50th Vienna Deep Learning Meetup

4th May 2023

#VDLM

@ Bosch

Vienna Deep Learning Meetup

The Organizers:

Thomas Lidy
Utopia Music

Alex Schindler
AIT & TU Wien

Jan Schlüter
JKU Linz

René Donner
mva.ai &
contextflow

Topics for Today

Vienna Deep Learning Meetup

Welcome by the organizers

Introduction by the host - *Josipa Basta & Elisabeth Weigl, Bosch*

Security of Machine Learning Systems - (How) Can We Get There?
Rudolf Mayer, SBA Research & Lecturer @ TU Wien

Announcements + Break

Intro to (Deep) Reinforcement Learning
Sharwin Rezagholi, FH Technikum Wien

Hot Papers: Latest Developments in Large Language Models
Michael Pieler, OpenBioML.org & Stability.AI

Networking

Slides will be provided on: https://github.com/vdlm/meetups

Announcements

Vienna
Deep Learning
Meetup

VDLM on Github

https://github.com/vdlm/meetups

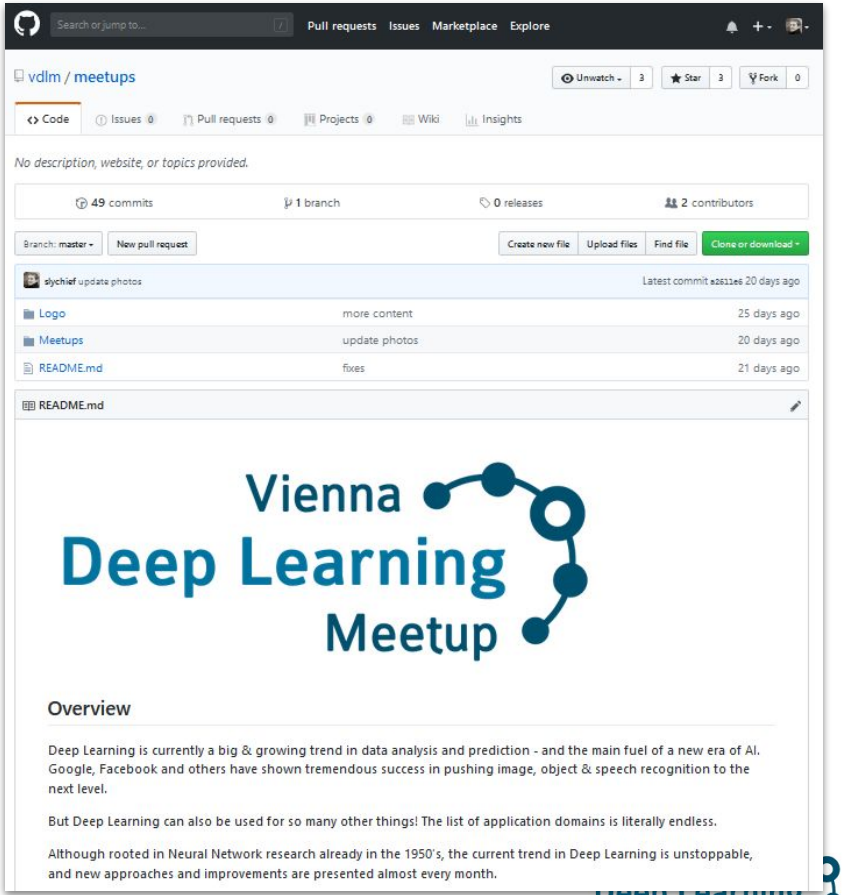
- all talks
- slides
- photos
- videos
- Wiki

Meetups

#	Date	Place	Topic	Link	Video	Meetup.com
1	2016-04-07	Sector 5	intro	more		link
2	2016-05-09	Sector 5		more		link
3	2016-06-06	Sector 5		more		link
4	2016-07-07	TU Wien		more		link
5	2016-09-22	Automic Software GmbH		more		link
6	2016-10-12	Sector 5		more		link
7	2016-12-01	Agentur Virtual Identity		more		link
8	2017-01-17	TU Wien Informatik		more		link
9	2017-02-21	bwin.party services (Austria) GmbH		more		link

Talks

Date	MU#	Speaker	Topic	Slides
2016-04-07	1	Thomas Lidy	An overview presentation of Deep Learning	pdf
2016-04-07	1	Jan Schlüter	History, Approaches, Applications	pdf
2016-05-09	2	Alex Champandard	Neural Networks for Image Synthesis	
2016-05-09	2	Gregor Mitscha-Baude	Recurrent Neural Networks	pdf
2016-06-06	3	Jan Schlüter	Open-source Deep Learning with Theano and Lasagne	pdf
2016-09-22	5	Josef Puchinger	Deep Learning & The Future of Automation	
2016-09-22	5	Christoph Körner	Going Deeper with GoogLeNet and CaffeJS	pdf



Deep Learning
Meetup

Bzzzz - Konferenz der österreichischen Musikwirtschaft

- **5th May 2023** (tomorrow!)
- all day 10-20 h
- WKO

14:00 Uhr | **Everything is data. Data is everything**

In der modernen Musikwirtschaft geht nichts ohne Daten. Die Wiener Plattformen Fortunes und MusiMap gehören zum internationalen Musitech-Konzern Utopia und loten seit Langem die Möglichkeiten aus, die Algorithmen, künstliche Intelligenz und Datenanalysen bieten. Vom frühen Erkennen von Trends zu exaktem Finden von Zielgruppen, von exaktem Monitoring von Radio Airplays bis emotionsgesteuerten Musikempfehlungen; kaum eine Tätigkeit in der Musikwirtschaft lässt sich heute nicht von Daten (unter)stützen. Ein intensiver Ein- und Überblick.

