Learning from natural language instructions

Qiqi Chen

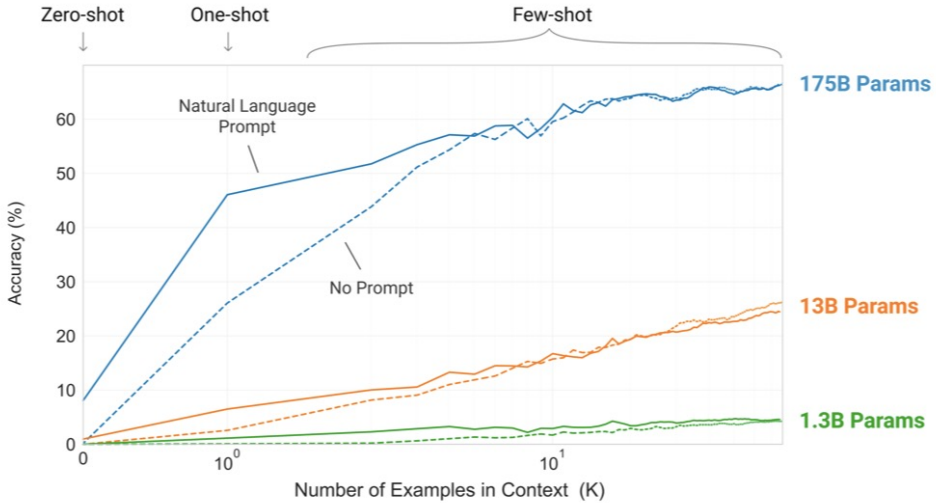
May 19, 2023

TABLE OF CONTENTS

01

Motivation

Motivation



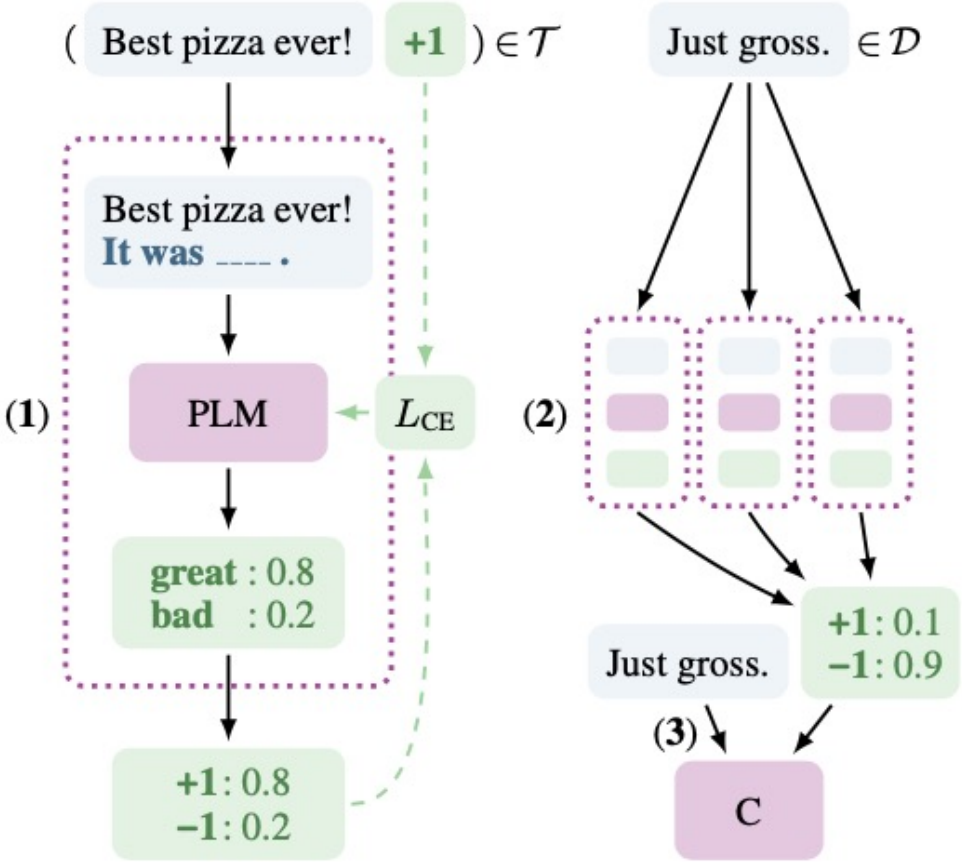
- traditional supervised learning: given large amount of data and label -> find hidden pattern
- few-shot-learning -> task description
- pretrained language model with task descriptions underperforms its supervised counterpart

PET & iPET

Pattern-Exploiting Training (PET), a semi-supervised training procedure that reformulates input examples as cloze-style phrases to help language models understand a given task.

