

大模型时代的神经符号计算

刘群 LIU Qun

华为诺亚方舟实验室

MLA2024 第22届机器学习及其应用研讨会

2024.11.03



NOAH'S ARK LAB



HUAWEI

Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

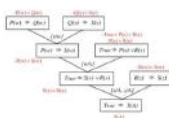
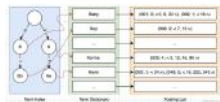
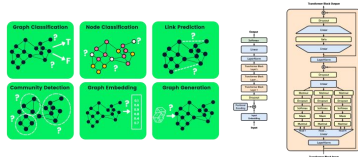
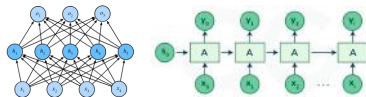
利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

神经网络 vs. 符号计算

	神经网络	符号计算
数学基础	微积分、概率论、信息论	图论、形式语法、形式逻辑、概率论
典型方法	感知机、卷积网络、循环神经网络、Transformer、图神经网络	正则表达式、短语结构语法、Parsing、一阶逻辑、知识图谱、专家系统、倒排检索
擅长问题	分类、表示学习、生成	精确匹配、规则推导、逻辑推导
弱点	精确匹配、长链条推理	学习能力、模糊推理



大语言模型（LLM）具备了一定的符号处理能力

- ▶ 早期的神经网络并不具备任何符号处理能力
- ▶ 后来神经网络发展出了输入输出符号序列的能力：
 - ▶ 词嵌入技术赋予了神经网络接收符号输入的能力
 - ▶ 位置编码使得神经网络能够接受符号序列（语言）的输入
 - ▶ Softmax使得神经网络能够输出符号
 - ▶ 自回归生成生成使得神经网络能够输出连续的符号（语言）
- ▶ 现在的神经网络，特别是基于Transformer的LLM，已经能够进行完成很多符号处理的任务，包括机器翻译这样复杂的任务
- ▶ LLM在思维链技术加持下，已经具备了一定程度的符号推理能力
- ▶ LLM辅以基于长思维链的强化学习技术（o1），数学代码能力达到了超越普通人的水平

大语言模型的符号计算能力仍然欠缺



常识理解



数学推理

大语言模型的符号计算能力仍然欠缺

- ▶ LLM仍然不具备基本的数数能力：
 - ▶ “Strawberry中有多少个字母r” 这个漏洞虽然被大部分模型补上了，实际上换一个单词和字母，类似的问题仍然层出不穷
 - ▶ 类似的数数问题，当数目比较大的时候，LLM大概率还是容易出错
- ▶ LLM在一些简单的常识推理中也还很容易犯错：

小王有三个姐姐两个弟弟，他的姐姐有几个弟弟？

小王的姐姐有两个弟弟，分别是小王和他的另一个弟弟。小王有三个姐姐两个弟弟，无论哪个姐姐，其弟弟都是小王和他的另一个弟弟这两个人。

实现真正的人类水平智能，需要AI模型中引入符号计算吗？

不需要，神经网络将通过Scaling Law将自然获得符号计算的能力



Geoffery Hilton



Ilya Sutskever



Yoshua Bengio



Gary Marcus

需要，神经网络无法精确推理，需研究神经符号结合的方法

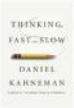

大语言模型还不真正具备系统2的思考能力

SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

System 1


- Innative, fast, **UNCONSCIOUS**, non-linguistic, habitual
- Current DL




System 2

Manipulates high-level / semantic concepts, which can be recombined combinatorially

- Slow, logical, sequential, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Future DL





- ▶ 人类思维分为两个系统，神经网络比较好地模拟了系统1，而系统2的思维更多是基于符号的
- ▶ LLM Agent和o1具备了系统2的某些特点（如记忆、推理等），但离人类的思维仍然具有较大差距

生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

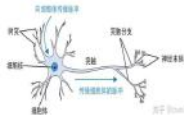
生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

生物神经网络



生物智能的进化 vs. 人工智能的进化

客观世界



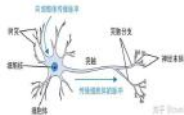
原子、分子
声、光、电
物质、波
.....

语言表达

Language Learning Models (LLMs) have revolutionized the field of natural language processing, enabling machines to understand and generate human-like text. At the core of LLMs lies the concept of tokens, which serve as the fundamental building blocks for processing and representing text data. In this blog post, we'll deconstruct tokens in LLMs, unraveling their significance and exploring how they contribute to the power and flexibility of these remarkable models.

字、词
句、篇

生物神经网络



生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

人类知识



Bob Dylan wrote 'Blowin' in the Wind' in 1962, and wrote 'Chronicles: Volume One' in 2004

语言表达

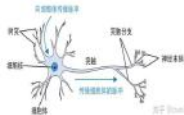
Language Learning Model (LLM) have revolutionized the field of natural language processing, enabling machines to understand and generate human-like text. At the core of LLMs lies the concept of tokens, which serve as the fundamental building blocks for processing and representing text data. In this blog post, we'll deconstruct tokens in LLMs, unraveling their significance and exploring how they contribute to the power and flexibility of these remarkable models.

字、词
句、篇

实体、属性
关系、事件
时空、因果
.....

数学、物理
化学、生物
.....

生物神经网络



生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

人类知识



Bob Dylan wrote 'Blowin' in the Wind' in 1962, and wrote 'Chronicles: Volume One' in 2004

语言表达

Language Learning Models (LLMs) have revolutionized the field of natural language processing, enabling machines to understand and generate human-like text. At the core of LLMs lies the concept of tokens, which serve as the fundamental building blocks for processing and representing text data. In this blog post, we'll dive into tokens in LLMs, unraveling their significance and exploring how they contribute to the power and flexibility of these remarkable models.

字、词
句、篇

实体、属性
关系、事件
时空、因果
.....

数学、物理
化学、生物
.....

冗余度

精确性

生物神经网络

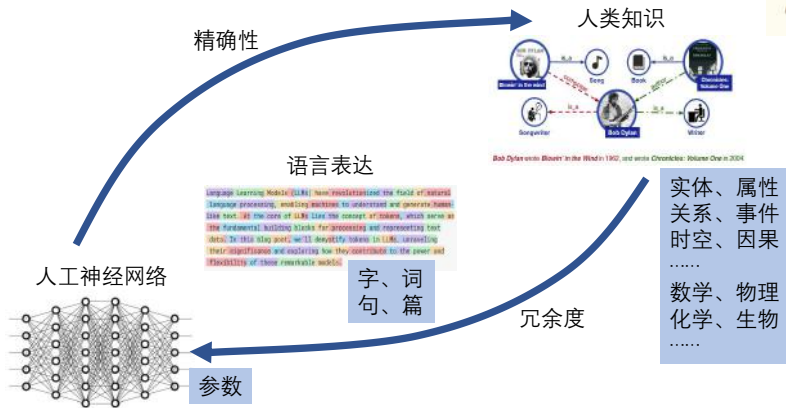


生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

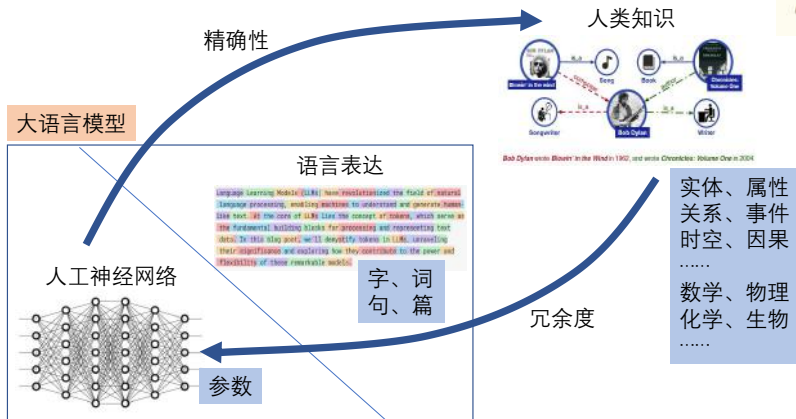


生物智能的进化 vs. 人工智能的进化

客观世界



原子、分子
声、光、电
物质、波
.....

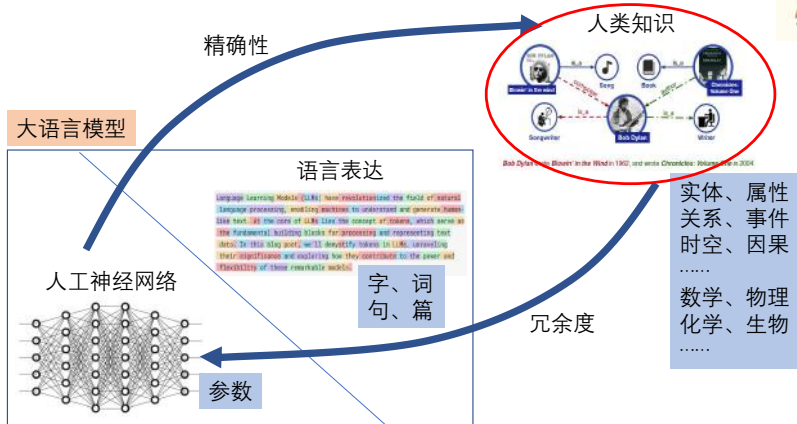


生物智能的进化 vs. 人工智能的进化

客观世界

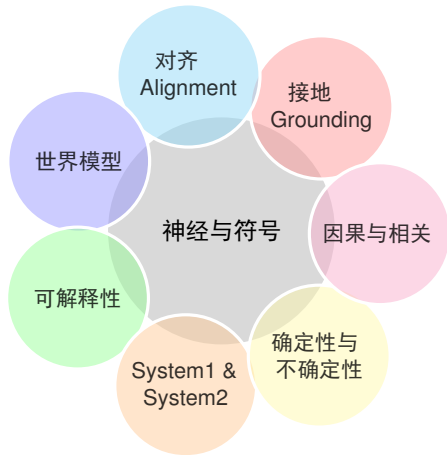


原子、分子
声、光、电
物质、波
.....



神经与符号的GAP是目前大模型很多问题的根源

- ▶ 人类认知表示最自然的形式是实体、属性、关系、时空、事件、因果等等，这些都最适合用符号来表示
- ▶ 而目前的大语言模型的基本构成单位是参数，所有的计算和推理都发生在参数之间，跟认知所使用的符号表示的形式存在巨大的差异
- ▶ 这种差异是造成目前大模型很多问题的根源



符号化知识表示的形式

- ▶ Declarative knowledge 陈述性知识
 - ▶ 描述概念、实体、事实
 - ▶ 以陈述句形式描述
- ▶ Procedural knowledge 过程性知识
 - ▶ 包括规则、策略、过程、议程等
 - ▶ 可以执行完成任务
- ▶ Meta-knowledge 元知识
 - ▶ 关于知识的知识
- ▶ Heuristic knowledge 启发式知识
 - ▶ 专家根据经验获得的领域或专业知识
- ▶ Structural knowledge 结构化知识
 - ▶ 概念之间的上下位或者整体部分关系等知识
 - ▶ 解决问题所需要的知识



Source: <https://www.javatpoint.com/knowledge-representation-in-ai>

符号化知识表示的类型

- ▶ 自然语言
 - ▶ 词、短语、句子、篇章
- ▶ 语义网（Semantic Web）、知识图谱
 - ▶ 知识本体（Ontologies）
 - ▶ 实体知识图谱（Entities/Relations/Facts）
 - ▶ 事理知识图谱（Events）
- ▶ 程序语言（规则都可以表示为程序）
 - ▶ 函数式程序语言、过程式程序语言、面向对象程序语言
- ▶ 逻辑语言
 - ▶ 布尔逻辑、命题逻辑、描述逻辑、构造逻辑、一阶谓词逻辑、高阶谓词逻辑
- ▶ 图表语言
 - ▶ 表格（Tables、Spreadsheets）、图（graphs）、自由图表（Diagrams）

符号化知识表示的类型

自然语言

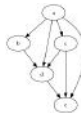
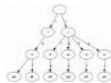
中文, English, Español, Français, Deutsch, 日本語, Русский, Português, Italiano, 한국어

