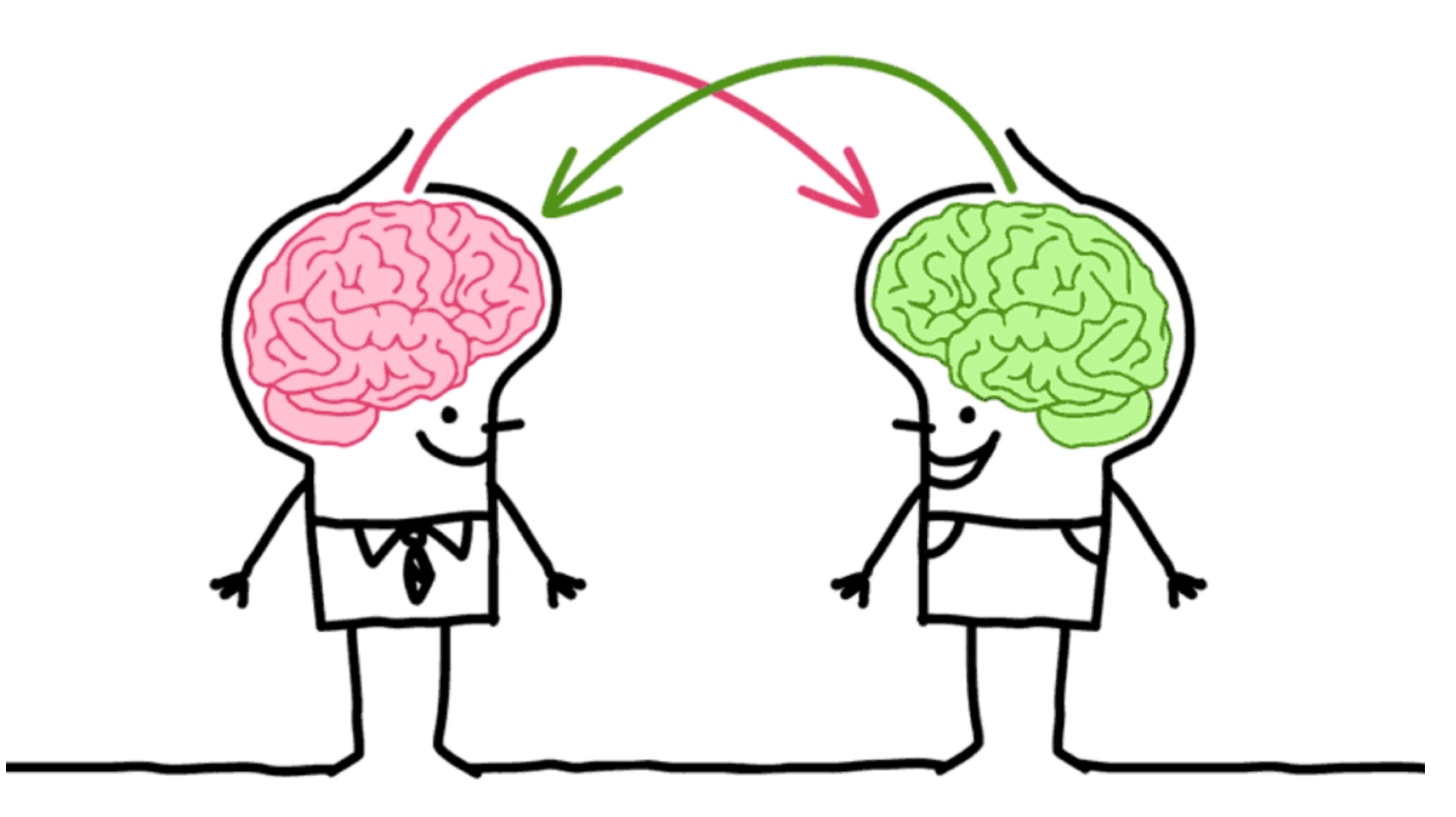
迁移Prompt–解决Prompt Tuning三大问题！

原创 Harris 夕小瑶的卖萌屋 2021-12-15 12:05



文 | Harris

刘鹏飞博士将近代NLP的研究划归为四种范式 [1] 并把预训练语言模型加持下的Prompt Learning看作是近代自然语言处理技术发展的“第四范式”。当我们使用新范式的方法的时候，能够意识到**它带来的优异性可能是以某种“人力”牺牲为代价的**。而**如何让这种人力代价降到最低**，往往就是新范式里需要解决的核心问题 [2]。Prompt Learning刚兴起之时，prompts大多是人工设计的，为了减少人工，后来涌现出一系列用自动化方式获取prompts的研究工作。Soft prompts由于其易用性、易理解性和优异的性能在近一年来获得了广泛的关注。今年四月份，谷歌提出**Prompt-tuning** [3]，其为每个任务分配一个可训练的soft prompt并保持预训练模型的参数不变，发现在使用较大的预训练模型时，Prompt-tuning可以媲美微调整个预训练模型（Model-tuning）的性能。虽然Prompt-tuning非常有效，其仍然面临如下三个问题：

- 在使用**较小的预训练模型**时，Prompt-tuning的表现仍旧与Model-tuning的表现有明显的差距
- 在**Few-shot的场景下**，Prompt-tuning的表现并不理想
- 相比于Model-tuning，Prompt-tuning的**收敛速度较慢**，需要训练更多的步数

幸运的是，这些问题都可以通过**更好地初始化soft prompts**解决！一个直观的想法是可以通过预训练模型词汇表中的一些词来初始化，比如对于分类问题就可以用类别标签对应的词来初始化。但是类别标签词数目有限而且并没有很多任务相关的信息。

