Few Shot Learning in NLP

Small Language Models Are Also Few-Shot Learners

Atelier Datcraft

Mindshake Time – Few shot Learning NLP

2022/02/04

Ekimetrics.

Data science for business

PARIS | LONDON | NEW YORK | HONG KONG

Ekimetrics NLP team

Gabriel Olympie
gabriel.olympie@ekimetrics.com

Nicolas Chesneau
nicolas.chesneau@ekimetrics.com

Bertrand De Vericourt
bertrand.devericourt@ekimetrics.com

Samuel Chaineau
samuel.chaineau@ekimetrics.com

Audrey Poinsot
audrey.poinsot@ekimetrics.com

Agenda.

Few Shot Learning in NLP

Small Language Models Are Also Few-Shot Learners

PET a semi-supervised training procedure for FSL

1. What is it ?
2. How to do it ?
3. Why is it interesting ?
4. Appendix, going into deeper details

What is “Pattern Exploiting Training” (PET) ?

Few shot learning, a quick reminder

Unsupervised
Learn without labels

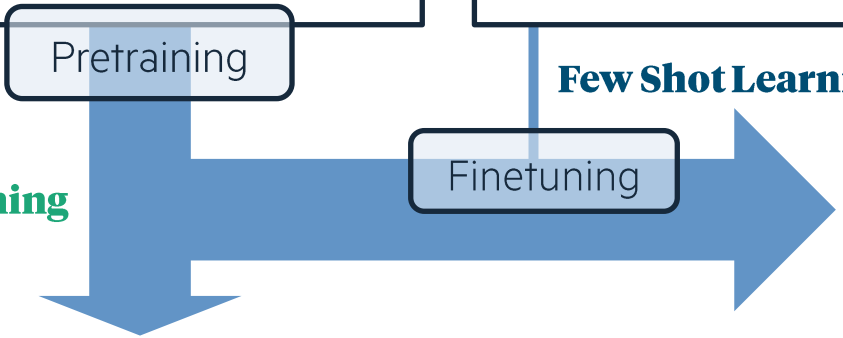
Clustering, Dimension reduction, Association, Vectorization

Supervised
Learn with labels

Classification, Regression, Generation

Semi supervised
Learn with soft-labels

Classification, Regression, Generation



Few Shot Learning

Zero Shot Learning

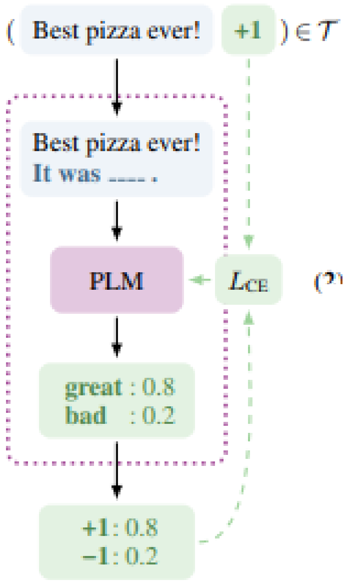
Finetuned Model

For a specific task and domain

Foundation Model

Huge Foundation models are general enough to treat a task without finetuning

PET: a promising solution for industries thanks to "prompting"



A Pattern Verbalizer Pair
to finetune
a Pretrained Language Model

Yelp For the Yelp Reviews Full Star dataset (Zhang et al., 2015), the task is to estimate the rating that a customer gave to a restaurant on a 1- to 5-star scale based on their review's text. We define the following patterns for an input text a:

$$P_1(a) = \text{It was \_\_\_\_. } a \qquad P_2(a) = \text{Just \_\_\_\_! } \| \ a$$

$$P_3(a) = a. \text{ All in all, it was \_\_\_\_.}$$

$$P_4(a) = a \ \| \text{ In summary, the restaurant is \_\_\_\_.}$$

We define a single verbalizer v for all patterns as

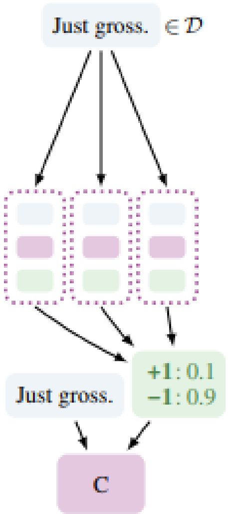
$$v(1) = \text{terrible} \quad v(2) = \text{bad} \quad v(3) = \text{okay}$$
$$v(4) = \text{good} \quad v(5) = \text{great}$$