- Nina Wöss (Female Founders/Fund F)
- Markus Riedler (Napalm Records)
- Florian Richling (ForTunes / Utopia)
- Franz Medwenitsch (IFPI Austria / LSG)
- Thomas Lidy (MusiMap / Utopia)

(15:30 - 16:00 Uhr: Pause)

16:00 Uhr | **AI – PANDORAS BOX ODER HEILIGER GRAL?**

"Künstliche Intelligenz (AI) hat nun auch im künstlerisch-kreativen Bereich einen Punkt erreicht, wo ein vermeintliches, neues Oasis-Album im Netz auftaucht, das stimmlich und musikalisch klingt wie die Band in den 90ern; oder eine Kollaboration der Superstars Drake und The Weeknd ohne deren Zutun entsteht. Die künstlerischen und technischen Möglichkeiten scheinen grenzenlos, doch hat die Entwicklung sämtliche Debatten über rechtliche und moralische Grenzen längst überholt. Kann, soll, muss die Musik(wirtschaft) als traditioneller Early Adopter technologischer Veränderungen einmal mehr den Weg vorgeben? Wie kann sie aussehen und wie kann sie das tun? Wird uns AI völlig überrollen oder können wir auch einen praktischen Nutzen aus den Errungenschaften ziehen?"

- Thomas Lidy (MusiMap/Utopia)
- Christine Bauer
- Franz Medwenitsch (IFPI Austria / LSG)
- Moderation: Claudia Zettel (Chefredakteurin Futurezone)

Vienna Deep Learning Meetup

Vienna Deep Learning Meetup

A lot of conferences next year in Vienna!

- **ICLR - International Conference on Learning Representations**
  https://iclr.cc/Conferences/FutureMeetings
  May 7 - 11, 2024

- **ICML - International Conference on Machine Learning**
  https://dev.icml.cc/Conferences/FutureMeetings
  July 19 - 27, 2024

… if you have events or jobs to announce, write us at **contact@vdlm.at**

Break

Topics for Today

Vienna Deep Learning Meetup

Welcome & Introduction

Security of Machine Learning Systems - (How) Can We Get There?
Rudolf Mayer, SBA Research & Lecturer @ TU Wien

Announcements + Break

Intro to (Deep) Reinforcement Learning
Sharwin Rezagholi, FH Technikum Wien

Hot Papers: Latest Developments in Large Language Models
Michael Pieler, OpenBioML.org & Stability.AI