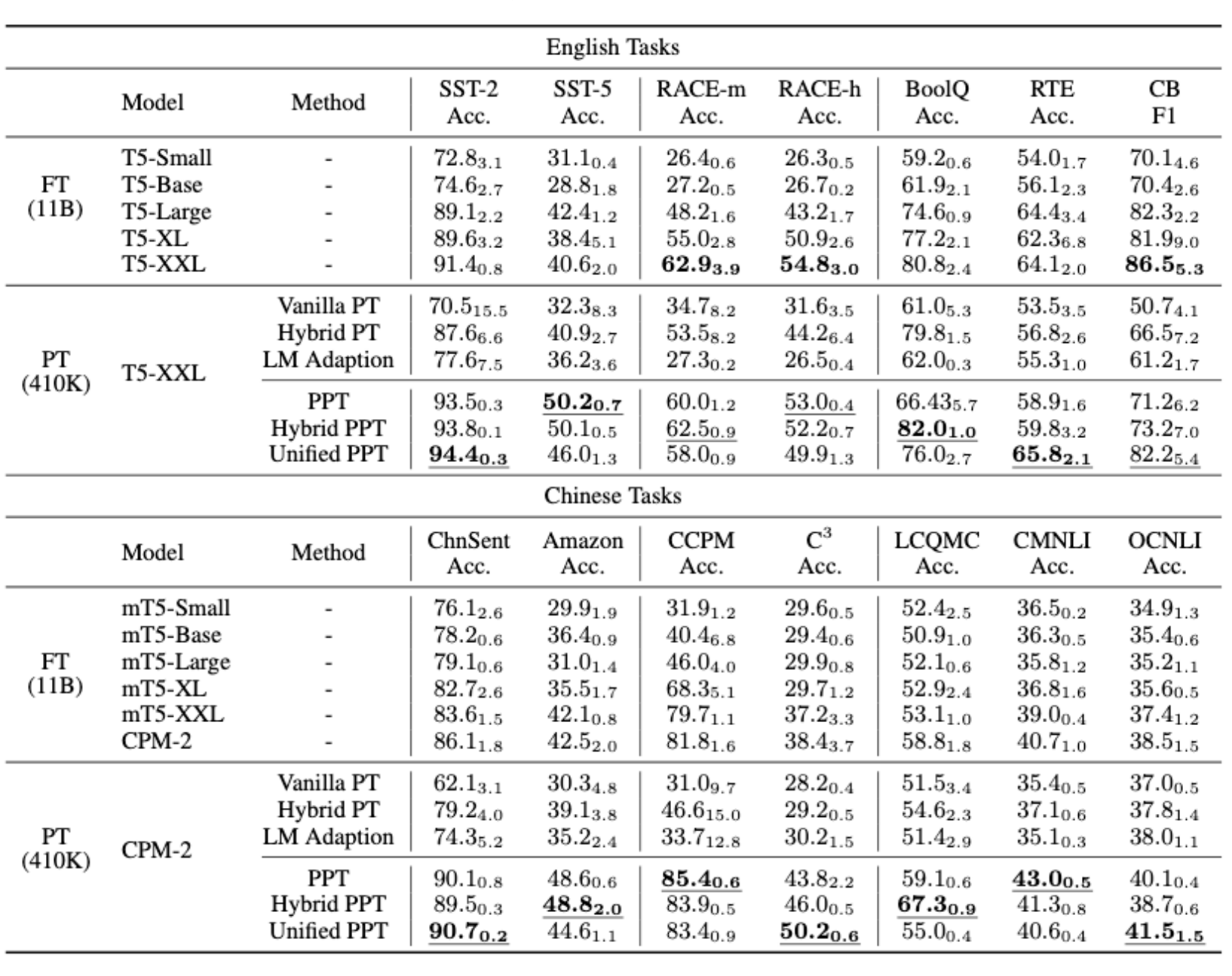
一个更好的方法是：**用在源任务上训练的soft prompts来初始化目标任务的soft prompts**！这个想法有点类似迁移学习的意思。迁移学习通俗来讲，就是运用已有的知识来学习新的知识。核心是找到已有知识和新知识之间的相似性。迁移prompts也类似，如果源任务与目标任务越相似，那么迁移的效果可能就越好。针对prompts的可迁移性问题，谷歌和清华的研究员们进行了初步的探索，笔者接下来为大家一一解读。想要快速浏览 takeaway 干货的读者，可以直接移步文末总结部分。

相关论文：

- *PPT: Pre-trained Prompt Tuning for Few-shot Learning*
  https://arxiv.org/abs/2109.04332
- *SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer*
  https://arxiv.org/abs/2110.07904
- *On Transferability of Prompt Tuning for Natural Language Understanding*
  https://arxiv.org/abs/2111.06719

PPT: Pre-trained Prompt Tuning for Few-shot Learning

清华研究人员发现在Few-shot的场景下，即便使用非常大的预训练模型，相比于Model-tuning，Prompt-tuning的表现仍要差得多。受预训练模型的启发，他们想**对prompts也进行预训练**！他们关注的是分类任务，并把分类任务分成了三类：单句分类任务、句对分类任务和多项选择分类任务。之后，他们针对每一类都设计了一个自监督预训练任务，然后用预训练任务的prompts去初始化下游任务的prompts。此外，他们还提供了一个unified的版本，将所有的分类任务都看作是多项选择分类任务。



