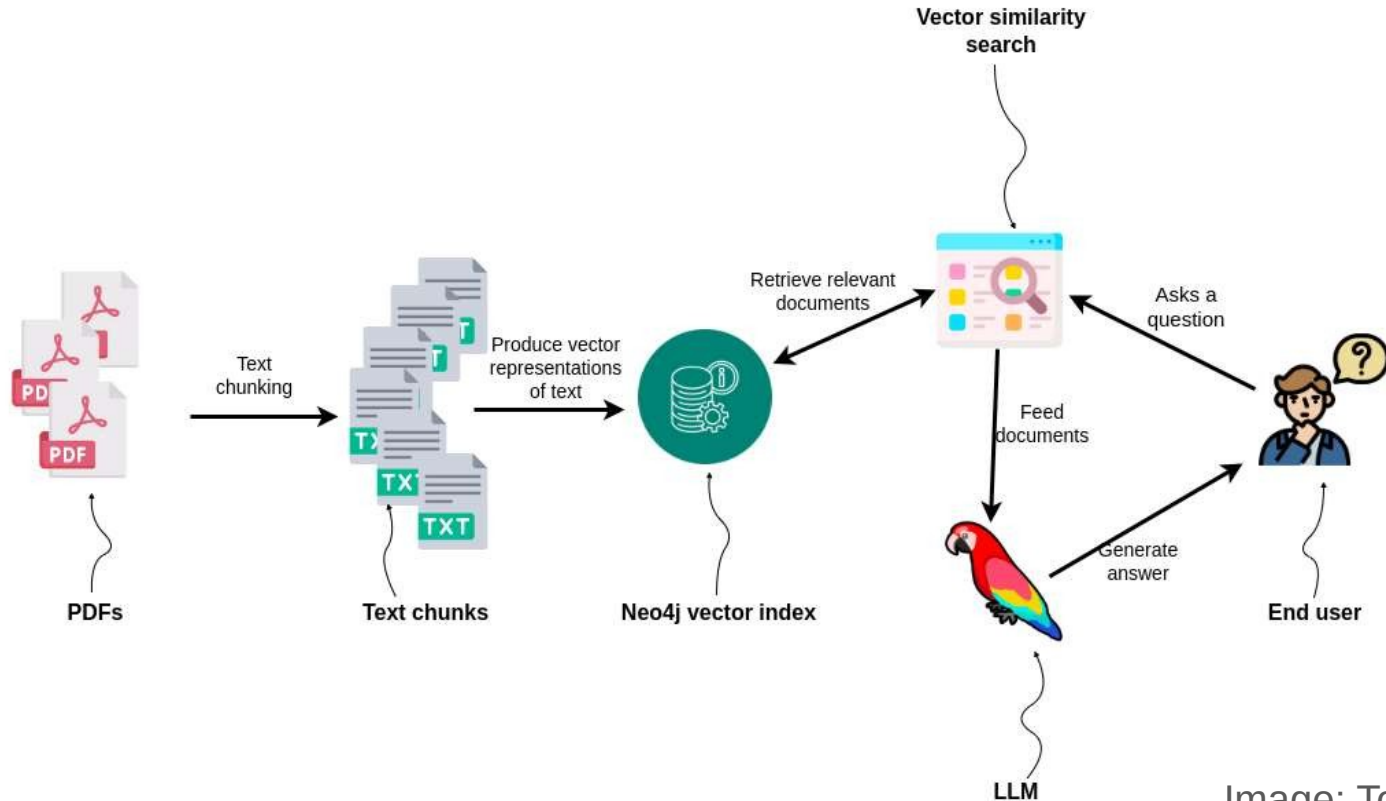
# RAG with GenAI Stack

## GenAI Stack (default options)



Graph database: Neo4j in Docker  
Vector database: Neo4j in Docker  
Embeddings: SentenceTransformers  
Local LLM: Llama2 from Ollama  
Document loaders: LangChain  
Text chunking: LangChain  
Conversation memory: LangChain  
User interface: Streamlit

# RAG from PDFs with GenAI Stack







# Chat with your pdf file

Upload your PDF



Drag and drop file here

Limit 200MB per file • PDF

Browse files



IWSDS-2016-65.pdf 77.4KB

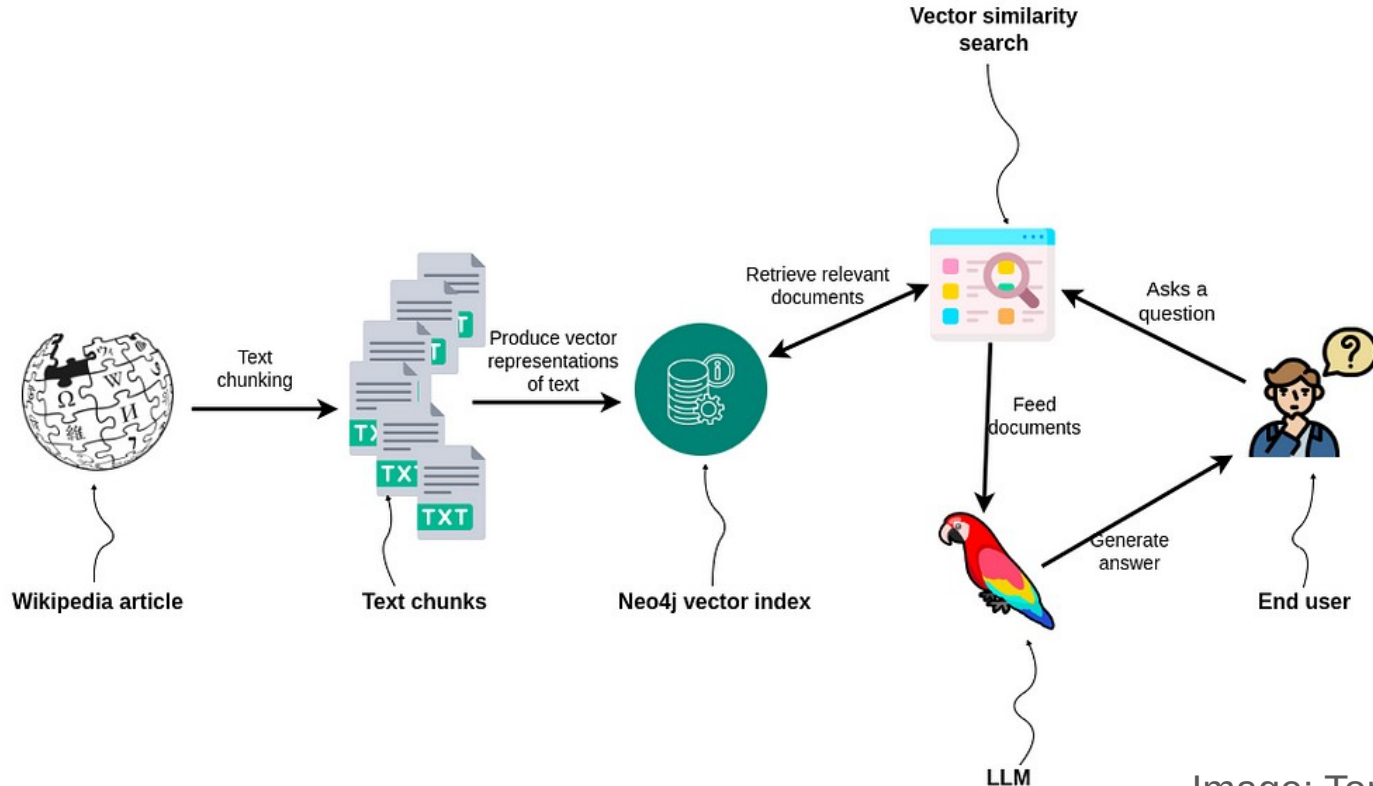


Ask questions about related your upload pdf file

Can robots speak the Sami language spoken in Lapland?

Yes, robots can speak the Sami language spoken in Lapland. The DigiSami project is currently underway to develop speech recognition and synthesis components for North Sami, one of the indigenous languages spoken in Lapland. These components are being developed using a new corpus provided by the University of Tromsø, and will be integrated into a humanoid robot called SamiTalk. Once completed, SamiTalk will be able to hold spoken dialogues with users in North Sami, access information from the Sami Wikipedia, and talk about requested topics using the Wikipedia texts.

# RAG from Wikipedia with GenAI Stack





# Chat with Wikipedia

Ask questions about Wikipedia article (William Shakespeare)

Did he have any children, grandchildren, or other descendants?

Yes, William Shakespeare had at least two known children:

1. Hamnet Shakespeare (baptized February 2, 1585 - died August 11, 1596)
2. Judith Shakespeare (twin of Hamnet, baptized February 2, 1585)

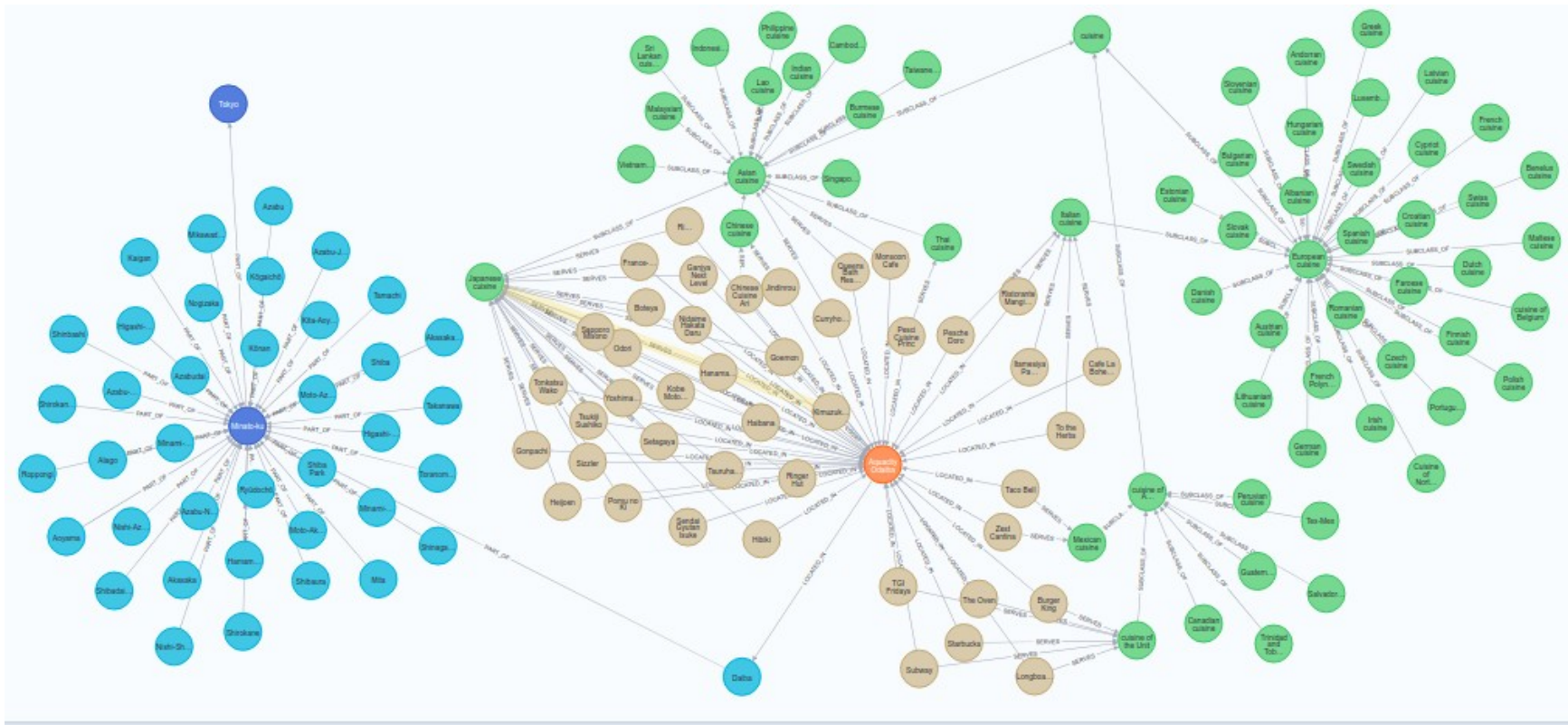
Hamnet's death at the age of 11 was a significant event in Shakespeare's life, and it is believed to have had an impact on his writing. However, there is no direct evidence that he had any other children or descendants.

It is worth noting that the exact number of Shakespeare's children and their names are not known for certain, as there are limited historical records available. Some scholars believe that Shakespeare may have had additional children beyond Hamnet and Judith, but this is purely speculative at this point.

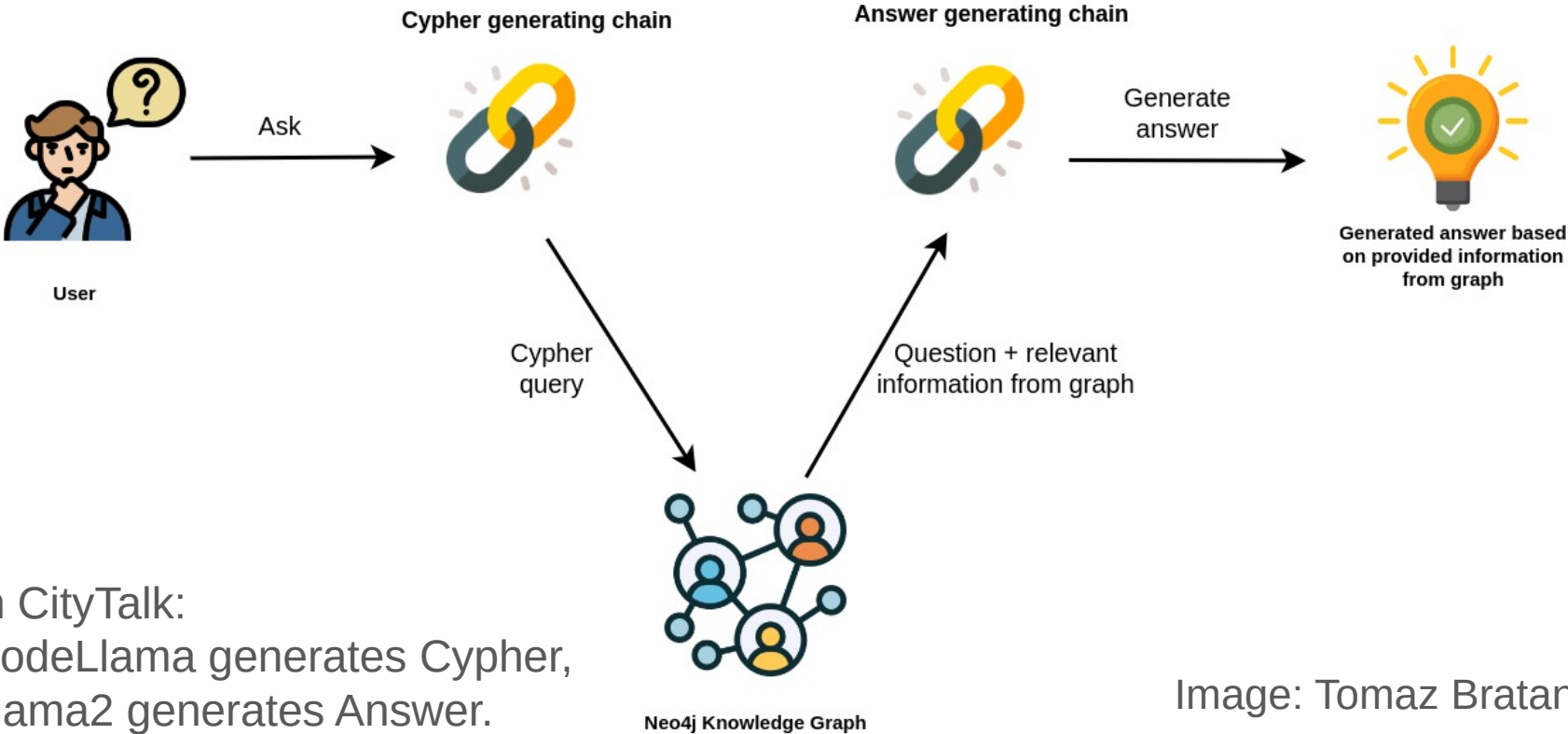
# CityTalk: RAG from Knowledge Graphs



# CityTalk Knowledge Graphs with Taxonomies (extract)



# RAG from Knowledge Graphs (GraphCypherQAChain)



In CityTalk:  
CodeLlama generates Cypher,  
Llama2 generates Answer.

