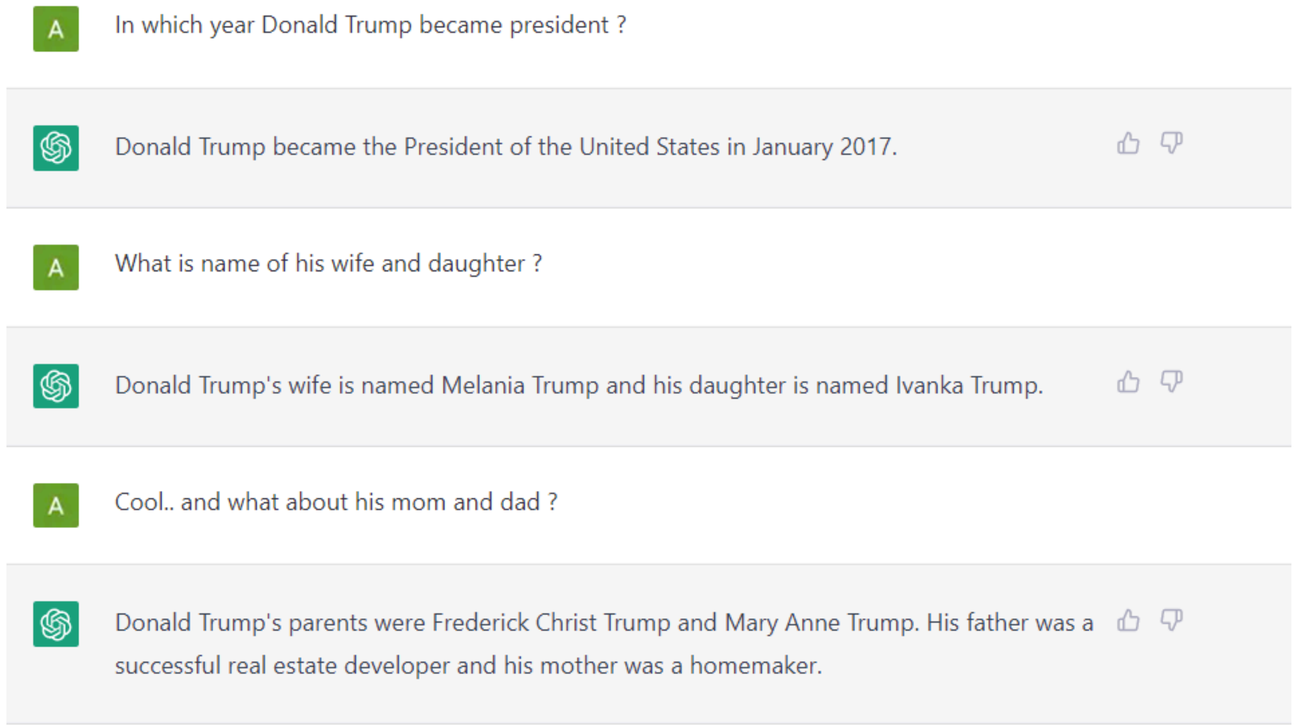
Optimizing Prompts for Large Language Models Using Reinforcement Learning

