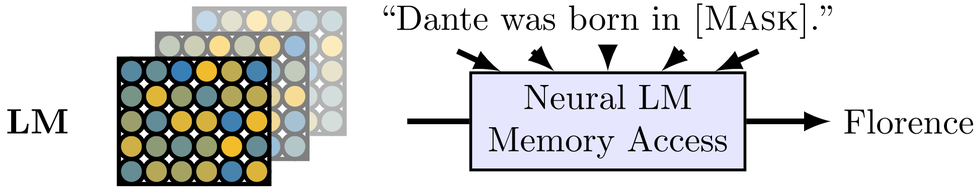
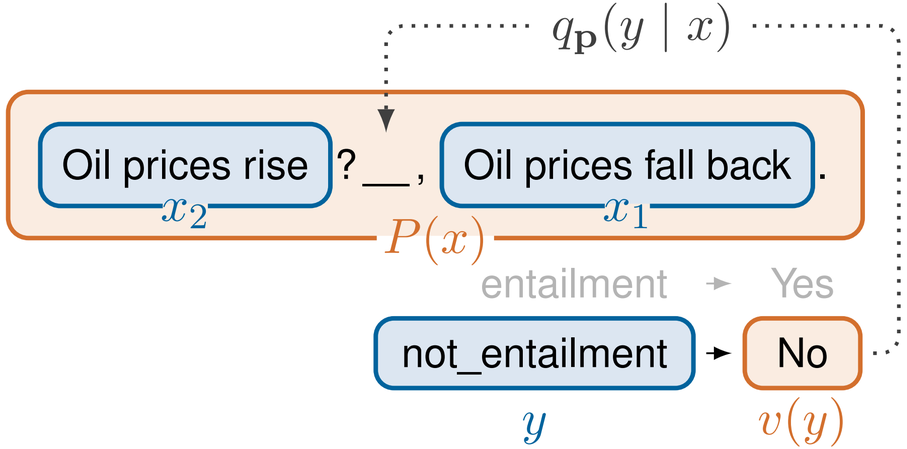
From prompt probing to prompt fine-tuning





Speaker: AntNLP(@Yijun Wang)

Development

- ☆ Prompt based Probing:
  - Language Models as Knowledge Bases? (EMNLP2019), Inducing Relational Knowledge from BERT (AAAI2020)
- ☆ Prompt Tuning:
  - LPAQA (TACL2020), AUTOPROMPT (EMNLP2020)
  - Prefix-Tuning, P-tuning, Soft prompts (NAACL2021), OPTIPROMPT (NAACL2021)
- ☆ Rethink:
  - Knowledgeable or Educated Guess (ACL2021), Quantify (NAACL2021), OPTIPROMPT (NAACL2021)
- ☆ Prompt based fine-tuning:
  - PET (EACL2021), PET2 (NAACL2021), Few-shot (ACL2021), Domain adaptation, Relation Extraction

Prompt Tuning: Classification of Prompts

- Manual: LAMA
- Data-driven
  - Discrete: LPAQA, AUTOPROMPT
  - Continuous: Prefix-Tuning, P-tuning, Soft prompts, OPTIPROMPT

Examples

Method	Prompt	Data-driven?
LAMA (Petroni et al., 2019)	`[X]` is `[MASK]` citizen	✗
LPAQA (Jiang et al., 2020)	`[X]` is a citizen of `[MASK]`	✓
AUTOPROMPT (Shin et al., 2020)	`[X]` m^3 badminton pieces internationally representing `[MASK]`	✓
OPTIPROMPT	`[X]` $[V]_1$ $[V]_2$ $[V]_3$ $[V]_4$ $[V]_5$ `[MASK]`	✓
OPTIPROMPT (manual)	`[X]` $[V]_1 :=$ is `[MASK]` $[V]_2 :=$ citizen	✓

Table 1: Comparison of prompts for the relation *country of citizenship*. `[X]` denotes the name of the subject and `[MASK]` is single-token object label to be predicted. In our OPTIPROMPT approach, we optimize a sequence of learned embeddings $[V]_i \in \mathbb{R}^d$ for each relation type. $[V]_i := w$ indicates that the vector is learned but initialized by the pre-trained embedding of word w and OPTIPROMPT (manual) indicates that we use a manual prompt as initialization (see Section 3 for more details).

Soft Prompts: _____x v_1 v_2 v_3 v_4 v_5 _____y v_6

Find better prompts for probing knowledge

Results

Method	1-1	N-1	N-M	All	UHN
Majority	1.8	23.9	22.0	22.0	23.8
LAMA (manual)	**68.0**	32.4	24.7	31.1	21.8
LPAQA (manual + paraphrased)	65.0	35.9	27.9	34.1	28.7
AUTOPROMPT (5 `[T]`s)	58.0	46.5	34.0	42.2	31.3
OPTIPROMPT (5 `[V]`s)	49.6	53.1	39.4	47.6	37.5
OPTIPROMPT (10 `[V]`s)	60.7	53.2	39.2	48.1	37.9
OPTIPROMPT (manual)	59.6	**54.1**	**40.1**	**48.6**	**38.4**

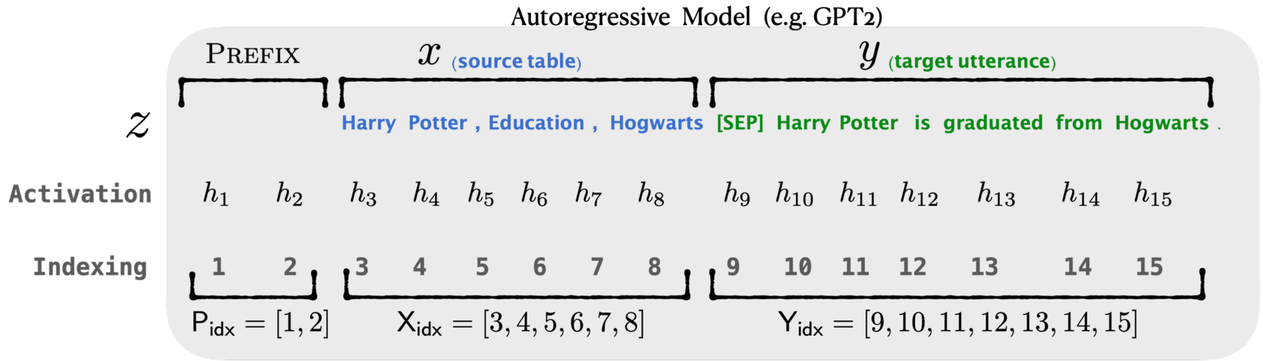
Table 2: Micro-averaged results (top-1) on the LAMA benchmark using the BERT-base-cased model, averaged over relations. UHN stands for UnHelpfulNames (Poerner et al., 2019), which is a subset of LAMA where questions with helpful entity names were deleted. The LAMA results are broken down by relation category. Examples from each category are *capital of* (1-1), *place of birth* (N-1), and *shares border with* (N-M).

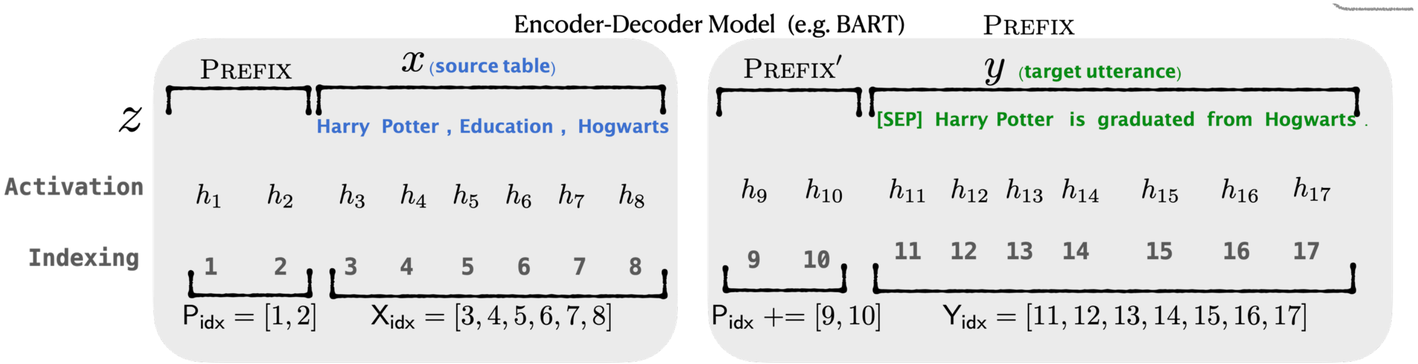
Prompt type	Model	P@1
Original (MP)	BERT-base	31.1
	BERT-large	32.3
	E-BERT	36.2
Discrete	LPAQA (BERT-base)	34.1
	LPAQA (BERT-large)	39.4
	AutoPrompt (BERT-base)	43.3
P-tuning	BERT-base	48.3
	BERT-large	**50.6**

