PJ3 Prompt learning

李菲菲 18307110500

I. 背景介绍

i. 预训练语言模型:

BERT 是一种语言表征模型，旨在通过联合条件作用于所有层中的左右上下文，预训练未标记文本的深度双向表示。BERT 的输入可以来自一个句子，或用特殊 token [SEP]分隔的一对句子，开头为 [CLS]标记。

BERT 的学习目标：两个无监督的预训练任务 Masked language model 和 Next sentence prediction。

- MLM 的训练使用 cloze task，即完形填空任务。具体而言，首先从 input 的 token 序列中随机选取并 mask 15% tokens，然后仅通过 masked 序列的上下文预测原 token，损失为预测的交叉熵损失，最后可以训练得到单个 token 的上下文表征；
- NSP 是指从语料库中选取连续的语句 A、B 作为“IsNext”样本，随机的不连续语句 A、B 作为“NotNext”样本，正例和负例 1：1，联合地学习 text pairs 的表征，最后学习到 token [CLS]的表征，它对应表示整个句子。

Fine-tune: 依据传统的 Pre-train, Fine-tune 训练范式，对于不同的下游任务，需要对 BERT 的所有参数进行微调。例如对于文本分类，数据只有单个序列，输入为文本-[空]文本对；分类是序列级别的任务，因此下游分类器使用[CLS]表征进行分类。

MaksedLM：实验中使用的模型都是 Masked LM，仅使用了完形填空任务预训练。

ii. prompt-based learning:

主要思想：将下游任务转换为完形填空任务，充分利用预训练模型通过预训练任务学习到的能力。

PVP, pattern verbalizer pair, 以文本情感分类任务为例，对 positive/negative 类分别设计对应标签词（verbalizer），预测到某个标签词则对应该类；然后在原句子上加一个带有[MASK]的提示短句，具体的添加方式依据手动选择的模板（pattern），然后使用[MASK]处的输出对 PTM 进行 finetune（few shot）。

II. 实验内容

i. 对比直接微调和基于 prompt 微调

三种实验设置: zero-shot, few-shot32/64/128，根据数据集和模板构造 prompt，生成新的数据集，不额外引入参数；few-shot32，添加分类头。使用的预训练模型为 BertMaskedLM, 除了 zero-shot 外都会对 PTM 进行微调；最后在验证集上计算准确率。

构造 prompt 使用的备选 pattern 和 verbalizer 如下表所示。经过在 zero-shot 上进行的预实验，第一种效果更好。

	Pattern	Verbalizer
Prompt[1]	[X] It is a [Z] film.	Z = {“great”, “bad”}

依据上述的三种实验设置，共进行 5 次实验，验证集准确率如下表所示：

Setting	Prompt	Epoch	Eval-Acc	Time(s/epo)
Zero-shot	√	--	**0.668**	--
Few-shot(32)	√	10	**0.794**	3.961
Few-shot(64)	√	14	**0.802**	8.714
Few-shot(128)	√	14	**0.805**	16.961
Few-shot(128)	×	10	**0.798**	4.041

其中，Prompt 表示构造 prompt 作为输入或者直接微调分类器 header；Epoch 表示训练迭代次数；Eval acc 表示验证集准确率；Time 表示一次迭代训练花费的训练时间；最后一列为每个实验对应的 notebook 文件名，文件内包含实验的过程记录。

ii. Template 和 Verbalizer 的优化

关于 prompt 的优化：比较使用不同的 template 对准确率的影响，使用 32 条训练样本。

设置 1：使用陈述句填空的方法，手动选择 Verbalizer 构造 prompt **设置 2**：模板使用陈述句填空，Verbalizer 可以更新；**设置 3**：构造 Demonstration，在所有训练集中随机选择正、负例各一个样本[2] 。模板示例如下：

	Pattern	Verbalizer	Target	
1	**[X] It is a [Z] film.**	**“great”, “bad”**	**max(Z[Y].logits)**	**Z 为 mask，目标**

				是令 Z 尽可能输出 token Y. 模板是手动选择的。
2	[X] It was [Z] .	“great”, “terrible”	max(Z[Y].logits)	参考[2]
3	[X] It is a [Z] film.	“great”, “bad” (Init)	Yk:=argmin(lossY(Z)); Update Yk max(Z[Y].logits)	每一次迭代，找到令 logit 损失最小的 Y，更新 Y
4	[X]=>It is a [Z] film. [Xneg]=>It is a bad film. [Xpos]=>It is a great film.	“great”, “bad”	max(Z[Y].logits)	随机选择 Xneg 和 Xpos

选择 few-shot 32 的设置，MLM 模型使用 BertMaskedLM 和参数串 Bert-base-uncased，在验证集上的准确率如下所示:

Setting	Prompt Type	Epoch	Eval-Acc	Time/epo
Few-shot(32)	1	10	**0.794**	3.961s
Few-shot(32)	2	10	**0.755**	5.485s
Few-shot(32)	3	10	**0.799**	5.548s
Few-shot(32)	4	10	**0.724**	5.420s

iii. Model

比较不同大小、不同类别的预训练模型，使用基于 prompt 的方法在验证集上的准确率：

Model	Size	Zero-shot	FewShot32	Train/epo	
Bert	Base	0.668	0.794	3.961s	
	Large	0.622	--	--	使用 Prompt type 1
Roberta	Base	0.796	0.847	7.180s	
	Large	0.851	--	--	
Albert	Base	0.701	--	--	

	Large	0.647	--	--

iv. Full data

使用全部数据的情况下，比较直接微调和基于提示微调的表现。

直接微调：在 BertMaskedLM 后添加分类头，基于 MLM 的隐藏层进行分类。

	Train data	Eval acc	Train/epo	Epo
Fine-tune	8596	0.904	256.248s	3

III. 实验总结

1. few-shot 设置下，随着可用训练集增多，单一格式的 prompt learning 表现会更好；随着训练迭代次数增加，通常会在 10 个 epoch 内达到过拟合。
2. zero-shot 设置下，BERT 本身预训练的 MLM 的能力起到决定因素：相对其他 BERT 的变体，Roberta Masked LM 表现最好。
3. Roberta 在 zero-shot 和 few-shot 的各种设置下表现都优于其他变体，原因在于：该模型的预训练任务语料更多、使用动态 Masking 机制，预训练任务是一种增强的完形填空任务，因此对 mask 预测更准确。
4. few-shot 设置下，在 pattern 和 verbalizer 中，模型对 verbalizer 的变化更加敏感，原因在于，基于 verbalizer 计算损失微调，会影响一组近义词的输出概率。
5. 不足之处：关于损失的设计考虑不够充分。实验中抽取出特定 verbalizer 对应的 logits 分数，softmax 后直接作为二分类的预测值计算损失，在模型太大的情况下无法训练，小样本微调时效果不稳定。

IV. 参考

[1] Schick, Timo and Hinrich Schütze. “It’s Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners.” ArXiv abs/2009.07118 (2020): n. pag.

[2] Gao, Tianyu, Adam Fisch, and Danqi Chen. "Making pre-trained language models better few-shot learners." arXiv preprint

arXiv:2012.15723 (2020).