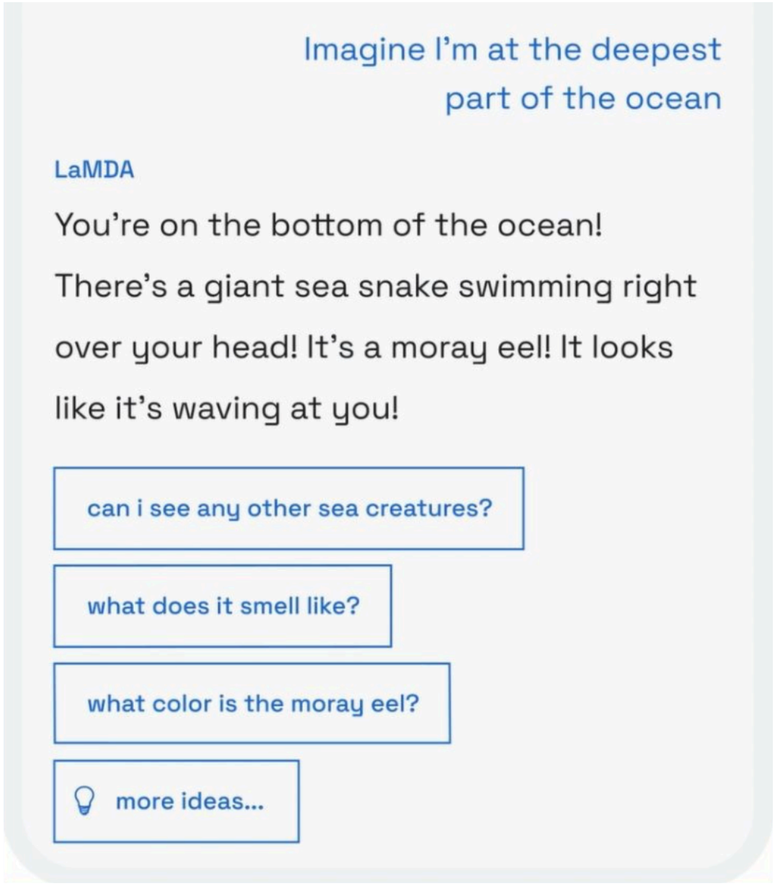
Mingkai Deng

Practical LLM Workshop

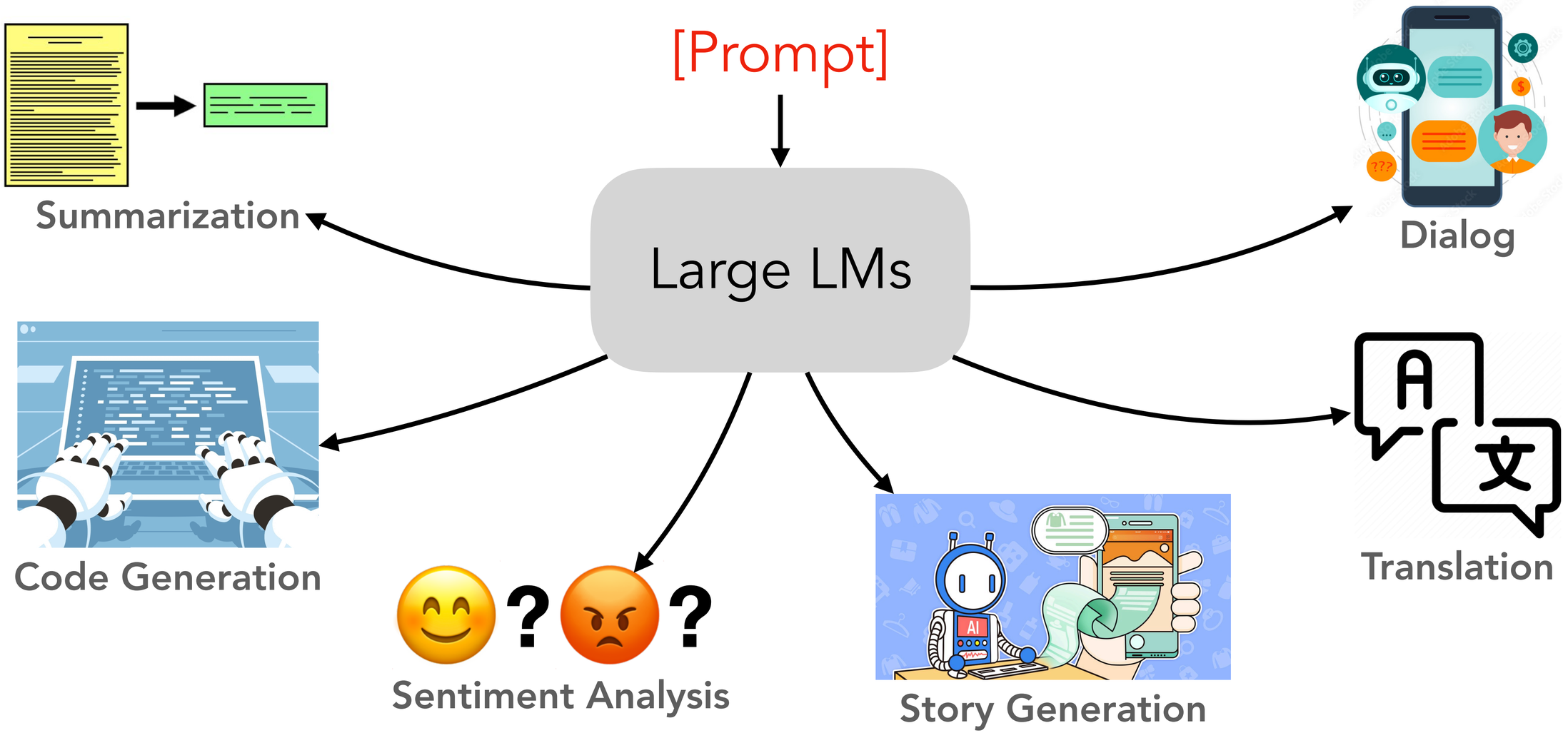
March 10, 2023

Large language models (LMs) show impressive success performing various tasks

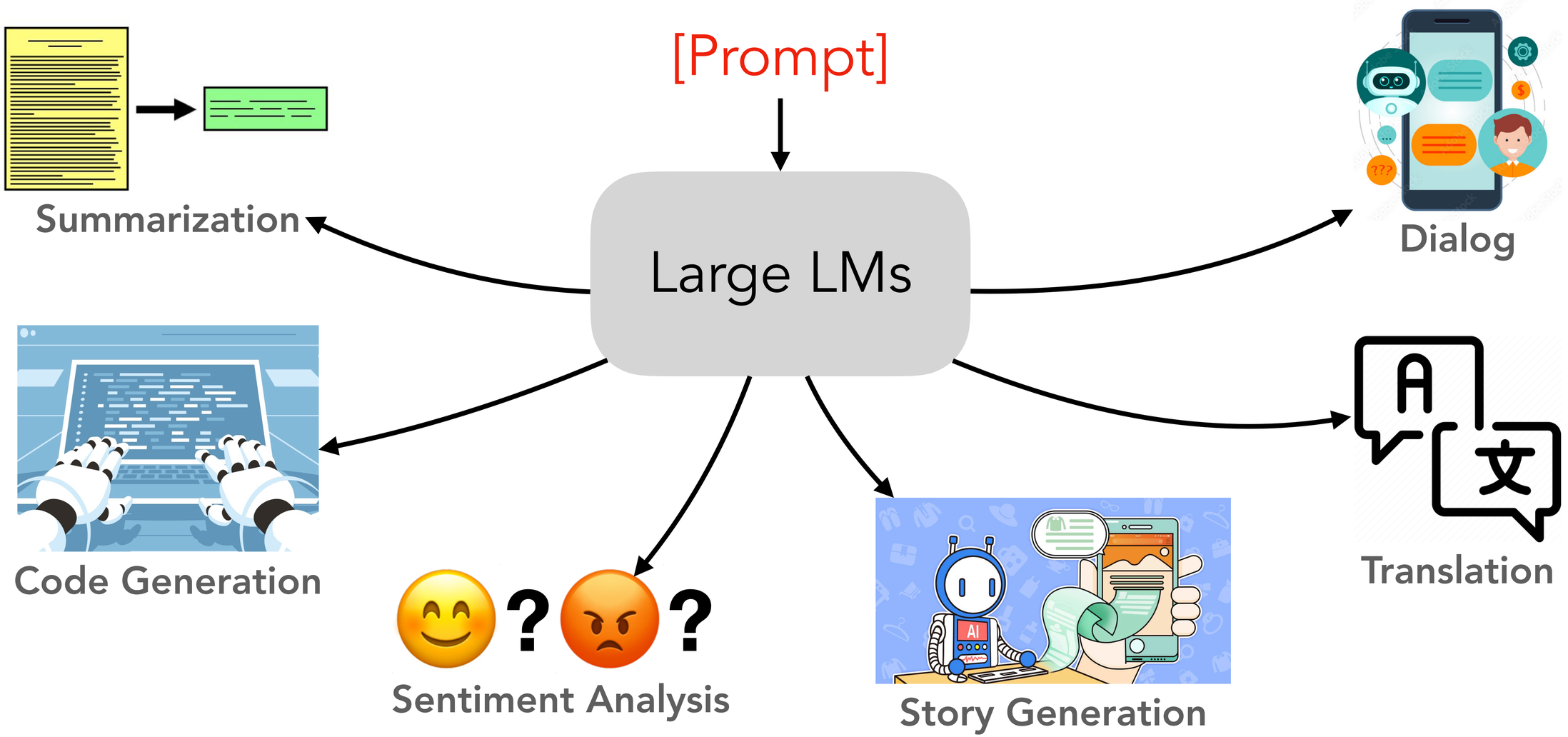




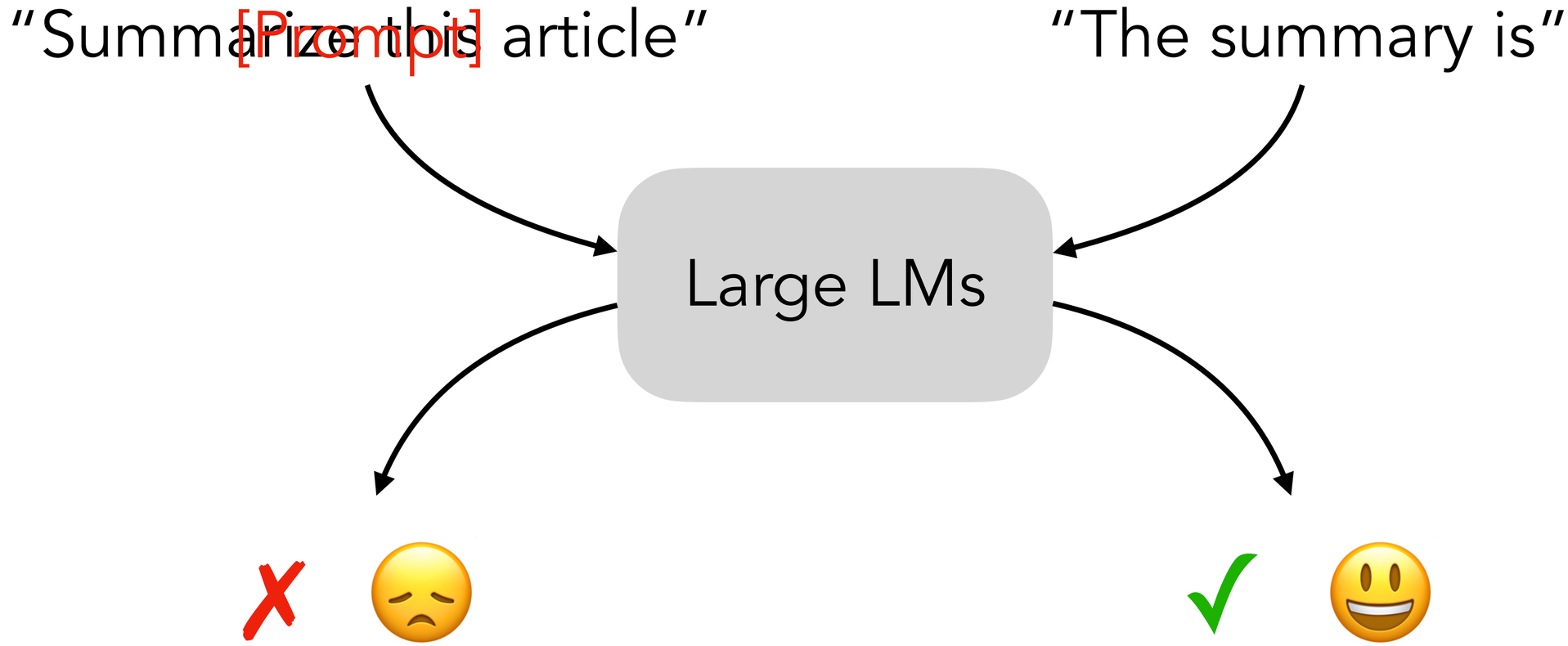
Large language models (LMs) show impressive success performing various tasks

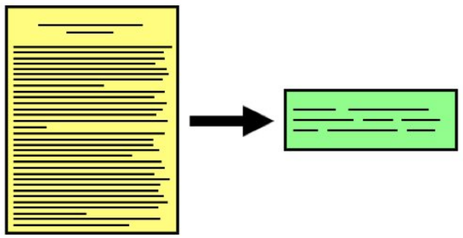


LLMs can be sensitive to prompt wordings



LLMs can be sensitive to prompt wordings



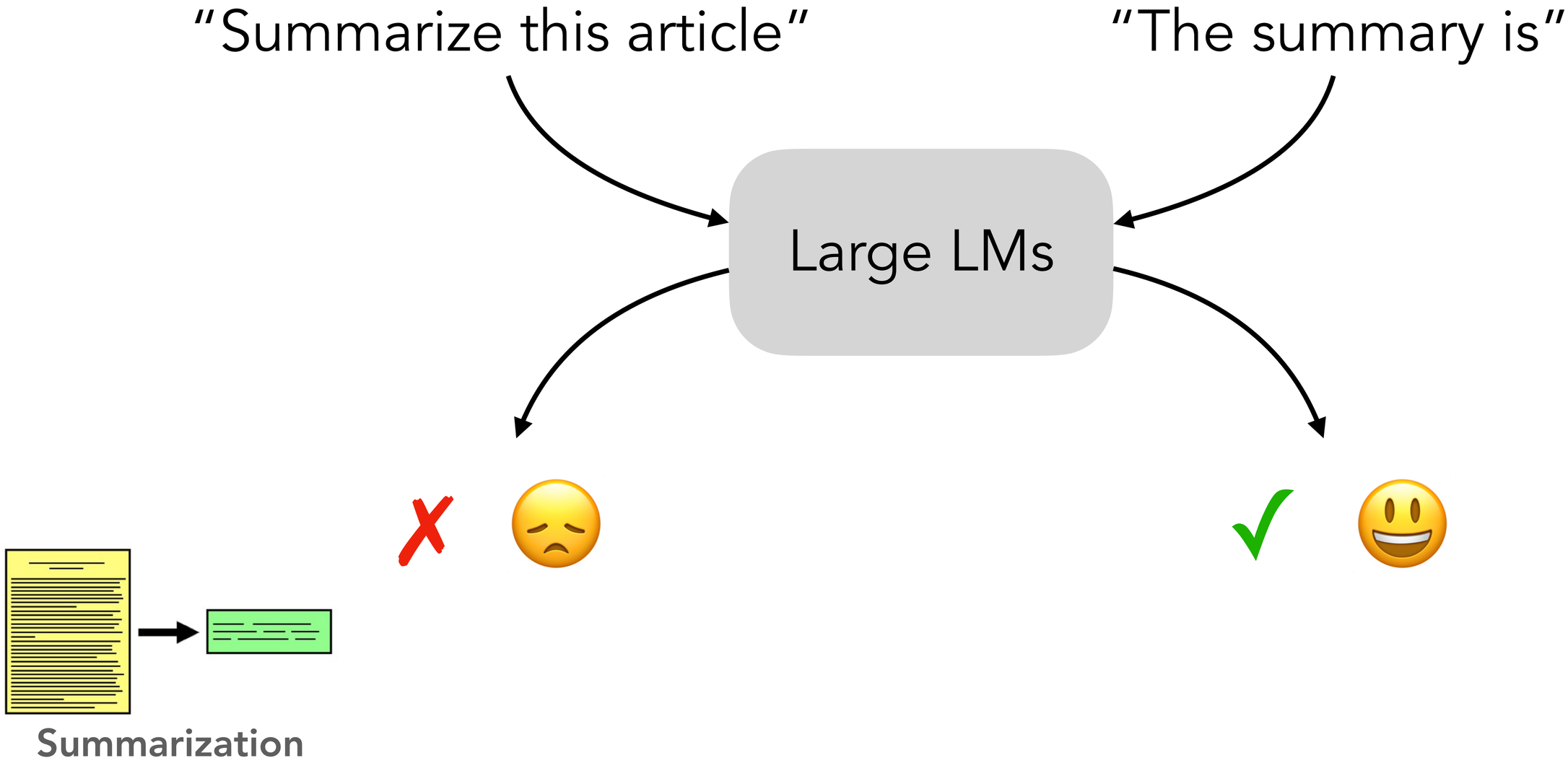


Summarization

What is the best way to talk to them?

How can we make LLMs do what we want?

