Mask is All You Need!

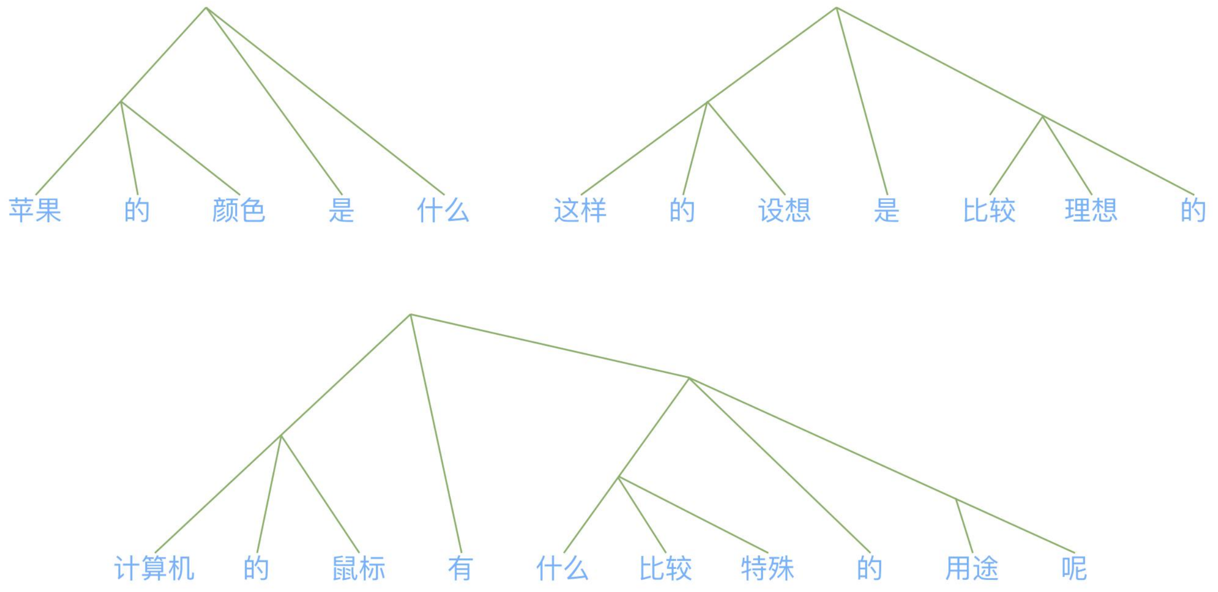
- Mask for interpretion
- Mask for few-shot
- Mask for adversarial attack
- An interesting work about CL in NLP

Jie Zhou
2021.04.29

Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT

Zhiyong Wu[1], **Yun Chen**[2], **Ben Kao**[1], **Qun Liu**[3]
[1]The University of Hong Kong, Hong Kong, China
[2]Shanghai University of Finance and Economics, Shanghai, China
[3]Huawei Noah's Ark Lab, Hong Kong, China
{zywu,kao}@cs.hku.hk, yunchen@sufe.edu.cn, qun.liu@huawei.com

Introduction



[u'习近平', u'总书记', u'6月', u'8日', u'赴', u'宁夏', u'考察', u'调研', u'。', u'当天', u'下午', u'，他先后', u'来到', u'吴忠', u'市', u'红寺堡镇', u'弘德', u'村', u'、黄河', u'吴忠', u'市城区段、', u'金星', u'镇金花园', u'社区', u'，', u'了解', u'当地', u'推进', u'脱贫', u'攻坚', u'、', u'加强', u'黄河流域', u'生态', u'保护', u'、', u'促进', u'民族团结', u'等', u'情况', u'。']

[u'大肠杆菌', u'是', u'人和', u'许多', u'动物', u'肠道', u'中最', u'主要', u'且数量', u'最多', u'的', u'一种', u'细菌']

[u'苏剑林', u'是', u'科学', u'空间', u'的博主']

[u'九寨沟', u'国家级', u'自然', u'保护', u'区', u'位于', u'四川', u'省', u'阿坝藏族羌族', u'自治', u'州', u'南坪县境内', u'，', u'距离', u'成都市400多公里', u'，', u'是', u'一条', u'纵深', u'40余公里', u'的山沟谷', u'地']