English Tasks

	Model	Method	SST-2 Acc.	SST-5 Acc.	RACE-m Acc.	RACE-h Acc.	BoolQ Acc.	RTE Acc.	CB F1
FT (11B)	T5-Small	-	$72.8_{3.1}$	$31.1_{0.4}$	$26.4_{0.6}$	$26.3_{0.5}$	$59.2_{0.6}$	$54.0_{1.7}$	$70.1_{4.6}$
	T5-Base	-	$74.6_{2.7}$	$28.8_{1.8}$	$27.2_{0.5}$	$26.7_{0.2}$	$61.9_{2.1}$	$56.1_{2.3}$	$70.4_{2.6}$
	T5-Large	-	$89.1_{2.2}$	$42.4_{1.2}$	$48.2_{1.6}$	$43.2_{1.7}$	$74.6_{0.9}$	$64.4_{3.4}$	$82.3_{2.2}$
	T5-XL	-	$89.6_{3.2}$	$38.4_{5.1}$	$55.0_{2.8}$	$50.9_{2.6}$	$77.2_{2.1}$	$62.3_{6.8}$	$81.9_{9.0}$
	T5-XXL	-	$91.4_{0.8}$	$40.6_{2.0}$	$\mathbf{62.9_{3.9}}$	$\mathbf{54.8_{3.0}}$	$80.8_{2.4}$	$64.1_{2.0}$	$\mathbf{86.5_{5.3}}$
PT (410K)	T5-XXL	Vanilla PT	$70.5_{15.5}$	$32.3_{8.3}$	$34.7_{8.2}$	$31.6_{3.5}$	$61.0_{5.3}$	$53.5_{3.5}$	$50.7_{4.1}$
		Hybrid PT	$87.6_{6.6}$	$40.9_{2.7}$	$53.5_{8.2}$	$44.2_{6.4}$	$79.8_{1.5}$	$56.8_{2.6}$	$66.5_{7.2}$
		LM Adaption	$77.6_{7.5}$	$36.2_{3.6}$	$27.3_{0.2}$	$26.5_{0.4}$	$62.0_{0.3}$	$55.3_{1.0}$	$61.2_{1.7}$
		PPT	$93.5_{0.3}$	$\mathbf{50.2_{0.7}}$	$60.0_{1.2}$	$53.0_{0.4}$	$66.43_{5.7}$	$58.9_{1.6}$	$71.2_{6.2}$
		Hybrid PPT	$93.8_{0.1}$	$50.1_{0.5}$	$62.5_{0.9}$	$52.2_{0.7}$	$\mathbf{82.0_{1.0}}$	$59.8_{3.2}$	$73.2_{7.0}$
		Unified PPT	$\mathbf{94.4_{0.3}}$	$46.0_{1.3}$	$58.0_{0.9}$	$49.9_{1.3}$	$76.0_{2.7}$	$\mathbf{65.8_{2.1}}$	$82.2_{5.4}$

Chinese Tasks

	Model	Method	ChnSent Acc.	Amazon Acc.	CCPM Acc.	C^3 Acc.	LCQMC Acc.	CMNLI Acc.	OCNLI Acc.
FT (11B)	mT5-Small	-	$76.1_{2.6}$	$29.9_{1.9}$	$31.9_{1.2}$	$29.6_{0.5}$	$52.4_{2.5}$	$36.5_{0.2}$	$34.9_{1.3}$
	mT5-Base	-	$78.2_{0.6}$	$36.4_{0.9}$	$40.4_{6.8}$	$29.4_{0.6}$	$50.9_{1.0}$	$36.3_{0.5}$	$35.4_{0.6}$
	mT5-Large	-	$79.1_{0.6}$	$31.0_{1.4}$	$46.0_{4.0}$	$29.9_{0.8}$	$52.1_{0.6}$	$35.8_{1.2}$	$35.2_{1.1}$
	mT5-XL	-	$82.7_{2.6}$	$35.5_{1.7}$	$68.3_{5.1}$	$29.7_{1.2}$	$52.9_{2.4}$	$36.8_{1.6}$	$35.6_{0.5}$
	mT5-XXL	-	$83.6_{1.5}$	$42.1_{0.8}$	$79.7_{1.1}$	$37.2_{3.3}$	$53.1_{1.0}$	$39.0_{0.4}$	$37.4_{1.2}$
	CPM-2	-	$86.1_{1.8}$	$42.5_{2.0}$	$81.8_{1.6}$	$38.4_{3.7}$	$58.8_{1.8}$	$40.7_{1.0}$	$38.5_{1.5}$
PT (410K)	CPM-2	Vanilla PT	$62.1_{3.1}$	$30.3_{4.8}$	$31.0_{9.7}$	$28.2_{0.4}$	$51.5_{3.4}$	$35.4_{0.5}$	$37.0_{0.5}$
		Hybrid PT	$79.2_{4.0}$	$39.1_{3.8}$	$46.6_{15.0}$	$29.2_{0.5}$	$54.6_{2.3}$	$37.1_{0.6}$	$37.8_{1.4}$
		LM Adaption	$74.3_{5.2}$	$35.2_{2.4}$	$33.7_{12.8}$	$30.2_{1.5}$	$51.4_{2.9}$	$35.1_{0.3}$	$38.0_{1.1}$
		PPT	$90.1_{0.8}$	$48.6_{0.6}$	$\mathbf{85.4_{0.6}}$	$43.8_{2.2}$	$59.1_{0.6}$	$\mathbf{43.0_{0.5}}$	$40.1_{0.4}$
		Hybrid PPT	$89.5_{0.3}$	$\mathbf{48.8_{2.0}}$	$83.9_{0.5}$	$46.0_{0.5}$	$\mathbf{67.3_{0.9}}$	$41.3_{0.8}$	$38.7_{0.6}$
		Unified PPT	$\mathbf{90.7_{0.2}}$	$44.6_{1.1}$	$83.4_{0.9}$	$\mathbf{50.2_{0.6}}$	$55.0_{0.4}$	$40.6_{0.4}$	$\mathbf{41.5_{1.5}}$

从上图可以看到，在Few-shot的场景下，PPT比Prompt-tuning（图中Vanilla PT）的表现明显要好。相比于Model-tuning（图中FT），PPT在所有中文任务上都取得了更好的表现，并在英文任务上取得了类似的性能。上述结果充分显示出预训练prompts在Few-shot场景下的有效性。但是当训练数据增多时，其优势就会越来越小，见下图