02

Pattern-Exploiting Training



(1) A number of patterns encoding some form of task description are created to convert training examples to cloze questions; for each pattern, a pretrained language model is finetuned.
(2) The ensemble of trained models annotates unlabeled data.
(3) A classifier is trained on the resulting soft-labeled dataset.

T: a small training set
D: a large unlabeled dataset

Figure 1: PET for sentiment classification.

convert training data into cloze questions

vocabulary V and mask token ___ ∈V

input $x = (s_1,...,s_k)$ with $s_i \in V^*$

x = (Mia likes pie, Mia hates pie)

define a *pattern* to be a function P that takes x as input and outputs a phrase or sentence $P(x) \in V^*$ that contains exactly one mask token, i.e., its output can be viewed as a cloze question

L be a set of labels for our target classification task

define a *verbalizer* as an injective function $v : L \rightarrow V$ that maps each label to a word from vocabulary V

We refer to (P, v) as a *pattern-verbalizer pair* (PVP).

P(a,b)= a? ___,b. combined with a verbalizer v that maps y_0 to “Yes” and y_1 to “No”.

P(x)= Mia likes pie? ___, Mia hates pie.

“Yes” or “No”.

core idea of PET

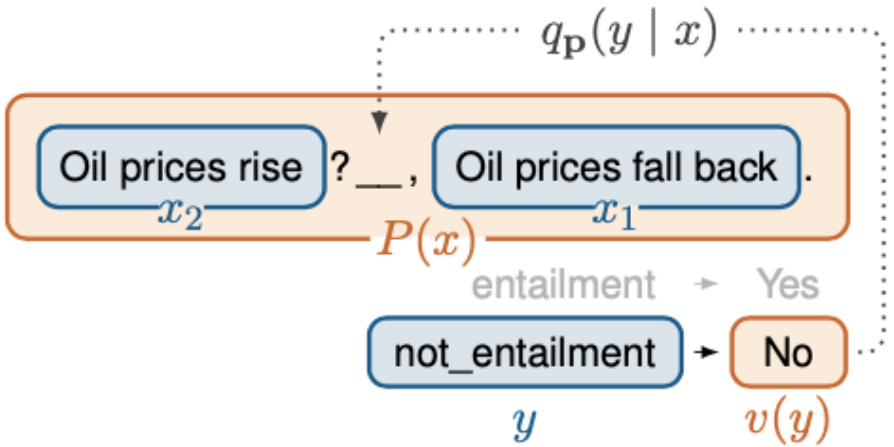


Figure 2: Application of a PVP $\mathbf{p} = (P, v)$ for recognizing textual entailment: An input $x = (x_1, x_2)$ is converted into a cloze question $P(x)$; $q_{\mathbf{p}}(y \mid x)$ for each y is derived from the probability of $v(y)$ being a plausible choice for the masked position.

the core idea of PET is to derive the probability of y being the correct output for x from the probability of v(y) being the “correct” token at the masked position in P(x)

Challenge: how to identify which PVPs perform well in the absence of a large development set?

-> combination of multiple PVPs P = {$p_1, \ldots, p_n$}

1. For each PVP $\mathbf{p}$, a MLM is finetuned on training examples (x, y) by minimizing the cross entropy between y and $q_{\mathbf{p}}(y \mid x)$. In practice, Schick and Schütze (2021) train three MLMs per pattern as performance can vary substantially between runs.

2. The ensemble of finetuned MLMs is used to annotate a set of unlabeled examples; each unlabeled example $x \in X$ is annotated with soft labels based on the probability distribution

$$q_{\mathbf{P}}(y \mid x) \propto \exp \sum_{\mathbf{p} \in \mathbf{P}} w_{\mathbf{p}} \cdot s_{\mathbf{p}}(y \mid x) \quad (2)$$

similar to Eq. 1 where $w_{\mathbf{p}}$ is a weighting term that is proportional to the accuracy achieved with $\mathbf{p}$ on the training set *before* training.

