Language Model Prompting

Shane Storks

EECS 595: Natural Language Processing

November 16, 2022

Pre-trained LMs

- The SOTA in NLP is dominated by **large-scale, pre-trained language models** (LMs)
  - Train a high-complexity transformer as a language model
  - Use massive amounts of text from the Web for training
  - Apply to downstream tasks
- Examples
  - Google: BERT, PaLM
  - Meta: RoBERTa
  - Baidu: ERNIE
  - OpenAI: GPT, GPT-2, GPT-3
  - Microsoft: Turing NLG

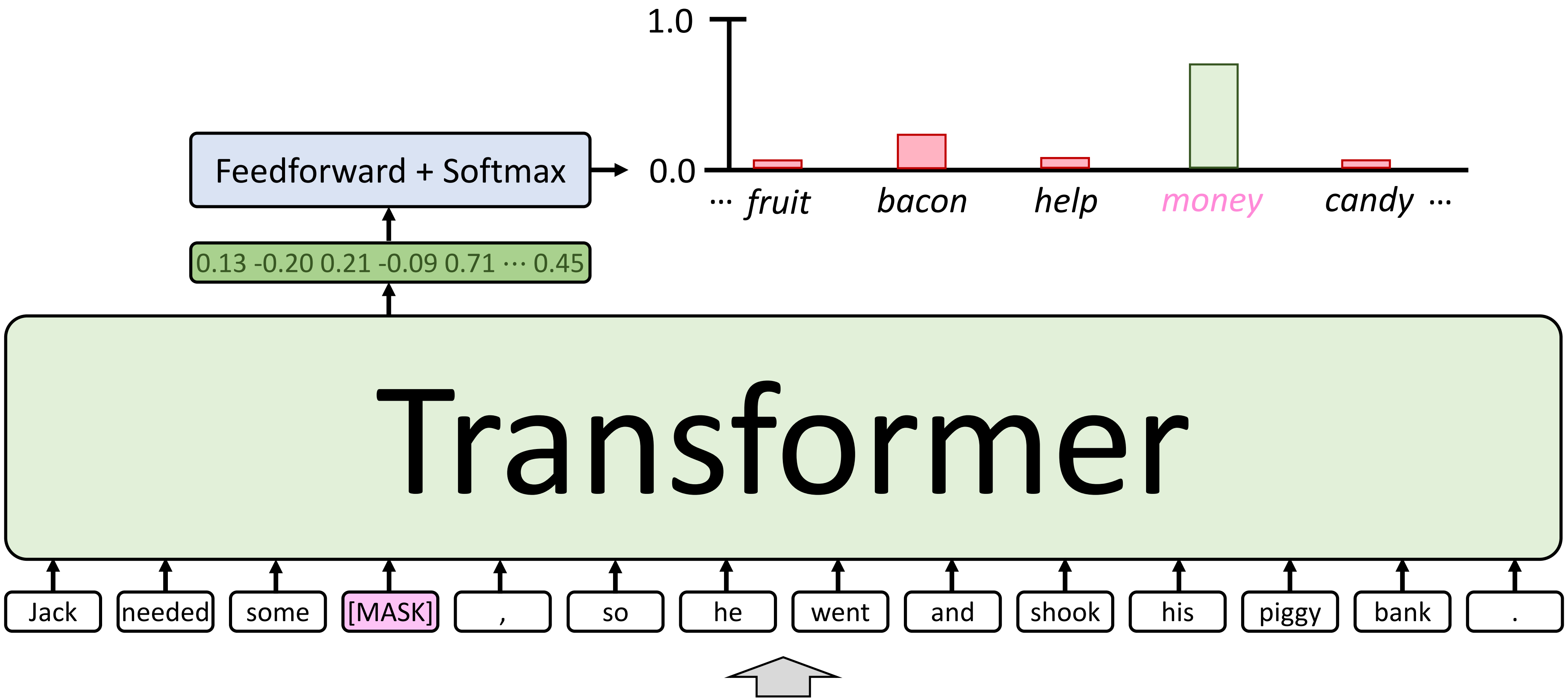
SQuAD1.1 Leaderboard

Here are the ExactMatch (EM) and F1 scores evaluated on the test set of SQuAD v1.1.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Jul 24, 2021	{ANNA} (single model) *LG AI Research*	**90.622**	**95.719**
2 Apr 10, 2020	LUKE (single model) *Studio Ousia & NAIST & RIKEN AIP* https://arxiv.org/abs/2010.01057	90.202	95.379
3 May 21, 2019	XLNet (single model) *Google Brain & CMU*	89.898	95.080
4 Dec 11, 2019	XLNET-123++ (single model) *MST/EOI* http://tia.today	89.856	94.903
4 Aug 11, 2019	XLNET-123 (single model) *MST/EOI*	89.646	94.930
5 Jul 21, 2019	SpanBERT (single model) *FAIR & UW*	88.839	94.635
6 Jul 03, 2019	BERT+WWM+MT (single model) *Xiaoi Research*	88.650	94.393
7 Jul 21, 2019	Tuned BERT-1seq Large Cased (single model) *FAIR & UW*	87.465	93.294
8 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	87.433	93.160
9 May 14, 2019	ATB (single model) *Anonymous*	86.940	92.641
10 Jul 21, 2019	Tuned BERT Large Cased (single model) *FAIR & UW*	86.521	92.617
10 Jul 04, 2019	BERT+MT (single model) *Xiaoi Research*	86.458	92.645

Vaswani, A., Shazeer, N., et al. (2017). Attention Is All You Need. NIPS 2017.

Training a Language Model

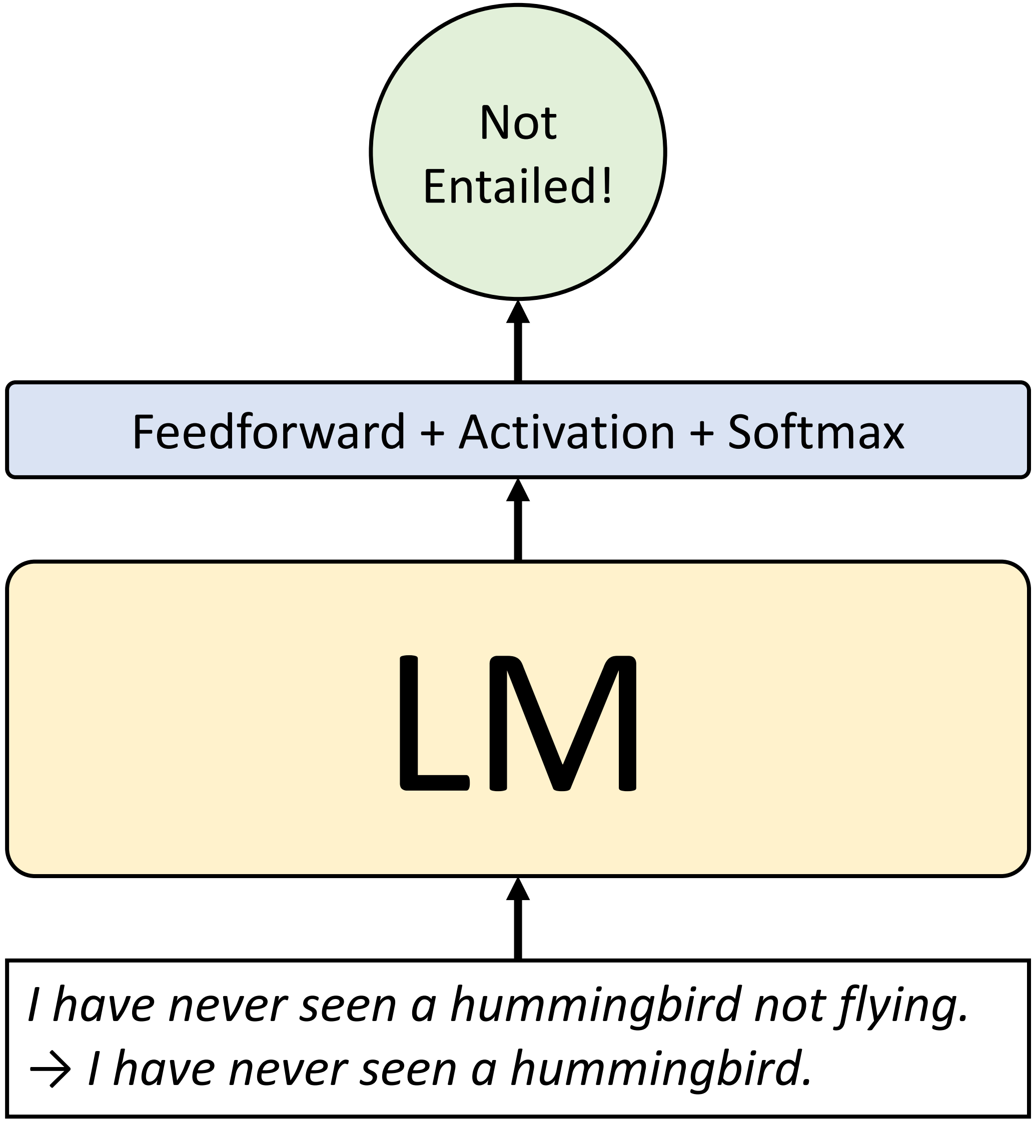


*"Jack needed some **money**, so he went and shook his piggy bank."*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In NAACL HLT 2019.
Vaswani, A. et al. (2017). Attention is All you Need. In NIPS 30.

Fine-Tuning

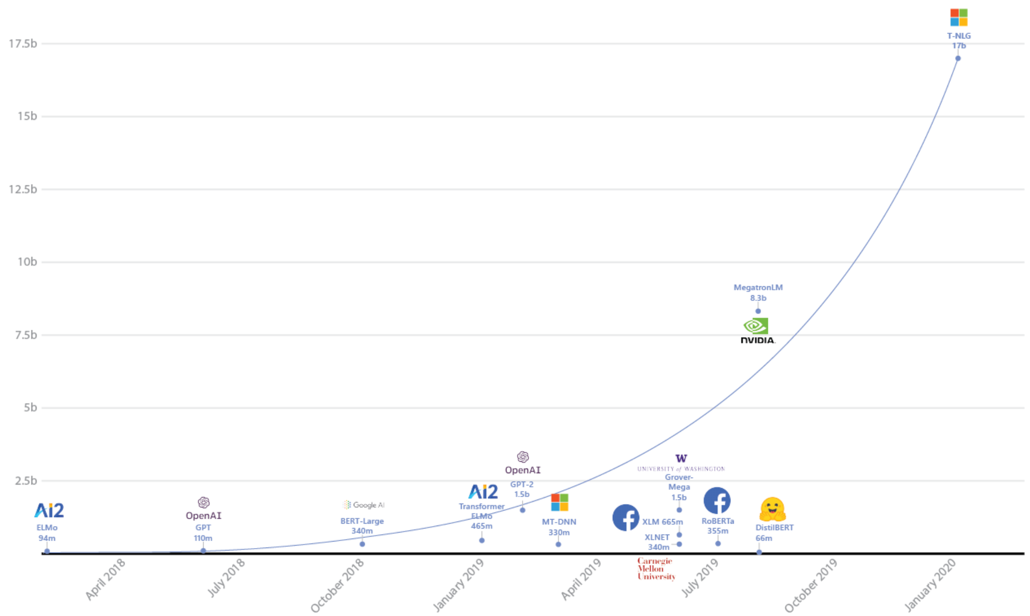
- We can **fine-tune** these LMs on **downstream tasks**
  - Train some classification head to classify LM embeddings
  - End-to-end with LM (back-propagate using downstream task supervision)



Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In NAACL HLT 2019.
Vaswani, A. et al. (2017). Attention is All you Need. In NIPS 30.
Wang, A., et al. (2019). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.

Limitations of Fine-Tuning

- Fine-tuned LMs can exploit biases in language data
  - Achieve artificially high performance (Niven and Kao, 2019)
  - Predictions tend to be supported by incoherent evidence (Storks and Chai, 2021)
- LMs are complex!
  - Limited insight into how conclusions are made
  - Computationally expensive



(figure from [Microsoft](#))

Niven, T. and Kao, H. (2019). Probing Neural Network Comprehension of Natural Language Arguments. ACL 2019.
Storks, S. and Chai, J. (2021). Beyond the Tip of the Iceberg: Assessing Coherence of Text Classifiers. Findings of EMNLP 2021.