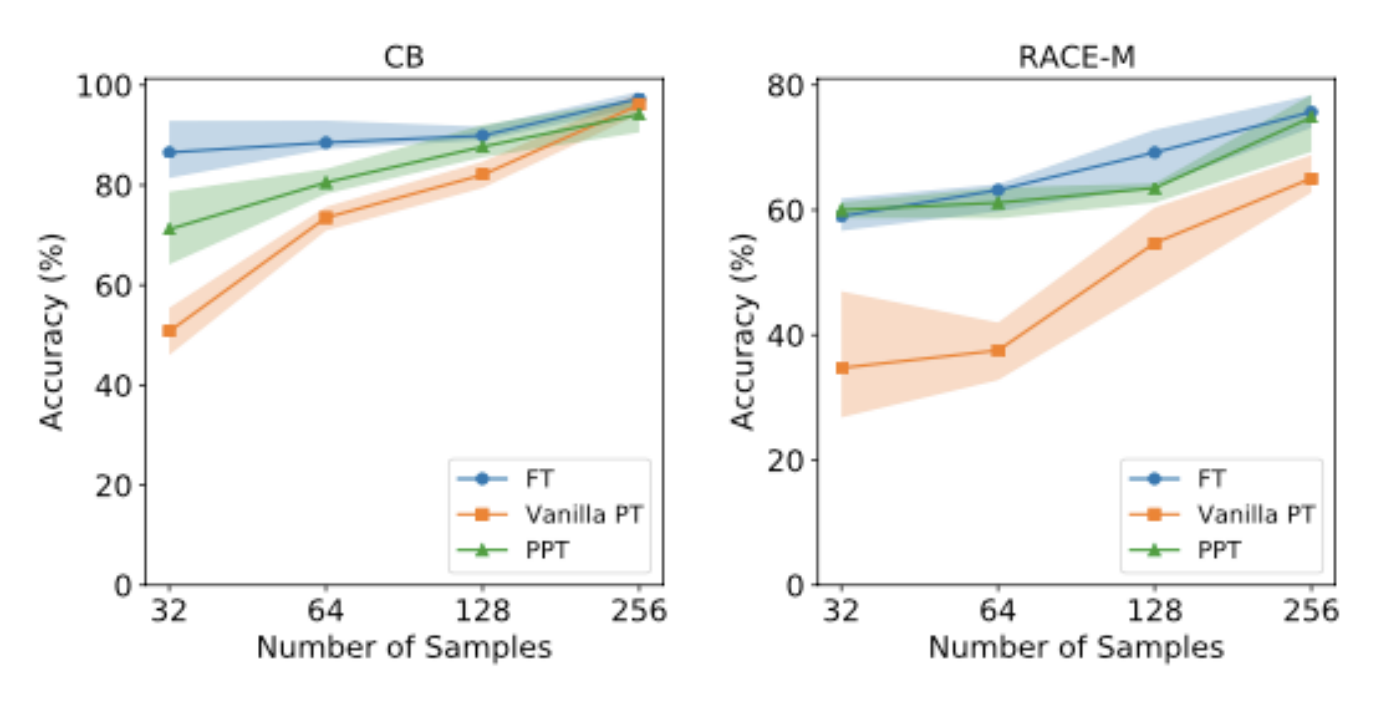
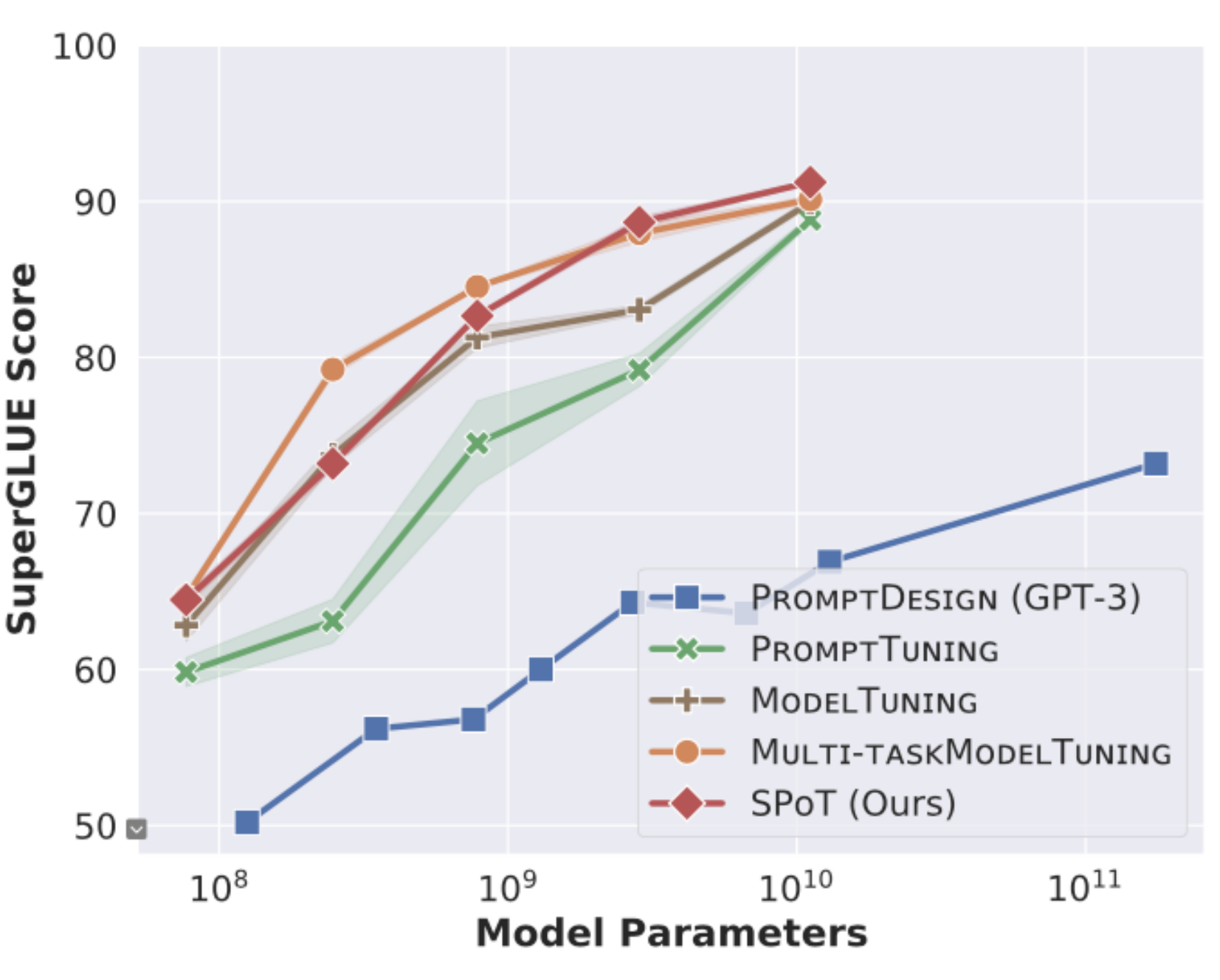