Image: Tomaz Bratanic



# Chat with CityTalk

Ask questions about restaurants in Tokyo Waterfront

Are there any restaurants in Daiba that serve European cuisine?

Yes, there are several restaurants in Daiba that serve European cuisine. Here are a few options:

- Pesche Doro
- Ristorante Storia
- To the Herbs
- Cafe La Boheme

All of these restaurants serve Italian cuisine, which is a type of European cuisine.

# From Conversational AI to Generative AI

- **Conversational AI**
  - Example open source tool: Rasa open source Conversational AI.
  - Successful for domain-specific dialogue systems, not open domain.
  - Transformers enabled successful domain-specific NLU.
  - NLG in Rasa has mainly been done by template-based generation.
- **Generative AI**
  - Example open source tools: Llama2, CodeLlama, LangChain.
  - Potential for success with open-domain dialogue systems.
  - LLMs can be successful for open-domain NLU.
  - LLMs can also be successful for open-domain NLG.

# Coupling KG and LLM: a few directions



Éric de la Clergerie

<Eric.De\_La\_Clergerie@inria.fr>

**Almanach/INRIA**



e-ViT workshop  
on Knowledge Graphs and Large Language Models  
Évry, March 8th, 2024

- 1 Introduction
- 2 LLMs for « base » Conversion/Translation tasks
- 3 Integration
- 4 Interaction
- 5 Conclusion

# Data sources

Huge amount of unstructured textual sources, used by LLMs  
but also large amount of structured knowledge sources

- Semantic WEB
- Linked Open Data (LoD) : **DBPEDIA** : 9.5B triples, **WIKIDATA** : 108M items
- many specialized and local knowledge bases, potentially derived from other structured knowledge sources (e.g. SQL DB)



Figure 1: Relationships between various entities. Source: [10].

SIB3_ID	NAME	DOB	Gender	...
10	"John"	2009-11-12	"M"	...
13	"Jane"	2017-10-25	"F"	...

ADM_ID	SIB3_ID	Age	Reason	...
231	10	81	"arrhythmia"	...
232	13	56	"cancer"	...

DX_ID	ADM_ID	ICD9	...
15	231	"481.9"	...
16	232	"162.9"	...

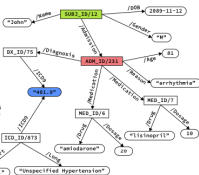
MED_ID	ADM_ID	DRUG	Dosage	...
6	231	"amlodarone"	20	...
7	232	"lisinapril"	10	...

ICD9 Dx Codes	ICD9 Short	Long
472	"481.9"	"Hypertension"
873	"481.9"	"Unsp. Hypertension"



Table to KG conversion

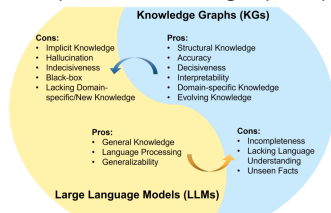


credit: Knowledge Graph-based Question Answering with Electronic Health Records

# Exploiting both data sources in a single setting

Many tasks may benefit from coupling language (LLMs) and knowledge (KGs) :

- access to more precise information (controlling hallucinations)
- access to up-to-date and/or private data
- entity linking (including disambiguation)
- question answering
- graph-based reasoning



credit: [Unifying Large Language Models and Knowledge Graphs: A Roadmap](#)

How to do it? Two main approaches (but many variants !):

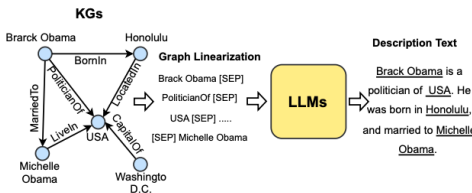
- **Integration** : coupling done at training time, coded in the parameters of a model (X-enhanced Y-model, fused model)
- **Interaction** : coupling done at inference time, through communications between two models

- 1 Introduction
- 2 LLMs for « base » Conversion/Translation tasks
- 3 Integration
- 4 Interaction
- 5 Conclusion

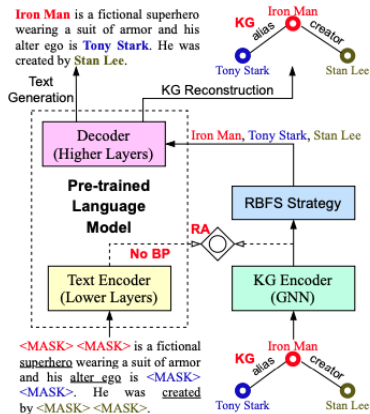


# Graph Linearization : from graph to text

- already useful to generate more readable texts from graph fragments
- could be done without LLMs but more diversity and fluency with LLMs



credit: Few-shot Knowledge Graph-to-Text Generation with Pretrained Language Models



credit: Investigating Pretrained Language Models for Graph-to-Text Generation

# From Natural Language queries to Graph queries

Formulating graph queries (SPARQL) may be complex  
Could be done assisted with LLMs



**Vous**

Write me a WikiData SPARQL query that finds all human genes



**ChatGPT**

To create a SPARQL query for WikiData that finds all human genes, you need to target items that are instances of "gene" (`Q7187`) and associated with the human species (`Q15978631`). The query would look something like this:

```
sparql Copy code
SELECT ?gene ?geneLabel
WHERE
(
  ?gene wdt:P31 wd:Q7187;      # Instance of gene
      wdt:P703 wd:Q15978631. # Found in human
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
)
```

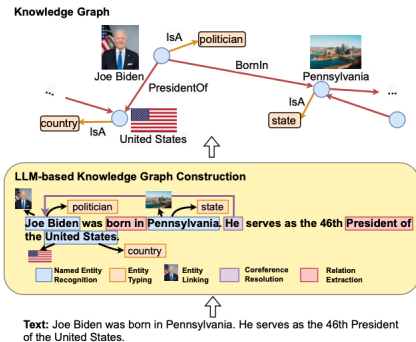
credit: author

SPARQA: Skeleton-based Semantic Parsing for Complex Questions over Knowledge Bases

- 1 Introduction
- 2 LLMs for « base » Conversion/Translation tasks
- 3 Integration**
- 4 Interaction
- 5 Conclusion

# Enriching KGs

Keeping KGs but enriching them with triples extracted from textual data through base LLMs or (Neural) Open Information Extraction



1 Vous

And as Turtle format for the following text:  
The sales for Xphone 27 have decreased by 27% over year 2032

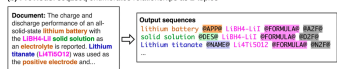
```
turtle
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ex: <http://example.org/>.

ex:xphone27 a ex:Product ;
ex:salesChange "2022"^^xsd:gYear [
  a ex:SalesChange ;
  ex:percentageChange "-27"^^xsd:decimal ;
] .
```

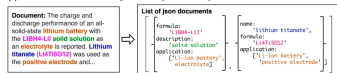
(a) Previous: Multi-step (pipeline) named entity recognition and relationship extraction



(b) Previous: seq2seq enumerates relationships as 2-tuples

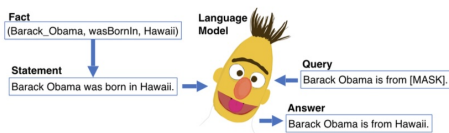


(c) This work: Hierarchical entity relationships without explicit enumeration



credit: Structured information extraction from scientific text with large language models

Transferring knowledge from KGs to LLM at pre-training time by linearizing graph triples or (random-walk) graph paths



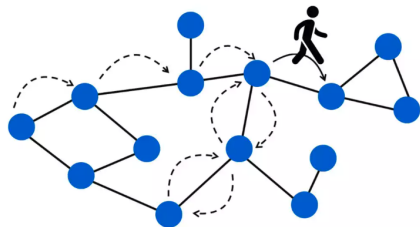
credit: Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries

in other words : one try to memorize KBs inside LLMs !  
but LLMs only memorize frequent facts  $\leadsto$  unsafe against hallucinations !

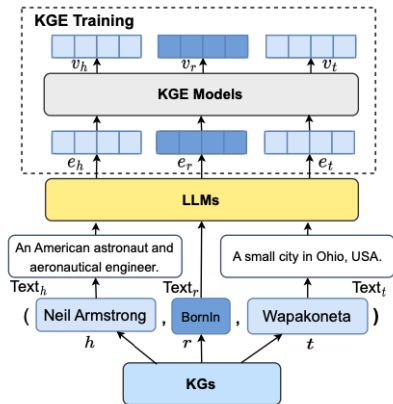
also a way to do **data augmentation** with **synthetic documents**  
e.g. **instruction tuning** on artificial but realistic queries and their answers

# Embeddings for KGs (or GNNs)

Embeddings may be computed on KGs based on their structure (random walks) but can also be enriched with (more semantic) LLM-based embeddings



topology-based embeddings  
(node2vec)



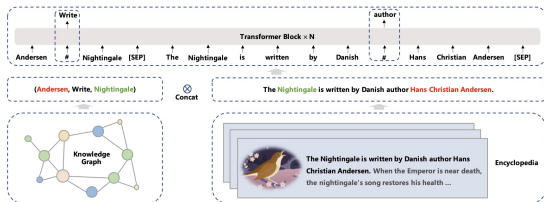
using LLM-embeddings

# Join pre-training in fused models

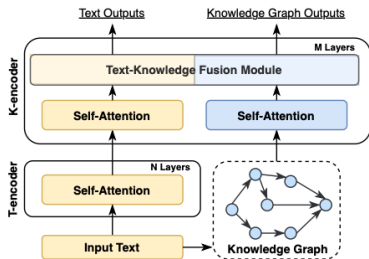
Contextual vector representations jointly learned on aligned texts and graphs masking elements on one side may benefit from the aligned other side

↪ fused models with

- two separate attention-based pipelines (Text and Graph)
- followed by one or more merging layers (cross-attention)



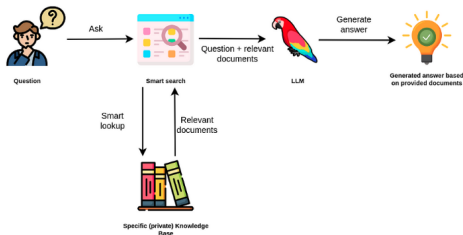
credit: ERNIE 3.0: LARGE-SCALE KNOWLEDGE ENHANCED PRE-TRAINING FOR LANGUAGE UNDERSTANDING AND GENERATION



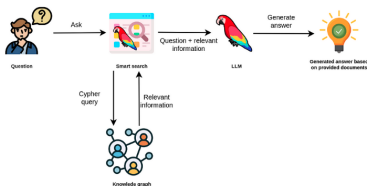


- 1 Introduction
- 2 LLMs for « base » Conversion/Translation tasks
- 3 Integration
- 4 Interaction**
- 5 Conclusion

Retrieval-Augmented Generation (**RAG**) : Given a query  $Q$ , documents most similar to its embedding  $e_Q$  are retrieved and added to  $Q$  as input to a LLM

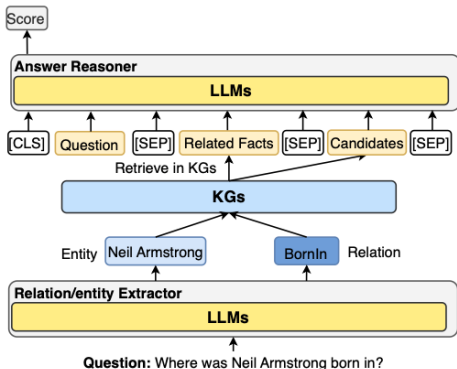


RAG may be adapted to KGs by retrieving **graph nodes or sub-graphs**, **linearize** them and add them to LLM context



# Augmented LLM querying KGs

Actually, LLMs may « **query** » KGs for information to be added to go further



- queries may be just entities, or more complex SPARQL queries
- several cycles of interactions between LLM and KG may occur (extending [chain-of-thought \[CoT\]](#) ideas, and X-of-thought variants)

# Multi-step interaction

## Also multi-hop reasoning

**Question:**  
What is the majority party now in the country where **Canberra** is located?

### LLM-only

(Chain-of-Thought Prompt): Let's think step by step.

Response: **Canberra** is the capital of **Australia**. According to my knowledge up to September 2021 the prime minister of Australia is **Scott Morrison**, who is a member of the **Liberal Party**. So the answer should be **Liberal Party**. ❌

(a)

### LLM ⊕ KG

(Prompt): Please generate a SPARQL query for this question.

Response: `SELECT ?country ?party WHERE { ?canberra dbprop:locatedIn ?country. ?country dbprop:majorityParty ?party. }`

Retrieve



Prompt

Response: Sorry, based on my query result from the knowledge base, I cannot answer your question since I do not have enough information. ❌

(b)

### LLM ⊕ KG

Looking for triples related to **Canberra**

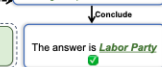
Think: The most relevant one is (**Canberra**, capital of, **Australia**). Information not enough for answering the question. Looking for triples related to **Australia**



Think: The most relevant one is (**Australia**, prime minister, **Anthony Albanese**).