Finding the Best Prompt Can Be Challenging

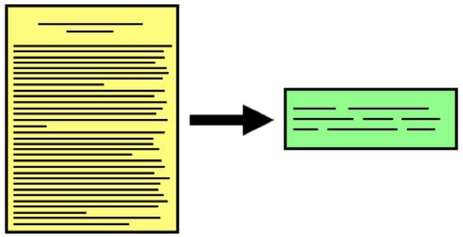


Finding the Best Prompt Can Be Challenging

- Manual Design

"Summarize this article"

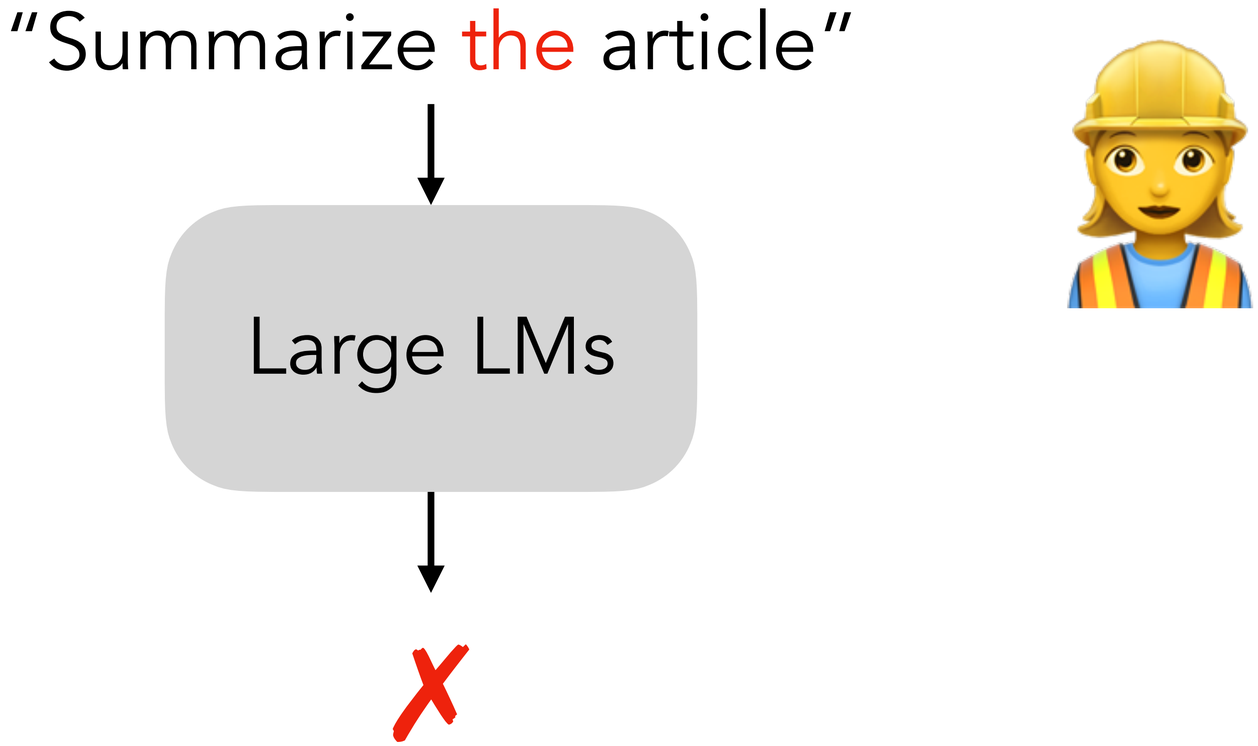
Large LMs

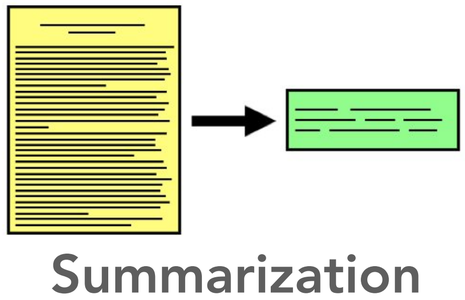


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design

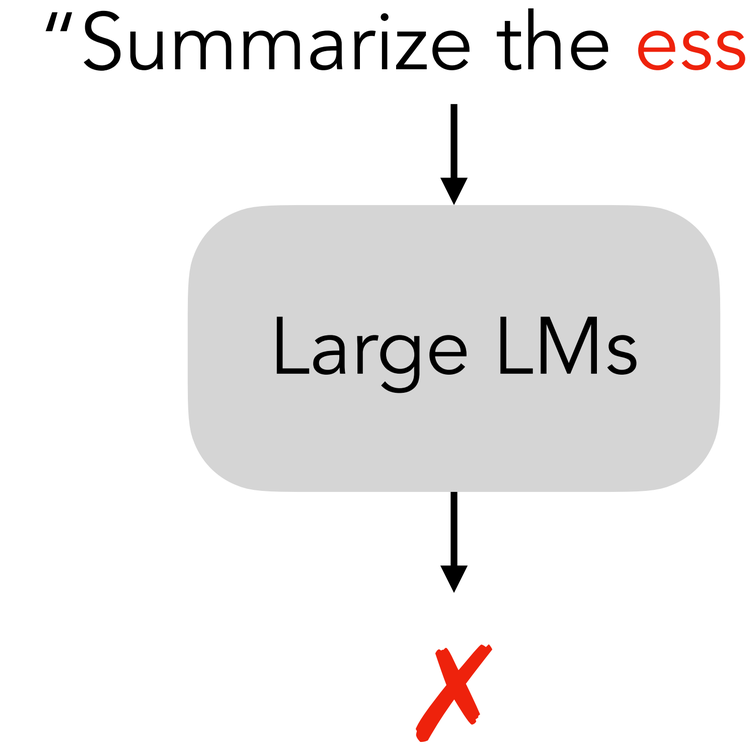


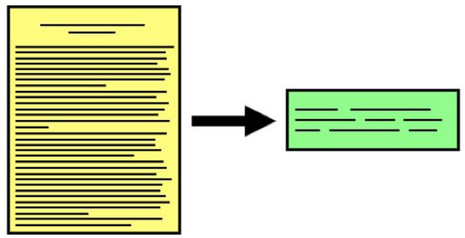


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design

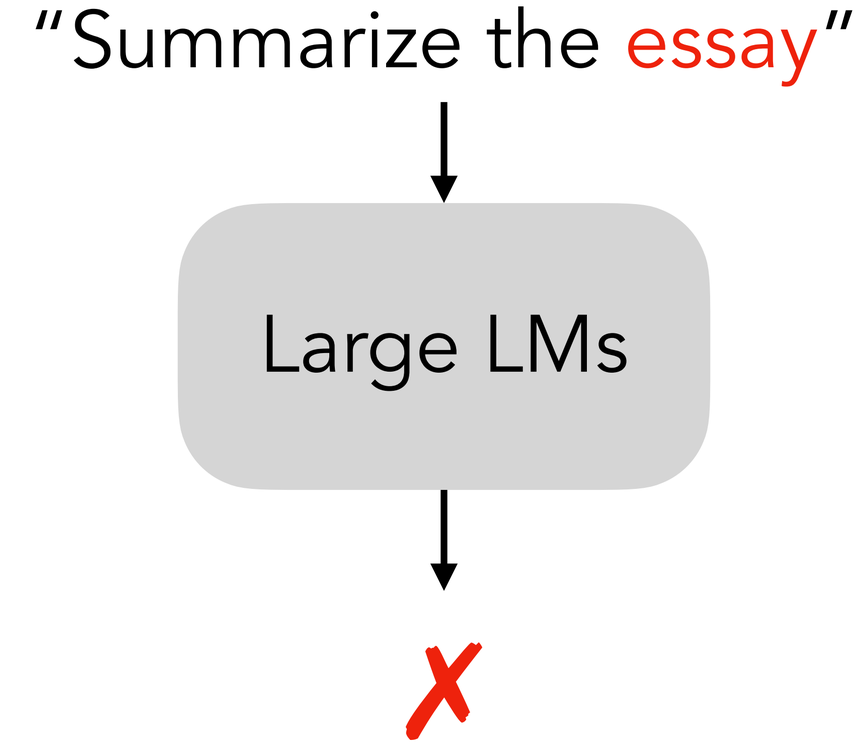


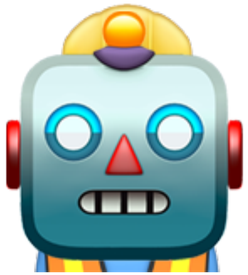


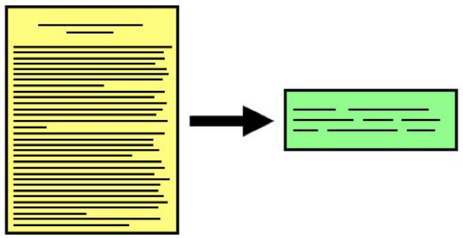
Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select



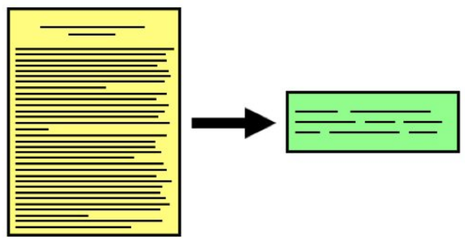




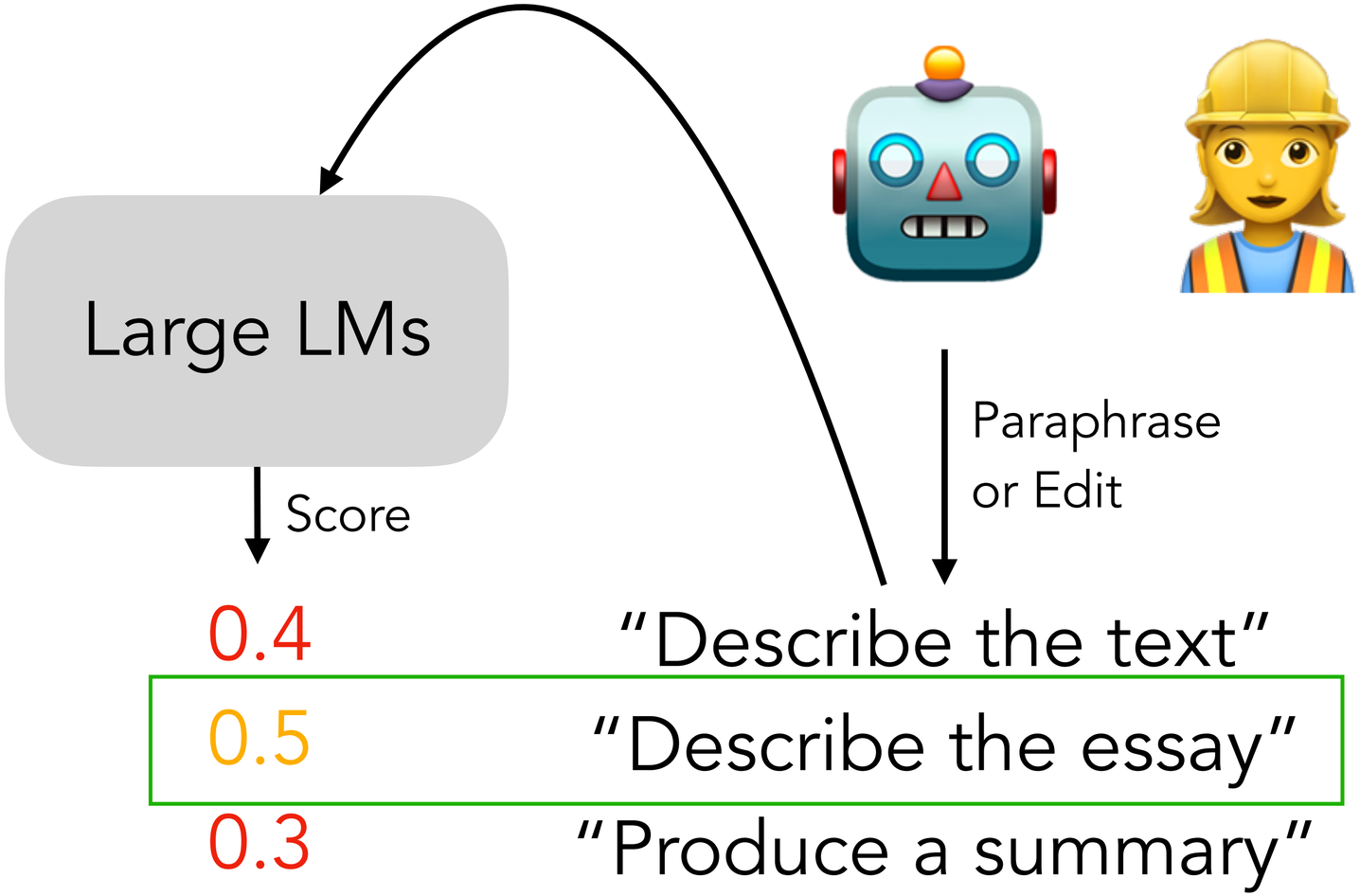
Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select

