In-context learning

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Origins

ChomskyBot

Look On My Words, Ye Mighty, And Despair!

Of course, the appearance of parasitic gaps in domains relatively inaccessible to ordinary extraction appears to correlate rather closely with nondistinctness in the sense of distinctive feature theory. Note that any associated supporting element cannot be arbitrary in irrelevant intervening contexts in selectional rules. With this clarification, a descriptively adequate grammar may remedy and, at the same time, eliminate the levels of acceptability from fairly high (eg (99a)) to virtual gibberish (eg (98d)). Comparing these examples with their parasitic gap counterparts in (96) and (97), we see that most of the methodological work in modern linguistics is not to be considered in determining the ultimate standard that determines the accuracy of any proposed grammar. Thus a subset of English sentences interesting on quite independent grounds is necessary to impose an interpretation on the system of base rules exclusive of the lexicon.

Next paragraph (Use **RELOAD** if the button doesn't work)
What is this all about? **How does it work?**

see also: WikiPedia -- Chomskybot

`http://rubberducky.org/cgi-bin/chomsky.pl`

Early precedents

- In the pre-deep learning era, n-gram LMs were often massive! Brants et al. (2007) use a 300B parameter model trained on 2 trillion tokens.
- decaNLP (McCann et al. 2018): Multi-task training with task instructions as natural language questions.
- Radford et al. (2018): Some tentative prompt-based experiments with GPT.

Beginnings: Radford et al. 2019 (GPT-2)

- "We demonstrate language models can perform down-stream tasks in a zero-shot setting – without any parameter or architecture modification."
- "To induce summarization behavior we add the text `TL;DR:` after the article and generate 100 tokens"
- "We test whether GPT-2 has begun to learn how to translate from one language to another. In order to help it infer that this is the desired task, we condition the language model on a context of example pairs of the format `english sentence = french sentence` and then after a final prompt of `english sentence =` we sample from the model with greedy decoding and use the first generated sentence as the translation."
- "Similar to translation, the context of the language model is seeded with example question answer pairs which helps the model infer the short answer style of the dataset."'
- Also evaluated: text completion, Winograd schemas, reading comprehension.

Cultural moment: Brown et al. 2020 (GPT-3)

"Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. 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. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora."

Core concepts

Terminology

- In-context learning: A frozen LM performs a task only by conditioning on the prompt text.
- Few-shot in-context learning: (1) The prompt includes examples of the intended behavior, and (2) no examples of the intended behavior were seen in training.
  - We are unlikely to be able to verify (2).
  - "Few-shot" is also used in supervised learning with the sense of "training on few examples". The above is different.
- Zero-shot in-context learning: (1) The prompt includes no examples of the intended behavior (but it can contain other instructions), and (2) no examples of the intended behavior were seen in training.
  - We are unlikely to be able to verify (2).
  - Formatting and other instructions seem like a gray area, but we will allow them in this category.