Model	T-REx orig.	T-REx ext.
LAMA (BEb)	31.1	26.4
LPAQA(BEb)	34.1	31.2
AutoPrompt	43.3	45.6
Soft (sin., BEb)	47.7 (+**16.6**$^?$)	49.6 (+**23.2**$^?$)
Soft (min., BEb)	**50.7**$^?$(+**16.6**$^?$)	**50.5**$^?$(+**19.3**$^?$)
Soft (par., BEb)	48.4 (+**12.8**$^?$)	49.7 (+**18.5**$^?$)
Soft (ran., BEb)	48.1 (+**47.4**)	50.6 (+**49.8**)
LAMA (BEl)	28.9†	24.0†
LPAQA(BEl)	39.4†	37.8†
Soft (sin., BEl)	51.1 (+**22.2**)	51.4 (+**27.4**)
Soft (min., BEl)	**51.6** (+**12.2**)	**52.5** (+**14.7**)
Soft (par., BEl)	51.1 (+**11.7**)	51.7 (+**13.9**)
Soft (ran., BEl)	51.9 (+**47.1**)	51.9 (+**50.5**)
AutoPrompt	40.0	-
Soft (min., Rob)	**40.6**$^?$(+**39.4**)	-

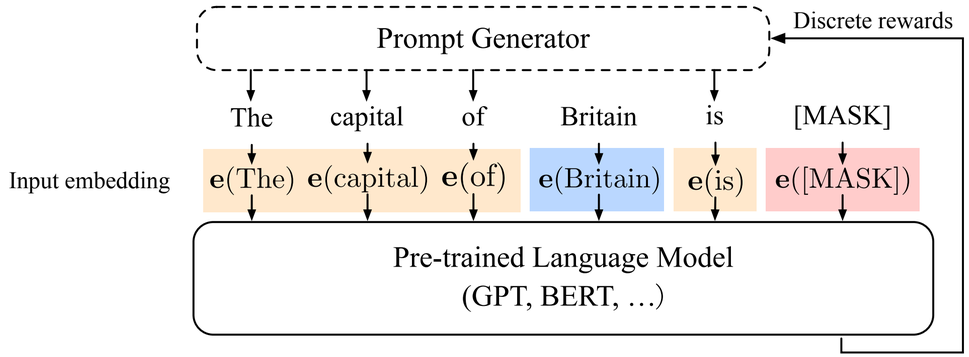
- (sin.) LAMA provides a **sin**gle manually created hard prompt for each relation type r.
- (par.) LPAQA (Jiang et al., 2020) provides a set of 13–30 hard prompts for each r, which are **par**aphrases of the LAMA prompt.[4]
- (min.) LPAQA also provides a set of 6–29 hard prompts for each r, based on text **min**ing.
- (ran.) For each (min.) prompt, we replace each word with a **ran**dom vector, drawn from a Gaussian distribution fit to all of the LM's word embeddings. The number of words and the position of the blanks are preserved.

Examples

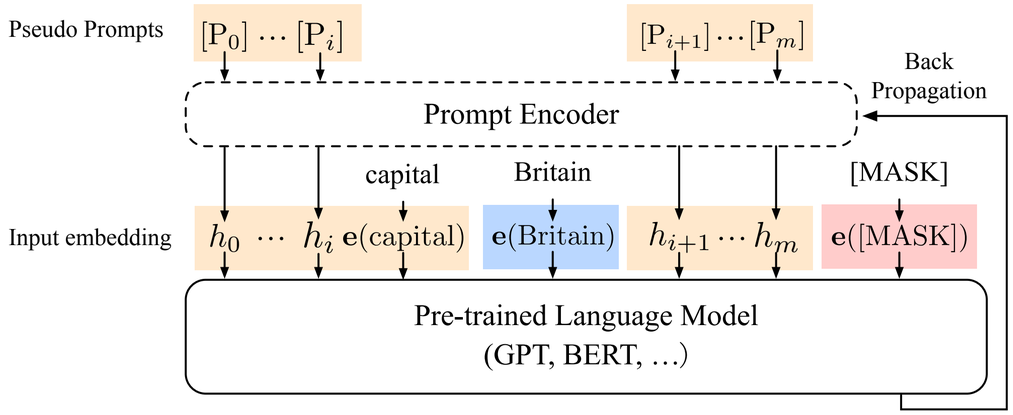




Prefix-tuning



(a) Discrete Prompt Search



(b) P-tuning

Replace fine-tuning with prompt-tuning

Results

	E2E					WebNLG									DART					
	BLEU	NIST	MET	R-L	CIDEr	BLEU			MET			TER ↓			BLEU	MET	TER ↓	Mover	BERT	BLEURT
						S	U	A	S	U	A	S	U	A						
GPT-2$_{\text{MEDIUM}}$																				
FINE-TUNE	68.2	8.62	**46.2**	71.0	2.47	**64.2**	27.7	46.5	**0.45**	0.30	0.38	**0.33**	0.76	0.53	46.2	**0.39**	**0.46**	**0.50**	**0.94**	**0.39**
FT-TOP2	68.1	8.59	46.0	70.8	2.41	53.6	18.9	36.0	0.38	0.23	0.31	0.49	0.99	0.72	41.0	0.34	0.56	0.43	0.93	0.21
ADAPTER(3%)	68.9	8.71	46.1	71.3	2.47	60.4	**48.3**	54.9	0.43	**0.38**	**0.41**	0.35	**0.45**	**0.39**	45.2	0.38	**0.46**	**0.50**	**0.94**	**0.39**
ADAPTER(0.1%)	66.3	8.41	45.0	69.8	2.40	54.5	45.1	50.2	0.39	0.36	0.38	0.40	0.46	0.43	42.4	0.36	0.48	0.47	**0.94**	0.33
PREFIX(0.1%)	**69.7**	**8.81**	46.1	**71.4**	**2.49**	62.9	45.6	**55.1**	0.44	**0.38**	**0.41**	0.35	0.49	0.41	**46.4**	0.38	**0.46**	**0.50**	**0.94**	**0.39**
GPT-2$_{\text{LARGE}}$																				
FINE-TUNE	68.5	8.78	46.0	69.9	2.45	**65.3**	43.1	55.5	**0.46**	0.38	**0.42**	**0.33**	0.53	0.42	**47.0**	**0.39**	0.46	**0.51**	**0.94**	**0.40**
Prefix	**70.3**	**8.85**	**46.2**	**71.7**	**2.47**	63.4	**47.7**	**56.3**	0.45	**0.39**	**0.42**	0.34	**0.48**	**0.40**	46.7	**0.39**	**0.45**	**0.51**	**0.94**	**0.40**
SOTA	68.6	8.70	45.3	70.8	2.37	63.9	52.8	57.1	0.46	0.41	0.44	-	-	-	-	-	-	-	-	-

Table 1: Metrics (higher is better, except for TER) for table-to-text generation on E2E (left), WebNLG (middle) and DART (right). With only 0.1% parameters, Prefix-tuning outperforms other lightweight baselines and achieves a comparable performance with fine-tuning. The best score is boldfaced for both GPT-2$_{\text{MEDIUM}}$ and GPT-2$_{\text{LARGE}}$.

Method	BoolQ (Acc.)	CB (F1)	CB (Acc.)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (Acc.)	Avg.
BERT-large-cased (335M)										
Fine-tune*	77.7	94.6	93.7	74.9	75.8	24.7	70.5	68.3	69.0	72.5
MP zero-shot	49.7	50.0	34.2	50.0	49.9	0.6	6.5	61.5	58.0	45.0
MP fine-tuning	77.2	91.1	93.5	70.5	73.6	17.7	67.0	80.8	75.0	73.1
P-tuning	77.8	96.4	97.4	72.7	75.5	17.1	65.6	81.7	76.0	74.6
GPT2-medium (345M)										
Fine-tune	71.0	73.2	51.2	65.2	72.2	19.2	65.8	62.5	66.0	63.1
MP zero-shot	56.3	44.6	26.6	54.1	51.3	2.2	32.5	63.5	53.0	47.3
MP fine-tuning	78.3	96.4	97.4	70.4	72.6	32.1	74.4	73.0	80.0	74.9
P-tuning	78.9 (+1.1)	98.2 (+1.8)	98.7 (+1.3)	69.4 (-5.5)	75.5 (-0.3)	29.3 (+4.6)	74.2 (+3.7)	74.0 (-7.7)	81.0 (+5.0)	75.6 (+1.0)

* We report the same results taken from SuperGLUE (Wang et al., 2019b).

