Humans Learn From Task Descriptions. And So Should Our Models!

Hinrich Schütze, Timo Schick

Center for Information and Language Processing, LMU Munich

2021-04-20

Outline

1. How do humans learn?
2. GPT3 & task descriptions
3. Pattern Exploiting Training (PET)
4. PET outperforms GPT3

How do humans learn?

How do humans learn?

How do humans learn?

Let's look at a typical example of human learning: How to open and eat a pomegranate

The BEST Way To Open & Eat A Pomegranate:
`https://www.youtube.com/watch?v=5BExPRwPdAs`
timestamps 10s to 45s
Read the closed captions
Pay attention to (i) descriptions, (ii) # training instances

Open & Eat A Pomegranate: What did we see?

- The teacher gives a detailed description of the task and of the solution
- Task description: way of opening/eating that is not "a pain in the butt" and not "messy"
- Solution description: "score the pomegranate along the ridges" etc.
- Very few training instances
- E.g., 3 instances of: "score the pomegranate along the ridge"

A typical form of human learning

- Detailed description
- Very few training instances (10 or fewer)

Typical machine learning setup

- No descriptions
- Large training sets
- Even few-shot learning often uses 1000s of examples

Motivation for our approach

- Humans take advantage of task descriptions, our machine learning models don't.
- This is specifically a problem in few-shot learning.
- How can task descriptions benefit machine learning?
- One success story in NLP: GPT3

Overview

1. How do humans learn?
2. GPT3 & task descriptions
3. Pattern Exploiting Training (PET)
4. PET outperforms GPT3

Team:

- Timo Schick (conception & actual work)
- Hinrich Schütze (PhD advisor)

Outline

1. How do humans learn?
2. GPT3 & task descriptions
3. Pattern Exploiting Training (PET)
4. PET outperforms GPT3

GPT3

- GPT3: a transformer-based language model,
  very large model,
  pretrained on very large corpus
- Key innovation:
  No supervised finetuning for a specific task
- Instead: "in-context learning" –
  I will call this priming in this talk
- The "priming" input to GPT3 consists of
  - Task description
  - A few training instances
  - A cloze question

GPT3 priming (in-context learning)

`Translate English to French:`	*(task description)*
`thanks => merci`	*(training instance 1)*
`hello => bonjour`	*(training instance 2)*
`mint => menthe`	*(training instance 3)*
`cheese =>`	*(cloze question)*

GPT3

- GPT3: a transformer-based language model,
  very large model,
  pretrained on very large corpus
- Key innovation:
  No supervised finetuning for a specific task
- Instead: "in-context learning" –
  I will call this priming in this talk
- The "priming" input to GPT3 consists of
  - Task description
  - A few training instances
  - A cloze question

GPT3

- GPT3: a transformer-based language model,
  very large model,
  pretrained on very large corpus
- Key innovation:
  No supervised finetuning for a specific task
- Instead: “in-context learning” –
  I will call this priming in this talk
- The “priming” input to GPT3 consists of
  - Task description
  - A few training instances
  - A cloze question
- No parameter updates during priming

GPT3

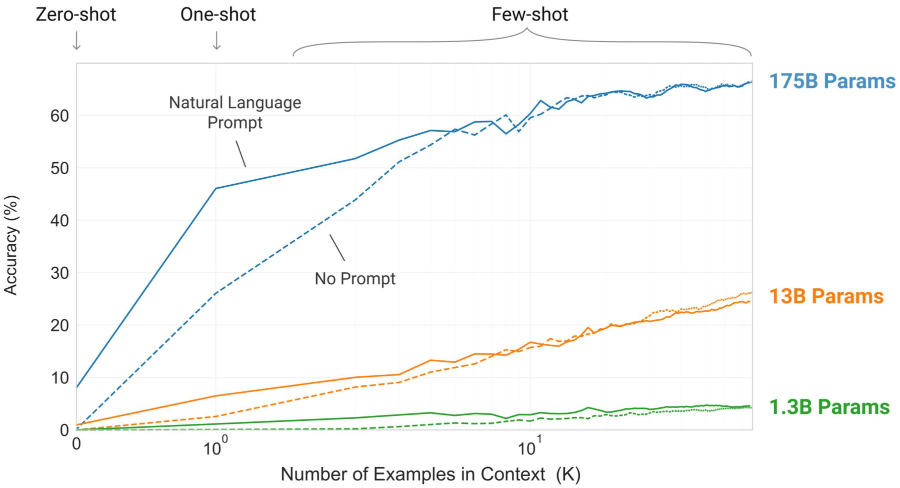
- GPT3: a transformer-based language model,
  very large model,
  pretrained on very large corpus
- Key innovation:
  No supervised finetuning for a specific task
- Instead: "in-context learning" –
  I will call this priming in this talk
- The "priming" input to GPT3 consists of
  - Task description
  - A few training instances
  - A cloze question
- No parameter updates during priming
- → No real learning takes place for a specific task.