Unsupervision parsing

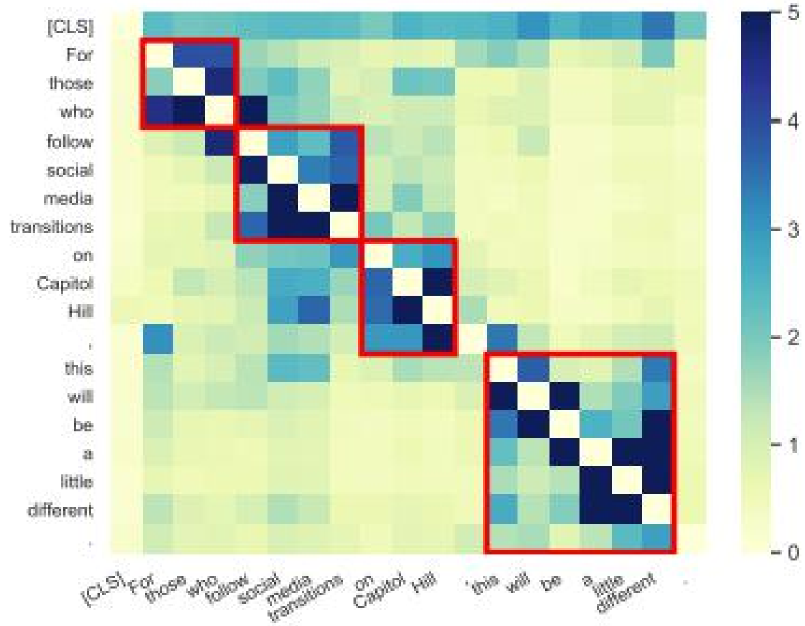
Via Mask in BERT

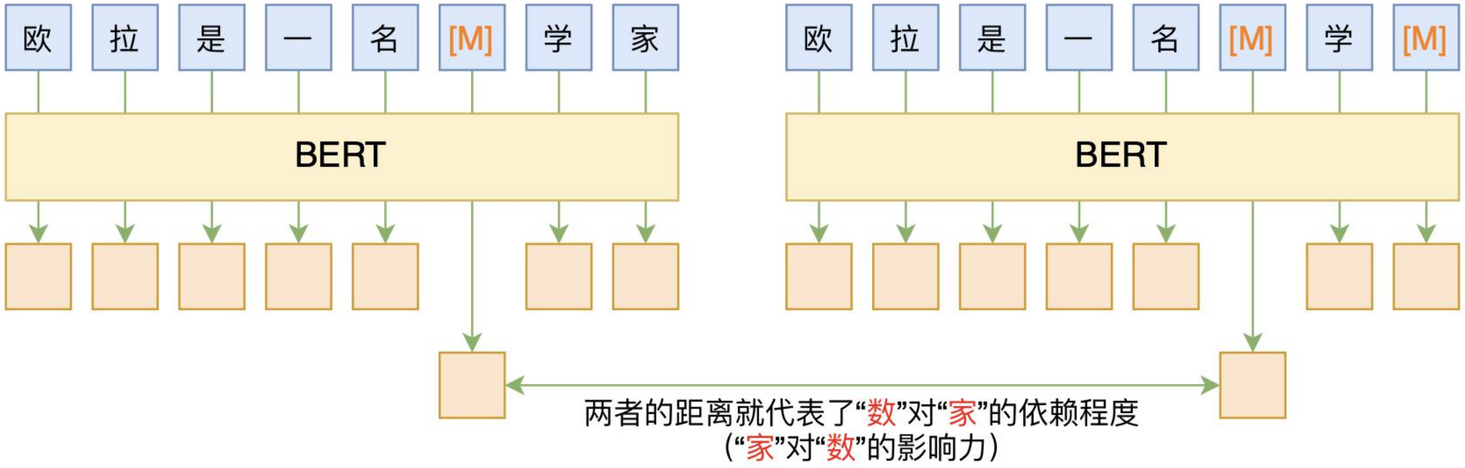
Not probing

Unsupervision word segment

Methods

Token- Token



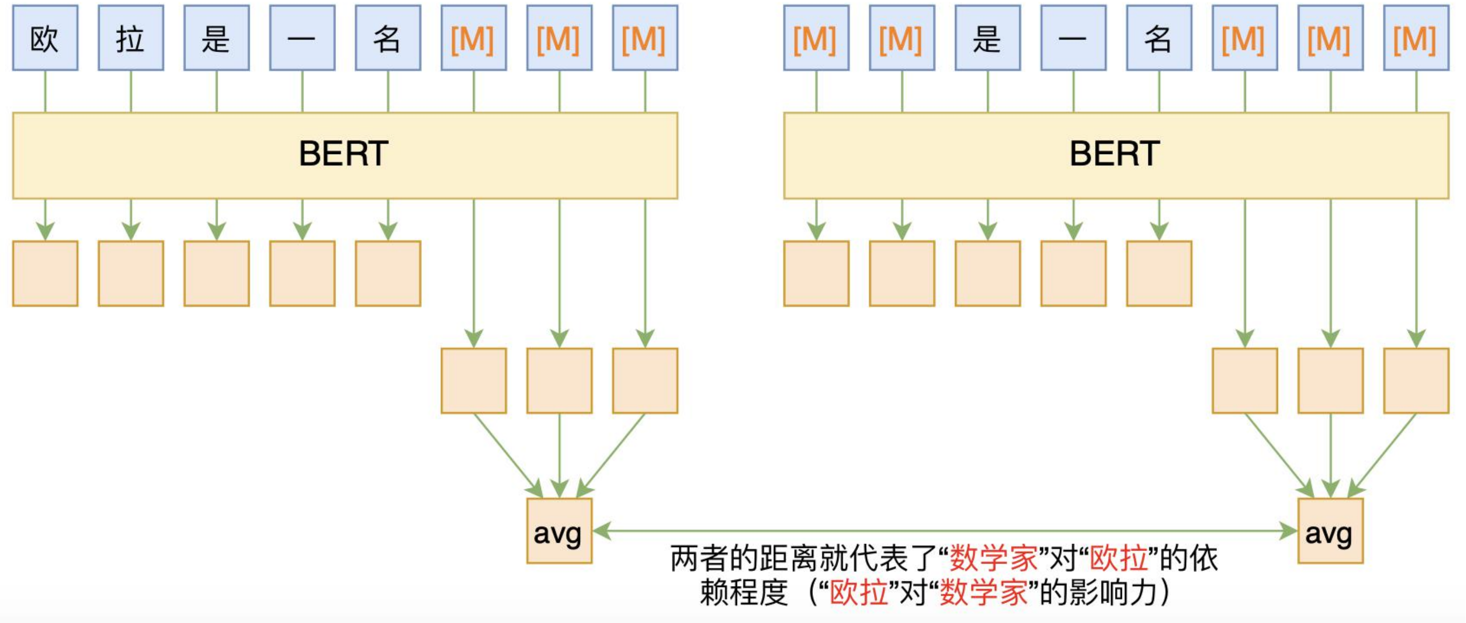


$$f(x_i, x_j) = d\big(H(\boldsymbol{x}\backslash\{x_i\})_i, H(\boldsymbol{x}\backslash\{x_i, x_j\})_i\big)$$

- $f(x_i,x_j)$ 表示第j个token对第i个token的“影响力”
- d(a,b)表示a和b的距离，欧式距离（DIST）、概率变化(Prob)

Methods

Span - Span



$$f(e_i, e_j) = d(H(D\backslash\{e_i\})_i, H(D\backslash\{e_i, e_j\})_i)$$

- $f(e_i,e_j)$ 表示第j个span对第i个span的“影响力”
- $H(D\backslash\{e_i\})$表示e_i中单词表示的平均值

Methods

Word Segment

计算相邻两个词相关性

$$\frac{f(x_i,x_i+1)+f(x_i+1,x_i)}{2}$$

[u'习近平', u'总书记', u'6月', u'8日', u'赴', u'宁夏', u'考察', u'调研', u'。', u'当天', u'下午', u'，他先后', u'来到', u'吴忠', u'市', u'红寺堡镇', u'弘德', u'村', u'、黄河', u'吴忠', u'市城区段、', u'金星', u'镇金花园', u'社区', u'，', u'了解', u'当地', u'推进', u'脱贫', u'攻坚', u'、', u'加强', u'黄河流域', u'生态', u'保护', u'、', u'促进', u'民族团结', u'等', u'情况', u'。']

[u'大肠杆菌', u'是', u'人和', u'许多', u'动物', u'肠道', u'中最', u'主要', u'且数量', u'最多', u'的', u'一种', u'细菌']

[u'苏剑林', u'是', u'科学', u'空间', u'的博主']

[u'九寨沟', u'国家级', u'自然', u'保护', u'区', u'位于', u'四川', u'省', u'阿坝藏族羌族', u'自治', u'州', u'南坪县境内', u'，', u'距离', u'成都市400多公里', u'，', u'是', u'一条', u'纵深', u'40余公里', u'的山沟谷', u'地']

Experiments

Dependency

Model	Parsing UAS WSJ10-U	Parsing UAS PUD
Right-chain	49.5	35.0
Left-chain	20.6	10.7
Random BERT	16.9	10.2
Eisner+Dist	**58.6**	**41.7**
Eisner+Prob	52.7	34.1
CLE+Dist	51.5	33.2

Table 1: UAS results of BERT on unsupervised dependency parsing.

Model	UAS	UUAS	NED
Eisner+Dist	41.7	52.1	69.6
Right-chain	35.0	39.9	41.2

Table 2: Performance on PUD when evaluated using UAS, UUAS, and NED.

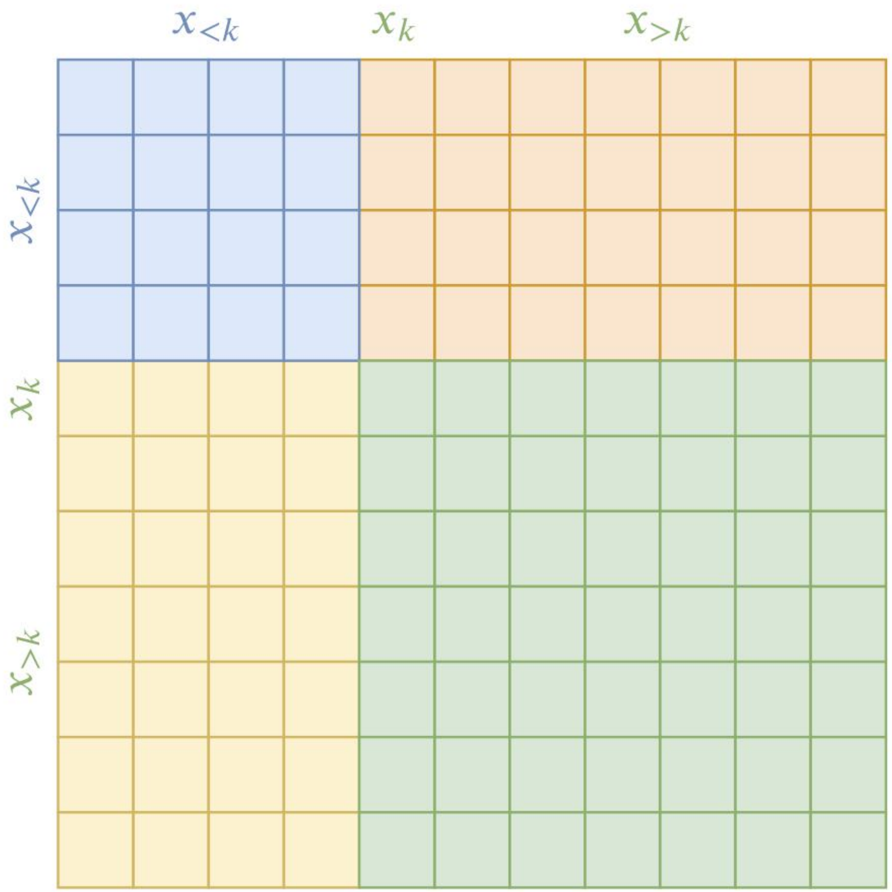
- Right/Left-chain是baseline
- Random BERT: 随机初始化BERT参数
- Eisner/CLE两种计算Parsing的方法
- Dist/Prob 计算距离的两种方式，欧拉距离和概率值

Experiments

Top-Down Parsing

$x=[x_1,x_2,\ldots,x_T]$ 划分为$((x<k),(x_k,(x>k)))$

$$\arg\max_k \underbrace{\frac{\sum_{i=1}^{k-1}\sum_{j=1}^{k-1} f(x_i,x_j)}{(k-1)^2}}_{\text{类内相关性}} + \underbrace{\frac{\sum_{i=k}^{T}\sum_{j=k}^{T} f(x_i,x_j)}{(T-k+1)^2}}_{\text{类内相关性}} - \underbrace{\frac{\sum_{i=1}^{k-1}\sum_{j=k}^{T} f(x_i,x_j)}{(k-1)(T-k+1)}}_{\text{类间相关性}} - \underbrace{\frac{\sum_{i=k}^{T}\sum_{j=1}^{k-1} f(x_i,x_j)}{(k-1)(T-k+1)}}_{\text{类间相关性}}$$



Experiments

Constituency

Model	Parsing F1		Accuracy on PTB23 by Tag				
	WSJ10	PTB23	NP	VP	PP	S	SBAR
PRPN-LM	70.5	37.4	63.9	-	24.4	-	-
ON-LSTM 1st-layer	42.8	24.0	23.8	15.6	18.3	48.1	16.3
ON-LSTM 2nd-layer	66.8	49.4	61.4	51.9	55.4	54.2	15.4
ON-LSTM 3rd-layer	57.6	40.4	57.5	13.5	47.2	48.6	10.4
300D ST-Gumbel w/o Leaf GRU	-	25.0	18.8	-	9.9	-	-
300D RL-SPINN w/o Leaf GRU	-	13.2	24.1	-	14.2	-	-
MART	58.0	42.1	44.6	47.0	50.6	66.1	51.9
Right-Branching	56.7	39.8	25.0	71.8	42.4	74.2	68.8
Left-Branching	19.6	9.0	11.3	0.8	5.0	44.1	5.5

Table 3: Unlabeled parsing F1 results evaluated on WSJ10 and PTB23.