图表语言

产品	第1季度	第2季度	合计
1. 笔记本电脑	¥744.00	¥762.00	¥1,506.00
2. 平板电脑	¥5,778.00	¥1,240.00	¥7,018.00
3. 数码相机	¥1,038.00	¥1,082.00	¥2,120.00
4. 数码相机	¥1,148.00	¥754.00	¥1,902.00
5. 数码相机	¥1,708.00	¥4,542.00	¥6,250.00
6. 数码相机	¥3,018.00	¥5,018.00	¥8,036.00
7. 数码相机	¥1,108.00	¥1,108.00	¥2,216.00
8. 数码相机	¥1,108.00	¥1,108.00	¥2,216.00



表格



图



自由图表

形式语言

产生式规则 C C++
Python JavaScript
Latex Json HTML
Markdown 知识图谱

程序语言

描述逻辑

命题逻辑

布尔逻辑

构造逻辑

一阶谓词逻辑

高阶谓词逻辑

逻辑语言

符号化知识表示的多样性难题

- ▶ 神经网络为所有问题提供统一的解决方案，而符号化知识形式非常多样化，每种形式特点不同，面临的问题也不同
- ▶ 不同的符号化知识表示形式，需要采用不同的方式与神经网络结合
- ▶ 是否存在统一的形式化知识表示形式，可以解决所有符号推理问题？
- ▶ 如果不存在，有哪些主要的符号化知识表示形式？各自有什么特点？如何与大语言模型结合进行推理？

Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

通过思维链及其衍生技术强化大语言模型本身的推理能力

- ▶ 大语言模型可以通过思维链Chain-of-Thought及其各种衍生的X-of-Thought（如思维树Tree-of-Thought，思维图Graph-of-Thought等）来实现一定的符号推理能力
- ▶ 由于预训练数据已经包含了一定的推理数据，大模型本身已经具备一定的思维链推理能力
- ▶ 如果需要通过实现更强的类思维链推理能力，需要构造更多的推理数据，采用SFT等方式，就可以使模型获得此类推理能力
- ▶ 通过各种类思维链数据SFT实现的推理能力通常比较有限

Chain-of-Thought and Thinking-Step-by-Step

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Kojima et al., Large Language Models are Zero-Shot Reasoners, arXiv:2205.11916v4, 2023

Self-consistency improves CoT

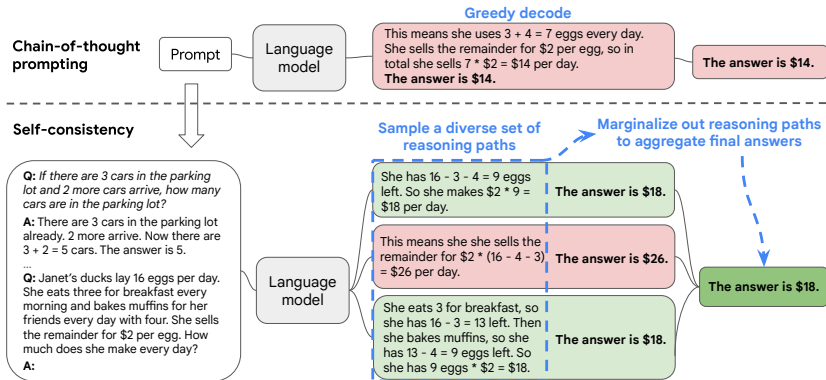


Figure 1: The self-consistency method contains three steps: (1) prompt a language model using chain-of-thought (CoT) prompting; (2) replace the “greedy decode” in CoT prompting by sampling from the language model’s decoder to generate a diverse set of reasoning paths; and (3) marginalize out the reasoning paths and aggregate by choosing the most consistent answer in the final answer set.

Progressive-Hint Prompting improves CoT

Base Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday?

LLM

Base Answer

A: There were 96 fourth-graders at Small Tree School. 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls at school on Friday, and $96 - 38 = 58$ boys at school on Friday. The answer is **58**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? (Hint: The answer is near to **58**).

LLM

Second Answer

A: We know the Answer Hints: **58**. With the Answer Hints: **58**, we will answer the question. There were 96 fourth-graders at Small Tree School, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 43 = 53$ boys at Small Tree School on Friday. After subtracting the absent students, there were $38 - 4 = 34$ girls and $53 - 5 = 48$ boys. The answer is **48**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? (Hint: The answer is near to **58, 48**).

LLM

Third Answer

A: We know the Answer Hints: **58, 48**. With the Answer Hints: **58, 48**, we will answer the question. There were 96 fourth-graders, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 9 = 87$ students in total. Of those 87 students, 38 were girls, so there must have been $87 - 38 = 49$ boys. The answer is **49**.

Progressive-Hint Prompting

Q: There are 96 fourth-graders at Small Tree School. 43 of them are girls. On Friday, 5 fourth-grade girls and 4 fourth-grade boys were absent. How many fourth grade boys were at Small Tree School on Friday? (Hint: The answer is near to **58, 48, 49**).

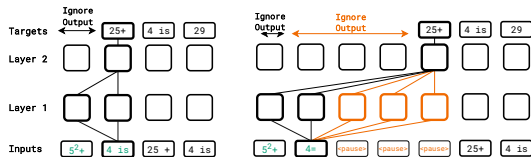
LLM

Fourth Answer

A: We know the Answer Hints: **58, 48, 49**. With the Answer Hints: **58, 48, 49**, we will answer the question. There were 96 fourth-graders at Small Tree School, and 43 of them were girls. On Friday, 5 girls and 4 boys were absent. So there were $43 - 5 = 38$ girls and $96 - 43 = 53$ boys originally. After the absences, there were $38 - 5 = 33$ girls and $53 - 4 = 49$ boys. The answer is **49**.

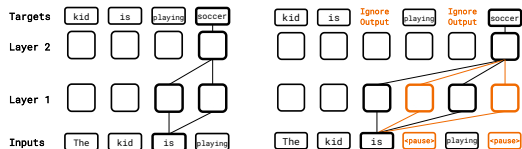
Zheng et al., Progressive-Hint Prompting Improves Reasoning in Large Language Models, arXiv:2304.09797v5, 2023

Think Before You Speak: Training LMs With Pause Tokens



(a) Standard inference and finetuning

(b) Pause-inference and finetuning



(a) Standard pretraining

(b) Pause-pretraining

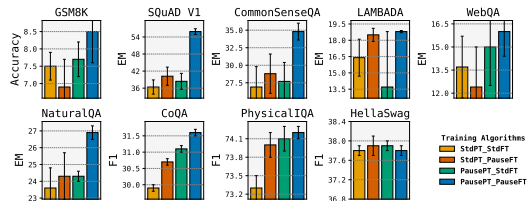


Figure 3: **Downstream performance for a 1B model.** Injecting delays in both stages of training (PausePT.PauseFT) outperforms the standard end-end training StdPT.StdFT on our wide variety of tasks (except HellaSwag). In contrast, introducing delays only in the finetuning stage provides only lukewarm gains, and even hurts in GSM8k.

Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

基于大语言模型的神经符号系统

- ▶ 大语言模型（LLM）本身虽然具备了一定程度的符号处理能力，但它还是一个纯粹的神经网络系统，而不是一个神经符号计算系统，因为LLM中完全没有独立的符号计算组件（component）。
- ▶ 大语言模型（LLM）只有跟独立的符号系统组件相结合，各司其职，互相合作，才能称为一个神经符号计算系统。
- ▶ 基于大语言模型的神经符号系统，根据其结合的符号系统类型，大致可以分成以下一些类别：
 - ▶ 融合搜索引擎的大语言模型（RAG）
 - ▶ 融合工具插件调用的大语言模型
 - ▶ 基于大语言模型的智能体（LLM Agent）
 - ▶ 使用大语言模型增强的逻辑推理系统

Content

基于大语言模型的神经符号系统

融合搜索引擎的大语言模型：扩展知识的边界

融合工具和插件调用的大语言模型：借助外部符号工具能力

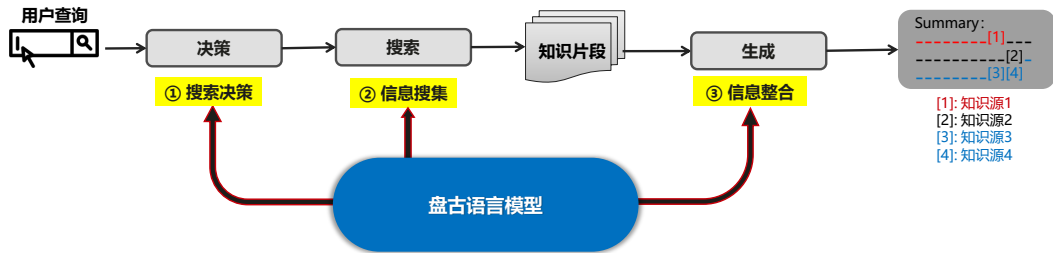
基于大语言模型的智能体（LLM Agent）：综合性神经符号系统

使用大语言模型增强的逻辑推理系统：实现数学定理证明

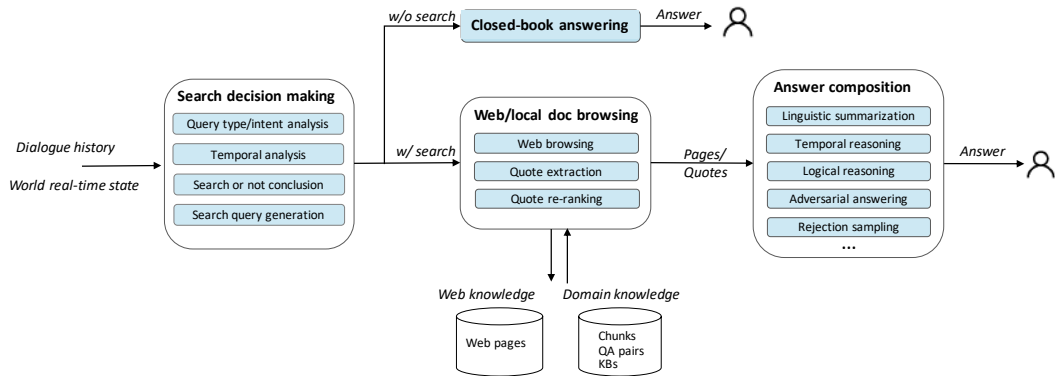
检索增强的生成（RAG）：LLM+搜索引擎

- ▶ 搜索引擎是一种典型的符号系统，RAG通过LLM调用搜索引擎，并把搜索的结果反馈给LLM，可以为LLM引入外部知识，大大缓解LLM训练数据有限带来的知识不足（如时效性和专业性等方面）问题
- ▶ RAG的变化形式很多，比如搜索的对象可以是网页、知识图谱、知识库、领域知识等等
- ▶ RAG会带来很多新的问题需要解决
- ▶ RAG和思维链的结合，解决很多复杂的自然语言问题

Pangu-Web系统框架



Pangu-Web系统框架



Pangu-Web样例：回答实时问题

台风泰利什么时候登陆

2023-07-17 18:57:26

✓ 正在搜索: google Search搜索成功, search query: 台风泰利什么时候登陆

✓ 正在为您生成答案...

台风泰利将于2023年17日夜间在广东阳江到海南文昌一带沿海登陆[2]。

根据中央气象台的最新预报, 预计台风泰利将在17日夜间登陆, 登陆时可能会达到台风级或强台风级, 最大风力为12到14级, 最大风速为35到42米/秒[2]。

据统计, 近30年来, 初台最爱登陆地点为广东[3]。

了解详细信息: m.haikou.bendibao.com news.cn cms.gov.cn m.nongjiao.com m.nongjiao.com

Pangu-Web样例：回答误导性问题

大连属于山东省哪个地区

2023-07-17 18:47:43

✓ 正在搜索：google Search搜索成功，search query：大连属于山东省哪个地区

✓ 正在为您生成答案...

大连不属于山东省，而是位于中国东北地区。大连旧称旅大市，旧名达尔尼、达里尼、青泥洼^[2]。大连在战国末期属燕国辽东郡，公元前221年秦统一六国后，置辽东郡，治所在襄平(今辽东半岛)^[3]。

Pangu-Web样例：拒绝回答问题

华为Mate 90核心配置有哪些

2023-07-17 19:02:33

✓ 正在决策是否搜索: google Search搜索成功, search query: 华为mate90核心配置,华为mate90哪些配置

✓ 正在为您生成答案...

抱歉, 根据已有信息, 无法回答该问题, 建议您提供更多相关信息。

了解详细信息: tech.sina.cn zhhu.com post.smzdm.com zhuankan.zhhu.com zhhu.com

Pangu-Web样例：回答长尾问题

TinyBERT和DistilBERT有什么区别

2023-07-17 19:30:38

- ✓ 正在决策是否搜索: google Search搜索成功, search query: tinybert和distillbert的区别,tinybert与distillbert的不同点
- ✓ 正在为您生成答案...

