# Language Model Programming: Themes and Prospects

**Kyle Richardson**

May 2024

**Allen Institute for Artificial Intelligence** (AI2)

natural language processing today

# language models

# Large language models as general-purpose reasoners

# In-context learning: learning through examples

Instruction

simulate 

Examples

example behavior

ChatGPT as 

**USER**

I am a 5 year old struggling at math. Please answer questions like me.

Here are some examples of where I get things wrong:
question: what is 2 + 2
my answer: Umm... 7. This is hard!

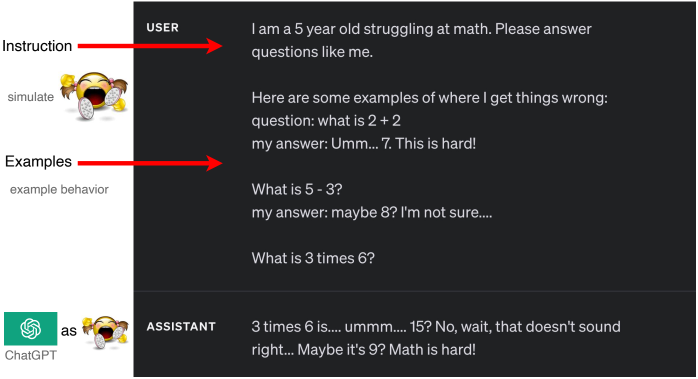
What is 5 - 3?
my answer: maybe 8? I'm not sure....

What is 3 times 6?

**ASSISTANT**

3 times 6 is.... ummm.... 15? No, wait, that doesn't sound right... Maybe it's 9? Math is hard!

# In-context learning: learning through examples



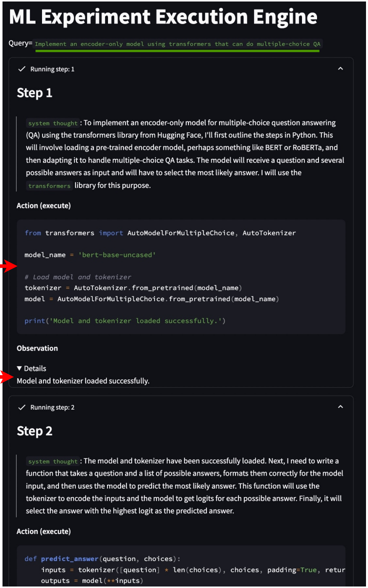

**Traditional learning** coupled with **in-context leaning** (no parameter updates, just new examples added to input)

# Language models as part of complex systems

ChatGPT

Model generated code

Code execution tool



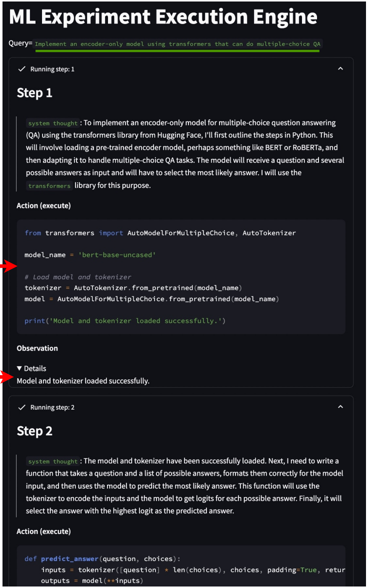

Can we leverage these tools to help automate and drive scientific discovery?

# Language models as part of complex systems

ChatGPT

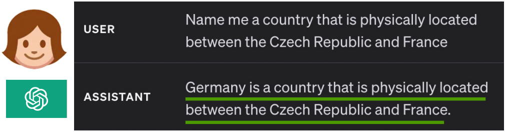
Model generated code

Code execution tool

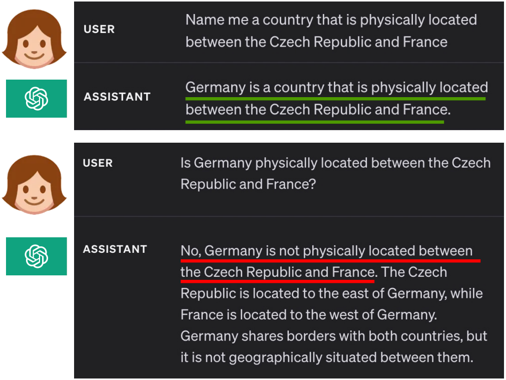



Language models with tools. **How to build such systems**?

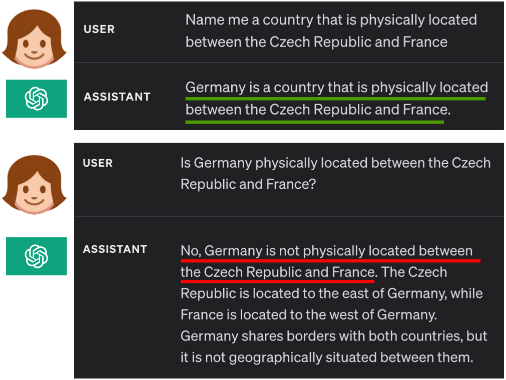
# Problems are still easy to find



# Problems are still easy to find



# Problems are still easy to find




Models can be wildly inconsistent, hallucinate, prompting is unstable (different results based on small changes).

# Problems are still easy to find

**USER**

Please tell me about the Tübingen AI center, including naming some of its core faculty and its address

**ASSISTANT**

The Tübingen AI Center (TAIC) is a leading research institution focused on artificial intelligence and machine learning. It is part of the University of Tübingen in Germany.