3. The resulting soft-labeled dataset is used to train a regular sequence classifier by minimizing cross entropy between its output and $q_{\mathbf{P}}$.

-> knowledge distillation

Iterative PET (iPET)

- As some patterns perform (possibly much) worse than others, the training set T_C for the final model may therefore contain many mislabeled examples.
- -> iPET, an iterative variant of PET. The core idea of iPET is to train several *generations* of models on datasets of increasing size.
- With minor adjustments, iPET can even be used in a zero-shot setting.

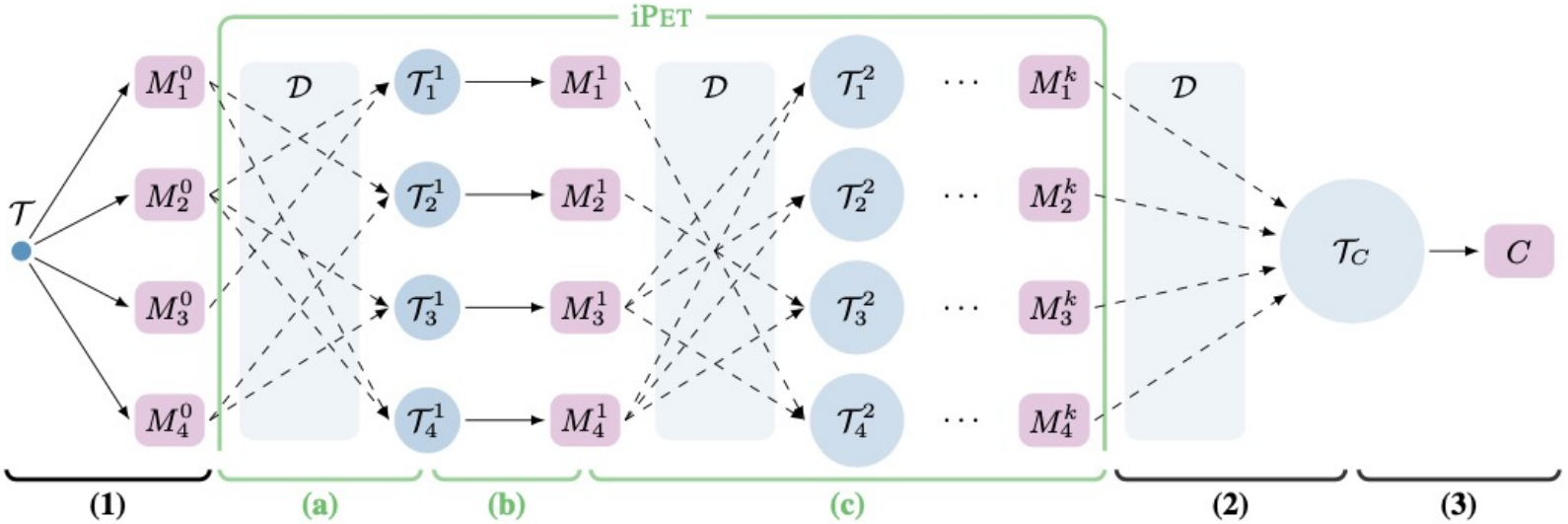


Figure 2: Schematic representation of PET (1-3) and iPET (a-c).
(1) The initial training set is used to finetune an ensemble of PLMs.
(a) For each model, a random subset of other models generates a new training set by labeling examples from D.
(b) A new set of PET models is trained using the larger, model-specific datasets.
(c) The previous two steps are repeated k times, each time increasing the size of the generated training sets by a factor of d.
(2) The final set of models is used to create a soft-labeled dataset T_C.
(3) A classifier C is trained on this dataset.

03

Experiments

Experiment settings

language	English	other languages
datasets	Yelp Reviews, AG's News, Yahoo Questions and MNLI	x-stance
language model	RoBERTa (large)	XLM-R (base)
Number of parameters	355M	270M

- each model is trained three times using different seeds and average results are reported.

