Text Generation with Prompting & PLMs

Presenter: Eunchan Lee (Research Intern, UNIST)

UNIST Language & Intelligence Lab

Outline

Step 1: Text Generation
(Before & After GPT)

Step 2: GPT-3 &
In-Context Learning (ICL)

Step 3: Prompting

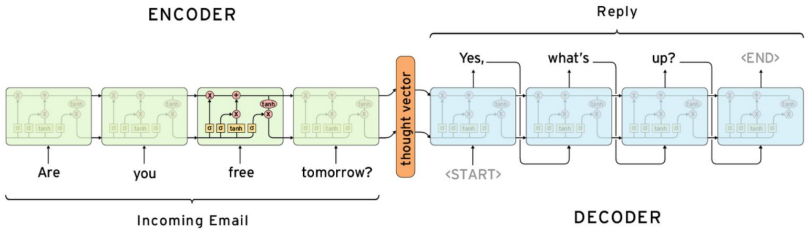
Step 4: What makes ICL work?

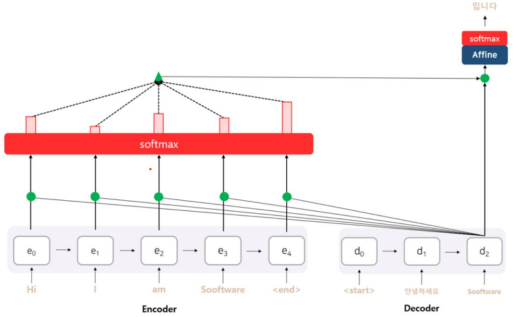
Step 5: Variants of Prompting
(Chain-of-Thought Prompting
/ Prompting for Multimodal)

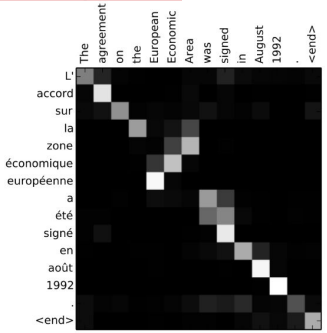
Step 6: Limitations & Future
Directions of Prompting

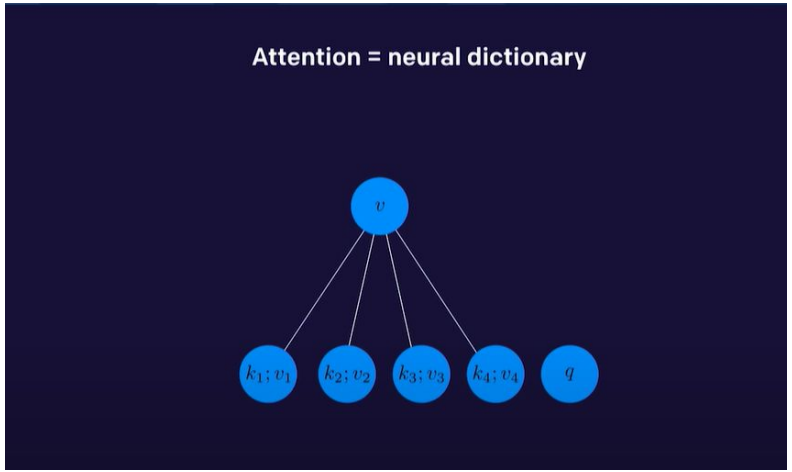
Text Generation (Before GPT)

Attention mechanism & seq2seq (via LSTMs): Predicting the Next Word sufficiently well.





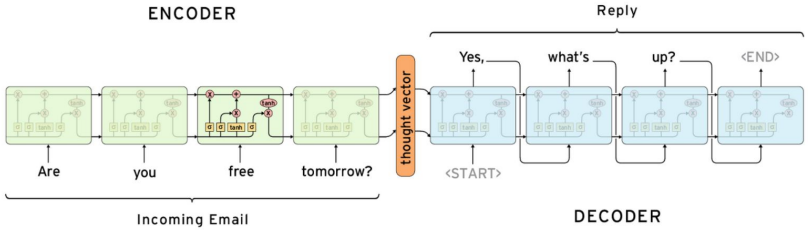


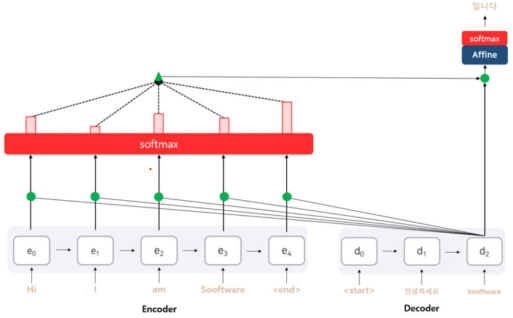


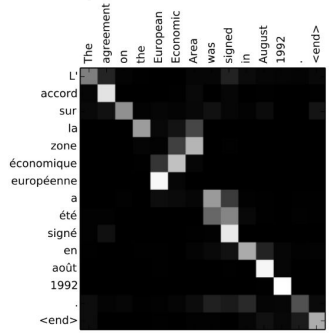
Figures from @soofware & @aistudy9314(blog)

Text Generation (Before GPT)

Attention mechanism & seq2seq (via LSTMs): Predicting the Next Word sufficiently well.



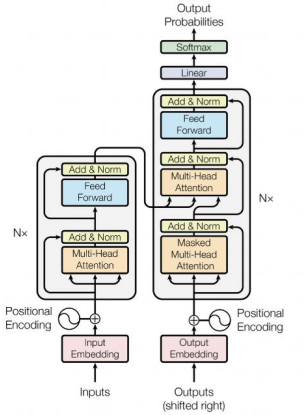


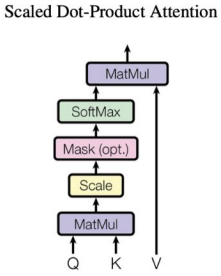


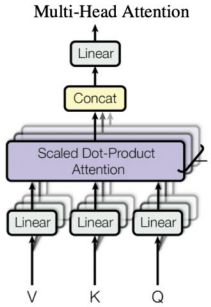
Transformer: improve seq2seq's next word prediction by using only attention mechanisms.