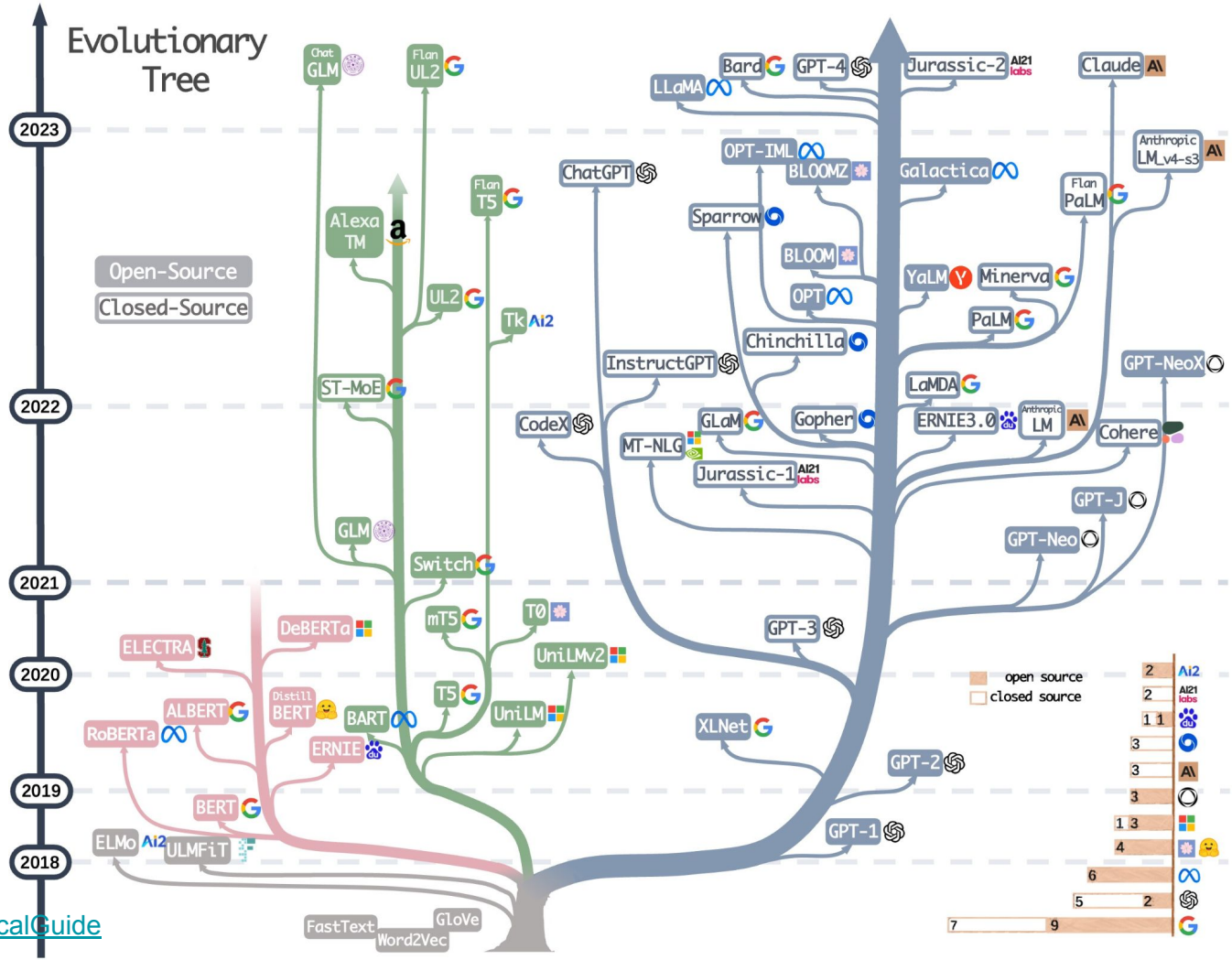
Networking

Slides will be provided on: https://github.com/vdlm/meetups

LLM Hot Papers

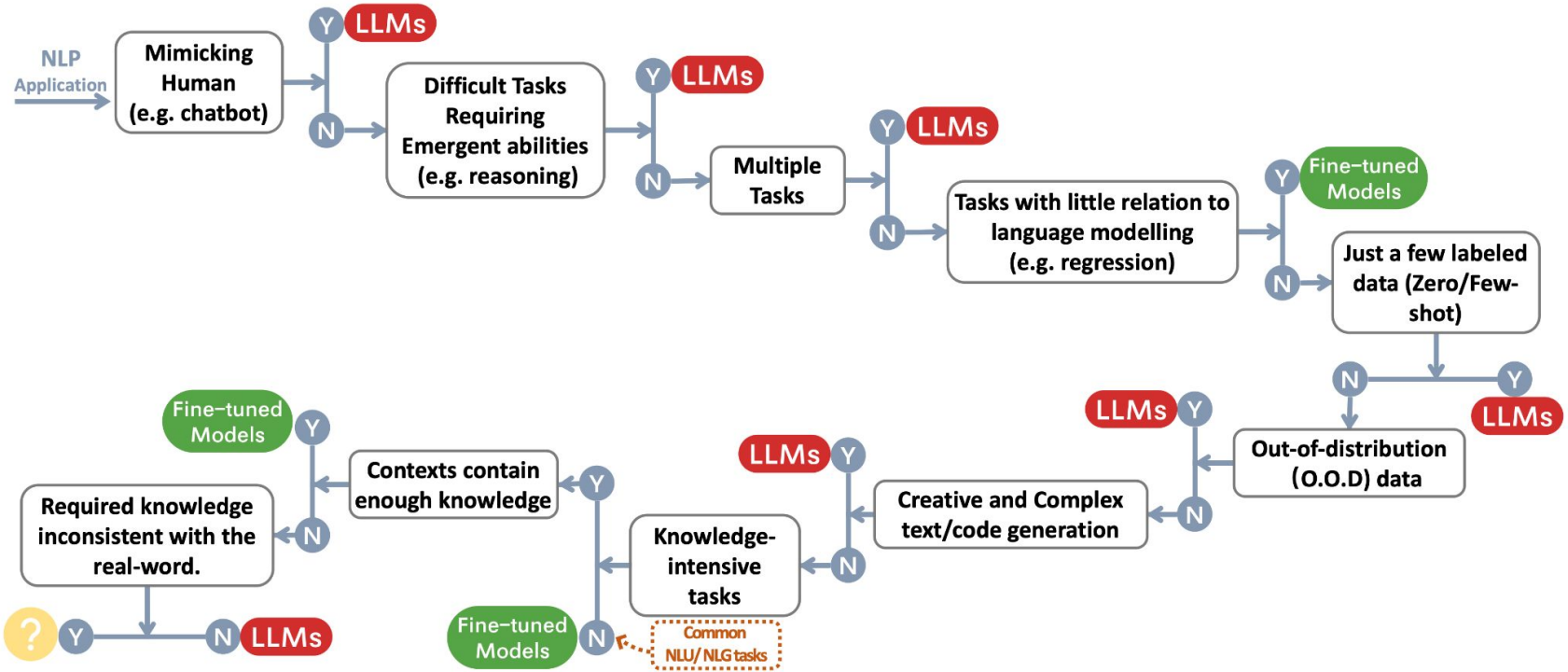
https://OpenBioML.org

The Practical Guides for LLMs



https://github.com/Mooler0410/LLMsPracticalGuide

Practical Guide for NLP Tasks



https://github.com/Mooler0410/LLMsPracticalGuide

Recurrent Memory Transformer (RMT)

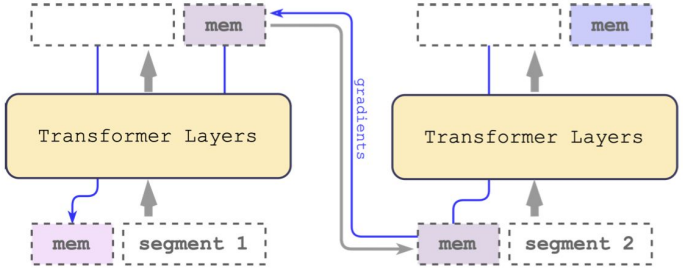


Figure 2: **Recurrent memory mechanism.** Memory is passed to Transformer along input sequence embeddings, and memory output is passed to the next segment. During training gradients flow from the current segment through memory to the previous segment.