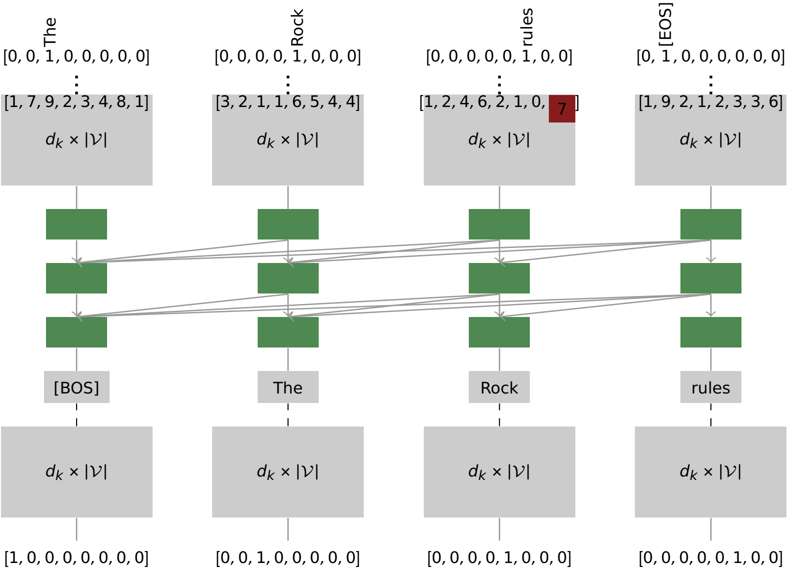
GPT: Autoregressive loss function

For vocabulary $\mathcal{V}$, sequence $\mathbf{x} = [x_1, \ldots, x_T]$, and word-level embedding e:

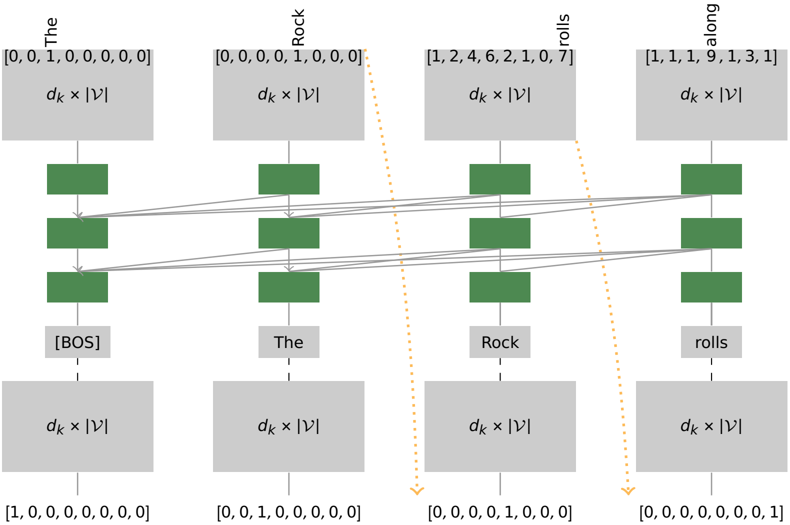
$$\max_\theta \sum_{t=1}^{T} \log \frac{\exp\big(e(x_t)^\top h_\theta(\mathbf{x}_{1:t-1})\big)}{\sum_{x' \in \mathcal{V}} \exp\big(e(x')^\top h_\theta(\mathbf{x}_{1:t-1})\big)}$$

for model parameters h_θ.

Autoregressive training with teacher forcing



Generation


The
Rock
rolls
along
[0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0]
[1, 2, 4, 6, 2, 1, 0, 7]
[1, 1, 1, 9, 1, 3, 1]
$d_k \times |\mathcal{V}|$
[BOS]
The
Rock
rolls
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]

A question

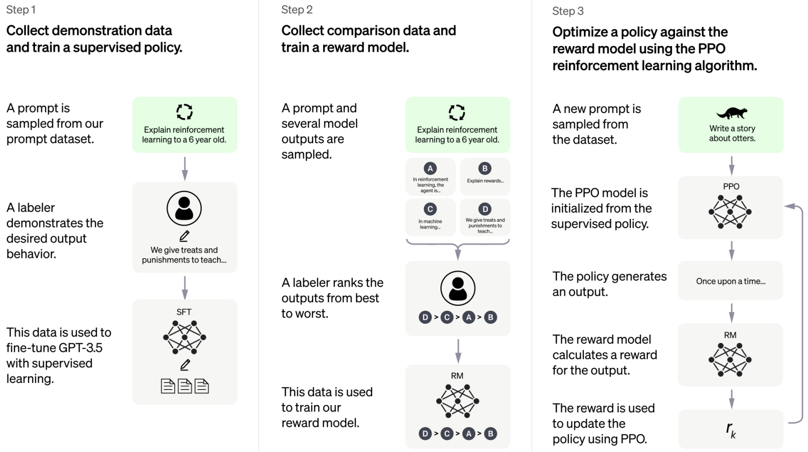
Do autoregressive LMs simply predict the next token?