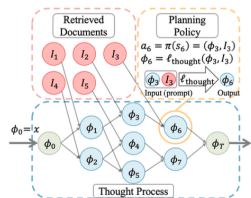


I know that **Anthony Albanese** is from **Labor Party**. Enough information is collected for answering this question.



Conclude: The answer is **Labor Party** ✅

(c)



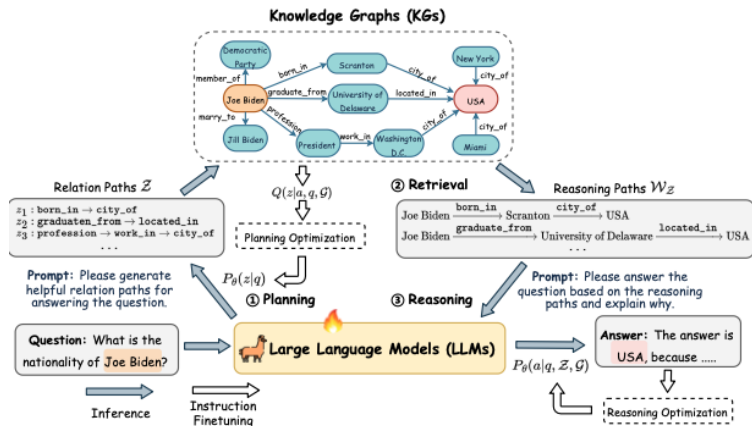
credit: Retrieval-Augmented Thought Process as Sequential Decision Making

credit: Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph

# LLM-guided graph reasoning

Besides queries and answers, LLMs may also be used

- generate hints to guide graph reasoning (reducing search space)
- generate explanations from retrieved sub-graphs



credit: Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning

- 1 Introduction
- 2 LLMs for « base » Conversion/Translation tasks
- 3 Integration
- 4 Interaction
- 5 Conclusion**

# Conclusion

- Language and Knowledge should play together
- Many ways to do it but interaction **much richer** than integration!
  - ▶ KGs flexible source of (local/dynamic/private) knowledge and allow for graph-based reasoning algorithms
  - ▶ LLMs generating queries and hints to navigate graphs
  - ▶ LLMs generating fluent answers (including sub-graph linearization)
- In other words, **KGs provide access to accurate facts**  
**LLMs provide language skills and some process knowledge**
- Probably useful to "**colorize**" a LLM for a given KB (~ domain adaptation) fine-tuning & instruction-tuning using KB's schema and vocabulary (**weak integration**)
- Maybe worth investigating specialized Language Models (as **agents**) wrapped around Knowledge Bases



# Conversational Grounding, Trustworthy Interaction and Generative AI

—

## Exploring LLMs for Active Healthy Aging

Kristiina Jokinen

AIRC, AIST Tokyo Waterfront

LLM-KG Workshop

March 8, 2024

# Paradigm Shift in Dialogue Modelling



# Paradigm Shift in Dialogue Modelling

## 1. Using large language models

- OpenAI: ChatGPT (now based on GPT4)
- Meta: LLaMA (Large Language Model Meta AI)
- Google: LaMDA (Language Model for Dialogue Applications)
- Huggingface: ChatGPT (based on GPT3.5-turbo)

## 2. Using knowledge graphs

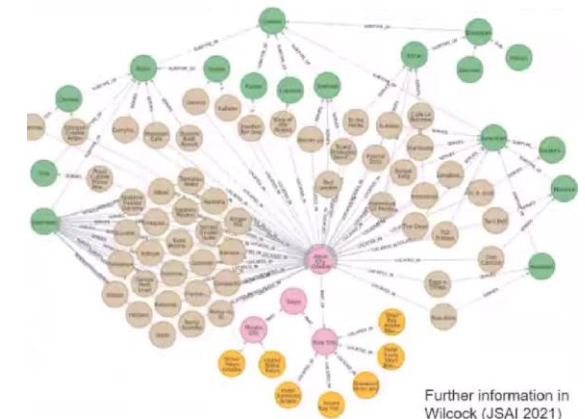
- Data provenance: knowledge curated by humans
- Truthfulness: Wikipedia, Wikidata taxonomy, taxonomies and ontologies for data augmentation
- Symbolic representation of objects, events, relations
- Graph search, Graph-to-text generation

## 3. Practical applications

- Balance between fluency and reliable information
- Support for various tasks besides providing useful information, send reminders, possibly give physical support



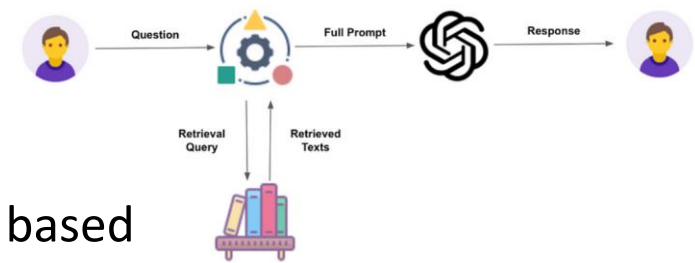
Icons from [Wikimedia Commons](#)



Further information in  
Wilcock (JSAI 2021)

# Starting Point

- Explore suitability of LLMs in practical real-world application of coaching
  - Coaching documents, prompt design, the user role
- Results:
  - The **GPT-model can distinguish** between interested and non-interested users based on the prompt instructions and the documents provided
  - **Able to provide training plans** and verbal descriptions of the information used
- However:
  - Content providers **need to check validity** of the interactions, training plans, etc.
  - Dialogue **continuation** needs to be secured
  - **Trustworthy** reliable information
  - Other issues that require further studies
    - network issues, rate limits, cost aspects, personal information
    - anthropomorphisation of the assistant, verbal imitation of the language





# Error types in human-robot interactions



## False implications

- Repeated questions about the same search parameter  
=>  
impression that there are items in the database that fulfil the user's request, although none exist



## Ontological errors

- False information and nonsense answers (LLM "hallucinations")  
=>  
lack of knowledge on semantic hierarchies, real world geography, synonyms, relations, ...



## Theory of Mind errors

- Different perspectives of the world (Baron-Cohen 1991)  
=>  
partitioning of knowledge bases into private vs shared beliefs
- Grounding of shared information



## Speech recognition errors

- Not LLM errors, but escalate the false information problem  
=>  
speech results should not be directly used as LLM input

## Solutions:

- More flexible knowledge graph searches,
  - Adding semantic metadata to knowledge graphs
- See video: <https://www.youtube.com/watch?v=QI5nbap5cRs>

Wilcock and Jokinen: To err is robotic; to earn trust, divine: comparing ChatGPT and knowledge graphs. RO-MAN conference August 2023.

# Towards Harnessing Large Language Models for Comprehension of Conversational Grounding

International Workshop on Spoken Dialogue System Technology IWSDS-2024

Sapporo 4-6 March, 2024

Kristiina Jokinen<sup>1</sup>, Phillip Schneider<sup>2</sup>, Taiga Mori<sup>1</sup>

<sup>1</sup>AI Research Center AIST, <sup>2</sup>Technical University of Munich

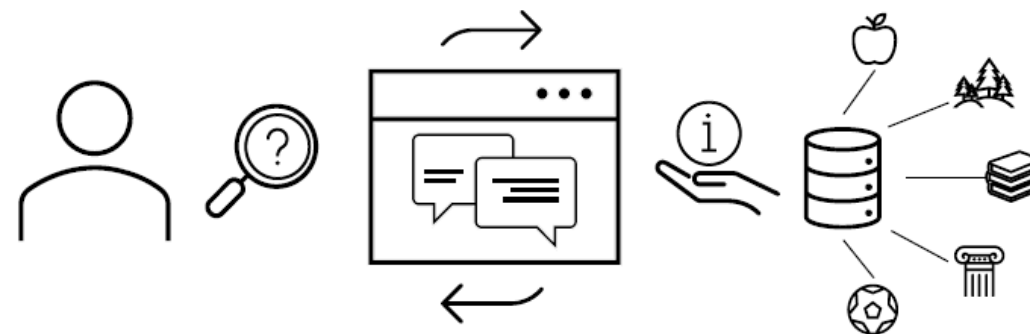


# Conversational Grounding

- General definition:  
Conversational grounding is a collaborative mechanism for **establishing mutual knowledge** among participants engaged in a dialogue
- Dialogue acts represent the communicative intention or function of a person's utterance, which can classify types of grounding
  - **Explicit** grounding: direct verbal feedback (e.g., "OK, great." or "Thanks!")
  - **Implicit** grounding: confirmation by moving forward with the conversation (e.g., inquiring about another concept)
  - **Clarification**: resolve uncertainty before moving forward with the conversation (e.g., clarifying a concept that was just introduced)

Information Seeker

Information Provider

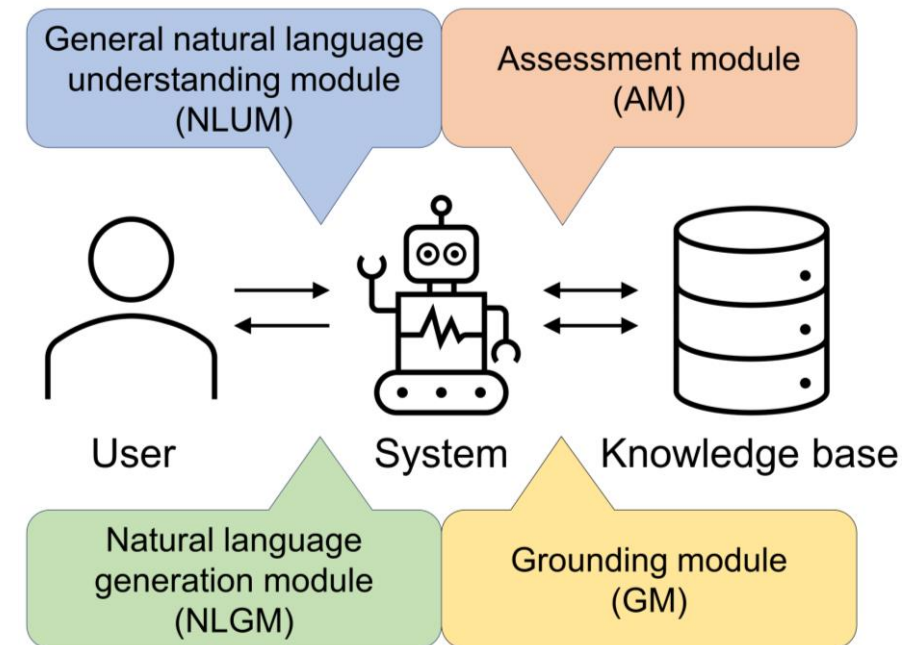


shared knowledge grows over time



# Annotation, Model Configuration and Prompts

- Preliminary analysis of a LLM in classifying grounding-related dialogue acts and extracting grounded knowledge elements
- We used an exploratory search dialogue corpus between two human participants that exchange information about a domain-specific tabular dataset (e.g., nature parks, media, nutrition, ...)
- In the corpus, information seeker (S) and provider (P) built up mutual knowledge about the tabular information in a chat room.
- Two researchers annotated **grounding types** (explicit, implicit, or clarification) and **grounded knowledge** elements in a JSON structure.
- We used the **GPT-3.5-Turbo** LLM for classifying the grounding type and extracting grounded knowledge.
- The system message contained the instruction and **few-shot prompt**, and the user message contained the complete conversation history up to the current turn.
- The token limit and the temperature were set to 256 and 0, respectively





# Prompts

## Classification Few-Shot Template

Predict the grounding label, representing when knowledge has been mutually grounded, for the last turn in the 'Input dialogue:'. The label can be 'explicit' if knowledge is verbally accepted, 'implicit' if accepted by moving forward with the conversation, or 'clarification' if a previous utterance must be clarified before acceptance.

USER: Input dialogue:

seeker: Can you tell me about the dataset's content?

provider: The dataset contains information about planets in our solar system.

seeker: What is the number of columns in the dataset?

ASSISTANT: Output label: implicit

...

## Information Extraction Few-Shot Template

Predict the newly grounded knowledge for the last turn in the 'Input dialogue:'. Use the JSON structure: {'table domain': str, 'table content': str, 'row count': int, 'column count': int, 'column info': [{'column name': str, 'values': [], 'distinct count': int, 'min value': int, 'max value': int}]}. Adhere strictly to the JSON structure, and only predict the attributes mentioned in the dialogue turns, leaving unmentioned attributes as null.

USER: Input dialogue:

seeker: Can you tell me about the dataset's content?

provider: The dataset contains information about planets in our solar system.

seeker: What is the number of columns in the dataset?

ASSISTANT: Output JSON: {'table content': 'planets of the solar system'}

...

# Results

- In the grounding type classification task, GPT-3.5-Turbo encountered **challenges**.
  - Explicit grounding was **mostly correctly classified** as in turn 7 of Dialogue B because it can be observed in the text in forms such as *OK* and *great*.
  - Implicit grounding and clarification were easily **confused** as in turn 8 of Dialogue A as both can involve questions and require contextual dialogue understanding.
  - There were two instances where the LLM predicts explicit grounding despite them being questions related to clarification or implicit grounding as in turn 5 of Dialogue B.
- Linguistic phenomena like **co-reference** and **ellipsis** might have added another level of complexity to classifying these grounding acts.
- In the grounded knowledge extraction task, GPT-3.5-Turbo demonstrated **better overall performance**.
  - The LLM accurately gathers **the relevant information** as in turn 4 of Dialogue A even though it mixes up the similar attributes “table domain” and “table content”.
  - The model adeptly handles **numerical information**, successfully determining the number of rows in a table or counts of unique values for specific columns as in turn 6 of Dialogue A.

Utterance	Grounding Type	Grounded Knowledge
<b>Dialogue A</b>		
4 S: How many rows are there in the dataset?	<i>I=I</i>	{'table domain': 'time travel works of fiction'} <del>≠</del> {'table content': 'time travel works of fiction'}
5 P: 500		
6 S: What are the attributes of the dataset?	<i>E≠I</i>	{'row count': 500}= <del>≠</del> {'row count': 500}
7 P: year, title, author, short text description		
8 S: Is there no column for the type of the work? How then can I determine if a work is a novel or a film?	<i>I≠C</i>	{'column names': ['year', 'title', 'author', 'short text description', 'type of work']} <del>≠</del> {'column names': ['year', 'title', 'author', 'short text description']}

**Table 1** Results of model predictions for sample dialogues. Seeker (S) and provider (P) roles are abbreviated for each numbered turn. Explicit (E), implicit (I), and clarification (C) grounding labels and shortened grounded knowledge are denoted as follows: prediction ( $= \oplus \neq$ ) ground-truth.