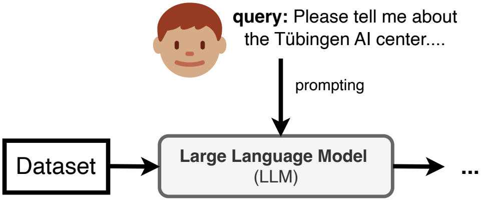
Some of the core faculty members at TAIC include:

1. Prof. Dr. Bernhard Schölkopf - Director of the Max Planck Institute for Intelligent Systems and Professor at the University of Tübingen
2. Prof. Dr. Andreas Zell - Professor of Bioinformatics and AI at the University of Tübingen
3. Prof. Dr. Michael A. Osborne - Professor of Machine Learning at the University of Tübingen

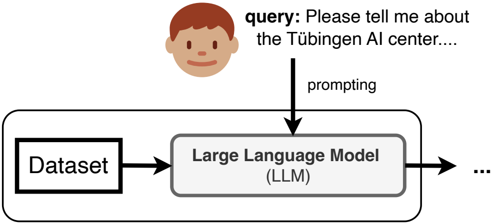
The address of the Tübingen AI Center is as follows:

Tübingen AI Center
Sand 13
72076 Tübingen
Germany

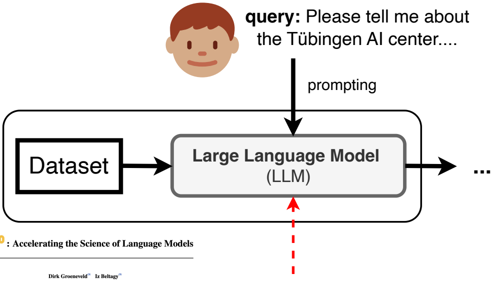
# The scale of models today

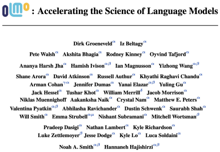


# The scale of models today



# The scale of models today



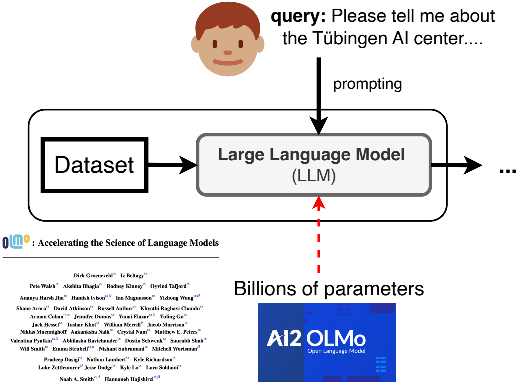


OLMo: Accelerating the Science of Language Models

Dirk Groeneveld Iz Beltagy
Pete Walsh Akshita Bhagia Rodney Kinney Oyvind Tafjord
Ananya Harsh Jha Hamish Ivison Ian Magnusson Yizhong Wang
Shane Arora David Atkinson Russell Authur Khyathi Raghavi Chandu
Arman Cohan Jennifer Dumas Yanai Elazar Yuling Gu
Jack Hessel Tushar Khot William Merrill Jacob Morrison
Niklas Muennighoff Aakanksha Naik Crystal Nam Matthew E. Peters
Valentina Pyatkin Abhilasha Ravichander Dustin Schwenk Saurabh Shah
Will Smith Emma Strubell Nishant Subramani Mitchell Wortsman
Pradeep Dasigi Nathan Lambert Kyle Richardson
Luke Zettlemoyer Jesse Dodge Kyle Lo Luca Soldaini
Noah A. Smith Hannaneh Hajishirzi

Billions of parameters

suite of 1B,7B,70B open-source models

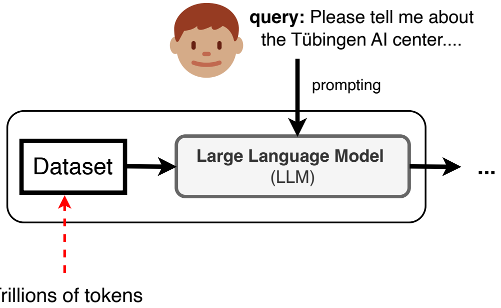
# The scale of models today




suite of 1B,7B,70B open-source models

- Open Language Model (OLMo) project (Groeneveld et al., 2024), open-source models, datasets, tools for LLMs, (Soldaini et al., 2024; Magnusson et al., 2023)

# The scale of models today

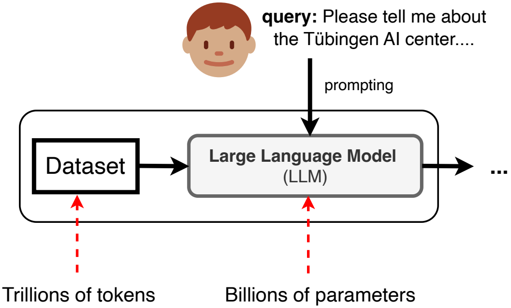



PALOMA : A BENCHMARK FOR EVALUATING LANGUAGE MODEL FIT

~3 trillion token pre-training corpus

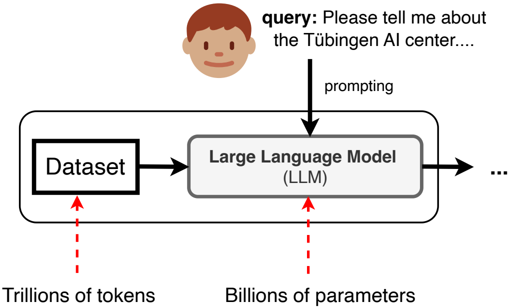
- Open Language Model (OLMo) project (Groeneveld et al., 2024), open-source models, datasets, tools for LLMs, (Soldaini et al., 2024; Magnusson et al., 2023)