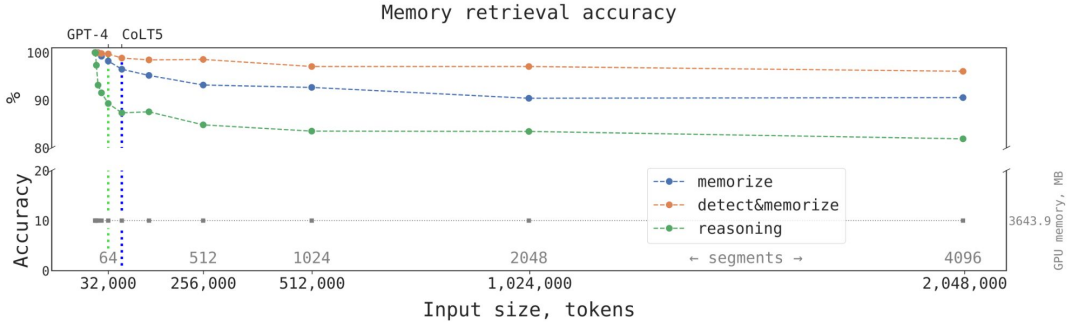


Figure 1: **Recurrent Memory Transformer retains information across up to 2×10^6 tokens**. By augmenting a pre-trained BERT model with recurrent memory (Bulatov et al., 2022), we enabled it to store task-specific information across 7 segments of 512 tokens each. During inference, the model effectively utilized memory for up to 4,096 segments with a total length of 2,048,000 tokens—significantly exceeding the largest input size reported for transformer models (64K tokens for CoLT5 (Ainslie et al., 2023), and 32K tokens for GPT-4 (OpenAI, 2023)). This augmentation maintains the base model's memory size at 3.6 GB in our experiments.

Scaling Transformer to 1M tokens and beyond with RMT

Training LMs with Language Feedback at Scale

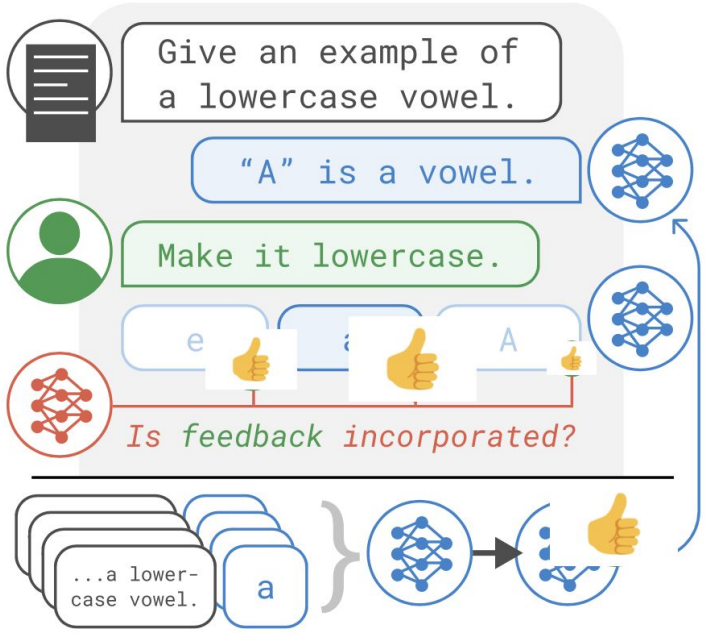
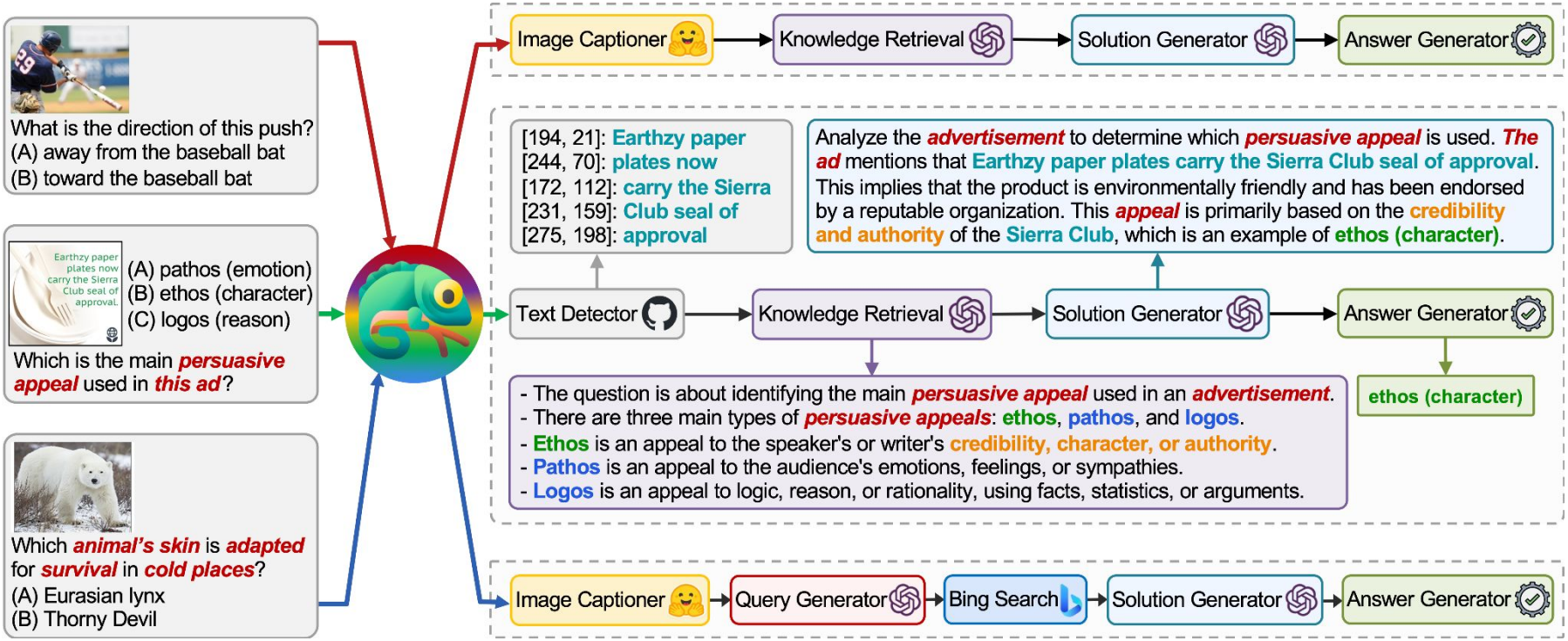


Figure 1: To learn from language feedback on a language model (LM) output, we have an LM generate multiple refinements of the original output based on the feedback. We use an LM to pick the best refinement and finetune the original LM to maximize the likelihood of the chosen refinement.