Dialogue B		
3 S: What is the dataset about in general?		
4 P: The dataset contains information about 98 nature parks in Germany. You can find in this dataset the name of the park, its year of establishment, its area etc.		
5 S: thanks, so if I understood correctly the dataset contains 3 columns, right? name of park, year, area	$E \neq C$	{'table content': 'information about 98 nature parks in Germany', 'column names': ['name of park', 'year', 'area']} ={ 'table content': 'nature parks in Germany', 'column names': ['park name', 'year', 'area']}
6 P: There are other attributes as well. Here are all the attributes: park name, the German state where the park is in, year of establishment, area in km2, and short text summary.		
7 S: great!	$E = E$	{'column names': ['park name', 'German state', 'year of establishment', 'area in km2', 'short text summary']}={ 'column names': ['park name', 'year', 'area', 'state', 'short text summary']}

**Table 1** Results of model predictions for sample dialogues. Seeker (S) and provider (P) roles are abbreviated for each numbered turn. Explicit (E), implicit (I), and clarification (C) grounding labels and shortened grounded knowledge are denoted as follows: prediction ( $= \oplus \neq$ ) ground-truth.

# Exploring a Japanese Cooking database

A robot uses GenAI and a knowledge graph  
to chat about culinary delights

19th Annual ACM/IEEE International Conference on Human Robot Interaction 2024



Kristiina Jokinen  
AI Research Center  
AIST Tokyo Waterfront



Graham Wilcock  
CDM Interact and  
University of Helsinki

# Contributions

- Build a knowledge graph in a Neo4j graph database from the existing open-source database (Kyoto culinary database)
- Enable interaction based on the KG and the latest advances in LLMs using GenAI
- Demonstrate a multilingual approach to developing applications by integrating modules and knowledge sources created in a different language than the application
- Support diversity by multilinguality in human-robot interaction



# KG construction: Kyoto culinary database

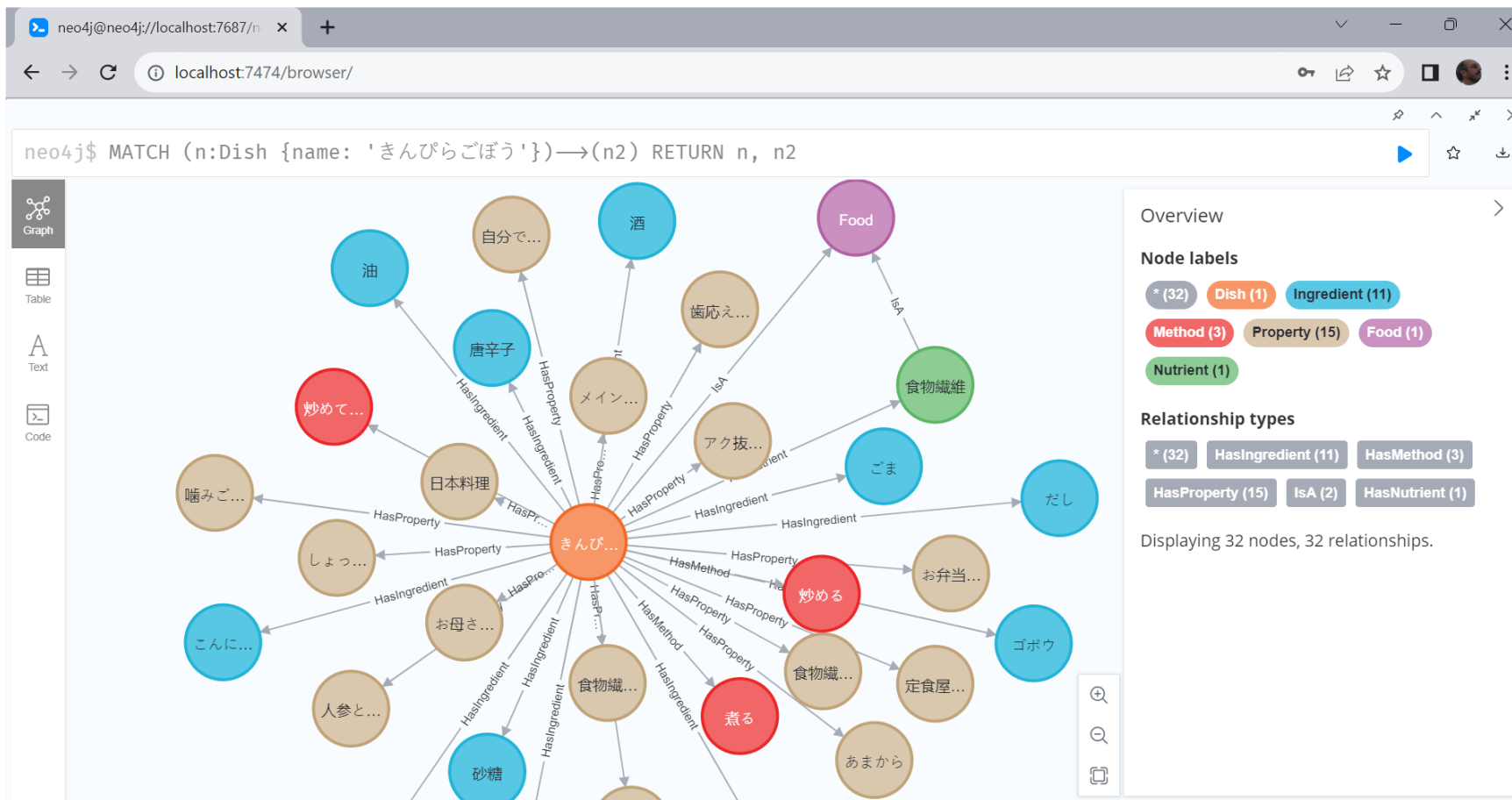
- Basic Cuisine Knowledge Base constructed at the Kyoto University in the joint project "Research on Knowledge Database Construction for Dialogue Processing" by the National Institute of Advanced Industrial Science and Technology (AIST), Kyoto University, and Panasonic Corporation.
- The knowledge base contains synonyms, ingredients, cooking methods, and attributes for approximately 400 basic dishes, selected from the "Cookpad Data", based on frequency and cooccurrence.
- Follows the notation of ConceptNet [26] except the relation types are different
- Attributes of the dishes include crowd-sourced surveys of cooking impressions, making the knowledgebase a realistic as well as locally and culturally reliable knowledge source for Japanese cuisine
- The database is in Japanese, and publically available.
- Supports multilingualism

# Attribute values and attributes for konpiragobou (braised burdock root) with relation confidence scores.

- [人参と一緒に/102/材料関連/0.5](#)
- [食物繊維が多い/103/栄養素/1.0](#)
- [食物繊維が豊富/103/栄養素/1.0](#)
- [日本料理/111/国・地域/0.5](#)
- [お弁当の脇役/121/料理ジャンル/1.0](#)
- [惣菜の鉄板/121/料理ジャンル/0.5](#)
- [定食屋で食べる/221/場所/0.5](#)
- [あまから/251/味/0.5](#)
- [しょっぱい味/251/味/0.5](#)
- [噛みごたえがある/252/食感/0.5](#)
- [歯応えがある/252/食感/0.5](#)
- [メインのおかずではない/261/印象/0.5](#)
- [お母さんの作ってくれるご飯/301/作り手/0.5](#)
- [自分では作らない/301/作り手/0.5](#)
- [アク抜きが必要/311/調理法・工程/0.5](#)
- [With carrots/102/Material related/0.5](#)
- [High in dietary fiber/103/Nutrients/1.0](#)
- [Rich in dietary fiber/103/Nutrients/1.0](#)
- [Japanese cuisine/111/Country/Region/0.5](#)
- [Side dish for bento/121/Cooking genre/1.0](#)
- [Side dish for teppan-yaki/121/Cooking genre/0.5](#)
- [Eating at a set meal restaurant/221/Place/0.5](#)
- [Sweet and salty/251/Taste/0.5](#)
- [Salty taste/251/Taste/0.5](#)
- [Chewy/252/Texture/0.5](#)
- [Chewy, tough/252/Texture/0.5](#)
- [Not the main side dish/261/Impression/0.5](#)
- [Food cooked by mother/301/Creator/0.5](#)
- [I don't make it myself/301/Maker/0.5](#)
- [Requires removal of scum/311/Cooking method/Process/0.5](#)



# KG construction: *kinpira-kobou* in the Neo4j knowledge graph (braised burdock root)



# LangChain Architecture for Neo4j Knowledge Graph

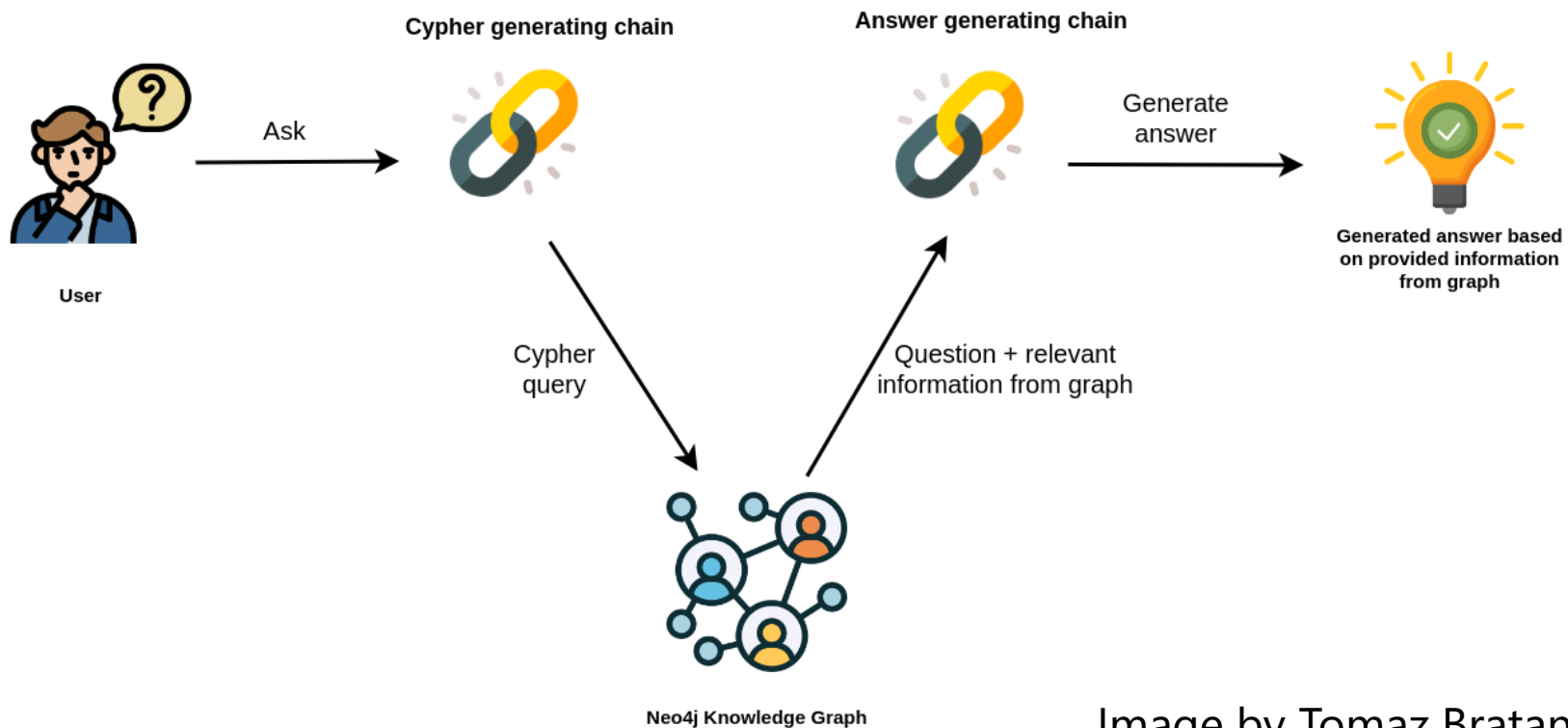


Image by Tomaz Bratanic 2023,  
from *neo4j\_cypher*, LangChain Templates

# Tell me six different Japanese dishes

```
> Entering new GraphCypherQAChain chain...
```

```
Generated Cypher:
```

```
  MATCH (d:Dish {region: '日本'})  
  RETURN d.name AS dish  
  LIMIT 6
```

```
Full Context:
```

```
[{'dish': 'あさりの酒蒸し'}, {'dish': 'あんぱん'}, {'dish': 'いちご大福'}, {'dish': 'いなり寿司'}, {'dish': 'おこわ'},  
{'dish': 'おじゃ'}]
```

```
> Finished chain.
```

```
Response: Sure, here are six different Japanese dishes you might enjoy: あさりの酒蒸し (Asari no sake-steamed), あんぱん (Anpan), いちご大福 (Ichigo daifuku), いなり寿司 (Inari sushi), おこわ (Okowa), おじゃ (Ojya).
```

# What cooking method is used for Okonomiyaki?

> Entering new GraphCypherQAChain chain...

Generated Cypher:

```
MATCH (d:Dish {name: 'お好み焼き'})-[:HasMethod]->(m:Method)
RETURN m.name AS method
```

Full Context:

```
[{'method': '炒める'}, {'method': '揚げる'}, {'method': 'おろす'}, {'method': '焼く'}]
```

> Finished chain.

Response: Sure, here's a friendly response:

炒める、揚げる、おろす、焼く! That's right, Okonomiyaki is made using a combination of these four cooking methods! 🔍🍳🍴🔥

# Demo

H: Tell me six different Japanese dishes.

R: Sure, here are six different Japanese dishes you might enjoy: Asari no sake-steamed, Anpan, Ichigo Daifuku, Inari sushi, Okowa (glutinous rice mixed with meat or vegetables), Ojiya (rice soup)

H: What ingredients are used to make Okonomiyaki?

R: The following are some of the main ingredients used to make Okonomiyaki is made with a delicious blend of ingredients, including soy sauce, cornstarch, cabbage, mountain yam, flour,...

H: What cooking method is used for Okonomiyaki?

R: itameru, ageru, orosu, yaku. That's right! Okonomiyaki is made using a combination of these four cooking methods: stir-fry, deep-fry, grating of vegetables, baking.



# Random Thoughts

- Different types of knowledge source => Information needs to be validated
- New risks with Language-Capable Robots
  - Identity, privacy, trust and influence  
*Tom Williams et al.: Voice in the Machine: Ethical Considerations for Language-Capable Robots. CACM 2023/8*
- Sustainable multimodal language processing (Bender et al. 2021)
  - Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? FAccT '21, pages 610–623, New York, USA. Association for Computing Machinery.
  - Powerful laptops which run the model
- Use for grounding and building common ground in interaction research
  - Conversational grounding (see our poster),
  - Grounding to the real-world sensor information

# Are we not yet there?

Extensive research on LLM properties, techniques etc.