# My research agenda



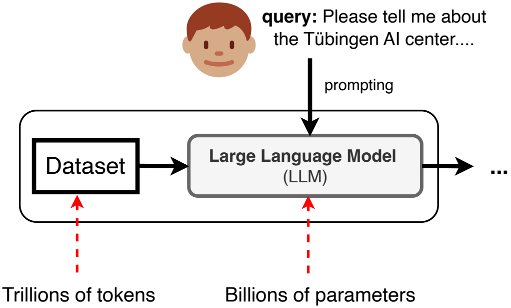

**Central theme** What do models know? How can we verify their correctness and reliability?

# My research agenda



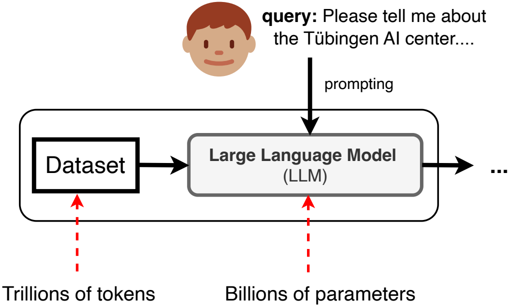

**General themes**: New open resources, programming paradigms and testing techniques for large language models.

# My research agenda




programming paradigms: frameworks for using and developing complex systems with LLMs, making their internal behavior more transparent.

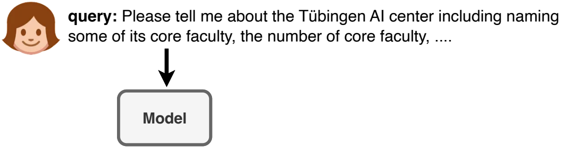
# General theme for today



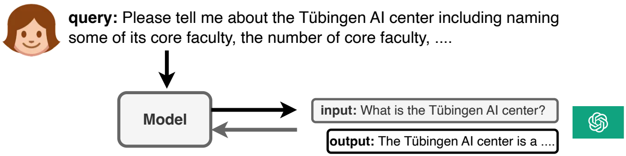

**Model Programming**: Building high-level programs on top of language models.

“programming” models

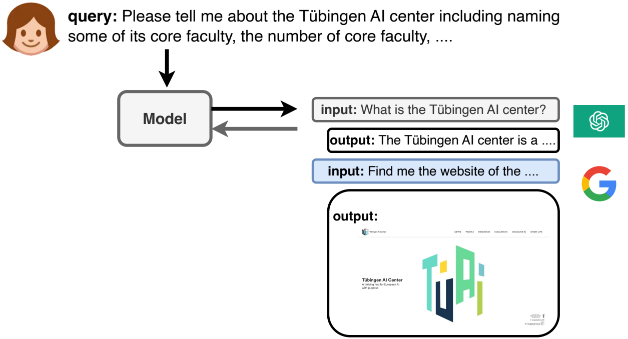
# Modular modeling: breaking problems into simple parts



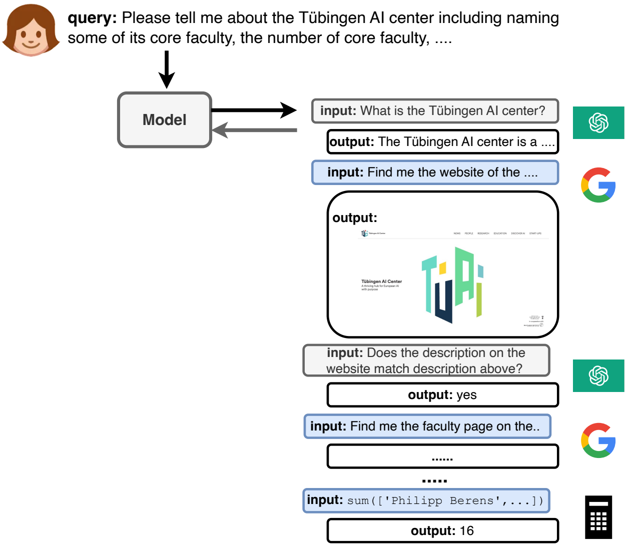
# Modular modeling: breaking problems into simple parts



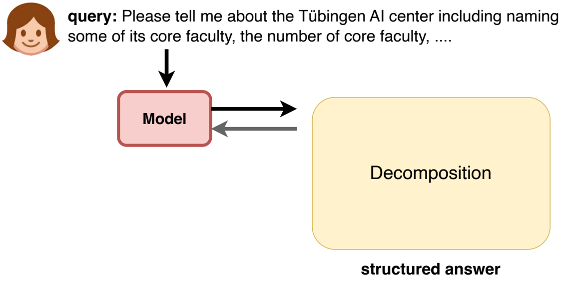
# Modular modeling: breaking problems into simple parts