Papers : 2009.07118.pdf (arxiv.org) and 2001.07676.pdf (arxiv.org)

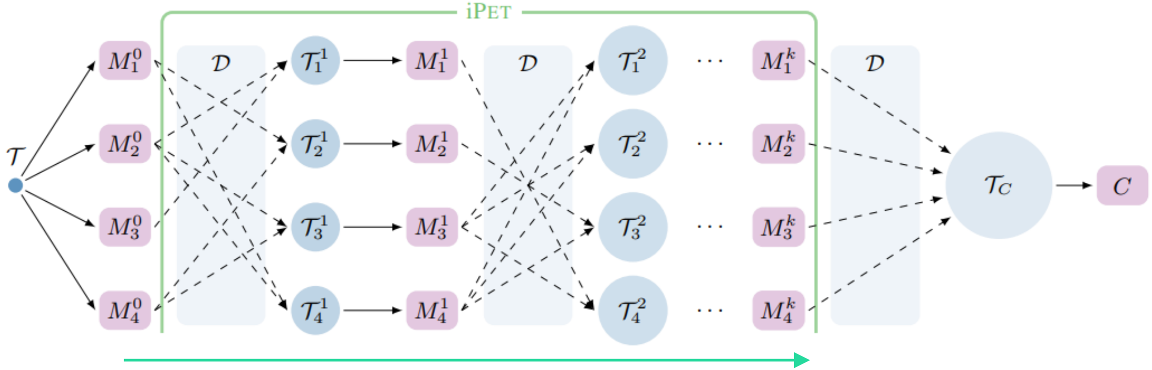
How to efficiently perform "Pattern Exploiting Training" ?

iPET, an efficient distillation method

"A set of PVPs to iteratively finetune a set of Pretrained Language Models to softly label unlabeled data to train a final classifier. "



Soft labelling with the set of finetuned PLMs



Iterations

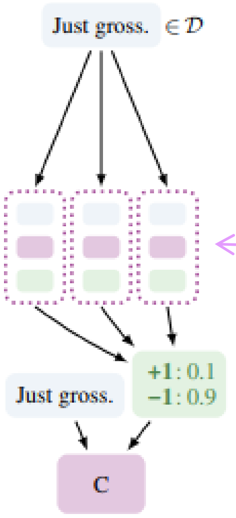
Increasing number of softly labeled data

Greater finetuning of the PLMs

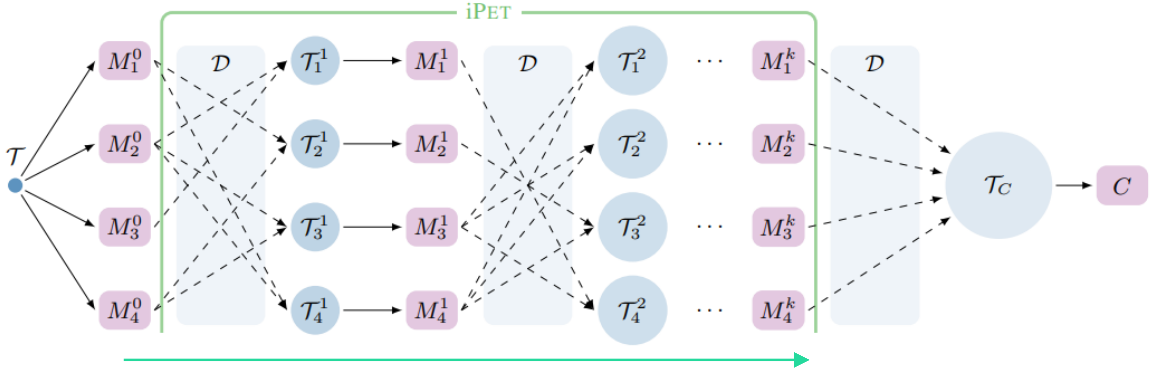
Papers : [2009.07118.pdf (arxiv.org)] and [2001.07676.pdf (arxiv.org)]

iPET, an efficient distillation method

"A set of PVPs to iteratively finetune a set of Pretrained Language Models to softly label unlabeled data to train a final classifier. "