1. Reliability of the LLM assistant's answers in a given domain
  - Open-source models, RAG, KGs, transparent operation
2. Grounding of responses in real-world representation,
  - Not just “stochastic parrots”
3. Interactive learning and adversarial learning to control the LLM agent's behaviour,
  - Frameworks that better match real-world use cases
4. Evaluation of the assistant's responses, supporting ethical and sustainable practices in building and using the assistant
  - Understanding and systematic testing of the model capabilities, limitations, potential misuse



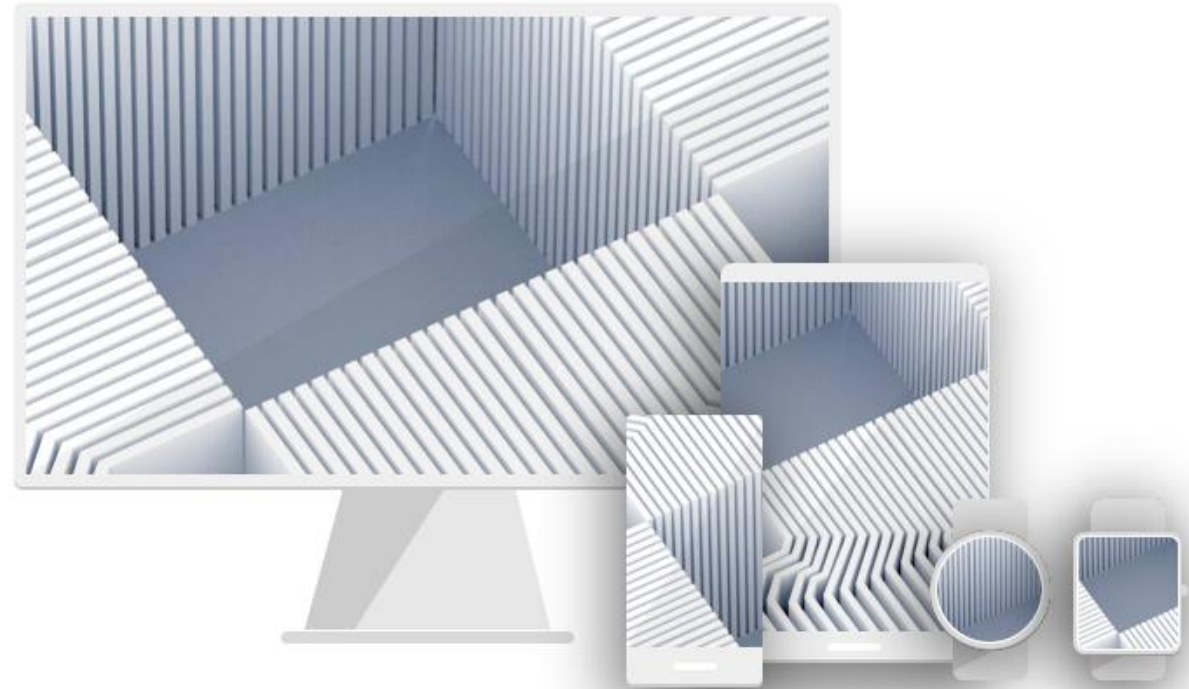


Thank you!

More information:

[kristiina.jokinen@aist.go.jp](mailto:kristiina.jokinen@aist.go.jp)

twitter: @pkjokinen





## Towards Hybrid Reasoning: Assimilating Structure into Subsymbolic Systems

<https://medium.com/@alcarazanthony1/towards-hybrid-reasoning-assimilating-structure-into-subsymbolic-systems-05cf9d34d13d?sk=aed32393c790b67cf14b6e090876406>



# Overview:

Recent advances in large language models (LLMs) show **impressive fluency** and **adaptability**

But LLMs struggle with deeper reasoning requiring:

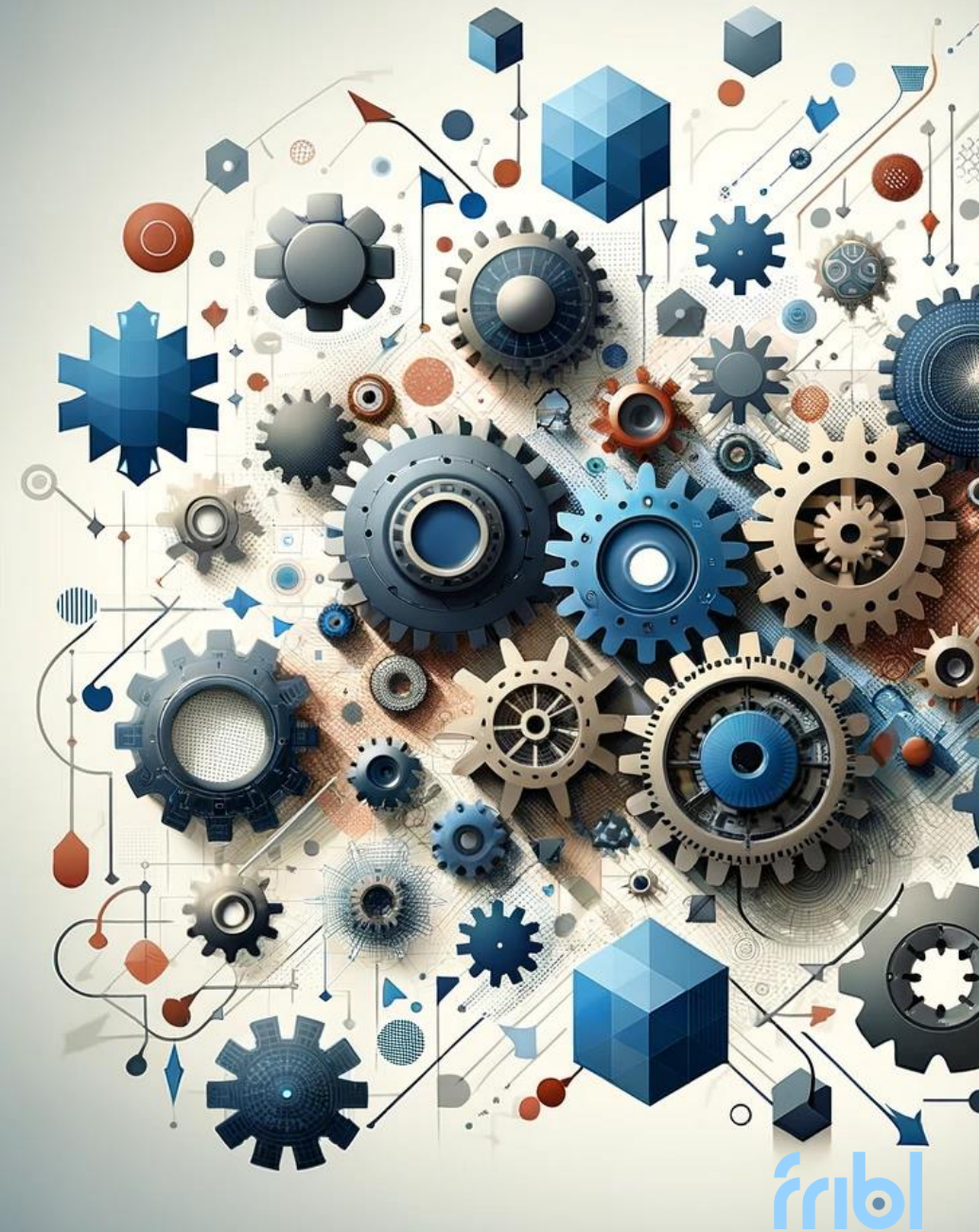
- **Compositional generalization**
- **Sustained causal chains**
- **Creatively hypothesizing mechanisms**

Knowledge graphs provide **structured representations** to address these gaps

However, knowledge graphs have challenges with:

- **Scale**
- **Noise**
- **Incompleteness**
- **Sparsity**

Proposes a **coordinated approach** leveraging strengths of both representations





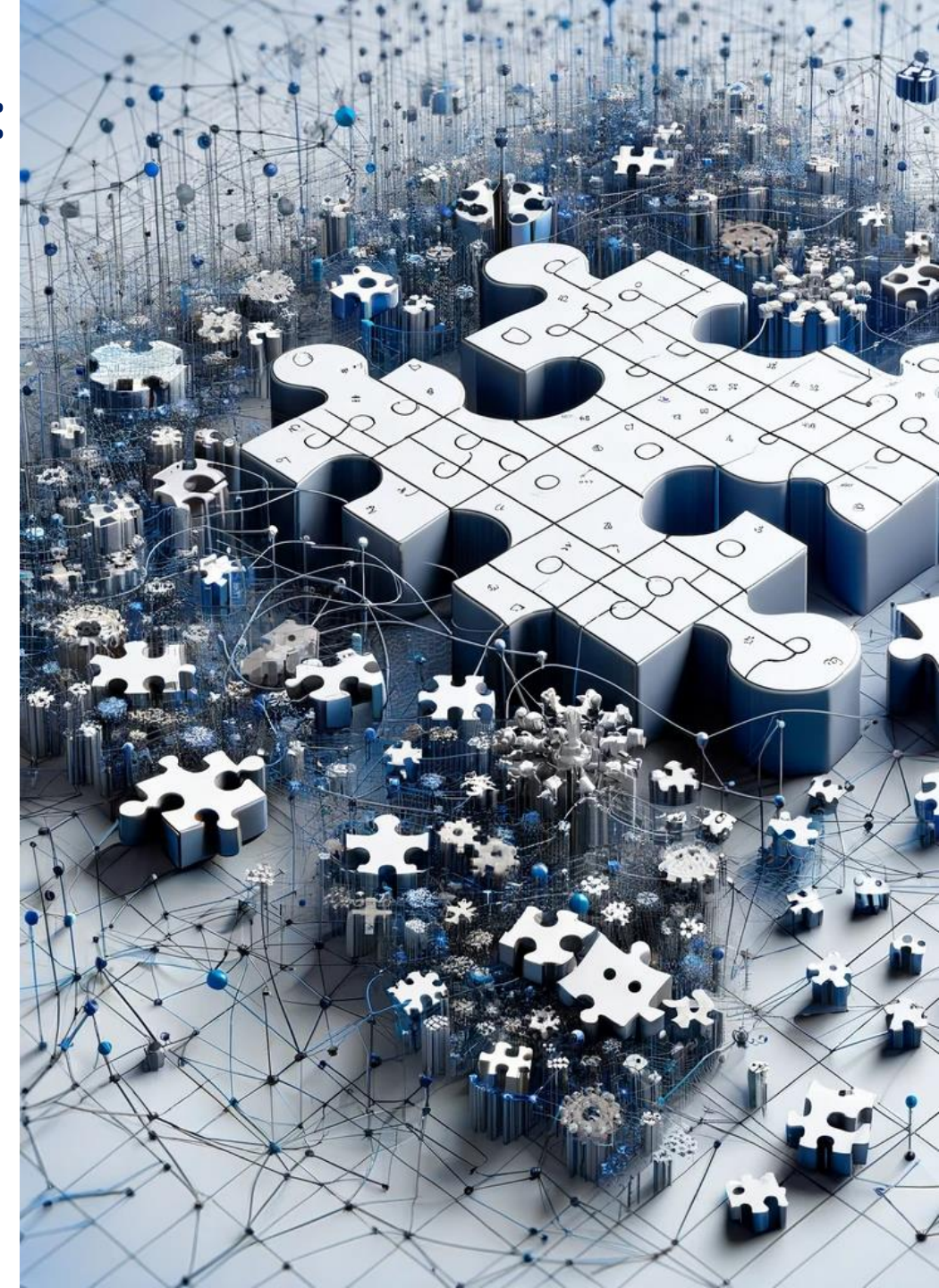
# Compositionality Challenges for LLMs:

**Brittle combination** of modular causal knowledge fragments

**Cannot reliably track intermediate conclusions** across long causal chains

**Struggle to smoothly transition** between interconnected causal chains

**Lack capacities for actively simulating and testing causal hypotheses**



## LANGUAGE MODEL AGENTS SUFFER FROM COMPOSITIONAL GENERALIZATION IN WEB AUTOMATION

Hiroki Furuta<sup>1,2\*</sup> Yutaka Matsuo<sup>2</sup> Aleksandra Faust<sup>1</sup> Izzeddin Gur<sup>1</sup>  
<sup>1</sup>Google DeepMind <sup>2</sup>The University of Tokyo  
furuta@eblab.t.u-tokyo.ac.jp

### ABSTRACT

Language model agents (LMA) recently emerged as a promising paradigm on multi-step decision making tasks, often outperforming humans and other reinforcement learning agents. Despite the promise, their performance on real-world applications that often involve combinations of tasks is still underexplored. In this work, we introduce a new benchmark, called *CompWoB* = 50 new compositional web automation tasks reflecting more realistic assumptions. We show that while existing prompted LMAs (gpt-3.5-turbo or gpt-4) achieve 94.0% average success rate on base tasks, their performance degrades to 24.9% success rate on compositional tasks. On the other hand, transferred LMAs (finetuned only on base tasks) show less generalization gap, dropping from 85.4% to 54.8%. By balancing data distribution across tasks, we train a new model, *HTML-T5++*, that surpasses human-level performance (95.2%) on *MiniWoB*, and achieves the best zero-shot performance on *CompWoB* (61.5%). While these highlight the promise of small-scale finetuned and transferred models for compositional generalization, their performance further degrades under different instruction compositions changing combinational order. In contrast to the recent remarkable success of LMA, our benchmark and detailed analysis emphasize the necessity of building LMAs that are robust and generalizable to task compositionality for real-world deployment.

### 1 INTRODUCTION

Based on the exceptional capability of large language models (LLMs) (OpenAI, 2023; Anil et al., 2023; Touvron et al., 2023) in commonsense understanding (Boswin et al., 2020; Chowdhury et al., 2022), multi-step reasoning (Wei et al., 2022; Kojima et al., 2022), program synthesis (Chen et al., 2021) and self-improvement (Shim et al., 2023; Madan et al., 2023; To et al., 2023), language model agents (LMA) have recently emerged to tackle various decision making problems, such as robotics (Huang et al., 2022a; Ahn et al., 2022), information retrieval (Nakano et al., 2021; Yao et al., 2022b), and external tool use (Wu et al., 2023; Shinn et al., 2023; Lu et al., 2023). Especially, in web automation (Shi et al., 2017), LMAs with prompting (Kim et al., 2023; Sun et al., 2023; Zheng et al., 2023) outperform humans and other learning-based agents, such as reinforcement learning (Lumppley et al., 2022) or finetuned language models (Gur et al., 2022; Furuta et al., 2023). Despite their proficiency in *MiniWoB* (Shi et al., 2017), a standard web automation benchmark, it is still unclear whether LMAs could deal with challenges in the real world, such as complex observation (Gur et al., 2023), domain generalization (Deng et al., 2023), and ambiguity of instructions (Zhu et al., 2023b). These challenges are exacerbated due to the open-ended nature of real-world tasks, making it infeasible to prepare exemplars and prompts in advance for any unseen task.