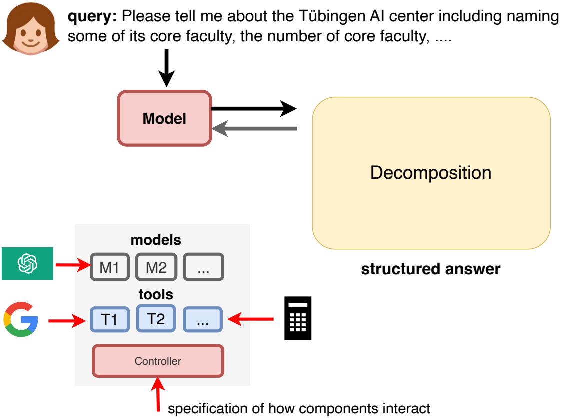
# Modular modeling: breaking problems into simple parts



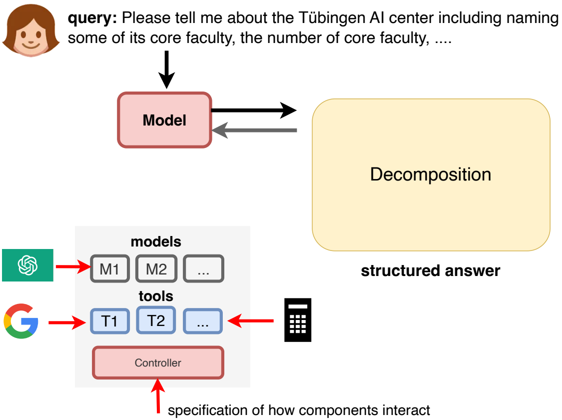
# Modular modeling: breaking problems into simple parts



# Modular modeling: breaking problems into simple parts

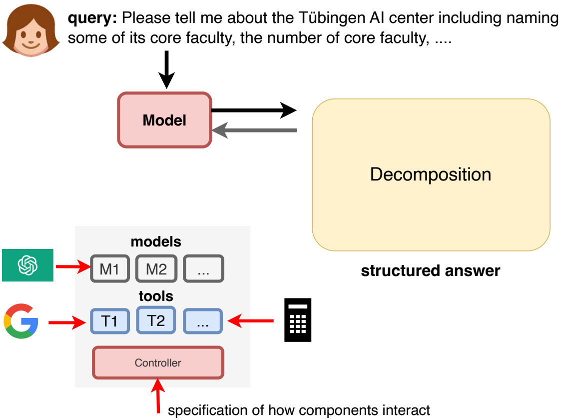


# Frameworks for modular modeling



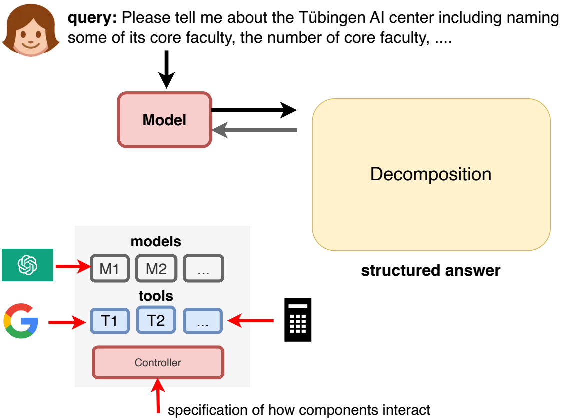

- **Frameworks**: text modular networks **[NAACL21]**, breakpoint transformers **[EMNLP2022]**, decomposed prompting **[ICLR23]**

# Frameworks for modular modeling



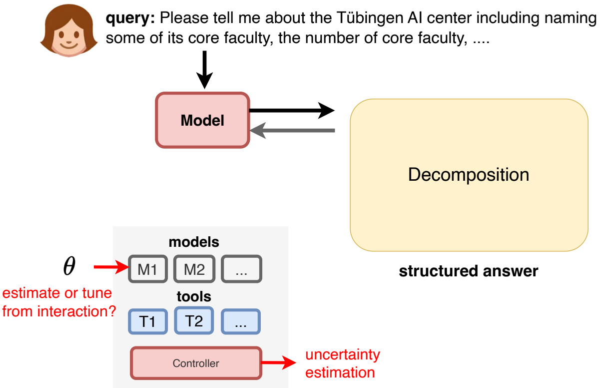

**Research questions**: what is the controller language, how do systems interact, how does inference and search work? Application-driven.

# Frameworks for modular modeling




paragraph QA, commonsense reasoning **[EMNLP22]**, argumentation modeling **[*SEM22]**, narrative understanding **[EMNLP22]**.

# A common feature of these frameworks

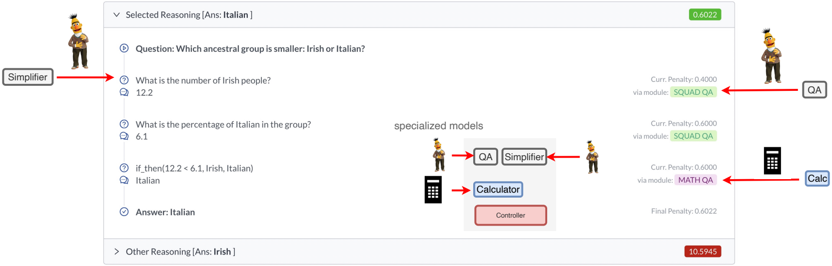



Often, designed for **inference-time** reasoning with frozen models; doesn't account for learning or model tuning, a limitation.