Soft labelling with the set of finetuned PLMs

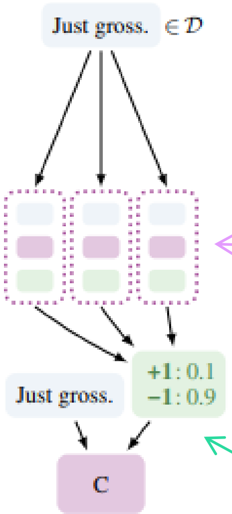


Iterations
Increasing number of softly labeled data
Greater finetuning of the PLMs

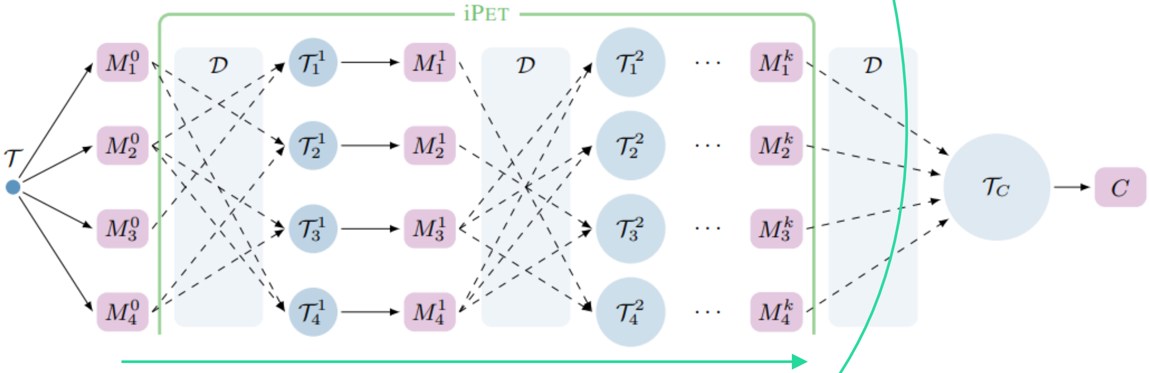
Papers : [2009.07118.pdf (arxiv.org)] and [2001.07676.pdf (arxiv.org)]

iPET, an efficient distillation method

"A set of PVPs to iteratively finetune a set of Pretrained Language Models to softly label unlabeled data to train a final classifier. "



Soft labelling with the set of finetuned PLMs

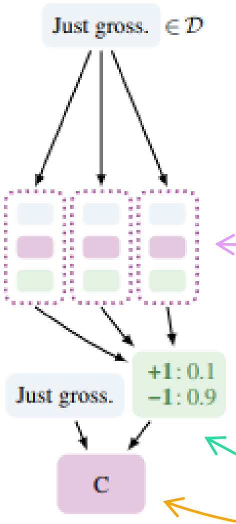


Iterations
Increasing number of softly labeled data
Greater finetuning of the PLMs

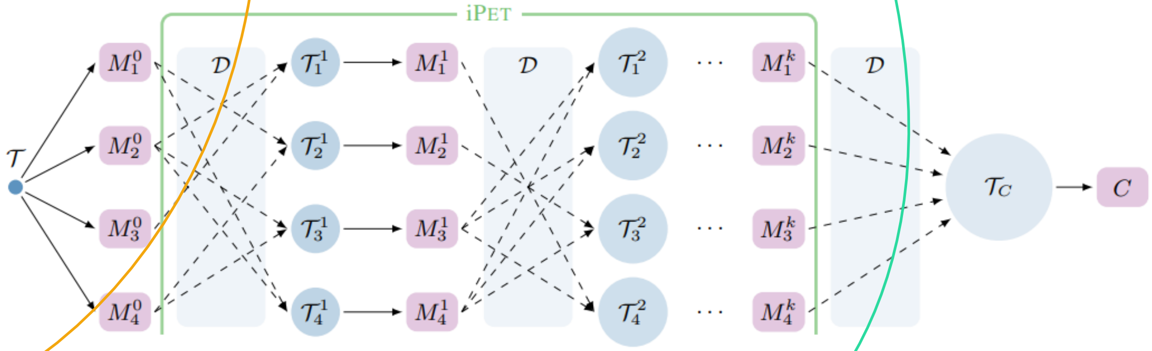
Papers : 2009.07118.pdf (arxiv.org) and 2001.07676.pdf (arxiv.org)

iPET, an efficient distillation method

"A set of PVPs to iteratively finetune a set of Pretrained Language Models to softly label unlabeled data to train a final classifier. "