GPT3: Excellent few-shot performance

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

GPT3 task description ("prompt") is key for few-shot learning



GPT3 vs. Supervised learning

- Arguably, humans do parameter updates when they learn.
- E.g., you don't start from scratch when you open a second pomegranate a day later.
- In contrast, GPT3 arguably doesn't learn anything after the completion of pretraining!
- So why not use: both task description and supervised learning?
- Which is what humans do ...
- $\rightarrow$ PET

GPT3 vs. Supervised learning

- Arguably, humans do parameter updates when they learn.
- E.g., you don't start from scratch when you open a second pomegranate a day later.
- In contrast, GPT3 arguably doesn't learn anything after the completion of pretraining!
- So why not use: both task description and supervised learning?
- Which is what humans do ...
- $\rightarrow$ PET

Task description: Terminological note

- Description of the task
- vs. Description of an aspect of the task
- vs. Description of the solution
- vs. Description of properties of training instances

Task description: Terminological note

- Description of the task
- vs. Description of an aspect of the task
- vs. Description of the solution
- vs. Description of properties of training instances
- I will use "task description" for all of these –
  to be discussed at the end.

Outline

1. How do humans learn?
2. GPT3 & task descriptions
3. Pattern Exploiting Training (PET)
4. PET outperforms GPT3

Pattern Exploiting Training (PET): Training set

- PET = Pattern Exploiting Training
- Task: Sentiment analysis
- Review: "Excellent pizza!"
- Gold label: 1 (positive)
- Training instance = ("Excellent pizza!",1)
- We vary the size of the training set from 0 to 1000, but are particularly interested in 10.

Pattern Exploiting Training (PET): Pattern

- Define a pattern for the task
- pattern ≈ cloze question
- Example pattern: *review* It was MASK.
  ("*Excellent pizza!* It was MASK.")
- Another example pattern:
  review In summary, the restaurant is MASK.
  ("*Excellent pizza!* In summary, the restaurant is MASK.")

Pattern Exploiting Training (PET): Verbalizer

- Define a verbalizer:
  It associates MASK substitutions with class labels.
- In our example:
  "good" $\leftrightarrow$ 1
  "bad" $\leftrightarrow$ 0
- Here, "good" and "bad" are label descriptions.
- Task description mainly in the form of label descriptions
- This taps into the masked language model's pretrained knowledge of the task.
- The MLM probably knows that
  "Excellent Pizza! It was good."
  is a lot more probable than
  "Excellent Pizza! It was bad."
  (even zero-shot)

Pattern Exploiting Training (PET): Overview

("Excellent pizza!",1)	*training instance*
$\Downarrow$	*use pattern:* "*review* It was MASK."
"Excellent pizza! It was MASK."	
$\Downarrow$	*input to MLM*
Masked Language Model (MLM)	*MLM predicts:* $P(\begin{smallmatrix}\text{good}\\\text{bad}\end{smallmatrix}\,\vert\text{MASK})$
$\Downarrow$	*verbalizer*
good 1 0.82 bad 0 0.18	
$\Uparrow$	*finetune MLM with cross-entropy*
("Excellent pizza!",1)	*training instance*