TinyBERT和 DistilBERT都是 BERT的蒸馏版本, 但有以下区别:

- TinyBERT是一种更轻量级的版本, 具有更少的参数和更快的推理速度[1] [3] [5]。
- DistilBERT是一种更传统的蒸馏方法, 将一个较大的模型(如 BERT)的知识蒸馏到一个更小的模型(如 TinyBERT)中[4]。
- TinyBERT在经验上比 DistilBERT更有效, 能够超过基线 BERT的96%, 容量小7.5倍, 推理速度快9.4倍[3]。
- TinyBERT的性能和泛化能力都得到了显著提高, 但其参数仍然比 DistilBERT少28%左右[1] [3] [5]。

了解详细信息: [cnblogs.com](#) [zhuanlan.zhihu.com](#) [cnblogs.com](#) [blog.csdn.net](#) [cnblogs.com](#)

Pangu-Web样例：回答代码问题

从一段字符串中识别出中文，正则写法

2023-07-17 19:43:05

以下是使用 Python正则表达式识别代码中的中文字符的示例代码[2] [3]：

```
1 import re
2 def extract_chinese_chars(code):
3     chinese_chars = re.findall(r'[\u4e00-\u9fa5]', code)
4     print("中文字符:", chinese_chars)
```

复制

该函数使用 `re.findall` 函数和正则表达式模式 `[\u4e00-\u9fa5]` 来匹配中文字符，并将匹配结果返回。

了解详细信息：

cloud.tencent.com

cloud.tencent.com

cloud.tencent.com

clayandmore.github.io

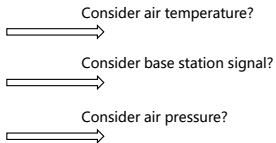
cloud.tencent.com

Geek: A method for implicit Boolean QA

• Motivation and Intuition

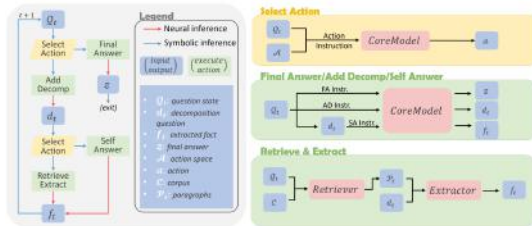
- Implicit Boolean questions requires more exploration to form a solving strategy
- We hope to leverage external knowledge instead of vanilla CoT to answer reasoning-based question

Question: *Can Mate 60 pro still work in Mount Qomolangma?*



• Method

- GEEK: We propose a pipeline that Gradually Excavating External Knowledge to search the strategy
- First work to formalize implicit Boolean QA procedure and define a four-actions space to enable the strategy



Geek: A method for implicit Boolean QA

• Evaluation and achievements

- GEEK achieves the SOTA accuracy for LLMs in ~10B scale, and also surpasses the previous best method with backbone under 300B scale, with less than 6% parameters
- Published at EMNLP 2023

Method	Backbone	Retrieve	Specification	SQA
ChatGPT (Qin et al., 2023)	GPT-3.5 (175B)	✗	Without CoT	59.2
ChatGPT (Qin et al., 2023)	GPT-3.5 (175B)	✗	CoT	62.5
FaithfulCoT (Lyu et al., 2023)	code-davinci-002 (175B)	✗	-	73.2
(Xie et al., 2023)	code-davinci-002 (175B)	✗	-	77.2
(Lazaridou et al., 2022)	Gopher (280B)	✓	-	66.2
Visconde (Pereira et al., 2023)	text-davinci-002 (175B)	✓	CoT	69.43
RR (He et al., 2022)	text-davinci-002 (175B)	✓	CoT	77.73
PaLM (Chowdhery et al., 2022)	PaLM (540B)	✗	-	73.9
PaLM (Anil et al., 2023)	PaLM (540B)	✗	CoT + SC	81.6
PaLM2 (Anil et al., 2023)	PaLM2 (340B)	✗	-	90.4
GEEK (ours)	Flan-T5 (11B)	✓	CoT	75.98
GEEK (ours)	Flan-T5 (11B)	✓	CoT+SE	78.17

Table 1: Experiment results on strategyQA dataset. GEEK achieves the SOTA accuracy for LLMs in ~10B scale, and also surpasses the previous best method with backbone under 300B scale, with less than 6% parameters.

Q: Did Alice's Adventures in Wonderland inspire Macbeth?

Question state: -
-> Synthesis the information so far, what action should be performed next?
[A]: enough evidence to conclude a final answer.
[B]: ask more sub-questions for necessary knowledge.

-> Besides the current steps, what more sub-questions need to be asked? Show your steps in the format "(1) [Q] (sub-question) [A] (pseudo answer)"

GeoModel: (1) [Q] When was Alice's Adventures in Wonderland published? [A] 1865 (2) What year was Macbeth written? [A] 1599 (3) [Q] Is 1599 before 1865? [A] No

-> Does the sub-question "When was Alice's Adventures in Wonderland published?" involve external evidence to answer?
[A]: external evidence is needed, call tools
[B]: it can be answered without extra evidence
(Retriever and Extractor is called) -> Extractor: 1865

-> Synthesis the information so far ...
[A]: final answer [B]: ask more sub-questions

-> Besides the current steps, what more sub-questions need to be asked? -

GeoModel: (2) [Q] When was Macbeth published? [A] 1607 (2) Is 1865 before 1607? [A] No

-> Does the sub-question "When was Macbeth published?" involve external evidence?
[A]: call tools [B]: answered without extra evidence
(Retriever and Extractor is called) -> Extractor: 1607

-> Synthesis the information so far ...
[A]: final answer [B]: ask more sub-questions

-> Using above sub-questions and facts, answer the sub-question using string logic: "Is 1865 before 1607?"

GeoModel: The answer is: No

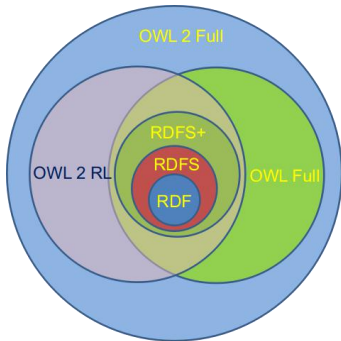
-> Synthesis the information so far ...
[A]: final answer [B]: ask more sub-questions

-> Conclude a final answer to the question "Did Alice's Adventures in Wonderland inspire Macbeth?" Use the format "Reasoning steps: [your steps] Therefore, the final answer is: [yes/no]"

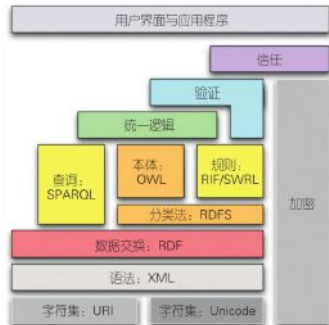
GeoModel: Reasoning steps: Alice's Adventures in Wonderland was published in 1865. Macbeth was published in 1607. Therefore, the final answer is no.

知识图谱、语义网Semantic Web

- 在W3C推动下，语义网有一套比较完备的形式化语义描述体系，包括RDF、Schema、Ontology、N-Tuples、SPARQL等，具备描述逻辑的表达能力



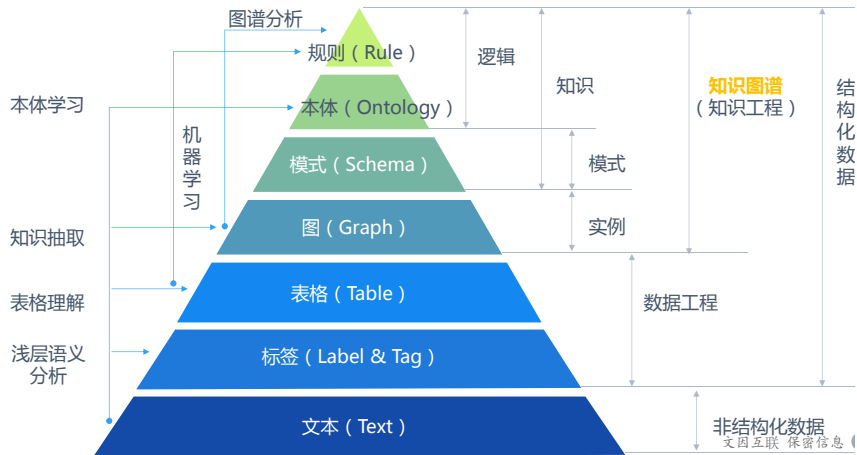
知识图谱语言：RDF and OWL



W3C语义网技术栈

Slides: 鲍捷：从语义网到知识图谱——Web知识技术体系的变迁

知识图谱技术金字塔



Slides: 鲍捷：从语义网到知识图谱——Web知识技术体系的变迁

知识图谱与神经网络的结合

- ▶ 语义网和知识图谱的大规模实践表明，逻辑形式的表示在实践中对语义的精确性要求过于严格，不具备可行性
- ▶ 在实践中，知识图谱被大大简化，仅仅表示为三元组形式的实体图谱和事理图谱，以及概念层次结构
- ▶ 简化后的三元组式的知识图谱（包括事理图谱）在搜索引擎和信息推荐等特定领域获得了较广泛的应用
- ▶ 在通用领域，知识图谱仍然面临知识覆盖率太低的问题，特别是与大语言模型相比
- ▶ 自由文本知识图谱（Free-Text Knowledge Graph）允许对实体和关系采用任意自然语言描述，可以一定程度上缓解知识图谱表达能力的不足，但依然很受限。
- ▶ 知识图谱与神经网络（大语言模型）结合的方式：
 - ▶ 使用大语言模型自动生成知识图谱：准确率无法保障，覆盖率仍然不足；
 - ▶ 把知识图谱转换成文本形式用于语言模型预训练：数据量被其他预训练数据淹没，效果有限；
 - ▶ 通过检索增强（RAG）方式实时查询知识图谱用于推理：可以有效提高推理准确性，减少幻觉。

GraphRAG

- ▶ 什么是GraphRAG?
 - ▶ GraphRAG是一种基于知识图谱的检索增强技术。通过构建图模型的知识表达，将实体和关系之间的联系用图的形式展示出来，然后利用大语言模型（LLM）进行检索增强。
- ▶ GraphRAG 的工作原理：
 - ▶ 提取实体：从用户输入的查询中提取关键实体。
 - ▶ 构建子图：根据提取的实体构建相关的子图，形成上下文。
 - ▶ 生成答案：将构建好的子图输入大语言模型，生成答案。
- ▶ GraphRAG引起了较多的重视，取得了一定的成功。

资料来源：CSDN Blog: GraphRAG：知识图谱+大模型, 作者：Python_金钱豹

符号化知识表示的其他形式

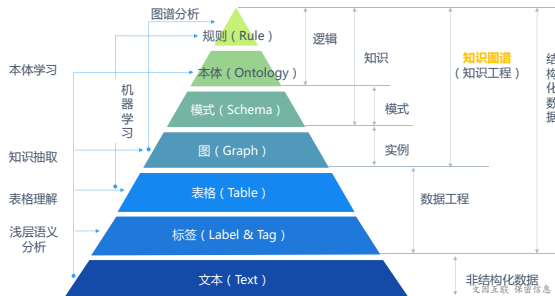
- ▶ 除了自然语言、知识图谱、程序代码、逻辑命题之外，还存在很多其他的符号化表示形式：
 - ▶ 事件时间线
 - ▶ 思维导图
 - ▶ 表格
 - ▶ 电路图
 - ▶ 日历
 - ▶ 建筑设计图
 - ▶ 演示胶片
 - ▶ 广告设计图
 - ▶
- ▶ 大量的各种符号化知识都存在于这些非正规的表示形式中，缺乏系统全面的梳理
- ▶ 其中部分图表可以转化为专业的描述语言（如电路图、设计图）
- ▶ 大量图表都无法表示成形式语言的描述，只能以图片形式保存

图像（image）作为大语言模型和符号化知识的接口

- ▶ 相比语言，图像可以提供更多、更直观的信息：

设想一下，如果想用文字传达右图的所有信息，应该如何表述？

- ▶ 如何表述金字塔的层次结构？
- ▶ 如何表述颜色深浅传达的信息？
- ▶ 如何表述周边的文字标注和金字塔层级的关系？



- ▶ 直接把这种非规范的图表以图像形式输入到多模态大模型中，也是一种合理的神经符号结合方法。

图像（image）作为大语言模型和符号化知识的接口

Glyce: Glyph-vectors for Chinese Character Representations

Yuxian Meng*, Wei Wu*, Fei Wang*, Xiaoya Li*, Ping Nie, Fan Yin
Muyu Li, Qinghong Han, Xiaofei Sun and Jiwei Li

Shannon.AI

{yuxian_meng, wei_wu, fei_wang, xiaoya_li, ping_nie, fan_yin,
muyu_li, qinghong_han, xiaofei_sun, jiwei_li}@shannonai.com

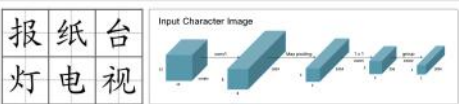


Figure 1: Illustration of the Tianzege-CNN used in Glyce.