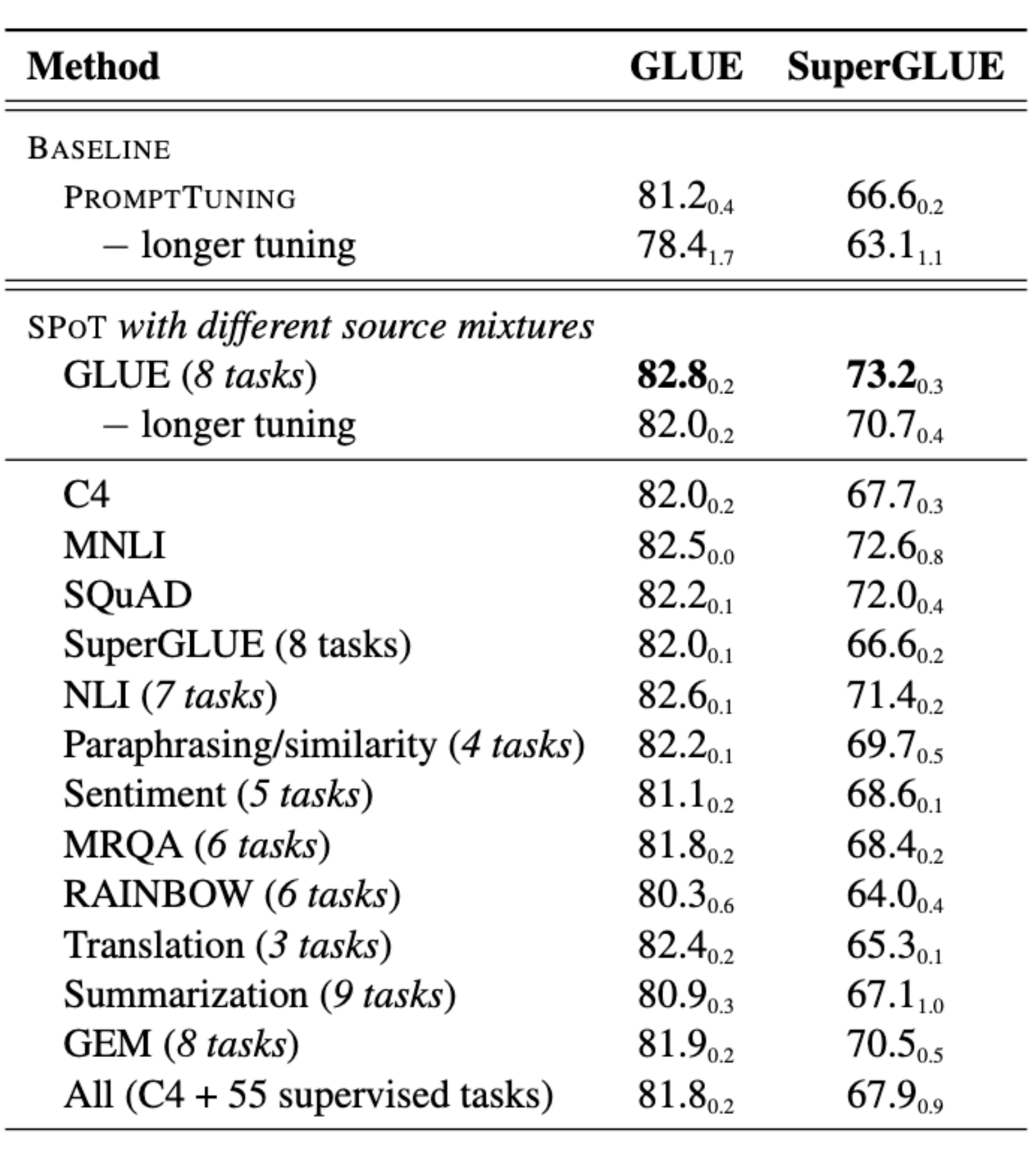
Figure 4: Comparison between full-model fine-tuning (FT), vanilla prompt tuning (Vanilla PT), and pre-trained prompt tuning (PPT) when different numbers training samples are available. For the small number of training samples, PPT is consistently the best. When the number grows, the performance of these methods becomes closer.

SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer

不同于PPT需要人工设计预训练任务，**SPOT用一个或多个源任务训练的prompts来初始化目标任务的prompts**。SPOT在预训练模型比较小时，仍可以达到与Model-tuning相近的性能，显著超越Prompt-tuning， 见下图



此外，SPOT关注的是full-data场景，研究人员发现即便当目标任务的数据充足时，选择合适的源任务训练的prompts仍能够对目标任务带来显著的增益。下图是用不同源任务的实验结果，多个任务是采用数据混合的方式进行训练，然后得到一个prompt。在GLUE的八个任务上训练的prompt在GLUE和SuperGLUE的多个下游任务上得到了最好的表现。值得关注的是使用单个任务如MNLI和SQUAD训练的prompt也带来了明显的提升。