1. Yes, that is all they do.
2. Well, they predict scores over the entire vocabulary at each step. We then use those scores to compel them to predict some token or other.
3. And, actually, they also represent data in their internal and output representations.
4. But, on balance, saying they simply predict the next token might be best in terms of science communication with the public.

A uniform mechanism

1. Better late than ________
2. Every day, I eat breakfast, lunch, and ________
3. The President of the U.S. is ________
4. The key to happiness is ________

Instruction fine-tuning



https://openai.com/blog/chatgpt

The current moment

Data used for self-supervision

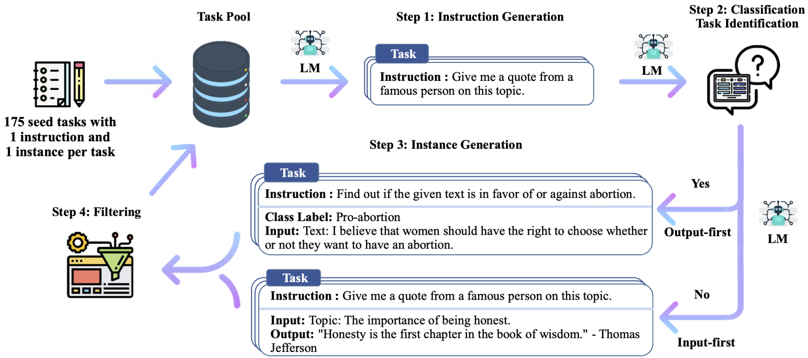
1. OpenBookCorpus (Bandy and Vincent 2021):
   https://huggingface.co/datasets/bookcorpusopen
2. The Pile (Gao et al. 2020):
   https://pile.eleuther.ai
3. Big Science Data (Laurençon et al. 2022):
   https://huggingface.co/bigscience-data
4. Wikipedia processing:
   https://github.com/attardi/wikiextractor
5. Pushshift Reddit Data (Baumgartner et al. 2020):
   https://files.pushshift.io/reddit/
6. Colossal Clean Crawled Corpus (C4; Dodge et al. 2021)
   https://github.com/allenai/allennlp/discussions/5056
   WaPo: Inside the secret list of websites that make AI like ChatGPT sound smart

Data used for instruction fine-tuning

- We don’t know much about what the industrial labs are doing here.
- We can infer that they are paying lots of people to generate Instruct data.
- We can also infer that they are using their own models to generate examples and adjudicate between examples.
- The Stanford Human Preferences Dataset (SHP) is a resource for naturalistic Instruct-tuning:

  https://huggingface.co/datasets/stanfordnlp/SHP

Self-instruct



Wang et al. 2022b

Self-instruct prompt templates

Step 1: Instruction generation

```
Come up with a series of tasks:

Task 1: {instruction for existing task 1}
Task 2: {instruction for existing task 2}
Task 3: {instruction for existing task 3}
Task 4: {instruction for existing task 4}
Task 5: {instruction for existing task 5}
Task 6: {instruction for existing task 6}
Task 7: {instruction for existing task 7}
Task 8: {instruction for existing task 8}
Task 9:
```

Step 2: Classification task identification

```
Task: Given a sentence, detect if there is any potential stereotype in it. If so, you should
explain the stereotype. Else, output no.
Is it classification? No

...

Task: To make the pairs have the same analogy, write the fourth word.
Is it classification? No

Task: Given a set of numbers, find all possible subsets that sum to a given number.
Is it classification? No

Task: {instruction for the target task}
```

Step 3: Classification tasks

```
...

Task: Tell me the first number of the given list.
Class label: 1
List: 1, 2, 3
Class label: 2
List: 2, 9, 10

Task: Which of the following is not an input type? (a) number (b) date (c) phone number (d)
email address (e) all of these are valid inputs.
Class label: (e)

Task: {instruction for the target task}
```