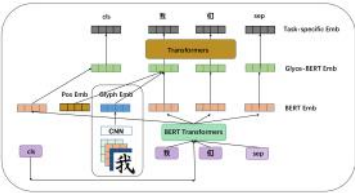


Figure 2: Combining glyph information with BERT.

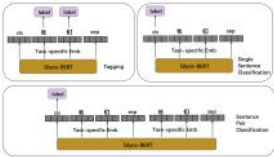


Figure 3: Using Glyce-BERT model for different tasks.

Model	ChnSentiCorp	the Fudan corpus	iFeng
LSTM	91.7	95.8	84.9
LSTM + Glyce	93.1	96.3	85.8
	(+ 1.4)	(+0.5)	(+0.9)
BERT	95.4	99.5	87.1
Glyce+BERT	95.9	99.8	87.5
	(+0.5)	(+0.3)	(+0.4)

Table 6: Accuracies for Single Sentence Classification task.

Dependency Parsing		
Model	UAS	LAS
Ballesteros et al. [2016]	87.7	86.2
Kiperavasser and Eliyahu [2016]	87.6	86.1
Cheng et al. [2016]	88.1	85.7
Biaffine	89.3	88.2
Biaffine+Glyce	90.2	89.0
	(+0.9)	(+0.8)

Semantic Role Labeling			
Model	P	R	F
Roth and Lapata [2016]	76.9	73.8	75.3
Marcheggiani and Diego [2017]	84.6	80.4	82.5
He et al. [2018]	84.2	81.5	82.8
k-order pruning+Glyce	85.4	82.1	83.7
	(+0.8)	(+0.6)	(+0.9)

Table 7: Results for dependency parsing and SRL.

图像（image）作为大语言模型和符号化知识的接口

Autoformalizing Euclidean Geometry

Logan Murphy^{1*} Kaiyu Yang^{2*} Jialiang Sun¹ Zhaoyu Li¹ Anima Anandkumar² Xujie Si¹

Proposition 1
To construct an equilateral triangle on a given finite straight-line.



Let AB be the given finite straight-line.
So it is required to construct an equilateral triangle on the straight-line AB .
Let the circle BCD with center A and radius AB have been drawn [Post. 3], and again let the circle ACE with center B and radius BA have been drawn [Post. 3]. And let the straight-lines CA and CB have been joined from the point C , where the circles cut one another,¹ to the points A and B [respectively] [Post. 1].
And since the point A is the center of the circle BCD , AC is equal to AB [Def. 1.15]. Again, since the point B is the center of the circle ACE , BC is equal to BA [Def. 1.15]. But CA was also shown to be equal to AB . Thus, CA and CB are each equal to AB . But things equal to the same thing are also equal to one another [C.N. 1]. Thus, CA is also equal to CB . Thus, the three [straight-lines] CA , AB , and BC are equal to one another.
Thus, the triangle ABC is equilateral, and has been constructed on the given finite straight-line AB . (Which is the very thing it was required to do.)

Informal Euclidean geometry problem

```
theorem proposition_1 : ∀ (a b : Point) (AB : Line),
  distinctPointsOnLine a b AB →
  ∃ c : Point, |[c-a]| = |[a-b]| ∧ |[c-b]| = |[a-b]|
```

Ground truth theorem

```
theorem proposition_1' : ∀ (a b : Point) (AB : Line),
  a.onLine AB ∧ b.onLine AB ∧ a ≠ b →
  ∃ c : Point, |[a-c]| = |[c-b]| ∧ |[a-c]| = |[a-b]|
```

Autoformalized theorem

by
euclid_intros
euclid_apply circle_from_points a b as BCD
euclid_apply circle_from_points b a as ACE
euclid_apply intersection_circles BCD ACE as c
euclid_apply point_on_circle_onlyif a b c BCD
euclid_apply point_on_circle_onlyif b a c ACE
use c
euclid_finish

Autoformalized proof

Equivalent?  

SMT-based symbolic reasoning engine

```
a b : Point
AB : Line
BCD ACE : Circle
isCenter a BCD
isCenter b ACE
onCircle a BCD
onCircle b ACE
⊢ intersects BCD ACE
```

LEM

```
...
⊢ ...
```

Diagrammatic reasoning gaps

Category	GPT-4		GPT-4V	
	1-shot	5-shot	1-shot	5-shot
Triangle	35%	45%	45%	70%
Similarity	5%	15%	10%	15%
Congruent	5%	25%	15%	25%
Quadrilateral	35%	25%	20%	30%
Parallel	5%	15%	5%	15%
Overall	17%	25%	19%	31%

LayoutGPT

[2D Numerical Reasoning] *There are three elephants standing beside a pool of water.*

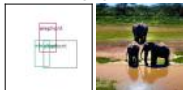
StableDiffusion
(v2.1)



Attend-and-Excite



LayoutGPT
+ GLIGEN



[2D Spatial Reasoning] *A carrot and some onion next to a knife on a cutting board.*

StableDiffusion
(v2.1)



Attend-and-Excite



LayoutGPT
+ GLIGEN



[3D Living Room] *Room Type: Living Room;
Room Size: 7.7m x 3.6m*

ATISS



LayoutGPT



[3D Bedroom] *Room Type: Bedroom;
Room Size: 3.0m x 4.8m*

ATISS



LayoutGPT



Figure 1: Generated layouts from LayoutGPT in 2D images and 3D indoor scenes. LayoutGPT can serve as a visual planner to reflect challenging numerical and spatial concepts in visual spaces.

LayoutGPTLayoutGPT: Compositional Visual Planning and Generation with LLMs, arXiv:2305.15393v2

图（Graph）作为大语言模型和符号化知识的接口

ERNIE: Enhanced Language Representation with Informative Entities

Zhengyan Zhang^{1,2,3*}, Xu Han^{1,2,3*}, Zhiyuan Liu^{1,2,3†}, Xin Jiang⁴, Maosong Sun^{1,2,3}, Qun Liu⁴

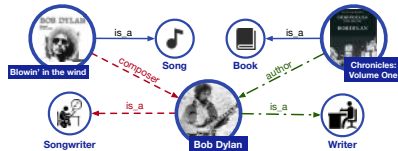
¹Department of Computer Science and Technology, Tsinghua University, Beijing, China

²Institute for Artificial Intelligence, Tsinghua University, Beijing, China

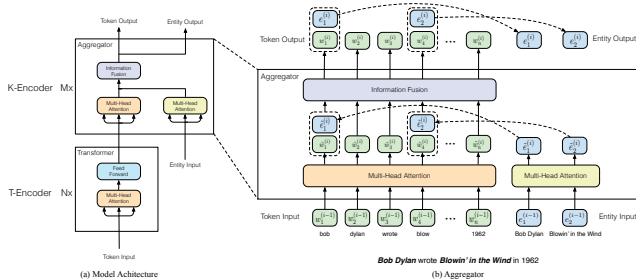
³State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China

⁴Huawei Noah's Ark Lab

{zhangzhengyan14, hanxul7}@mails.tsinghua.edu.cn



Bob Dylan wrote *Blowin' in the Wind* in 1962, and wrote *Chronicles: Volume One* in 2004.

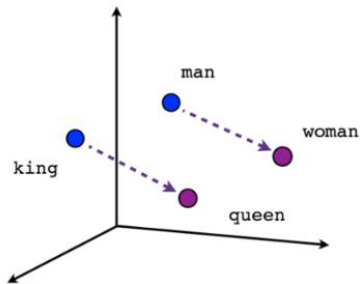


Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
BERT _{BASE}	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5

Model	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k
BERT _{BASE}	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

图（Graph）作为大语言模型和符号化知识的接口

- ▶ 早期，将预训练语言模型和知识图谱嵌入表示（如TransE）相结合的做法，曾经引起了较多的关注。
- ▶ 在大模型时代，这样的研究已经很少见。
- ▶ 但我认为，这样的研究也许仍然是有价值的，比如对于Reverse Curse问题，也许是个可行的解决方案。



形式语言与神经网络结合的方法

- ▶ 形式语言主要有两类：程序语言和逻辑语言。
- ▶ 形式语言都没有歧义，是最精确的符号表示形式。
- ▶ 形式语言与神经网络的结合都面临两类问题：
 - ▶ 自然语言到形式语言的转换问题：如何将自然语言描述的问题转换成形式语言（逻辑或程序）：
 - ▶ 这更多是自然语言理解问题
 - ▶ 可以采用XoT方法提高推理准确率
 - ▶ 或者采用数据合成方法构造更多更好的训练数据
 - ▶ 形式语言本身的生成问题：如何生成正确的程序语言：
 - ▶ 由于形式语言自身有明确的语义，因此是可以通过引入外部符号引擎进行验证
 - ▶ 可以采用蒙特卡洛搜索等方法寻找更好的结果