Soft labelling with the set of finetuned PLMs



Iterations

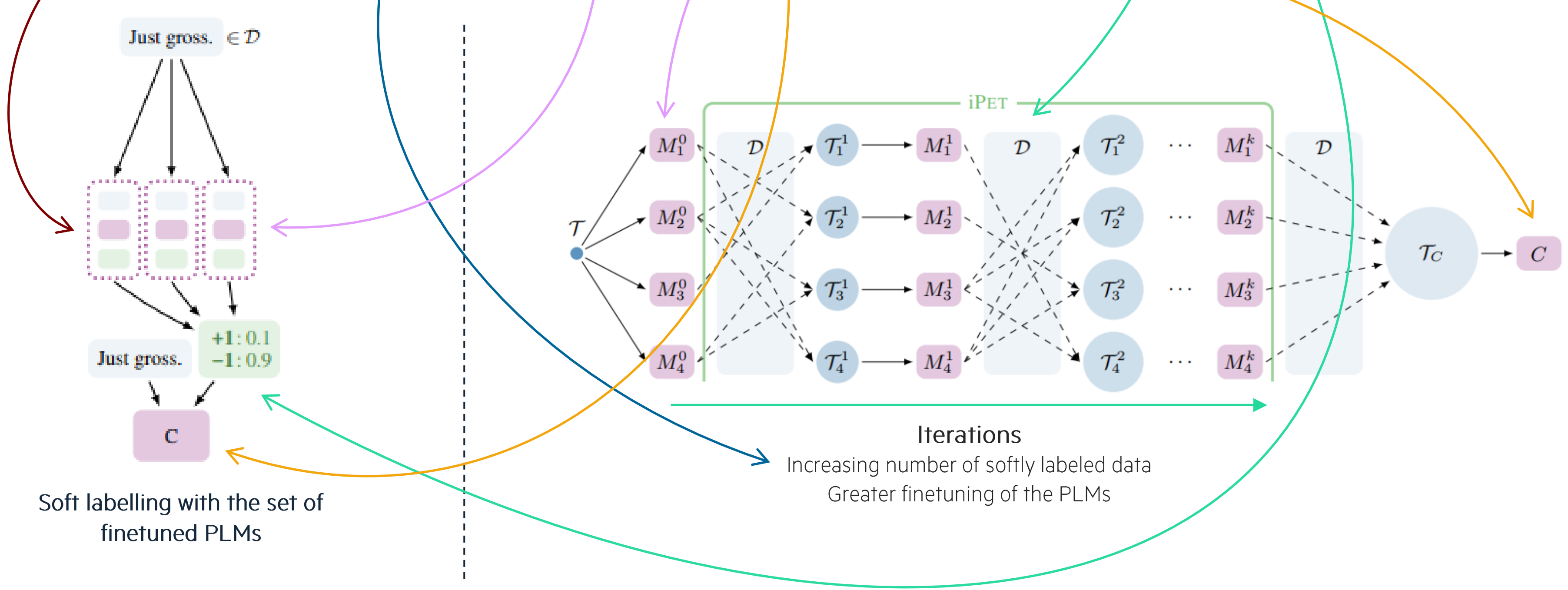
Increasing number of softly labeled data

Greater finetuning of the PLMs

Papers : 2009.07118.pdf (arxiv.org) and 2001.07676.pdf (arxiv.org)

iPET, an efficient distillation method

"A set of PVPs to iteratively finetune a set of Pretrained Language Models to softly label unlabeled data to train a final classifier. "



Soft labelling with the set of finetuned PLMs

Papers : 2009.07118.pdf (arxiv.org) and 2001.07676.pdf (arxiv.org)

Ekimetrics.

Why using “Pattern Exploiting Training” ?

Several use cases : sentiment analysis, classification, QA, logic

AG's News AG's News is a news classification dataset, where 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). For $\mathbf{x} = (a, b)$, we define the following patterns:

$$P_1(\mathbf{x}) = \_\_\_\_ : a\ b \qquad P_2(\mathbf{x}) = a\ (\ \_\_\_\_\ )\ b$$
$$P_3(\mathbf{x}) = \_\_\_\_ - a\ b \qquad P_4(\mathbf{x}) = a\ b\ (\ \_\_\_\_\ )$$
$$P_5(\mathbf{x}) = \_\_\_\_ \text{ News: } a\ b$$
$$P_6(\mathbf{x}) = [\ \text{Category: } \_\_\_\_\ ]\ a\ b$$

We use a verbalizer that maps 1–4 to "World", "Sports", "Business" and "Tech", respectively.