# Text Modular Networks (TMNs)

- Small modeling language and search framework, **inference-time reasoning** with plug-and-play (frozen) models and tools (Khot et al., 2021, 2022)

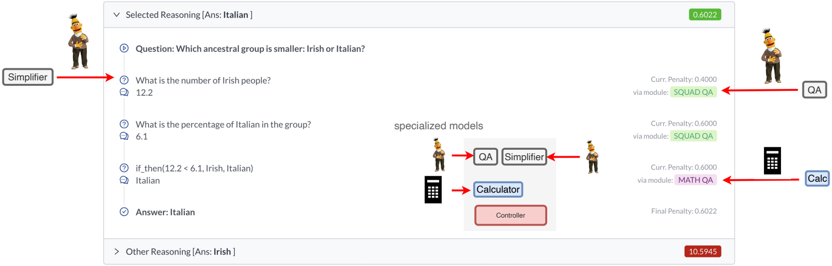
https://modularqa-demo.apps.allenai.org/



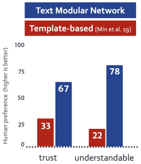
# Text Modular Networks (TMNs)

- Small modeling language and search framework, **inference-time reasoning** with plug-and-play (frozen) models and tools (Khot et al., 2021, 2022)

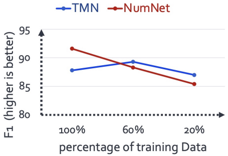
https://modularqa-demo.apps.allenai.org/




Improved explanation quality

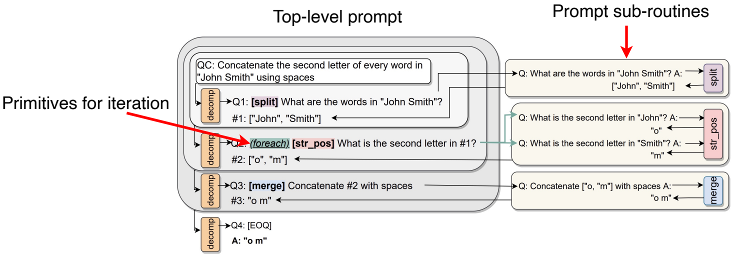



Improved training efficiency



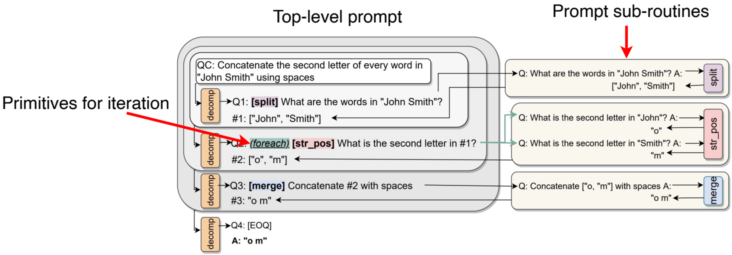
# Decomposed Prompting: Modular Prompting

- TMN-style modeling language for prompting (Khot et al., 2023), alternative to advanced prompting strategies, chain-of-thought (CoT) (Wei et al., 2022).



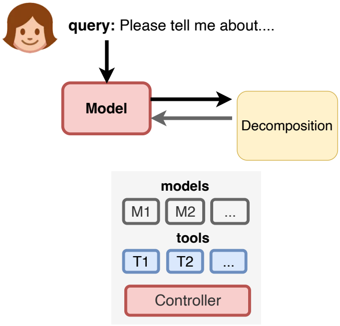
# Decomposed Prompting: Modular Prompting

- TMN-style modeling language for prompting (Khot et al., 2023), alternative to advanced prompting strategies, chain-of-thought (CoT) (Wei et al., 2022).

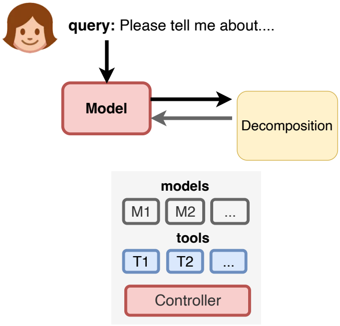


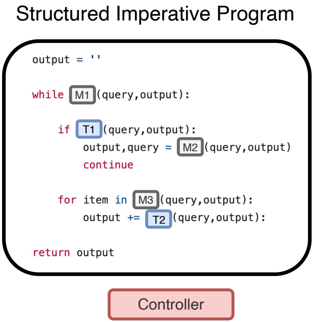

Much improved generalization over CoT, effective with smaller models.

# What's the bigger picture?

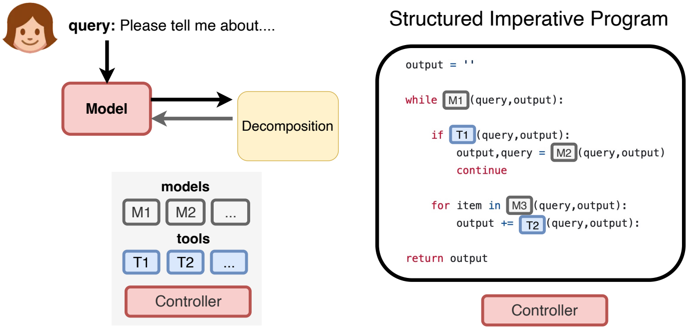


# What's the bigger picture?



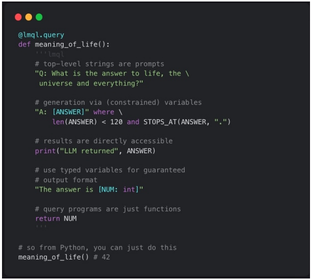



# What's the bigger picture?



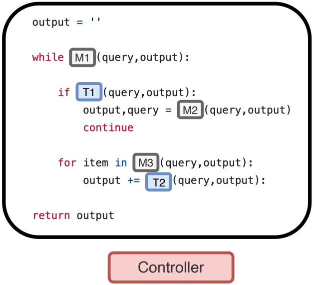

- What is the relationship between modular modeling and structured programming? Next generation of programming languages with LLMs.