MART: **MA**t**R**ix-based **T**op-down parser

Experiments

Discourse-EDUs

Model	UAS	Accuracy by distance			
		0	1	2	5
Right-chain	10.7	20.5	-	-	-
Left-chain	**41.5**	**79.5**	-	-	-
Random BERT	6.3	20.4	7.5	3.5	0.0
Eisner+Dist	34.2	61.6	7.3	**7.6**	**12.8**
CLE+Dist	34.4	63.8	3.3	3.5	2.6

Table 4: Performance of different discourse parser. The distance is defined as the number of EDUs between head and dependent.

BERT-based Trees VS Parser-provided Trees

Model	Laptop		Restaurant	
	Acc	Macro-F1	Acc	Macro-F1
LSTM	69.63	63.51	77.99	66.91
PWCN				
+Pos	75.23	71.71	81.12	71.81
+Dep	76.08	72.02	80.98	72.28
+Eisner	75.99	72.01	**81.21**	**73.00**
+right-chain	75.64	71.53	81.07	72.51
+left-chain	74.39	70.78	80.82	72.71

Table 5: Experimental results of aspect based sentiment classification.

Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference

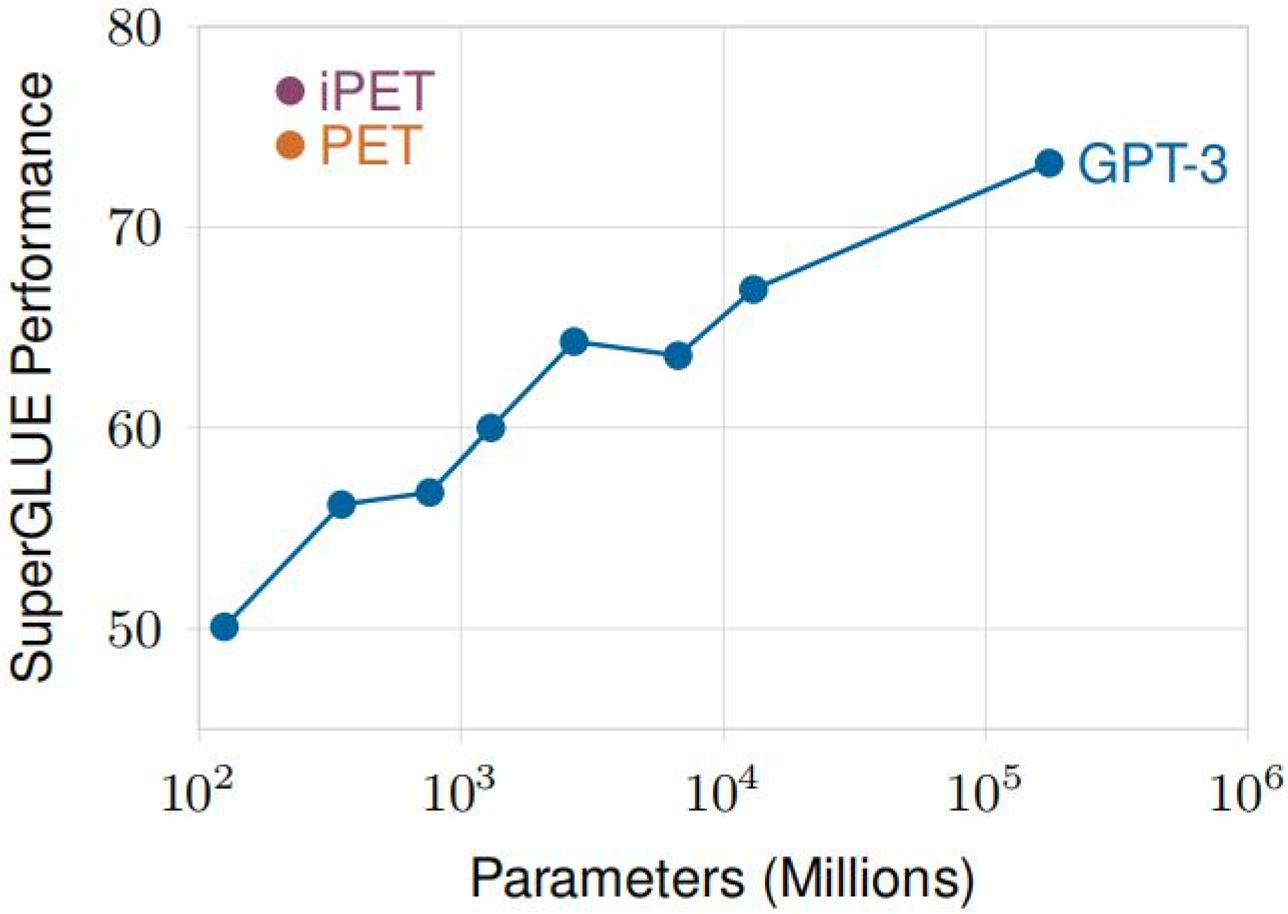
It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners

Timo Schick[1,2] and **Hinrich Schütze**[1]

[1] Center for Information and Language Processing, LMU Munich, Germany
[2] Sulzer GmbH, Munich, Germany

timo.schick@sulzer.de

Introduction



- Pattern-Exploiting Training (PET)
- Cloze question + MLM
- Few/Zero-shot Learning

Methods

Pattern

情感分析：

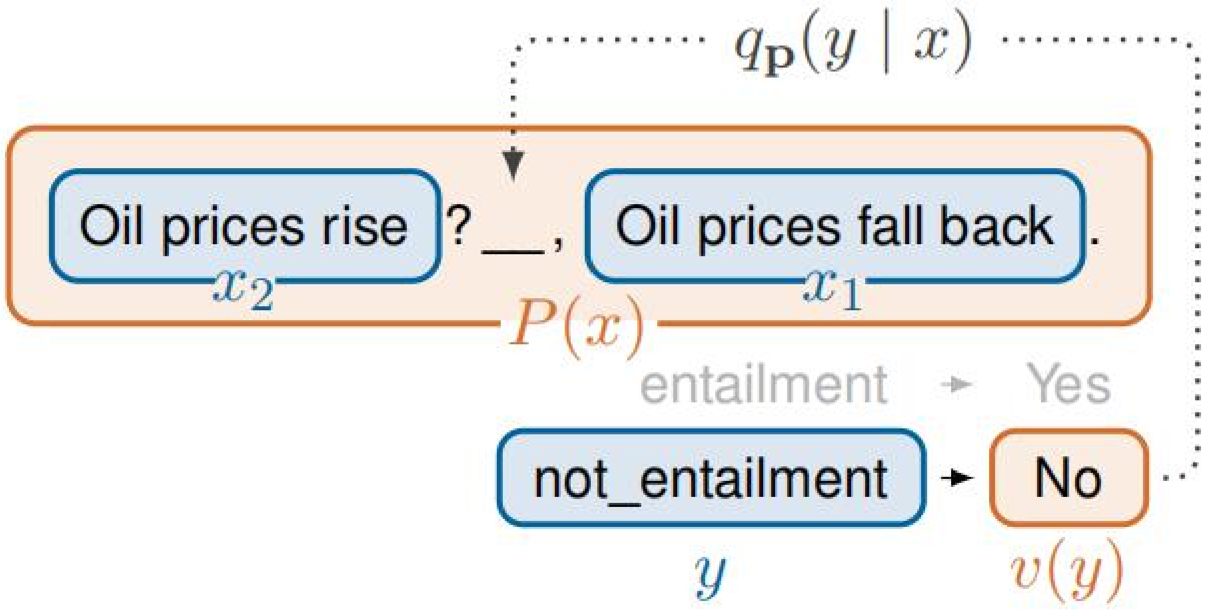
_______满意。这趟北京之旅我感觉很不错。　　很/不

文本分类：

下面报导一则_______新闻。八个月了，终于又能在赛场上看到女排姑娘们了。
体育/金融/娱乐/....

Methods

Pattern-Exploiting Training (PET)



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

1、对于每种Pattern，单独用训练集Finetune一个MLM模型出来；

2、然后将不同Pattern对应的模型进行集成，得到融合模型；

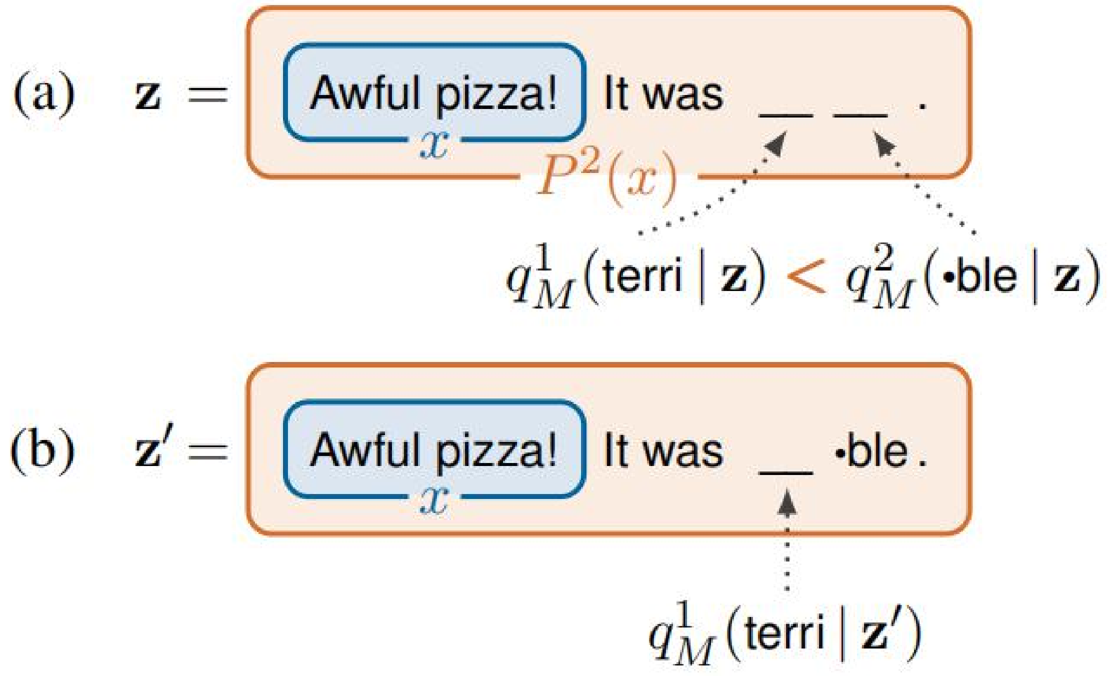
3、用融合模型预测未标注数据的伪标签；

4、用伪标签数据Finetune一个常规的（非MLM的）模型。 (Distillation)

5、iPET: 迭代上面1-4

Methods

PET with Multiple Masks



Experiments

	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., 2020) are shown in italics.