BoolQ (Clark et al., 2019) is a QA task where each example consists of a passage p and a yes/no question q. We use the following patterns:

- p. Question: q? Answer: __.
- p. Based on the previous passage, q? __.
- Based on the following passage, q? __. p

MultiRC (Khashabi et al., 2018) is a QA task. Given a passage p, a question q and an answer candidate a, the task is to decide whether a is a correct answer for q. We use the same verbalizer as for BoolQ and similar patterns:

- p. Question: q? Is it a? __.
- p. Question: q? Is the correct answer "a"? __.
- p. Based on the previous passage, q? Is "a" a correct answer? __.

MNLI The MNLI dataset (Williams et al., 2018) consists of text pairs $\mathbf{x} = (a, b)$. The task is to find out whether a implies b (0), a and b contradict each other (1) or neither (2). We define

$$P_1(\mathbf{x}) = \text{"}a\text{"?} \parallel \_\_\_\_, \text{"}b\text{"} \qquad P_2(\mathbf{x}) = a? \parallel \_\_\_\_, b$$

and consider two different verbalizers v_1 and v_2:

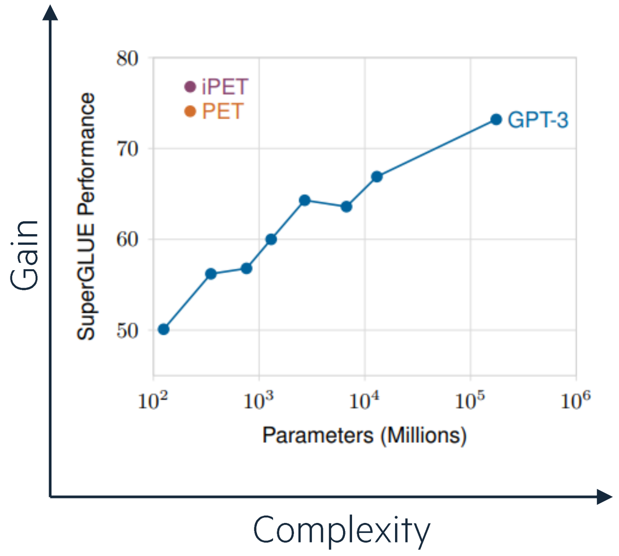
$$v_1(0) = \text{Wrong} \quad v_1(1) = \text{Right} \quad v_1(2) = \text{Maybe}$$
$$v_2(0) = \text{No} \quad v_2(1) = \text{Yes} \quad v_2(2) = \text{Maybe}$$

Combining the two patterns with the two verbalizers results in a total of 4 PVPs.

Papers : 2009.07118.pdf (arxiv.org) and 2001.07676.pdf (arxiv.org)

Ekimetrics.

iPET has on average better performances than GPT-3 with fewer parameters



	Model	Params (M)	BoolQ Acc.	CB Acc. / F1	COPA Acc.	RTE Acc.	WiC Acc.	WSC Acc.	MultiRC EM / F1a	ReCoRD Acc. / F1	Avg –
dev	GPT-3 Small	125	43.1	42.9 / 26.1	67.0	52.3	49.8	58.7	6.1 / 45.0	69.8 / 70.7	50.1
	GPT-3 Med	350	60.6	58.9 / 40.4	64.0	48.4	55.0	60.6	11.8 / 55.9	77.2 / 77.9	56.2
	GPT-3 Large	760	62.0	53.6 / 32.6	72.0	46.9	53.0	54.8	16.8 / 64.2	81.3 / 82.1	56.8
	GPT-3 XL	1,300	64.1	69.6 / 48.3	77.0	50.9	53.0	49.0	20.8 / 65.4	83.1 / 84.0	60.0
	GPT-3 2.7B	2,700	70.3	67.9 / 45.7	83.0	56.3	51.6	62.5	24.7 / 69.5	86.6 / 87.5	64.3
	GPT-3 6.7B	6,700	70.0	60.7 / 44.6	83.0	49.5	53.1	67.3	23.8 / 66.4	87.9 / 88.8	63.6
	GPT-3 13B	13,000	70.2	66.1 / 46.0	86.0	60.6	51.1	75.0	25.0 / 69.3	88.9 / 89.8	66.9
	GPT-3	175,000	77.5	82.1 / 57.2	92.0	72.9	**55.3**	75.0	32.5 / 74.8	**89.0 / 90.1**	73.2
	PET	223	79.4	85.1 / 59.4	**95.0**	69.8	52.4	**80.1**	**37.9 / 77.3**	86.0 / 86.5	74.1
	iPET	223	**80.6**	**92.9 / 92.4**	**95.0**	**74.0**	52.2	**80.1**	33.0 / 74.0	86.0 / 86.5	**76.8**
test	GPT-3	175,000	76.4	75.6 / 52.0	**92.0**	69.0	49.4	80.1	30.5 / 75.4	**90.2 / 91.1**	71.8
	PET	223	79.1	87.2 / 60.2	90.8	67.2	**50.7**	**88.4**	**36.4 / 76.6**	85.4 / 85.9	74.0
	iPET	223	**81.2**	**88.8 / 79.9**	90.8	**70.8**	49.3	**88.4**	31.7 / 74.1	85.4 / 85.9	**75.4**
	SotA	11,000	*91.2*	*93.9 / 96.8*	*94.8*	*92.5*	*76.9*	*93.8*	*88.1 / 63.3*	*94.1 / 93.4*	*89.3*