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	







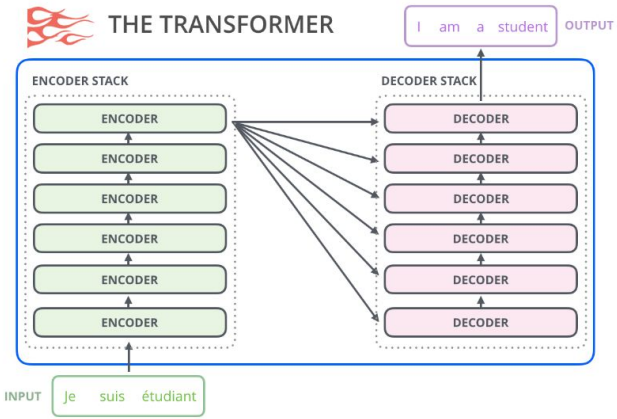
Figures from @soofware & @aistudy9314(blog)

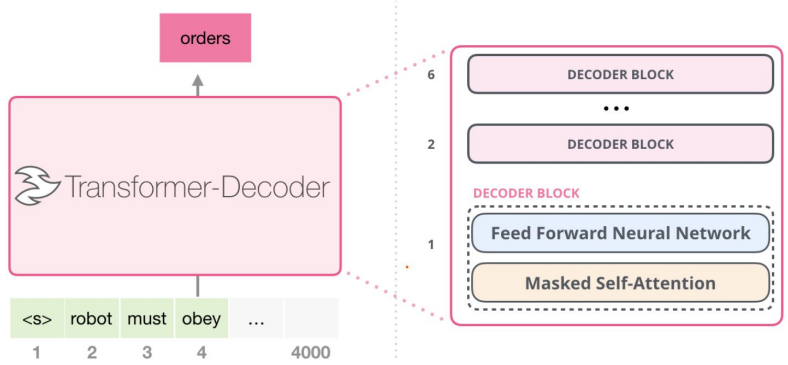
Text Generation (GPT)

Decoder-Only Block & Unsupervised Learning & Scaling Up: lead from Transformer to GPT-2 (2019)

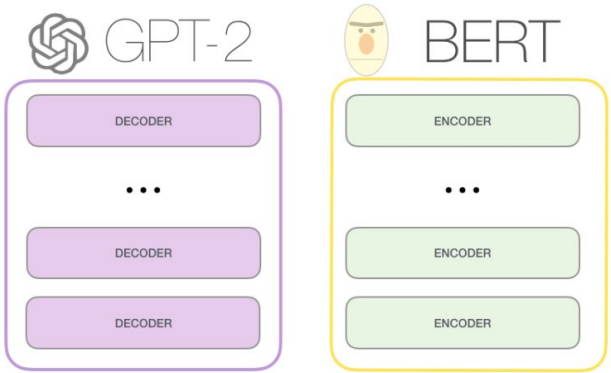
Text Generation (GPT)

Decoder-Only Block & Unsupervised Learning & Scaling Up: lead from Transformer to GPT-2 (2019)





- Encoder-Only ⟶ BERT (2019)
- Decoder-Only ⟶ GPT-2 (2019)



Figures from @JayAlammar's post
(If you interested in GPT-2, this post will help you.)

Text Generation (GPT)

Decoder-Only Block & Unsupervised Learning & Scaling Up: lead from Transformer to GPT-2 (2019)

(dataset)

WebText = lots of variety

(for unsupervised Pre-training)

- *From external links shared at least 3 times on Reddit*
- *Human-annotated filtering*
- *Deduplications (중복 제거작업)*
- *40GB of text*

(Main point)

The Goal of GPT-2

Language Model that performs multitasks only through unsupervised pre-training without fine-tuning. (Zero-Shot)

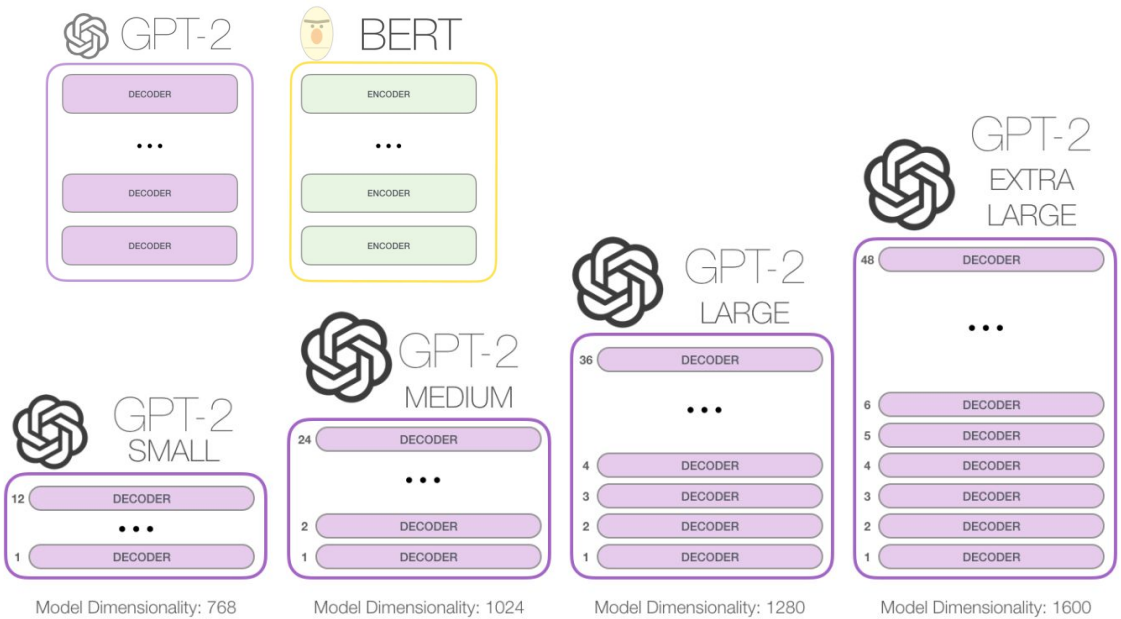
Training Objective:

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, ..., s_{n-1})$$

OpenAI trained GPT-2 with 100 Volta GPU (V100) for a week.

Text Generation (GPT)

Decoder-Only Block & Unsupervised Learning & Scaling Up: lead from Transformer to GPT-2 (2019)



< GPT-2 >

- **Small (0.1B)**
- **Medium (0.3B)**
- **Large (0.76B)**
- **XL (1.5B)**

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

Figures from @JayAlammar's post
(If you interested in GPT-2, this post will help you.)