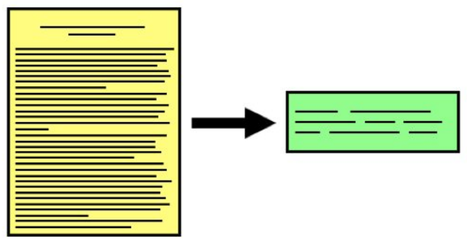


Summarization

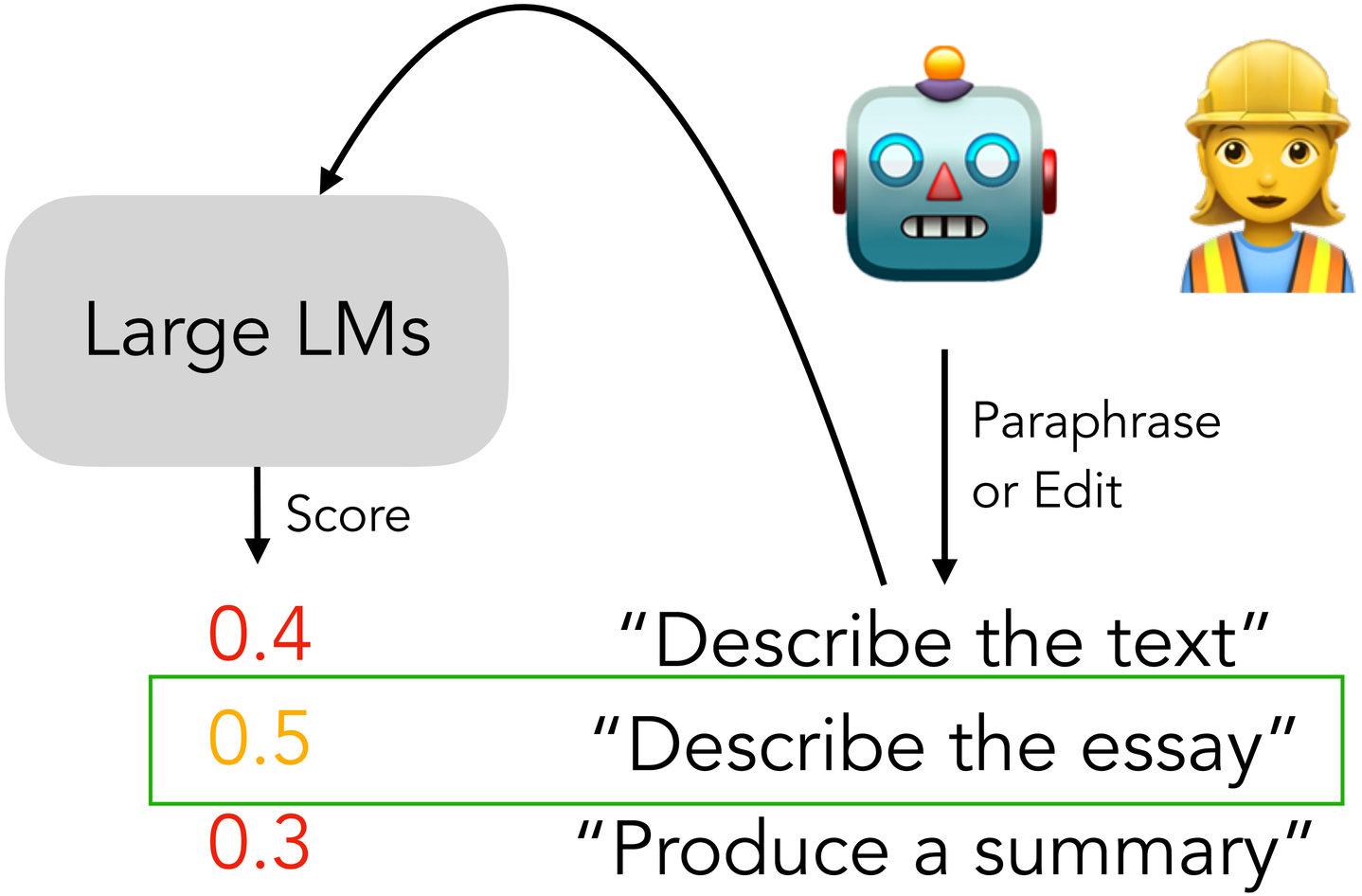


Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning



Summarization



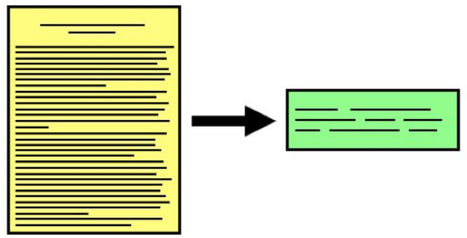
Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning

0.5	0.3	-0.1
0.2	-0.3	0.6
-0.6	0.4	0.8
0.1	-0.2	0.3

Large LMs

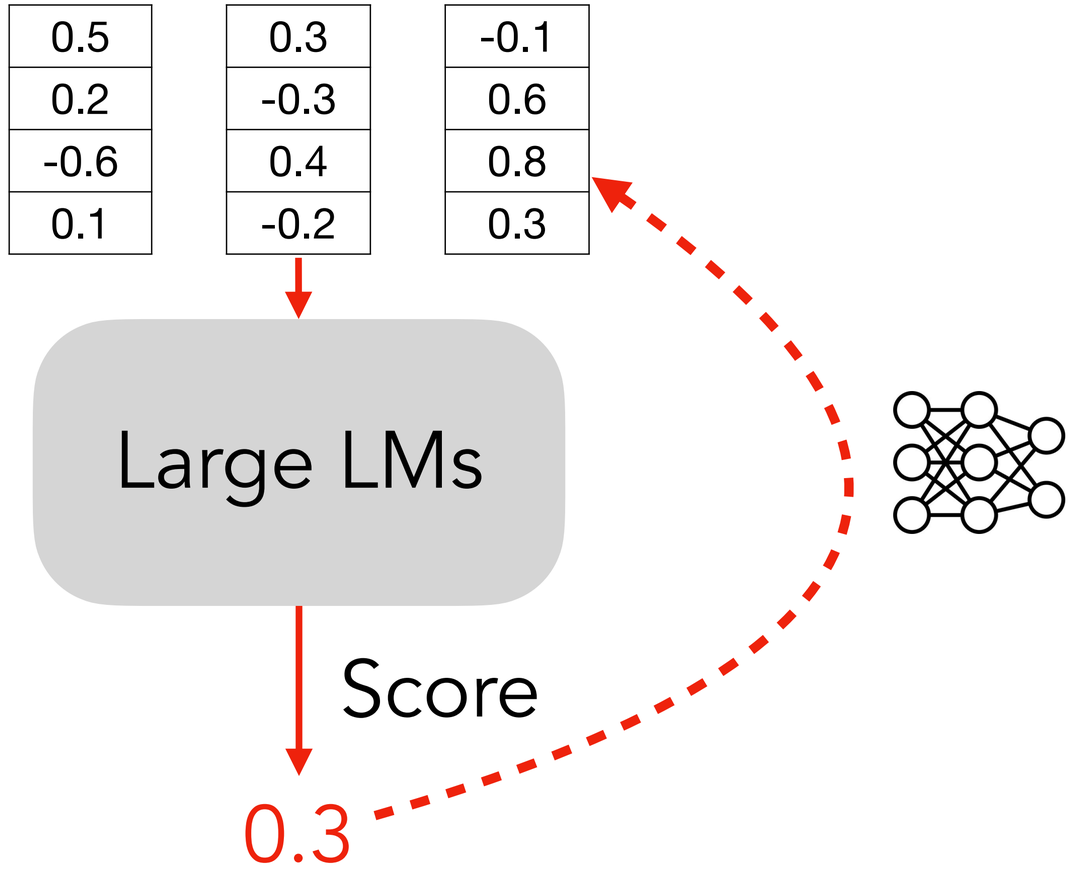
“Summarize this article”