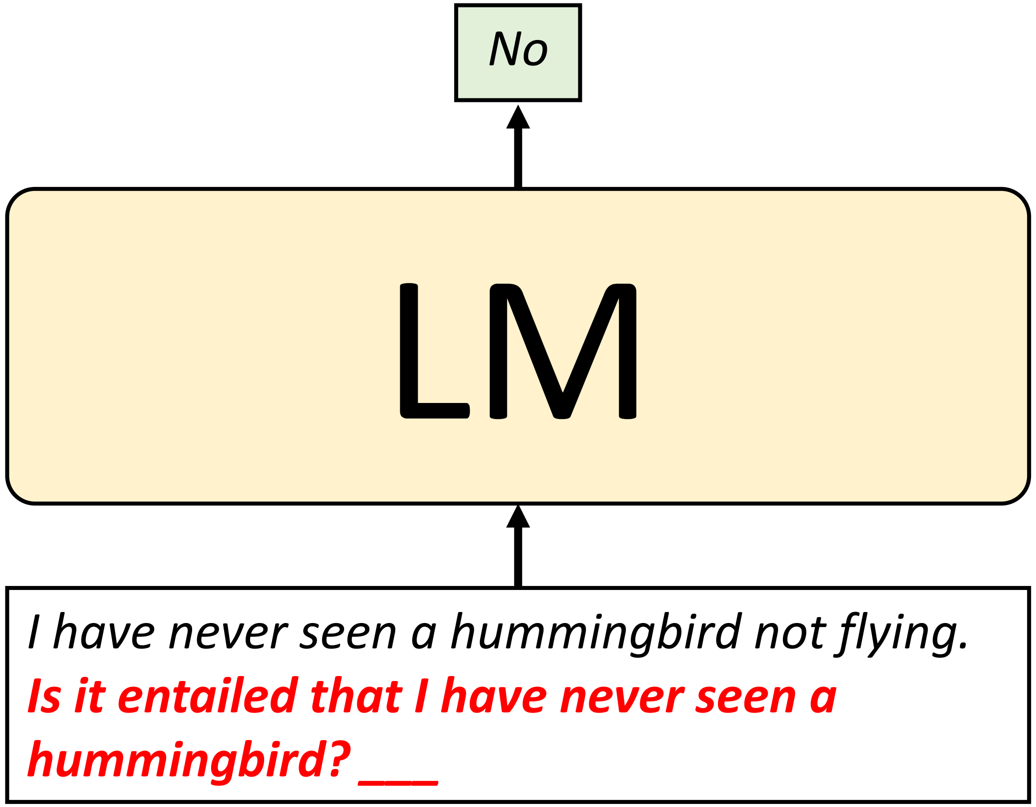
What do LMs Actually Know?

- LMs are trained on massive amounts of text data
- Latest LMs have billions of learned parameters
- What knowledge is captured? How do we extract it?

The Wrap

Prompting

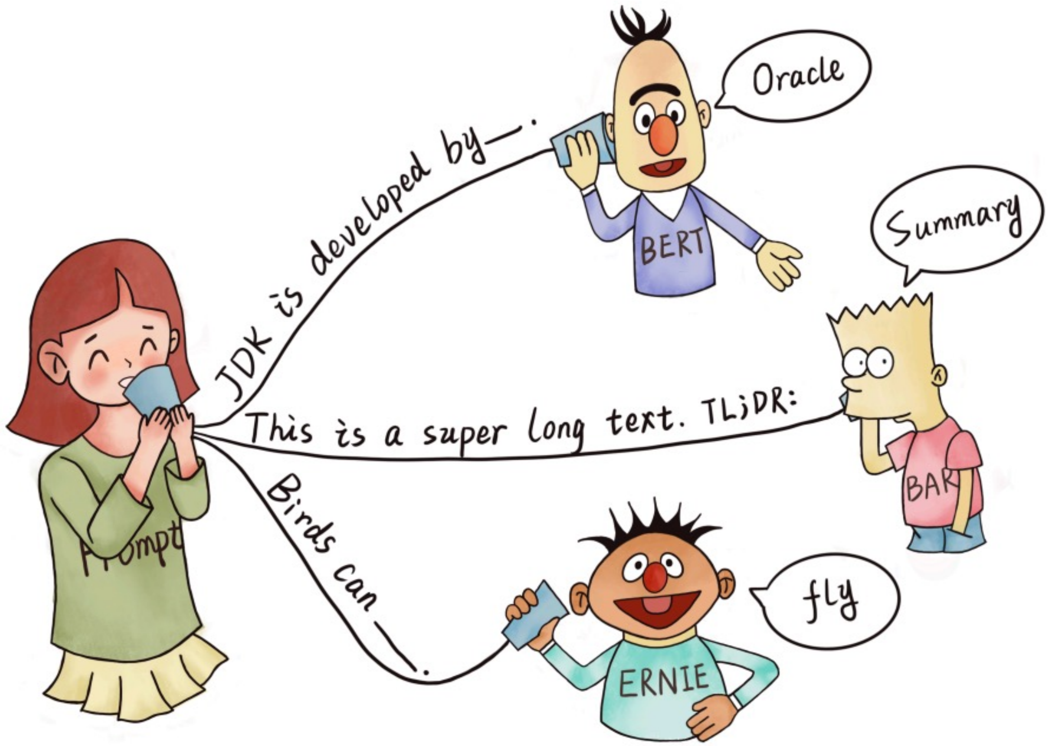
- Don't fine-tune, instead **prompt** the LM with targeted language at inference time!
  - LM outputs answer as natural language
  - **Zero-shot** setting
- Beneficial over fine-tuning when we don't have much training data
  - Access the knowledge already stored in the LM



Liu, P., Yuan, W., et al. (2021). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv preprint.

Outline

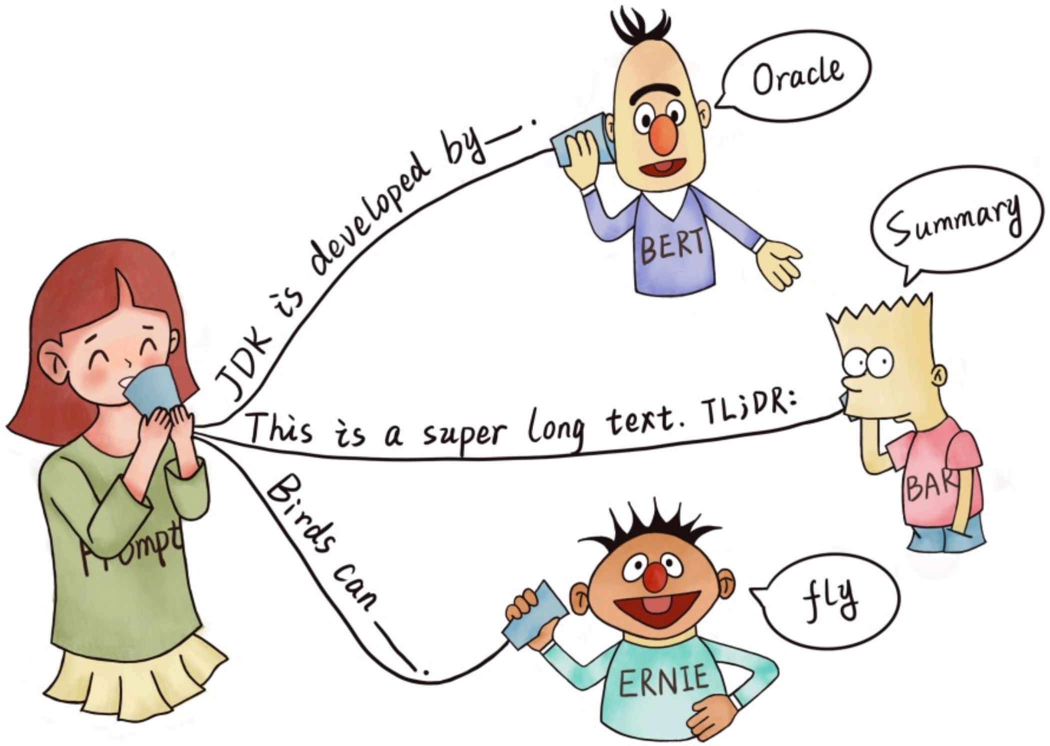
- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Outline

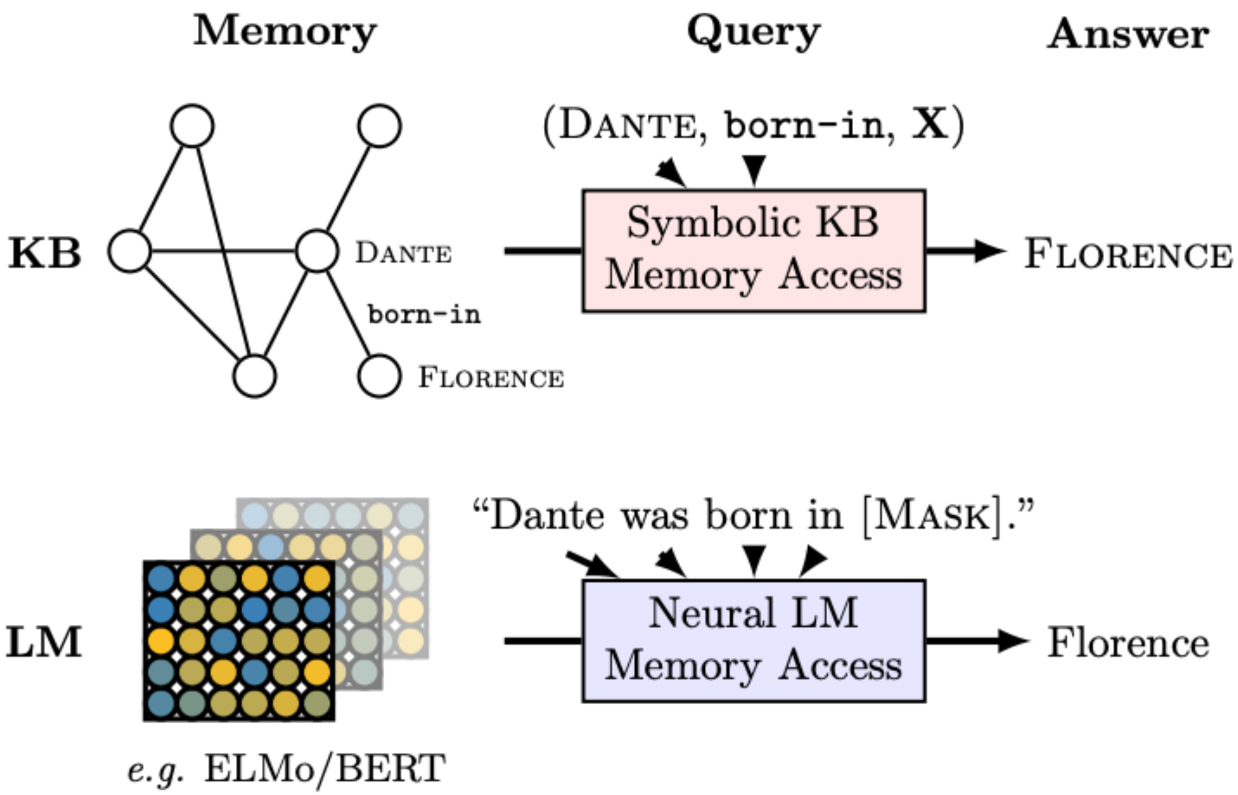
- Extracting knowledge with prompts
  - **Relational prompts**
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Relational Prompts

- Can LMs be used like knowledge bases?
- *Approach*: prompt the LM with an incomplete relation, generate the rest of it
- Advantages:
  - No schema engineering
  - No human annotation
  - Support any query



Petroni, F., Rocktaeschel, T., et al. (2019). Language Models as Knowledge Bases? EMNLP 2019.

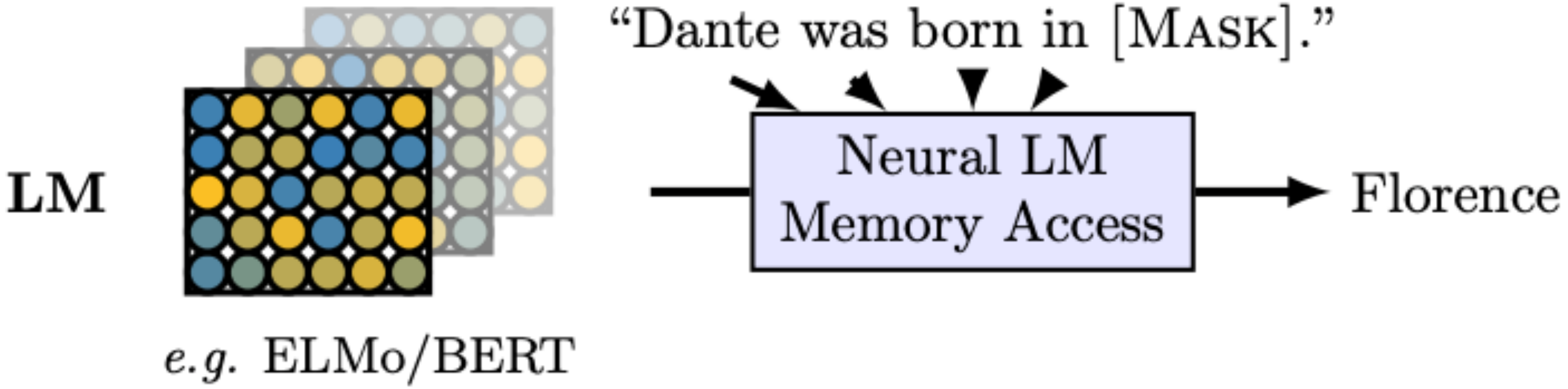
Relational Prompts

- LAMA (Language Model Analysis) dataset compiles this type of *relational knowledge*
- Consists of several pre-compiled knowledge resources:
  - Wikipedia
    - Google-RE (relational facts)
    - T-REx (relational facts)
    - SQuAD (facts from passages)
  - ConceptNet

Petroni, F., Rocktaeschel, T., et al. (2019). Language Models as Knowledge Bases? EMNLP 2019.