Datasets & Patterns

dataset	task	patterns	verbalizer
Yelp	estimate the rating that a customer gave to a restaurant on a 1 to 5-star scale based on their review's text	$P_1(a) =$ It was ____. a $P_2(a) =$ Just ____! || a $P_3(a) =$ a. All in all, it was ____. $P_4(a) =$ a || In summary, the restaurant is ____.	$v(1) =$ terrible $v(2) =$ bad $v(3) =$ okay $v(4) =$ good $v(5) =$ great
AG's News	news classification: given a headline *a* and text body *b*, news have to be classified as belonging to one of the categories World (1), Sports (2), Business (3) or Science/Tech (4).	$P_1(\mathbf{x}) =$ ____: a b $P_2(\mathbf{x}) =$ a (____) b $P_3(\mathbf{x}) =$ ____ – a b $P_4(\mathbf{x}) =$ a b (____) $P_5(\mathbf{x}) =$ ____ News: a b $P_6(\mathbf{x}) =$ [Category: ____] a b	verbalizer maps 1–4 to "World", "Sports", "Business" and "Tech", respectively
Yahoo	text classification: given a question *a* and an answer *b*, one of ten possible categories has to be assigned	same patterns as for AG's News, but replace the word "News" in P5 with the word "Question"	verbalizer that maps categories 1–10 to "Society", "Science", "Health", "Education", "Computer", "Sports", "Business", "Entertainment", "Relation- ship" and "Politics"

Datasets & Patterns

dataset	task	patterns	verbalizer
MNLI	Given text pairs x = (a, b). The task is to find out whether *a* implies *b* (0), *a* and *b* contradict each other (1) or neither (2).	$P_1(\mathbf{x}) = \text{“}a\text{”? } \| \text{ ____, “}b\text{”}$ $P_2(\mathbf{x}) = a? \| \text{ ____, } b$	$v_1(0) = \text{Wrong}$ $v_1(1) = \text{Right}$ $v_1(2) = \text{Maybe}$ $v_2(0) = \text{No}$ $v_2(1) = \text{Yes}$ $v_2(2) = \text{Maybe}$
X-Stance	a multilingual stance detection dataset with German, French and Italian examples. Each example x = (a, b) consists of a question *a* concerning some political issue and a comment *b*; the task is to identify whether the writer of *b* supports the subject of the question (0) or not (1).	$P_1(\mathbf{x}) = \text{“}a\text{” } \| \text{ ____. “}b\text{”}$ $P_2(\mathbf{x}) = a \| \text{ ____. } b$	define an English verbalizer v_{En} mapping 0 to “Yes” and 1 to “No” as well as a French (German) verbalizer v_{Fr} (v_{De}), replacing “Yes” and “No” with “Oui” and “Non” (“Ja” and “Nein”). We do not define an Italian verbalizer because x-stance does not contain any Italian training examples.

Results: English Datasets

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 $|\mathcal{T}|$.

Results: Comparison with SotA

Ex.	Method	Yelp	AG's	Yahoo	MNLI
$\lvert\mathcal{T}\rvert = 10$	UDA	27.3	72.6	36.7	34.7
	MixText	20.4	81.1	20.6	32.9
	PET	48.8	84.1	59.0	39.5
	iPET	**52.9**	**87.5**	**67.0**	**42.1**
$\lvert\mathcal{T}\rvert = 50$	UDA	46.6	83.0	60.2	40.8
	MixText	31.3	84.8	61.5	34.8
	PET	55.3	86.4	63.3	55.1
	iPET	**56.7**	**87.3**	**66.4**	**56.3**

Table 2: Comparison of PET with two state-of-the-art semi-supervised methods using RoBERTa (base)

Results: X-Stance

Examples	Method	De	Fr	It
$\|\mathcal{T}\| = 1000$	supervised	43.3	49.5	41.0
	PET	**66.4**	**68.7**	**64.7**
$\|\mathcal{T}\| = 2000$	supervised	57.4	62.1	52.8
	PET	**69.5**	**71.7**	**67.3**
$\|\mathcal{T}\| = 4000$	supervised	63.2	66.7	58.7
	PET	**71.7**	**74.0**	**69.5**
$\mathcal{T}_{\text{De}}$, $\mathcal{T}_{\text{Fr}}$	supervised	76.6	76.0	71.0
	PET	**77.9**	**79.0**	**73.6**
$\mathcal{T}_{\text{De}} + \mathcal{T}_{\text{Fr}}$	sup. (*)	76.8	76.7	70.2
	supervised	77.6	79.1	75.9
	PET	**78.8**	**80.6**	**77.2**

Table 3: Results on x-stance intra-target for XLM-R (base) trained on subsets of $\mathcal{T}_{\text{De}}$ and $\mathcal{T}_{\text{Fr}}$ and for joint training on all data ($\mathcal{T}_{\text{De}} + \mathcal{T}_{\text{Fr}}$). (*): Best results for mBERT reported in Vamvas and Sennrich (2020).