# What’s the bigger picture?




LMQL

Structured Imperative Program




General-purpose (imperative) programming languages exist for LLMs and prompting (Beurer-Kellner et al., 2023), PL semantics + LLM algorithms.

# Different kinds of model programming

- **Model Programming:** The problem of how modeling components are assembled, how they interact, and the language of how this is specified.

# Different kinds of model programming

- **Model Programming:** The problem of how modeling components are assembled, how they interact, and the language of how this is specified.

Imperative Style

```
output = ''

while M1(query,output):

    if T1(query,output):
        output,query = M2(query,output)
        continue

    for item in M3(query,output):
        output += T2(query,output):

return output
```

Declarative Style

```
Implies(
    And(M1(query,output),T2(query,output)),
    M2(query,output)
)
Biconditional(
    M1(query,output),
    Not(M3(query,output))
)
Or(T1(query,output),Not(M2(query,output)))
```

# Different kinds of model programming

- **Model Programming:** The problem of how modeling components are assembled, how they interact, and the language of how this is specified.

Imperative Style

```
output = ''

while M1(query,output):

    if T1(query,output):
        output,query = M2(query,output)
        continue

    for item in M3(query,output):
        output += T2(query,output):

return output
```

Declarative Style

```
Implies(
    And(M1(query,output),T2(query,output)),
    M2(query,output)
)
Biconditional(
    M1(query,output),
    Not(M3(query,output))
)
Or(T1(query,output),Not(M2(query,output)))
```

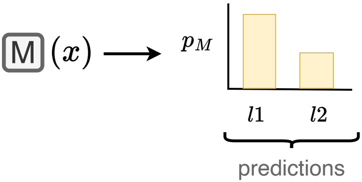
| Paradigm | Basic Units | Applications |
|---|---|---|
| **imperative** | models as **subroutines**, for/while loops, recursion, if-then control | structure building, combining LLMs w/ tools, explanation. |
| **declarative** | models predictions as **symbolic objects**, logical constraints | probabilistic reasoning, learning, self-correction, consistency. |

# Declarative Model Programming

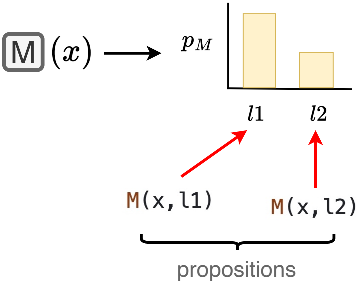
# Declarative Modeling: Predictions as Propositions

$$\underbrace{\boxed{\mathsf{M}}\,(x)}_{\text{inference}} \longrightarrow$$

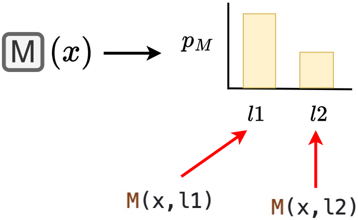
# Declarative Modeling: Predictions as Propositions



# Declarative Modeling: Predictions as Propositions



# Declarative Modeling: Predictions as Propositions




$$\underbrace{\theta(\ \texttt{M(x,l)}\ )}_{\text{proposition weight}} = p_M(l \mid x)$$

# Declarative Modeling: Predictions as Propositions

`M(x1,l1)` `M(x2,l2)`

$$\boxed{D = \{(x_1, l_1), (x_2, l_2), \ldots\}}$$

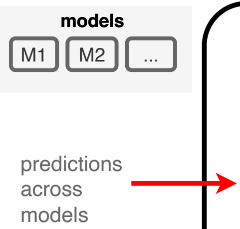
# Declarative Modeling: Programs as constraints on predictions

Prediction symmetries

negative information

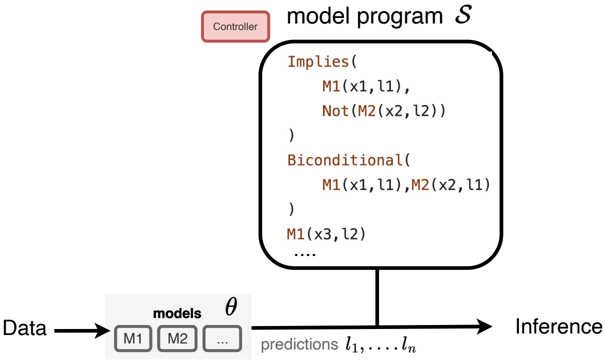
```
Implies(
    M(x1,l1),
    Not(M(x2,l2))
)
Biconditional(
    M(x1,l1),M(x2,l1)
)
Not(M(x3,l2))
```

# Declarative Modeling: Programs as constraints on predictions

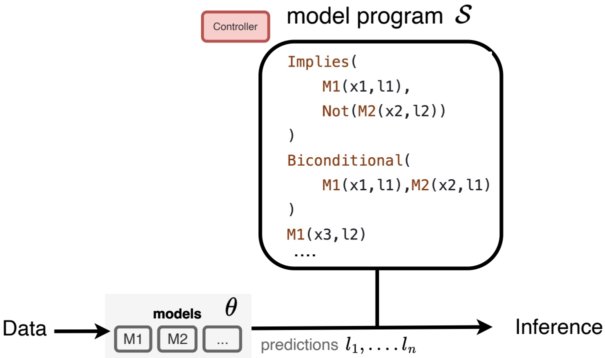



```
Implies(
    M1(x1,l1),
    Not(M2(x2,l2))
)
Biconditional(
    M1(x1,l1),M2(x2,l1)
)
M1(x3,l2)
```