Formalization

- Pattern $P(x)$,
  function from input to cloze question
- Verbalizer $v(l)$,
  injective function: class labels $\mapsto$ English words
- PVP (pattern-verbalizer pair): (P, v)
- $q(v(l)|P(x))$: for input $P(x)$, the probability that the MLM assigns to substituting $v(l)$ for MASK
  - softmax over "label" words
- Training objective: cross-entropy between $q(v(l)|P(x))$ and truth (discrete distribution)

How to exploit multiple patterns

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

Multiple patterns: Example for sentiment

Verbalizer

$v(\star) =$ terrible
$v(\star\star) =$ bad
$v(\star\star\star) =$ okay
$v(\star\star\star\star) =$ good
$v(\star\star\star\star\star) =$ great

Multiple patterns: Example for sentiment

Verbalizer

$v(\star) =$ terrible
$v(\star\star) =$ bad
$v(\star\star\star) =$ okay
$v(\star\star\star\star) =$ good
$v(\star\star\star\star\star) =$ great

Patterns

$P_1(\textit{review}) =$ "It was MASK. *review*"
$P_2(\textit{review}) =$ "Just MASK. *review*"
$P_3(\textit{review}) =$ "*review*. All in all, it was MASK."
$P_4(\textit{review}) =$ "*review*. In summary, the restaurant is MASK."

Why multiple patterns are critical

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

Why multiple patterns are critical

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

- The patterns provide human expertise – the more the better!

Why multiple patterns are critical

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

- The patterns provide human expertise – the more the better!
- Realistic few-shot learning difficult without human expertise

Why multiple patterns are critical

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

- The patterns provide human expertise – the more the better!
- Realistic few-shot learning difficult without human expertise
- Can we try out multiple patterns and just keep the best one?

Why multiple patterns are critical

x	x	x	*unlabeled instance x*
	$\Downarrow$		
$P_1(x)$	$P_2(x)$	$P_3(x)$	*multiple patterns*
	$\Downarrow$		
MLM_1	MLM_2	MLM_3	*finetuned MLM for each pattern*
	$\Downarrow$		
score_1	score_2	score_3	*each pattern scored by its MLM*
	$\Downarrow$		
	aggregate score		
	$\Downarrow$		
	classifier		*final classification decision*

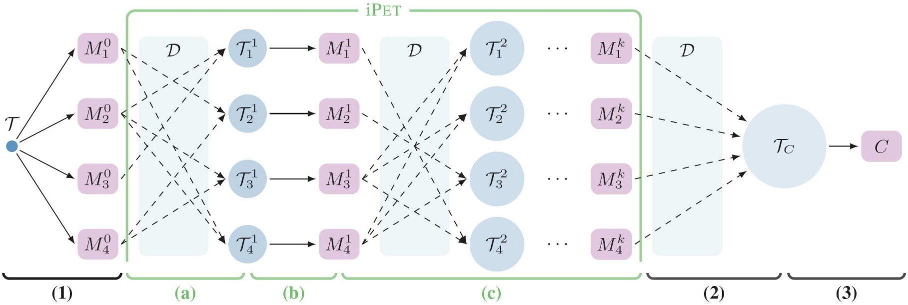
- The patterns provide human expertise – the more the better!
- Realistic few-shot learning difficult without human expertise
- Can we try out multiple patterns and just keep the best one?
- No: no dev set in true few-shot learning

Distillation creates single model from pattern-specific individual models

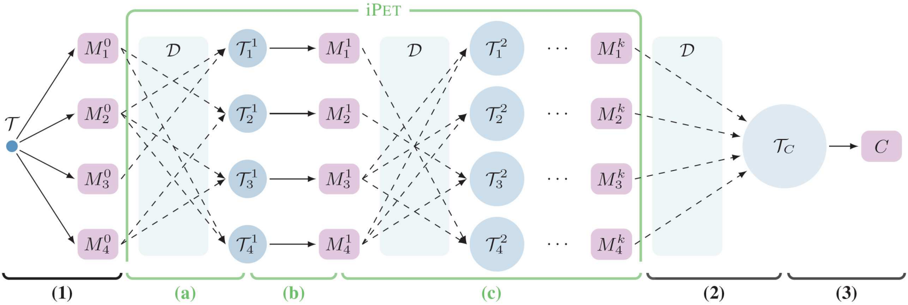
Distillation:

- Use individual models to label an unlabeled dataset $\mathcal{T}$
- Aggregrate scores to label $\mathcal{T}$
- Train final PET model on $\mathcal{T}$

iPET: Iterative training



iPET: Iterative training



$\boxed{\text{iPET} = \text{iterative PET}}$

PET: Key points

- Pattern+verbalizer taps into MLM's pretrained knowledge of the task:
  - Chances are the MLM knows, based on pretraining, that "Excellent Pizza! It was good." is better than "Excellent Pizza! It was bad."
- Patterns are a way of incorporating human expertise into the learning problem.
- PET exploits multiple patterns
  – important to use all human expertise available.
- Truly few-shot: no tuning on dev set
  (which is not available in a true few-shot setup)
- In contrast to GPT3, PET is supervised:
  It takes full advantage of the (small) training set.
- Excellent few-shot performance (next section)

What exactly is a task description?

A straightforward task description

`Translate English to French:`	*(task description)*
`thanks => merci`	*(training instance 1)*
`hello => bonjour`	*(training instance 2)*
`mint => menthe`	*(training instance 3)*
`cheese =>`	*(cloze question)*

Actually, it is not that straightforward

`Translate English to French:`	*(task description)*
`thanks => merci`	*(training instance 1)*
`hello => bonjour`	*(training instance 2)*
`mint => menthe`	*(training instance 3)*
`cheese =>`	*(cloze question)*

PET sentiment: Pattern and verbalizer interact

Verbalizer ("label description")

$v(\star) =$ terrible
$v(\star\star) =$ bad
$v(\star\star\star) =$ okay
$v(\star\star\star\star) =$ good
$v(\star\star\star\star\star) =$ great

Patterns

$P_1(review) =$ "It was MASK. *review*"
$P_2(review) =$ "Just MASK. *review*"
$P_3(review) =$ "*review*. All in all, it was MASK."
$P_4(review) =$ "*review*. In summary, the restaurant is MASK."

PET "Word in Context": Task description as question

Verbalizer ("label description")

$$
\begin{aligned}
v(\text{same\_sense}) &= \text{yes} \\
v(\text{different\_senses}) &= \text{no}
\end{aligned}
$$

Pattern

$P_1(s_1, s_2, w) = s_1\ s_2$ Does w have the same meaning in both sentences? MASK

PET "Winograd Schema Challenge": No use of label descriptions

Verbalizer (not a label description)

$v(w) = w$ (identity, for all words)

Pattern

$P_1(s) = s$ In the previous sentence, the pronoun "$\star p \star$" refers to MASK.

What exactly is a task description?

What exactly is a task description?

- Task descriptions are not simple descriptions of the task.

What exactly is a task description?

- Task descriptions are not simple descriptions of the task.
- They can be complex translations of the structure of the task into plain text (plus a MASK).

What exactly is a task description?

- Task descriptions are not simple descriptions of the task.
- They can be complex translations of the structure of the task into plain text (plus a MASK).
- Task descriptions are created by the system designer based on their understanding of task and language model.

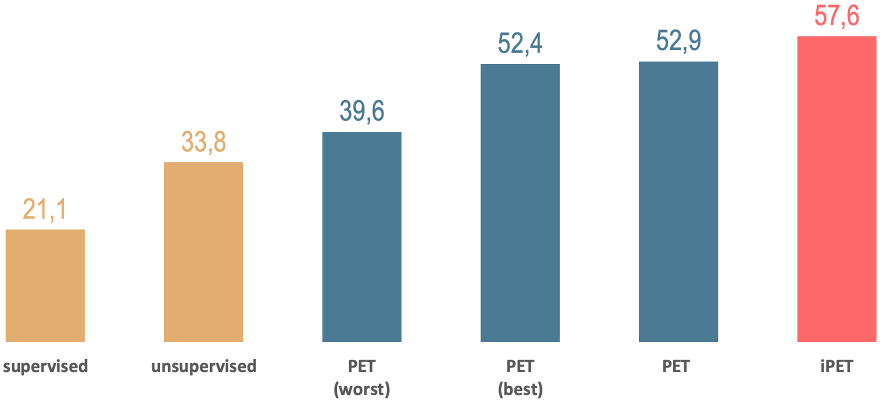
What exactly is a task description?