Experiments

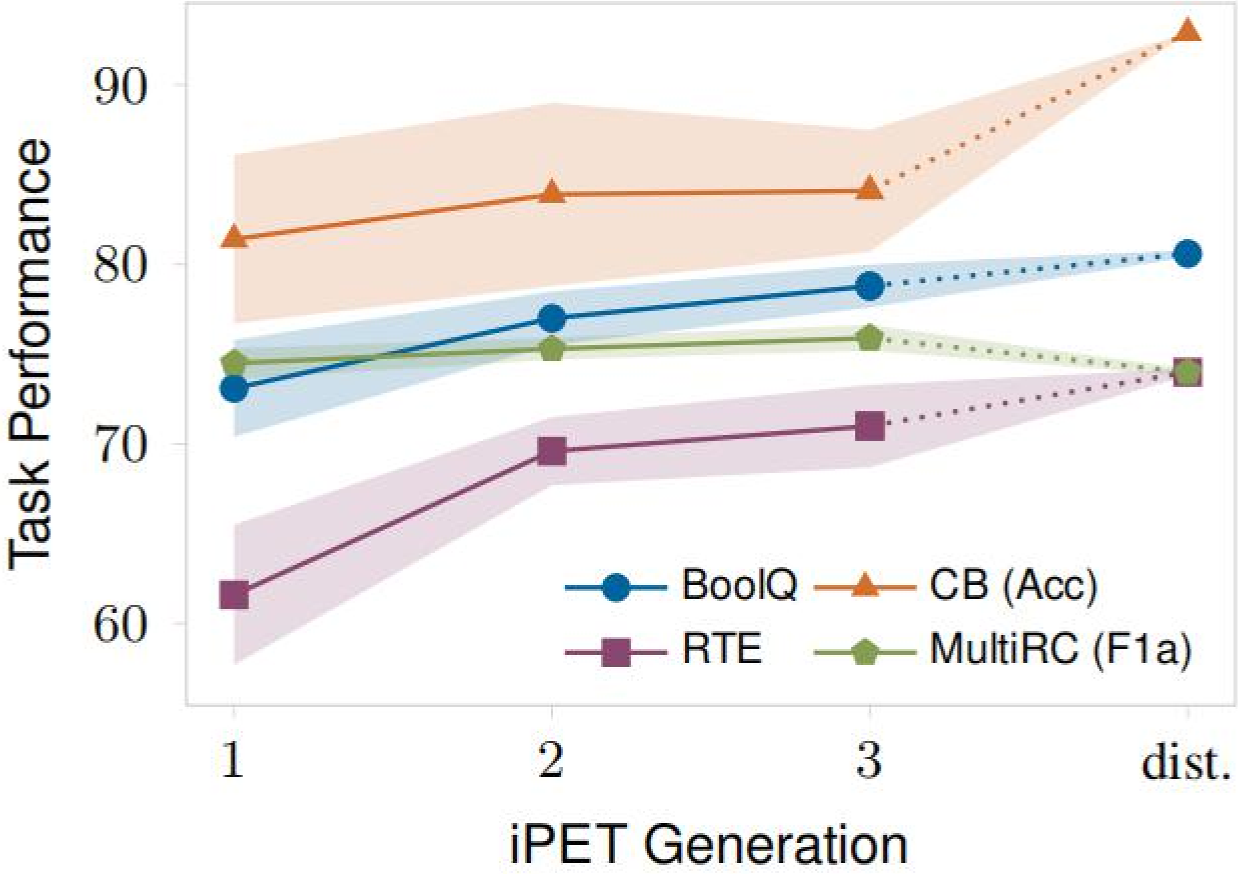
Pattern

Model	CB Acc. / F1	RTE Acc.	MultiRC EM / F1a	Avg –
PET ($\mathbf{p}_{\text{ours}}$)	**85.1** / 59.4	69.8	37.9 / 77.3	66.6
PET ($\mathbf{p}_{\text{GPT-3}}$)	83.3 / 58.1	71.8	25.4 / 68.3	63.1
PET ($\mathbf{p}_{\text{comb}}$)	84.5 / 59.0	**74.7**	39.1 / **77.7**	68.3
PET ($\mathbf{p}_{\text{ours}}$) ¬dist	83.9 / **76.2**	66.4	38.9 / 76.2	68.0
PET ($\mathbf{p}_{\text{comb}}$) ¬dist	83.9 / **76.2**	72.9	**39.6** / 76.6	**70.4**

Table 2: Results on selected tasks for various sets of PVPs for regular PET and for an ensemble of PET models with no knowledge distillation ("¬dist")

Experiments

Usage of labeled and unlabeled data



Model	CB Acc. / F1	RTE Acc.	MultiRC EM / F1a	Avg –
PET	**85.1 / 59.4**	**69.8**	**37.9 / 77.3**	**66.6**
unsupervised	33.5 / 23.1	55.0	3.9 / 60.3	38.5
supervised	60.7 / 42.5	50.2	4.3 / 49.8	43.0
PET (XLNet)	**88.7 / 83.0**	**60.4**	**21.4 / 66.6**	**63.4**
Priming (XLNet)	56.3 / 37.7	49.5	– / –	–

Table 3: Results on selected tasks for various ways of using the labeled examples available in FewGLUE

Experiments

Performance on Chinese

不同模型不同Pattern的零样本学习效果

	P1	P2	P3	P4	P5
M1	66.94 / 67.60	57.56 / 56.13	58.83 / 59.69	83.70 / 83.33	75.98 / 76.13
M2	85.17 / 84.27	70.63 / 68.69	58.55 / 59.12	81.81 / 82.28	80.25 / 81.62
M3	66.75 / 68.64	50.45 / 50.97	68.97 / 70.11	81.95 / 81.48	61.49 / 62.58
M4	83.56 / 85.08	72.52 / 72.10	76.46 / 77.03	88.25 / 87.45	82.43 / 83.56

P1：____满意。这趟北京之旅我感觉很不错。

P2：这趟北京之旅我感觉很不错。____满意。

P3：____好。这趟北京之旅我感觉很不错。

P4：____理想。这趟北京之旅我感觉很不错。

P5：感觉如何？____满意。这趟北京之旅我感觉很不错。

M1：Google开源的中文版BERT Base（链接）；

M2：哈工大开源的RoBERTa-wwm-ext Base（链接）：

M3：腾讯UER开源的BERT Base（链接）；

M4：腾讯UER开源的BERT Large（链接）。

Experiments

结果汇总比较

	P1	P2	P3	P4	P5
M2	85.17 / 84.27	70.63 / 68.69	58.55 / 59.12	81.81 / 82.28	80.25 / 81.62
$M2^{+无监督}$	88.05 / 87.53	71.01 / 68.78	81.05 / 81.24	86.40 / 85.65	87.26 / 87.40
$M2^{+小样本}$	89.29 / 89.18	84.71 / 82.76	88.91 / 89.05	89.31 / 89.13	89.07 / 88.75
$M2^{+半监督}$	90.09 / 89.76	79.58 / 79.35	90.19 / 88.96	90.05 / 89.54	89.88 / 89.23

GPT Understands, Too

Xiao Liu [*,1,2] Yanan Zheng [*,1,2] Zhengxiao Du [1,2] Ming Ding [1,2] Yujie Qian [3] Zhilin Yang [4,2] Jie Tang [1,2]

Introduction

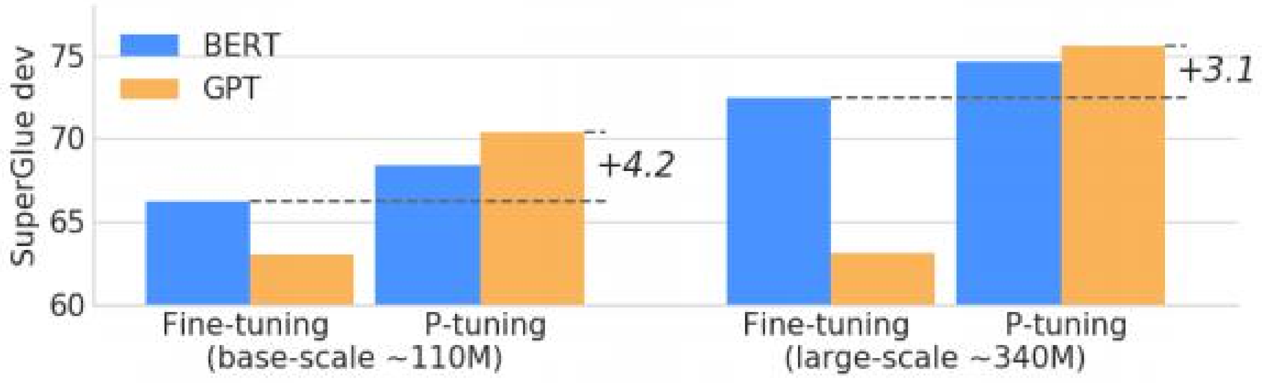


Figure 1. Average scores on 7 dev datasets of SuperGlue. GPTs can be better than similar-sized BERTs on NLU with P-tuning.