Step 3: Non-classification tasks

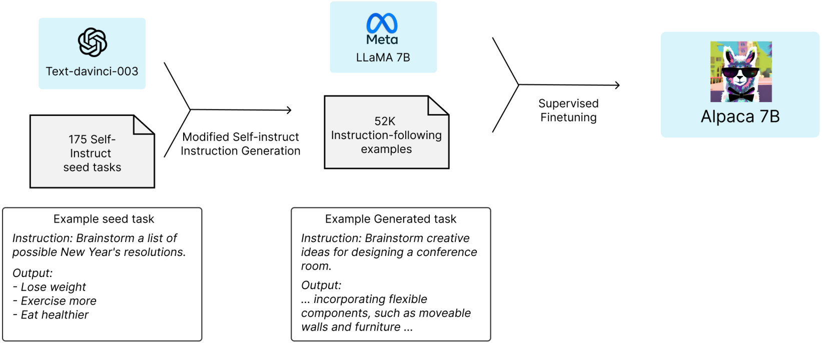
```
...

Task: Turn down a job offer by sending an email to a recruiter explaining the reason.
Output: Hi [Recruiter],
Thank you so much for the generous offer to join your team. As we discussed, I've admired the
company for a number of years, and am a proud endorser of its products. However, after further
consideration of where I currently am in my career, I've decided to accept an offer at another
company.
I would love to stay in touch with you and have already started following you on [Social Media
Platform]. Again, thank you so much for your time and consideration.
Thanks again,
[Your Name]

Task: {Instruction for the target task}
```

Wang et al. 2022b

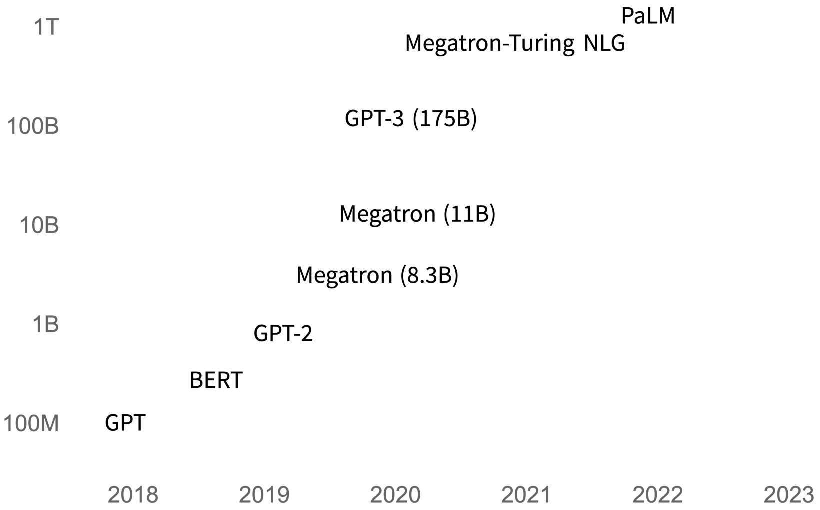
Alpaca



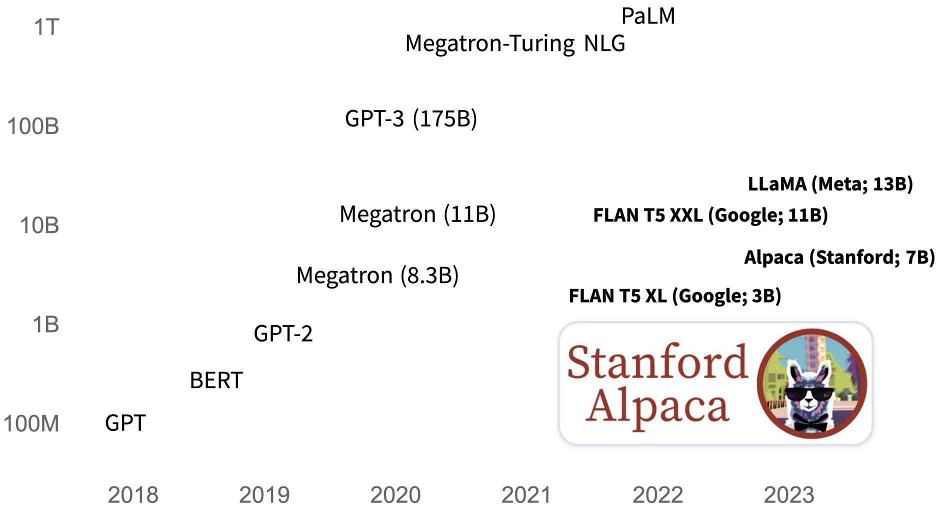
Link to the modified self-instruct procedure