Training Language Models with Language Feedback at Scale

Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models



https://chameleon-llm.github.io/, https://github.com/lupantech/chameleon-llm

Chameleon

▷ *Instruction for the planner model*

You need to act as a policy model, that given a question and a modular set, determines the sequence of modules that can be executed sequentially can solve the question.

The modules are defined as follows:

Query_Generator: This module generates a search engine query for the given question. Normally, we consider using "Query_Generator" when the question involves domain-specific knowledge.
Bing_Search: This module searches the web for relevant information to the question. Normally, we consider using "Bing_Search" when the question involves domain-specific knowledge.
Image_Captioner: This module generates a caption for the given image. Normally, we consider using "Image_Captioner" when the question involves the semantic understanding of the image, and the "has_image" field in the metadata is True.
Text_Detector: This module detects the text in the given image. Normally, we consider using "Text_Detector" when the question involves the unfolding of the text in the image, e.g., diagram, chart, table, map, etc., and the "has_image" field in the metadata is True.
Knowledge_Retrieval: This module retrieves background knowledge as the hint for the given question. Normally, we consider using "Knowledge_Retrieval" when the background knowledge is helpful to guide the solution.
Solution_Generator: This module generates a detailed solution to the question based on the information provided. Normally, "Solution_Generator" will incorporate the information from "Query_Generator", "Bing_Search", "Image_Captioner", "Text_Detector", and "Knowledge_Retrieval".
Answer_Generator: This module extracts the final answer in a short form from the solution or execution result. This module normally is the last module in the prediction pipeline.

Below are some examples that map the problem to the modules.

▷ *In-context example(s)*

Question: Compare the average kinetic energies of the particles in each sample. Which sample has the higher temperature?

Context: The diagrams below show two pure samples of gas in identical closed, rigid containers. Each colored ball represents one gas particle. Both samples have the same number of particles.

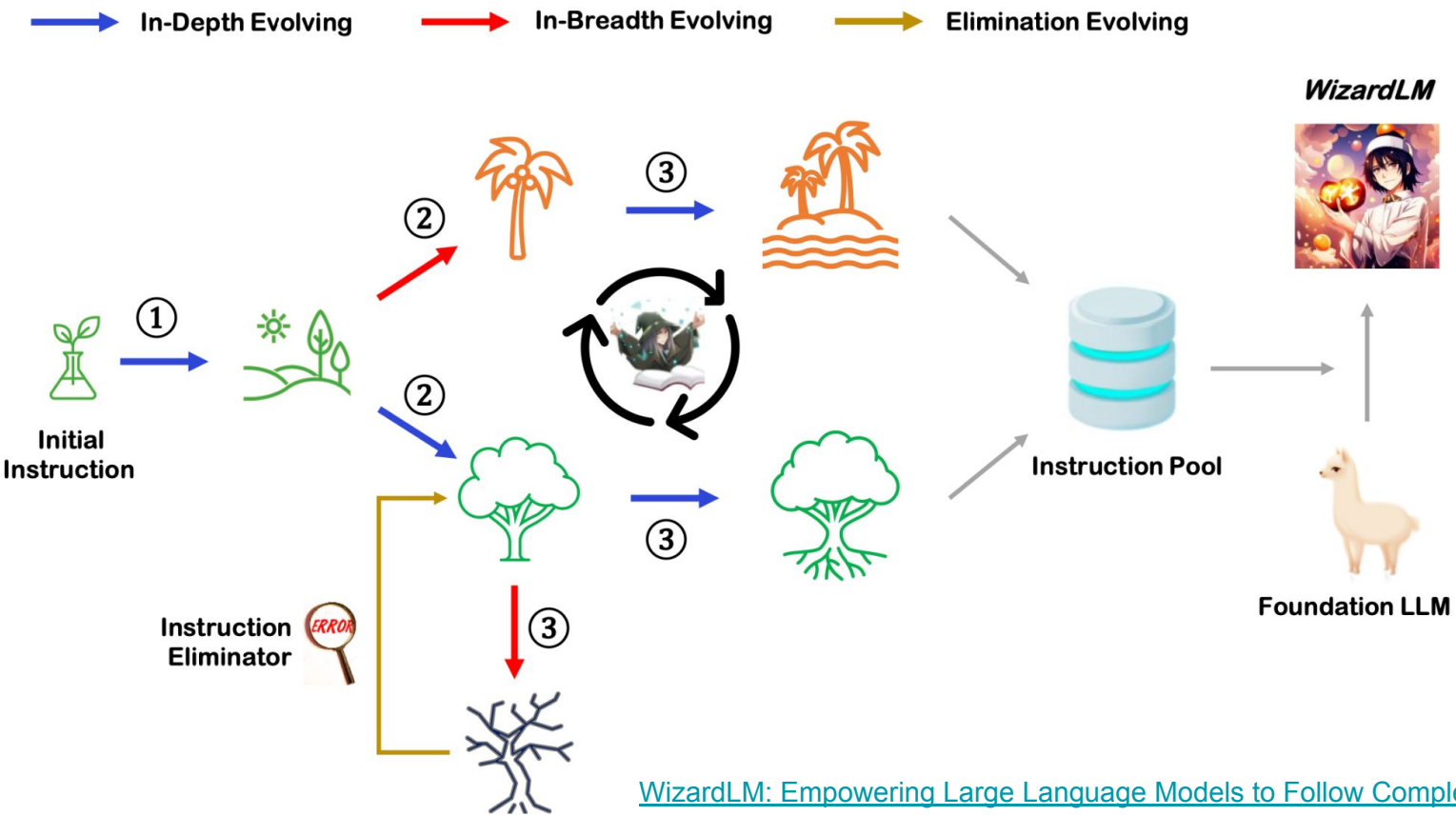
Options: (A) neither; the samples have the same temperature (B) sample A (C) sample B