Table 1: Results on SuperGLUE for GPT-3 primed with 32 randomly selected examples and for PET / iPET with ALBERT-xxlarge-v2 after training on FewGLUE. State-of-the-art results when using the regular, full size training sets for all tasks (Raffel et al., 2019) are shown in italics.

Thank you !

Appendix

iPET: specificities

3.2 Auxiliary Language Modeling

In our application scenario, only a few training examples are available and catastrophic forgetting can occur. As a PLM finetuned for some PVP is still a language model at its core, we address this by using language modeling as auxiliary task. With L_{CE} denoting cross-entropy loss and L_{MLM} language modeling loss, we compute the final loss as

$$L = (1 - \alpha) \cdot L_{\text{CE}} + \alpha \cdot L_{\text{MLM}}$$

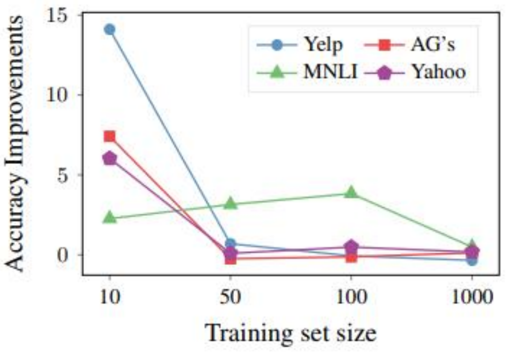


Figure 3: Accuracy improvements for PET due to adding L_{MLM} during training

E Automatic Verbalizer Search

Given a set of patterns $P_1, \ldots, P_n$, manually finding a verbalization $v(l)$ for each $l \in \mathcal{L}$ that represents the meaning of l well and corresponds to a single token in V can be difficult. We therefore devise *automatic verbalizer search* (AVS), a procedure that automatically finds suitable verbalizers given a training set $\mathcal{T}$ and a language model M.

	Yelp	AG's	Yahoo	MNLI
supervised	44.8	82.1	52.5	45.6
PET	**60.0**	**86.3**	**66.2**	**63.9**
PET + AVS	55.2	85.0	58.2	52.6

Table 7: Results for supervised learning, PET and PET with AVS (PET + AVS) after training on 50 examples

y	Top Verbalizers
1	worthless, BAD, useless, appalling
2	worse, slow, frustrating, annoying
3	edible, mixed, cute, tasty, Okay
4	marvelous, loved, love, divine, fab
5	golden, magical, marvelous, perfection

Table 8: Most probable verbalizers according to AVS for Yelp with 50 training examples

3.1 PET with Multiple Masks

An important limitation of PET is that the verbalizer v must map each output to a *single* token, which is impossible for many tasks. We thus generalize verbalizers to functions $v : Y \to T^*$; this requires some modifications to inference and training.[3]

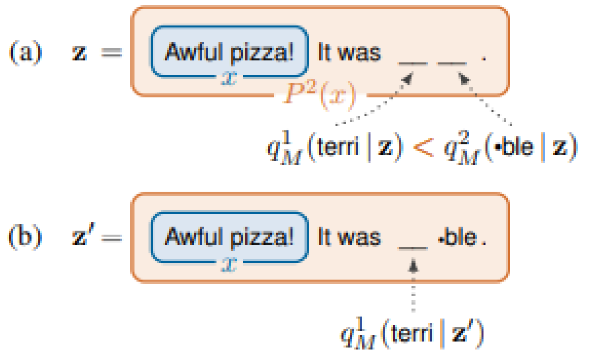


Figure 3: Inference for a verbalization consisting of the two tokens terri and •ble. (a) We first compute the probability of each token at its position in the cloze question $P^2(x)$ and identify the token with the highest probability. (b) We insert this token into the cloze question and compute the probability of the remaining token.