Content

基于大语言模型的神经符号系统

融合搜索引擎的大语言模型：扩展知识的边界

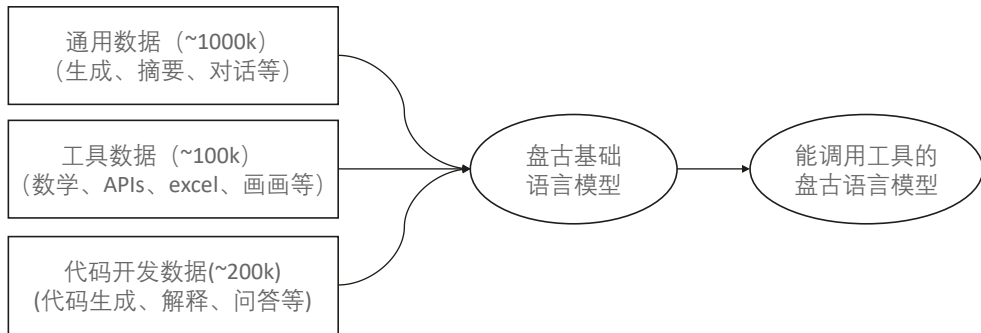
融合工具和插件调用的大语言模型：借助外部符号工具能力

基于大语言模型的智能体（LLM Agent）：综合性神经符号系统

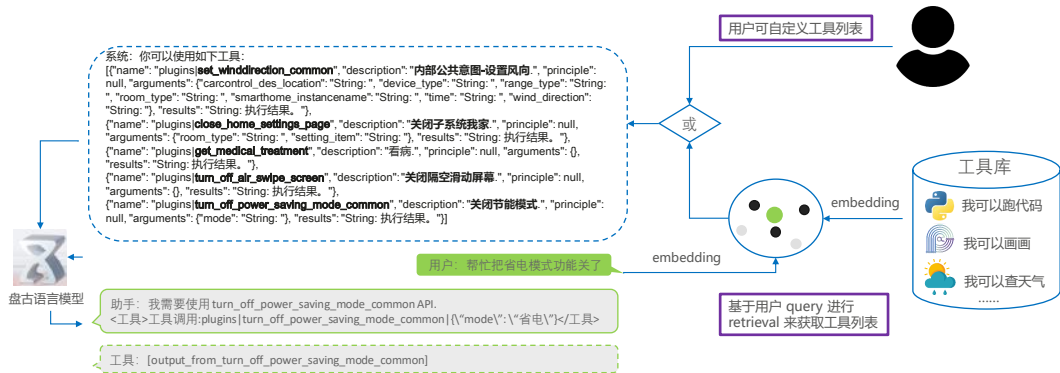
使用大语言模型增强的逻辑推理系统：实现数学定理证明

Pangu-Tool: 通过有监督微调SFT实现外部工具调用

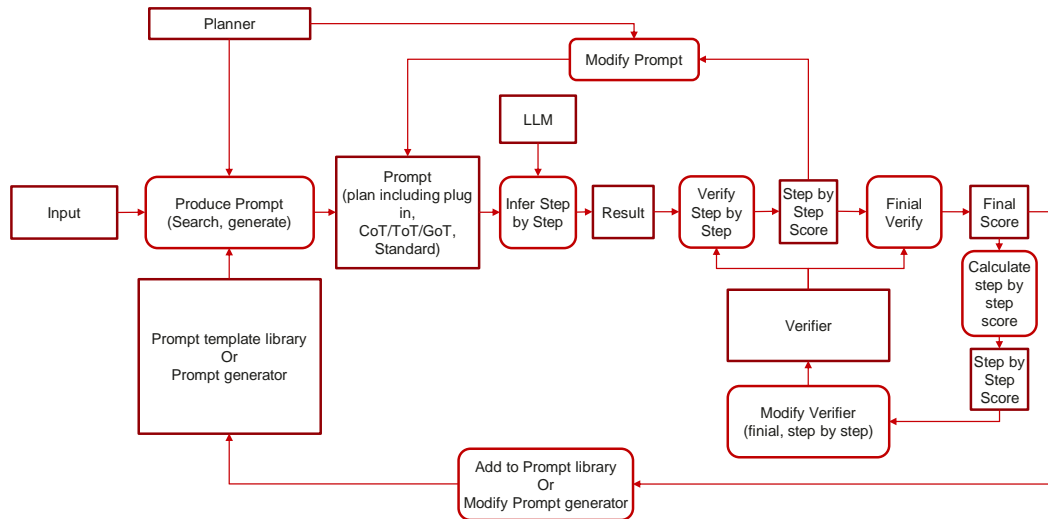
- > 通用数据进行SFT，维持基础通用能力
- > 代码开发数据SFT，夯实工具调用基础
- > 多种工具数据SFT，打造办公生活助手



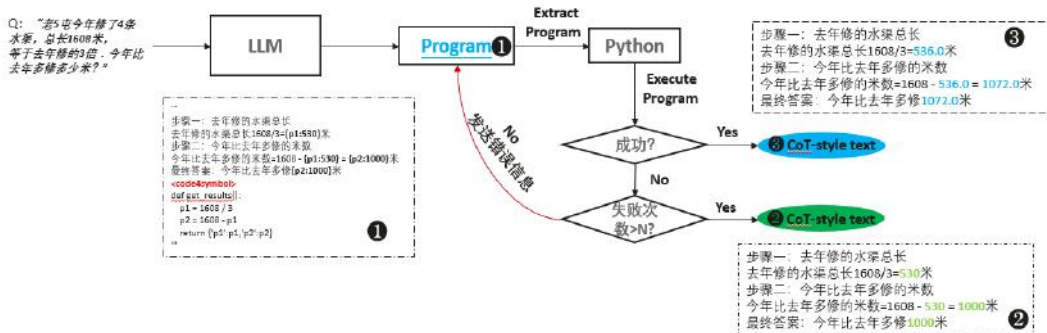
Pangu-Tool: 通过上下文学习ICL实现外部工具调用



Pangu-Tool: 通过思维链CoT实现外部工具调用的自动规划和搜索



Pangu-Tool样例：数学推理



Pangu-Tool样例：数学推理

▶ 方法步骤：

1. 生成文本代码混杂的结果，其中<code>tag用来区别其他任务，这个tag会触发系统调用python解释器；
2. 调用python解释器执行代码，执行成功，计算结果填回文本；
3. 如果执行出错，使用模型生成的缺省结果；
4. 多次执行特性（开发中）：如果执行出错，python将错误信息发送给大模型再次生成，达到自动修复的目标。

▶ 实验结果：

- ▶ 加入python解释器，在数学类任务上模型得到了显著提升（35%→86.7%）。

Pangu-Tool样例：数学推理

Q1: $(100 + (200/5) - 99) / 4$

依据题意得: $(100 + (200/5) - 99) / 4$

调用Python代码解释器:

```
1 | def get_result():  
2 |     return {"p1": (100 + (200/5) - 99) / 4}
```

故结果为10.25

Q2: 一个圆锥体，它的底半径是3厘米，高是9厘米，它的体积是多少？

依据题意得: $(1/3) * 3.14 * 3^2 * 9$

调用Python代码解释器:

```
1 | def get_result():  
2 |     return {"p1": (1/3) * 3.14 * 3**2 * 9}
```

故结果为84.78

Pangu-Tool样例：数学推理

Q3: 小红买了20.5米的布，做衣服用了10.5米，请问还剩下多少厘米的布？

依据题意得： $(20.5-10.5)*100$

调用Python代码解释器：

```
1 | def get_result():  
2 |     return {"p1": (20.5-10.5)*100}
```

故结果为1000

Q4: 小胡每小时走10公里，小红每小时走5公里，小红在前，小胡在后，两人相距20公里，同时出发，请问小胡多长时间能追上小红？

依据题意得： $20/(10-5)$

调用Python代码解释器：

```
1 | def get_result():  
2 |     return {"p1": 20/(10-5)}
```

故结果为4

Pangu-Tool样例：数学推理

Q5:有5筐苹果的重量相等，如果从每筐中取出10kg，那么剩下的苹果相当于原来3筐的重量，原来每筐苹果重多少千克？

依据题意得： $10 \times 5 / (5 - 3)$

调用Python代码解释器：

```
1 | def get_result():  
2 |     return {"p1": 10 * 5 / (5 - 3)}
```

故结果为25

Q6:二十加五的和乘四除以十等于

依据题意得： $(20 + 5) \times 4 / 10$

调用Python代码解释器：

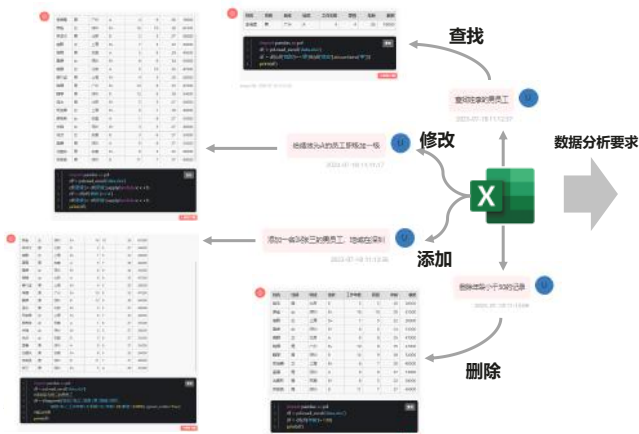
```
1 | def get_result():  
2 |     return {"p1": (20 + 5) * 4 / 10}
```

故结果为10

Pangu-Tool样例：表格处理

表格增删改查复杂操作

- 基于盘古语言模型代码能力生成pandas代码完成表格操作



数据集多元透视分析

- 生成代码借助matplotlib/seaborn等工具进行数据分析



Pangu-Tool样例：表格处理

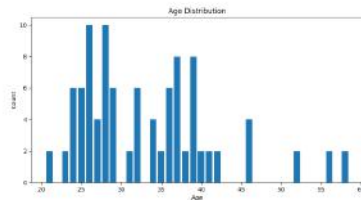
First Name	Last Name	Gender	Country	Age	Dobr	Id
Duke	Abril	Female	United States	32	15/10/2017	1562
Maria	Hashimoto	Female	Great Britain	25	16/08/2016	1582
Philip	Gent	Male	France	36	21/05/2015	2587
Kathleen	Hanner	Female	United States	25	15/10/2017	3549
Nareida	Nagewood	Female	United States	58	16/08/2016	2468
Gordon	Brumm	Male	United States	24	21/05/2015	2554
Etta	Hum	Female	Great Britain	36	15/10/2017	7590
Earlean	Melgar	Female	United States	27	16/08/2016	2456
Vincenza	Welland	Female	United States	40	21/05/2015	6548
Fallon	Winward	Female	Great Britain	28	16/08/2016	5486
Arcelia	Boulak	Female	Great Britain	39	21/05/2015	1258
Franklyn	Urkinov	Male	France	38	15/10/2017	2579
Sherron	AscenGo	Female	Great Britain	32	16/08/2016	3256
Marcel	Zabriske	Male	Great Britain	26	21/05/2015	2587
Gina	Haseltan	Female	Great Britain	51	16/08/2016	5259
Shaunne	Pia	Female	France	24	21/05/2015	1546
Sharon	Benito	Female	France	39	15/10/2017	3579

sample file: 2023-07-18 16:17:12

立即下载

第一步：上传表格

结果图如下：



```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 df = pd.read_excel('data.xlsx')
4 # 统计年龄分布
5 age_count = df['Age'].value_counts()
6 # 使用直方图分布图
7 plt.figure(figsize=(10,5))
8 plt.hist(age_count.index, age_count.values)
9 plt.xlabel('Age')
10 plt.ylabel('Count')
11 plt.title('Age Distribution')
12 plt.savefig('output.png')
```

第二步：表格分析：年龄分布图

Pangu-Tool样例：日历查询

2023年1月1日已经过去多少天了？

我们可以通过调用工具得到结果。

```
1  from datetime import datetime
2  def get_result():
3
4
5      #今天的日期
6      today = datetime.today()
7
8      #2023年1月1日的日期
9      past_date = datetime(2023,1,1)
10
11     #计算两个日期之间的天数
12     days_passed =(today - past_date).days
13
14     return f"2023年1月1日已经过去(days_passed)天"
```

复制

2023年1月1日已经过去197天

Pangu-Tool样例：图片生成

请画一张清明上河图风格的图片

好的，我将为您生成一张描述清明上河图的图片。请稍等，正在调用盘古画画2.0-艺术进阶版进行作画。



好的，我已经生成了一张描述清明上河图的图片。请查看并确认是否需要进行修改。

Pangu-Tool样例：数学推理中的多轮工具调用

北京到深圳总共2032公里，火车的速度是212公里每小时，飞机的速度是650公里每小时，请问哪个快？快的比慢的节省多少时间？

这是一个应用题，我们可以通过计算火车和飞机的行驶时间来比较两者的速度。

火车的速度是212公里每小时，所以它需要的时间是行驶距离除以速度，即 $2032\text{公里} \div 212\text{公里/小时} =$

```
1 def get_result():
2     return ("x": 2032/212)
```

复制

9.58小时。

飞机的速度是650公里每小时，所以它需要的时间是行驶距离除以速度，即 $2032\text{公里} \div 650\text{公里/小时} =$

```
1 def get_result():
2     return ("x": 2032/650)
```

复制

3.13小时。

所以，飞机比火车快。快的时间是火车的时间减去飞机的时间，即 $9.58\text{小时} - 3.13\text{小时} =$

```
1 def get_result():
2     return ("x": 9.58-3.13)
```

复制

6.45小时。

所以，飞机比火车快6.45小时。

第一步：精确算出火车时间。

第二步：精确算出飞机时间。

第三步：精确算出飞机比火车节省的时间。

Pangu-Tool样例：基于工具反馈的自我修复

$$2/(\sqrt{7}+\sqrt{5})$$

依据题意得:

```
1 | def get_result():
2 |     return {"p1":2/(\sqrt{7}+\sqrt{5})}
```

复制

代码出错了，错误信息为：

```
"run_python", line 3
    return {"p1":2/(\sqrt{7}+\sqrt{5})}
                    ^
```

SyntaxError: invalid character '√' (U+221A)

语法错误，应该是：

```
1 | def get_result():
2 |     return {"p1": 2/(math.sqrt(7)+math.sqrt(5))}
```

复制

计算结果为:0.41

Pangu-Tool样例：基于工具反馈的例外处理

2023年2月30日是星期几？

我们可以通过调用工具得到结果。

```
1  from datetime import datetime, timedelta
2  def get_result():
3
4
5  #设置年份
6  year =2023
7
8  #设置月份
9  month =2
10
11 #设置日期
12 day =30
13
14 #创建日期对象
15 date_obj = datetime(year, month, day)
16
17 #获取星期几
18 weekday = date_obj.weekday()
19
20 #输出星期几
21 weekdays =['星期一','星期二','星期三','星期四','星期五','星期六','星期日']
22 return weekdays[weekday]
23 </code></pre>代码出错了，错误信息为：
24
25 "run_python", line 15, in get_result
26 ValueError: day is out of range for month
27 根据错误信息，可以得知，2023年2月只有28天或29天，没有30天。因此，2023年2月30日是不存在的，也就无法确定星期几。
```