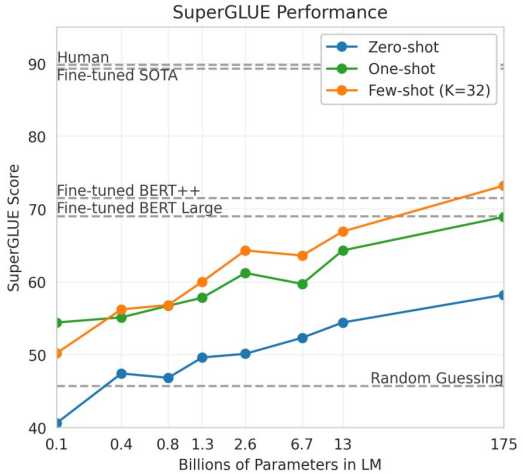
GPT-3 (2020)

- Scale up to 175B
- Language Models are Few-shot Learners
- Can achieve strong performances on *text understanding-based tasks* using few-shot left-to-right text generation.

< GPT-2 >

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

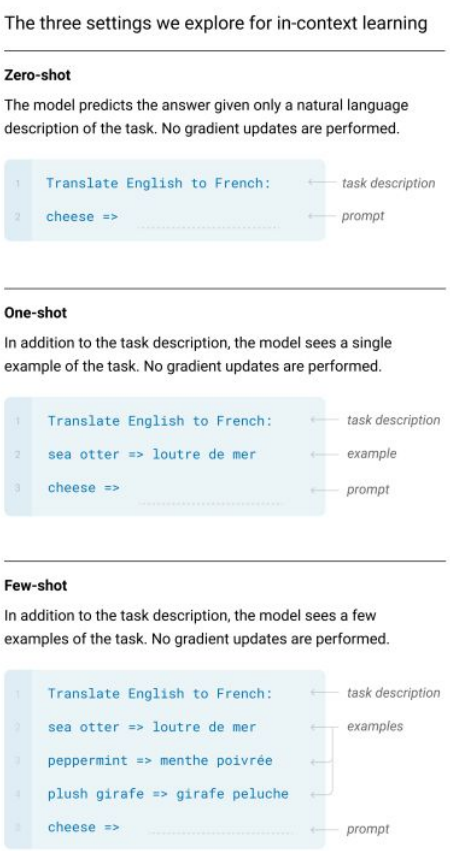
Table 2. Architecture hyperparameters for the 4 model sizes.



Model Name	n_{params}	n_{layers}	d_{model}
GPT-3 Small	125M	12	768
GPT-3 Medium	350M	24	1024
GPT-3 Large	760M	24	1536
GPT-3 XL	1.3B	24	2048
GPT-3 2.7B	2.7B	32	2560
GPT-3 6.7B	6.7B	32	4096
GPT-3 13B	13.0B	40	5140
GPT-3 175B or "GPT-3"	175.0B	96	12288

Figures from @GPT-3_PAPER

In-context Learning (GPT3; Brown et al., 2020)



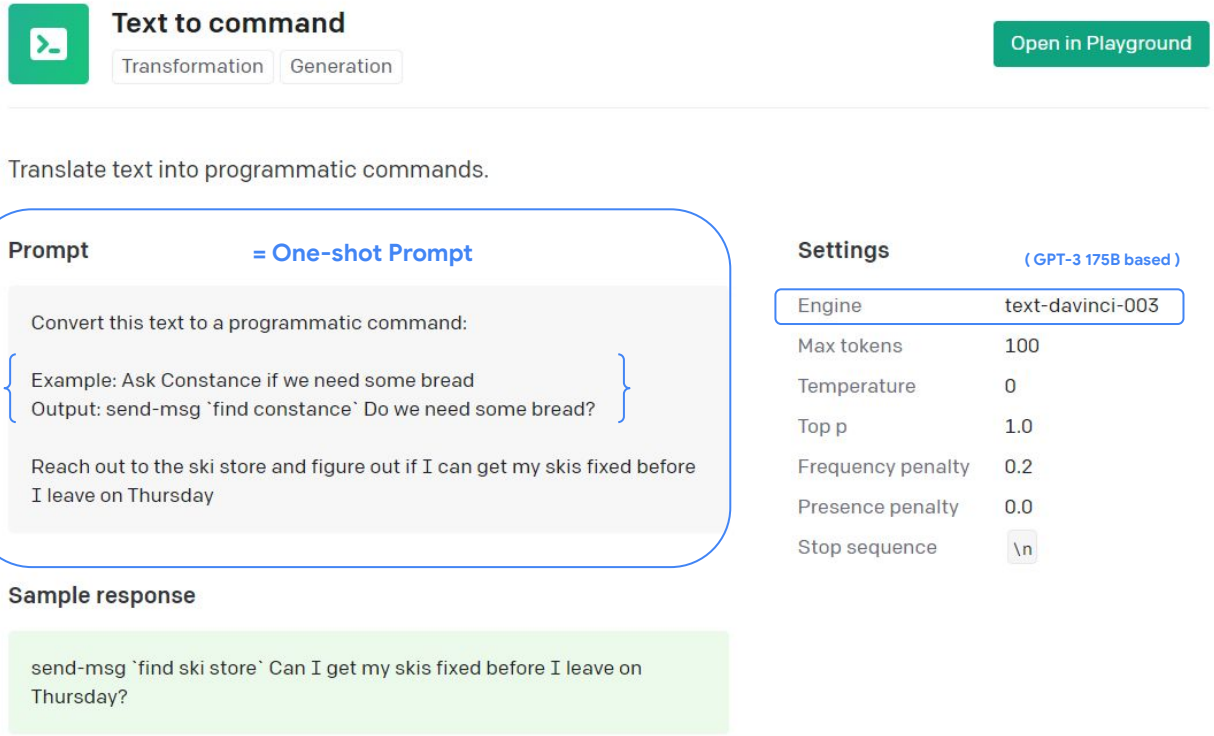
By using **'Prompting'**,
GPT3 performs learning w/o gradient updates!

For all tasks, GPT-3 is applied *without any gradient updates or fine-tuning*, with tasks and *few-shot demonstrations specified purely* ***via text interaction with the model.(=prompting)***

Use case: *Configure text input like a blue box (left) to generate right task answers.*

GPT's Prompting Example

▶**GPT's Prompting = Task description + example of data (zero-, one-, few-shot)**



What is Prompting?