In this work, we extensively study the generalization of LMAs to more realistic task compositions. We first design a new controlled test bed, called *CompWoB*, with 50 compositional tasks by combining a set of base tasks in a single-page or multi-page environment with instructions linked together using simple connectors such as “and then”. Only providing the knowledge about base tasks, we investigate the generalization performance of existing SoTA prompted LMAs (Kim et al., 2023; Sun et al., 2023;

## Causal Reasoning and Large Language Models: Opening a New Frontier for Causality

Emre Kiciman<sup>\*</sup> Robert Ness  
Microsoft Research Microsoft Research  
emrek@microsoft.com robertness@microsoft.com  
Amit Sharma Chenhao Tan  
Microsoft Research University of Chicago  
amshar@microsoft.com chenhao@uchicago.edu

Working Paper May 9, 2023

### Abstract

The causal capabilities of large language models (LLMs) is a matter of significant debate, with critical implications for the use of LLMs in societally impactful domains such as medicine, science, law, and policy. We further our understanding of LLMs and their causal implications, considering the distinctions between different types of causal reasoning tasks, as well as the entangled threats of construct and measurement validity. We find that LLM-based methods establish new state-of-the-art accuracy on multiple causal benchmarks. Algorithms based on GPT-3.5 and 4 outperform existing algorithms on a pairwise causal discovery task (97%, 13 points gain), counterfactual reasoning task (92%, 20 points gain) and actual causality (86% accuracy in determining necessary and sufficient causes in vignettes). At the same time, LLMs exhibit unpredictable failure modes and we provide some techniques to interpret their robustness.

Crucially, LLMs perform these causal tasks while relying on sources of knowledge and methods distinct from and complementary to non-LLM based approaches. Specifically, LLMs bring capabilities so far understood to be restricted to humans, such as using collected knowledge to generate causal graphs or identifying background causal context from natural language. We envision LMAs to be used alongside existing causal methods, as a proxy for human domain knowledge and to reduce human effort in setting up a causal analysis, one of the biggest impediments to the widespread adoption of causal methods. We also see existing causal methods as promising tools for LLMs to formalize, validate, and communicate their reasoning.

\* Authors listed alphabetically, all contributed equally

especially in high-stakes scenarios.

Our experiments do not imply that complex causal reasoning has spontaneously emerged in LLMs. However, in capturing common sense and domain knowledge about causal mechanisms and supporting translation between natural language and formal methods, LLMs open new frontiers for advancing the research, practice, and adoption of causality.

### 1 Introduction

Recent advances in scaling large language models (LLMs) have led to breakthroughs in AI capabilities. As language models increase in number of parameters and are trained on larger datasets, they gain complex, emergent behaviors, such as abilities to write code in programming languages, generate stories, poems, essays, and other texts, and demonstrate strong performance in certain reasoning tasks (Chen et al., 2021; Nguyen & Nadi, 2022; Bubeck et al., 2023; Katz et al., 2023; Wei et al., 2022a). Impressively, when asked to explain their outputs, update their conclusions given new evidence, and even generate counterfactuals, LLMs can create plausible responses (Nori et al., 2023; Lee et al., 2023a,b). This apparent capacity for both implicit and explicit consideration of causal factors has generated excitement towards understanding their reasoning capabilities (Hobbbahn et al., 2022; Kosoy et al., 2022; Willig et al., 2022; Liu et al., 2023; Zhang et al., 2023). Figure 2(a) shows an example of such reasoning.

At the same time, LLMs are imperfect: they can make absurd claims and are often observed to make basic errors of logic and mathematics, much less complex rea-



# Knowledge Graphs vs. Vector Search

Model **richer semantic relationships**

- **Taxonomic, logical, procedural, etc.**
- **Beyond just similarity scores**

Enable **explainable inference trails**

**Trace paths over entities and relations**

**Understand reasoning process**

Provide **modular, structure-learnable components**

Custom **subgraphs** with unique **constraints**

- **Add new facts and ontologies**
- Allow **focused exploration**

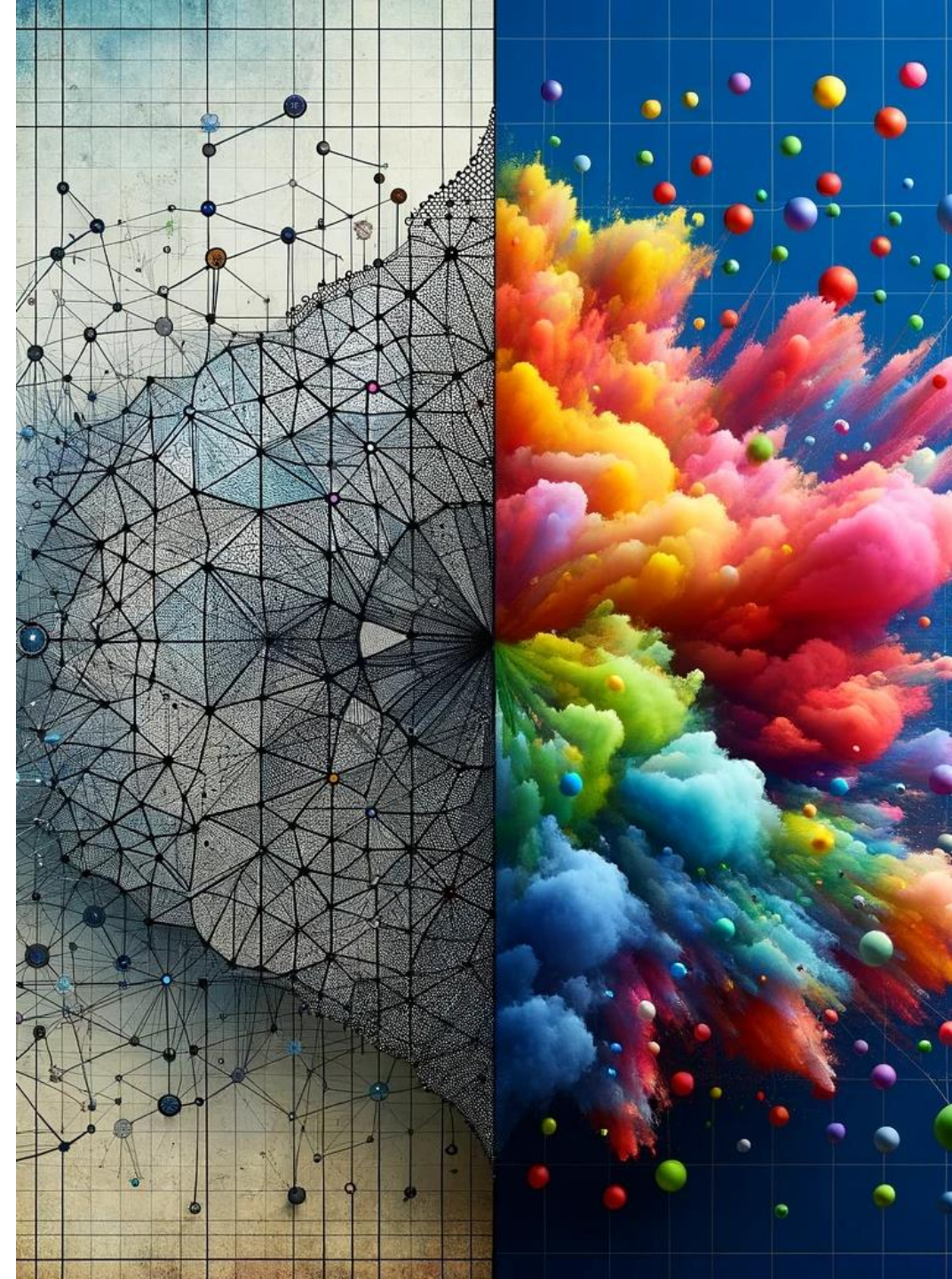
**Directly retrieve interconnected content**

Avoid drifting to tangentially related info

Empower more systematic reasoning

**Leverage validated connections**

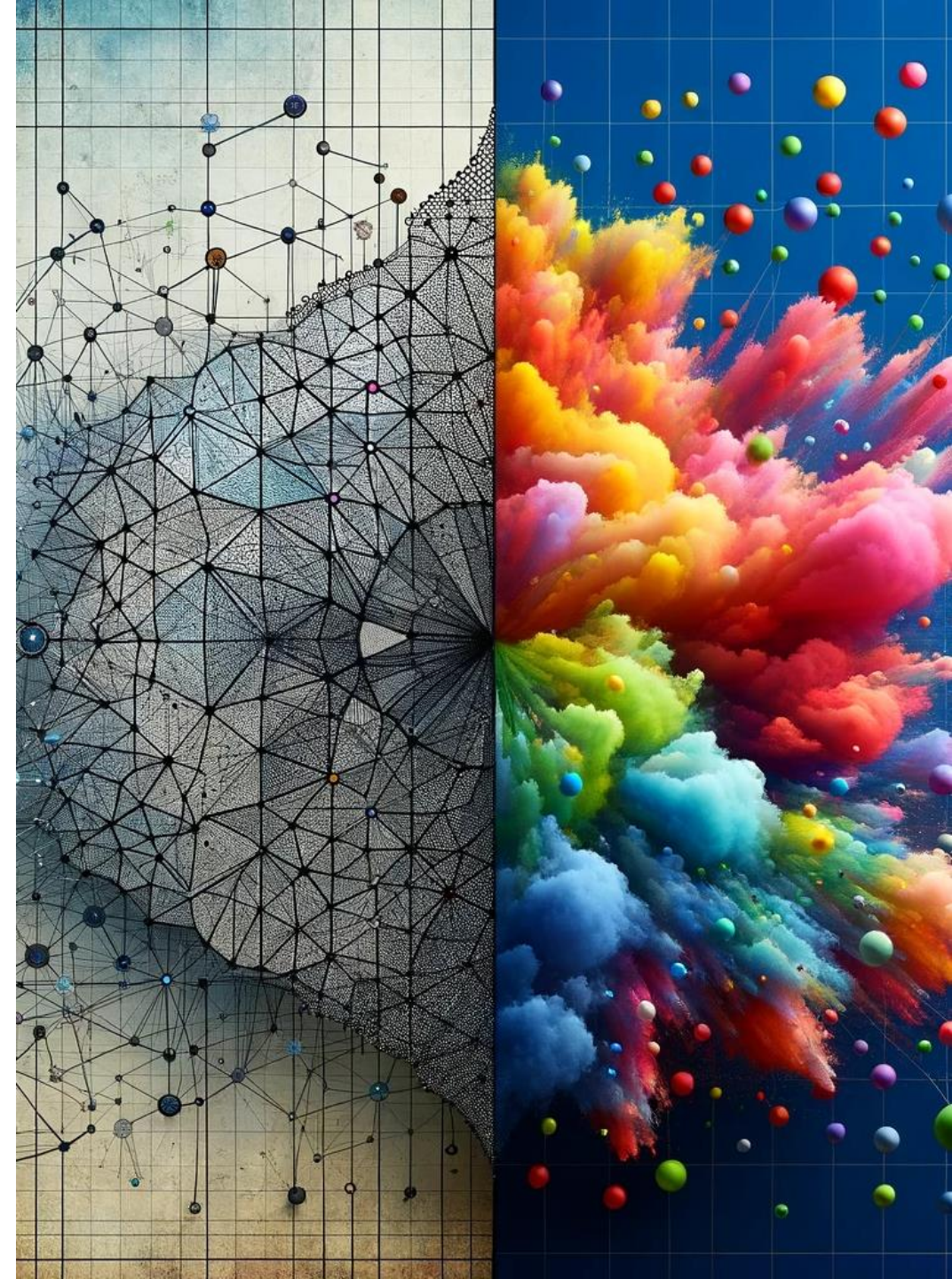
**Qualify facts with metadata like time, location, etc.**





# Knowledge Graphs vs. Vector Search

	<b>Vector Search RAG</b>	<b>Knowledge Graph RAG</b>
Relationships	Passages linked by semantic vector similarity scores	Diverse relationships between entities - taxonomic, logical, temporal
Reasoning Style	Similarity-chain based	Multi-hop inference over graph schema
Inference Trail	Opaque neural projections	Explicit interpretation over graph paths
Exploration Dynamics	Potential semantic drift	Focused traversal anchored to key entities
Modularity	Lack native support	Custom subgraphs with unique constraints
Evolution	Requires external model changes	Continuous structure learning from data
Trustworthiness	Questionable relevance signals	Validated relations and explainable trails
Limitations	Precision capped by loose implicit associations	Pragmatic balance between depth and scale





# Challenges of Complex Knowledge Graphs:

## Massive Scale

- Billions of facts creates computational bottlenecks
- Exponential complexity for algorithms

## Noise

- Inaccurate facts from information extraction
- Propagates to degrade query responses

## Incompleteness

- Gaps relative to full scope of world knowledge
- Important concepts and relations missed

## Sparsity

- Power law distribution of connections
- Islands of facts with minimal links
- Hampers lookup and inference

## Difficulty of Query Formulation

- Mapping questions to formal query languages challenging
- Requires understanding precise semantics
- Steep learning curve for domain experts





# The Gates

## Cypher Queries

- Formulate precise graph pattern matching queries in Cypher to extract entities and relationships
- Requires expertise in query language to translate information needs
- Retrieves subgraphs that can provide contextual facts to guide LLM

## Vector Similarity Search

- Encode knowledge graph contents into embeddings vector space
- Allows approximate semantic search for relevant entities/relations instead of keywords
- Blazing fast indexed retrieval to contextualize language generation





# The Gates

## Graph Algorithms

- Graph algorithms equip language models with topological knowledge about explanatory reasoning chains, influential entities, contextual modularity, and similarity embeddings
- Elevating inference through structural perspective beyond individual facts.