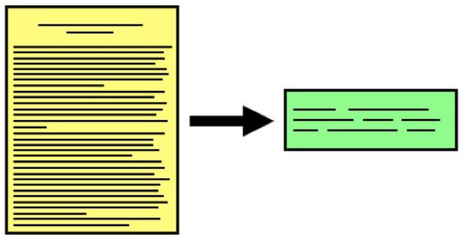


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning

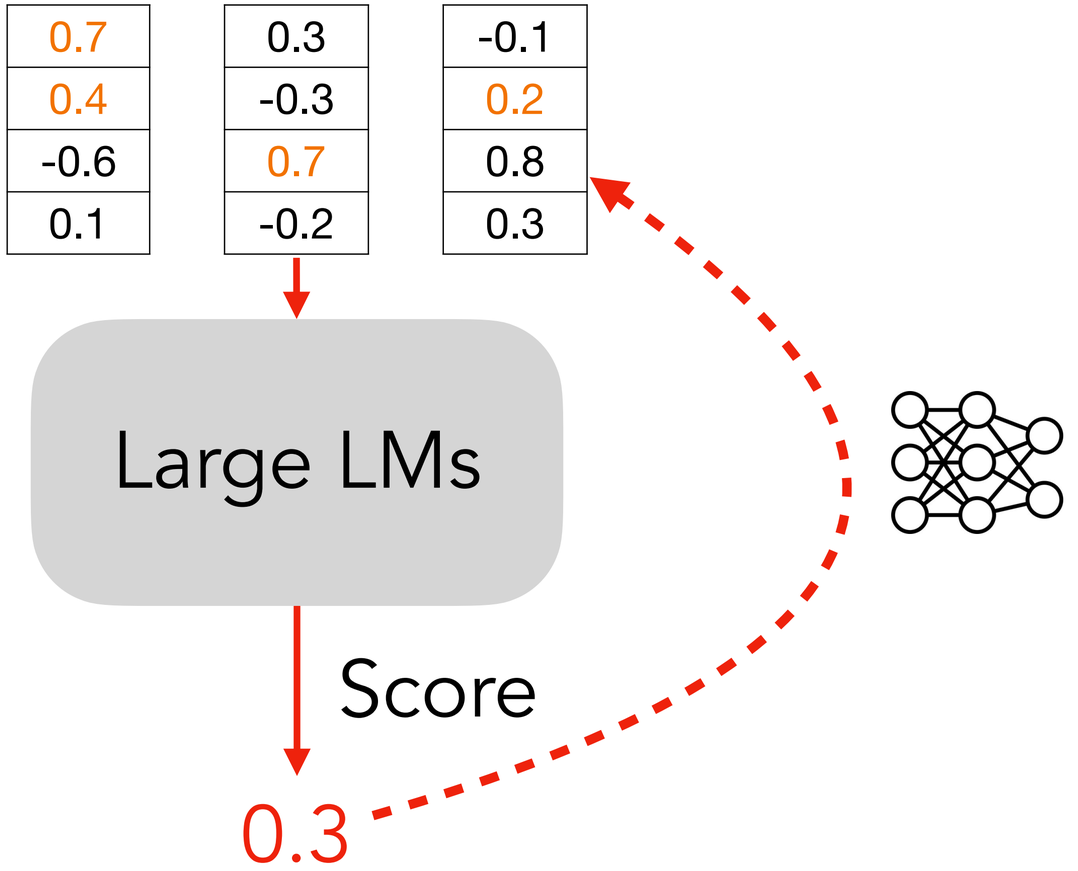


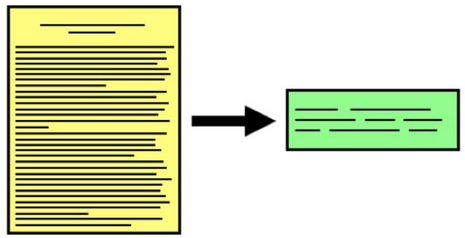


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning

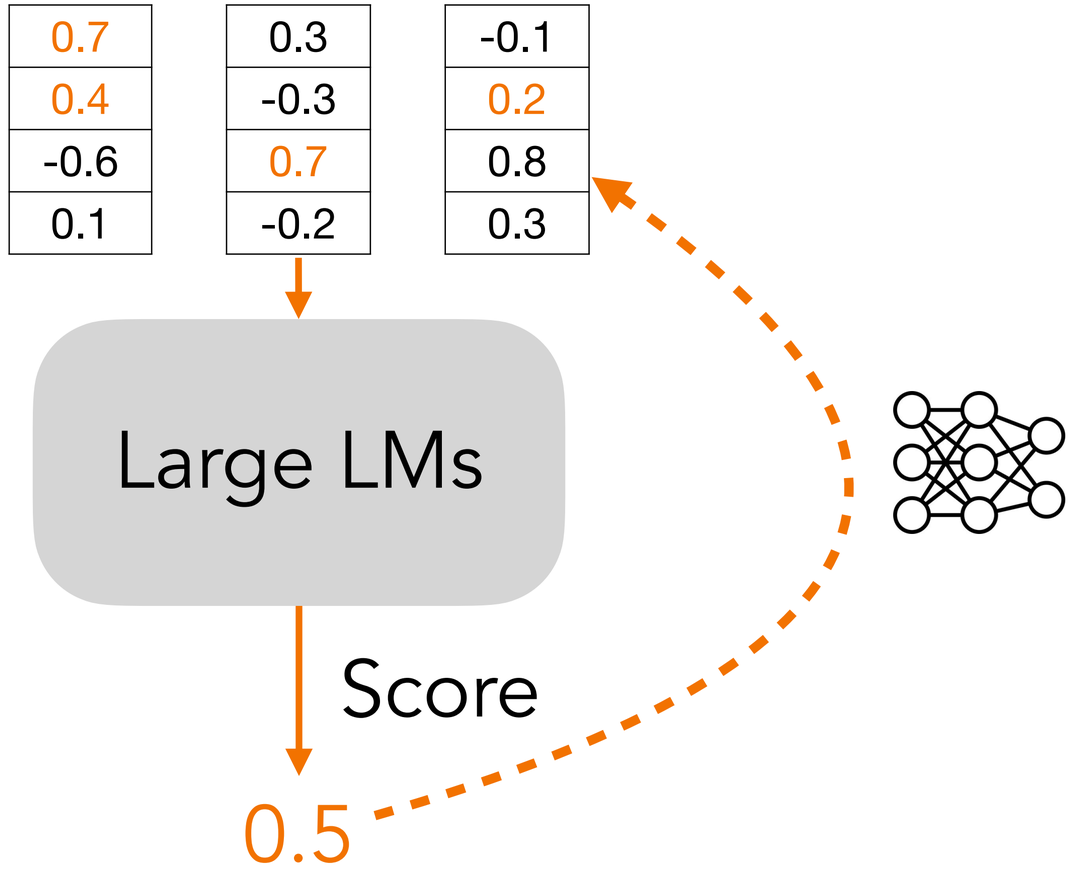


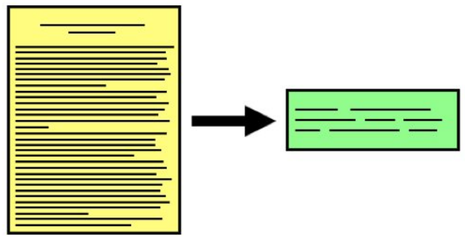


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning

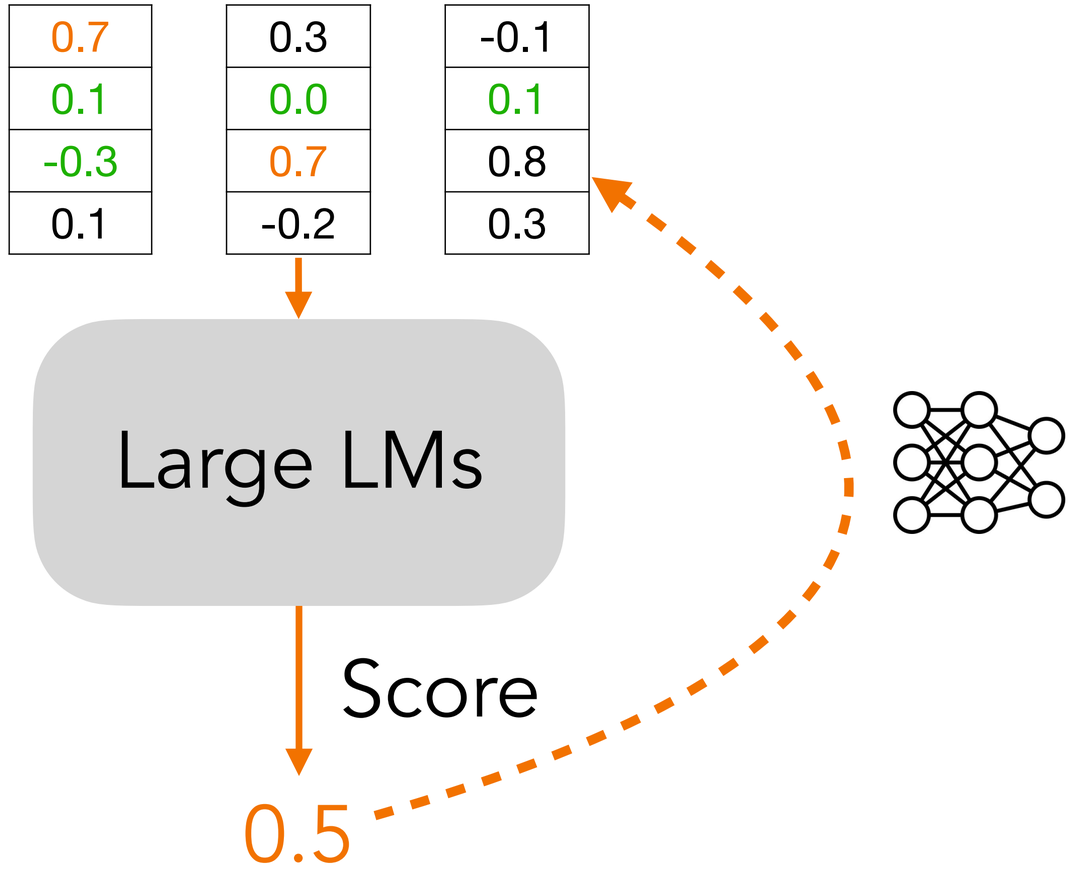


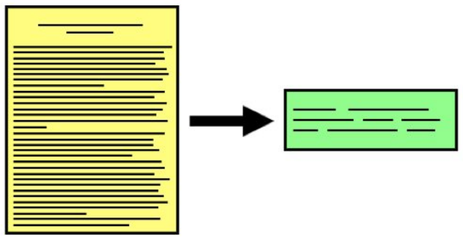


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning

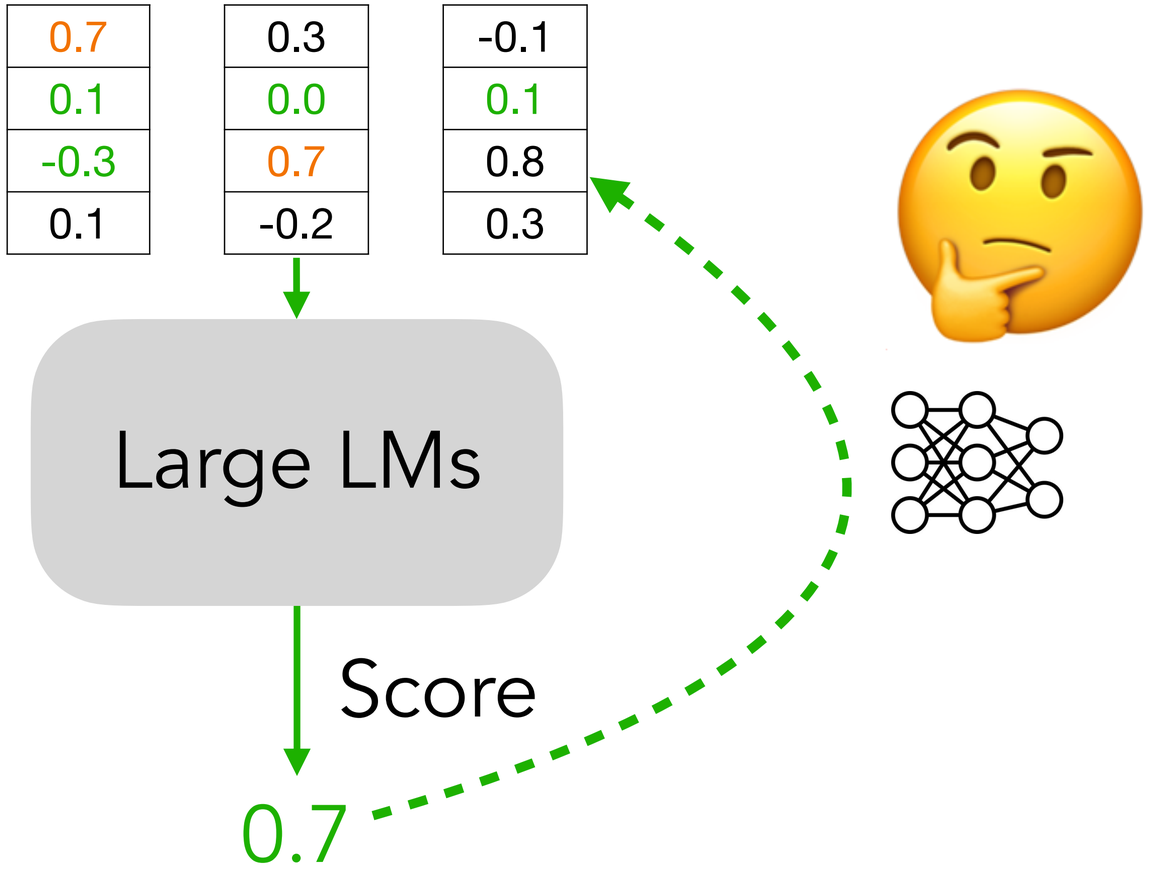


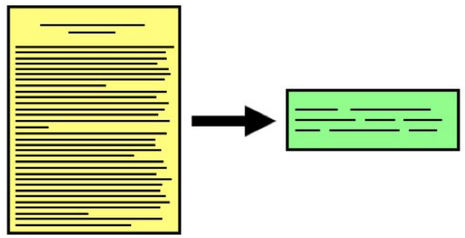


Summarization

Finding the Best Prompt Can Be Challenging

- Manual Design
- Enumerate and Select
- Numerical Tuning





Summarization

Our Solution

EMNLP 2022

RLPROMPT: Optimizing Discrete Text Prompts with Reinforcement Learning

Mingkai Deng[1]*, **Jianyu Wang**[2*], **Cheng-Ping Hsieh**[2*], **Yihan Wang**[2], **Han Guo**[1], **Tianmin Shu**[3], **Meng Song**[2], **Eric P. Xing**[1,4,5], **Zhiting Hu**[2]

[1]Carnegie Mellon University, [2]UC San Diego,
[3]MIT, [4]Mohamed bin Zayed University of Artificial Intelligence, [5]Petuum Inc.