Metadata: 'pid': 19, 'has_image': True, 'grade': 8, 'subject': 'natural science', 'topic': 'physics', 'category': 'Particle motion and energy', 'skill': 'Identify how particle motion affects temperature and pressure'

Modules: `["Text_Detector", "Knowledge_Retrieval", "Solution_Generat or", "Answer_Generator"]`

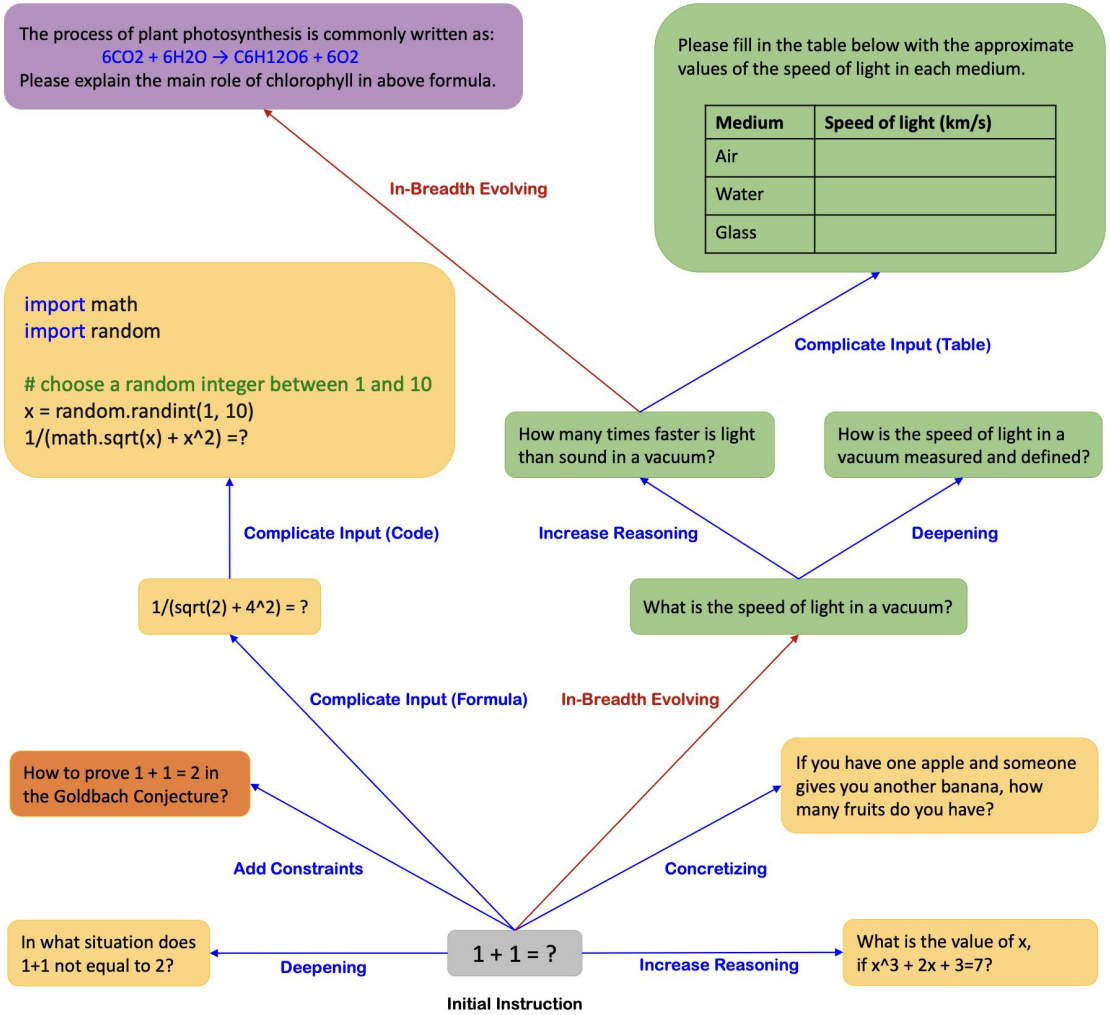
https://arxiv.org/pdf/2304.09842.pdf#page=17

WizardLM



[WizardLM: Empowering Large Language Models to Follow Complex Instructions]

WizardLM



[WizardLM: Empowering Large Language Models to Follow Complex Instructions]

WizardLM: In-Depth Evolving

```
I want you act as a Prompt Rewriter.
Your objective is to rewrite a given prompt into a more complex version to make those famous AI systems
(e.g., chatgpt and GPT4) a bit harder to handle.
But the rewritten prompt must be reasonable and must be understood and responded by humans.
Your rewriting cannot omit the non-text parts such as the table and code in #Given Prompt#:.  Also, please
do not omit the input in #Given Prompt#.
You SHOULD complicate the given prompt using the following method:

Please add one more constraints/requirements into #Given Prompt# or
If #Given Prompt# contains inquiries about certain issues, the depth and breadth of the inquiry can be
increased.  or
Please replace general concepts with more specific concepts.  or
If #Given Prompt# can be solved with just a few simple thinking processes, you can rewrite it to
explicitly request multiple-step reasoning.

You should try your best not to make the #Rewritten Prompt# become verbose, #Rewritten Prompt# can only
add 10 to 20 words into #Given Prompt#.
'#Given Prompt#', '#Rewritten Prompt#', 'given prompt' and 'rewritten prompt' are not allowed to appear in
#Rewritten Prompt#
#Given Prompt#:
<Here is instruction.>
#Rewritten Prompt#:
```