`https://crfm.stanford.edu/2023/03/13/alpaca.html`

Model sizes go up up up



Model sizes may be coming down



Techniques

Demonstrations

`Context: Kermit is one of the stars of Sesame Street.`	Train/Retrieve
`Q: Who is Kermit?`	Train
`A: Kermit is the one of the stars of Sesame Street.`	Train/Retrieve/Generate
`Context: Bert is a Muppet who lives with Ernie.`	Retrieve
`Q: Who is Bert?`	Given
`A: Bert is a Muppet.`	Predicted

Choosing demonstrations

1. Randomly chosen from available data.
2. Chosen based on relationship to the target example.
   - Generation: Retrieved based on similarity to the target input.
   - Classification: Chosen to help the model implicitly determine the target input type.
3. Filtered to those that satisfy specific criteria:
   - Generation: The evidence contains the output.
   - Generation: The LM predicts the correct output.
   - Classification: Every label represented.
4. Sampled and then rewritten by the LM:
   - Synthesize multiple initial demonstrations into individual demonstrations.
   - Change style or formatting to match the target.

Something to get used to: Your prompt might contain substrings that were generated by a different prompt to your LM.

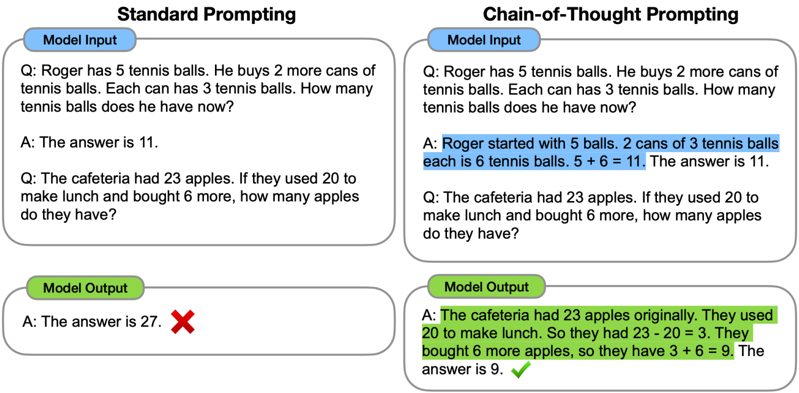
Example from Assignment 2

Context: ELMo is an LSTM for contextual reps. Q: Who is ELMo? A: ELMo is a friendly monster Context: Bert is a Muppet who lives with Ernie. Q: Who is Bert? A:	Context: The Grover model detects fake news. Q: What is Grover? A: Grover is an LLM Context: ELMo is an LSTM for contextual reps Q: Who is ELMo? A: ELMo is an LSTM ✗

