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?

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
How do humans learn?

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
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)
How do humans learn?

GPT3 & task descriptions

Pattern Exploiting Training (PET)

PET outperforms GPT3

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)

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Training Instance 1</th>
<th>Training Instance 2</th>
<th>Training Instance 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate English to French:</td>
<td>thanks =&gt; merci</td>
<td>hello =&gt; bonjour</td>
<td>mint =&gt; menthe</td>
</tr>
<tr>
<td>cheese =&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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: 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

<table>
<thead>
<tr>
<th></th>
<th>SuperGLUE Average</th>
<th>BoolQ Accuracy</th>
<th>CB Accuracy</th>
<th>CB F1</th>
<th>COPA Accuracy</th>
<th>RTE Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned SOTA</td>
<td>89.0</td>
<td>91.0</td>
<td>96.9</td>
<td>93.9</td>
<td>94.8</td>
<td>92.5</td>
</tr>
<tr>
<td>Fine-tuned BERT-Large</td>
<td>69.0</td>
<td>77.4</td>
<td>83.6</td>
<td>75.7</td>
<td>70.6</td>
<td>71.7</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>71.8</td>
<td>76.4</td>
<td>75.6</td>
<td>52.0</td>
<td>92.0</td>
<td>69.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>WiC Accuracy</th>
<th>WSC Accuracy</th>
<th>MultiRC Accuracy</th>
<th>MultiRC F1a</th>
<th>ReCoRD Accuracy</th>
<th>ReCoRD F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-tuned SOTA</td>
<td>76.1</td>
<td>93.8</td>
<td>62.3</td>
<td>88.2</td>
<td>92.5</td>
<td>93.3</td>
</tr>
<tr>
<td>Fine-tuned BERT-Large</td>
<td>69.6</td>
<td>64.6</td>
<td>24.1</td>
<td>70.0</td>
<td>71.3</td>
<td>72.0</td>
</tr>
<tr>
<td>GPT-3 Few-Shot</td>
<td>49.4</td>
<td>80.1</td>
<td>30.5</td>
<td>75.4</td>
<td>90.2</td>
<td>91.1</td>
</tr>
</tbody>
</table>
GPT3 task description ("prompt") is key for few-shot learning

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
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 . . .

→ 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 . . .
- → 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 \(\approx\) 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

(training instance)

use pattern: “review It was MASK.”

input to MLM

MLM predicts: \(P(\text{good} | \text{MASK})\)

verbalizer

finetune MLM with cross-entropy

(training instance)

("Excellent pizza!", 1)

⇓

"Excellent pizza! It was MASK."

⇓

Masked Language Model (MLM)

⇓

good 1 0.82
bad 0 0.18

⇑

("Excellent pizza!", 1)
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

\[ x \quad x \quad x \quad x \]
\[ \Downarrow \]
\[ P_1(x) \quad P_2(x) \quad P_3(x) \]
\[ \Downarrow \]
\[ \text{MLM}_1 \quad \text{MLM}_2 \quad \text{MLM}_3 \]
\[ \Downarrow \]
\[ \text{score}_1 \quad \text{score}_2 \quad \text{score}_3 \]
\[ \Downarrow \]
\[ \text{aggregate score} \]
\[ \Downarrow \]
\[ \text{classifier} \]

unlabeled instance \( x \)

multiple patterns

finetuned MLM for each pattern

each pattern scored by its MLM

final classification decision
Multiple patterns: Example for sentiment

Verbalizer

\[ v(\star) = \text{terrible} \]
\[ v(\star\star) = \text{bad} \]
\[ v(\star\star\star) = \text{okay} \]
\[ v(\star\star\star\star) = \text{good} \]
\[ v(\star\star\star\star\star) = \text{great} \]
Multiple patterns: Example for sentiment

**Verbalizer**

\[ v(\star) = \text{terrible} \]
\[ v(\star\star) = \text{bad} \]
\[ v(\star\star\star) = \text{okay} \]
\[ v(\star\star\star\star) = \text{good} \]
\[ v(\star\star\star\star\star) = \text{great} \]