- Task descriptions are not simple descriptions of the task.
- They can be complex translations of the structure of the task into plain text (plus a MASK).
- Task descriptions are created by the system designer based on their understanding of task and language model.
- Difficult to automate, requires the ingenuity of the system designer.

Outline

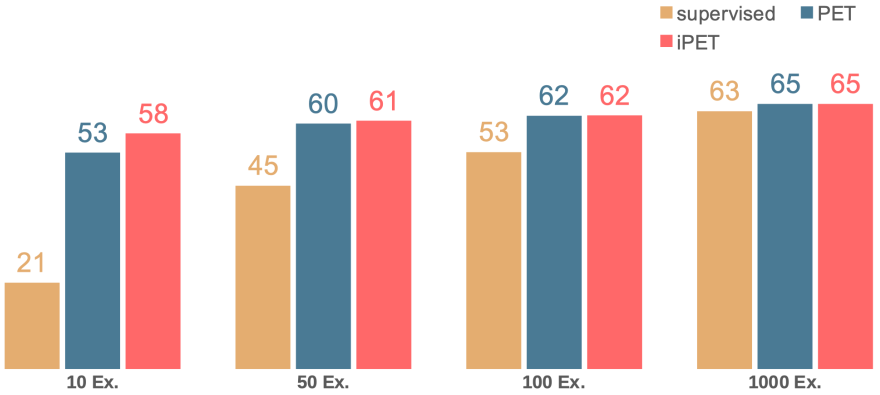
1. How do humans learn?
2. GPT3 & task descriptions
3. Pattern Exploiting Training (PET)
4. PET outperforms GPT3