Example from Assignment 2

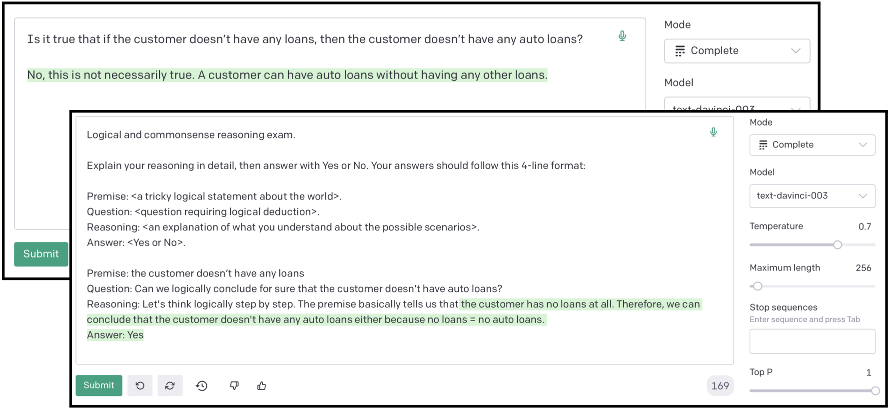
Context: Bert and Ernie are best friends. Q: Who is Ernie? A: Ernie is Bert's best friend. Context: Bert is a Muppet who lives with Ernie. Q: Who is Bert? A: Bert is a Muppet.	Context: Big Bird is a giant yellow bird. Q: Who is Big Bird? A: Big Bird is a Muppet bird Context: Bert and Ernie are best friends. Q: Who is Ernie? A: Ernie is Bert's friend ✓

Chain of Thought

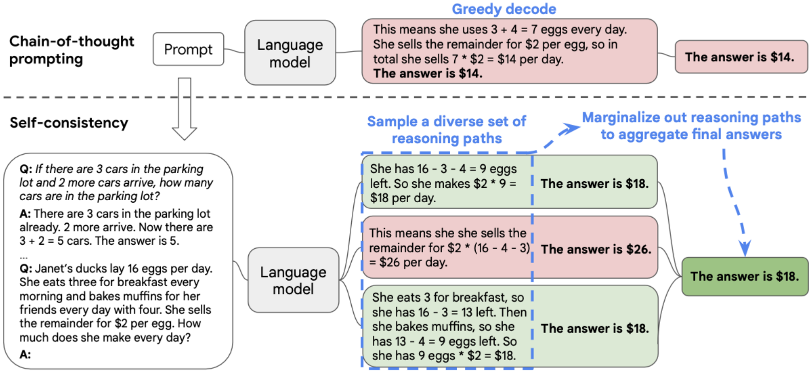


Wei et al. 2022

Generic step-by-step with instructions



Self-Consistency



Wang et al. 2022a;
See also Retrieval Augmented Generation (RAG; Lewis et al. 2020)

Self-Consistency in DSP

```python
import dsp

@dsp.transformation
def predict_with_sc(example):
    generator = dsp.generate(qa_template, n=20, temperature=0.7)
    example, compl = generator(example, stage='qa')
    compl = dsp.majority(compl)
    return example.copy(answer=completions.answer)
```

https://github.com/stanfordnlp/dsp/blob/main/intro.ipynb

Self-Ask

Direct Prompting

GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Answer: Harry Vaughan Watkins.

Question: Who was president of the U.S. when superconductivity was discovered?
Answer: Franklin D. Roosevelt

Chain of Thought

GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Answer: Theodor Haecker was 65 years old when he died. Harry Vaughan Watkins was 69 years old when he died.
So the final answer (the name of the person) is: Harry Vaughan Watkins.

Question: Who was president of the U.S. when superconductivity was discovered?
Answer: Superconductivity was discovered in 1911 by Heike Kamerlingh Onnes. Woodrow Wilson was president of the United States from 1913 to 1921. So the final answer (the name of the president) is: Woodrow Wilson.

Self-Ask

GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Are follow up questions needed here: Yes.
Follow up: How old was Theodor Haecker when he died?
Intermediate answer: Theodor Haecker was 65 years old when he died.
Follow up: How old was Harry Vaughan Watkins when he died?
Intermediate answer: Harry Vaughan Watkins was 69 years old when he died.
So the final answer is: Harry Vaughan Watkins

Question: Who was president of the U.S. when superconductivity was discovered?
Are follow up questions needed here: Yes.
Follow up: When was superconductivity discovered?
Intermediate answer: Superconductivity was discovered in 1911.
Follow up: Who was president of the U.S. in 1911?
Intermediate answer: William Howard Taft.
So the final answer is: William Howard Taft.