WizardLM: Empowering Large Language Models to Follow Complex Instructions

WizardLM: In-Breadth Evolving

```
I want you act as a Prompt Creator.
Your goal is to draw inspiration from the #Given Prompt# to create a brand new prompt.
This new prompt should belong to the same domain as the #Given Prompt# but be even more rare.
The LENGTH and difficulty level of the #Created Prompt# should be similar to that of the #Given Prompt#.
The #Created Prompt# must be reasonable and must be understood and responded by humans.
‘#Given Prompt#’, ‘#Created Prompt#’, ‘given prompt’ and ‘created prompt’ are not allowed to appear in
#Created Prompt#.

#Given Prompt#:
<Here is instruction.>

#Created Prompt#:
```

WizardLM: Empowering Large Language Models to Follow Complex Instructions

Generative Agents: Interactive Simulacra of Human Behavior

Figure 1: Generative agents create believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents they plan their days, share news, form relationships, and coordinate group activities.

Generative Agents: Interactive Simulacra of Human Behavior

Generative Agents: Interactive Simulacra of Human Behavior

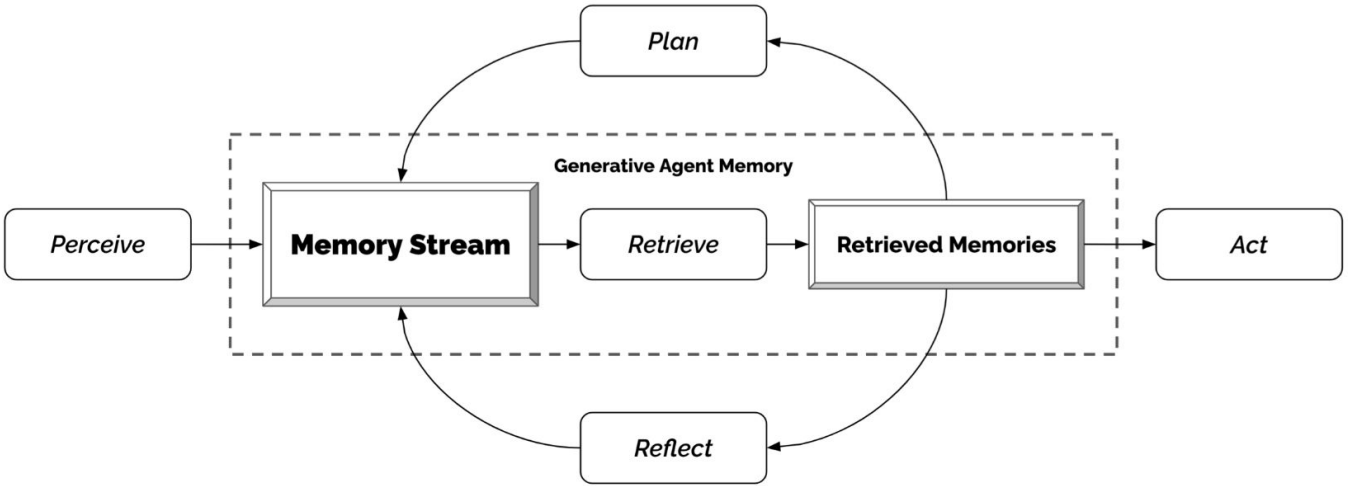


Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories, then uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans, and to create higher-level reflections, which are both entered into the memory stream for future use.

Generative Agents: Interactive Simulacra of Human Behavior

Generative Agents: Interactive Simulacra of Human Behavior

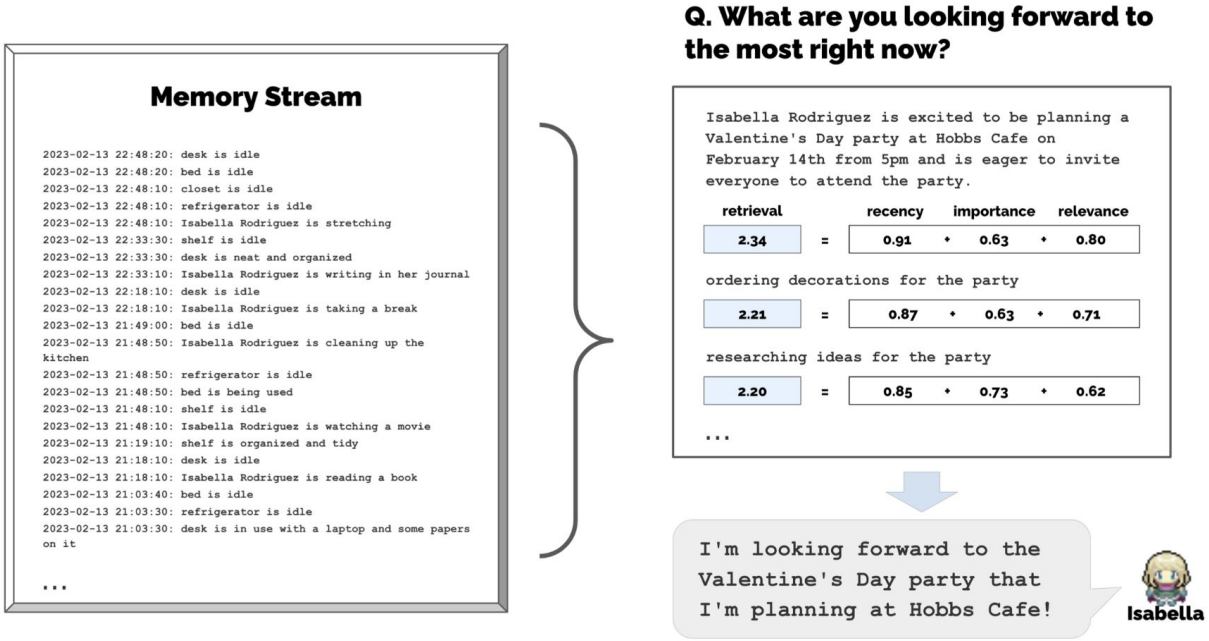


Figure 6: The memory stream comprises a large number of observations that are relevant and irrelevant to the agent's current situation. Retrieval identifies a subset of these observations that should be passed to the language model to condition its response to the situation.