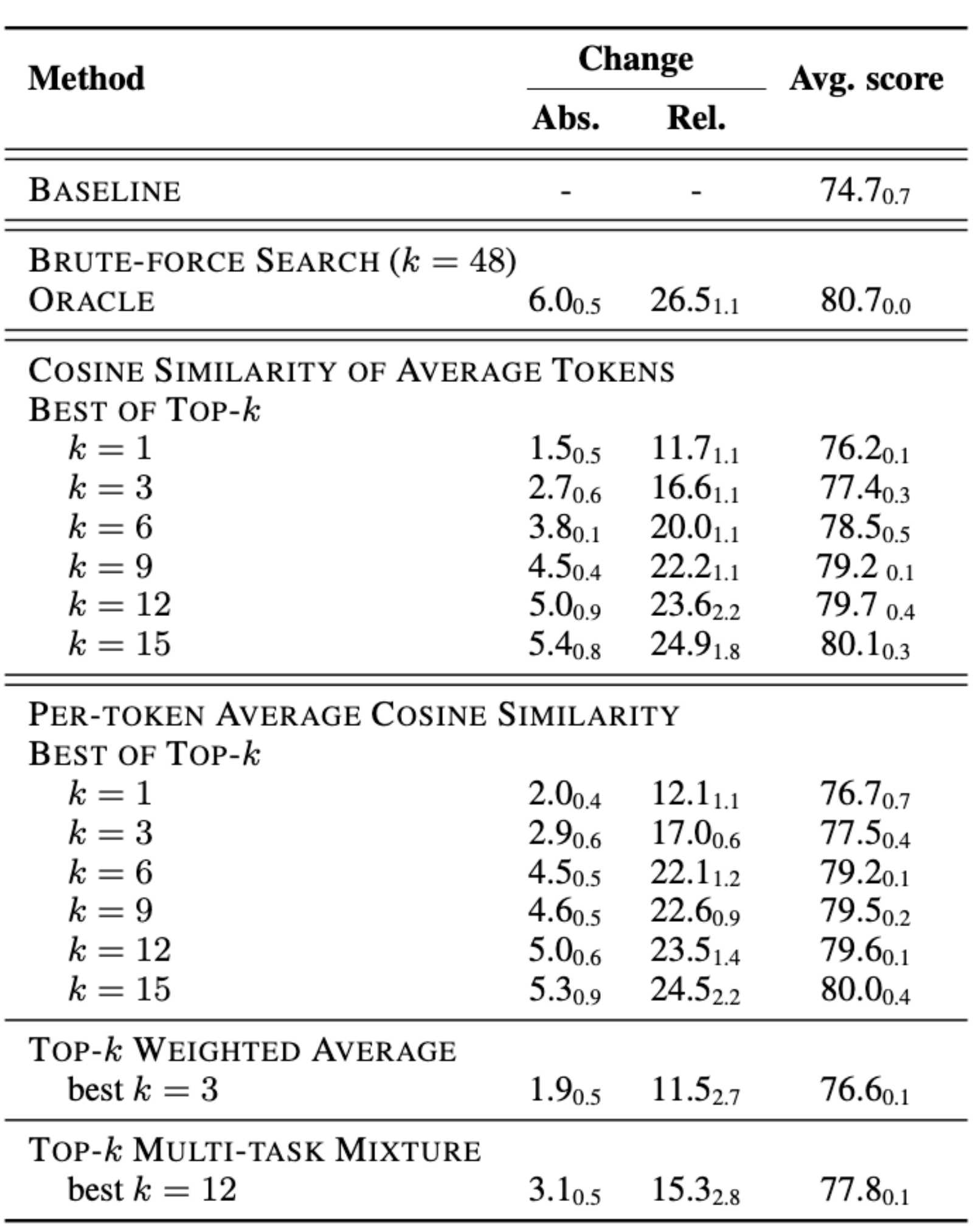


Method	GLUE	SuperGLUE
BASELINE		
PROMPTTUNING	$81.2_{0.4}$	$66.6_{0.2}$
– longer tuning	$78.4_{1.7}$	$63.1_{1.1}$
SPoT *with different source mixtures*		
GLUE (*8 tasks*)	$\mathbf{82.8}_{0.2}$	$\mathbf{73.2}_{0.3}$
– longer tuning	$82.0_{0.2}$	$70.7_{0.4}$
C4	$82.0_{0.2}$	$67.7_{0.3}$
MNLI	$82.5_{0.0}$	$72.6_{0.8}$
SQuAD	$82.2_{0.1}$	$72.0_{0.4}$
SuperGLUE (8 tasks)	$82.0_{0.1}$	$66.6_{0.2}$
NLI (*7 tasks*)	$82.6_{0.1}$	$71.4_{0.2}$
Paraphrasing/similarity (*4 tasks*)	$82.2_{0.1}$	$69.7_{0.5}$
Sentiment (*5 tasks*)	$81.1_{0.2}$	$68.6_{0.1}$
MRQA (*6 tasks*)	$81.8_{0.2}$	$68.4_{0.2}$
RAINBOW (*6 tasks*)	$80.3_{0.6}$	$64.0_{0.4}$
Translation (*3 tasks*)	$82.4_{0.2}$	$65.3_{0.1}$
Summarization (*9 tasks*)	$80.9_{0.3}$	$67.1_{1.0}$
GEM (*8 tasks*)	$81.9_{0.2}$	$70.5_{0.5}$
All (C4 + 55 supervised tasks)	$81.8_{0.2}$	$67.9_{0.9}$

为了探究源任务到目标任务的prompt可迁移性受什么因素影响，作者用16个源任务和10个目标任务构造了160个任务对。对于每一对任务，用源任务训练得到的prompt去初始化目标任务的prompt。他们发现任务之间的相似度是一个重要的影响因素。作者假设不同任务prompts之间的相似度可以反映任务之间的相似度。这样的话，给定一个目标任务就可以去检索相似的源任务，用这些源任务来帮助模型执行目标任务。有如下几种方式

- **Best of top-k**：依次使用top-k源任务的prompts去初始化目标任务的prompt，然后选择表现最好的那个
- **Top-k weighted average**：用top-k源任务的prompts的线性加权去初始化目标任务的prompt，权重就是各个任务与目标任务的prompts的相似度
- **Top-k multi-task mixture**：混合top-k源任务的数据进行训练得到一个prompt去初始化目标任务的prompt