- GPT can be better than BERT on NLU
- P-tuning: pattern must be nature language?

Introduction

Prompt	P@1
[X] is located in [Y]. *(original)*	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

Table 1. Case study on LAMA-TREx P17 with bert-base-cased. A single-word change in prompts could yield a drastic difference.

Performance of mannal pattern is also volatile.

Methods

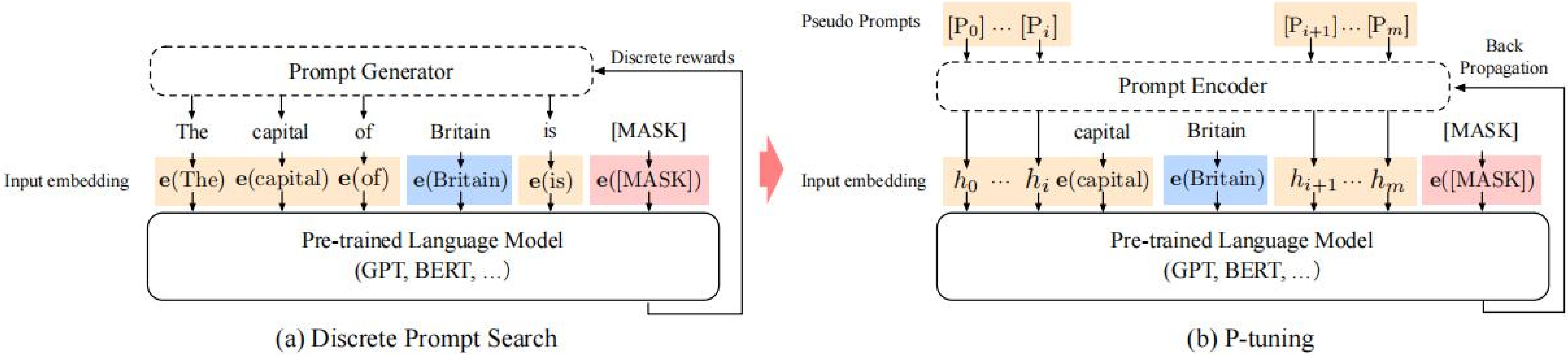


Figure 2. An example of prompt search for "The capital of Britain is [MASK]". Given the context (blue zone, "Britain") and target (red zone, "[MASK]"), the orange zone refer to the prompt tokens. In (a), the prompt generator only receives discrete rewards; on the contrary, in (b) the pseudo prompts and prompt encoder can be optimized in a differentiable way. Sometimes, adding few task-related anchor tokens (such as "capital" in (b)) will bring further improvement.

- Few-shot: fine-tune the embedding
- Enough samples: fine-tune the emebedding with the model parameters

$$\{\mathbf{e}([\mathrm{P}_{0:i}]), \mathbf{e}(\mathbf{x}), \mathbf{e}([\mathrm{P}_{i+1:m}]), \mathbf{e}(\mathbf{y})\}$$

$$\hat{h}_{0:m} = \arg\min_{h} \mathcal{L}(\mathcal{M}(\mathbf{x}, \mathbf{y}))$$

Experiments

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	MP	FT	MP+FT	P-tuning
BERT-base (109M)	31.7	51.6	52.1	52.3 (+20.6)
-AutoPrompt (Shin et al., 2020)	-	-	-	45.2
BERT-large (335M)	33.5	54.0	55.0	54.6 (+21.1)
RoBERTa-base (125M)	18.4	49.2	50.0	49.3 (+30.9)
-AutoPrompt (Shin et al., 2020)	-	-	-	40.0
RoBERTa-large (355M)	22.1	52.3	52.4	53.5 (+31.4)
GPT2-medium (345M)	20.3	41.9	38.2	46.5 (+26.2)
GPT2-xl (1.5B)	22.8	44.9	46.5	54.4 (+31.6)
MegatronLM (11B)	23.1	OOM*	OOM*	**64.2** (+41.1)

* MegatronLM (11B) is too large for effective fine-tuning.

Table 2. Knowledge probing Precision@1 on LAMA-34k (left) and LAMA-29k (right). P-tuning outperforms all the discrete prompt searching baselines. And interestingly, despite fixed pre-trained model parameters, P-tuning overwhelms the fine-tuning GPTs in LAMA-29k. (MP: Manual prompt; FT: Fine-tuning; MP+FT: Manual prompt augmented fine-tuning; PT: P-tuning).

Experiments

Method	BoolQ (Acc.)	CB (Acc.)	CB (F1)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (Acc.)	Avg.
BERT-base-cased (109M)										
Fine-tuning	72.9	85.1	73.9	71.1	68.4	16.2	66.3	63.5	67.0	66.2
MP zero-shot	59.1	41.1	19.4	49.8	54.5	0.4	0.9	62.5	65.0	46.0
MP fine-tuning	73.7	87.5	90.8	67.9	70.4	13.7	62.5	60.6	70.0	67.1
P-tuning	73.9	89.2	92.1	68.8	71.1	14.8	63.3	63.5	72.0	68.4
GPT2-base (117M)										
Fine-tune	71.2	78.6	55.8	65.5	67.8	17.4	65.8	63.0	64.4	63.0
MP zero-shot	61.3	44.6	33.3	54.1	49.5	2.2	23.8	62.5	58.0	48.2
MP fine-tuning	74.8	87.5	88.1	68.0	70.0	23.5	69.7	66.3	78.0	70.2
P-tuning	75.0 (+1.1)	91.1 (+1.9)	93.2 (+1.1)	68.3 (-2.8)	70.8 (-0.3)	23.5 (+7.3)	69.8 (+3.5)	63.5 (+0.0)	76.0 (+4.0)	70.4 (+2.0)

Table 3. Fully-supervised learning on SuperGLUE dev with base-scale models. MP refers to manual prompt. For a 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. Subscript in red represents advantages of GPT with P-tuning over the best results of BERT.

Experiments

Dev size	Method	BoolQ (Acc.)	CB (Acc.)	CB (F1)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (Acc.)
32	PET*	73.2±3.1	82.9±4.3	74.8±9.2	51.8±2.7	62.1±5.3	33.6±3.2	74.5±1.2	79.8±3.5	85.3±5.1
	PET best†	75.1	86.9	83.5	52.6	65.7	35.2	75.0	80.4	83.3
	P-tuning	77.8 (+4.6)	92.9 (+10.0)	92.3 (+17.5)	56.3 (+4.5)	76.5 (+14.4)	36.1 (+2.5)	75.0 (+0.5)	84.6 (+4.8)	87.0 (+1.7)
Full	GPT-3	77.5	82.1	57.2	55.3	72.9	32.5	74.8	75.0	92.0
	PET‡	79.4	85.1	59.4	52.4	69.8	37.9	77.3	80.1	95.0
	iPET§	80.6	92.9	92.4	52.2	74.0	33.0	74.0	-	-

* We report the average and standard deviation of each candidate prompt's average performance.
† We report the best performed prompt selected on *full* dev dataset among all candidate prompts.
‡ With additional ensemble and distillation.
§ With additional data augmentation, ensemble, distillation and self-training.

Table 5. Few-shot learning (32 train samples) on SuperGLUE dev. Previous few-shot learning approaches use the original full dev set ($\mathcal{D}_{dev}$) for validation, which does not make sense. We construct a new dev set ($\mathcal{D}_{dev32}$) with 32 unused samples from original training set. Under fair comparison, P-tuning significantly outperforms PET ($\mathcal{D}_{dev32}$) and PET best ($\mathcal{D}_{dev32}$) on all tasks. More interestingly, P-tuning even outperforms GPT-3, PET ($\mathcal{D}_{dev}$) and iPET ($\mathcal{D}_{dev}$) on 4 out of 7 tasks. Subscripts in red represents the improvements of P-tuning over PET($\mathcal{D}_{dev32}$).