# Declarative Modeling: Inference and Learning

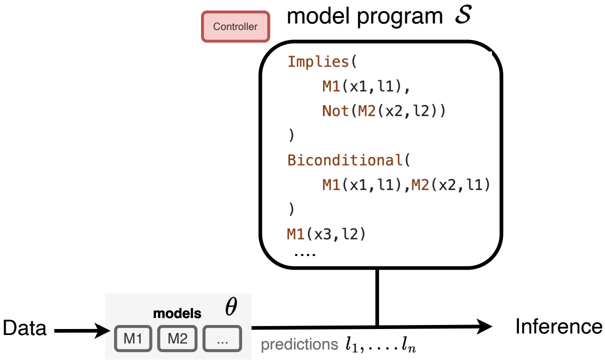


# Declarative Modeling: Inference and Learning




$$\mathrm{S}_\theta(\mathcal{S}) = \underbrace{\sum_{I \models \mathcal{S}} \prod_{l \in I} \theta(l)}_{\text{weighted model counting}}$$

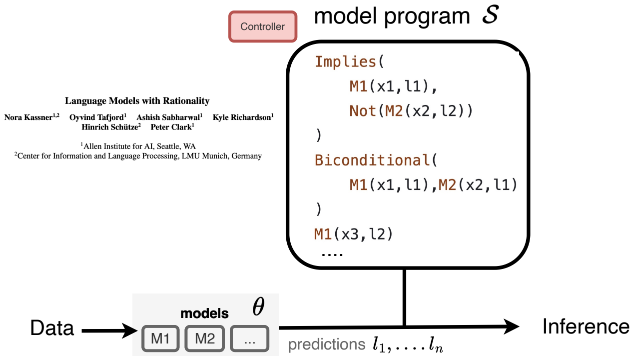
# Declarative Modeling: Inference and Learning




| inference | $\bigoplus$ | $\bigotimes$ |
|---|---|---|
| PROB | $\sum$ | $\prod$ |
| MPE | max | $\prod$ |
| FUZZY | max | min |

$$\mathrm{S}_\theta(\mathcal{S}) = \bigoplus_{I \models \mathcal{S}} \bigotimes_{l \in I} \theta(l)$$

# Declarative Modeling: Inference and Learning

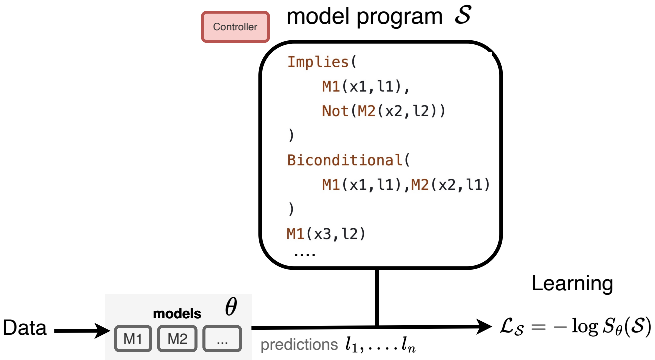



| inference | $\bigoplus$ | $\bigotimes$ |
|---|---|---|
| PROB | $\sum$ | $\prod$ |
| MPE | max | $\prod$ |
| FUZZY | max | min |

$$\mathrm{S}_\theta(\mathcal{S}) = \bigoplus_{I \models \mathcal{S}} \bigotimes_{l \in I} \theta(l)$$

Used probabilistic reasoning for correcting LLM *beliefs* and inconsistencies **[EMNLP23]**, efficient in practice via SAT technology.

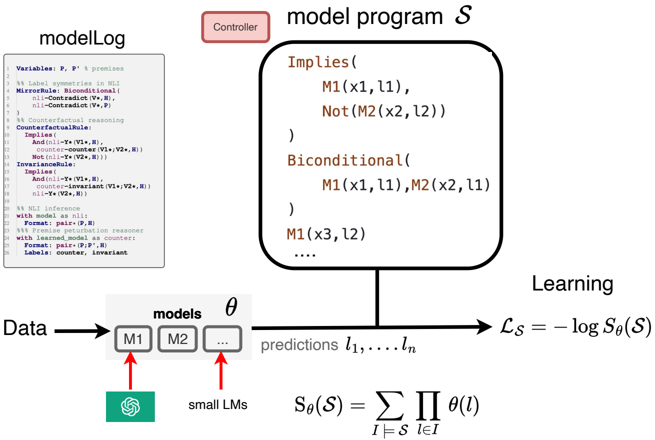
# Declarative Modeling: Inference and Learning




$$\mathrm{S}_\theta(\mathcal{S}) = \sum_{I \models \mathcal{S}} \prod_{l \in I} \theta(l)$$

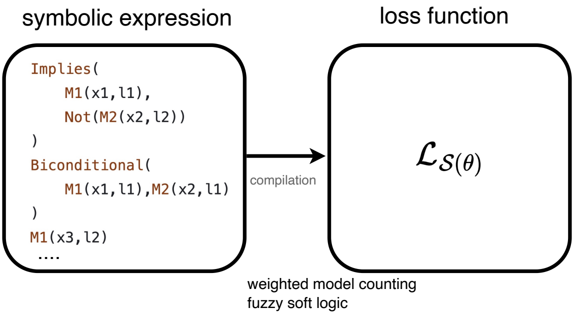
Known and efficient techniques for compiling symbolic expressions into loss, model-counting based (Xu et al., 2018) and fuzzy logic (Li et al., 2019).