Relational Prompts

- Automatically convert relational data into prompts using templates
  - For simplicity, only consider single-token targets from the data, e.g., "Florence"
  - LM can just rank all tokens in vocabulary to fill in the blank



Petroni, F., Rocktaeschel, T., et al. (2019). Language Models as Knowledge Bases? EMNLP 2019.

Corpus	Relation	Statistics		Baselines		KB		LM				Prompting BERT	
		#Facts	#Rel	Freq	DrQA	RE_n	RE_o	Fs	Txl	Eb	E5B	Bb	Bl
Google-RE	`birth-place`	2937	1	4.6	-	3.5	13.8	4.4	2.7	5.5	7.5	14.9	**16.1**
	`birth-date`	1825	1	1.9	-	0.0	**1.9**	0.3	1.1	0.1	0.1	1.5	1.4
	`death-place`	765	1	6.8	-	0.1	7.2	3.0	0.9	0.3	1.3	13.1	**14.0**
	Total	5527	3	4.4	-	1.2	7.6	2.6	1.6	2.0	3.0	9.8	**10.5**
T-REx	1-1	937	2	1.78	-	0.6	10.0	17.0	36.5	10.1	13.1	68.0	**74.5**
	N-1	20006	23	23.85	-	5.4	**33.8**	6.1	18.0	3.6	6.5	32.4	34.2
	N-*M*	13096	16	21.95	-	7.7	**36.7**	12.0	16.5	5.7	7.4	24.7	24.3
	Total	34039	41	22.03	-	6.1	**33.8**	8.9	18.3	4.7	7.1	31.1	32.3
ConceptNet	Total	11458	16	4.8	-	-	-	3.6	5.7	6.1	6.2	15.6	**19.2**
SQuAD	Total	305	-	-	**37.5**	-	-	3.6	3.9	1.6	4.3	14.1	17.4

Table 2: Mean precision at one (P@1) for a frequency baseline (Freq), DrQA, a relation extraction with naïve entity linking (RE_n), oracle entity linking (RE_o), fairseq-fconv (Fs), Transformer-XL large (Txl), ELMo original (Eb), ELMo 5.5B (E5B), BERT-base (Bb) and BERT-large (Bl) across the set of evaluation corpora.

Petroni, F., Rocktaeschel, T., et al. (2019). Language Models as Knowledge Bases? EMNLP 2019.

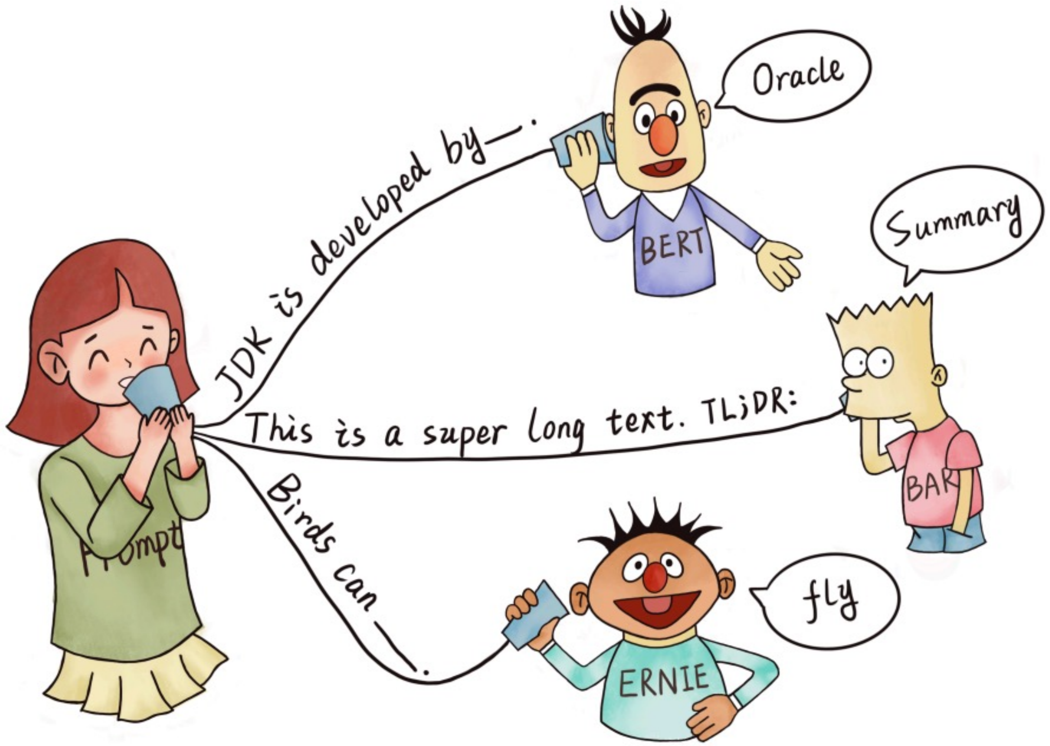
Takeaways

- Using prompts to sample relational knowledge from large LMs works to some degree
  - Fairly competitive with baselines
- While BERT performs best, still much room for improvement in zero-shot setting
  - Maybe we're not ready to let go of fine-tuning…

Petroni, F., Rocktaeschel, T., et al. (2019). Language Models as Knowledge Bases? EMNLP 2019.

Outline

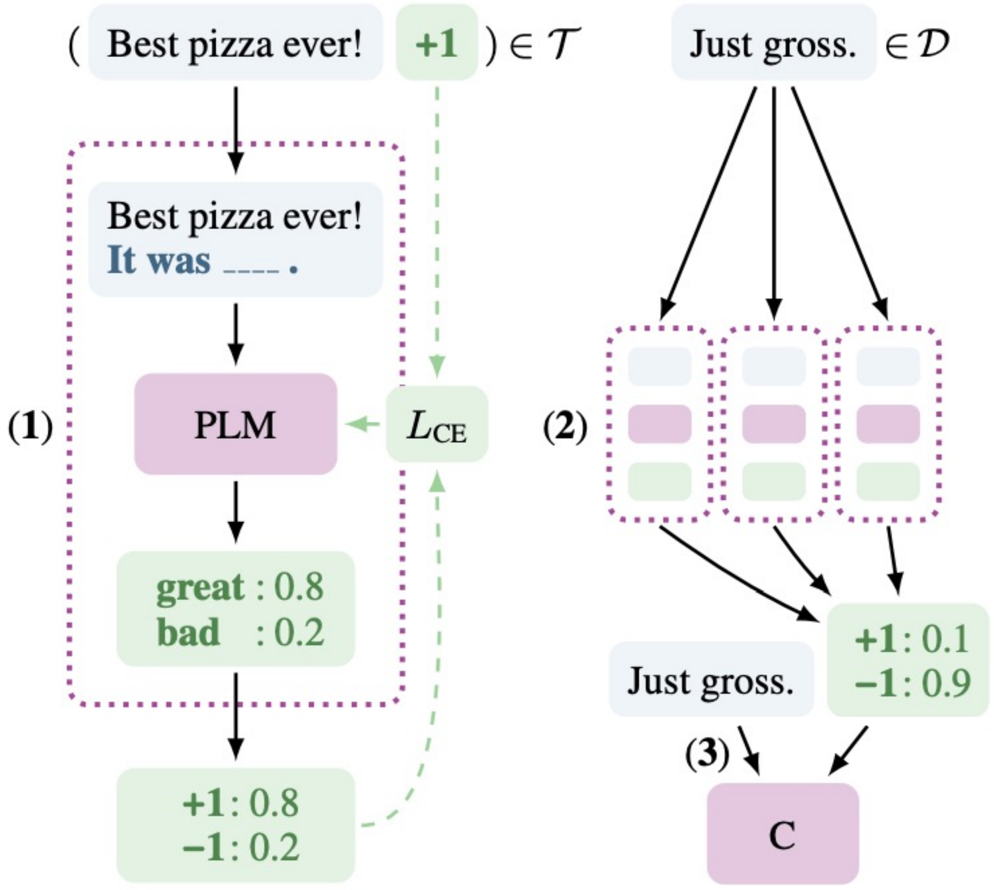
- Extracting knowledge with prompts
  - Relational prompts
  - **Prompts to improve fine-tuning**
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Prompts to Improve Fine-Tuning

- Fine-tuning requires a large training dataset
  - Difficult to learn from small dataset
- Improve learning from small dataset with **pattern-exploiting training (PET)**
- *Approach*:
  1. Define several fill-in-the-blank templates (**patterns**) to use as prompts
     - Fine-tune separate LMs to generate supporting knowledge when prompted with each pattern
  2. Use ensemble of all patterns to generate soft labels for unlabeled data
  3. Fine-tune another LM on labeled data and soft-labeled data



Schick, T., and Schütze, H. (2020). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. EACL 2020.

Line	Examples	Method	Yelp	AG's	Yahoo	MNLI (m/mm)
1	$\|\mathcal{T}\| = 0$	unsupervised (avg)	33.8 ±9.6	69.5 ±7.2	44.0 ±9.1	39.1 ±4.3 / 39.8 ±5.1
2		unsupervised (max)	40.8 ±0.0	79.4 ±0.0	56.4 ±0.0	43.8 ±0.0 / 45.0 ±0.0
3		iPET	**56.7** ±0.2	**87.5** ±0.1	**70.7** ±0.1	**53.6** ±0.1 / **54.2** ±0.1
4	$\|\mathcal{T}\| = 10$	supervised	21.1 ±1.6	25.0 ±0.1	10.1 ±0.1	34.2 ±2.1 / 34.1 ±2.0
5		PET	52.9 ±0.1	87.5 ±0.0	63.8 ±0.2	41.8 ±0.1 / 41.5 ±0.2
6		iPET	**57.6** ±0.0	**89.3** ±0.1	**70.7** ±0.1	**43.2** ±0.0 / **45.7** ±0.1
7	$\|\mathcal{T}\| = 50$	supervised	44.8 ±2.7	82.1 ±2.5	52.5 ±3.1	45.6 ±1.8 / 47.6 ±2.4
8		PET	60.0 ±0.1	86.3 ±0.0	66.2 ±0.1	63.9 ±0.0 / 64.2 ±0.0
9		iPET	**60.7** ±0.1	**88.4** ±0.1	**69.7** ±0.0	**67.4** ±0.3 / **68.3** ±0.3
10	$\|\mathcal{T}\| = 100$	supervised	53.0 ±3.1	86.0 ±0.7	62.9 ±0.9	47.9 ±2.8 / 51.2 ±2.6
11		PET	61.9 ±0.0	88.3 ±0.1	69.2 ±0.0	74.7 ±0.3 / 75.9 ±0.4
12		iPET	**62.9** ±0.0	**89.6** ±0.1	**71.2** ±0.1	**78.4** ±0.7 / **78.6** ±0.5
13	$\|\mathcal{T}\| = 1000$	supervised	63.0 ±0.5	**86.9** ±0.4	70.5 ±0.3	73.1 ±0.2 / 74.8 ±0.3
14		PET	**64.8** ±0.1	**86.9** ±0.2	**72.7** ±0.0	**85.3** ±0.2 / **85.5** ±0.4

Table 1: Average accuracy and standard deviation for RoBERTa (large) on Yelp, AG's News, Yahoo and MNLI (m:matched/mm:mismatched) for five training set sizes $|\mathcal{T}|$.

Schick, T., and Schütze, H. (2020). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. EACL 2020.

Takeaways

- If we have only a small amount of training data, we can use prompting to augment the dataset and enhance fine-tuning
  - Outperform supervised (fine-tuning) and unsupervised (zero-shot) approaches
- Improvement is largest for smaller training dataset sizes