Practically, how to build the right Patterns

Abstract

Recent work has presented intriguing results examining the knowledge contained in language models (LM) by having the LM fill in the blanks of prompts such as "*Obama is a _ by profession*". These prompts are usually manually created, and quite possibly sub-optimal; another prompt such as "*Obama worked as a _*" may result in more accurately predicting the correct profession. Because of this, given an inappropriate prompt, we might fail to retrieve facts that the LM *does* know, and thus any given prompt only provides a lower bound estimate of the knowledge contained in an LM. In this paper, we attempt to more accurately estimate the knowledge contained in LMs by automatically discovering better prompts to use in this querying process. Specifically, we propose mining-based and paraphrasing-based methods to automatically generate high-quality and diverse prompts, as well as ensemble methods to combine answers from different prompts. Extensive experiments on the LAMA benchmark for extracting relational knowledge from LMs demonstrate that our methods can improve accuracy from 31.1% to 39.6%, providing a tighter lower bound on what LMs know. We have released the code and the resulting LM Prompt And Query Archive (LPAQA) at `https://github.com/jzbjyb/LPAQA`.

	Prompts
manual	`DirectX` *is developed by* y_{man}
mined	y_{mine} *released the* `DirectX`
paraphrased	`DirectX` *is created by* y_{para}

Top 5 predictions and log probabilities

	y_{man}		y_{mine}		y_{para}	
1	`Intel`	-1.06	`Microsoft`	-1.77	`Microsoft`	-2.23
2	`Microsoft`	-2.21	`They`	-2.43	`Intel`	-2.30
3	`IBM`	-2.76	`It`	-2.80	`default`	-2.96
4	`Google`	-3.40	`Sega`	-3.01	`Apple`	-3.44
5	`Nokia`	-3.58	`Sony`	-3.19	`Google`	-3.45

Figure 1: Top-5 predictions and their log probabilities using different prompts (manual, mined, and paraphrased) to query BERT. Correct answer is underlined.

ID	Modifications	Acc. Gain
P413	x plays in→at y position	+23.2
P495	x was created→made in y	+10.8
P495	x was→is created in y	+10.0
P361	x is a part of y	+2.7
P413	x plays ~~in~~ y position	+2.2

Table 6: Small modifications (update, insert, and delete) in paraphrase lead to large accuracy gain (%).

ID	Relations	Manual Prompts	Mined Prompts	Acc. Gain
P140	religion	x is affiliated with the y religion	x who converted to y	+60.0
P159	headquarters location	The headquarter of x is in y	x is based in y	+4.9
P20	place of death	x died in y	x died at his home in y	+4.6
P264	record label	x is represented by music label y	x recorded for y	+17.2
P279	subclass of	x is a subclass of y	x is a type of y	+22.7
P39	position held	x has the position of y	x is elected y	+7.9

Table 4: Micro-averaged accuracy gain (%) of the mined prompts over the manual prompts.

8 Conclusion

In this paper, we examined the importance of the prompts used in retrieving factual knowledge from language models. We propose mining-based and paraphrasing-based methods to systematically generate diverse prompts to query specific pieces of relational knowledge. Those prompts, when combined together, improve factual knowledge retrieval accuracy by 8%, outperforming manually designed prompts by a large margin. Our analysis indicates that LMs are indeed more knowledgeable than initially indicated by previous results, but they are also quite sensitive to how we query them. This indicates potential future directions such as (1) more robust LMs that can be queried in different ways but still return similar results, (2) methods to incorporate factual knowledge in LMs, and (3) further improvements in optimizing methods to query LMs for knowledge. Finally, we have released all our learned prompts to the community as the LM Prompt and Query Archive (LPAQA), available at: `https://github.com/jzbjyb/LPAQA`.

Paper : 1911.12543.pdf (arxiv.org)