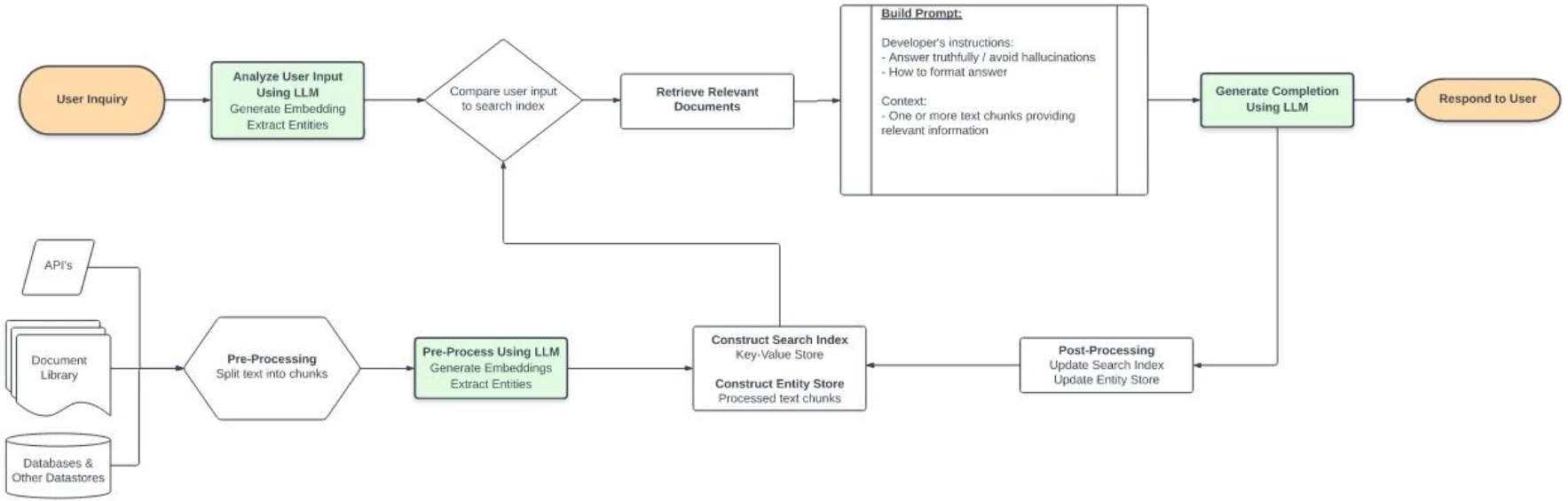
Generative Agents: Interactive Simulacra of Human Behavior

Release of StableLM

Size	StableLM-Base-Alpha	StableLM-Tuned-Alpha	Training Tokens	Parameters	Web Demo
3B	checkpoint	checkpoint	800B	3,638,525,952	
7B	checkpoint	checkpoint	800B	7,869,358,080	Hugging Face
15B	(in progress)	(pending)			
30B	(in progress)	(pending)			
65B	(in progress)	(pending)			
175B	(planned)				

https://github.com/Stability-AI/StableLM

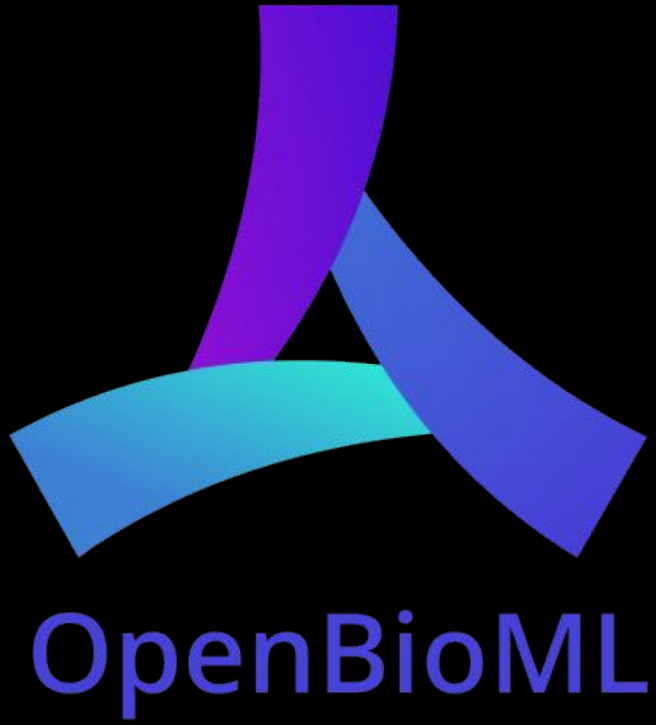
Knowledge Retrieval Architecture for LLMs



https://mattboegner.com/knowledge-retrieval-architecture-for-llms/

Other interestings things

- 2023 State of AI in 14 Charts
- ChatGPT Prompt Engineering for Developers - DeepLearning.AI
- Google "We Have No Moat, And Neither Does OpenAI" (brand new!)



https://OpenBioML.org

4th May 2023

#VDLM

@ Bosch

THANK YOU!

Next DL Meetup

Vienna Deep Learning Meetup

~probably around 14/15th June

After ChatGPT: The dawn of machines that self-reflect and self-direct

Matthias Samwald, MedUni Wien

Segment Anything - Architecture and Use Cases

René Donner, mva.ai