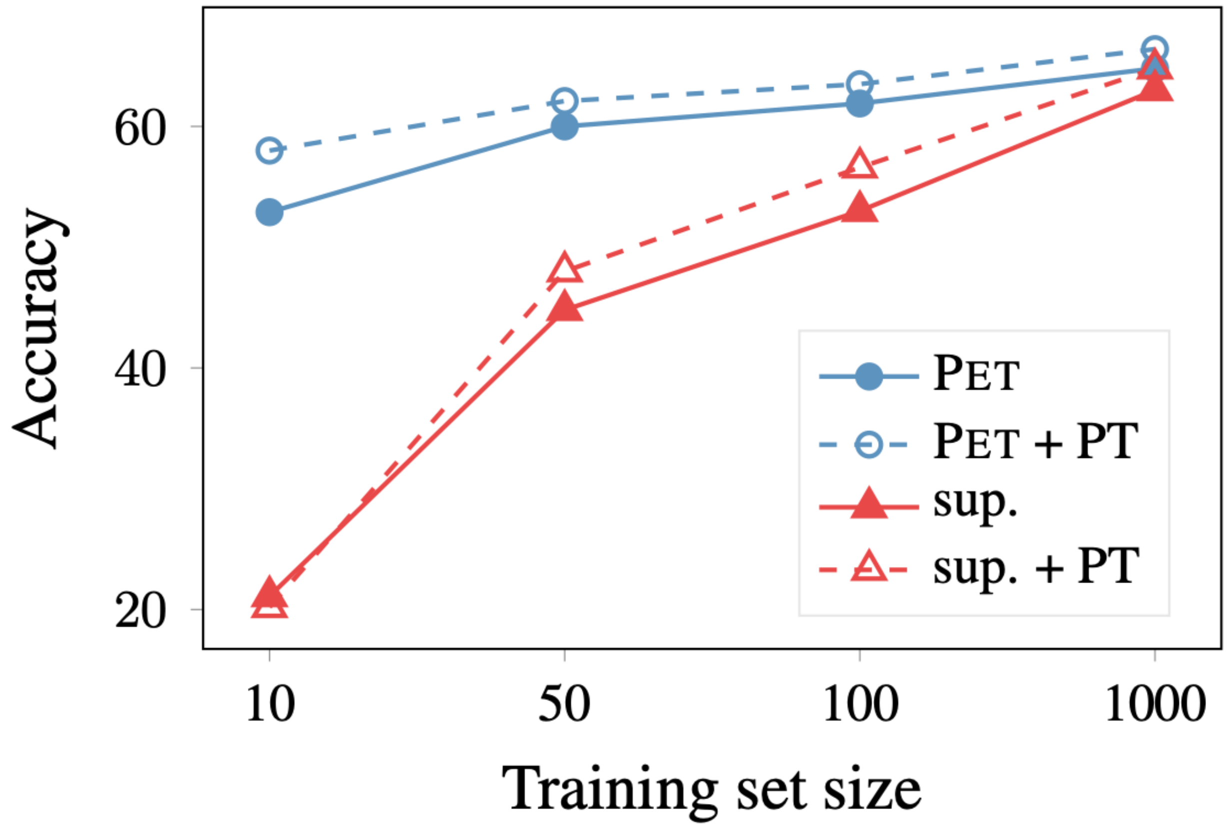
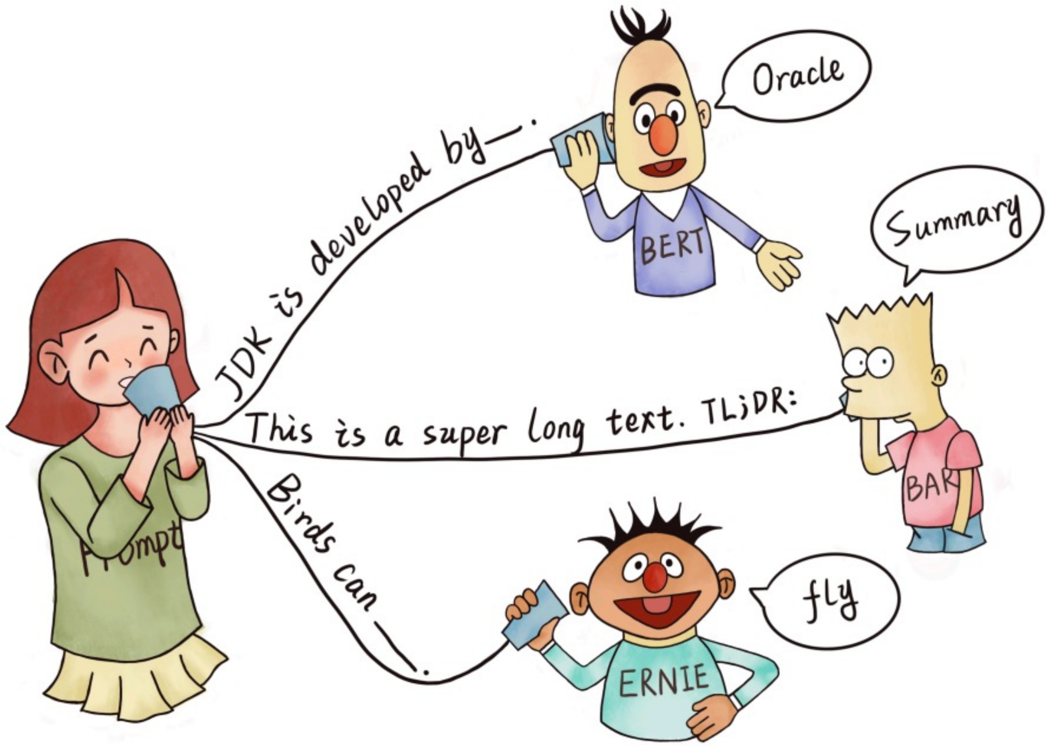


Figure 5: Accuracy of supervised learning (sup.) and PET both with and without pretraining (PT) on Yelp

Schick, T., and Schütze, H. (2020). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. EACL 2020.

Outline

- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - **Prompts to improve zero-shot inference**
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



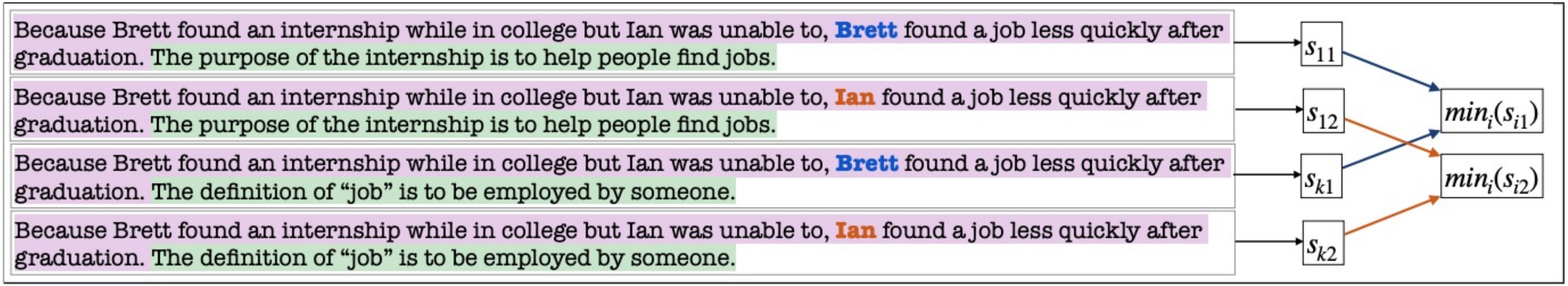
(from Pre-train, Prompt, and Predict Survey Paper)

Prompting to Improve Zero-Shot Inference

- *Recall*: zero-shot inference is hard
  - Can we prompt LM for additional knowledge to support prediction?
- *Approach*: Define several templates we can use to gather clarifying knowledge for a language task
  - Example: *Because Brett found an internship while in college but Ian was unable to, **he** found a job less quickly after graduation.*
    - ***he*** = **Brett** or **Ian**?
  - Ask: What's the purpose of an *internship*? What is a *job*?
    - LM: The purpose of the *internship* is to help people find jobs.
    - LM: The definition of *job* is to be employed by someone.

Shwarz, V., West, P., et al. (2020). Unsupervised Commonsense Question Answering with Self-Talk. EMNLP 2020.

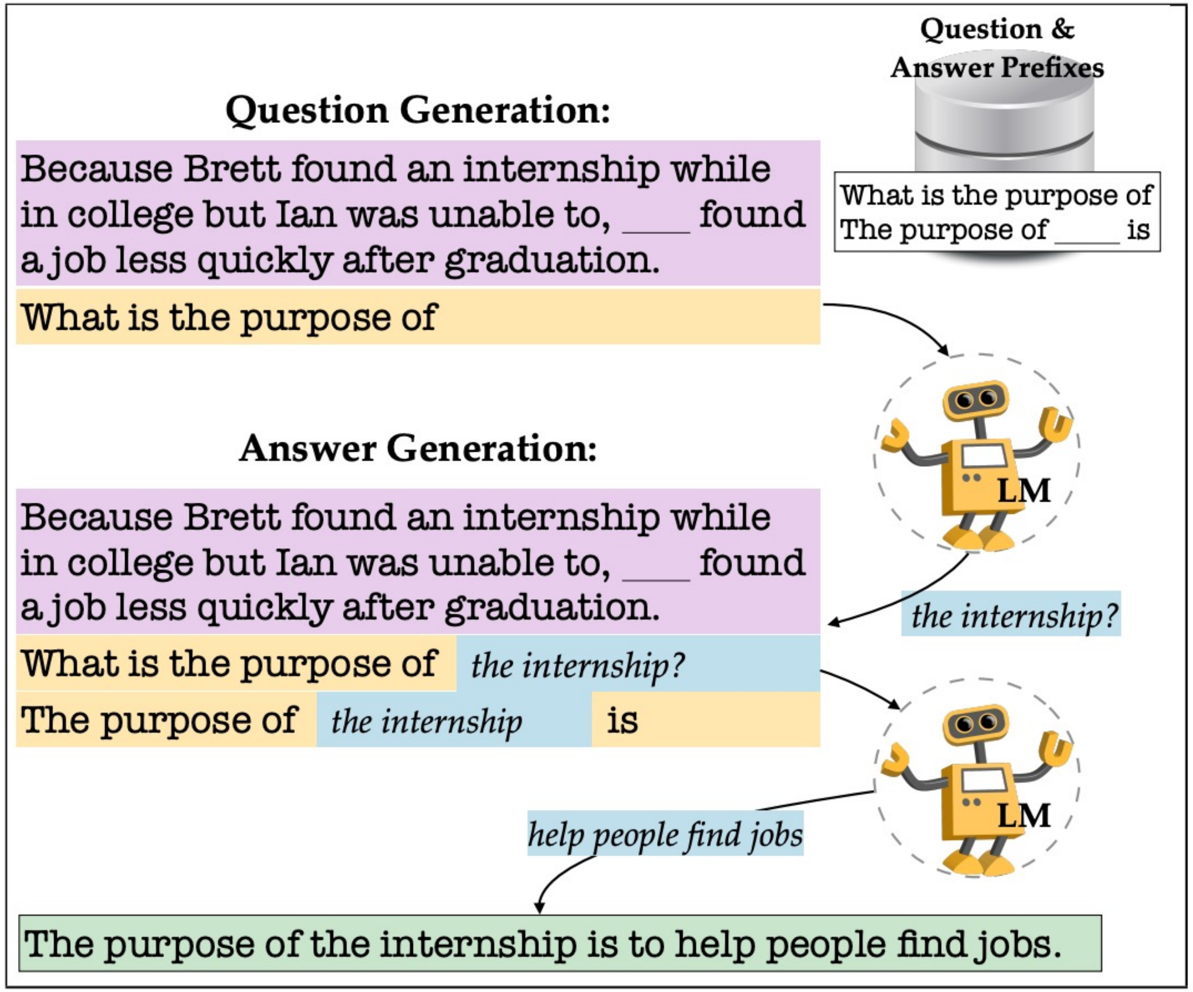
Prompting to Improve Zero-Shot Inference



Shwarz, V., West, P., et al. (2020). Unsupervised Commonsense Question Answering with Self-Talk. EMNLP 2020.

Prompting to Improve Zero-Shot Inference

- In practice, we can also prompt the LM for the concept that needs clarification
- "Self-talk"



Shwarz, V., West, P., et al. (2020). Unsupervised Commonsense Question Answering with Self-Talk. EMNLP 2020.

Prompting to Improve Zero-Shot Inference

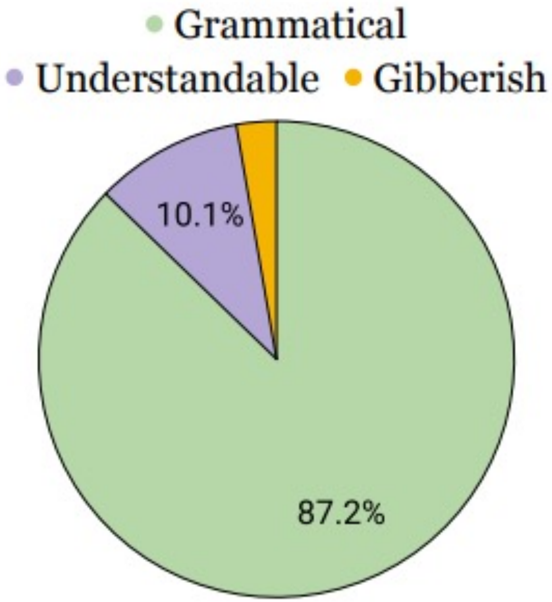
	COMeT	ConceptNet	Google Ngrams	GPT	Distil-GPT2	GPT2	GPT2-M	GPT2-L	GPT2-XL	XLNet	XLNet-L
COPA	10.25	6.87	7.50	7.25	5.37	7.12	7.37	4.37	7.75	6.87	7.37
CSQA	0.39	-3.23	-0.30	-4.04	-3.79	-3.58	-3.09	-3.26	-3.65	-3.91	-3.55
MC-TACO	1.90	3.35	3.53	2.36	2.59	3.15	2.56	3.06	2.92	1.84	1.75
Social IQa	2.74	1.21	1.49	1.71	1.87	1.66	1.75	1.95	2.24	1.74	1.79
PIQA	3.77	4.07	4.36	4.01	3.61	3.80	3.89	3.88	3.96	3.82	4.10
WinoGrande	0.01	-0.01	-0.11	0.13	-0.17	-0.03	-0.04	0.04	0.08	-0.10	-0.25

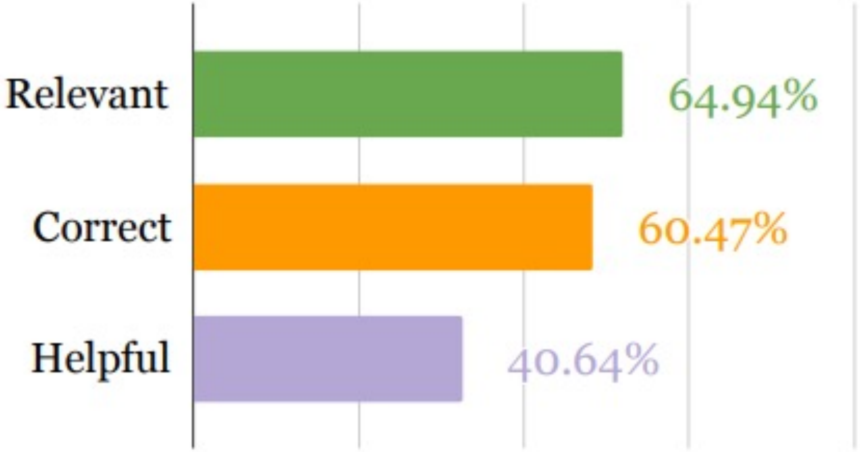
Table 1: Relative improvement upon the zero-shot baseline in terms of development accuracy, for each knowledge source averaged across LMs for each dataset.