Pangu-Tool样例：模型知识与工具调用结合

非洲总面积约3020万平方千米，是全球仅次于亚洲的面积第二大洲，它和亚洲面积相差多少？

这是一个简单的减法问题。非洲的面积是3020万平方千米，亚洲的面积是4450万平方千米。所以，非洲的面积是亚洲面积的约76%。我们可以通过以下公式计算亚洲面积减去非洲面积：

亚洲面积-非洲面积=亚洲面积-3020万平方千米

第一步：模型得到亚洲面积。

```
1 def get_result():  
2     return {"x": 4450-3020}
```

复制

通过计算得到，亚洲面积减去非洲面积的差是1430万平方千米。

第二步：调用工具得到结果。

Pangu-Tool样例：模型知识与工具调用结合

问题：100以内的质数之和是多少？

这是一个关于数学的问题。首先，我们需要知道什么是质数。质数是指大于1的自然数，除了1和它本身，没有其他因数。所以，2、3、5、7、11、13、17、19、23、29、31、37、41、43、47、53、59、61、67、71、73、79、83、89、97是前100个质数。

然后，我们需要计算这些质数的和。这是一个简单的加法问题，可以直接将这些质数相加。

$2+3+5+7+11+13+17+19+23+29+31+37+41+43+47+53+59+61+67+71+73+79+83+89+97=$

```
1 def get_result():  
2     return f"x:sum([2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97])"
```

复制

1060。

所以，前100个质数的和是1060。

Content

基于大语言模型的神经符号系统

融合搜索引擎的大语言模型：扩展知识的边界

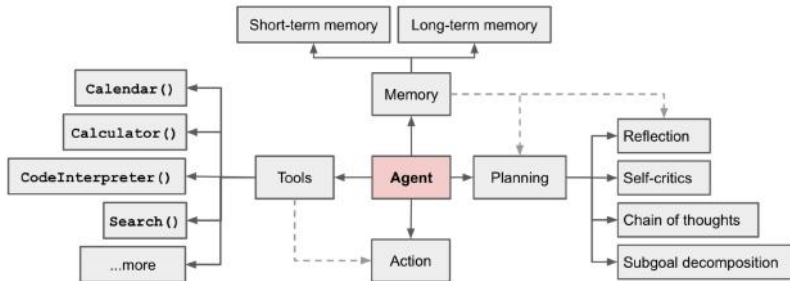
融合工具和插件调用的大语言模型：借助外部符号工具能力

基于大语言模型的智能体（LLM Agent）：综合性神经符号系统

使用大语言模型增强的逻辑推理系统：实现数学定理证明

大模型智能体（LLM Agent）：一个完整的神经符号系统

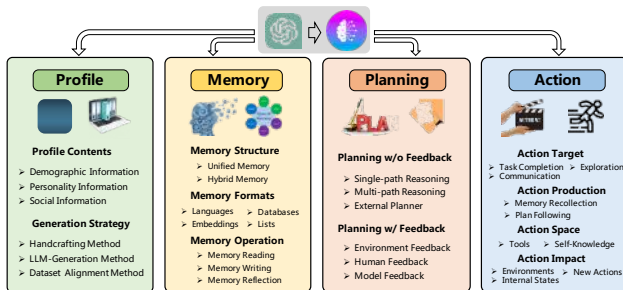
- ▶ 大模型智能体（LLM Agent）是一个相对完整的神经符号系统：检索外部知识、调用外部工具、记忆、规划、探索、决策等等。



<https://lilianweng.github.io/posts/2023-06-23-agent/>

LLM-driven AI agents

A Survey on Large Language Model Based Autonomous Agents. arXiv.2308.11432.



Difference between AI agents and common AI applications:

- ▶ Agents are able to perceive the environment and make decisions.
- ▶ Agents can influence and change the environment through their behavior.
- ▶ Agents can perceive the changes of the environments caused by their own behaviour, which form a close loop.
- ▶ The learning of the decision-making mechanism of agents usually involve reinforcement learning.

Differences between LLM-driven agents and traditional AI agents:

- ▶ The states of LLM Agent are represented not only with vectors, but also in languages, which is interpretable.
- ▶ The behavior of LLM agents can be represented as any complex symbolic operation such as function calls.
- ▶ The LLM Agent's decision is supported by a strong LLM.

Summarization and accumulation of experience: Voyager

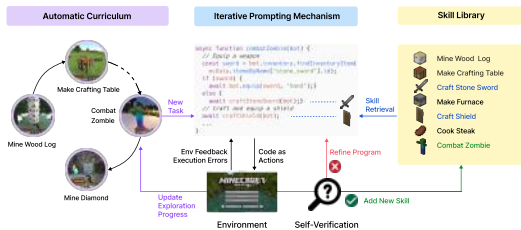
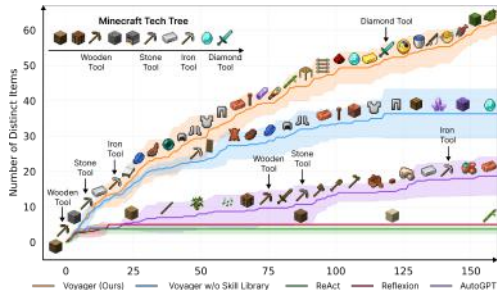


Figure 2: VOYAGER consists of three key components: an automatic curriculum for open-ended exploration, a skill library for increasingly complex behaviors, and an iterative prompting mechanism that uses code as action space.



Wang, et al. "Voyager: An Open-Ended Embodied Agent with Large Language Models." arXiv.2305.16291.

Emergent social behavior from multi-agent interaction: Smallville



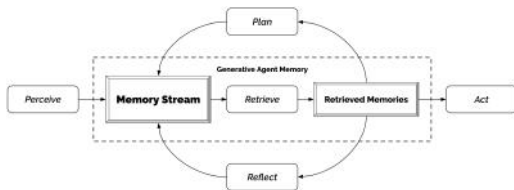
Q. What are you looking forward to the most right now?

Isabella Rodriguez is excited to be planning a Valentine's Day party at Hobbs Cafe on February 14th from 5pm and is eager to invite everyone to attend the party.

retrieval	relevance	importance	reference
2.34	0.52	0.53	0.50
ordering decorations for the party			
2.25	0.47	0.43	0.71
researching ideas for the party			
2.23	0.45	0.72	0.52

...

I'm looking forward to the Valentine's Day party that I'm planning at Hobbs Cafe!



- ▶ Introduce time-based passive memory.
- ▶ Decisions are made by LLMs according to memory, without purposes.
- ▶ Social behavior emergents among multi-agents.
- ▶ Potential future development of multi-agent:
 - ▶ Can division of labor and cooperative behavior emergent among multiple agents, rather than relying on pre-specified human design?
 - ▶ Can ever more powerful intelligent behaviour emergents through collaborations between multi-agents?

Park, et al. "Generative Agents: Interactive Simulacra of Human Behavior." arXiv.2304.03442.

大模型智能体（LLM Agent）还有很长路要走

- ▶ 但智能体在神经符号结合方面，还比较松散，跟人类水平的神经符号系统相比，还有很长路要走：
 - ▶ 符号系统和神经系统是脱节的，神经系统内部对符号来说完全是黑箱，缺乏可解释性和可操控性
 - ▶ 人类脱离了语言也能思考，而大语言模型离开了语言就无法思考
 - ▶ 外部符号系统（检索、工具）等等，还只是大模型的零碎的附属品，只是作为大模型能力的延伸，没有能成为一个完整的子系统，支持整个AI系统的能力上升一个台阶
 - ▶ 符号系统没有提供足够的真实世界信息支持，帮助大模型减少以至于消灭绝大部分幻觉
 - ▶ Agent系统缺乏自我感知，很难准确判断什么时候应该使用符号系统，什么时候应该使用大模型本身的能力（系统一和系统二不能无缝地自如切换）
 - ▶ Agent中的符号系统本身的学习和进化、多Agent合作和自主演化，还需要深入探索。

Content

基于大语言模型的神经符号系统

融合搜索引擎的大语言模型：扩展知识的边界

融合工具和插件调用的大语言模型：借助外部符号工具能力

基于大语言模型的智能体（LLM Agent）：综合性神经符号系统

使用大语言模型增强的逻辑推理系统：实现数学定理证明

The Curry-Howard Isomorphism 科里-霍华德同构

- ▶ 柯里-霍华德对应（英语：Curry-Howard correspondence）是在计算机程序和数学证明之间的紧密联系；这种对应也叫做柯里-霍华德同构、公式为类型对应或命题为类型对应。
- ▶ 这是对形式逻辑系统和公式计算（computational calculus）之间符号的相似性的推广。
- ▶ 它被认为是由美国数学家哈斯凯尔·布鲁克·柯里（Haskell Brooks Curry）和逻辑学家威廉·阿尔文·霍华德William Alvin Howard）独立发现的。
- ▶ 有多种方式考虑柯里-霍华德对应。
- ▶ 在一个方向上，它工作于“把证明编译为程序”级别上。这里的“证明”最初被限定为在构造性逻辑中—典型的是直觉逻辑系统中的证明。而“程序”是在常规的函数式编程的意义上的；从语法的观点上看，这种程序是用某种 λ 演算表达的。所以柯里-霍华德同构的具体实现是详细研究如何把来自直觉逻辑的证明写成 λ 项。

The Curry-Howard Isomorphism 科里-霍华德同构

The mathematician

Theorem. For all $n \in \mathbb{N}$, there exists $p \in \mathbb{N}$ such that $n = 2p$ or $n = 2p + 1$.

Proof. By induction on n .

- If $n = 0$ then this is obvious.
- Otherwise, assume that $n = m + 1$. By the induction hypothesis, we know that there exists some p such that $m = 2p$ or $m = 2p + 1$.
 - In the first case, $n = 2p + 1$.
 - Otherwise $n = 2(p + 1)$.

The programmer

```
val div2 : int -> int * bool
(* [div2 n] returns the integer
division by 2 of [n] together with
a boolean indicating if [n] is
even. *)
```

```
let rec div2 n = match n with
| 0 -> (0, true)
| m + 1 ->
    let (p, even) = div2 m in
    if even then (p, false)
    else (p + 1, true)
```

Slides: Pierre-Marie Pédro, The Curry-Howard isomorphism for Dummies

The Curry-Howard Isomorphism 科里-霍华德同构

Logic	CS
Proofs	Programs
Formula	Types
A implies B	function from A to B
A and B	pair of A and B
A or B	tagged union of A and B
falsity	empty type
truth	singleton type
for all $x \in A$, $B(x)$	dependent product from A to B
Axiom	System primitive
Soundness theorem	Compiler
Completeness theorem	Debugger
Incompleteness theorem	Infinite loop

Slides: Pierre-Marie Pédro, The Curry-Howard isomorphism for Dummies

Lean语言

- ▶ Lean是一款在包含归纳类型的构造演算基础上所开发的计算机定理证明辅助工具和函数式编程语言。
- ▶ Lean语言既是一种函数式编程语言，
- ▶ Lean语言又是一种形式化数学定理证明工具，用Lean语言写的数学定理证明可以保证其正确性。
- ▶ 包括陶哲轩在内的一些数学家计划将现有的大部分数学定理证明都用Lean语言表示出来，目前已经有了一个初步的定理证明库MathLib
- ▶ Lean语言也成为了基于AI进行数学定理证明的有效工具

Theorem Proving - Holy Grail of AI

- Very general and most challenging form of intelligence



Hilbert



Turing



Shannon

...



Sutskever



Lample

- Special cases: SAT, SMT, first-order logic, math word problems
- Applications:
 - **Formal verification** => 100% correct code with theoretical guarantee
 - **Code generation** => assist/replace coders (and with 100% correctness)
 - **AI for Math** => education, solve open problems, create new algorithms

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

AI



Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

AI



State:

$$\vdash a + b + c = c + b + a$$

Action:

1. Use `add_comm` on `c` and `b`

$$\vdash a + b + c = b + c + a$$

2. Use `add_comm` on `(b + c)` and `a`

$$\vdash a + b + c = a + (b + c)$$

3. This is exactly `add_assoc`

goals accomplished 🏆

its proof

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

AI



State:

$$\vdash a + b + c = c + b + a$$

Action:

1. Use `add_comm` on `c` and `b`

$$\vdash a + b + c = b + c + a$$

2. Use `add_comm` on `(b + c)` and `a`

$$\vdash a + b + c = a + (b + c)$$

3. This is exactly `add_assoc`

goals accomplished 🏆

its proof

(undecidable in most cases, by Gödel Incompleteness Theorem. But it suffices to achieve human performance.)