BAE: BERT-based Adversarial Examples for Text Classification

EMNLP 2020

Siddhant Garg[*†]
Amazon Alexa AI Search
Manhattan Beach, CA, USA
sidgarg@amazon.com

Goutham Ramakrishnan[*†]
Health at Scale Corporation
San Jose, CA, USA
gouthamr@cs.wisc.edu

Introduction

Adversarial attack

Original [Positive Sentiment]: This film offers many delights and surprises.	
TextFooler: This flick citations disparate revel and surprises.	
BAE-R: This movie offers enough delights and surprises	
BAE-I: This lovely film platform offers many pleasant delights and surprises	
BAE-R/I: This lovely film serves several pleasure and surprises .	
BAE-R+I: This beautiful movie offers many pleasant delights and surprises .	
Original [Positive Sentiment]: Our server was great and we had perfect service.	
TextFooler: Our server was tremendous and we assumed faultless services.	
BAE-R: Our server was decent and we had outstanding service.	
BAE-I: Our server was great enough and we had perfect service but.	
BAE-R/I: Our server was great enough and we needed perfect service but.	
BAE-R+I: Our server was decent company and we had adequate service.	

Table 3: Qualitative examples of each attack on the BERT classifier (Replacements: Red, Inserts: Blue)

Introduction

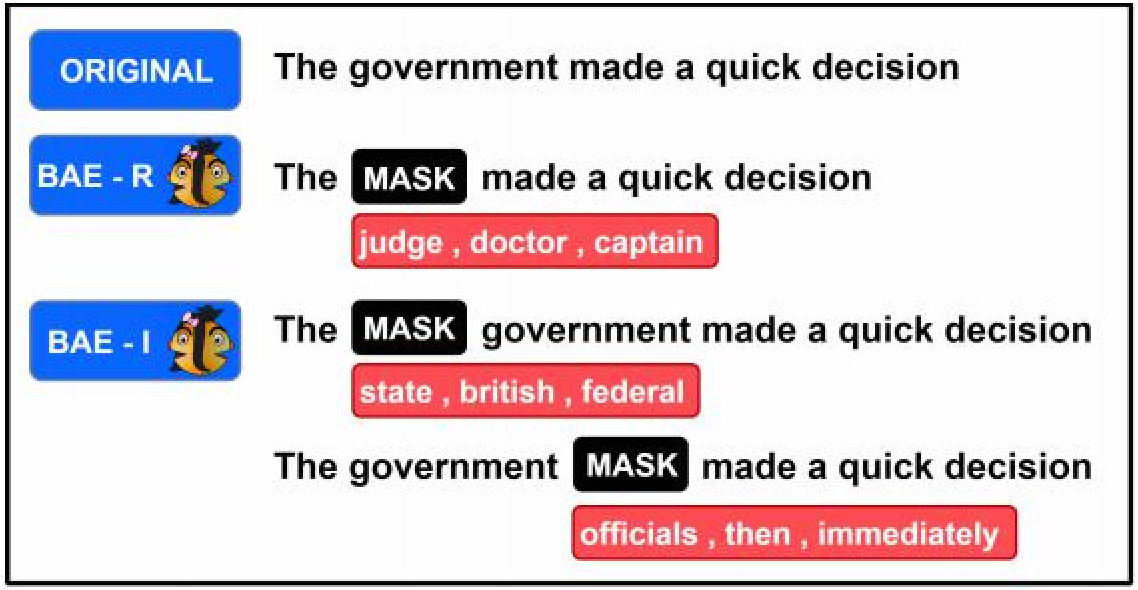


Figure 1: We use BERT-MLM to predict masked tokens in the text for generating adversarial examples. The MASK token replaces a word (`BAE-R` attack) or is inserted to the left/right of the word (`BAE-I`).

- Rule-based synonym replacement strategies: out-of-context and unnaturally.
- BAE: contextual perturbations from a BERT masked language model.

Methods

- Estimate token importance.
  The food is good!
- Replace (R) or Insert (I) a mask token.
  The food is [MASK]!
- Top-k tokens predicted by BERT-MLM
  - Problem: Good -> Bad
  - Filter using USE based on sentence similarity scorer

Algorithm 1: `BAE-R` Pseudocode

Input: Sentence $\mathbb{S} = [t_1, \ldots, t_n]$, ground truth label y, classifier model C
Output: Adversarial Example $\mathbb{S}_{adv}$
Initialization: $\mathbb{S}_{adv} \leftarrow \mathbb{S}$
Compute token importance $I_i \; \forall \; t_i \in \mathbb{S}$
for i *in descending order of* I_i **do**
  $\mathbb{S}_M \leftarrow \mathbb{S}_{adv[1:i-1]}[M]\mathbb{S}_{adv[i+1:n]}$
  Predict top-K tokens $\mathbb{T}$ for mask M $\in \mathbb{S}_M$
  $\mathbb{T} \leftarrow$ FILTER($\mathbb{T}$)
  $\mathbb{L} = \{\}$ `// python-style dict`
  for $t \in \mathbb{T}$ **do**
    $\mathbb{L}[t] = \mathbb{S}_{adv[1:i-1]}[t]\mathbb{S}_{adv[i+1:n]}$
  end
  if $\exists \, t \in \mathbb{T}$ s.t $C(\mathbb{L}[t]) \neq y$ **then**
    Return: $\mathbb{S}_{adv} \leftarrow \mathbb{L}[t']$ where $C(\mathbb{L}[t']) \neq y$, $\mathbb{L}[t']$ has maximum similarity with $\mathbb{S}$
  else
    $\mathbb{S}_{adv} \leftarrow \mathbb{L}[t']$ where $\mathbb{L}[t']$ causes maximum reduction in probability of y in $C(\mathbb{L}[t'])$
  end if
end
Return: $\mathbb{S}_{adv} \leftarrow None$

Experiments

Model	Adversarial Attack	Datasets			
		Amazon	Yelp	IMDB	MR
wordLSTM	Original	88.0	85.0	82.0	81.16
	TextFooler	31.0 (0.747)	28.0 (0.829)	20.0 (0.828)	25.49 (0.906)
	BAE-R	21.0 (0.827)	20.0 (0.885)	22.0 (0.852)	24.17 (0.914)
	BAE-I	17.0 (0.924)	22.0 (0.928)	23.0 (0.933)	19.11 (0.966)
	BAE-R/I	16.0 (0.902)	19.0 (0.924)	8.0 (0.896)	15.08 (0.949)
	BAE-R+I	**4.0 (0.848)**	**9.0 (0.902)**	**5.0 (0.871)**	**7.50 (0.935)**
wordCNN	Original	82.0	85.0	81.0	76.66
	TextFooler	42.0 (0.776)	36.0 (0.827)	31.0 (0.854)	21.18 (0.910)
	BAE-R	16.0 (0.821)	23.0 (0.846)	23.0 (0.856)	20.81 (0.920)
	BAE-I	18.0 (0.934)	26.0 (0.941)	29.0 (0.924)	19.49 (0.971)
	BAE-R/I	13.0 (0.904)	17.0 (0.916)	20.0 (0.892)	15.56 (0.956)
	BAE-R+I	**2.0 (0.859)**	**9.0 (0.891)**	**14.0 (0.861)**	**7.87 (0.938)**
BERT	Original	96.0	95.0	85.0	85.28
	TextFooler	30.0 (0.787)	27.0 (0.833)	32.0 (0.877)	30.74 (0.902)
	BAE-R	36.0 (0.772)	31.0 (0.856)	46.0 (0.835)	44.05 (0.871)
	BAE-I	20.0 (0.922)	25.0 (0.936)	31.0 (0.929)	32.05 (0.958)
	BAE-R/I	**11.0 (0.899)**	16.0 (0.916)	22.0 (0.909)	20.34 (0.941)
	BAE-R+I	14.0 (0.830)	**12.0 (0.871)**	**16.0 (0.856)**	**19.21 (0.917)**

Table 1: Automatic evaluation of adversarial attacks on 4 Sentiment Classification tasks. We report the test set accuracy. The average semantic similarity, between the original and adversarial examples, obtained from USE are reported in parentheses. Best performance, in terms of maximum drop in test accuracy, is highlighted in **boldface**.

Experiments

Dataset	Sentiment Accuracy (%)			
	Original	TF	R	R+I
Amazon	95.7	79.1	**85.2**	83.8
IMDB	90.3	83.1	**84.3**	79.3
MR	93.3	82.0	**84.6**	82.4

Dataset	Naturalness (1-5)			
	Original	TF	R	R+I
Amazon	4.26	3.17	**3.91**	3.71
IMDB	4.35	3.41	**3.89**	3.76
MR	4.19	3.35	**3.84**	3.74

Table 4: Human evaluation results (TF: TextFooler and R(R+I): `BAE-R(R+I)`).