Shwarz, V., West, P., et al. (2020). Unsupervised Commonsense Question Answering with Self-Talk. EMNLP 2020.

Takeaways

- Prompting LM for clarification ("self-talking") on language tasks improves zero-shot task performance!
- Paper also includes excellent analysis on the quality and helpfulness of generated clarifications

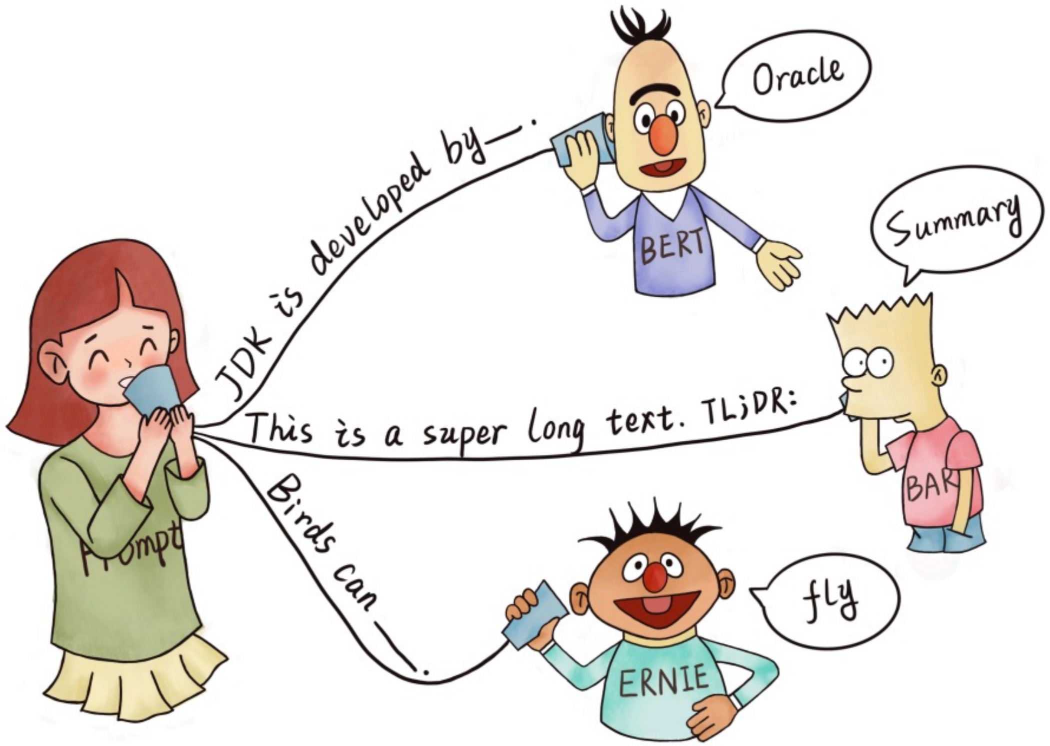




Shwarz, V., West, P., et al. (2020). Unsupervised Commonsense Question Answering with Self-Talk. EMNLP 2020.

Outline

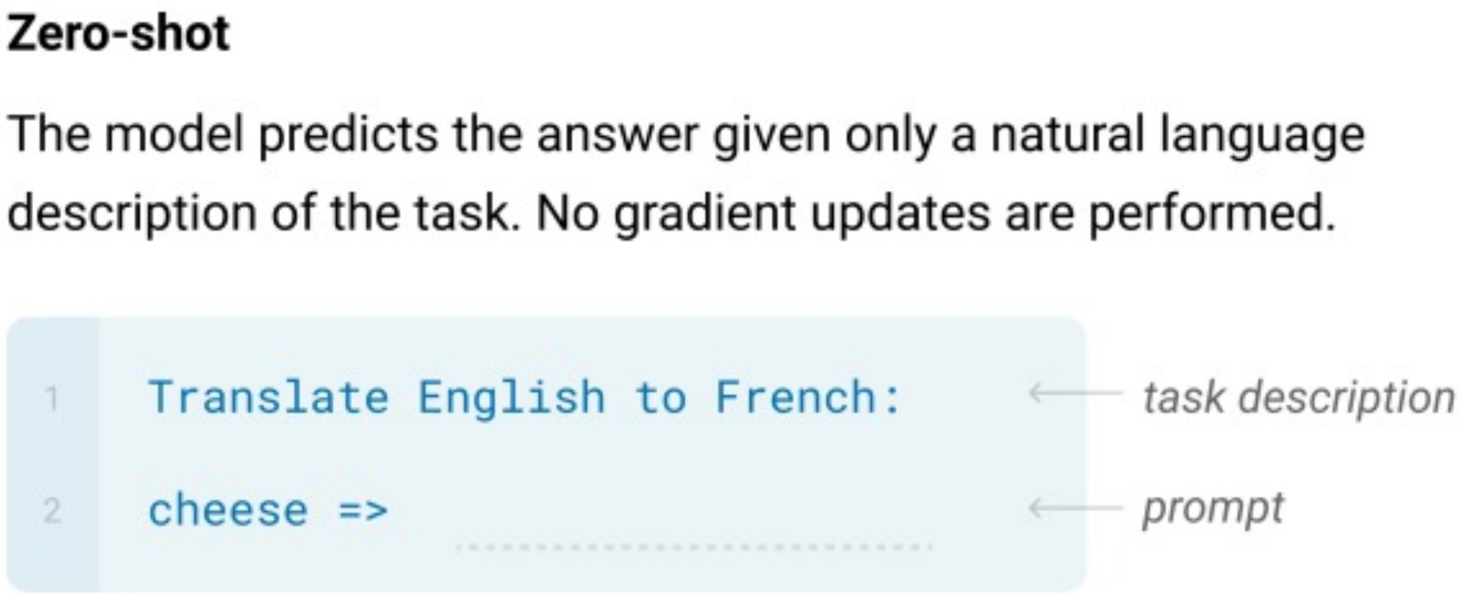
- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - **Few-shot inference with LMs**
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Prompting Massive LMs

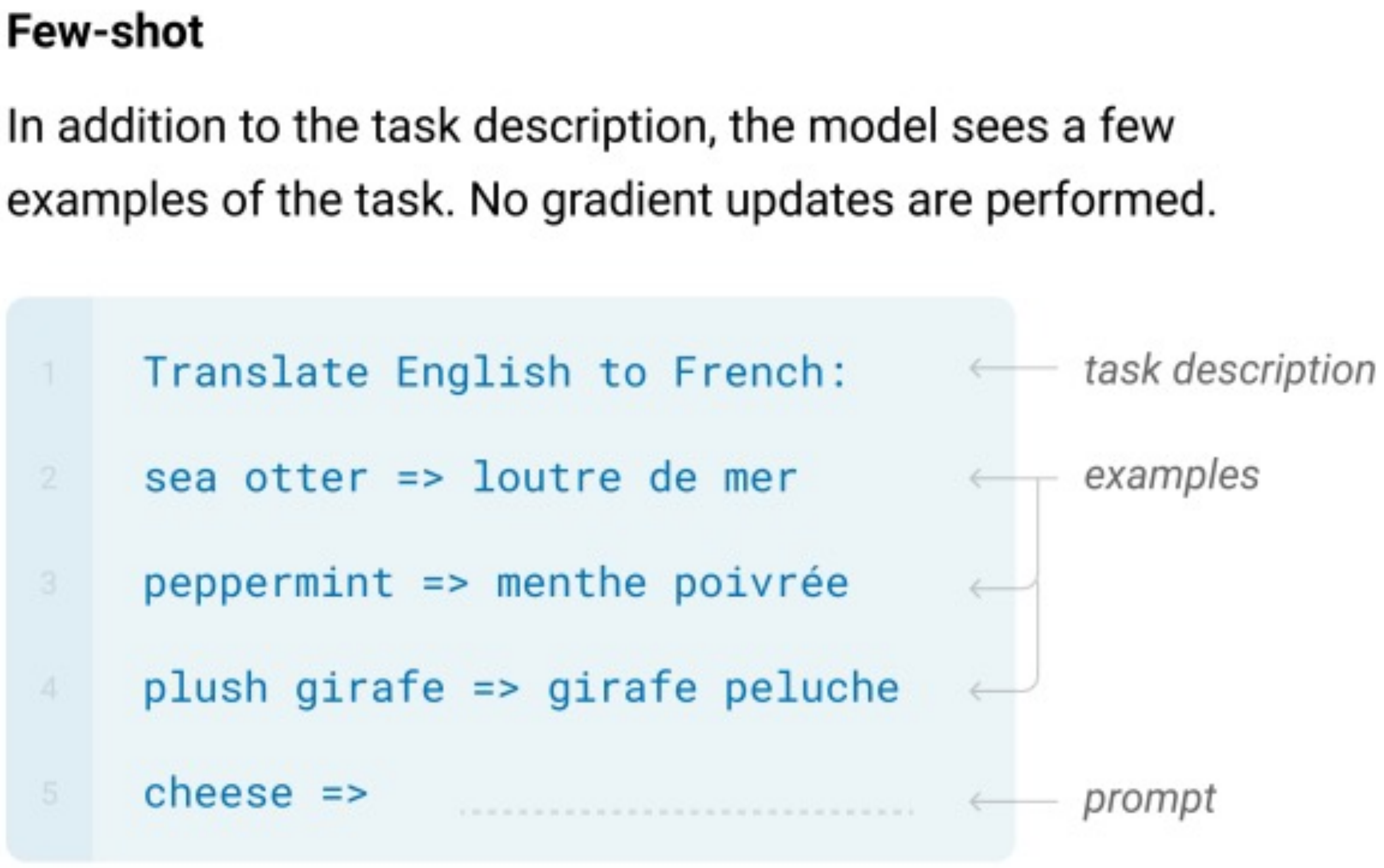
- As LMs evolve and grow, they become more capable to solve language tasks in a zero-shot setting
    - **Prompt engineering** plays a big role
    - What if we prompt the LM with a few examples of the task first?
    - **Few-shot** setting



Brown, T.B., Mann, B., et al. (2020). Language Models are Few-Shot Learners. arXiv pre-print.

Prompting Massive LMs

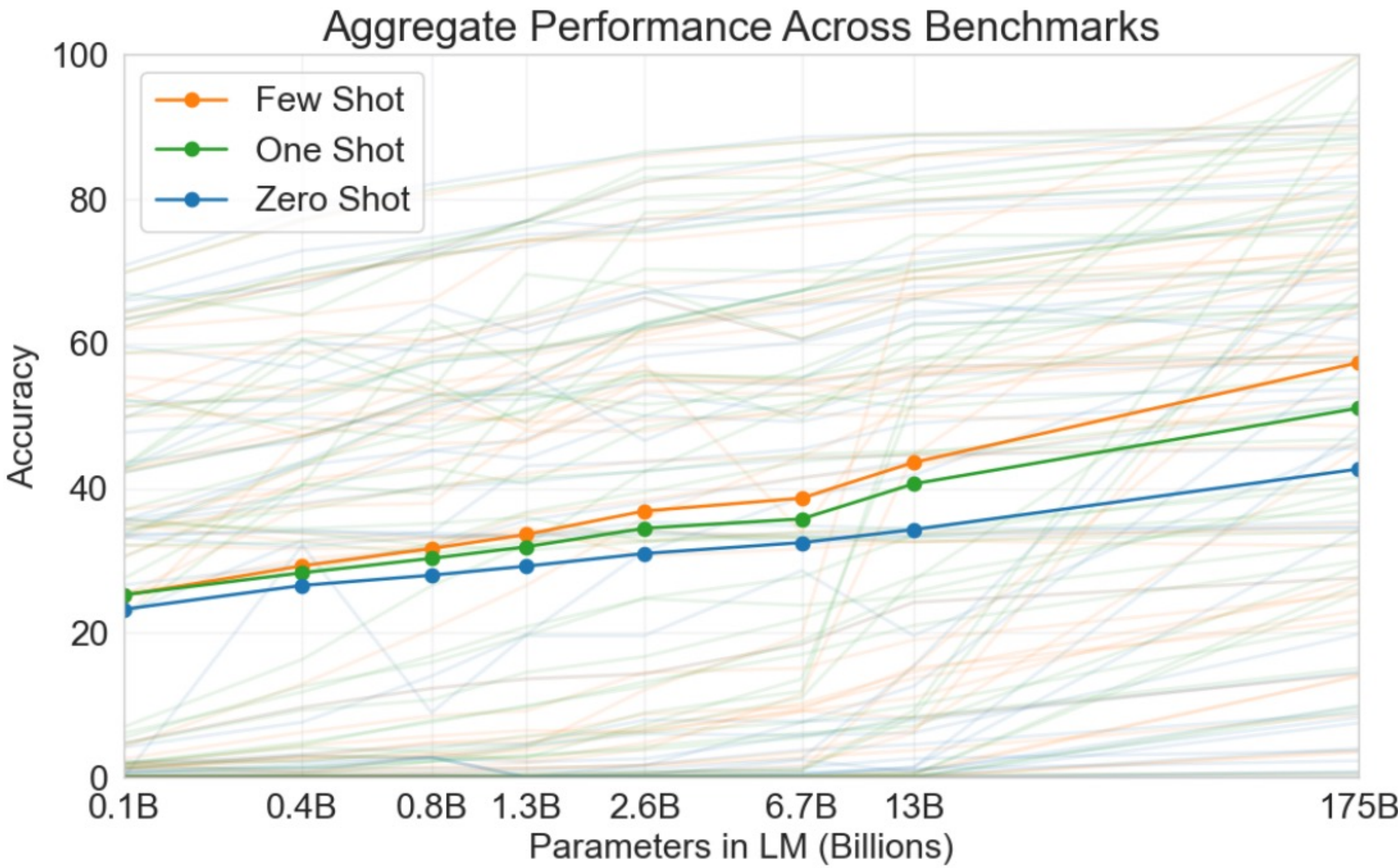
- As LMs evolve and grow, they become more capable to solve language tasks in a zero-shot setting
  - **Prompt engineering** plays a big role
  - What if we prompt the LM with a few examples of the task first?
  - **Few-shot** setting



[Brown, T.B., Mann, B., et al. (2020). Language Models are Few-Shot Learners. arXiv pre-print.]