PET results on YELP FULL, 10 training examples



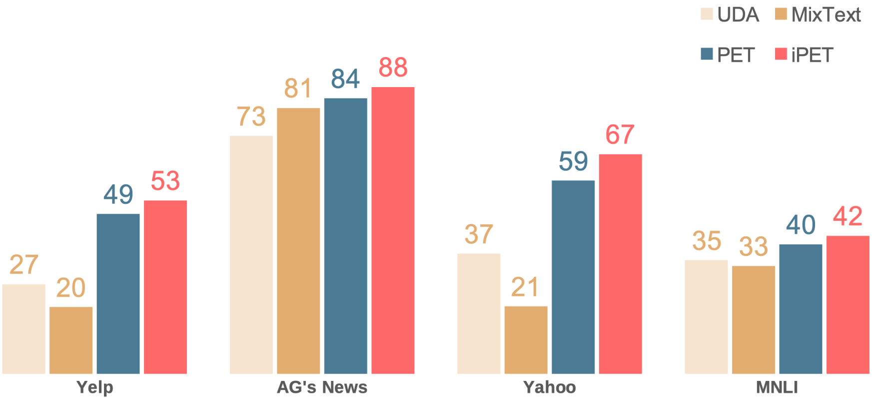
RoBERTa large

PET results on YELP FULL, effect of training set size



RoBERTa large

PET/iPET vs. UDA/MixText, 10 training examples



RoBERTa base

(i)PET vs. GPT3: Size of model

model	# params	
GPT3	175G	100.0%
GPT3 med	350M	0.2%
(i)PET	223M	0.1%

ALBERT xxlarge

(i)PET vs. GPT3: Size of model

model	# params	
GPT3	175G	100.0%
GPT3 med	350M	0.2%
(i)PET	223M	0.1%

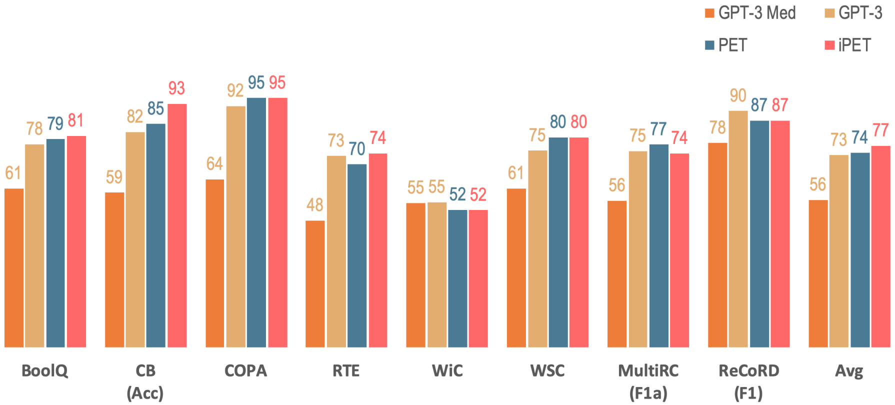
ALBERT xxlarge

(i)PET vs. GPT3: Size of model

model	# params	
GPT3	175G	100.0%
GPT3 med	350M	0.2%
(i)PET	223M	0.1%

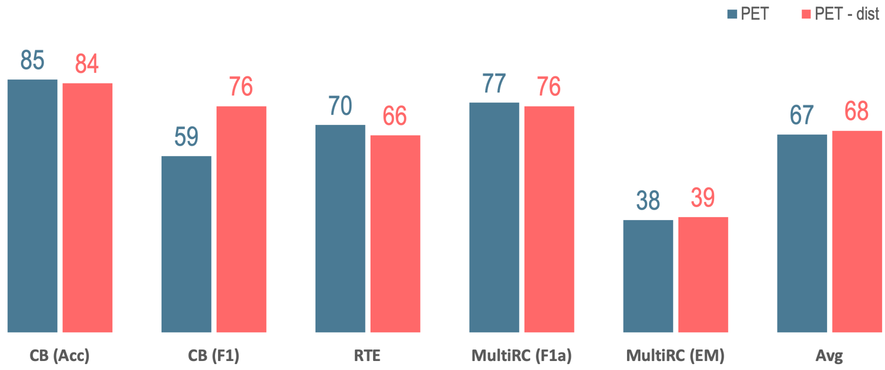
ALBERT xxlarge

PET vs. GPT3 on SuperGLUE, 32 training examples



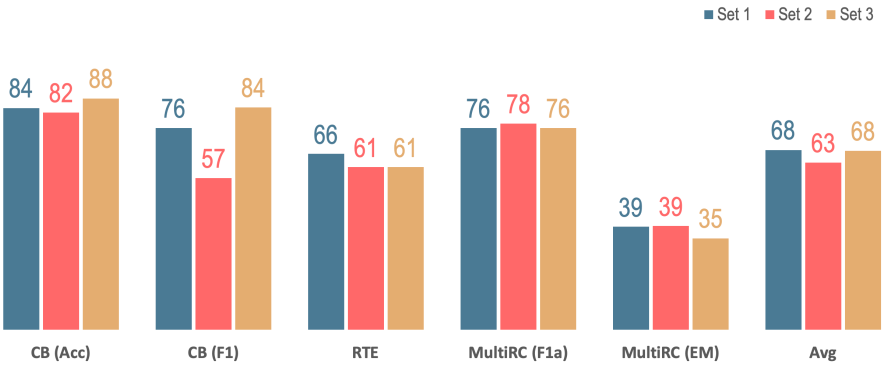
ALBERT xxlarge

Effect of (not) using unlabeled data



ALBERT xxlarge

Different sets of 32 training examples: The choice of shots matters



ALBERT xxlarge

PET vs. GPT3

	PET	GPT3	
perform.	great	great	
model size	small	huge	$\rightarrow$ PET broadly deployable
few shots	no restriction	ctx w. limit	$\rightarrow$ PET can exploit all train data
dev set	not needed?	needed?	few-shot $\rightarrow$ no dev set
supervision	supervised	unsupervised	supervision improves performance
supervision	supervised	unsupervised	different PET model for each task
fluidity	nonfluid	fluid	GPT3 mimicks human fluidity
generation	hard	easy	GPT3 easily handles generative tasks

PET: Summary

- PET leverages task descriptions for better few-shot learning.
- Task descriptions / patterns are a way of incorporating human expertise into the learning problem.
- PET exploits multiple patterns
  – important to use all human expertise available.
- Truly few-shot: no tuning on dev set
  (which is not available in a true few-shot setup)
- In contrast to GPT3, PET is supervised:
  It takes full advantage of the (small) training set.
- Excellent few-shot performance

The full potential of descriptions

The full potential of descriptions

- We have seen diverse types of task descriptions.