几种不同策略的实验结果见下图

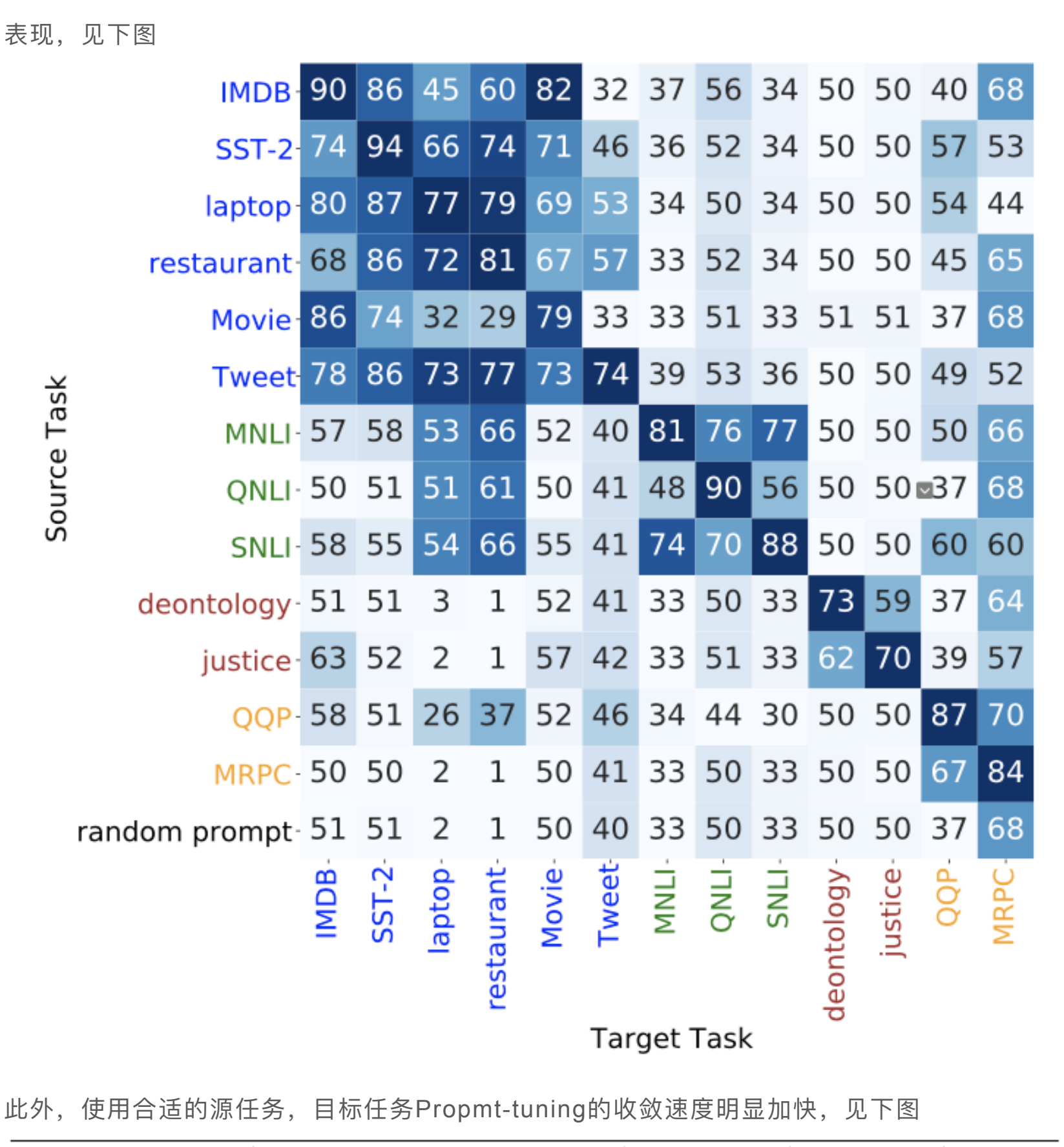


Method	Change		Avg. score
	Abs.	Rel.	
BASELINE	-	-	$74.7_{0.7}$
BRUTE-FORCE SEARCH ($k = 48$)			
ORACLE	$6.0_{0.5}$	$26.5_{1.1}$	$80.7_{0.0}$
COSINE SIMILARITY OF AVERAGE TOKENS			
BEST OF TOP-k			
$k = 1$	$1.5_{0.5}$	$11.7_{1.1}$	$76.2_{0.1}$
$k = 3$	$2.7_{0.6}$	$16.6_{1.1}$	$77.4_{0.3}$
$k = 6$	$3.8_{0.1}$	$20.0_{1.1}$	$78.5_{0.5}$
$k = 9$	$4.5_{0.4}$	$22.2_{1.1}$	$79.2\ _{0.1}$
$k = 12$	$5.0_{0.9}$	$23.6_{2.2}$	$79.7\ _{0.4}$
$k = 15$	$5.4_{0.8}$	$24.9_{1.8}$	$80.1_{0.3}$
PER-TOKEN AVERAGE COSINE SIMILARITY			
BEST OF TOP-k			
$k = 1$	$2.0_{0.4}$	$12.1_{1.1}$	$76.7_{0.7}$
$k = 3$	$2.9_{0.6}$	$17.0_{0.6}$	$77.5_{0.4}$
$k = 6$	$4.5_{0.5}$	$22.1_{1.2}$	$79.2_{0.1}$
$k = 9$	$4.6_{0.5}$	$22.6_{0.9}$	$79.5_{0.2}$
$k = 12$	$5.0_{0.6}$	$23.5_{1.4}$	$79.6_{0.1}$
$k = 15$	$5.3_{0.9}$	$24.5_{2.2}$	$80.0_{0.4}$
TOP-k WEIGHTED AVERAGE			
best $k = 3$	$1.9_{0.5}$	$11.5_{2.7}$	$76.6_{0.1}$
TOP-k MULTI-TASK MIXTURE			
best $k = 12$	$3.1_{0.5}$	$15.3_{2.8}$	$77.8_{0.1}$