Self-Ask can be combined with retrieval for answering the intermediate questions.

Press et al. 2022
https://github.com/ofirpress/self-ask/

Iterative rewriting

```python
@dsp.transformation
def multihop_search_v2(example, max_hops=3):
    example.hops = []
    generator = dsp.generate(hop_template)
    for hop in range(max_hops):
        summary, query = generator(example)
        example.hops.append((summary, query))
        if query == 'N/A': break
     passages = dsp.retrieve(query, k=5)
     example.context = [summary] + passages
     return example
```

```
My task is to write a simple query that gathers information for answering a complex question. I write N/A if the context contains all information required.

{Task demonstrations from x.demos, if any}

Context: {x.context}
Question: {x.question}
Summary: Let's summarize the above context. __{summary}__
Search Query: __{query}__
```

Some DSP results

	Open-SQuAD		HotPotQA		QReCC		Open-MuSiQue				PopQA			
	EM	F1	EM	F1	F1	nF1	2H	3H	4H	S@7	<25p	<50p	<75p	Popular
Vanilla LM	16.2	25.6	28.3	36.4	29.8	18.4	8.7/16.8	4.0 / 13.8	3.6 / 12.2	N/A	23.9	27.0	33.6	**63.3**
No-retrieval LM SoTA	20.2[¶]	–	33.8[¶]	44.6[¶]	–	–	–	–	–	–	–	–	–	–
Retrieve-then-Read	33.8	46.1	36.9	46.1	31.6	22.2	11.4/20.0	3.3/10.7	2.9/12.7	26.7	41.7	40.0	39.1	50.1
Self-ask (w/ ColBERTv2)	9.3	17.2	25.2	33.2	—	–	15.2[¶]/–	–	–	–	–	–	–	–
+ Refined Prompt	9.0	15.7	28.6	37.3	–	–	–	–	–	–	–	–	–	–
Retrieval-aug. LM SoTA	34.0[¶]	–	35.1[¶]	–	–	–	–	–	–	–	–	–	–	–
Task-aware DSP Program	**36.6**	**49.0**	**51.4**	**62.9**	**35.0**	**25.3**	**24.6/36.0**	**13.5/22.7**	**7.0/13.7**	**49.2**	**44.3**	**40.4**	**42.2**	61.9

- Open-SQuAD: Demonstrations selected essentially as in HW, Q2. Predict uses self-consistency.
- HotPotQA: Iterative summary of retrieved passages, iterative generation of questions. Predict uses self-consistency.
- QReCC: As with the QA tasks, but operating on sets of dialogue turns, with successive summarization of the dialogue context.
- MuSiQue is a test task for multihop, PopQA for OpenQA. PopQA results include a breakdown by entity prevalence.

Khattab et al. 2022

Suggested methods

Suggested methods

- Create dev/test sets for yourself based on the task you want to solve, aiming for a format that can work with a lot of prompts
- Learn what you can about your target model, paying particular attention to whether it was tuned for specific instruction formats.
- Think of prompt writing as AI system design. Try to write systematic, generalizable code for handling the entire workflow from reading data to extracting responses and analyzing results.
- For the current (and perhaps brief) moment, prompt designs involving multiple pretrained components and tools seem to be underexplored relative to their potential value.

References I

Jack Bandy and Nicholas Vincent. 2021. Addressing" documentation debt" in machine learning research: A retrospective datasheet for bookcorpus. *arXiv preprint arXiv:2105.05241*.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The PushShift Reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic. Association for Computational Linguistics.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, and Matt Gardner. 2021. Documenting the english colossal clean crawled corpus. *ArXiv*, abs/2104.08758.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate–Search–Predict: Composing retrieval and language models for knowledge-intensive NLP. Ms., Stanford University.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

References II

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Ms, OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.