The full potential of descriptions

- We have seen diverse types of task descriptions.
- Both in GPT3 and PET

The full potential of descriptions

- We have seen diverse types of task descriptions.
- Both in GPT3 and PET
- Task descriptions in PET are pattern-verbalizer combinations where the verbalizer mostly provides label descriptions.

The full potential of descriptions

- We have seen diverse types of task descriptions.
- Both in GPT3 and PET
- Task descriptions in PET are pattern-verbalizer combinations where the verbalizer mostly provides label descriptions.
- What is key: the method exploits the MLM's understanding of language descriptions for understanding/solving the task.

The full potential of descriptions

- We have seen diverse types of task descriptions.
- Both in GPT3 and PET
- Task descriptions in PET are pattern-verbalizer combinations where the verbalizer mostly provides label descriptions.
- What is key: the method exploits the MLM's understanding of language descriptions for understanding/solving the task.
- This gives the method a head start compared to other few-shot learners.

The full potential of descriptions

- We have seen diverse types of task descriptions.
- Both in GPT3 and PET
- Task descriptions in PET are pattern-verbalizer combinations where the verbalizer mostly provides label descriptions.
- What is key: the method exploits the MLM's understanding of language descriptions for understanding/solving the task.
- This gives the method a head start compared to other few-shot learners.
- Other types of descriptions:
  solution description
  comments on training instances
  useful background information
  ...

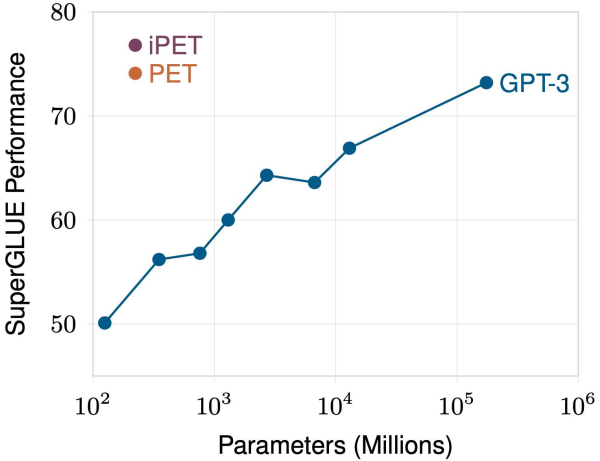
Related work

- Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723, 2020.
  URL https://arxiv.org/abs/2012.15723
- Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training, 2021
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020
- Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. Warp: Word-level adversarial reprogramming, 2021
- Teven Le Scao and Alexander M. Rush. How many data points is a prompt worth?, 2021
- Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts, 2021
- Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Adaprompt: Adaptive prompt-based finetuning for relation extraction, 2021

PET publications

- Timo Schick and Hinrich Schütze. Exploiting cloze questions for few-shot text classification and natural language inference.
  CoRR, abs/2001.07676, 2020b.
  URL `https://arxiv.org/abs/2001.07676` (EACL 2021)
- Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners.
  CoRR, abs/2009.07118, 2020a.
  URL `https://arxiv.org/abs/2009.07118` (NAACL 2021)
- Timo Schick, Helmut Schmid, and Hinrich Schütze. Automatically identifying words that can serve as labels for few-shot text classification.
  In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics, 2020.
  doi: 10.18653/v1/2020.coling-main.488.
  URL `https://doi.org/10.18653/v1/2020.coling-main.488`
- Timo Schick, Sahana Udupa, and Hinrich Schütze. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP.
  CoRR, abs/2103.00453, 2021.
  URL `https://arxiv.org/abs/2103.00453`
- Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models, 2021
- Timo Schick and Hinrich Schütze. Few-shot text generation with pattern-exploiting training, 2020

GPT3/PET: Size vs. performance



PET/iPET performance = single points