Table 4. Fully-supervised learning on SuperGLUE dev with large-scale models. MP refers to manual prompt. For fair comparison, MP zero-shot and MP fine-tuning report results of a single pattern, while anchors for P-tuning are selected from the same prompt. Subscripts in red represents improvements of GPT with P-tuning over the best results of BERT.

Knowledgeable or Educated Guess: Rethinking Masked Language Models as Factual Knowledge Bases

ACL2021

Erros

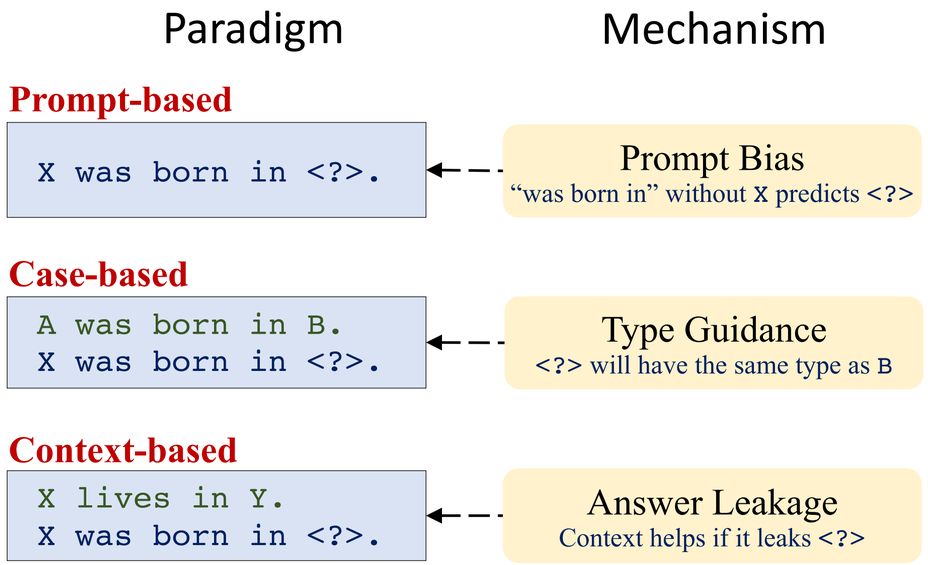
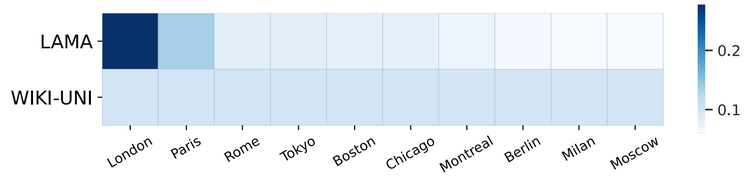


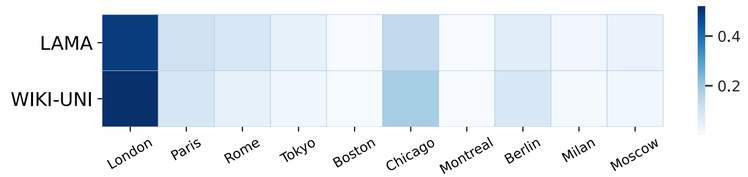
Figure 1: This paper explores three different kinds of factual knowledge extraction paradigms from MLMs, and reveal the underlying predicting mechanisms behind them.

- **Prompt-based retrieval** (Petroni et al., 2019; Jiang et al., 2020b; Shin et al., 2020), which queries MLM for object answer only given the subject and the corresponding relation prompt as input, e.g., *"Jobs was born in [MASK]."*
- **Case-based analogy** (Brown et al., 2020; Madotto et al., 2020; Gao et al., 2020), which enhances the prompt-based retrieval with several illustrative cases, e.g., *"Obama was born in Hawaii. [SEP] Jobs was born in [MASK]."*
- **Context-based inference** (Petroni et al., 2020; Bian et al., 2021), which augments the prompt-based retrieval with external relevant contexts, e.g., *"Jobs lives in California. [SEP] Jobs was born in [MASK]."*

Prompt-based Retrieval



(a) The true answer distributions are very different between LAMA and WIKI-UNI.



(b) However, the prediction distribution made by MLMs on them are still very similar.

Figure 2: An illustration example of the vastly different answer distributions but similar prediction distributions on LAMA and WIKI-UNI on "`place-of-birth`" relation.

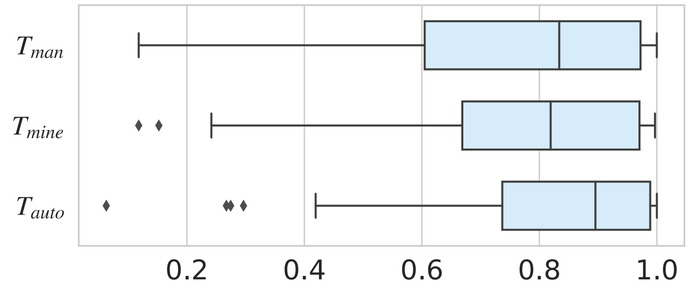


Figure 3: Correlations of the prediction distributions on LAMA and WIKI-UNI. Even these two datasets have totally different answer distributions, MLMs still make highly correlated predictions.

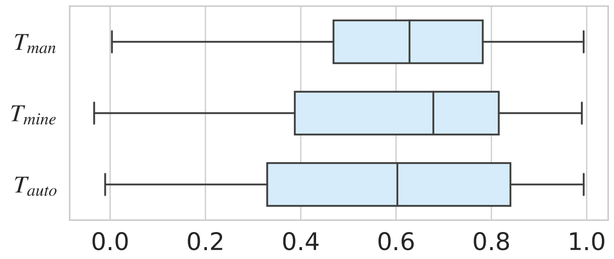


Figure 4: Correlations between the prompt-only distribution and prediction distribution on WIKI-UNI. MLMs make correlated predictions w. or w/o. subjects.

Prompt	Precision	KL divergence
T_{man}	30.36	12.27
T_{mine}	39.49	10.79
T_{auto}	40.36	10.27

Table 2: The smaller KL divergence between the prompt-only distribution and golden answer distribution of LAMA, the better performance of the prompt.

Relation	Prompt	Source	Prec.
`occupation`	x is a y by profession	T_{man}	0.63
	x is an american y	T_{mine}	18.27
`citizenship`	x is y citizen	T_{man}	0.00
	x returned to y	T_{mine}	43.58
`work location`	x used to work in y	T_{man}	11.01
	x was born in y	T_{mine}	40.25
`instance of`	x is a y	T_{man}	30.15
	x is a small y	T_{mine}	52.60

Table 3: Examples of prompts that can achieve significant improvements on LAMA. We can see that the better performance actually stems from over-fitting: the better prompts are not prompts with a stronger semantic association to the relation.

Case-based Analogy

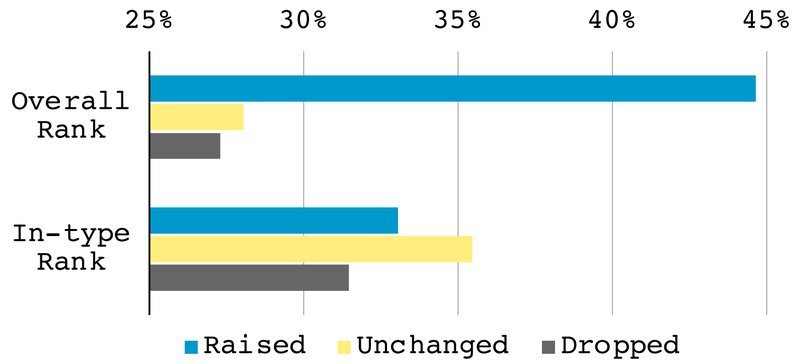


Figure 6: Percentages on the change of overall rank (among all candidates) and the in-type rank (among candidates with the same type) of golden answer. We can see that the illustrative cases mainly raise the overall rank but cannot raise the in-type rank, which means the performance improvements mainly come from better type recognition.