▶Prompting is like a sort of interacting with the model with text-direction. It basically aims to few-shot or zero-shot with Large (Pretrained) LMs.

prompt-based learning is based on language models that model the probability of text directly. To use these models to perform prediction tasks, the original input x is modified using a template into a textual string prompt x 0 that has some unfilled slots, and then the language model is used to probabilistically fill the unfilled information to obtain a final string xˆ, from which the final output y can be derived. This framework is powerful and attractive for a number of reasons: it allows the language model to be pre-trained on massive amounts of raw text, and by defining a new prompting function the model is able to perform few-shot or even zero-shot learning, adapting to new scenarios with few or no labeled data.

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

Pengfei Liu
Carnegie Mellon University
pliu3@cs.cmu.edu

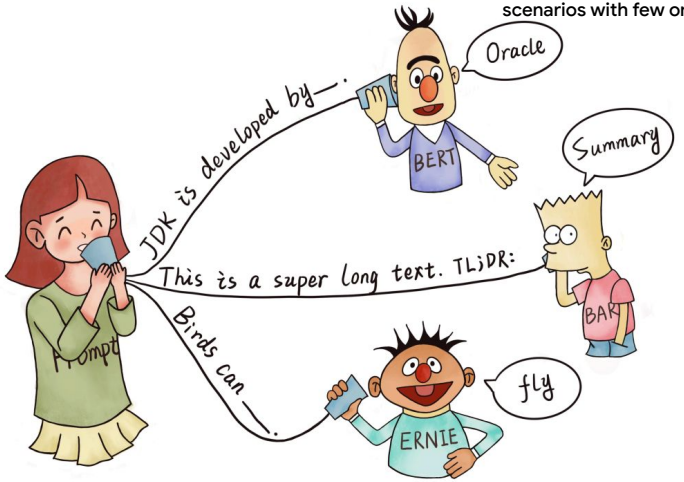
Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Zhengbao Jiang
Carnegie Mellon University
zhengbaj@cs.cmu.edu

Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

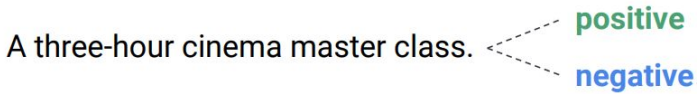


Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing { 2021, Liu et al. (Carnegie Mellon University) }

A simple example of Prompting

(▸Perform the task w/o fine-tuning. How does it work?)

For example, let's say we consider sentiment analysis on movie review.



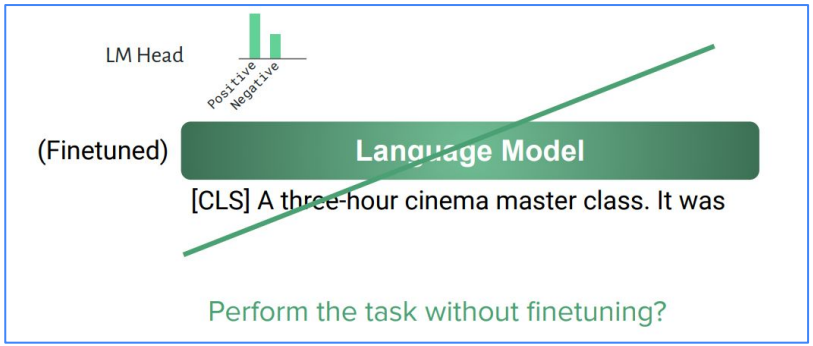
Figures from
@ACL22Tutorial

A simple example of Prompting

(▸Perform the task w/o fine-tuning. How does it work?)

For example, let's say we consider sentiment analysis on movie review.

A three-hour cinema master class. → positive / negative



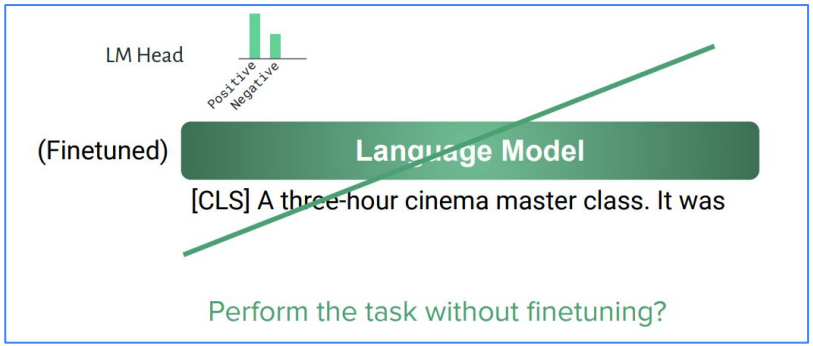
Figures from
@ACL22Tutorial

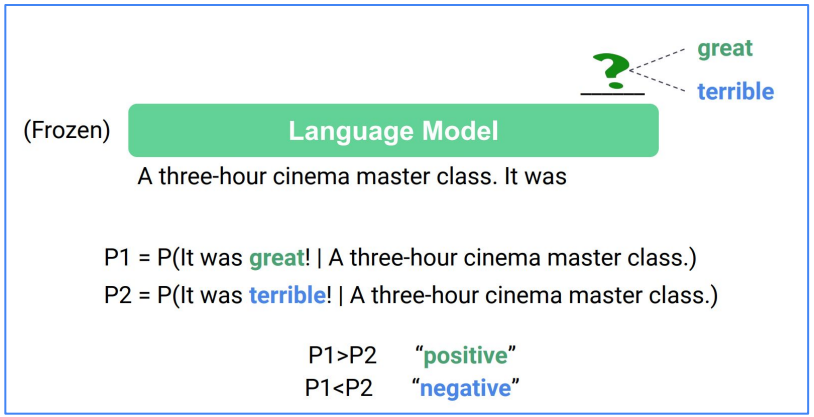
A simple example of Prompting

(▸Perform the task w/o fine-tuning. How does it work?)

For example, let's say we consider sentiment analysis on movie review.

A three-hour cinema master class. → positive / negative





Figures from
@ACL22Tutorial

In-Context Learning

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

An effortlessly accomplished and richly resonant work. It was great!

A mostly tired retread of several other mob tales. It was terrible!