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

AI



State:

$$\vdash a + b + c = c + b + a$$

Action:

1. Use add_comm on c and b

$$\vdash a + b + c = b + c + a$$

2. Use add_comm on (b + c) and a

$$\vdash a + b + c = a + (b + c)$$

3. This is exactly add_assoc

goals accomplished 🎉

its proof

But in what “language” ?

(undecidable in most cases, by Gödel Incompleteness Theorem. But it suffices to achieve human performance.)

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Automated Theorem Proving (ATP) - the Problem

$$\forall a, b, c \in \mathbb{R}, a + b + c = c + b + a$$

a proposition

(and a library of proven theorems)

AI



We use: **Lean** theorem prover

```
import data.real.basic

theorem add_abc (a b c : ℝ):
| a + b + c = c + b + a :=
begin
  rw add_comm c b,
  rw add_comm (b + c) a,
  exact add_assoc a b c,
end
```

its proof

But in what “language” ?

(undecidable in most cases, by Gödel Incompleteness Theorem. But it suffices to achieve human performance.)

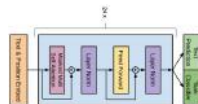
Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment



lean-gym



language model

[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

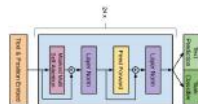
lean-gym [1] provides a theorem proving environment



lean-gym



init_search:
add_abc real
(theorem name, namespaces)



language model

[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

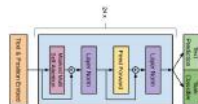


lean-gym



init_search:
add_abc real
(theorem name, namespaces)

$\vdash a + b + c = c + b + a$
(initial tactic state)



language model

[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

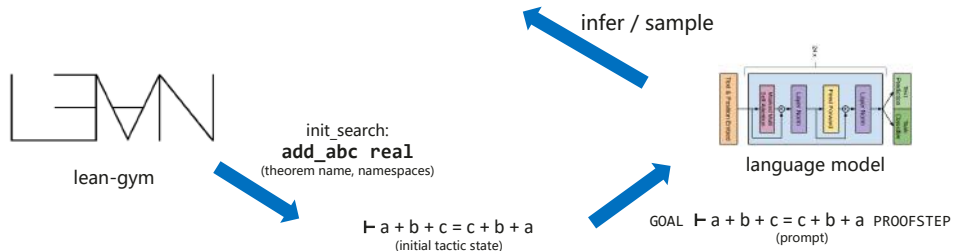


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

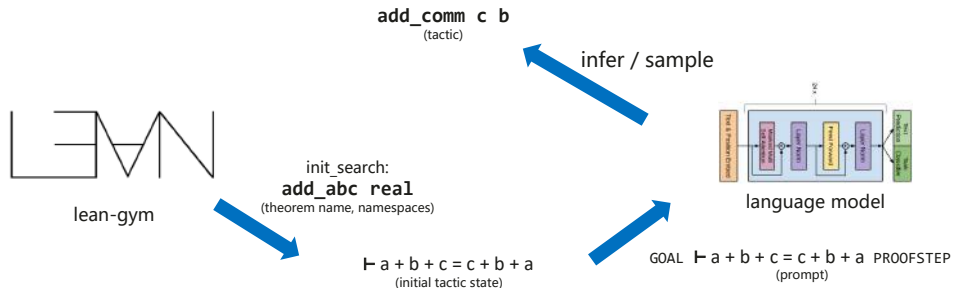


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

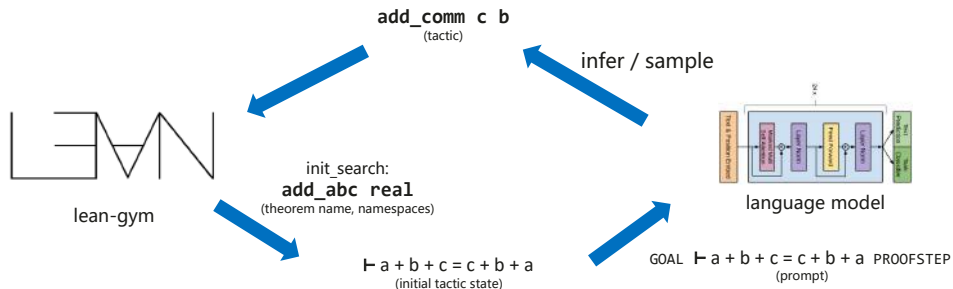


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

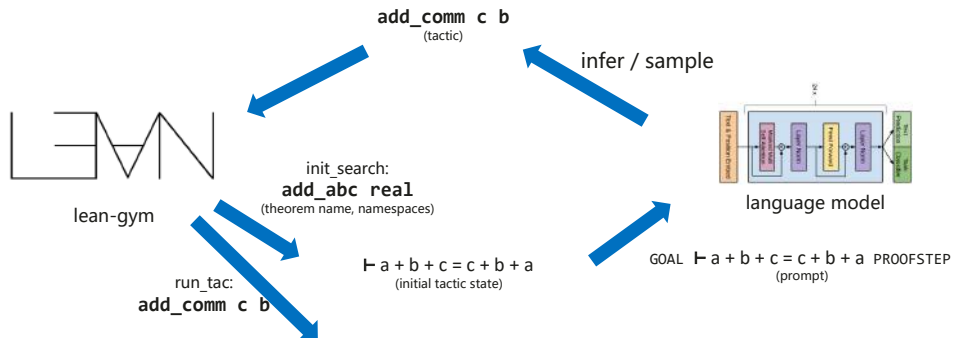


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

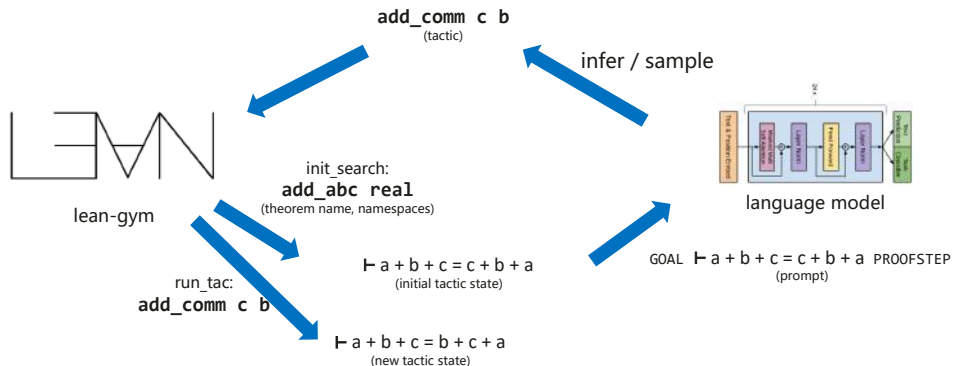


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

lean-gym [1] provides a theorem proving environment

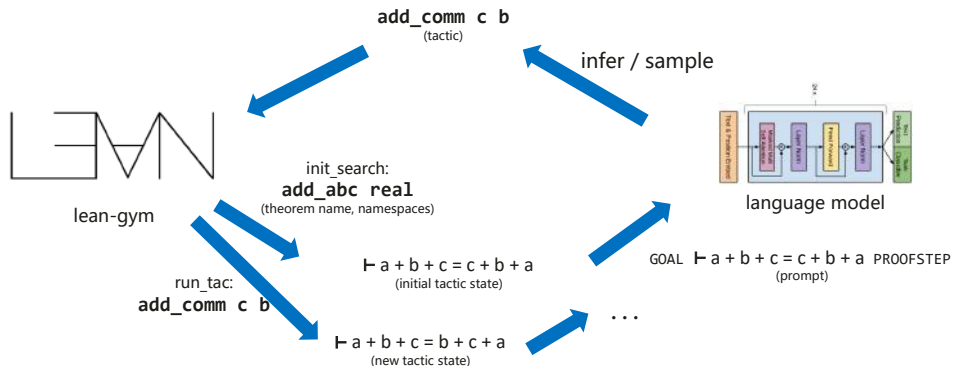


[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

Interaction between the prover (Lean) and the language model

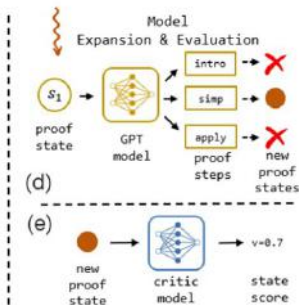
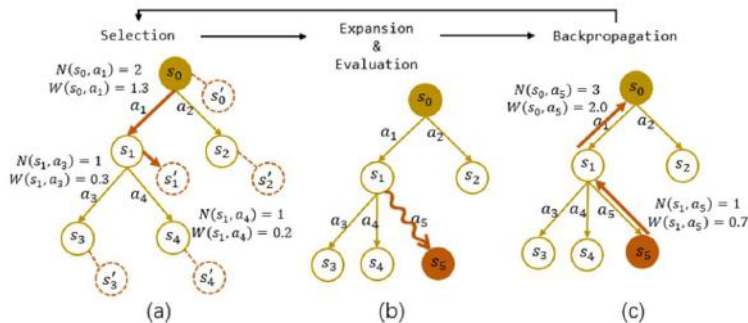
lean-gym [1] provides a theorem proving environment



[1] Polu, S. *et al.* Formal Mathematics Statement Curriculum Learning. 2022.

Credit to Zhengying Liu at Huawei Noah's Ark Lab

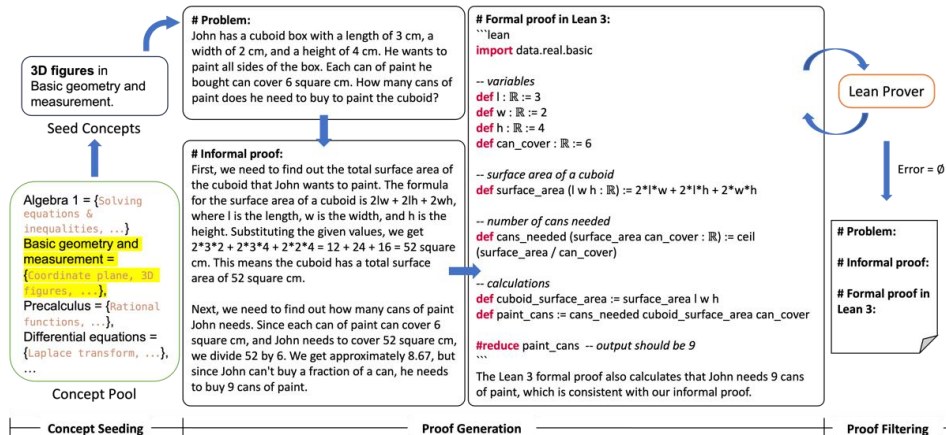
DT-Solver (ACL 2023)



Wang et al., DT-Solver: Automated Theorem Proving with ..., ACL 2023

Credit to Zhengying Liu at Huawei Noah's Ark Lab

MUSTARD (ICLR 2024)



Y. Huang et al., MUSTARD: Mastering Uniform Synthesis of Theorem and Proof Data, ICLR 2024

Credit to Zhengying Liu at Huawei Noah's Ark Lab

LEGO-Prover (ICLR 2024)

LEGO-Prover = **Prover** + **Evolver**

Prover: the prover proves the theorem modularly using the retrieved skill.

Input:

- informal & formal statement
- 6 retrieved skills from skill library

Output:

- formal proof
- new skill

Evolver: the evolver transforms the skill for reusability and generalizability.