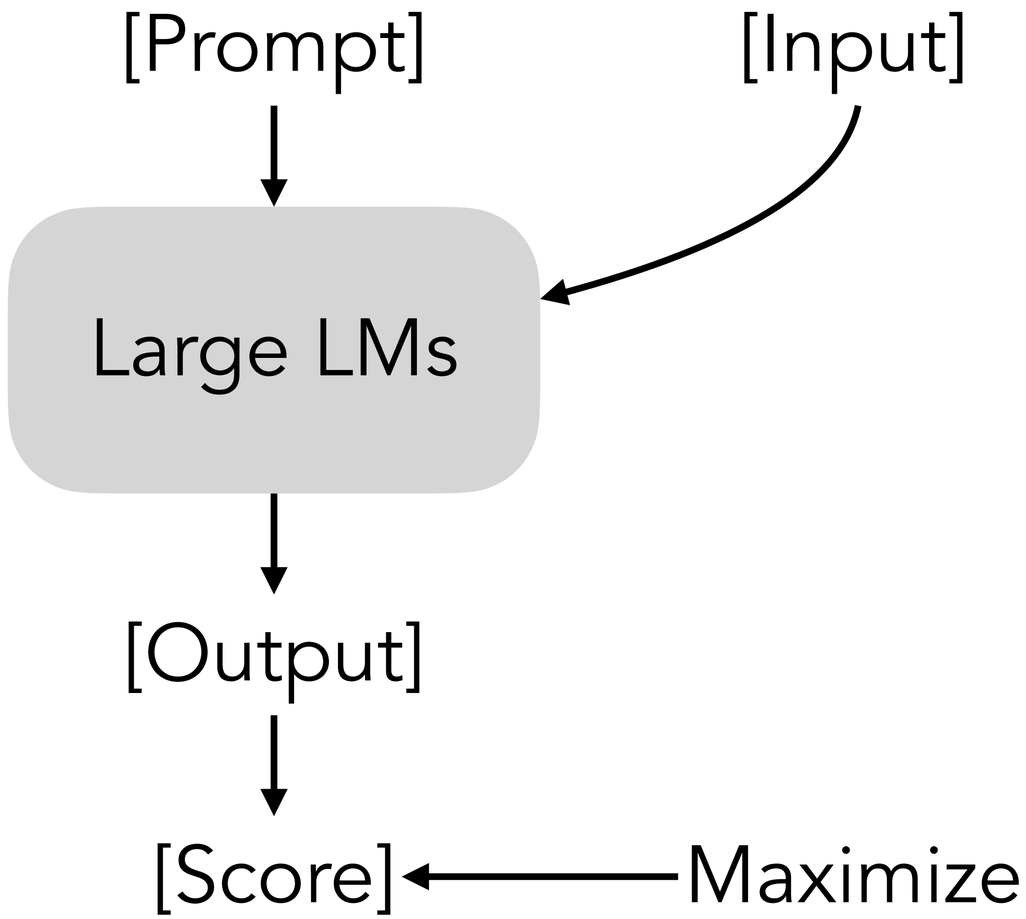
{mingkaid,hanguo}@cs.cmu.edu, {jiw102,c2hsieh,yiw007,zhh019}@ucsd.edu