GPT-3 Zero-Shot and Few-Shot Inference

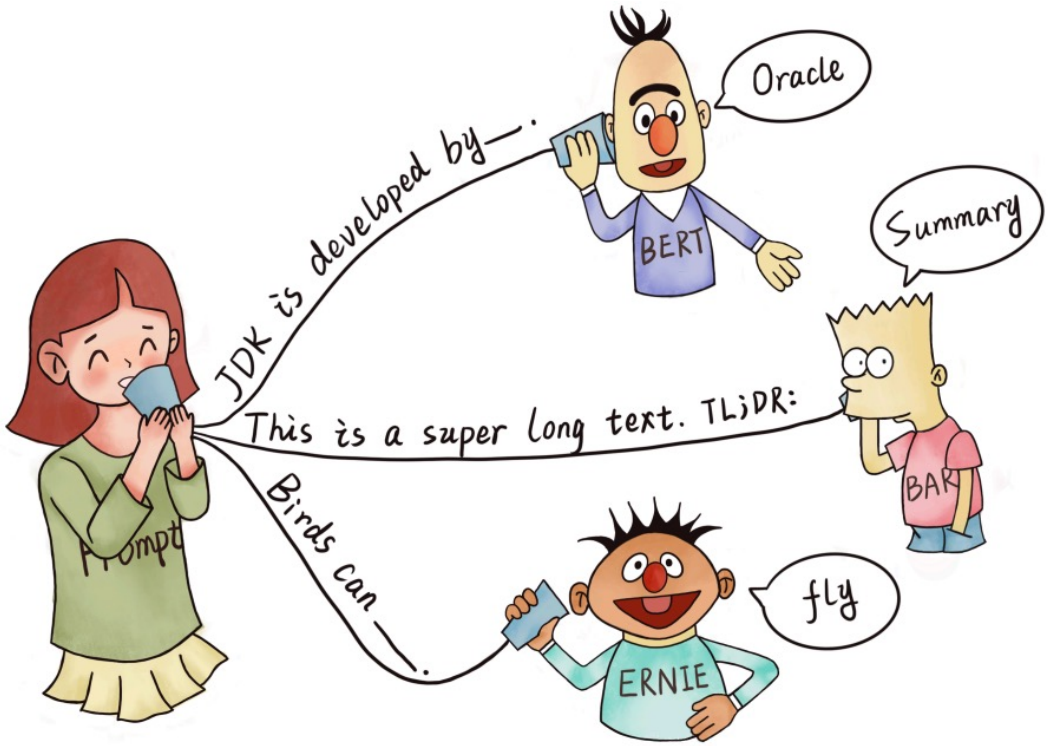
- GPT-3 succeeds in zero-shot and few-shot settings across several language tasks!
  - Zero-shot and few-shot performance increase as model complexity increases



[Brown, T.B., Mann, B., et al. (2020). Language Models are Few-Shot Learners. arXiv pre-print.]

Outline

- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - **Reasoning with LMs**
- Learning better prompts
  - Learning to prompt
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Step-by-Step Reasoning for Massive LMs

- Reasoning tasks are especially challenging
  - May require several steps of internal monologue to arrive at the answer
- Even in a few-shot setting, **prompt engineering** may play a big role
- How can we best solicit reasoning from the LM?

Chain of Thought Prompting

Standard Prompting

Input

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: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain of Thought Prompting

Input

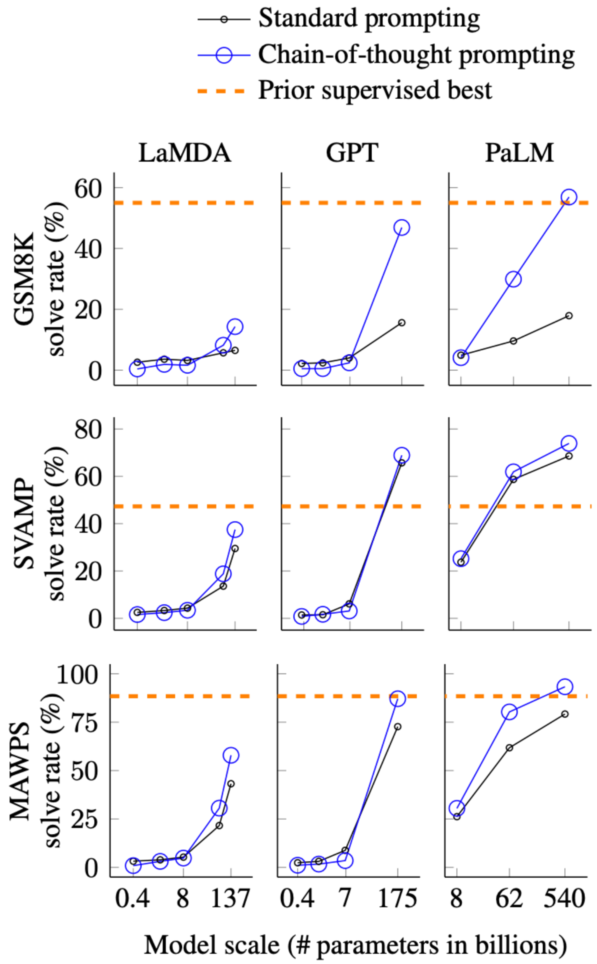
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

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: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

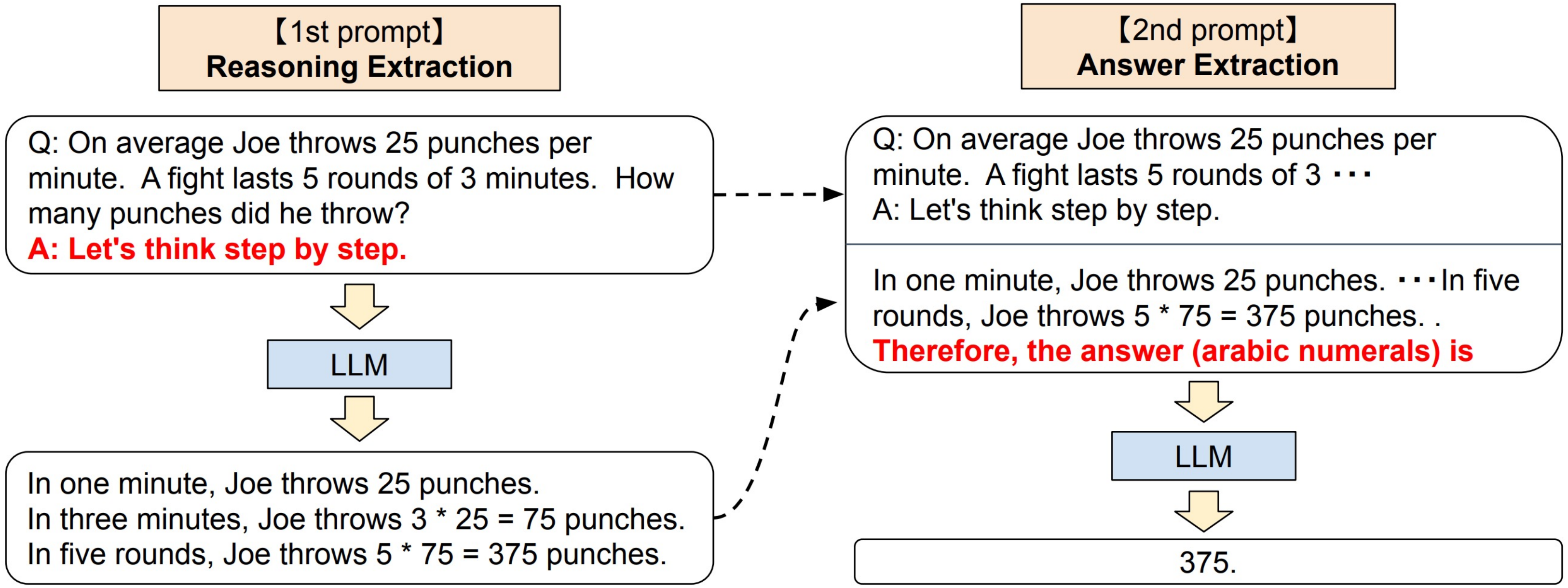
Model Output

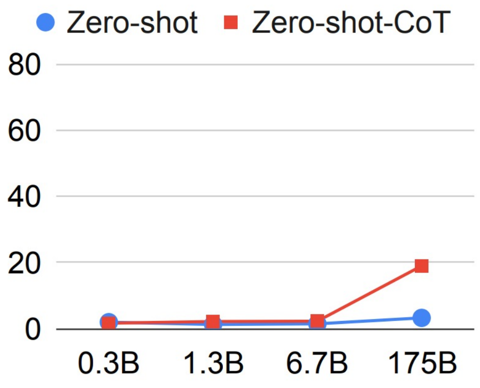
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✔



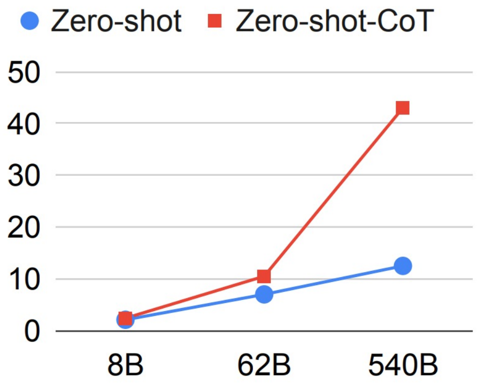
Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022.

“Let’s Think Step by Step”





(a) MultiArith on Original GPT-3



(c) GMS8K on PaLM

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. NeurIPS 2022.

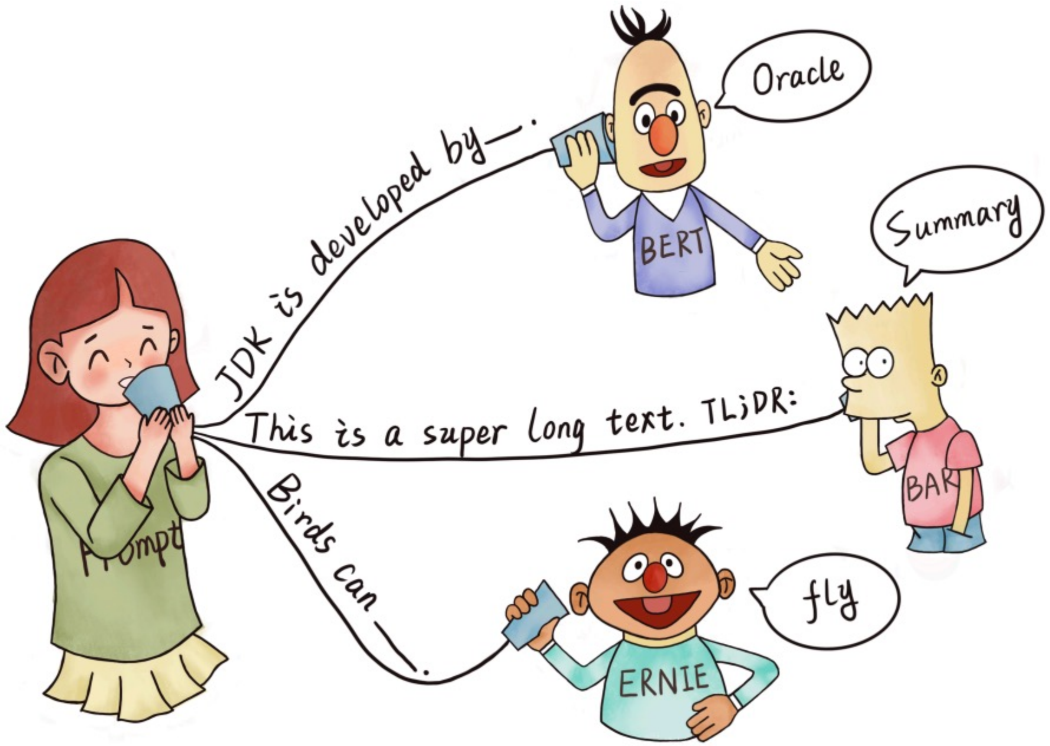
Takeaways

- Massive LMs can successfully perform language understanding tasks without fine-tuning on thousands of examples
  - Just prompt with a few examples
  - Can even elicit step-by-step reasoning with chain of thought
  - Compete with supervised SOTA approaches
- NLP is now moving away from fine-tuning, and toward prompting!