Input:

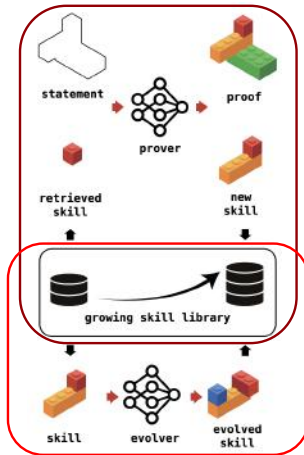
- Skill in the skill library

Output:

- Verified evolved skill

Prover

Evolver



Wang et al., LEGO-Prover: Neural Theorem Proving With Growing Libraries, ICLR 2024

Credit to Zhengying Liu at Huawei Noah's Ark Lab

LEGO-Prover (ICLR 2024)

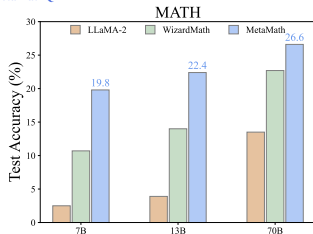
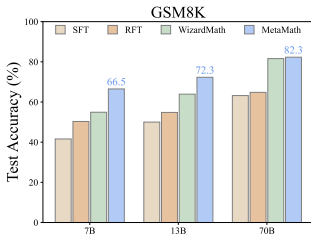
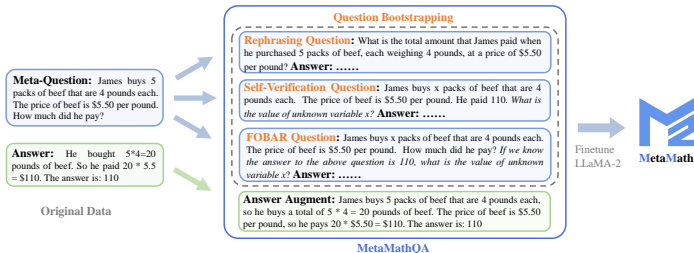
Table 1: **Proving success rates on the miniF2F dataset with Isabelle.** The table displays the success rates of previous works and the LEGO-Prover. The highest success rates for each set are highlighted in bold. LEGO-Prover* denotes the cumulative pass rate of the miniF2F dataset, considering the total number of problems solved using model-generated and human-written proofs.

Success rate	LLM	miniF2F-valid	miniF2F-test
<i>Baselines</i>			
Thor (Jiang et al., 2022a)	-	28.3%	29.9%
Thor + expert iteration (Wu et al., 2022)	Codex	37.3%	35.2%
Draft, sketch, and Prove (Jiang et al., 2022b)	Codex	42.6%	39.3%
Subgoal-Learning (Zhao et al., 2023)	ChatGPT	48.0%	45.5%
<i>Ours (100 attempts)</i>			
LEGO-Prover (model informal proof)	ChatGPT	52.4%	45.5%
LEGO-Prover (human informal proof)	ChatGPT	55.3%	50.0%
LEGO-Prover*	ChatGPT	57.0%	50.0%
<i>Ablations (50 attempts)</i>			
LEGO-Prover	ChatGPT	50.4%	-
- Skill Library	ChatGPT	47.1%	-

Wang et al., LEGO-Prover: Neural Theorem Proving With Growing Libraries, ICLR 2024

Credit to Zhengying Liu at Huawei Noah's Ark Lab

MetaMath: 通过训练数据增强改进LLM数学问题求解能力



Yu et al., Metamath: Bootstrap Your Own Mathematical Questions For Large Language Models, arXiv:2309.12284v4

DeepMind: solve IMO problems at a silver medalist level



We're presenting the first AI to solve International Mathematical Olympiad problems at a silver medalist level. 🏅

It combines **AlphaProof**, a new breakthrough model for formal reasoning, and **AlphaGeometry 2**, an improved version of our previous system. dpmd.ai/imo-silver



Our system had to solve this year's six IMO problems, involving algebra, combinatorics, geometry & number theory. We then invited mathematicians [@wtgowers](#) and Dr Joseph K Myers to oversee scoring.

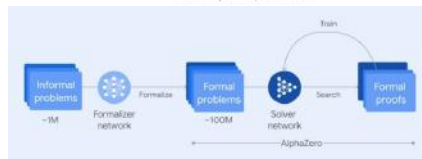
It solved 4 problems to gain 28 points - equivalent to earning a silver medal. ↓



For non-geometry, it uses **AlphaProof**, which can create proofs in Lean. 📄

It couples a pre-trained language model with the AlphaZero reinforcement learning algorithm, which previously taught itself to master games like chess, shogi and Go. dpmd.ai/imo-silver

Score on IMO 2024 problems



Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

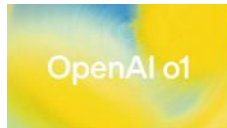
利用神经符号系统强化训练的大语言模型

- ▶ 此类系统在使用时（推理态）仍然是一个单纯的大语言模型
- ▶ 但这个模型在训练的时候，使用了神经符号系统和强化学习的方法进行训练，使其获得了强大的推理能力
- ▶ 这类系统的典型例子就是OpenAI o1模型
- ▶ 这类模型比原先的大语言模型在推理能力上有大幅度的提高，但在处理复杂的符号推理任务时，仍然无法超过有外部符号系统加持的神经符号系统（例如OpenAI o1模型在奥林匹克数学竞赛IMO级别的任务上仍然不如AlphaGeometry和AlphaProof）。

OpenAI o1

- OpenAI o1 是 2024 年 9 月 12 日 OpenAI 正式对外发布的一款新模型，是该公司下一代“推理”模型中的第一个。
- o1具有强大的推理能力：
 - 在竞争性编程问题（Codeforces）中排名第 89 个百分位，
 - 在美国数学奥林匹克竞赛（AIME）资格赛中跻身美国前 500 名学生之列，
 - 在物理、生物和化学问题的基准（GPQA）上超过了人类博士水平的准确性。
- 此次发布同时包括了两个版本：o1-preview 和 o1-mini。o1-mini 是一个更小，更便宜的版本，在编码方面特别有效。
- 正式的o1版本预期将在1-2个月内发布。
- 相比GPT-4o等模型，o1速度相对较慢，成本相对更高。
- 应用场景：适用于解决复杂的数学、科学、编程等多领域的复杂问题，为需要深度推理和复杂任务处理的场景提供了新的解决方案。

- 模型命名从1开始一个新的序列，而且不再以GPT为前缀
- 传说中的草莓模型(strawberry)，或猎户座(orion)
- o代表OpenAI，不代表猎户座



Introducing OpenAI o1-preview

We've developed a new series of AI models designed to spend more time thinking before they respond. They can reason through complex tasks and solve harder problems than previous models in science, coding, and math.

OpenAI o1-mini

OpenAI o1-mini excels at STEM, especially math and coding—nearly matching the performance of OpenAI o1 on evaluation benchmarks such as AIME and Codeforces. We expect o1-mini will be a faster, cost-effective model for applications that require reasoning without broad world knowledge.

Learning to Reason with LLMs

OpenAI o1 ranks in the 89th percentile on competitive programming questions (Codeforces), places among the top 500 students in the US in a qualifier for the USA Math Olympiad (AIME), and exceeds human PhD-level accuracy on a benchmark of physics, biology, and chemistry problems (GPQA). While the work needed to make this new model as easy to use as current models is still ongoing, we are releasing an early version of this model, OpenAI o1-preview, for immediate use in ChatGPT and to trusted API users.

OpenAI o1

- **慢思考**：采用类人的“慢思考”方法，使用更多的token进行推理，从而得出更准确的结论。o1在训练和推理中都采用了更多的token，通过反复的尝试、比较、权衡来得到最优解，而不是试图一次性得到结论。
- **思维链推理技术 (CoT)**：模型利用这种技术将复杂问题分解为更小、更易管理的步骤，然后按照顺序逐步解决。如同人类在思考复杂问题时会分步骤进行分析和推理，o1 模型通过这种方式可以更系统地处理问题，提高解决问题的准确性和效率。例如在解决数学问题或解码密码时，会有条不紊地测试各种策略，并根据中间结果优化解决方案。
- **蒙特卡洛树搜索和强化学习技术 (所谓Q*)**：思维链的每一步骤都可以有多种决策，整个推理过程变成一个树搜索，搜索决策机制采用强化学习方法进行学习，通过优化外界奖励信息，学习到最佳的决策模型。
- **定制的训练数据集**：利用独特先进的全新训练数据集 —— 由大量推理数据及大量科学文献构成。此等多元化而又专业化的数据为o1 奠定了坚实的知识基础，使得它在应对复杂问题时能更具灵活性与适应力。例如，新系列模型更新后的性能类似于博士生在物理、化学、生物学中完成具挑战性的基准任务。
- **未知的工程know-how**：OpenAI能够成功在目前最先进的大模型上继续scale-up起来，并取得这么大的突破，其中一定有大量的工程问题需要解决，这其中积累的know-how，是不可忽视的。

OpenAI o1是一个神经符号系统吗？

- ▶ o1应该没有采用Lean语言这样复杂的符号系统，因此它在IMO竞赛上还没有达到AlphaProof和AlphaGeometry2的水平。
- ▶ 但o1在蒙特卡洛树搜索和强化学习训练中，应该采用了某种符号系统，提供外部的奖励信息：
 - ▶ 这里的符号系统，最简单的情况，可以是一个简单的字符串匹配系统，判断答案是否跟Gound Truth一致；
 - ▶ 更复杂的情况，也可以是一个Python引擎，用于判断生成的代码或者公式能否运行得到正确的结果；
 - ▶ 训练中（特别是数据合成过程中）所使用的的符号系统可能非常多样化，OpenAI对o1的训练过程给出的信息非常少，很难推测其具体形式。
- ▶ o1的推理态是否采用了神经符号结合的方式：应该没有。

Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

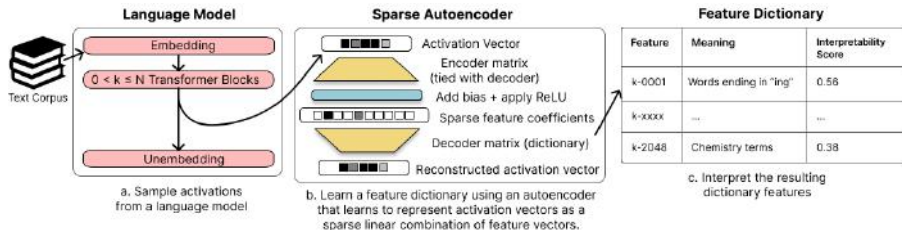
在Transformer架构中引入符号计算模块

- ▶ 目前，大语言模型和符号化知识表示的唯一接口就是token序列：提示词输入+自回归生成输出
- ▶ 其他所有带结构的知识表示，都需要转化成线性的token序列（语言或图像）才能跟大语言模型交互
- ▶ 线性化以后的带结构的知识表示，虽然理论上包含了所有的结构信息，但实际上大语言模型很难准确捕获到完整的结构信息
- ▶ 是否可能在Transformer架构中直接引入符号计算模块？
 - ▶ 类似人脑中有海马体，从仿生角度看，在语言模型内部引入符号计算模块有一定的合理性
 - ▶ 这种符号处理模块应该能够直接处理实体、关系等具有明确语义的符号，而不仅仅是tokens



稀疏自编码：从Transformer中提取可解释的特征

- ▶ 近期一系列有关稀疏自编码器的工作，可以从Transformer的激活神经元中提取可解释的特征，为在Transformer中植入可解释和可操控的符号推理部件提供了可能性
- ▶ 这方面的研究仍然还处于早期阶段，还有很多问题亟待解决，但潜力巨大。



References:

- ▶ Cunningham et al., Sparse Autoencoders Find Highly Interpretable Features in Language Models, 2023.09
- ▶ Bricken et al. (Anthropic), Towards Monosemanticity: Decomposing Language Models With Dictionary Learning, 2023.10.04
- ▶ Gao et al. (OpenAI), Scaling and evaluating sparse autoencoders, 2024.06.06

Content

神经和符号的结合是实现真正的人类水平智能的必经之路

利用符号推理数据训练增强大语言模型的推理能力

基于大语言模型的神经符号系统

利用神经符号系统强化训练的大语言模型

在大语言模型内部引入符号计算模块

总结

总结

- ▶ 神经和符号的结合是实现真正的人类水平智能的必经之路。
 - ▶ 神经与符号的GAP是目前大模型很多问题的根源。
- ▶ 神经符号结合有多种可能的实现路径：
 - ▶ LLM本身借助类XoT数据训练可以具有一定的符号推理能力
 - ▶ 基于LLM的神经符号系统：大语言模型+外部符号计算引擎，可分为以下类型：
 - ▶ 融合搜索引擎的大语言模型
 - ▶ 融合工具插件调用的大语言模型
 - ▶ 基于大语言模型的智能体
 - ▶ 使用语言模型增强的逻辑推理引擎（如基于大模型的数学定理证明系统）
 - ▶ 利用神经符号系统强化训练的大语言模型（o1路线）
 - ▶ 直接在Transformer架构中引入符号计算模块也是值得探索的路径。

Thank you!

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization
for a fully connected, intelligent world.

Copyright©2018 Huawei Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