## Generative Knowledge Graphs

- Transform symbolic graphs into continuous probability distributions
- Allows sampling plausible new triples and uncertainty modeling
- Compatible with language model generation for grounding
- Handles noise and missing facts via joint distributions
- Constraint-aware generation respecting ontology
- Augmentation by extracting relations from text





# Proposed Orchestration Workflow:

## Iterative Analysis

- Comprehend reasoning needs
- Identify key entities and relationships
- Deconstruct question into information needs

## Modularization

- Encapsulate targeted search operations
- Create reusable reasoning components
- Define interfaces for interoperability

## Parallel Evidence Retrieval

- Configure & launch concurrent query tools
- Rapidly focus on relevant regions
- Continual optimization based on signals

## Propagate Intermediate Results

- Directly populate centralized state store
- Resolve co-references across retrieved content





# Proposed Orchestration Workflow:

## Iterative Analysis

- Comprehend reasoning needs
- Identify key entities and relationships
- Deconstruct question into information needs

## Modularization

- Encapsulate targeted search operations
- Create reusable reasoning components
- Define interfaces for interoperability

## Parallel Evidence Retrieval

- Configure & launch concurrent query tools
- Rapidly focus on relevant regions
- Continual optimization based on signals

## Propagate Intermediate Results

- Directly populate centralized state store
- Resolve co-references across retrieved content

<https://towardsdatascience.com/achieving-structured-reasoning-with-llms-in-chaotic-contexts-with-thread-of-thought-prompting-and-a4b8018b619a?sk=5d0c86d418b35886138edfc586809e30>





# Proposed Orchestration Workflow:

## Recursive Re-planning

- Re-evaluate open needs based on evidence
- Dynamically launch additional queries
- Track progress towards completeness

## Assimilation by Language Models

- Batch updated state digest for ingestion
- Disambiguate and reconcile evidence
- Highlight speculative interpretations

## Evaluation & Explanation

- Assess alignment with original query
- Construct response elucidating reasoning
- Expose key graph traversal paths

## Leverage Asynchrony and Concurrency

- Concurrent operations reduce waiting time
- Parallelism increases computational efficiency
- Accelerate overall workflow

## Strategies for Symbolic/Subsymbolic Blending:

- Joint vector embeddings
- Inject symbolic graph schemas
- Differentiable graph programming



# THANK YOU!

Synchroteam  
by nomadia

François Pichon, Co-Founder

*“As a SMB operating in Europe and USA, it is imperative for us to optimize our processes and avoid costly errors. Our conventional recruitment procedure typically spans several weeks to finalize candidate preselection and confirm the hiring decision. The integration of Fribl has transformed our approach to talent acquisition, introducing a level of efficiency and cost-effectiveness that was previously unattainable. What used to be a time-intensive process of candidate selection now unfolds within mere minutes. This streamlined efficiency allows us to redirect our efforts towards cultivating meaningful connections with our chosen candidates. The rapidity and precision afforded by GenAI have significantly enhanced our recruitment strategy, reaching unprecedented levels of seamlessness and satisfaction. It stands as a pivotal advancement in the realm of talent acquisition, serving as a true game-changer for our organisation.”*

Anthony ALCARAZ  
Chief AI Officer  
anthony@fribl.co  
M- +33 641860945





Appotek

CimateGPT



# AppTek Company Overview

## Data

AppTek's data packs and services are highly valued for training AI and machine learning models

**250K**

Transcribed audio hours

**1.5M**

Audio hours for unsupervised training

**60+**

Languages with 100s of dialects

## AI Models

Advanced engines for automatic speech recognition, machine translation, natural language understanding / processing, and text to speech

**ASR**

Automatic transcription of broadcast, media and entertainment, microphone and telephony in 60+ languages

**MT**

Utilizes software to translate text or speech into different languages, featuring 600+ language pairs

**NLU/P**

Context from ASR used to discern meaning and execute an intent from voice commands

**TTS**

Reading out text in human-like, expressive and adapted voices

## The People

Highly experienced team of AI scientists and engineers, providing word-class expertise to help customers refine their licensed or in-house AI models

**~60**

Scientists

**32**

PhDs

**~20**

Research engineers

**100s**

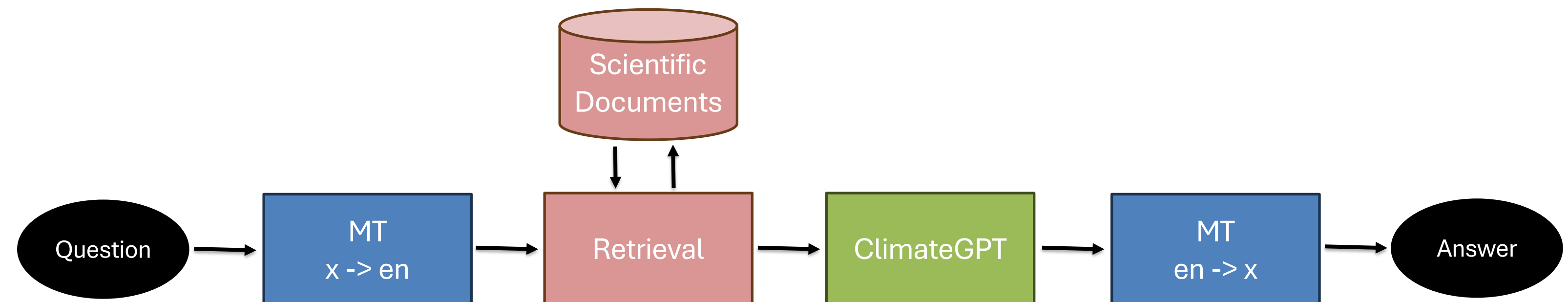
Peer-reviewed papers

**9**

Patents

# ClimateGPT

- Developed and fine-tuned a generative LLM model to improve fluency of scientific climate change output
- 3 dimensions/perspectives: Natural Science, Economics, and Sociology
- Baselines are Llama2-7B, Llama2-13B and Llama2-70b trained on 2T tokens
- Continuous pre-training on 4.2B tokens climate-related text
- Instruction Fine Tuning augmented with climate-scientist curated data (10k demonstration pairs)
- Hierarchical retrieval augmentation
- Multilinguality through cascaded system



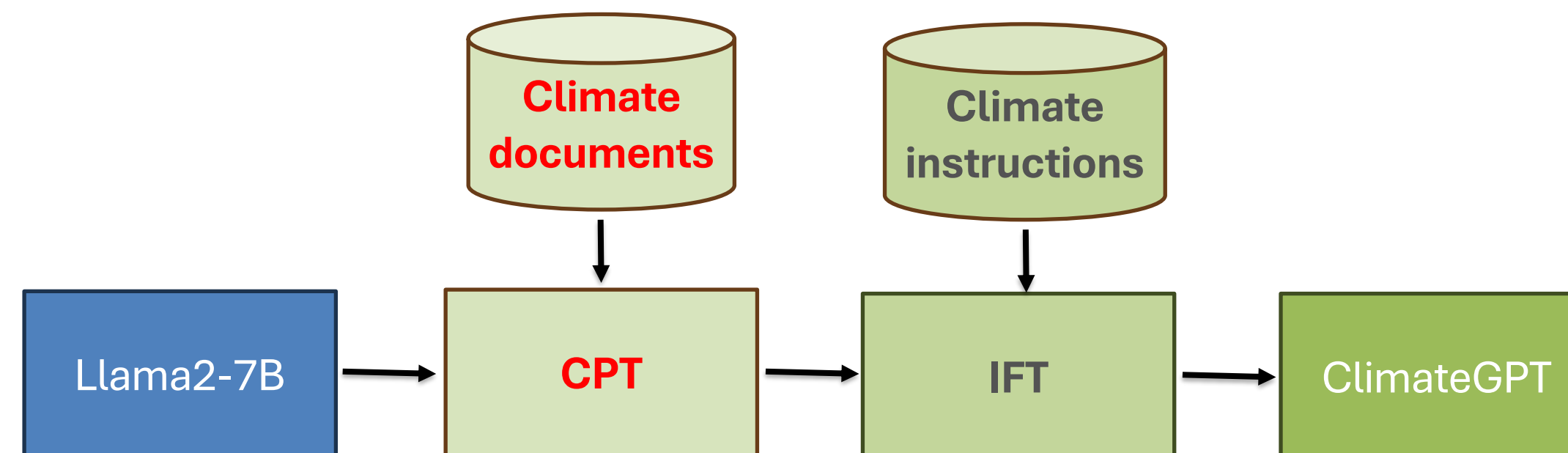


# ClimateGPT

Training

CPT + IFT

- Continuous pre-training on 4.2B climate-related text
  - Extreme Weather reports (10 years \* 1M articles)
  - Technical Game-Changing Breakthroughs (153 themes x.1000 articles)
  - Selection through Sustainable Development Goals (17 SDGs)
  - Climate Change News
  - Climate Change reports
    - World Bank, OECD, IPCC, UN, EU, TFCD, US, NASA, ESA, WRI, IREA, WEF, Nature Finance
  - Climate Academic Research





# ClimateGPT

Training

CPT + IFT

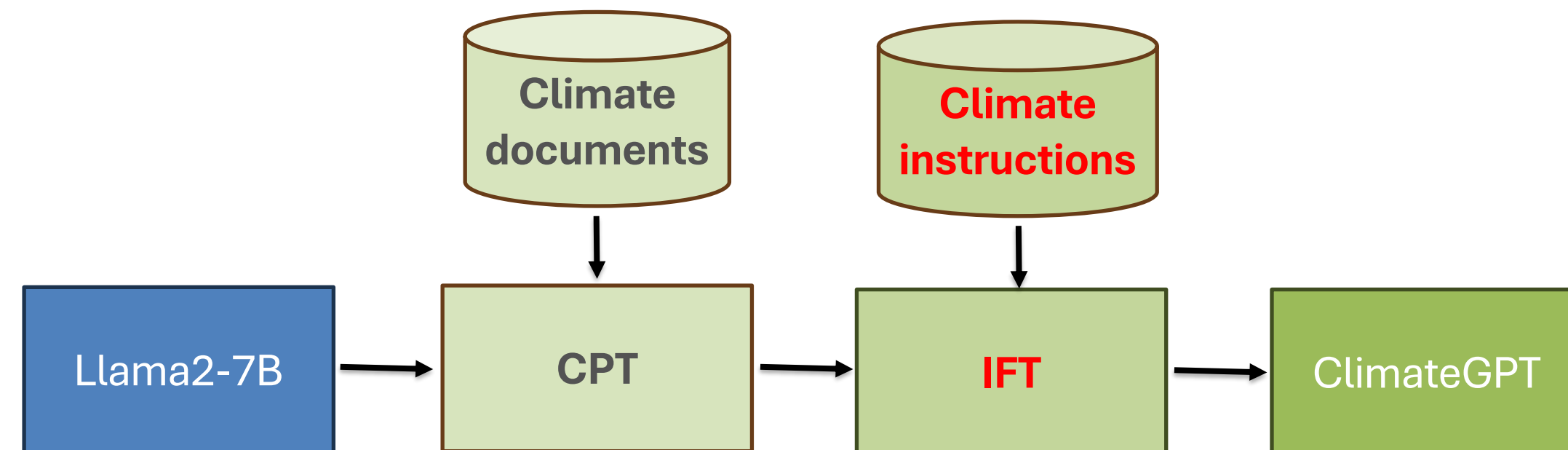
- Instruction Fine Tuning augmented with climate-scientist curated data (10k demonstration pairs)

Domain	Name	Total Size	Training Samples
Climate	Senior Expert Interviews	74	1,332
	Grounded Expert Demonstration	403	7,254
	Grounded Non-Expert Demonstrations	9,663	146,871
	Synthetically Generated Demonstrations	57,609	0
	Climate-dimension specific StackExchange	3,282	9,846
General	AppTek General	700	2,100
	OASST-1	3,783	11,349
	Dolly	15,001	45,003
	Llama-2 Safety	939	2,817
	FLAN	38,909	30,000
	CoT	448,439	15,000