SimCSE: Simple Contrastive Learning of Sentence Embeddings

Tianyu Gao[†*] **Xingcheng Yao**[‡*] **Danqi Chen**[†]
[†]Department of Computer Science, Princeton University
[‡]Institute for Interdisciplinary Information Sciences, Tsinghua University
{tianyug,danqic}@cs.princeton.edu
yxc18@mails.tsinghua.edu.cn

Introduction

Contrastive Learning

$$\ell_i = \log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}},$$

- x^+为x的正样本 .
- 图像连续，NLP离散

Introduction

Data Augmentation

$$\ell_i = \log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}},$$

- x^+为x的正样本 .
- 数据增强：图像连续，NLP离散

Introduction

SimCSE

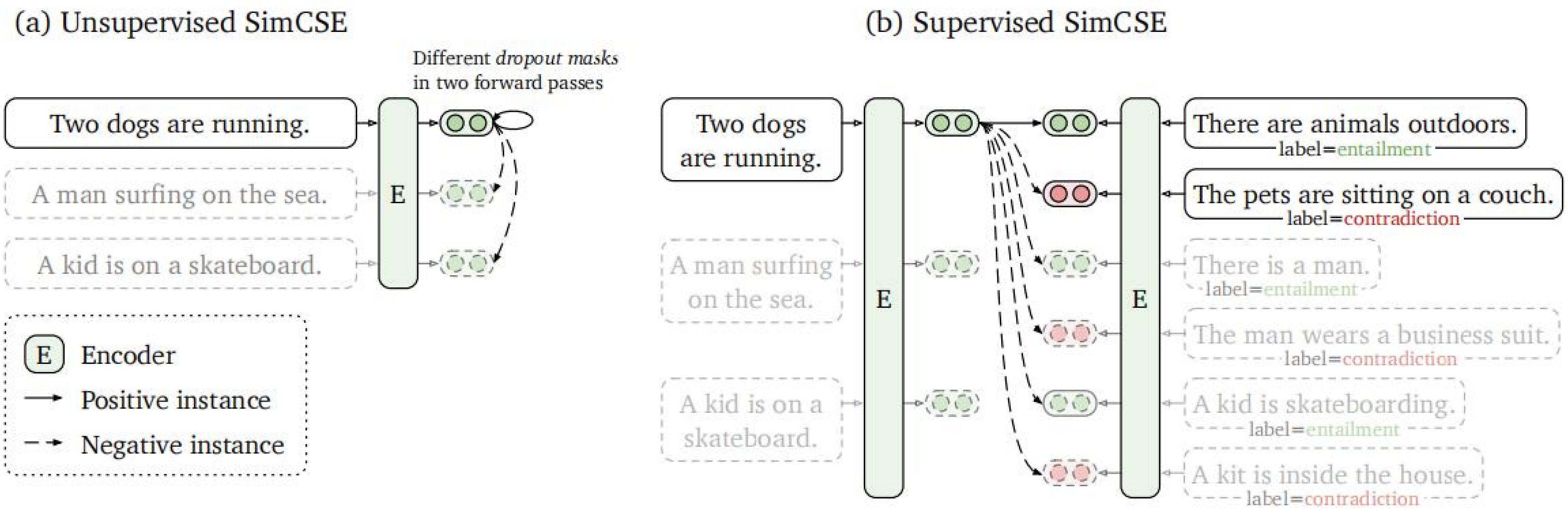


Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

Methods

SimCSE

$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z'_i})/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z'_j})/\tau}},$$

Different dropout masks z, z‘

Data augmentation			STS-B
None			**79.1**
Crop	*10%*	*20%*	*30%*
	75.4	70.1	63.7
Word deletion	*10%*	*20%*	*30%*
	74.7	71.2	70.2
Delete one word			74.8
w/o dropout			71.4
MLM 15%			66.8
Crop 10% + MLM 15%			70.8

Table 2: Comparison of different data augmentations on STS-B development set (Spearman's correlation). *Crop k%*: randomly crop and keep a continuous span with 100-*k*% of the length; *word deletion k%*: randomly delete *k*% words; *delete one word*: randomly delete one word; *MLM k%*: use $\text{BERT}_{\texttt{base}}$ to replace *k*% of words. All of them include the standard 10% dropout (except "w/o dropout").

Methods

SimCSE

$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z'_i})/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z'_j})/\tau}},$$

Different dropout masks z, z‘

Data augmentation			STS-B
None			**79.1**
Crop	*10%*	*20%*	*30%*
	75.4	70.1	63.7
Word deletion	*10%*	*20%*	*30%*
	74.7	71.2	70.2
Delete one word			74.8
w/o dropout			71.4
MLM 15%			66.8
Crop 10% + MLM 15%			70.8

Table 2: Comparison of different data augmentations on STS-B development set (Spearman's correlation). *Crop k%*: randomly crop and keep a continuous span with 100-*k*% of the length; *word deletion k%*: randomly delete *k*% words; *delete one word*: randomly delete one word; *MLM k%*: use $\text{BERT}_{\texttt{base}}$ to replace *k*% of words. All of them include the standard 10% dropout (except "w/o dropout").

Methods

SimCSE

Training objective	f_θ	$(f_{\theta_1}, f_{\theta_2})$
Next sentence	66.8	67.7
Next 3 sentences	68.7	69.7
Delete one word	74.8	70.4
Unsupervised SimCSE	**79.1**	70.7

Table 3: Comparison of different unsupervised objectives. Results are Spearman's correlation on the STS-B development set using $\text{BERT}_{\texttt{base}}$, trained on 1-million pairs from Wikipedia. The two columns denote whether we use one encoder f_θ or two independent encoders f_{θ_1} and f_{θ_2} ("dual-encoder"). *Next 3 sentences*: randomly sample one from the next 3 sentences. *Delete one word*: delete one word randomly (see Table 2).

p	*0.0*	*0.01*	*0.05*	*0.1*
STS-B	64.9	69.5	78.0	**79.1**
p	*0.15*	*0.2*	*0.5*	*Fixed 0.1*
STS-B	78.6	78.2	67.4	45.2

Table 4: Effects of different dropout probabilities p on the STS-B development set (Spearman's correlation, $\text{BERT}_{\texttt{base}}$). *Fixed 0.1*: use the default 0.1 dropout rate but apply the same dropout mask on both x_i and x_i^+.

Methods

Supervised SimCSE

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)},$$

- x is the premise.
- x^+ and x^- are entailment and contradiction hypotheses

Dataset	sample	full
Unsup. SimCSE (1m)	-	79.1
QQP (134k)	81.8	81.8
Flickr30k (318k)	81.5	81.4
ParaNMT (5m)	79.7	78.7
SNLI+MNLI		
entailment (314k)	**84.1**	**84.9**
neutral (314k)[3]	82.6	82.9
contradiction (314k)	77.5	77.6
SNLI+MNLI		
entailment + hard neg.	-	**86.2**
+ ANLI (52k)	-	85.0

Table 5: Comparisons of different supervised datasets as positive pairs. Results are Spearman's correlation on the STS-B development set using BERT$_{\texttt{base}}$. Numbers in brackets denote the # of pairs. *Sample*: subsampling 134k positive pairs for a fair comparison between datasets; *full*: using the full dataset. In the last block, we use entailment pairs as positives and contradiction

Methods

Connection to Anisotropy

- Anisotropy problem in language representation.
  - A few dominating singular values, all others are close to zero.

- Postprocessing methods
  - Eliminate the dominant principal components
  - Map embeddings to an isotropic distribution
  - Add regularization during training

- Contrastive learning objective can inherently "flatten" the singular value distribution

Methods

Connection to Anisotropy

Alignment

$$-\frac{1}{\tau}\underset{(x,x^+)\sim p_{\text{pos}}}{\mathbb{E}}\left[f(x)^\top f(x^+)\right]$$

$$+\underset{x\sim p_{\text{data}}}{\mathbb{E}}\left[\log \underset{x^-\sim p_{\text{data}}}{\mathbb{E}}\left[e^{f(x)^\top f(x^-)/\tau}\right]\right],$$

Uniformity

Uniformity

$$\begin{aligned}&\underset{x\sim p_{\text{data}}}{\mathbb{E}}\left[\log \underset{x^-\sim p_{\text{data}}}{\mathbb{E}}\left[e^{f(x)^\top f(x^-)/\tau}\right]\right]\\ =&\frac{1}{m}\sum_{i=1}^{m}\log\left(\frac{1}{m}\sum_{j=1}^{m}e^{\mathbf{h}_i^\top \mathbf{h}_j/\tau}\right)\\ \geq&\frac{1}{\tau m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{h}_i^\top \mathbf{h}_j.\end{aligned}$$

$$\text{Sum}(\mathbf{W}\mathbf{W}^\top)=\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{h}_i^\top \mathbf{h}_j$$

ing to Merikoski (1984), if all elements in $\mathbf{W}\mathbf{W}^\top$ are positive, which is the case in most times from Gao et al. (2019), then $\text{Sum}(\mathbf{W}\mathbf{W}^\top)$ is an upper bound for the largest eigenvalue of $\mathbf{W}\mathbf{W}^\top$. There-

Experiments

Semantic textual similarity tasks

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\texttt{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\texttt{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\texttt{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\texttt{base}}$ ♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
∗ SimCSE-$\text{BERT}_{\texttt{base}}$	**66.68**	**81.43**	**71.38**	**78.43**	**78.47**	**75.49**	**69.92**	**74.54**
$\text{RoBERTa}_{\texttt{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
$\text{RoBERTa}_{\texttt{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
∗ SimCSE-$\text{RoBERTa}_{\texttt{base}}$	**68.68**	**82.62**	**73.56**	**81.49**	**80.82**	**80.48**	**67.87**	**76.50**
∗ SimCSE-$\text{RoBERTa}_{\texttt{large}}$	**69.87**	**82.97**	**74.25**	**83.01**	**79.52**	**81.23**	**71.47**	**77.47**
Supervised models								
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{SBERT}_{\texttt{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\texttt{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\texttt{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
∗ SimCSE-$\text{BERT}_{\texttt{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
$\text{SRoBERTa}_{\texttt{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\texttt{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ SimCSE-$\text{RoBERTa}_{\texttt{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ SimCSE-$\text{RoBERTa}_{\texttt{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Experiments