Brown, T.B., Mann, B., et al. (2020). Language Models are Few-Shot Learners. arXiv pre-print.

Outline

- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- **Learning better prompts**
  - Learning to prompt
  - Learning soft prompts



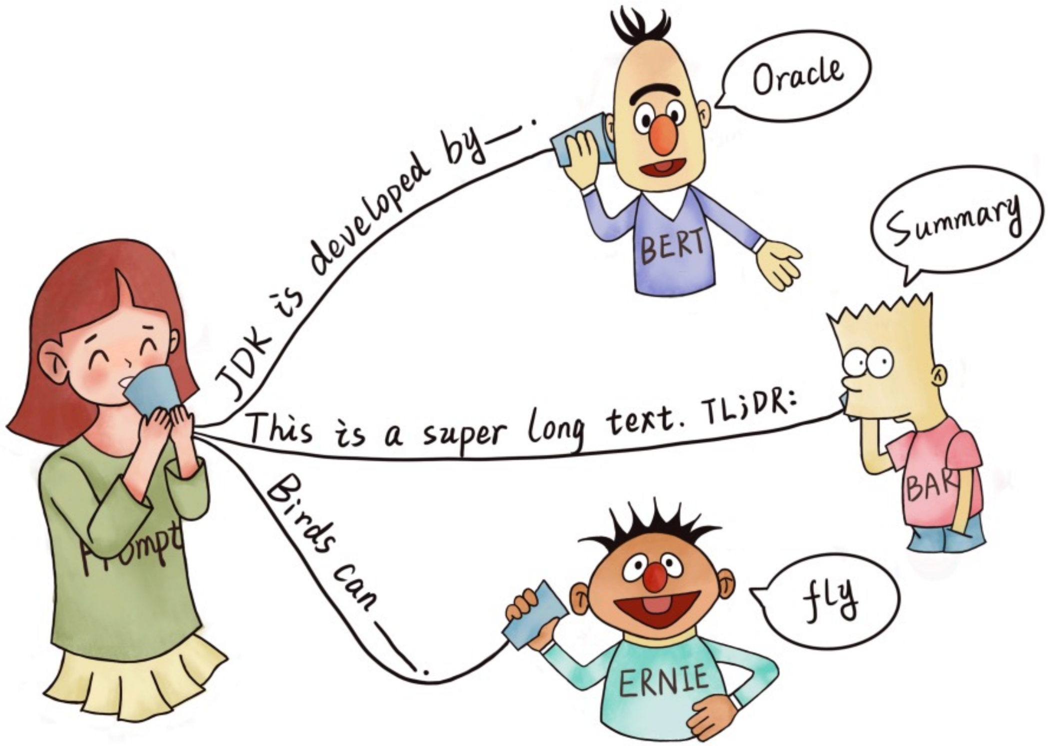
(from Pre-train, Prompt, and Predict Survey Paper)

Learning Better Prompts

- Prompts so far have been manually engineered based on various templates or pre-compiled benchmark data…
  - Can we do better than this? How can we find an optimal prompt?
- Approaches:
  - Learning to generate LM prompt text
  - Learning to generate LM prompt vectors

Outline

- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - **Learning to prompt**
  - Learning soft prompts



(from Pre-train, Prompt, and Predict Survey Paper)

Learning New Prompts

- How can we *learn* the optimal words for a prompt?
- *Approach*: given some manually defined prompt, select several learned **trigger tokens** with a gradient-based search
  - Improve the likelihood of the LM producing the correct answer
  - Learn which tokens are best suited to be associated with class labels

Shin, T., Razeghi, Y., et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP 2020.

Learning New Prompts

A real joy . [T] ______

"Positive"

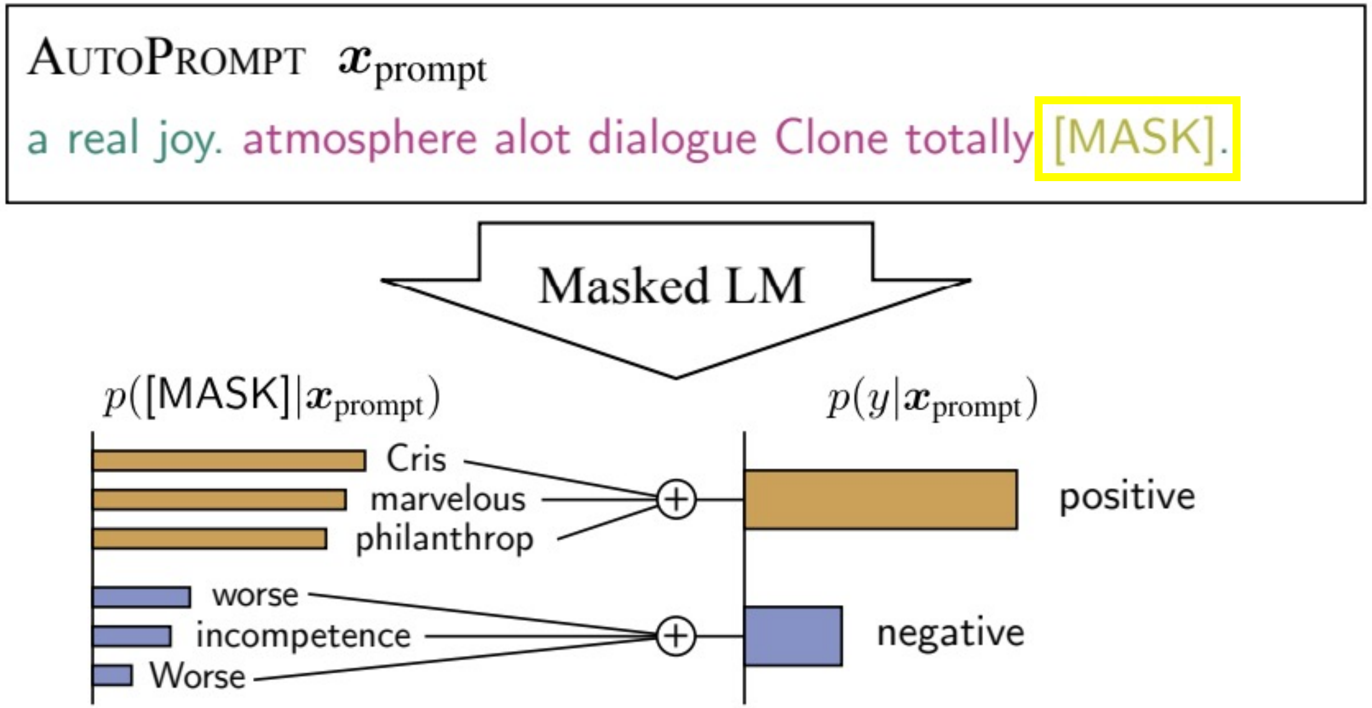
$$\mathcal{V}_{\text{cand}} = \underset{w \in \mathcal{V}}{\text{top-}k} \left[\boldsymbol{w}_{\text{in}}^{T} \nabla \log p(y | \boldsymbol{x}_{\text{prompt}}) \right]$$

A real joy . atmosphere alot dialogue Clone totally ______

Shin, T., Razeghi, Y., et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP 2020.

Learning Mapping from Tokens to Classes

- Given a prompt, an LM will rank all tokens in the vocabulary by likelihood to appear after the prompt
  - The most likely tokens are not necessary the desired token relating to a class, e.g., “positive”
- Can we learn a better mapping from generated tokens to predicted classes?



Shin, T., Razeghi, Y., et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP 2020.

Takeaways

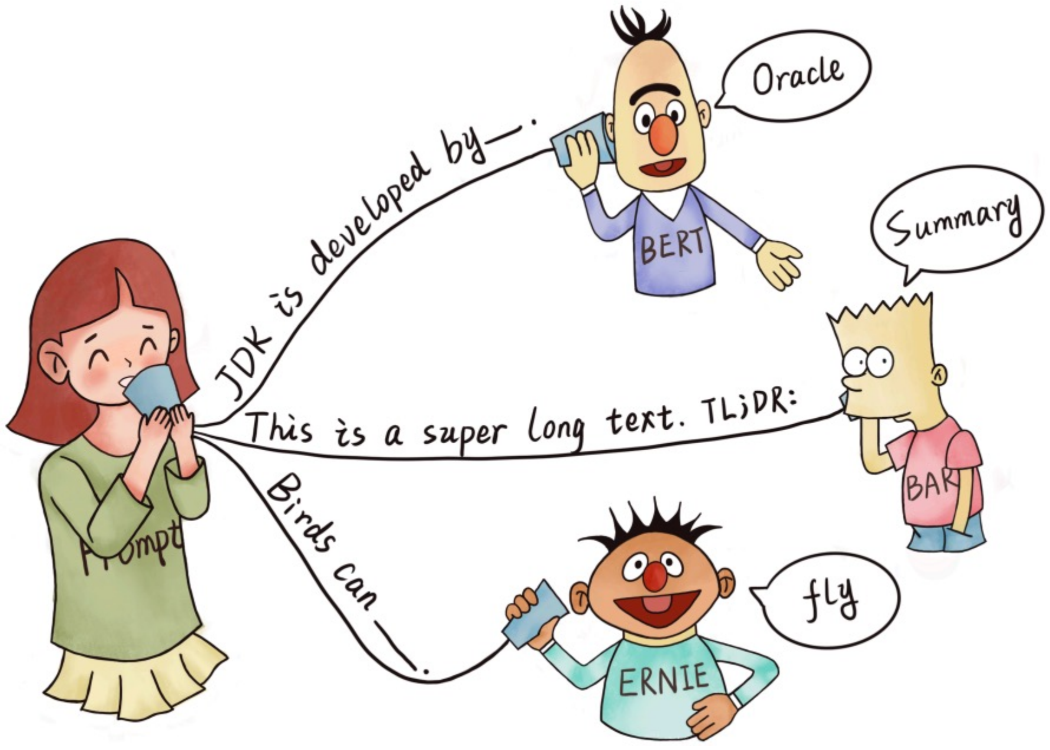
- AutoPrompt drastically improves performance over manually defined prompts!
- Performance comes close to supervised approaches even with BERT and RoBERTa
  - Much smaller than GPT-3 😎

Model	Dev	Test
BERT (finetuned)	-	93.5†
RoBERTa (finetuned)	-	96.7†
BERT (manual)	63.2	63.2
BERT (AUTOPROMPT)	80.9	82.3
RoBERTa (manual)	85.3	85.2
RoBERTa (AUTOPROMPT)	91.2	91.4

Table 1: **Sentiment Analysis** performance on the SST-2 test set of supervised classifiers (top) and fill-in-the-blank MLMs (bottom). Scores marked with † are from the GLUE leaderboard: http://gluebenchmark.com/leaderboard.

Outline

- Extracting knowledge with prompts
  - Relational prompts
  - Prompts to improve fine-tuning
  - Prompts to improve zero-shot inference
- Directly solving tasks with prompts
  - Few-shot inference with LMs
  - Reasoning with LMs
- Learning better prompts
  - Learning to prompt
  - **Learning soft prompts**



(from Pre-train, Prompt, and Predict Survey Paper)

Learning Soft Prompts

- *Lastly*: Why limit ourselves to human-interpretable tokens?
  - Past prompting works have focused on the tokens in prompts
  - In SOTA LMs, tokens are converted into numerical vector embeddings using several embedding layers before being processed by the transformer
    - Word embedding
    - Position embedding
    - Segment embedding
  - Can we learn a dense query vector, i.e., **soft prompt**, that is most likely to produce the correct answer for a task?
  - **Prompt is no longer a sequence of words – it's a sequence of vectors!**

Qin, G. and Eisner, J. (2021). Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. NAACL 2021 (Best Short Paper).

Learning Soft Prompts

- *Motivation*: Some **hard prompts** will not apply to all cases
  - *Example:*
    - *"____ performed until his death in ____"*
    - Only applicable to male performers!
- Generate an initial soft prompt from the hard prompt's word embeddings:
  - Before: *"____ performed until his death in ____"*
  - After: "____ $v_{performed}$ v_{until} v_{his} v_{death} v_{in} ____"
- Vectors can now be tuned continuously through small perturbations

Qin, G. and Eisner, J. (2021). Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. NAACL 2021 (Best Short Paper).