# Declarative Modeling: Inference and Learning

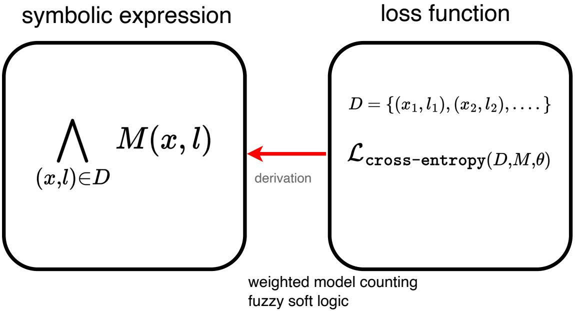



*(Forthcoming)* A declarative programming language (modelLog), used for learning systems on top of frozen LLMs, calibration, consistency training.

# Understanding the semantics of learning

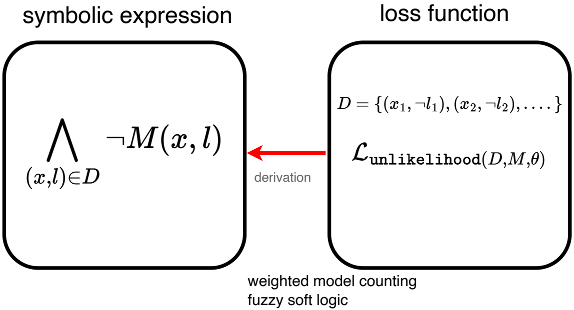


# Understanding the semantics of learning



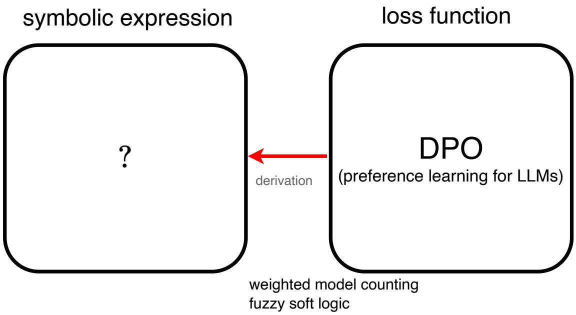

- Finding symbolic expressions that compile into known loss functions, better understand these losses, derive new ones from first principles.

# Understanding the semantics of learning




- Finding symbolic expressions that compile into known loss functions, better understand these losses, derive new ones from first principles.

# Understanding the semantics of learning




- Finding symbolic expressions that compile into known loss functions, better understand these losses, derive new ones from first principles.

# Declarative and neuro-symbolic modeling

Formal foundations of neuro-symbolic modeling

Kyle Richardson and Vivek Srikumar

March 11, 2024

## 1 Introduction

When doing neural-symbolic modeling of any kind, one must first fix on a particular framework for symbolic inference. In this survey, we will focus on approaches based on formal logic and, in particular, on fragments of classical propositional logic. The goal of this chapter, therefore, is to introduce the basic propositional calculus. In particular, our review aims to be *algorithmic* in nature by focusing on the technical tools needed for *building* the kinds of automated reasoning tools that underlie current neural-symbolic systems based on logic. Given the inherent computational difficulty of working with logic, and the need for ultimately unifying logic with machine learning systems, a central focus will be on *tractable representations of logic* that are compatible with the kinds of *gradient-based learning* that we describe in the next chapter.

- Many technical topics related to efficient inference, systems for compiling logic to loss; ESSLLI course 2023-2024.

# Conclusion

- Much of what we do in NLP can be viewed as a kind of programming, **model programming**. Useful both conceptually and technically:

# Conclusion

- Much of what we do in NLP can be viewed as a kind of programming, **model programming**. Useful both conceptually and technically:

  Imperative style: modeling with tools, building more transparent models, advanced prompting.

  Declarative style: modeling with constraints, integrating learning and probabilistic reasoning, correcting model errors.

Thank You

# References I


Beurer-Kellner, L., Fischer, M., and Vechev, M. (2023). Prompting is programming: A query language for large language models. *Proceedings of the ACM on Programming Languages*, 7(PLDI):1946–1969.

Groeneveld, D., Beltagy, I., Walsh, P., Bhagia, A., Kinney, R., Tafjord, O., Jha, A. H., Ivison, H., Magnusson, I., Wang, Y., et al. (2024). Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Khot, T., Khashabi, D., Richardson, K., Clark, P., and Sabharwal, A. (2021). Text modular networks: Learning to decompose tasks in the language of existing models. *Proceedings of NAACL*.

Khot, T., Richardson, K., Khashabi, D., and Sabharwal, A. (2022). Learning to solve complex tasks by talking to agents. *Findings of ACL*.

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. (2023). Decomposed prompting: A modular approach for solving complex tasks. *ICLR*.

Li, T., Gupta, V., Mehta, M., and Srikumar, V. (2019). A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*.

Magnusson, I., Bhagia, A., Hofmann, V., Soldaini, L., Harsh Jha, A., Tafjord, O., Schwenk, D., Walsh, E. P., Elazar, Y., Lo, K., et al. (2023). Paloma: A benchmark for evaluating language model fit. *arXiv e-prints*, pages arXiv–2312.

# References II


Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., et al. (2024). Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv e-prints*, pages arXiv–2402.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Broeck, G. (2018). A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pages 5502–5511. PMLR.