Relation	**Induced Object Type**	**Precision Δ**	**Type Prec. Δ**	**Wrong → Right w/ Type Change**	**Right → Wrong w/o Type Change**
country of citizenship	sovereign state	43.37	84.16	100.00	-
position held	religious servant	36.88	80.26	91.15	90.00
religion	religion	33.20	34.88	100.00	-
work location	city	26.10	70.55	85.04	100.00
instrument	musical instrument	17.07	55.75	89.08	75.00
country	sovereign state	14.30	29.04	88.48	87.93
employer	business	12.01	99.22	100.00	-
continent	continent	10.87	51.18	96.86	88.24

Table 4: Detailed analysis on relations where the mean precision increased more than 10%. Precision Δ and Type Prec. Δ represents the precision changes on the answer and the type of the answer respectively. "w/ Type Change" and "w/o Type Change" represents the type of prediction changed/unchanged before/after introducing illustrative cases. "-" indicate there is no queries whose predictions are mistakenly reversed.

Context-based Inference

Answer in context	Prompt-based	Context-based	Δ
Present (45.30%)	34.83	64.13	+29.30
Absent (54.70 %)	25.37	23.26	-2.11

Table 5: Comparison between prompt-based and context-based paradigms grouped by whether the answer presents or absents in the context. We can see that only contexts containing the answer can significantly improve the performance.

Prompt-based	Context-based	Masked Context-based
30.36	41.44	35.66

Table 6: Overall performance when introducing different kinds of contexts. "Masked Context-based" indicates that we mask the golden answer in contexts, and there is still a significant performance improvement.

Answer Reconstructable	Prompt-based	Context-based	Δ
Reconstructable (60.23%)	39.58	60.82	+21.24
Not-reconstructable (39.77 %)	28.84	35.83	+6.99

Table 7: Comparison between prompt-based and context-based paradigms grouped by whether the masked answer in the context can be reconstructed from the remaining context. We can see that contexts can reconstruct the masked answer is more likely to improve the performance.

Factual Probing Is [MASK]: Learning vs. Learning to Recall

Zexuan Zhong* **Dan Friedman*** **Danqi Chen**
Department of Computer Science
Princeton University

NAACL2021

Facts in training data

Method	1-1	N-1	N-M	All	UHN
Majority	1.8	23.9	22.0	22.0	23.8
LAMA (manual)	**68.0**	32.4	24.7	31.1	21.8
LPAQA (manual + paraphrased)	65.0	35.9	27.9	34.1	28.7
AUTOPROMPT (5 `[T]`s)	58.0	46.5	34.0	42.2	31.3
OPTIPROMPT (5 `[V]`s)	49.6	53.1	39.4	47.6	37.5
OPTIPROMPT (10 `[V]`s)	60.7	53.2	39.2	48.1	37.9
OPTIPROMPT (manual)	59.6	**54.1**	**40.1**	**48.6**	**38.4**

Table 2: Micro-averaged results (top-1) on the LAMA benchmark using the BERT-base-cased model, averaged over relations. UHN stands for UnHelpfulNames (Poerner et al., 2019), which is a subset of LAMA where questions with helpful entity names were deleted. The LAMA results are broken down by relation category. Examples from each category are *capital of* (1-1), *place of birth* (N-1), and *shares border with* (N-M).

Relation	Class Prior	Naive Bayes
All	17.3	24.6
1-1	0.2	0.3
N-1	23.2	28.6
N-M	11.0	21.8
member of	2.2	59.6
manufacturer	8.9	62.0

Table 3: Results for simple classifiers fit to the Wikidata training data and evaluated on the LAMA test set. We highlight two relations for which object labels are correlated with particular subject tokens: In the *member of* category, the model appears to learn that any subject with "football" in its name, such as *Ghana Football Association*, is likely to be a member of *FIFA*. In the *manufacturer* category, the model learns to predict that *Chevrolet* manufactures the *Chevrolet Impala*, *BMW* manufactures the *BMW M Coupe*, and so on.

Control Task

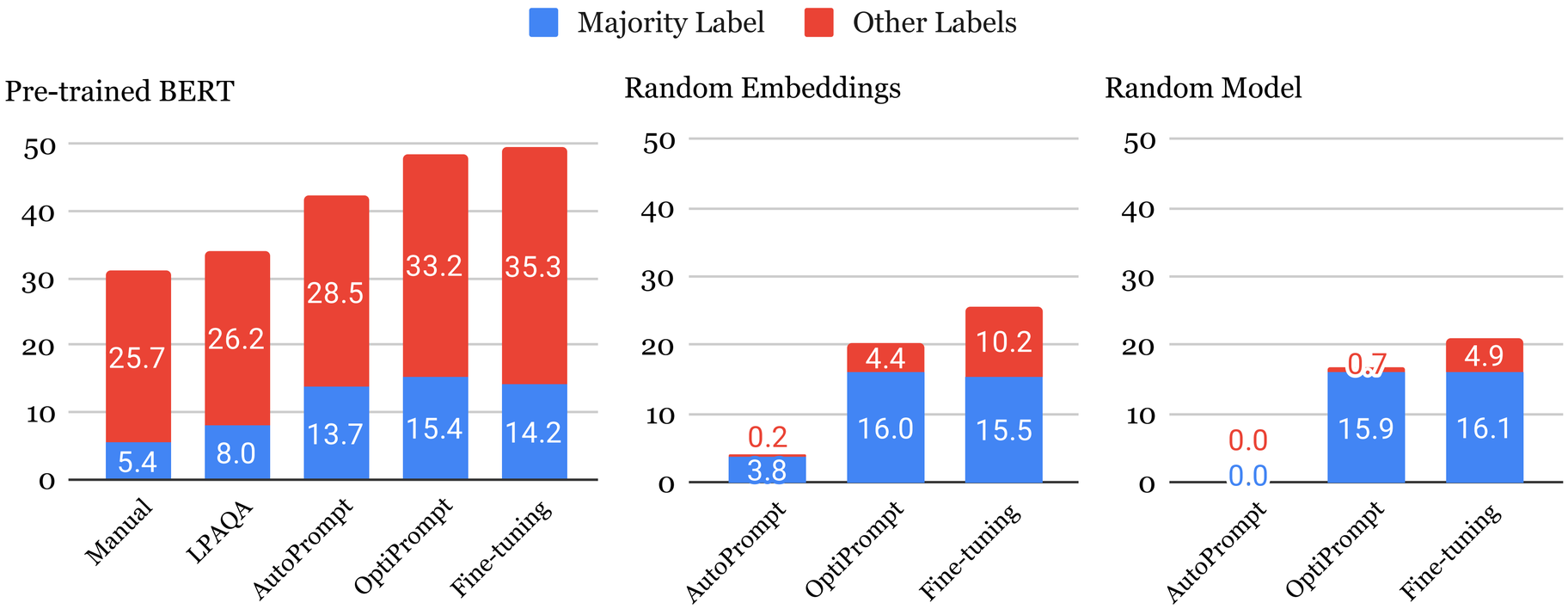


Figure 2: Accuracy on LAMA obtained by prompting BERT-base-cased, either the pre-trained model, reinitializing the input embeddings, or reinitializing all parameters. Each bar represents total accuracy micro-averaged over relations and divided into two categories: accuracy obtained by predicting the training set majority class label, and accuracy obtained by predicting other object labels. We also fine-tune BERT, which, in the random control settings, can be thought of as a better lower bound on the entropy of the task distribution.

Control Task

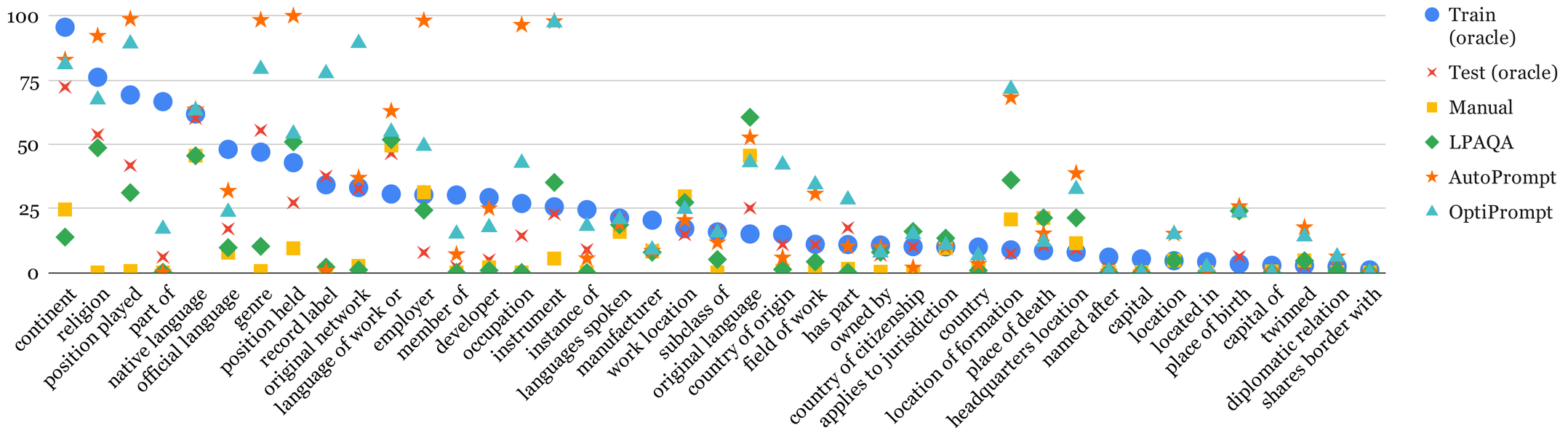


Figure 3: The percentage of LAMA examples for which a prompt elicits the training set majority label, compared with the percentage of training and test facts with that label. Optimized prompts show a strong tendency to over-predict the majority class relative to manual prompts and the ground truth. "Train (oracle)" is calculated from the set of Wikidata facts collected by Shin et al. (2020), which is used to train AUTOPROMPT and OPTIPROMPT.