Learning Soft Prompts

- Consider a set of soft prompts $\mathcal{T}_r$ for some relation type in LAMA
  - Model probability of LM's generated token as a weighted sum of soft prompt outputs, where $p(\boldsymbol{t}|r)$ is a learned weight for the soft prompt $\boldsymbol{t}$:

$$p(y \mid x, r) = \sum_{\mathbf{t} \in \mathcal{T}_r} \underbrace{p(\mathbf{t} \mid r)}_{\text{prompt weight (learned)}} \cdot \underbrace{p_{\mathrm{LM}}(y \mid \mathbf{t}, x)}_{\text{correct token likelihood for this prompt}}$$

  - Optimize model by maximizing the likelihood of correct token being predicted
    - Freeze weights of LM, instead adjust prompt vectors and weights
    - Weights of soft prompts are learned implicitly based on the inputs
    - Instead of learning to complete task with LM, learn how to ask the LM to complete it

Qin, G. and Eisner, J. (2021). Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. NAACL 2021 (Best Short Paper).

Learning Soft Prompts

- Start with pre-made hard prompts (**min.**) or randomly initialize the soft prompts instead (**ran.**)
- Compare BERT-base (**BEb**) and BERT-large (**BEl**) on LAMA
- *Metrics*: P@1, P@10 for correct token, mean reciprocal rank (MRR)

Model	P@1	P@10	MRR
LAMA (BEb)	0.1†	2.6†	1.5†
LAMA (BEl)	0.1†	5.0†	1.9†
Soft (min.,BEb)	11.3(+**11.2**)	36.4(+**33.8**)	19.3(+**17.8**)
Soft (ran.,BEb)	**11.8**(+11.8)	**34.8**(+31.9)	**19.8**(+19.6)
Soft (min.,BEl)	**12.8**(+12.7)	**37.0**(+32.0)	**20.9**(+19.0)
Soft (ran.,BEl)	**14.5**(+14.5)	**38.6**(+34.2)	**22.1**(+21.9)

Table 3: Results on ConceptNet (winner: random init).

Qin, G. and Eisner, J. (2021). Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. NAACL 2021 (Best Short Paper).

Takeaways

- We don't need language-based prompts to extract knowledge out of large LMs!
- We can get away with learning vector prompts that are randomly initialized
  - **No need to write prompts!**
- *Limitation*: loss of interpretability 😬

Qin, G. and Eisner, J. (2021). Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. NAACL 2021 (Best Short Paper).

Summary

1. It’s difficult to extract knowledge from early large LMs, e.g., BERT, using manually-defined prompts
2. Manually-defined prompts can be combined with LM fine-tuning for better performance when training data is small
3. Prompts can be used to gather supporting information to solve language tasks in zero-shot settings
4. More complex language models, e.g., GPT-3, can solve language tasks directly in zero- and few-shot settings
5. Learning prompts for LMs further improves performance, even on zero-shot setting for early large LMs

Prompting LLMs for Task Planning in Robotics

Yinpei Dai
Computer Science and Engineering, University of Michigan
Nov 16th, 2022

Outline

- What is task planning ?
- How to use prompts to do task planning?
- Main challenges

What is task/robotic planning?

https://say-can.github.io/

What is task/robotic planning?

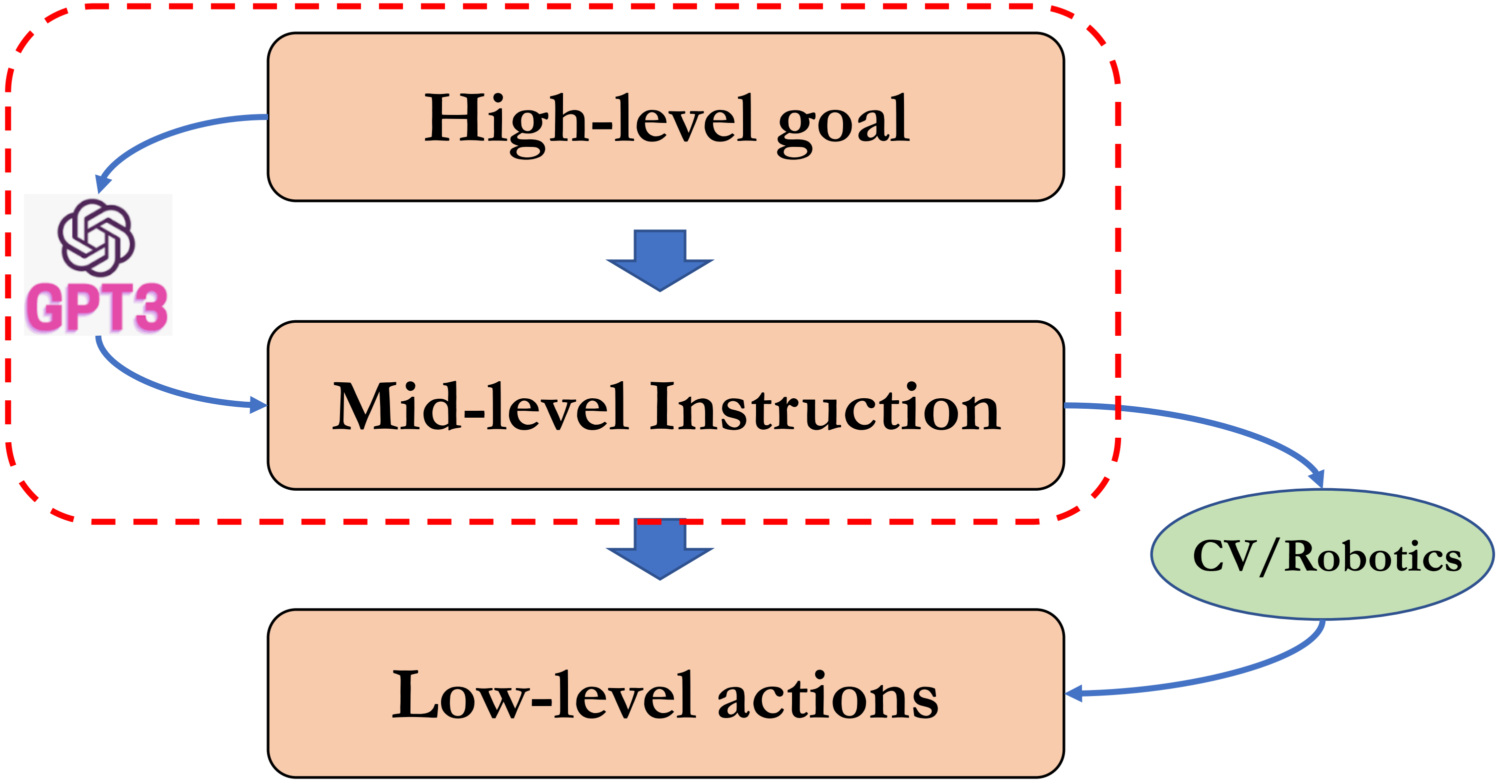
- Task planning: How to plan actions to achieve certain tasks.
- Three levels:
  - High-level goals/tasks/missions.
    - E.g., “I spilled my coke, throw the coke can”
  - Mid-level instructions.
    - E.g., “find a coke can” “go to the trash can” “put down the coke can”
  - Low-level (primitive) actions.
    - E.g., “go forward 5 meters, turn left 30 degrees, go forward 3 inches, ”

I spilled my coke on the table, how would you throw it away and bring me something to help clean?

Robot: I would: 1. find a coke can, 2. pick up the coke can, 3.go to trash can, 4. put down the coke can, 5.__

What is task/robotic planning?

- Task planning: How to plan actions to achieve certain tasks.
- Three levels:



LLMs are few-shot planners

- Method: Using demonstration data as the prompt

(Prefix) Prompt

Robot: Hi there, I'm a robot operating in an office kitchen.
Robot: You can ask me to do various tasks and I'll tell you the sequence of actions I would do to accomplish your task.
Human: How would you hold the snickers?
Robot: 1. pick up the snickers, 2. done.
Human: How would you put the trailmix here?
Robot: 1. put down the trailmix, 2. done.
Human: How would you put a water bottle and oatmeal next to the microwave?
Robot: 1. find a water bottle, 2. pick up the water bottle, 3. go to microwave, 4. put down the water bottle, 5. find an oatmeal, 6. pick up the oatmeal, 7. go to the microwave, 8. put down the oatmeal, 9. Done.

Question

Human: How to scramble eggs?

Answer

Robot: 1. get a pan, 2. put the pan on the stove, 3. turn on the stove, 4. get a bowl, 5. crack eggs into the bowl, 6. whisk the eggs, 7. pour the eggs into the pan, 8. scramble the eggs, 9. put the eggs on a plate, 10. Done.

Main Challenges for LLM Planning

- **Challenge 1:** LLMs are not situated.

[Prefix prompts are omitted..]

Human: Bring me a cup of juice.

Robot: 1. find a cup, 2. pick up the cup, 3. go to the fridge, 4. get juice from the fridge, 5. pour juice into the cup, 6. bring the cup of juice to you, 7. put down the cup of juice, 8. done.

Main Challenges for LLM Planning

- **Challenge 1**: LLMs are not situated.
- Solution: Using CV/Robotic Models to re-rank

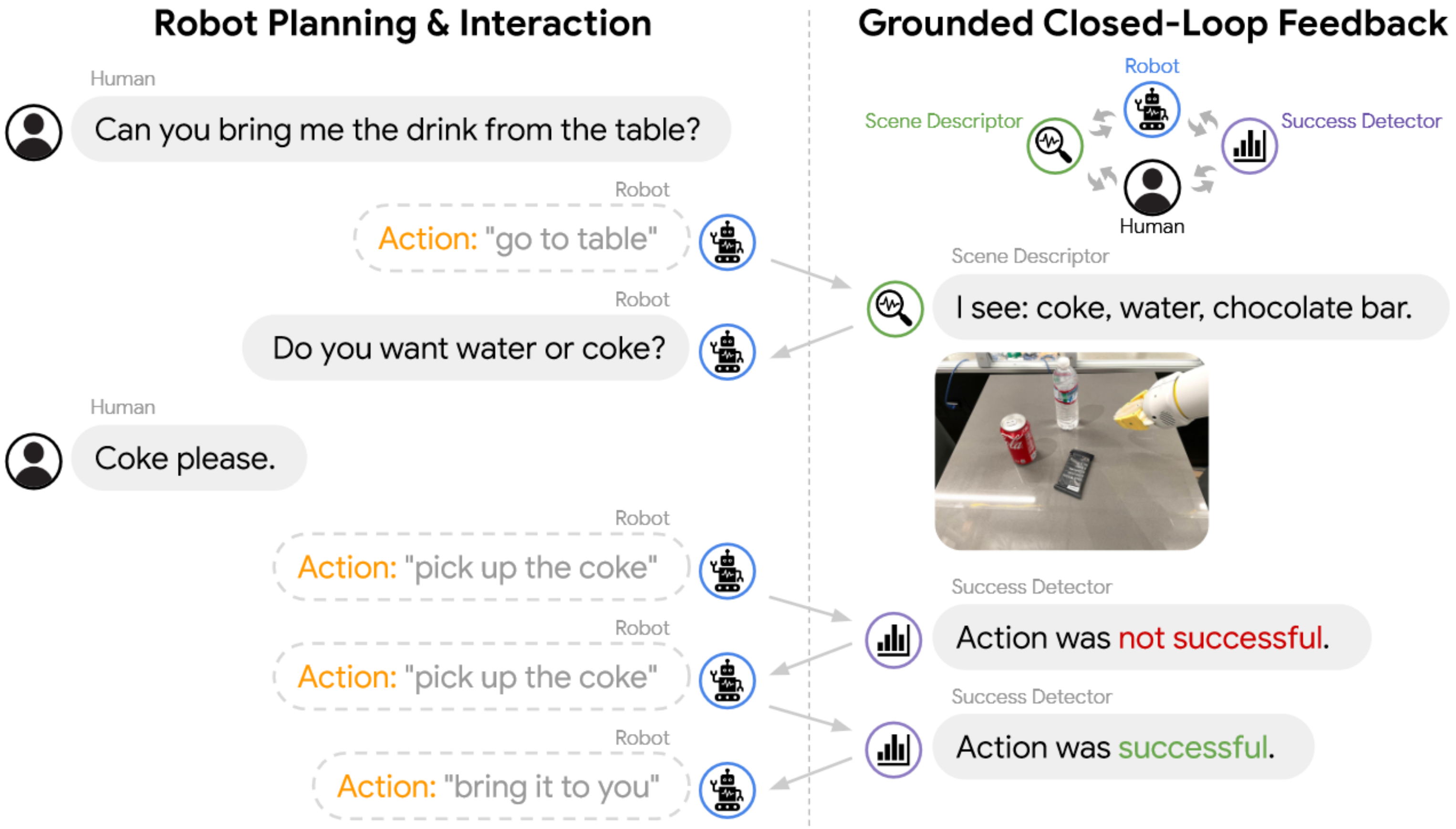
Brohan A, Chebotar Y, Finn C, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances[C]//6th Annual Conference on Robot Learning. 2022.

Main Challenges for LLM Planning

- **Challenge 2:** Exception handling.
  - Action Failure
    - E.g., unsuccessful pick up
  - High-level goal change
    - E.g., “I want to drink coke” “I change my mind, I want to drink tea”
  - Environment change
    - E.g., scene change, water run out, power off
  - Uncertain Case
    - E.g., “There are two apples on the table, which one do you want?”
  - …

Main Challenges for LLM Planning

- **Challenge 2:** Exception handling
- Solution: Add feedbacks to LLMs



Uncertain Case

Action Failure

Huang W, Xia F, Xiao T, et al. Inner monologue: Embodied reasoning through planning with language models[J]. arXiv preprint arXiv:2207.05608, 2022.