A three-hour cinema master class. It was _____

Language Model

P1 = P(It was **great**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P2 = P(It was **terrible**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

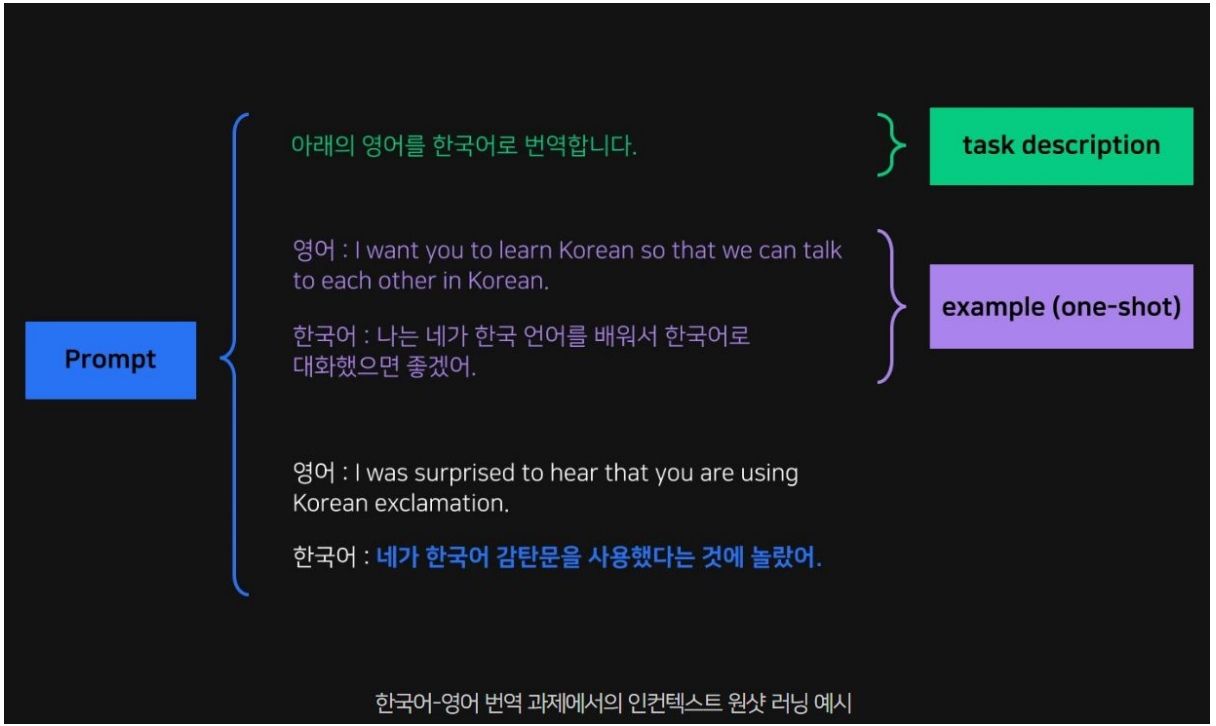
P1>P2 "**positive**"

P1<P2 "**negative**"

Figures from
@ACL22Tutorial

A nice example of In-Context Learning

▸**A nice example of in-context learning for English→Korean neural machine translation**



Few-shot Machine translation (English to Korean)

What makes ICL work? (EMNLP' 22)

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] **Xinxi Lyu**[1] **Ari Holtzman**[1] **Mikel Artetxe**[2]
Mike Lewis[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,alrope,ahai,hannaneh,lsz}@cs.washington.edu
{artetxe,mikelewis}@meta.com

In order to what makes in-context learning work, they performed various factor analyses as shown below.
(Used two LLMs: GPT-J, MetaICL)

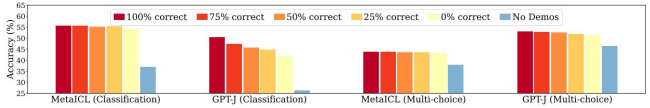


Figure 4: Results with varying number of correct labels in the demonstrations. Channel and Direct used for classification and multi-choice, respectively. Performance with no demonstrations (blue) is reported as a reference.

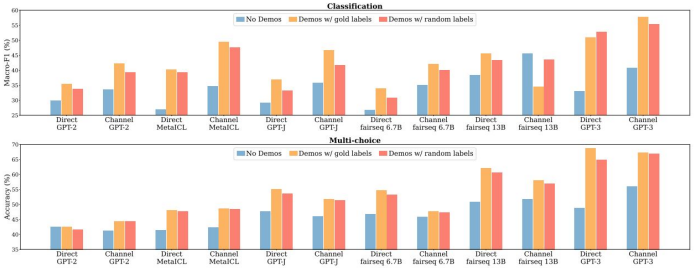


Figure 3: Results when using no-demonstrations, demonstrations with gold labels, and demonstrations with random labels in classification (top) and multi-choice tasks (bottom). The first eight models are evaluated on 16 classification and 10 multi-choice datasets, and the last four models are evaluated on 3 classification and 3 multi-choice datasets. See Figure 11 for numbers comparable across all models. **Model performance with random labels is very close to performance with gold labels** (more discussion in Section 4.1).

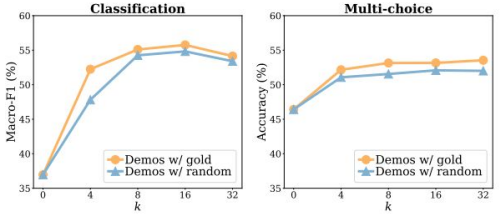


Figure 5: Ablations on varying numbers of examples in the demonstrations (k). Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used.

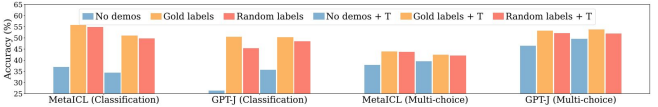


Figure 6: Results with minimal templates and manual templates. '+T' indicates that manual templates are used. Channel and Direct used for classification and multi-choice, respectively.

[Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? (2022, Min et al.)]

What makes ICL work? (EMNLP' 22)

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] **Xinxi Lyu**[1] **Ari Holtzman**[1] **Mikel Artetxe**[2]
Mike Lewis[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,alrope,ahai,hannaneh,lsz}@cs.washington.edu
{artetxe,mikelewis}@meta.com

They performed various factor analyses.

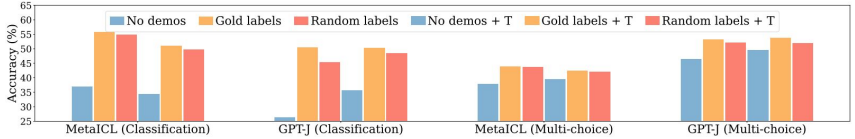


Figure 6: Results with minimal templates and manual templates. '+T' indicates that manual templates are used. Channel and Direct used for classification and multi-choice, respectively.