How Many Data Points is a Prompt Worth?

Teven Le Scao
Hugging Face
teven@huggingface.co

Alexander M. Rush
Hugging Face
sasha@huggingface.co

NAACL2021

Comparison: Heads vs Prompts

- Head-based: standard fine-tuning
- Prompt-based: PET

"Posthumous marriage – Posthumous marriage (or necrogamy) is a marriage in which one of the participating members is deceased. It is legal in France and similar forms are practiced in Sudan and China. Since World War I, France has had hundreds of requests each year, of which many have been accepted. **Based on the previous passage,** ***can u marry a dead person in france ?*** **<MASK>**"

Experiments

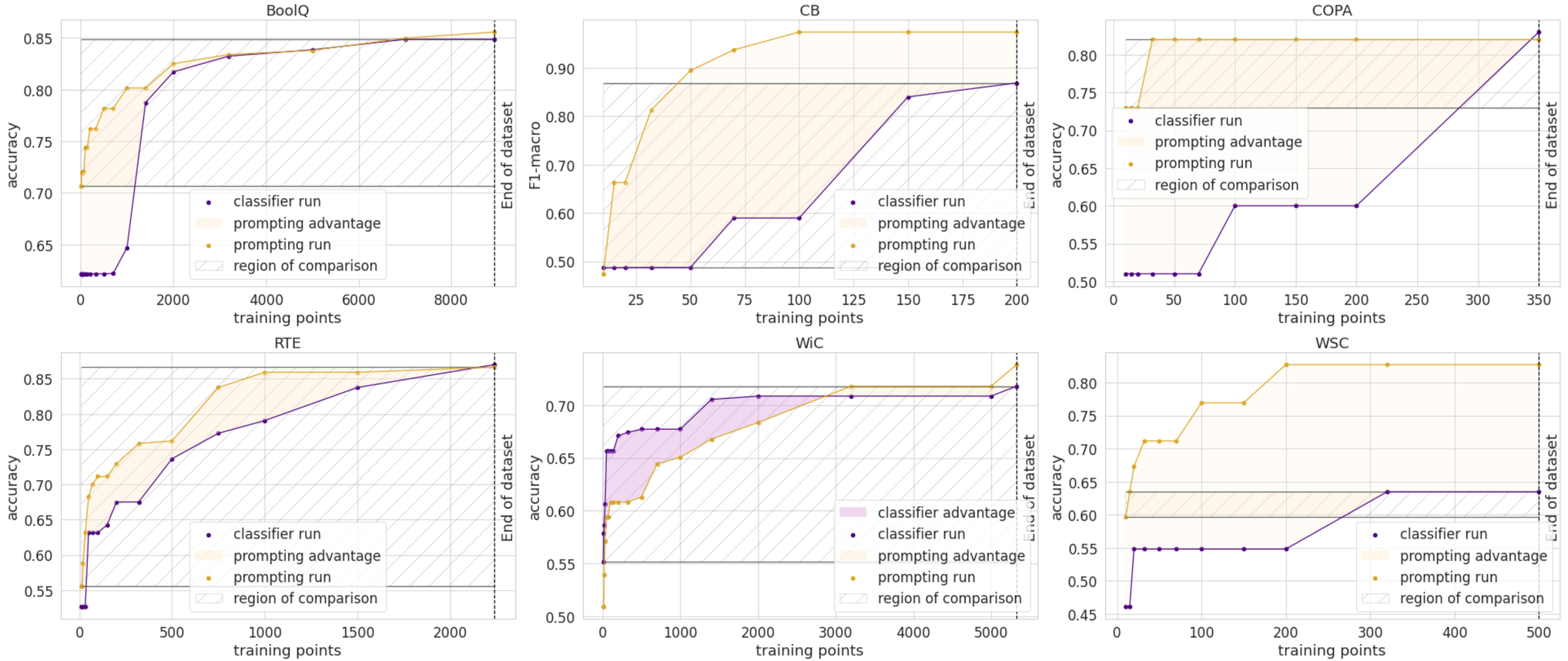


Figure 1: Prompting vs head (classifier) performance across data scales, up to the full dataset, for six SuperGLUE tasks. Compares the best prompt and head performance at each level of training data across 4 runs. Highlighted region shows the accuracy difference of the models. Cross-hatch region highlights the lowest- and highest- accuracy matched region in the curves. The highlighted area in this region is used to estimate the data advantage.

Results

	Average Advantage (# Training Points)							
	MNLI	BoolQ	CB	COPA	MultiRC*	RTE	WiC	WSC
P vs H	3506 ± 536	752 ± 46	90 ± 2	288 ± 242	384 ± 378	282 ± 34	-424 ± 74	281 ± 137
P vs N	150 ± 252	299 ± 81	78 ± 2	-	74 ± 56	404 ± 68	-354 ± 166	-
N vs H	3355 ± 612	453 ± 90	12 ± 1	-	309 ± 320	-122 ± 62	-70 ± 160	-

Table 1: Average prompting advantage in number of data points for MNLI & SuperGLUE tasks. *P* denotes the prompt model, *H* the head model. On average across performance levels, an MNLI prompt model yields the results of an MNLI head model trained with 3500 additional data points. Confidence levels are based on a multiple random runs (see text). *N* indicates a null-verbalizer prompting task that replaces the verbalizer with a non-sensical mapping. *The comparison band of MultiRC is too small as the head baseline fails to learn beyond majority class; we use the full region for a lower-bound result.

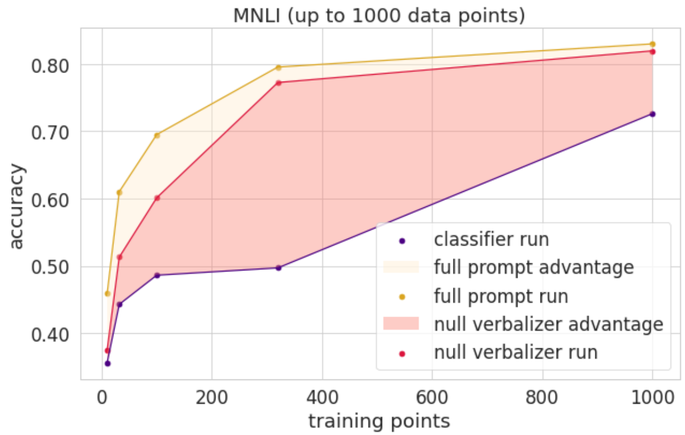


Figure 2: Comparison of full prompt and null verbalizer advantage on MNLI at lower data scales.

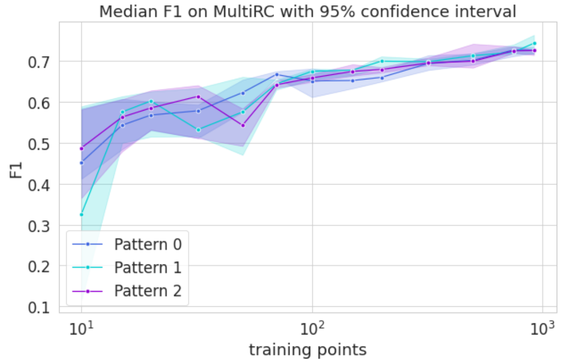


Figure 3: Median performance on MultiRC across runs for three prompts. Differences are inconsistent and eclipsed by the variance within one prompt's runs.

Results

- ☆ Prompt based fine-tuning is mostly robust to pattern choice.
- ☆ Prompting can even learn without an informative verbalizer.
- ☆ Prompting is similarly helpful in terms of data points.