- $|T_{Fr}|$ = 11 790, $|T_{De}|$ = 33 850
- $|T_{Fr} + T_{De}|$ = 45 640
- Reported are the macro-average of the F1 scores for labels 0 and 1, averaged over three runs

04

Analysis

PET's ability to cope with situations where some PVPs perform much worse than others

Method	Yelp	AG's	Yahoo	MNLI
min	39.6	82.1	50.2	36.4
max	52.4	85.0	63.6	40.2
PET (no distillation)	51.7	87.0	62.8	40.6
PET uniform	52.7	87.3	**63.8**	**42.0**
PET weighted	**52.9**	**87.5**	**63.8**	41.8

Table 4: Minimum (min) and maximum (max) accuracy of models based on individual PVPs as well as PET with and without knowledge distillation ($|\mathcal{T}| = 10$).

- uniform : set weighting terms for all PVPs to 1
- weighted: set weighting terms to be the accuracy obtained using p on the training set before training
- -> no clear difference between the uniform and weighted variants of PET
- Distillation brings consistent improvements over the ensemble

Auxiliary Language Modeling

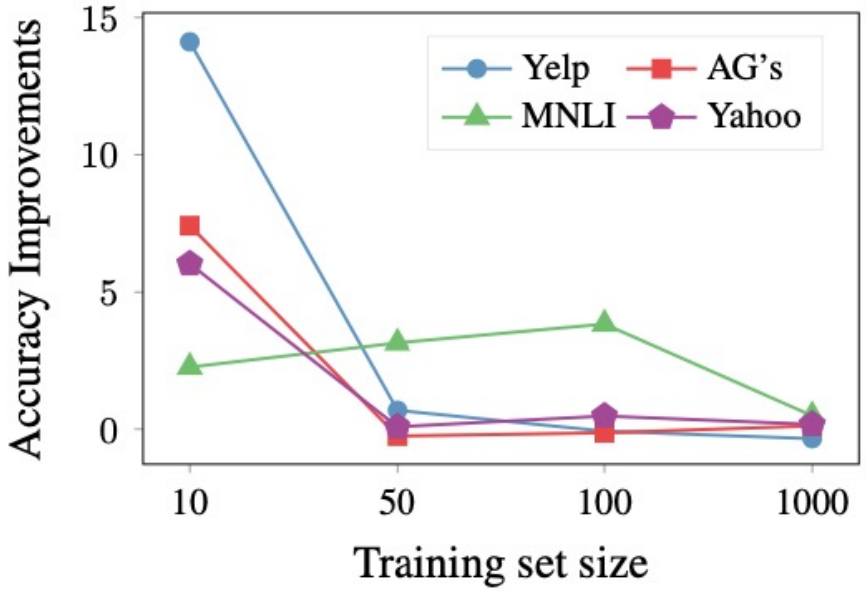


Figure 3: Accuracy improvements for PET due to adding L_{MLM} during training

- $L = (1-\alpha)\cdot L_{CE} + \alpha\cdot L_{MLM}$
  L : final loss
  L_{CE} : cross-entropy loss
  L_{MLM} : language modeling loss