**Patterns**

\[ P_1(\text{review}) = \text{“It was MASK. } \text{review”} \]
\[ P_2(\text{review}) = \text{“Just MASK. } \text{review”} \]
\[ P_3(\text{review}) = \text{“review. All in all, it was MASK.”} \]
\[ P_4(\text{review}) = \text{“review. In summary, the restaurant is MASK.”} \]
Why multiple patterns are critical

\[ \begin{align*}
\text{unlabeled instance } x \\
\text{multiple patterns} \\
\text{finetuned MLM for each pattern} \\
\text{each pattern scored by its MLM} \\
\text{final classification decision}
\end{align*} \]

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
Why multiple patterns are critical

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

\[
x \quad \times \quad \times \\
P_1(x) \quad P_2(x) \quad P_3(x) \\
\downarrow \\
\text{MLM}_1 \quad \text{MLM}_2 \quad \text{MLM}_3 \\
\downarrow \\
\text{score}_1 \quad \text{score}_2 \quad \text{score}_3 \\
\downarrow \\
\text{aggregate score} \\
\downarrow \\
\text{classifier}
\]

unlabeled instance \( x \)

multiple patterns

finetuned MLM for each pattern

each pattern scored by its MLM

final classification decision
Why multiple patterns are critical

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

<table>
<thead>
<tr>
<th>unlabeled instance x</th>
<th>multiple patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1(x) )</td>
<td>( P_2(x) )</td>
</tr>
<tr>
<td>( P_3(x) )</td>
<td>( P_3(x) )</td>
</tr>
<tr>
<td>MLM (_1 )</td>
<td>MLM (_2 )</td>
</tr>
<tr>
<td>MLM (_3 )</td>
<td>MLM (_3 )</td>
</tr>
<tr>
<td>score (_1 )</td>
<td>score (_2 )</td>
</tr>
<tr>
<td>score (_3 )</td>
<td>score (_3 )</td>
</tr>
<tr>
<td>aggregate score</td>
<td></td>
</tr>
<tr>
<td>classifier</td>
<td></td>
</tr>
</tbody>
</table>

The patterns provide human expertise – the more the better!

Realistic few-shot learning difficult without human expertise
Why multiple patterns are critical

- 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?

unlabeled instance $x$

multiple patterns

$P_1(x)$ $P_2(x)$ $P_3(x)$

finetuned MLM for each pattern

MLM$_1$ MLM$_2$ MLM$_3$

each pattern scored by its MLM

score$_1$ score$_2$ score$_3$

aggregate score

classifier

final classification decision
Why multiple patterns are critical

- 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

\[
\begin{align*}
\text{unlabeled instance } x \\
\downarrow \\
P_1(x) & P_2(x) & P_3(x) \\
\downarrow \\
\text{multiple patterns} \\
\downarrow \\
\text{finetuned MLM for each pattern} \\
\downarrow \\
\text{each pattern scored by its MLM} \\
\downarrow \\
\text{aggregate score} \\
\downarrow \\
\text{classifier} \\
\end{align*}
\]
Distillation creates single model from pattern-specific individual models

Distillation:
- Use individual models to label an unlabeled dataset $\mathcal{T}$
- Aggregate scores to label $\mathcal{T}$
- Train final PET model on $\mathcal{T}$
iPET: Iterative training
iPET: Iterative training

\begin{align*}
\text{iPET} &= \text{iterative PET}
\end{align*}
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:

thanks => merci
hello => bonjour
mint => menthe
cheese =>

(task description)
(training instance 1)
(training instance 2)
(training instance 3)
(cloze question)
Actually, it is not that straightforward

Translate English to French:
thanks => merci
hello => bonjour
mint => menthe
cheese =>

(task description)
(training instance 1)
(training instance 2)
(training instance 3)
(cloze question)
PET sentiment: Pattern and verbalizer interact

Verbalizer ("label description")

\[ v(\star) = \text{terrible} \]
\[ v(\star\star) = \text{bad} \]
\[ v(\star\star\star) = \text{okay} \]
\[ v(\star\star\star\star) = \text{good} \]
\[ v(\star\star\star\star\star) = \text{great} \]