Making Pre-trained Language Models Better Few-shot Learners

Tianyu Gao[†*] **Adam Fisch**[‡*] **Danqi Chen**[†]
[†]Princeton University [‡]Massachusetts Institute of Technology

ACL2021

Method

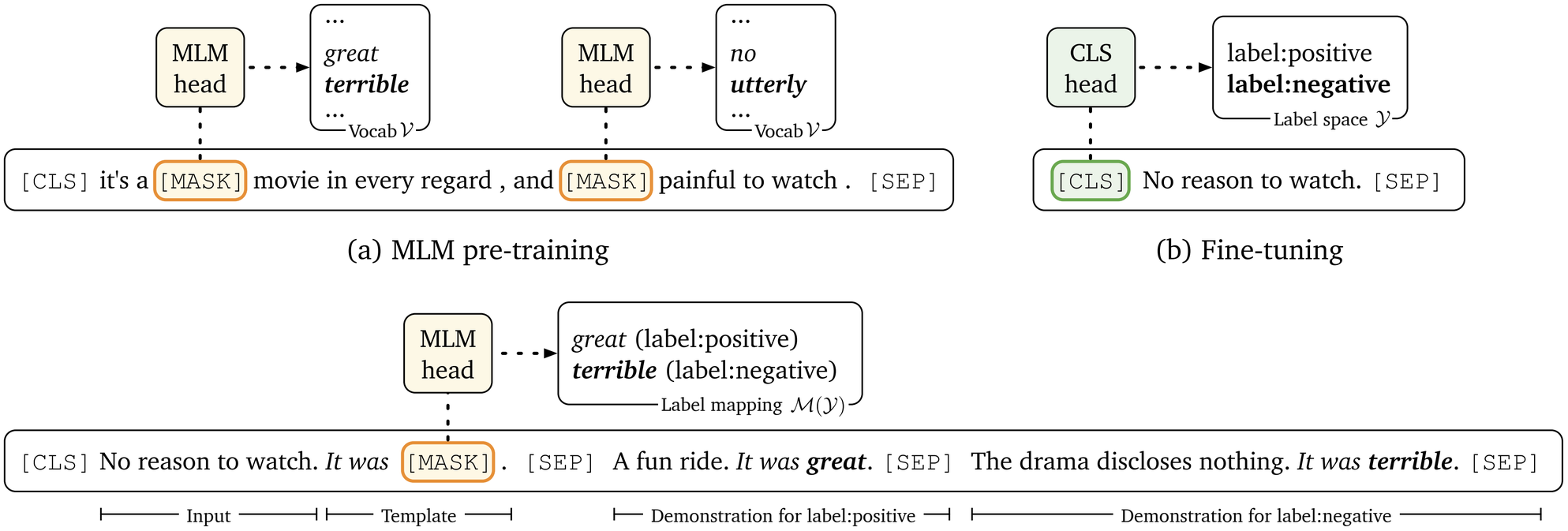


Figure 1: An illustration of (a) masked language model (MLM) pre-training, (b) standard fine-tuning, and (c) our proposed LM-BFF using prompt-based fine-tuning with demonstrations.

Automatic Prompt Generation

Automatic selection of label words

$$\underset{v \in \mathcal{V}}{\text{Top-}k} \left\{ \sum_{x_{\text{in}} \in \mathcal{D}^c_{\text{train}}} \log P_{\mathcal{L}}\Big(\texttt{[MASK]} = v \mid \mathcal{T}(x_{\text{in}}) \Big) \right\},$$

Automatic generation of templates

$$\begin{aligned} <S_1> &\longrightarrow <\texttt{X}> \mathcal{M}(y) <\texttt{Y}> <S_1> \\ <S_1> &\longrightarrow <S_1> <\texttt{X}> \mathcal{M}(y) <\texttt{Y}> \\ <S_1>, <S_2> &\longrightarrow <S_1> <\texttt{X}> \mathcal{M}(y) <\texttt{Y}> <S_2> \end{aligned}$$

$$\sum_{j=1}^{|\mathcal{T}|} \sum_{(x_{\text{in}}, y) \in \mathcal{D}_{\text{train}}} \log P_{\text{T5}}\big(t_j \mid t_1, ..., t_{j-1}, \mathcal{T}_{\text{g}}(x_{\text{in}}, y)\big),$$

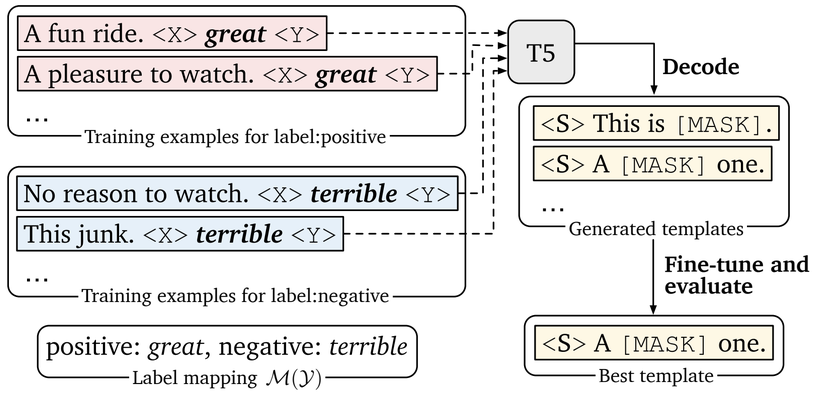


Figure 2: Our approach for template generation.

Fine-tuning with Demonstrations

- Demonstrations: cases for steering PLM

$$\mathcal{T}(x_{\text{in}}) \oplus \tilde{\mathcal{T}}(x_{\text{in}}^{(1)}, y^{(1)}) \oplus \cdots \oplus \tilde{\mathcal{T}}(x_{\text{in}}^{(|\mathcal{Y}|)}, y^{(|\mathcal{Y}|)}),$$

- Sampling similar demonstrations
  - Obtain sentence embeddings by Sentence-BERT
  - Sort demonstrations according cosine similarity
  - Sample the top 50% demonstrations

Results

	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)
Majority†	*50.9*	*23.1*	*50.0*	*50.0*	*50.0*	*50.0*	*18.8*	*0.0*
Prompt-based zero-shot‡	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
"GPT-3" in-context learning	84.8 (1.3)	30.6 (0.9)	80.5 (1.7)	87.4 (0.8)	63.8 (2.1)	53.6 (1.0)	26.2 (2.4)	-1.5 (2.4)
Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)	**33.9** (14.3)
Prompt-based FT (man)	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	91.2 (1.1)	84.8 (5.1)	9.3 (7.3)
+ demonstrations	92.6 (0.5)	**50.6** (1.4)	86.6 (2.2)	90.2 (1.2)	**87.0** (1.1)	**92.3** (0.8)	87.5 (3.2)	18.7 (8.8)
Prompt-based FT (auto)	92.3 (1.0)	49.2 (1.6)	85.5 (2.8)	89.0 (1.4)	85.8 (1.9)	91.2 (1.1)	88.2 (2.0)	14.0 (14.1)
+ demonstrations	**93.0** (0.6)	49.5 (1.7)	**87.7** (1.4)	**91.0** (0.9)	86.5 (2.6)	91.4 (1.8)	**89.4** (1.7)	21.8 (15.9)
Fine-tuning (full)†	*95.0*	*58.7*	*90.8*	*89.4*	*87.8*	*97.0*	*97.4*	*62.6*

	MNLI (acc)	MNLI-mm (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (Pear.)
Majority†	*32.7*	*33.0*	*33.8*	*49.5*	*52.7*	*81.2*	*0.0*	-
Prompt-based zero-shot‡	50.8	51.7	49.5	50.8	51.3	61.9	49.7	-3.2
"GPT-3" in-context learning	52.0 (0.7)	53.4 (0.6)	47.1 (0.6)	53.8 (0.4)	60.4 (1.4)	45.7 (6.0)	36.1 (5.2)	14.3 (2.8)
Fine-tuning	45.8 (6.4)	47.8 (6.8)	48.4 (4.8)	60.2 (6.5)	54.4 (3.9)	76.6 (2.5)	60.7 (4.3)	53.5 (8.5)
Prompt-based FT (man)	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)	71.0 (7.0)
+ demonstrations	**70.7** (1.3)	**72.0** (1.2)	**79.7** (1.5)	**69.2** (1.9)	68.7 (2.3)	77.8 (2.0)	**69.8** (1.8)	73.5 (5.1)
Prompt-based FT (auto)	68.3 (2.5)	70.1 (2.6)	77.1 (2.1)	68.3 (7.4)	**73.9** (2.2)	76.2 (2.3)	67.0 (3.0)	75.0 (3.3)
+ demonstrations	70.0 (3.6)	**72.0** (3.1)	77.5 (3.5)	68.5 (5.4)	71.1 (5.3)	**78.1** (3.4)	67.7 (5.8)	**76.4** (6.2)
Fine-tuning (full)†	*89.8*	*89.5*	*92.6*	*93.3*	*80.9*	*91.4*	*81.7*	*91.9*

Table 3: Our main results using RoBERTa-large. †: full training set is used (see dataset sizes in Table 1); ‡: no training examples are used; otherwise we use $K = 16$ (# examples per class) for few-shot experiments. We report mean (and standard deviation) performance over 5 different splits (§3.3). Majority: majority class; FT: fine-tuning; man: manual prompt (Table B.1); auto: automatically searched templates (§5.2). "GPT-3" in-context learning: using the in-context learning proposed in Brown et al. (2020) with RoBERTa-large (no parameter updates).