Automatic ✓ Flexible ✓

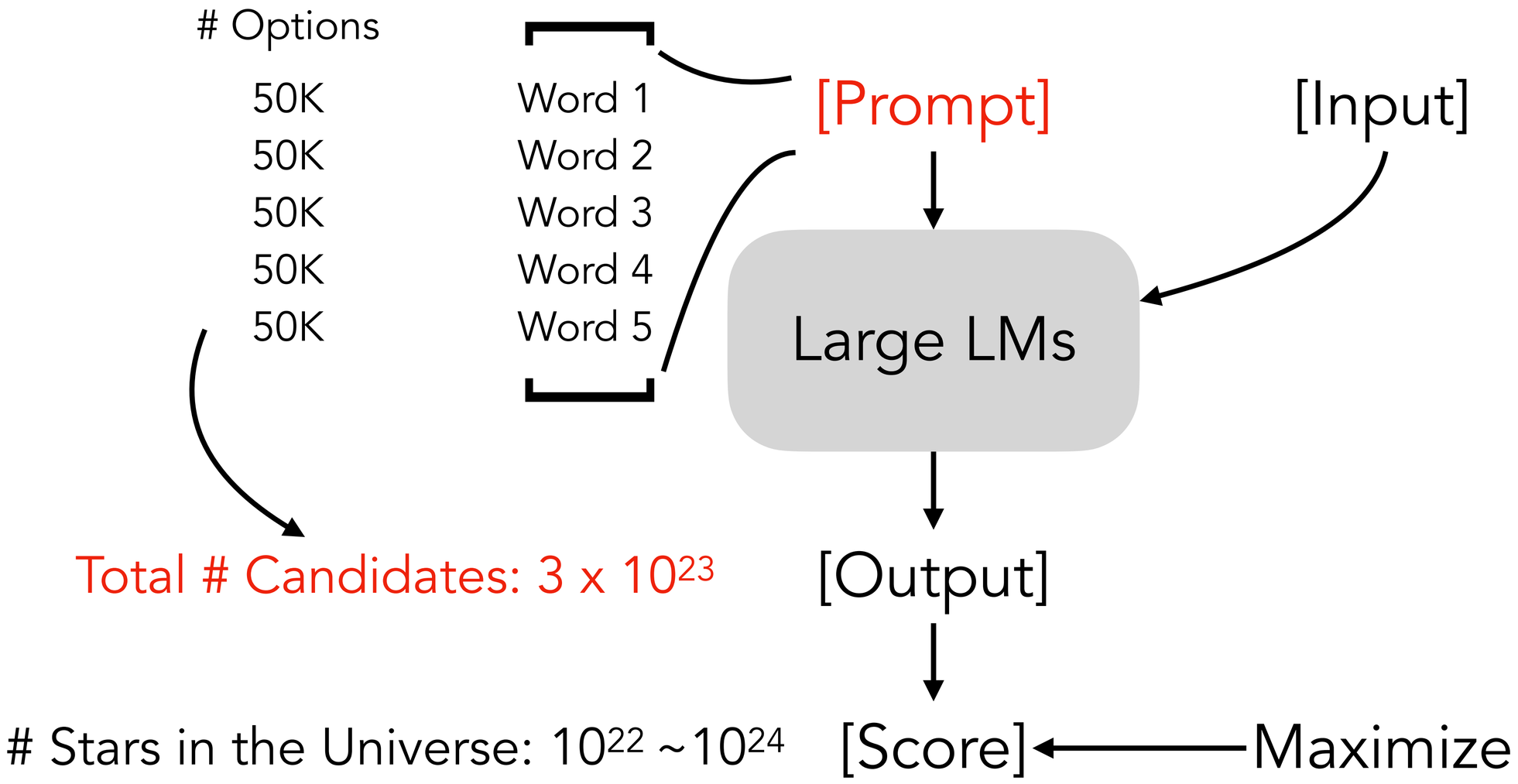
Systematic ✓ Transferrable ✓

Accessible ✓ Transparent ✓

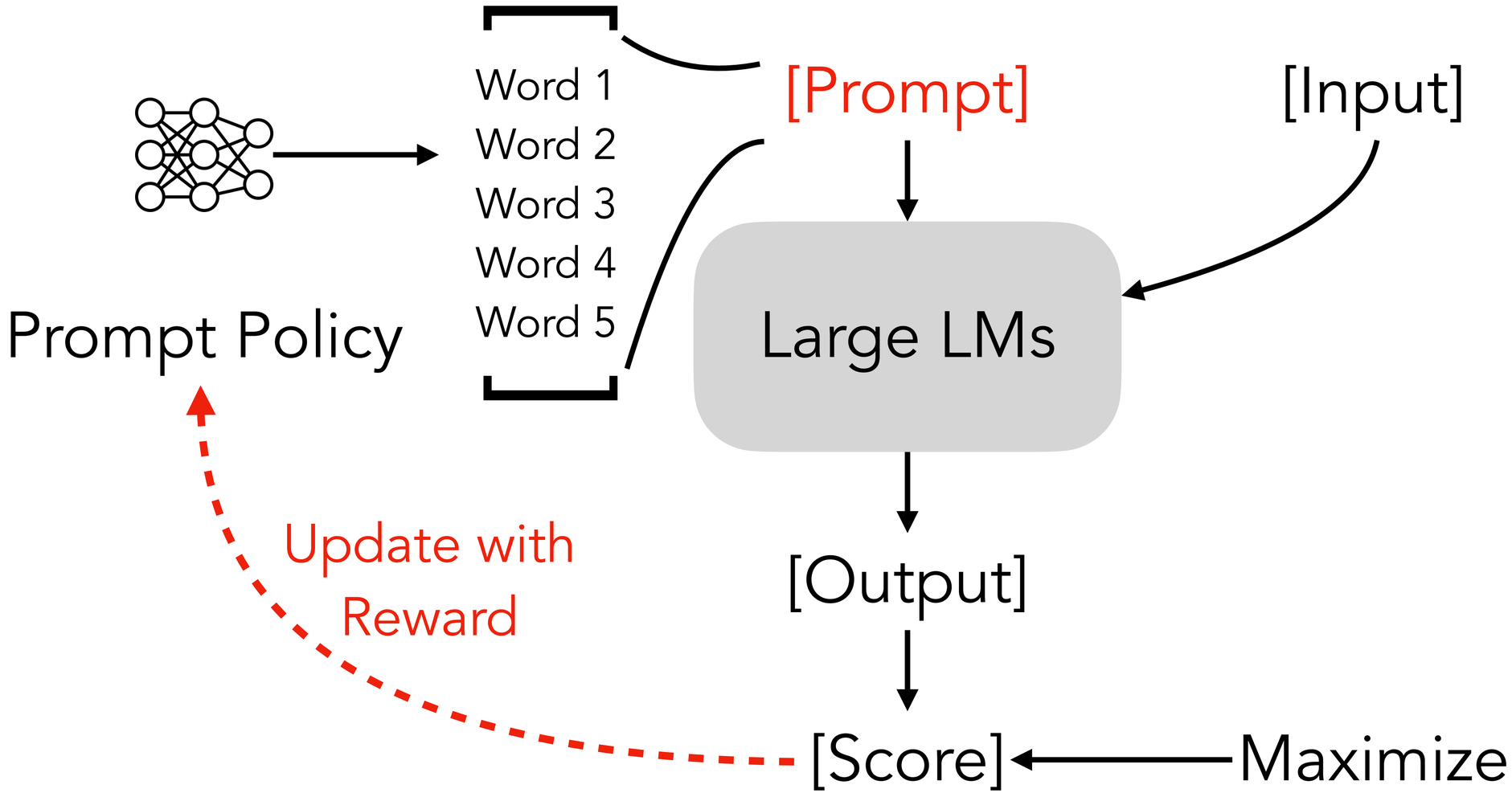
The Prompt Optimization Problem



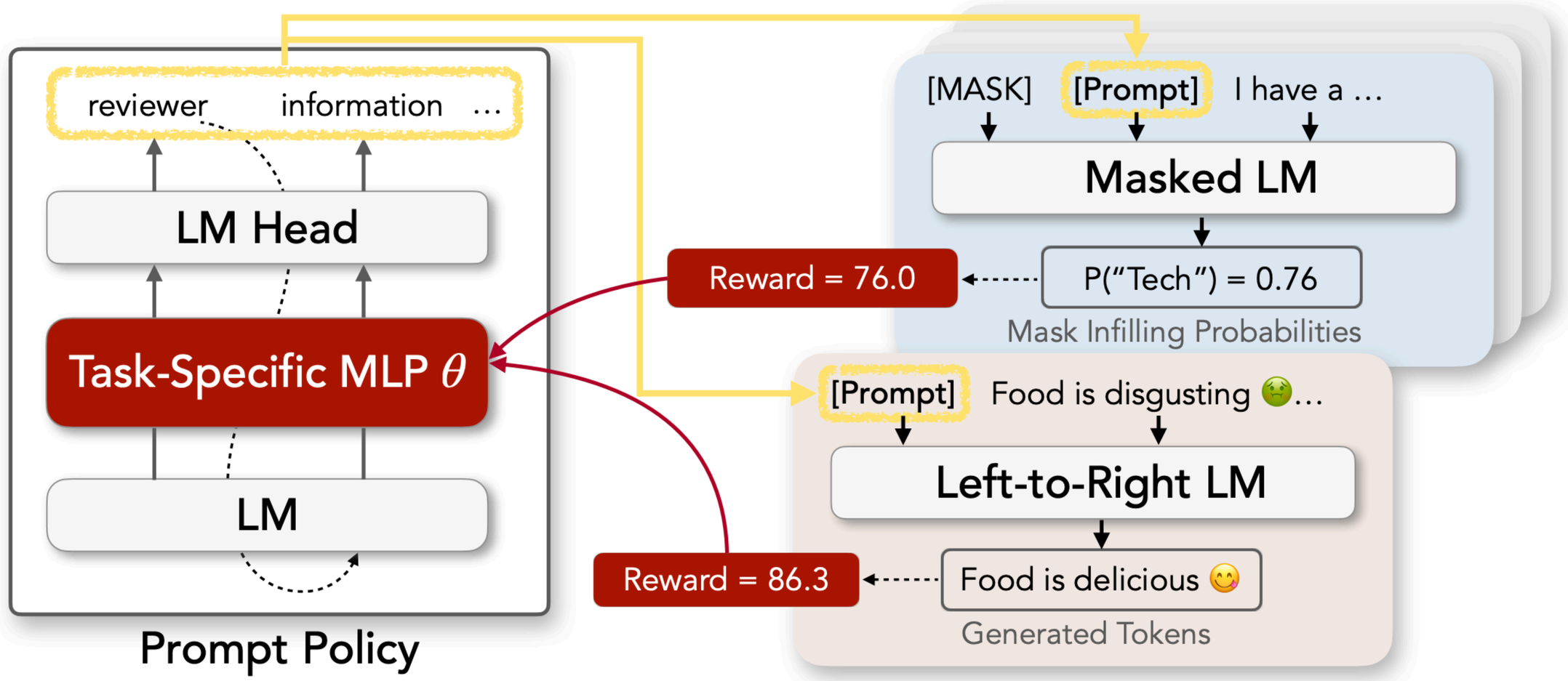
The Prompt Optimization Problem



The Reinforcement Learning Formulation

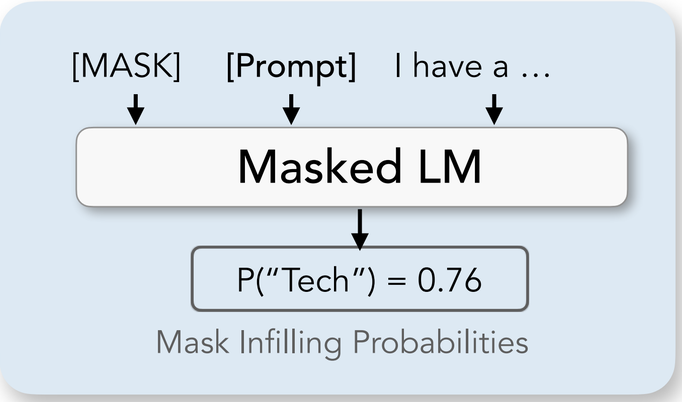


Concrete Overview of the Process

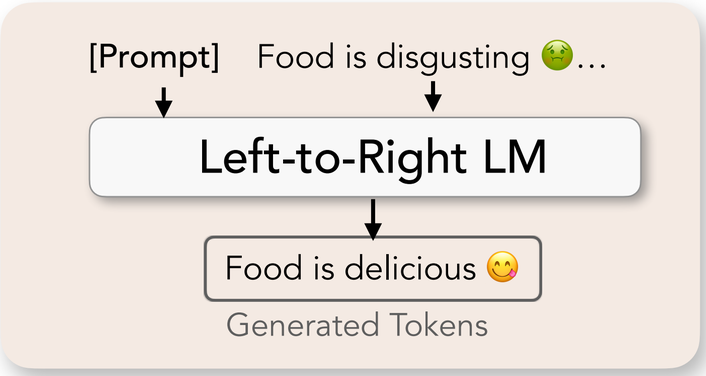


Experiments / Example Use Cases

- Few-Shot Text Classification
  (e.g., sentiment and topic analysis)



- Unsupervised, Controlled Text Generation
  (e.g., text style transfer - rewriting Shakespeare into modern English)

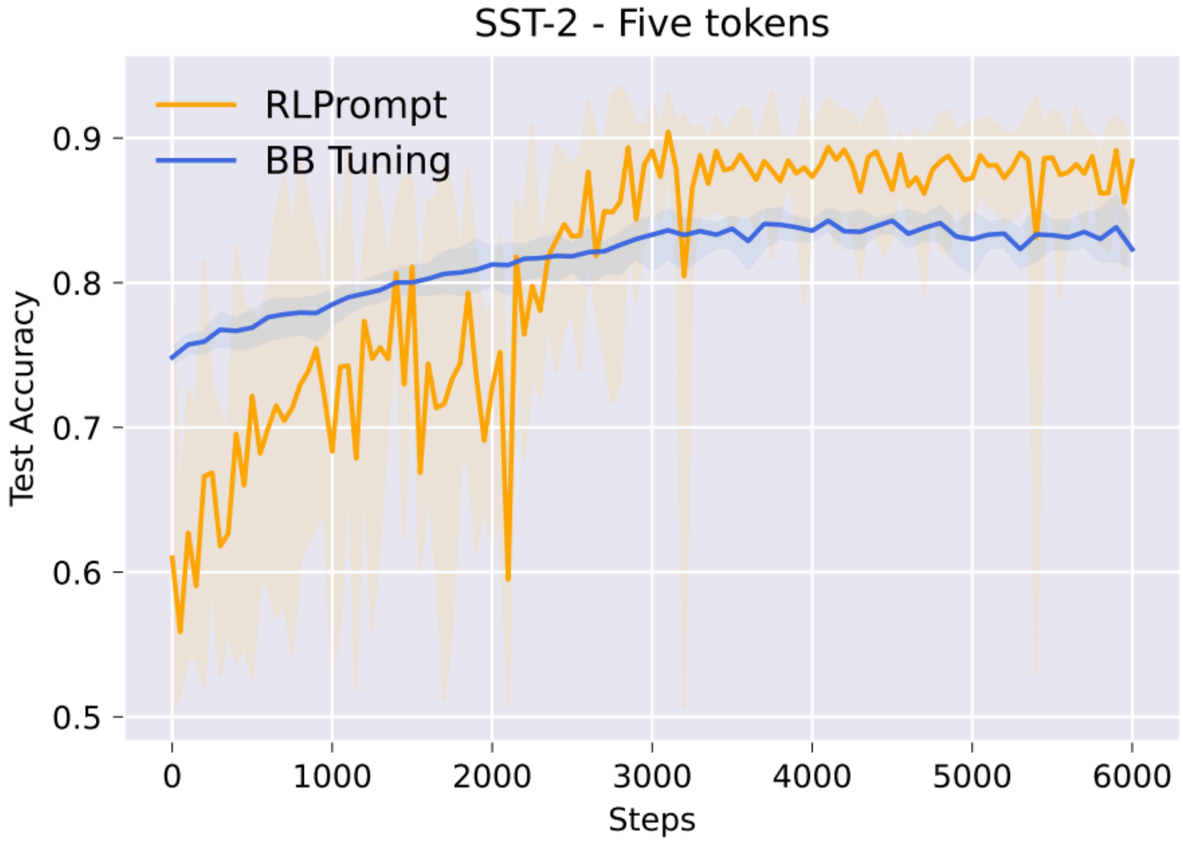


Experiments - Few-Shot Classification

Setting: 16 training examples per category

Reward Function: Gap between probabilities of correct and incorrect categories

	Avg. Acc.
Fine-Tuning	69.5
Manual Prompt	68.6
Instructions	58.5
In-Context Demo.	72.2
Prompt Tuning *(Soft Prompt Tuning)*	69.2
Black-Box Tuning *(2 soft tokens)*	69.2
Black-Box Tuning *(5 soft tokens)*	67.1
Black-Box Tuning *(Mixed, 50 soft tokens)*	74.7
GrIPS *(Discrete Prompt Enumeration)*	69.4
AutoPrompt	56.7
RLPrompt (Ours, 2 discrete tokens)	74.4
RLPrompt (Ours, 5 discrete tokens)	**75.8**