- auxiliary task is extremely valuable when training on just 10 examples. With more data, it becomes less important, sometimes even leading to worse performance. Only for MNLI, we find language modeling to consistently help

iPET's ability to improve models over multiple generations

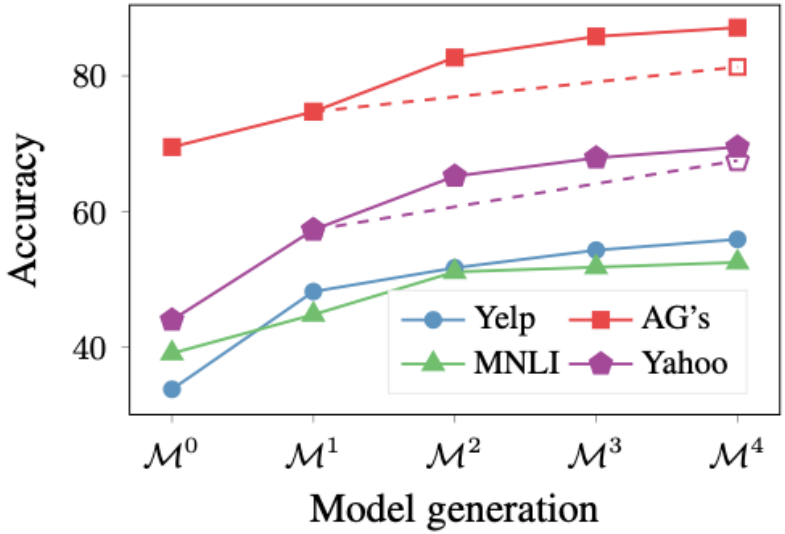


Figure 4: Average accuracy for each generation of models with iPET in a zero-shot setting. Accuracy on AG's News and Yahoo when skipping generation 2 and 3 is indicated through dashed lines.

- whether similar results can be obtained with fewer iterations by increasing the training set size more aggressively
  -> skip generations 2 and 3
  -> leads to worse performance, highlighting the importance of only gradually increasing the training set size

In-Domain Pretraining

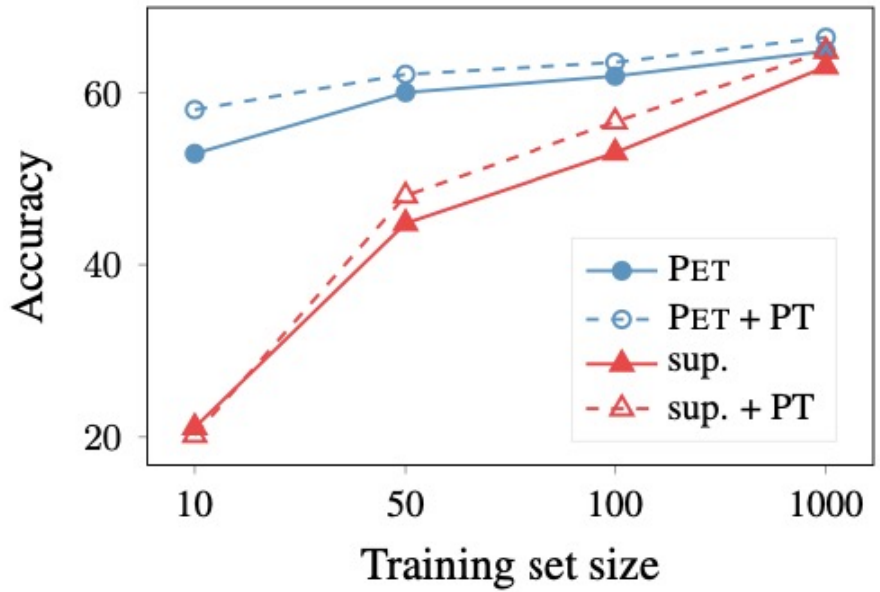


Figure 5: Accuracy of supervised learning (sup.) and PET both with and without pretraining (PT) on Yelp

- pretraining does indeed improve accuracy for supervised training
- the supervised model still clearly performs worse than PET
- in-domain pretraining is also helpful for PET, indicating that PET leverages unlabeled data in a way that is clearly different from standard masked language model pretraining.

05

Conclusion

Conclusion

- providing task descriptions to pretrained language models can be combined with standard supervised training
- When the initial amount of training data is limited, PET gives large improvements over standard supervised training and strong semi-supervised approaches.

THANKS!

DO YOU HAVE ANY QUESTIONS?

REFERENCES

- Exploiting cloze questions for few shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Kyiv, Ukraine (Online). International Committee on Computational Linguistics.

- Schick T, Schütze H. It's not just size that matters: Small language models are also few-shot learners[J]. arXiv preprint arXiv:2009.07118, 2020.

- Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

Backup: PET details

Finetune a language model using a unnormalized score
Given some input x, we define the score for label l ∈ L as

$$s_{\mathbf{p}}(l \mid \mathbf{x}) = M(v(l) \mid P(\mathbf{x}))$$

and obtain a probability distribution over labels using softmax:

$$q_{\mathbf{p}}(l \mid \mathbf{x}) = \frac{e^{s_{\mathbf{p}}(l|\mathbf{x})}}{\sum_{l' \in \mathcal{L}} e^{s_{\mathbf{p}}(l'|\mathbf{x})}}$$

For the ensemble M={M_p |p∈P} of finetuned models: combine the unnormalized class scores for each example x ∈ D as

$$s_{\mathcal{M}}(l \mid \mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) \cdot s_{\mathbf{p}}(l \mid \mathbf{x})$$

where $Z = \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p})$ and the $w(\mathbf{p})$ are weighting terms for the PVPs.

Backup: generating $\mathcal{T}_i^j$ for iPET

1. We obtain $\mathcal{N} \subset \mathcal{M}^{j-1} \setminus \{M_i^{j-1}\}$ by randomly choosing $\lambda \cdot (n-1)$ models from the previous generation with $\lambda \in (0, 1]$ being a hyperparameter.

2. Using this subset, we create a labeled dataset

$$\mathcal{T}_{\mathcal{N}} = \{(\mathbf{x}, \arg\max_{l \in \mathcal{L}} s_{\mathcal{N}}(l \mid \mathbf{x})) \mid \mathbf{x} \in \mathcal{D}\}.$$

   For each $l \in \mathcal{L}$, we obtain $\mathcal{T}_{\mathcal{N}}(l) \subset \mathcal{T}_{\mathcal{N}}$ by randomly choosing $c_j(l) - c_0(l)$ examples with label l from $\mathcal{T}_{\mathcal{N}}$. To avoid training future generations on mislabeled data, we prefer examples for which the ensemble of models is confident in its prediction. The underlying intuition is that even without calibration, examples for which labels are predicted with high confidence are typically more likely to be classified correctly (Guo et al., 2017). Therefore, when drawing from $\mathcal{T}_{\mathcal{N}}$, we set the probability of each $(\mathbf{x}, y)$ proportional to $s_{\mathcal{N}}(l \mid \mathbf{x})$.

3. We define $\mathcal{T}_i^j = \mathcal{T} \cup \bigcup_{l \in \mathcal{L}} \mathcal{T}_{\mathcal{N}}(l)$. As can easily be verified, this dataset contains $c_j(l)$ examples for each $l \in \mathcal{L}$.

Backup: improved PET

- a severe limitation of PET: the verbalizer v must map each output to a *single* token, which is impossible for many tasks

Backup: Compare 2 versions of PET

	PET with Single Mask	PET with Multiple Masks
Verbalizer function	v : L → V (label → vocabulary)	v : L → V* (label → token sequences)
conditional probability distribution	$$q_{\mathbf{p}}(y \mid x) = \frac{\exp s_{\mathbf{p}}(y \mid x)}{\sum_{y' \in Y} \exp s_{\mathbf{p}}(y' \mid x)} \quad (1)$$ where $s_{\mathbf{p}}(y \mid x) = s_M^1(v(y) \mid P(x))$ is the raw score of $v(y)$ at the masked position in $P(x)$.	we set $q_{\mathbf{p}}(y \mid x) = q(v(y) \mid P^k(x))$ where $$q(t_1 \ldots t_k|\mathbf{z}) = \begin{cases} 1 & \text{if } k = 0 \\ q_M^j(t_j|\mathbf{z}) \cdot q(t'|\mathbf{z}') & \text{if } k \geq 1 \end{cases} \quad (3)$$ with $j = \arg\max_{i=1}^{k} q_M^i(t_i \mid \mathbf{z})$, $\mathbf{z}'$ is $\mathbf{z}$ except $\mathbf{z}'_j = t_j$ and $t' = t_1 \ldots t_{j-1} t_{j+1} \ldots t_k$. $$\tilde{q}_{\mathbf{p}}(y' \mid x) = \prod_{i=1}^{k} q_M^i(t_i \mid P^{l(x)}(x))$$
Traning objective	cross entropy loss	multiclass hinge loss