PADA: A Prompt-based Autoregressive Approach for Adaptation to Unseen Domains

Eyal Ben-David * **Nadav Oved** * **Roi Reichart**
Technion, Israel Institute of Technology

Motivation

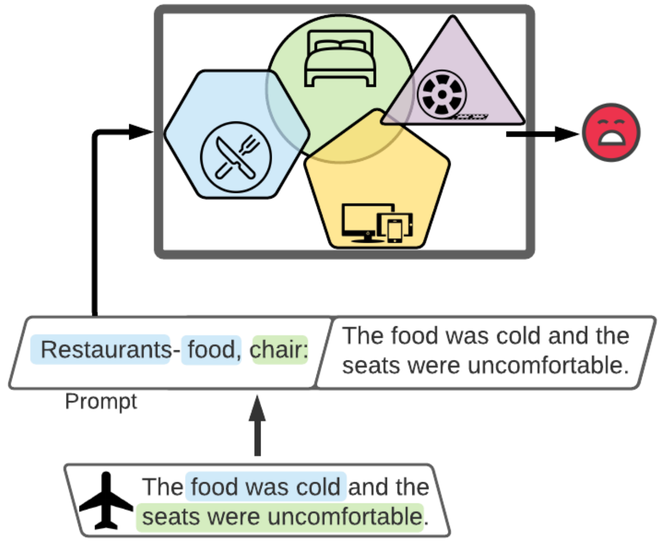
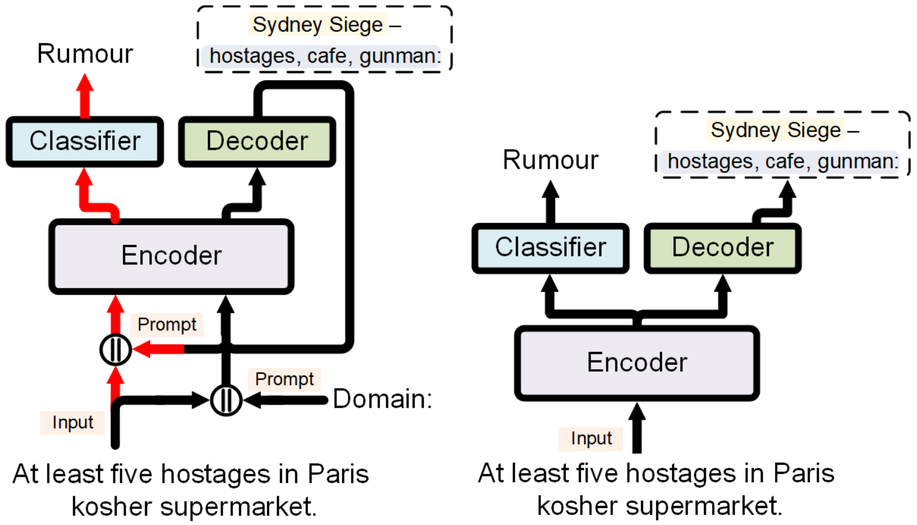


Figure 1: Text classification with PADA. Colored texts signify relation to a specific source domain. *PADA* first generates the domain name, followed by a set of DRFs related to the input example. Then it uses the prompt to predict the task label.

Any-Domain Adaptation

Motivation



(a) *PADA*. (b) *PADA-NC*.

Figure 2: A multi-task model with a generative head trained for DRF generation and a discriminative head for Rumour Detection. Text marked with blue signifies the DRFs and text marked with yellow signifies the domain name. Black arrows (→) mark the first inference step and red arrows (→) mark the second inference step. The presented models are trained on 4 different source domains: *Ferguson*, *Germanwings-crash*, *Ottawa-shooting*, and *Sidney-siege*, while the test example arrives from the *Charlie-Hebdo* domain. The models identify the origin of the example to be *Sydney-siege*, and associates it with 3 DRFs: *hostages*, *cafe*, and *gunman*.

Method

- Generates the name of the training domain
- Generate the Domain Related Features
  - mutual-information (MI) between all tokens and the domain label

$$\frac{C_{\mathcal{S}\setminus\mathcal{S}_j}(n)}{C_{\mathcal{S}_j}(n)} \leq \rho, \;\; C_{\mathcal{S}_j}(n) > 0$$

  - For each example, choose m DRF according to Euclidean distance

Results

	Rumour Detection						MNLI					
	All → C	**All → FR**	**All → GW**	**All → OS**	**All → S**	**AVG**	**All → F**	**All → G**	**All → SL**	**All → TE**	**All → TR**	**AVG**
CCRF	63.6	46.5	70.4	69.0	61.2	62.1	-	-	-	-	-	-
Tr-MoE	68.0	46.1	74.8	58.2	64.9	62.4	64.3	73.9	65.3	62.4	69.8	67.1
T5-NoDA	64.1	46.9	**75.1**	72.0	71.0	65.8	76.4	83.5	75.5	74.9	81.3	78.3
T5-DAN	64.9	52.4	69.1	72.7	64.4	64.7	74.4	76.3	61.0	72.4	77.7	72.4
T5-IRM	63.5	39.4	70.1	44.2	65.7	56.6	72.0	81.5	73.2	69.3	78.9	75.0
T5-MoE	68.1	46.0	73.6	65.3	66.3	63.9	74.0	82.0	73.4	74.6	78.3	76.5
PADA-DN	66.4	53.7	72.4	71.4	70.1	66.8	**77.0**	**84.4**	75.6	76.3	80.5	78.8
PADA-NC	65.8	**54.8**	71.6	72.2	74.0	67.7	76.2	83.6	75.4	77.2	81.4	78.8
PADA	**68.6**	54.4	73.0	**75.2**	**75.1**	**69.3**	76.4	83.4	**76.9**	**78.9**	**82.5**	**79.6**

Table 2: Binary-F1 scores for the Rumour Detection task and macro-F1 scores for the MNLI task.

	Aspect Prediction				
	All → D	**All → L**	**All → R**	**All → SE**	**AVG**
T5-NoDA	31.1	45.6	40.2	37.9	38.7
T5-DAN	28.4	38.0	49.1	33.4	33.2
T5-IRM	37.1	44.6	47.4	**41.5**	42.7
T5-MoE	39.5	31.4	31.4	30.9	33.3
PADA-DN	41.1	42.6	29.0	30.8	35.9
PADA-NC	41.7	48.2	50.1	40.1	45.0
PADA	**41.9**	**50.9**	**50.8**	41.3	**46.2**

Table 3: Binary-F1 scores for Aspect Prediction.

AdaPrompt: Adaptive Prompt-based Finetuning for Relation Extraction

Xiang Chen[1,2 *], **Xin Xie** [1,2 *], **Ningyu Zhang**[1,2 *,†], **Jiahuan Yan**[1 *], **Shumin Deng**[1,2],
Chuanqi Tan[3], **Fei Huang**[3], **Luo Si**[3], **Huajun Chen**[1,2 †]
[1] Zhejiang University [2] AZFT Joint Lab for Knowledge Engine [3] Alibaba Group

Motivation

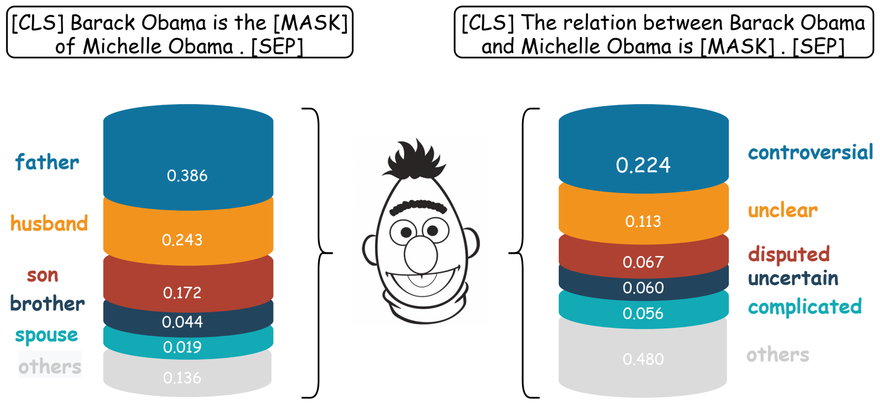


Figure 1: Benefit from pretraining in the large corpus text, BERT without fine-tuning is able to predict quite good results which can be seen in the Figure above that BERT predicts the correct relation type "husband" with 0.243 confidence .