Patterns

\[ P_1(\text{review}) = "\text{It was MASK. } \text{review}" \]
\[ P_2(\text{review}) = "\text{Just MASK. } \text{review}" \]
\[ P_3(\text{review}) = "\text{review. All in all, it was MASK.}" \]
\[ P_4(\text{review}) = "\text{review. In summary, the restaurant is MASK.}" \]
PET “Word in Context”: Task description as question

Verbalizer ("label description")

\[ \nu(\text{same\_sense}) = \text{yes} \]
\[ \nu(\text{different\_senses}) = \text{no} \]

Pattern

\[ P_1(s_1, s_2, w) = s_1 s_2 \text{ Does } w \text{ have the same meaning in both sentences? MASK} \]
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 “⋆p⋆” 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

- Supervised: 21.1
- Unsupervised: 33.8
- PET (worst): 39.6
- PET (best): 52.4
- PET: 52.9
- iPET: 57.6

RoBERTa large
PET results on YELP FULL, effect of training set size

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
How do humans learn? GPT3 & task descriptions  
Pattern Exploiting Training (PET)  
PET outperforms GPT3

(i) PET vs. GPT3: Size of model

<table>
<thead>
<tr>
<th>model</th>
<th># params</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT3</td>
<td>175G</td>
<td>100.0%</td>
</tr>
<tr>
<td>GPT3 med</td>
<td>350M</td>
<td>0.2%</td>
</tr>
<tr>
<td>(i)PET</td>
<td>223M</td>
<td>0.1%</td>
</tr>
</tbody>
</table>
(i) PET vs. GPT3: Size of model

<table>
<thead>
<tr>
<th>model</th>
<th># params</th>
<th># params as %</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT3</td>
<td>175G</td>
<td>100.0%</td>
</tr>
<tr>
<td>GPT3 med</td>
<td>350M</td>
<td>0.2%</td>
</tr>
<tr>
<td>(i)PET</td>
<td>223M</td>
<td>0.1%</td>
</tr>
</tbody>
</table>
### (i) PET vs. GPT3: Size of model

<table>
<thead>
<tr>
<th>Model</th>
<th>Size</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT3</td>
<td>175G</td>
<td>100.0%</td>
</tr>
<tr>
<td>GPT3 med</td>
<td>350M</td>
<td>0.2%</td>
</tr>
<tr>
<td>(i)PET</td>
<td>223M</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models.
PET vs. GPT3 on SuperGLUE, 32 training examples
Effect of (not) using unlabeled data
How do humans learn? GPT3 & task descriptions 

Pattern Exploiting Training (PET) 

PET outperforms GPT3 

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

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
# PET vs. GPT3

<table>
<thead>
<tr>
<th>Feature</th>
<th>PET</th>
<th>GPT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform.</td>
<td>great</td>
<td>great</td>
</tr>
<tr>
<td>Model size</td>
<td>small</td>
<td>huge</td>
</tr>
<tr>
<td>Few shots</td>
<td>no restriction</td>
<td>ctx w. limit</td>
</tr>
<tr>
<td>Dev set</td>
<td>not needed?</td>
<td>needed?</td>
</tr>
<tr>
<td>Supervision</td>
<td>supervised</td>
<td>unsupervised</td>
</tr>
<tr>
<td>Supervision</td>
<td>supervised</td>
<td>unsupervised</td>
</tr>
<tr>
<td>Fluidity</td>
<td>nonfluid</td>
<td>fluid</td>
</tr>
<tr>
<td>Generation</td>
<td>hard</td>
<td>easy</td>
</tr>
</tbody>
</table>

→ PET broadly deployable
→ PET can exploit all train data
few-shot → no dev set
supervision improves performance
different PET model for each task
GPT3 mimicks human fluidity
GPT3 easily handles generative tasks

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
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

- Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. *Improving and simplifying pattern exploiting training,* 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. Generating datasets with pretrained language models, 2021

- Timo Schick and Hinrich Schütze. Few-shot text generation with pattern-exploiting training, 2020
Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models
GPT3/PET: Size vs. performance

PET/iPET performance = single points

Schütze & Schick (LMU Munich): Humans learn from task descriptions and so should our models