60.8%

39.2%

271,572 demonstration pairs

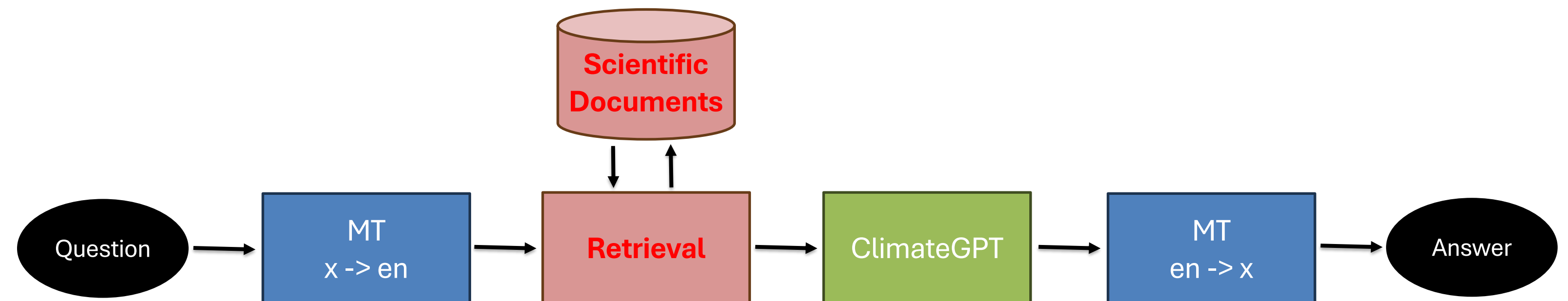


# ClimateGPT

## Inference time

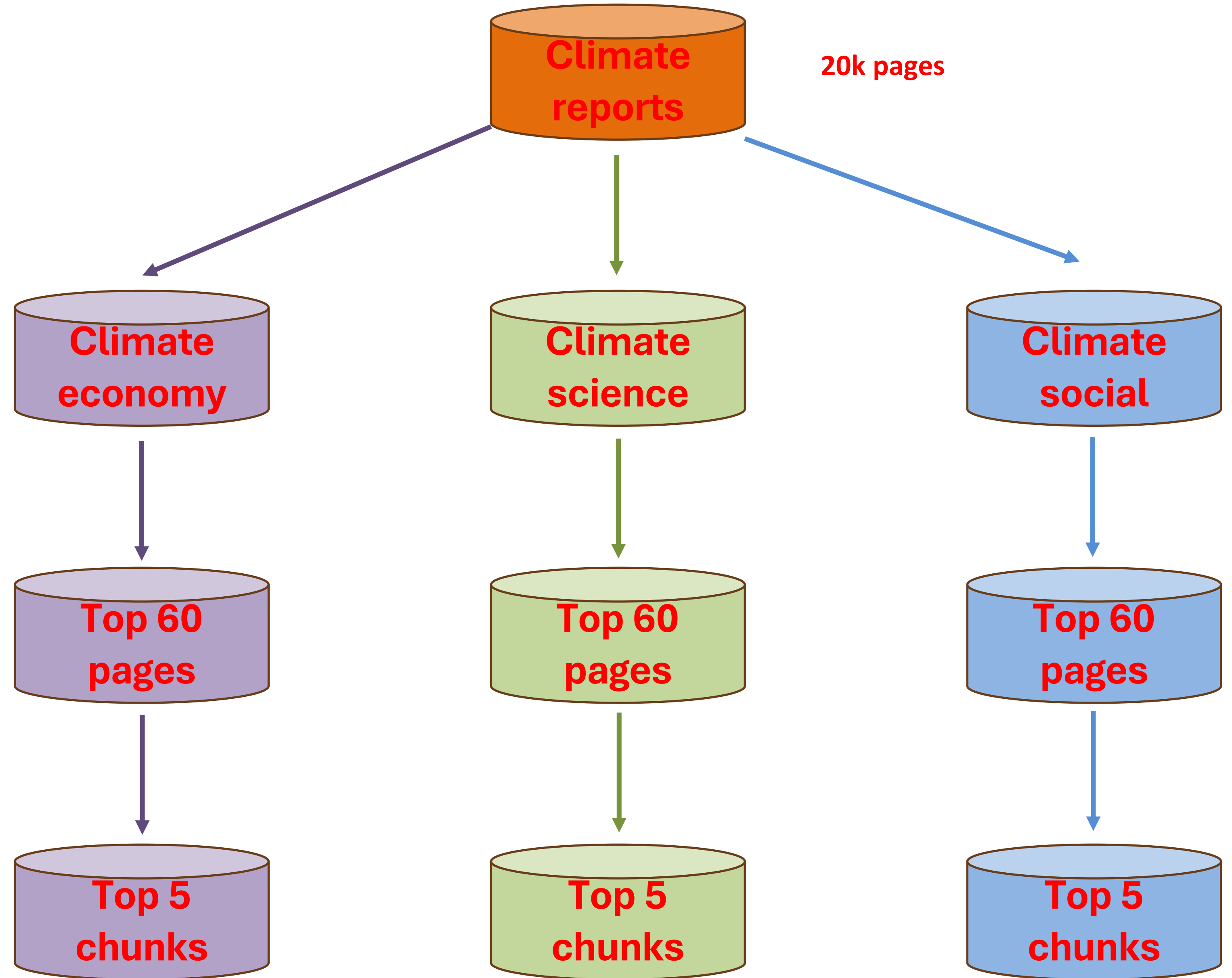
## RAG

- 700 documents (IPCC\* reports + academic papers cleaned from tables and references)
  - 20k pages
  - GPT-3.5 tagged along 3 dimensions (economy, social, science)
  - Vector search (transformer bi-encoder)
  - Hierarchical retrieval
    - Page level search (top 60)
    - Chunks of 115 tokens per page (top 5)
    - Citations provided through selected chunk
- ➔  $5 * 115 + \text{meta-data} == 154$  tokens added per dimension





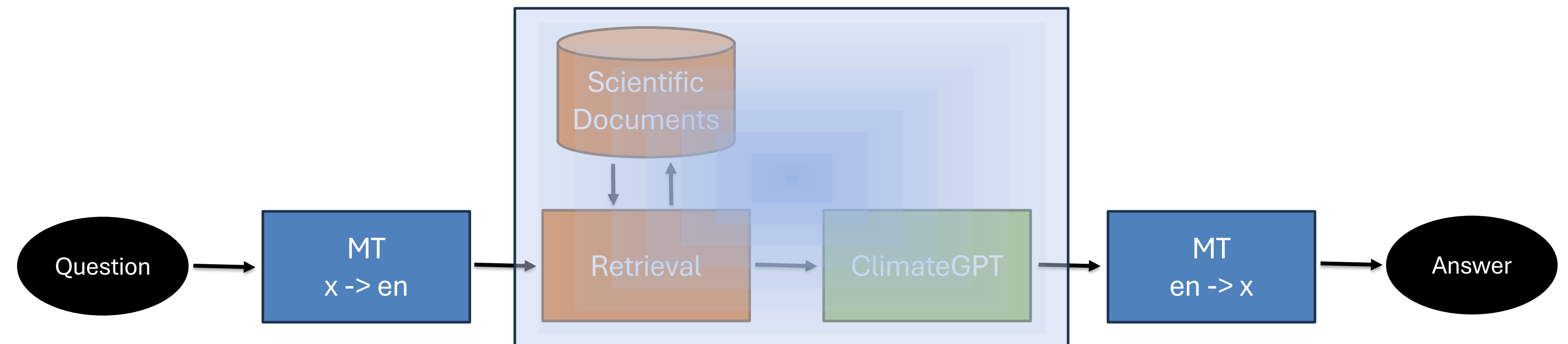
# Hierarchical RAG



# ClimateGPT

Inference time  
MT

- Multilinguality through cascaded system
  - No truly multilingual open source LLM available
  - Allows to keep compactness and LLM model precision
  - Answer quality for low resourced languages (science)
  - May not be adapted to culture





# ClimateGPT

## Results

- Evaluated on
  - standard language comprehension tasks
  - climate related comprehension tasks
- ClimateGPT-7B models equals performance of Llama2-70B on climate tasks
  - 10 times smaller
  - 12 time less energy needed at inference time
- Incremental training at a tiny fraction of the cost needed to train the base model
- Multilinguality addressed with a cascaded approach



# Appotek

**RAG: vector search quality**



# KGs: Pros and Cons



## PROS

- ✓ Manual KGs are factual
- ✓ Contains explicit alternatives / complementarity / inconsistencies
- ✓ Allows reasoning
- ✓ Does not always have an answer



## CONS

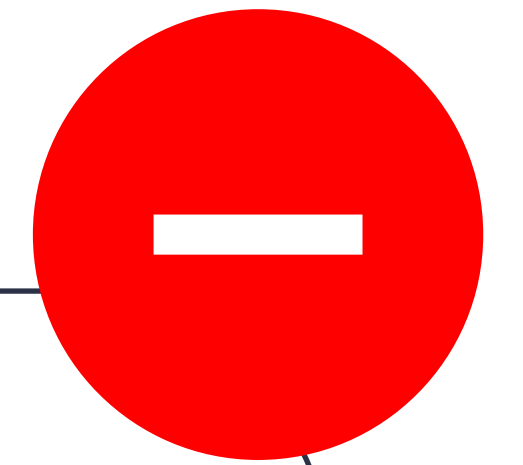
- ✗ Relations are based on hard-coded ontologies
- ✗ Intensive manual work for high quality
- ✗ To be efficient, KG expansion is task dependent
- ✗ Precision impacts flexibility

# LLMs: Pros and Cons



## PROS

- ✓ Based on data
- ✓ Automatic
- ✓ Task/Domain independent



## CONS

- ✗ Hallucinates
- ✗ One answer per perspective
- ✗ No abstraction: No reasoning structure
- ✗ Always has an answer



# KGs to Improve RAG\*

Document-based KG generation has good results when intention/goal is known

- Given a question to the LLM (Q-Intention + Q-Entities)
- Given a set of documents used as a priori knowledge indexed on D-Entities
- Select subset of documents based on Q-Entities
- Apply Q-Intention Recognition on the subset of documents
- Extract document snippets with ranked Q-Intention
- Provide LLM with question + snippets
- Build a (dynamic) KG from subset
- Get KG-facts related by Q-Intention
- Provide LLM with question, KG facts and document snippet related to KG facts

\*work in progress

Reduce snippet vector search to intention recognition  
and  
keep entities (abstraction and resolved value) as hard as possible



Appotek

LLM

Thank You!

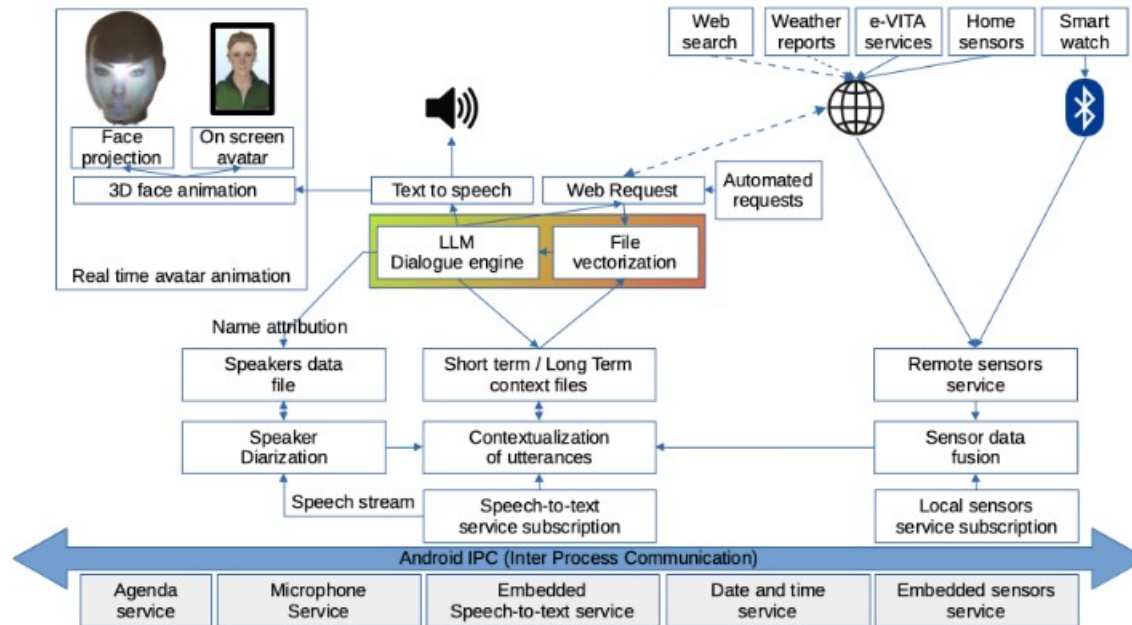


# Lifeline on a phone in e-ViTA

Hugues Sansen

# Initial objective

## e-Vita embedded on a phone



# Disappointments

- The LLM we installed on a Pixel6 was too slow for a realistic dialogue
  - Expected <400ms (equivalent to telephony with geostationary satellite)
  - A jitter is detected when over 200ms and becomes uncomfortable
  - Reality > 1mn
- Too “generative” to be usable
  - Funny answers on “What is it like to be a bat?” (Thomas Nagel)
- This was before Google’s Gemini on Pixel8, that we will use in a short future.
- => **we revived the Lifeline project**

# Lifeline

- Is a graph that represents what a user can tell about her life
- Is built from a dialogue with the user according to the graph theory
- Milestones are temporal vertices.
- The knowledge graph is built through the dialogue. It reflects who the user is and what she knows or believes.
- It can be seen as a ghost writer that will write the bio with a LLM from the knowledge extracted from the graph.

# What we developed

- A graph DB, with small initial knowledge (the 5 countries of the e-ViTA project as a graph)
- A rule based dialogue
  - technically simple if not naive, efficient, precise, relatively well suited for graph construction, but tedious, and incapable of detecting speech recognition errors
  - Spoken text is localized for easy translation,
  - Large use of localized Regular expressions and distance measure: the speech recognizer is not reliable.
  - Unfortunately, we receive text generated by a speech recognizer, not by a keyboard thus available text distance measurement APIs are of little value if we do not know the words used by the speaker:
    - “the wolves are made of stones” instead of “the walls are made of stones”

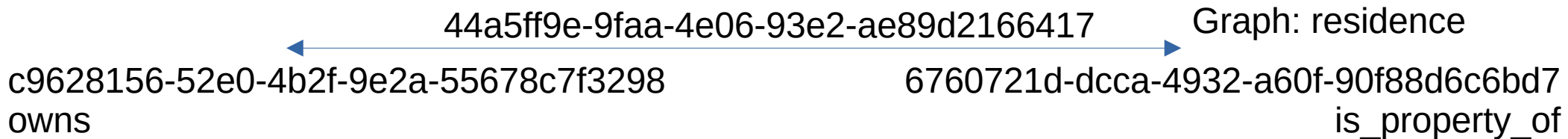
# Vertices

- An oid field (UUID)
- A type = vertex
- A subtype
- A creation timestamp
- A data field (string)
- A field that indicates whether a vertex is unique (e.g. there is only one Paris, France)
- A vector field (not used yet: e.g. to represent synonyms and antonyms on the surface of a Poincaré sphere)



# Edges

- Oid (UUID)
- Type = edge
- Subtype
- Graph name
- Input oid
- Input name
- Input vector
- Output oid
- Output name
- Output vector



# Milestones are specialized vertices

- They represent a period from 1ns to years.
  - They have a start date
  - And an end date in addition to a standard vertex fields.
  - A date can be inaccurate e.g. a year

This was chosen to represent the fuzziness of dates that cannot be instants but periods

# Benefits

- Vertices do not reference graphs
- Serialization of complex graphs is easy
- An object table, references objects by their id.
- Sufficiently fast compared to speech required time
- Can be saved either as json files or in a 2 Table SQL database (3 if we want to have a dedicated table for milestones), graphs are created by the names of the edges.

# Access to created vertices and edges

- 3 contexts:
  - The whole graph (all the graphs),
  - A session context,
  - A short term context (per sub dialogue)
- 1 (oid,vertex or edge) map: object table as for Object DBs
- 1 (name,vertex) map
- Access to graphs through their names.



# Graph theory and dialogue

- Difficulty to determine transitivity automatically in language since it is semantically based:
  - A cheap horse is rare, what is rare is expensive, thus a cheap horse is expensive.
- Non directional edges for automatic graph browsing: Inverse sentence of an edge, usually passive form, (input\_sentence ↔ output\_sentence) is not trivial and must be adaptable to the vertices an edge is connected to. Easy localization must also be taken into account.

# What is left

- We only had 3 months to have something running on a phone, in which, one month has been dedicated to adapt a LLM.
- For complex answers we must add:
  - either a 80's Chomskian grammar analysis
  - Or a LLM based analysis
- LLM for bio redaction
- Integration: Diarization, Weather, sensor integration etc.
- Use the contacts in the phones
- Develop a dialogue editor, a project in itself
- Integrate photos and videos

# What we did wrong

## Graph Programming is not Object Programming

- Smalltalk programmer by education, ex Gemstone Systems employee, our early vertices were too object like which implied dedicated code.
- Automatic browsing became too much case by case.
- => Unlearn object programming and make the vertices minimal (unlike the examples provided by some graph DB vendors).

Thank you



Ministry of Internal Affairs  
and Communications, JAPAN



# Appendix

# Example of loop dialogue node

```
{
  "id": "20_1",
  "name": "user describes her house",
  "condition": "default",
  "action": [
    "create_vertex subtype: utterance name: answer.value constraint: unique",
    "create_edge graph: residence between OWNER and short_term_context_last.inputName said outputName as_said_by",
    "create_edge graph: residence between short_term_context_last and HOME.inputName said_about outputName is_described_as"
  ],
  "response_timer": "30s",
  "random_sentence_choice": true,
  "loop_exit": "loop_exit_regex",
  "sentences": [
    {
      "sentence": "Okay",
      "variables": ""
    },
    {
      "sentence": "Excellent",
      "variables": ""
    },
    {
      "sentence": "Cool",
      "variables": ""
    }
  ],
  "children": ["20_1"],
  "response_timer_children": ["21_1"],
  "loop_exit_child": "23_1",
  "requires_answer": true,
  "on_error": "on_error_1"
}
```