prompt-based fine-tuning for relation extraction

Method

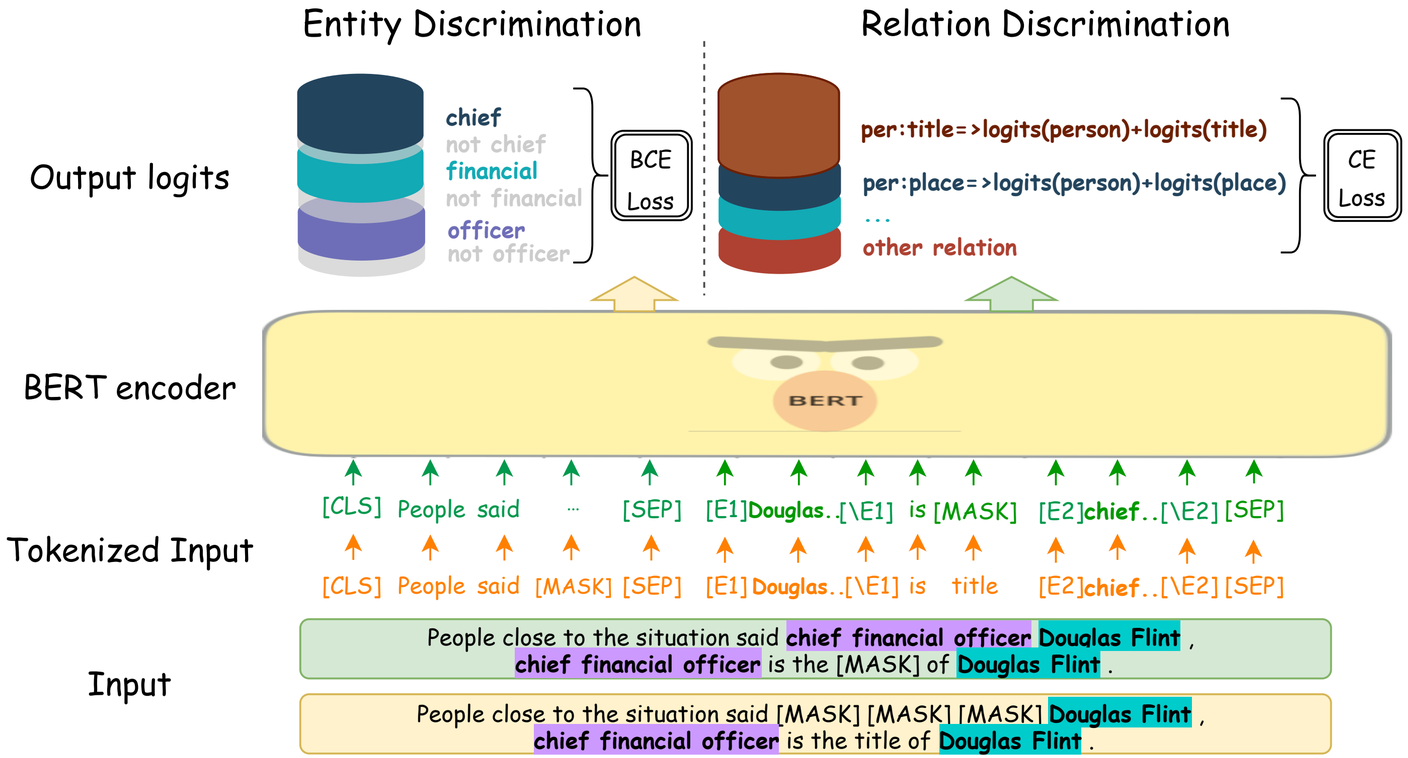


Figure 2: The model architecture is shown as above, for every sample we construct two inputs by masking the relation word and entity name respectively with the purpose of using the knowledge stored in the BERT model itself. As BERT decoder provide the probability over the vocabulary at the `[MASK]` position, we select the output logits of the entity name and compute the BCE Loss forcing the model to memorize the entity name. And similarly we determine the output of relation word by the label name like "per:title" can be seen as "person" + "title".

Method

- Label Words Selection:
  - disassemble the relation label, per:city_of_death, $\mathcal{M}(y) = \{person, city, death\}$
- Relation Discrimination Objective:

$$p(y|X_{\text{prompt}}) = \frac{\exp\left(\mathbf{w}_{\mathcal{M}(y)} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\mathbf{w}_{\mathcal{M}(y')} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)} \quad (4)$$

- Entity Discrimination Objective:

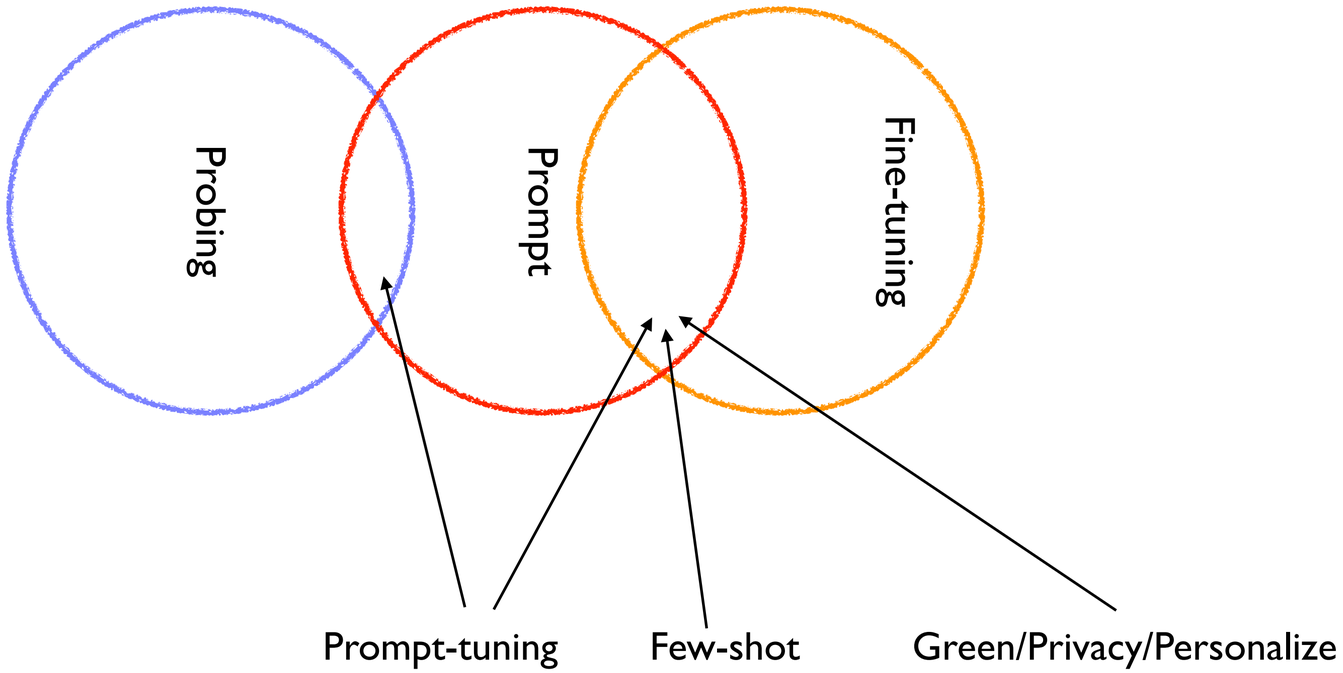
$$q(x^m|x', y) = \frac{\exp([\![L_(x', y)]\!]_{x^m})}{\sum_{v' \in \mathcal{V}} \exp([\![L(x', y)]\!]_{v'})}$$

$$\mathcal{L}_E = \sum_{m \in M} \text{BCE}(q(x^m|x', y).$$

Reflection

- ⭐ Prompt-based probing ❌
- ⭐ From tuning-prompt to prompt-based finetuning ✅
- ⭐ Prompt provide more information for few-shot situation ✅

Reflection



Prompt-based fine-tuning keeps the same learning paradigm as pre-training.
Is prompt-based fine-tuning a future fine-tuning?

THANK YOU !