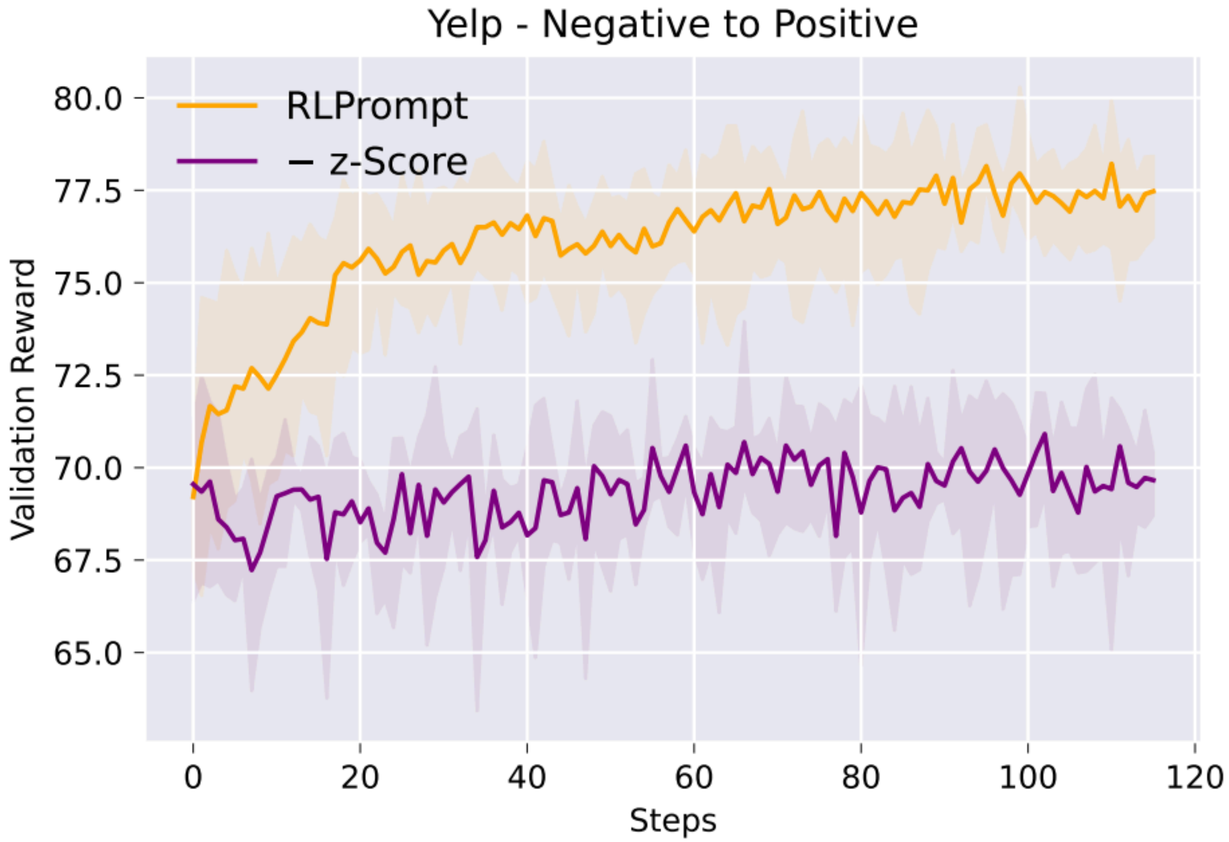


Experiments - Unsupervised Text Style Transfer

Setting: No training example, just two corpora representing each style

Reward Function: Metrics for style correspondence and content preservation

Model	$J(C, S, F)$
Training Baselines	
Style Transformer	46.1
DiRR	59.6
Prompting Baselines (GPT-2-xl)	
Null Prompt	33.6
Random Prompt	34.7
Manual Prompt	53.4 (7.9)
RLPROMPT *(Ours)*	
distilGPT-2	46.0 (0.9)
GPT-2-small	50.7 (1.3)
GPT-2-medium	56.1 (1.0)
GPT-2-large	56.5 (1.3)
GPT-2-xl	**61.4** (2.2)



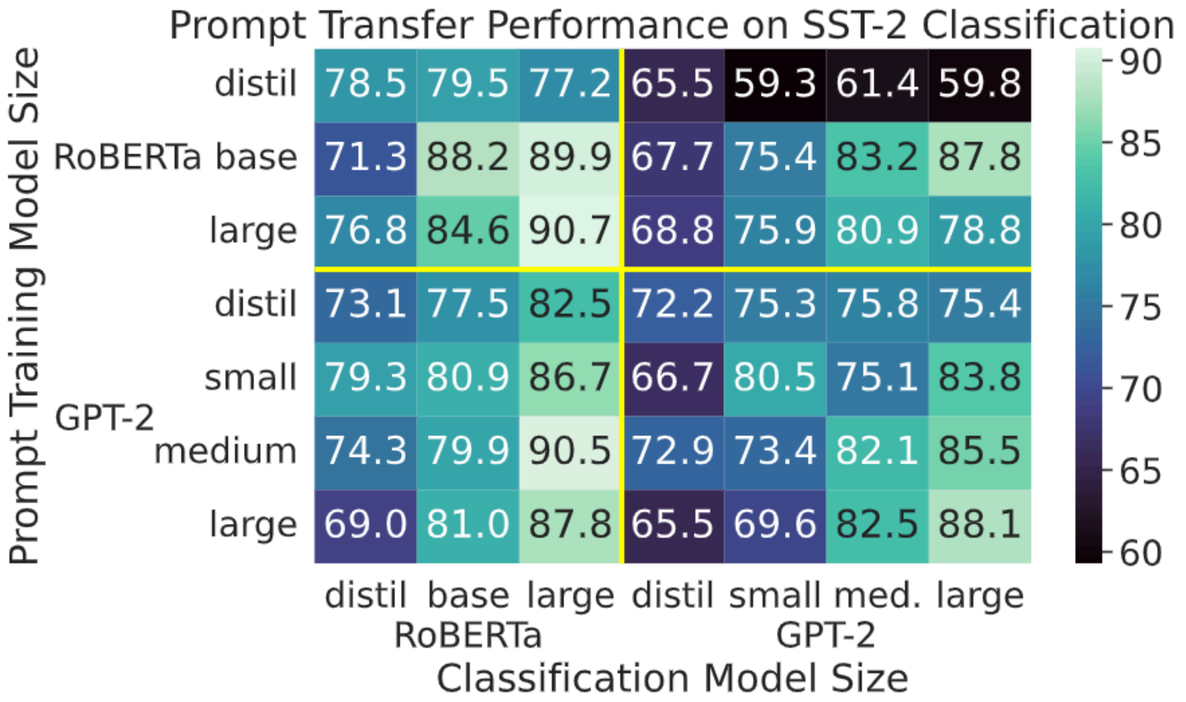
Analysis Insights

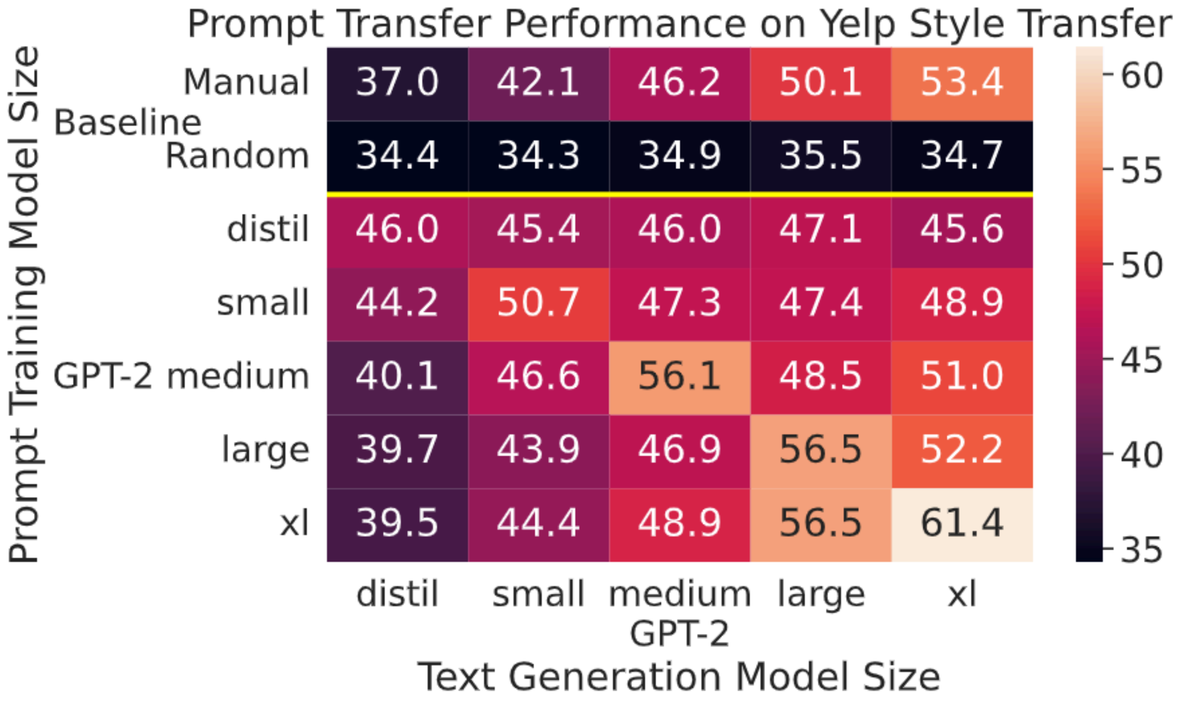
- Optimal prompts don’t follow human language
- Optimized prompts transfer trivially across models

Analysis - Optimal prompts don't follow human language

Prompt Template [to negative | to positive]	$J(C, S, F)$
Manual Prompt	
Here is some text: "{input}". Here is a rewrite of the text, which is more [negative | positive]: "	62.3
Change the following sentence from [positive | negative] sentiment to [negative | positive] sentiment but keep its semantics. "{input}" "	50.5
"{input}". Rewrite the sentence to be [sadder | happier] but have the same meaning. "	47.4
RLPROMPT *(Ours)*	
[Fixed (– contrasts (– contrasts | Dutch English excellent Correct (>] "{input}" "	**62.8**
[Fixed RemovedChanged Prevent outcomes | Parameters Comparison)=(Compare either] "{input}" "	58.9
[Affect differed judgments (– analysis | Difference experiences (– contrasting experience] "{input}" "	62.6

Analysis - Learned prompts transfer trivially across LMs





Codebase and Library on GitHub

Demo

Takeaways

- Optimizing prompts for LLMs is challenging should be done carefully
- The best way to talk to LLMs is not necessarily human language
- RL-based optimization allows us to find out the best prompts effectively

Thank You!

Blog Post

Paper

Code & API

Email: mingkaid@cs.cmu.edu
Twitter: twitter.com/mdeng34
Website: mingkaid.github.io