Transfer tasks

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
				Unsupervised models				
GloVe embeddings (avg.)♣	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought♡	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings♣	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT-[CLS] embedding♣	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT$_{\texttt{base}}$♡	**81.09**	**87.18**	**94.96**	**88.75**	**85.96**	88.64	74.24	**85.83**
∗ SimCSE-BERT$_{\texttt{base}}$	80.41	85.30	94.46	88.43	85.39	87.60	71.13	84.67
w/ MLM	80.74	85.67	94.68	87.21	84.95	89.40	**74.38**	85.29
∗ SimCSE-RoBERTa$_{\texttt{base}}$	79.67	84.61	91.68	85.96	84.73	84.20	64.93	82.25
w/ MLM	**82.02**	**87.52**	**94.13**	**86.24**	**88.58**	**90.20**	**74.55**	**86.18**
∗ SimCSE-RoBERTa$_{\texttt{large}}$	80.83	85.30	91.68	86.10	85.06	89.20	**75.65**	84.83
w/ MLM	**83.30**	**87.50**	**95.27**	**86.82**	**87.86**	**94.00**	75.36	**87.16**
				Supervised models				
InferSent-GloVe♣	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder♣	80.09	85.19	93.98	86.70	86.38	93.20	70.14	85.10
SBERT$_{\texttt{base}}$♣	**83.64**	**89.43**	94.39	**89.86**	**88.96**	**89.60**	76.00	**87.41**
∗ SimCSE-BERT$_{\texttt{base}}$	82.69	89.25	**94.81**	89.59	87.31	88.40	73.51	86.51
w/ MLM	82.68	88.88	94.52	89.82	88.41	87.60	**76.12**	86.86
SRoBERTa$_{\texttt{base}}$	84.91	90.83	92.56	88.75	90.50	88.60	**78.14**	87.76
∗ SimCSE-RoBERTa$_{\texttt{base}}$	84.92	**92.00**	**94.11**	**89.82**	91.27	88.80	75.65	88.08
w/ MLM	**85.08**	91.76	94.02	89.72	**92.31**	**91.20**	76.52	**88.66**
∗ SimCSE-RoBERTa$_{\texttt{large}}$	88.12	92.37	95.11	90.49	92.75	91.80	76.64	89.61
w/ MLM	**88.45**	**92.53**	**95.19**	**90.58**	**93.30**	**93.80**	**77.74**	**90.23**

Experiments

MLM auxiliary task

Model	STS-B	Avg. transfer
[CLS]	**86.2**	85.8
First-last avg.	86.1	86.1
w/o MLM	**86.2**	85.8
w/ MLM		
$\lambda = 0.01$	85.7	86.1
$\lambda = 0.1$	85.7	**86.2**
$\lambda = 1$	85.1	85.8

Table 9: Ablation studies of different pooling methods and incorporating the MLM objective. The results are based on the development sets using $\text{BERT}_{\texttt{base}}$.

Experiments

Uniformity and alignment

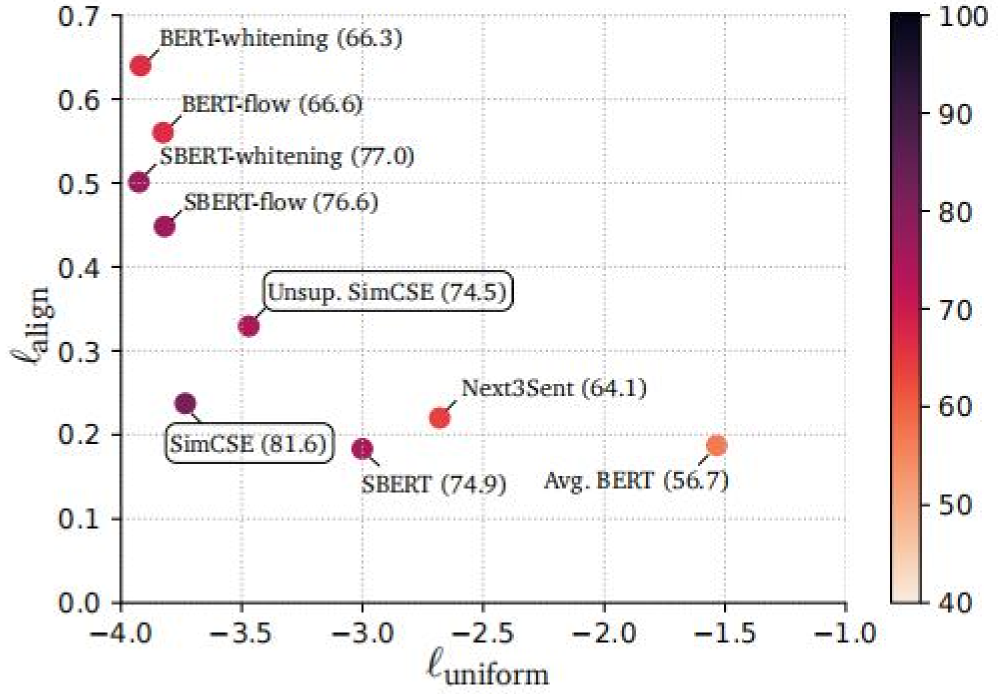


Figure 3: ℓ_{align}-ℓ_{uniform} plot of models based on $\text{BERT}_{\texttt{base}}$. Color of points and numbers in brackets represent average STS performance (Spearman's correlation). *Next3Sent*: "next 3 sentences" from Table 3.

Experiments

Cosine-similarity distribution

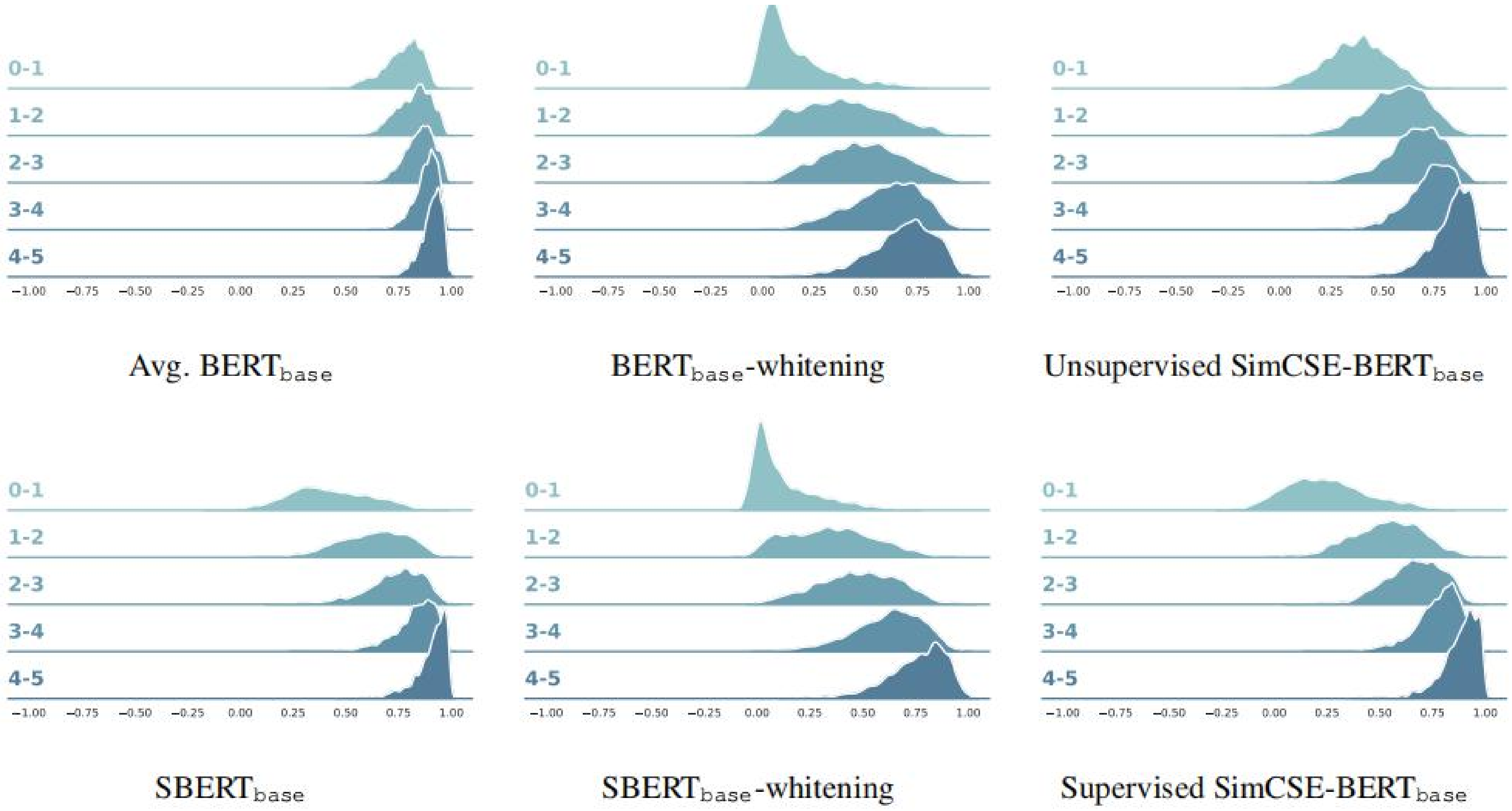


Figure 4: Density plots of cosine similarities between sentence pairs in full STS-B. Pairs are divided into 5 groups based on ground truth ratings (higher means more similar) along the y-axis, and x-axis is the cosine similarity.

Thanks！ Q&A