Does the model learn at test time(=in-context learning)?

If we take a strict definition of learning: capturing the input label correspondence given in the training data, then our findings suggest that ***LMs do not learn new tasks at test time***.
Our analysis shows that the model may ignore the task defined by the demonstrations and instead use prior from pretraining.

However, learning a new task can be interpreted more broadly: it may include adapting to specific input and label distributions and the format suggested by the demonstrations, and ultimately getting to make a prediction more accurately. With this definition of learning, *the* ***model does learn the task from the demonstrations.***

The model adapts to make predictions more accurate, but does not learn new tasks at test time.

[Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? (2022, Min et al.)]

What makes ICL work? (EMNLP' 22)

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] **Xinxi Lyu**[1] **Ari Holtzman**[1] **Mikel Artetxe**[2]
Mike Lewis[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,alrope,ahai,hannaneh,lsz}@cs.washington.edu
{artetxe,mikelewis}@meta.com

They performed various factor analyses.

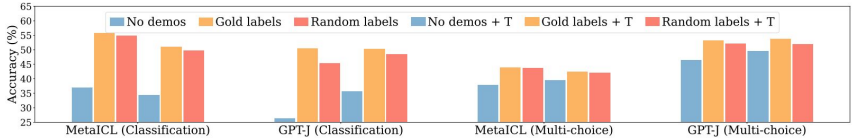


Figure 6: Results with minimal templates and manual templates. '+T' indicates that manual templates are used. Channel and Direct used for classification and multi-choice, respectively.

Capacity of LMs

The model performs a downstream task without relying on the input-label correspondence from the demonstrations.

This suggests that the model has learned the (implicit notion of) input-label correspondence from the language modeling objective alone, e.g., associating a positive review with the word 'positive'.

This is in line with Reynolds and McDonell (2021) who claim that the demonstrations are for task location and the intrinsic ability to perform the task is obtained at pretraining time.

Demonstrations are for task location and the intrinsic ability to perform the task is obtained at pretraining time.

[Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? (2022, Min et al.)]

How/Why in-context learning works?

- Demonstrations do not teach a new task; instead, it is about locating an already-learned task during pretraining (Reynolds & McDonell, 2021)
- LMs do not exactly understand the meaning of their prompt (Webson & Pavlick, 2021)
- Demonstrations are about providing a latent concept so that LM generates coherent next tokens (Xie et al. 2022)
- In-context learning performance is highly correlated with term frequencies during pretraining (Razeghi et al. 2022)
- LMs do not need input-label mapping in demonstrations, instead, it uses the specification of the input & label distribution separately (Min et al. 2022)
- Data properties lead to the emergence of few-shot learning (burstiness, long-tailedness, many-to-one or one-to-many mappings, a Zipfian distribution) (Chan et al. 2022)

Chain-of-Thought Prompting (NeurIPS' 22)

(A variant of Prompting, more improved.)

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

a few chain of thought demonstrations are provided as exemplars in prompting.

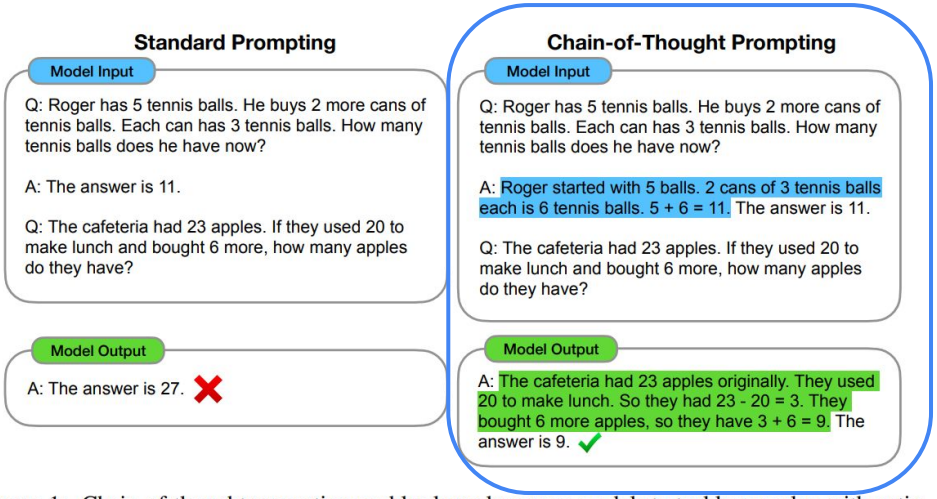
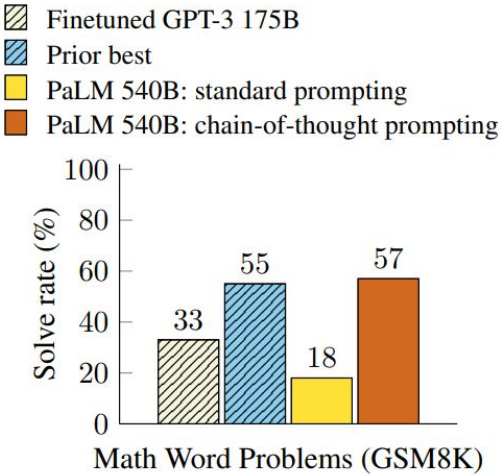


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.



https://arxiv.org/abs/2201.11903

Chain-of-Thought Prompting (NeurIPS' 22)

(A variant of Prompting, more improved.)

Math Word Problems (free response)

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.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation

Q: Take the last letters of the words in "Lady Gaga" and concatenate them.

A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

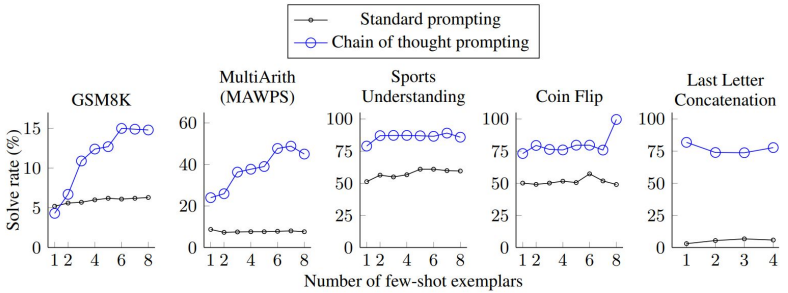
Coin Flip (state tracking)

Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

Task-specific Chain-of-Thought Prompting also improves the task accuracy. (e.g. Reasoning Coin Flip)



https://arxiv.org/abs/2201.11903

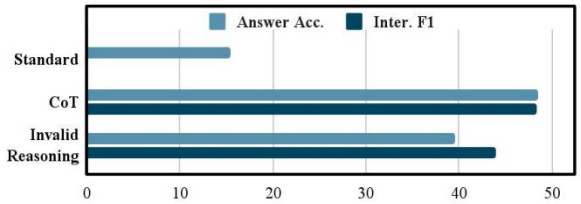
Why does Chain-of-Thought Prompting work?

Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters

Boshi Wang[1] **Sewon Min**[2] **Xiang Deng**[1] **Jiaming Shen**[3] **You Wu**[3]
Luke Zettlemoyer[2] **Huan Sun**[1]
[1]The Ohio State University [2]University of Washington [3]Google Research
{wang.13930,deng.595,sun.397}@osu.edu
{sewon,lsz}@cs.washington.edu, {jmshen,wuyou}@google.com

series of ablation experiments that unveil the impact of different aspects of a CoT rationale. We find that 1) the validity of reasoning matters only a small portion to the performance; 2) relevance to the input query and following the order along the reasoning steps are the key to the effectiveness of CoT prompting. Overall, our findings deepen the

	In-context Demonstration	Inference by LLM
Query	Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?	Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?
Standard	The answer is 39.	The answer is 18. ✗
CoT	Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39 pieces left in total. The answer is 39.	Julie is reading a 120-page book. Yesterday, she read 12 pages and today, she read 24 pages. So she read a total of 12 + 24 = 36 pages. Now she has 120 - 36 = 84 pages left. Since she wants to read half of the remaining pages, she should read 84 / 2 = 42 pages. The answer is 42. ✓
Invalid Reasoning	Originally, Leah had 32 chocolates and her sister had 42. So her sister had 42 - 32 = 10 chocolates more than Leah has. After eating 35, since 10 + 35 = 45, they had 45 - 6 = 39 pieces left in total. The answer is 39.	Yesterday, Julie read 12 pages. Today, she read 12 * 2 = 24 pages. So she read a total of 12 + 24 = 36 pages. Now she needs to read 120 - 36 = 84 more pages. She wants to read half of the remaining pages tomorrow, so she needs to read 84 / 2 = 42 pages tomorrow. The answer is 42. ✓



[Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters (Wang et al., Dec 2022)]

Least-to-Most Prompting

(Another variant of Chain-of-Thought Prompting)

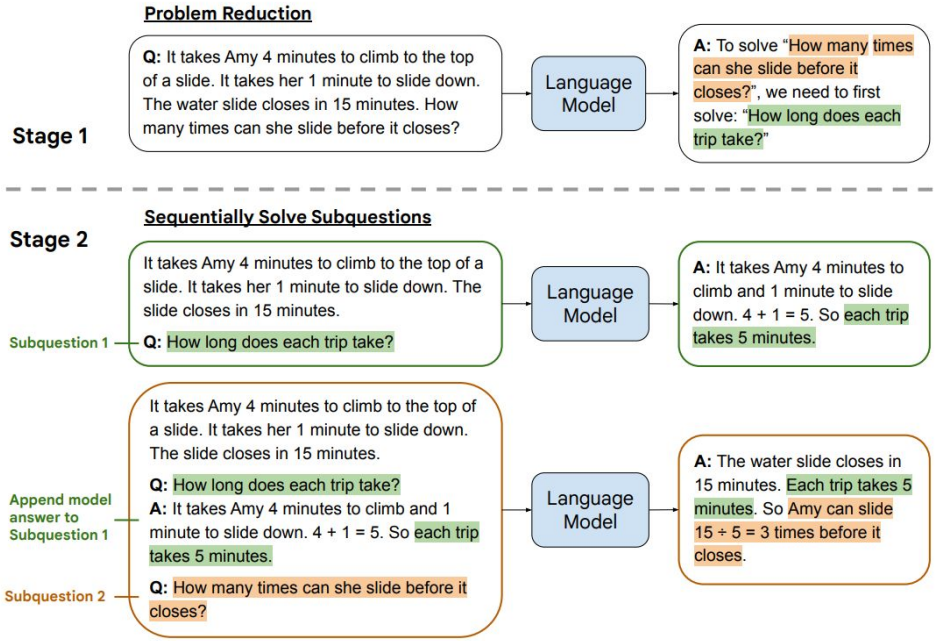
Google Research

LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS

Denny Zhou* **Nathanael Schärli** **Le Hou** **Jason Wei** **Nathan Scales** **Xuezhi Wang**

Dale Schuurmans **Claire Cui** **Olivier Bousquet** **Quoc Le** **Ed Chi**

Google Research



Least-to-Most Prompting Enables Complex Reasoning in Large Language Models (2022, Zhou et al. (Google))

Least-to-Most Prompting

(Another variant of Chain-of-Thought Prompting)

Google Research

LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS

Denny Zhou* **Nathanael Schärli** **Le Hou** **Jason Wei** **Nathan Scales** **Xuezhi Wang**

Dale Schuurmans **Claire Cui** **Olivier Bousquet** **Quoc Le** **Ed Chi**

Google Research

Chain-of-thought prompting	Least-to-most prompting (solving stage)
Q: "think, machine" A: The last letter of "think" is "k". The last letter of "machine" is "e". Concatenating "k", "e" leads to "ke". So, "think, machine" outputs "ke". Q: "learning, reasoning, generalization" A: The last letter of "learning" is "g". The last letter of "reasoning" is "g". The last letter of "generalization" is "n". Concatenating "g", "g", "n" leads to "ggn". So, "learning, reasoning, generalization" outputs "ggn".	Q: "think, machine" A: The last letter of "think" is "k". The last letter of "machine" is "e". Concatenating "k", "e" leads to "ke". So, "think, machine" outputs "ke". Q: "think, machine, learning" A: "think, machine" outputs "ke". The last letter of "learning" is "g". Concatenating "ke", "g" leads to "keg". So, "think, machine, learning" outputs "keg".

Table 1: Chain-of-thought (left column) and least-to-most (right column) prompts for the last-letter-concatenation task. Note that in least-to-most, the second example is built on the first one.