可以看到使用不同的策略相比于baseline即随机初始化都能够得到显著的提升。

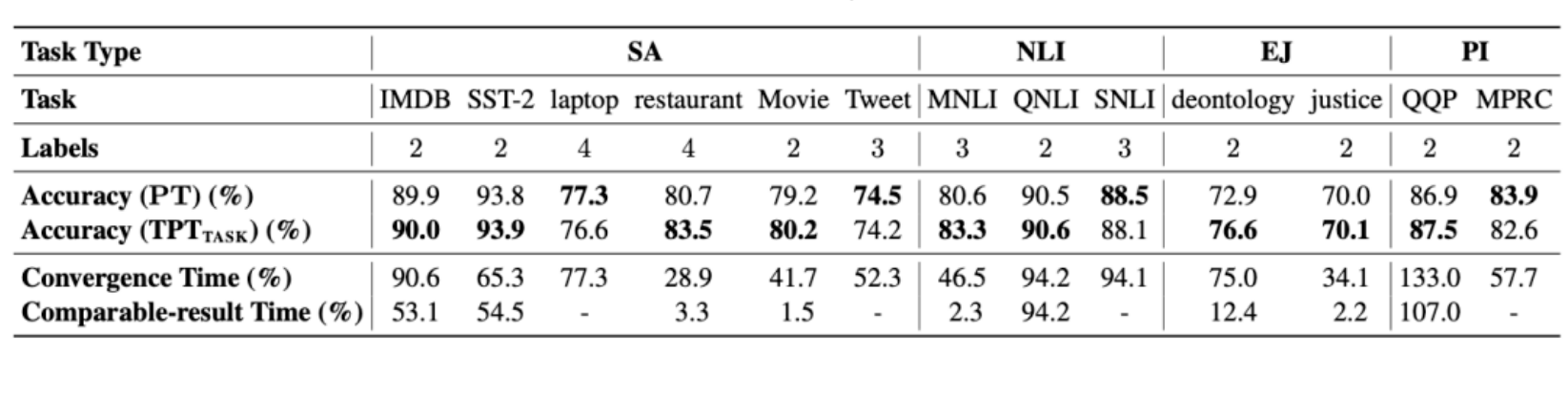
On Transferability of Prompt Tuning for Natural Language Understanding

相比于之前的两篇工作，这个工作定义了更多的衡量prompts之间的相似度的指标。作者还探究了prompts的zero-shot transfer的能力，即在源任务上训练的prompts直接用到目标任务上而不用目标任务的数据finetune，选择合适的源任务可以比使用随机prompts得到明显更好的表现，见下图



Source Task \ Target Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
IMDB	90	86	45	60	82	32	37	56	34	50	50	40	68
SST-2	74	94	66	74	71	46	36	52	34	50	50	57	53
laptop	80	87	77	79	69	53	34	50	34	50	50	54	44
restaurant	68	86	72	81	67	57	33	52	34	50	50	45	65
Movie	86	74	32	29	79	33	33	51	33	51	51	37	68
Tweet	78	86	73	77	73	74	39	53	36	50	50	49	52
MNLI	57	58	53	66	52	40	81	76	77	50	50	50	66
QNLI	50	51	51	61	50	41	48	90	56	50	50	37	68
SNLI	58	55	54	66	55	41	74	70	88	50	50	60	60
deontology	51	51	3	1	52	41	33	50	33	73	59	37	64
justice	63	52	2	1	57	42	33	51	33	62	70	39	57
QQP	58	51	26	37	52	46	34	44	30	50	50	87	70
MRPC	50	50	2	1	50	41	33	50	33	50	50	67	84
random prompt	51	51	2	1	50	40	33	50	33	50	50	37	68

此外，使用合适的源任务，目标任务Propmt-tuning的收敛速度明显加快，见下图



Task Type	SA						NLI			EJ		PI	
Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MPRC
Labels	2	2	4	4	2	3	3	2	3	2	2	2	2
Accuracy (PT) (%)	89.9	93.8	**77.3**	80.7	79.2	**74.5**	80.6	90.5	**88.5**	72.9	70.0	86.9	**83.9**
Accuracy ($\mathrm{TPT_{TASK}}$) (%)	**90.0**	**93.9**	76.6	**83.5**	**80.2**	74.2	**83.3**	**90.6**	88.1	**76.6**	**70.1**	**87.5**	82.6
Convergence Time (%)	90.6	65.3	77.3	28.9	41.7	52.3	46.5	94.2	94.1	75.0	34.1	133.0	57.7
Comparable-result Time (%)	53.1	54.5	-	3.3	1.5	-	2.3	94.2	-	12.4	2.2	107.0	-

总结与展望

基于之前三篇研究工作，我们可以得到如下结论：

- **好的初始化对Prompt-tuning十分重要，可以解决Prompt-tuning的三大问题。Prompt transfer是初始化目标任务prompts的一种非常有效的方式。**
- **源任务与目标任务之间的相似度是迁移有效性的一个重要因素**。可以利用任务的prompts之间的相似度来估计任务的相似度。值得注意的是，这些实验结果都只是显示了一种趋势，**更高的相似度并不能保证更好的迁移**。
- 通过为多个源任务训练prompts可以得到一个prompts库，对于某个目标任务就可以检索一个或者是多个源任务的prompts去初始化目标任务的prompt。目标任务的prompt可以通过训练一定步数得到（不一定要等到收敛），检索可以通过prompts的相似度去进行排序，也可以通过源任务的prompts在目标任务上zero-shot transfer的表现去排序。如果使用多个源任务的prompts，有多种方式，例如逐个使用选最好或者是加权平均。更复杂的方式可能获得更好的性能。
- PPT在目标任务训练数据较多时提升有限，而SPOT仍能有显著的提升。这可能是因为预训练任务与下游任务之间仍旧有显著的差距，更合适的源任务可能是更好的选择。

基于此，我们有如下展望：

- 如果可以将一个任务分解成多个子任务，那么**是否可以训练多个子任务的prompts然后用这些子任务的prompts去得到原任务的prompt**？比如任务型对话系统有多个模块，并不是每个数据集都有所有模块的标注，通过这种方式就可以利用几乎所有现存的数据集，每个prompt只需要编码对应子任务的信息，然后将多个子任务的prompt组合起来就可以为原任务提供很好的初始化。
- 更进一步，**是否可以设计任务相关的prompts，综合考虑原任务与子任务的特点与联系，使得子任务上训练的prompt直接在原任务上复用**？
- **是否可以结合Prompt-tuning和Model-tuning在目标任务上得到更好的表现**？结合第二点展望，**能否将此推广到多任务学习，多个任务之间设计相关的prompts然后一起训练**？

笔者认为Prompt-tuning是一个十分有前景的技术，如何使其更加有效、适用更多场景还需要进一步探索。这就仰仗各位研究者（包括正在读文章的你）的努力啦~~

参考文献

[1] Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. https://arxiv.org/abs/2107.13586

[2] https://mp.weixin.qq.com/s/2U6sk-LzYx4GtRitwEQnBw

[3] The Power of Scale for Parameter-Efficient Prompt Tuning. https://arxiv.org/abs/2104.08691