Q: "mind, emerge"
A: The last letter of "mind" is "d". The last letter of "emerge" is "e". Concatenating "d", "e" leads to "de". So, "mind, emerge" outputs "de".

Q: "mind, emerge, from"
A: "mind, emerge" outputs "de". The last letter of "from" is "m". Concatenating "de", "m" leads to "dem". So, "mind, emerge, from" outputs "dem".

Q: **"mind, emerge, from, neuron"**
A: "mind, emerge, from" outputs "dem". The last letter of "neuron" is "n". Concatenating "dem", "n" leads to "demn". So, "mind, emerge, from, neuron" outputs "demn".

Table 2: A test case of least-to-most prompting for the last-letter-concatenation task. Generated with `code-davinci-002` in GPT-3. The prompt context is shown on the right column of Table 1.

Method	L = 4	L = 6	L = 8	L = 10	L = 12
Standard prompting	0.0	0.0	0.0	0.0	0.0
Chain-of-Thought	89.4	75.0	51.8	39.8	33.6
Least-to-Most	**94.0**	**88.4**	**83.0**	**76.4**	**74.0**

Table 3: Accuracies of different prompting methods with `code-davinci-002` on the last-letter-concatenation task with the length of lists increasing from 4 to 12. All the methods are 2-shot.

Least-to-Most Prompting Enables Complex Reasoning in Large Language Models (2022, Zhou et al. (Google))

Least-to-Most Prompting

(Another variant of Chain-of-Thought Prompting)

Google Research

LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS

Denny Zhou* **Nathanael Schärli** **Le Hou** **Jason Wei** **Nathan Scales** **Xuezhi Wang**

Dale Schuurmans **Claire Cui** **Olivier Bousquet** **Quoc Le** **Ed Chi**

Google Research

Q: "jump right"
A: The output of "jump right" concatenates: the output of "turn right", the output of "jump". "turn right" outputs "TURN_RIGHT". "jump" outputs "JUMP". So concatenating the output of "turn right" and the output of "JUMP" leads to "TURN_RIGHT" + "JUMP". So the output of "jump right" is "TURN_RIGHT" + "JUMP".

Q: "jump right twice"
A: The output of "jump right twice" concatenates: the output of "jump right", the output of "jump right". "jump right" outputs "TURN_RIGHT" + "JUMP". So repeating the output of "jump right" two times leads to ("TURN_RIGHT" + "JUMP") * 2. So the output of "jump right twice" is ("TURN_RIGHT" + "JUMP") * 2.

Q: **"jump right twice after walk"**
A: The output of "jump right twice after walk" concatenates: the output of "walk", the output of "jump right twice". "walk" outputs "WALK". "jump right twice" outputs ("TURN_RIGHT" + "JUMP") * 2. So concatenating the output of "walk" and the output of "jump right twice" leads to "WALK" + ("TURN_RIGHT" + "JUMP") * 2. So the output of "jump right twice after walk" is "WALK" + ("TURN_RIGHT" + "JUMP") * 2.

Table 7: A test case of least-to-most prompting on SCAN. Generated with `code-davinci-002`. The outputs of longer commands are progressively built on the outputs of shorter commands.

Method	Standard prompting	Chain-of-Thought	Least-to-Most
`code-davinci-002`	16.7	16.2	**99.7**
`text-davinci-002`	6.0	0.0	**76.0**
`code-davinci-001`	0.4	0.0	**60.7**

Table 8: Accuracies (%) of different prompting methods on the test set of SCAN under length split. The results of `text-davinci-002` are based on a random subset of 100 commands.

[Least-to-Most Prompting Enables Complex Reasoning in Large Language Models (2022, Zhou et al. (Google))]

Prompting for Multimodal Reasoning

(Another variant of Prompting)

Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, Pete Florence

Google

4.1 Socratic Image Captioning on MS COCO Captions: VLM + LM

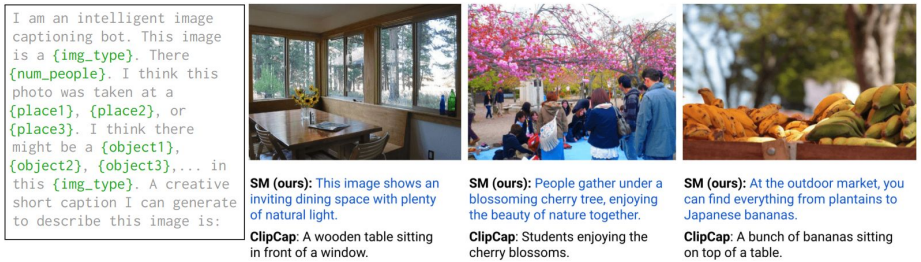


Figure 3: SMs with VLM and LM prompting (left) can zero-shot generate captions for generic Internet images (e.g., from MS COCO), and can be as expressive as task-specific finetuned methods such as ClipCap [45].

Method	BLEU-4	METEOR	CIDEr	SPICE	ROUGE-L
*ClipCap [45]	40.7	30.4	152.4	25.2	60.9
†MAGIC [61]	11.4	16.4	56.2	11.3	39.0
ZeroCap [62]	0.0	8.8	18.0	5.6	18.3
SMs 0-shot (ours)	6.9	15.0	44.5	10.1	34.1
SMs 3-shot (ours)	**18.3**	**18.8**	**76.3**	**14.8**	**43.7**

*finetuned on full training set with image-text pairs.
†finetuned on unpaired training set, zero-shot on image-text pairs.

Table 1: Image captioning comparisons on a random subset of $N = 100$ MS COCO test examples.

[Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language { 2022, Zeng et al. (Google) }]

Limitations and Future Directions of Prompting

Limitations

- **Still in progress on understanding how/why it works**,
  (Can we predict whether in-context learning would work on a given task or not?)

- Need to be cautious in **evaluation**

- We have **only scratched** the surface of using prompts to improve model learning.
  Prompts will become more elaborate(= 더 자세히, 정교하게**)!**

Insights from
@ACL22Tutorial

Limitations and Future Directions of Prompting

Future Directions

Better in-context learning

- Make it less sensitive to the order of training examples or patterns/verbalizers?
- It increases inference cost – how to make it efficient?
- How to scale it (longer context, more training examples)?

Better understanding

- Can we better understand how and why it works?
- Can we predict whether in-context learning would work on a given task or not?

Explainable NLP

Prompts may also be a more natural way to incorporate natural language explanations into model training.

Insights from
@ACL22Tutorial

Thank you for your attention!