

加了元学习之后，少样本学习竟然可以变得这么简单！

原创 Rukawa\_Y 夕小瑶的卖萌屋 2021-12-19 22:05



文 | Rukawa\_Y  
编 | Sheryc\_王苏, 小铁

去年年初 GPT-3 的论文在 arxiv 上出现，论文名为 "Language Models are Few-Shot Learners"，引起一阵轰动。除了前无古人的模型规模外，最抓人眼球的是，GPT-3 能够不需要 fine-tuning 进一步更新参数，直接解决下游任务。

这种完全依赖语言模型从预训练过程中习得的推理能力，通过上下文语境（task description）直接解决新任务的方式，叫做语境学习（in-context learning）。然而利用语境学习让模型学会处理一个新的任务的方法效果往往比不上传统的微调，预测的结果也有很大的方差，并且如何把多种多样的任务转化为语境的模板（Prompt）是很难被设计出来的。

近期，Facebook AI 实验室所发表的论文 MetalCL: Learning to learn in context，提出了一种训练方法 MetalCL，通过元学习的方式让模型更加简单有效地进行语境学习——不需要在输入中提供任务描述模板，只需提供训练样例和目标输入。论文作者希望通过在元训练任务上进行多任务学习，让模型能够自动学习到如何通过输入中的少量训练样本重构任务信息，从而省去人工设计模板的麻烦。

## 论文标题

MetalCL: Learning to Learn In Context

## 论文链接

https://arxiv.org/pdf/2110.15943.pdf

Arxiv 访问慢的小伙伴也可以在 【夕小瑶的卖萌屋】 订阅号后台回复关键词 【1219】 下载论文 PDF~

## 语境学习的几种情况

根据 GPT-3 的论文，语境学习分为 3 种情形，分别是：零样本学习（Zero-shot Learning）、单样本学习（One-shot Learning）和少样本学习（Few-shot Learning）。

### The three settings we explore for in-context learning

#### Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French:
2 cheese =>
```

#### One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese =>
```

#### Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese =>
```

- 零样本学习（Zero-shot Learning）：在模型输入中只提供任务描述和测试样例输入，得到测试样例输出；
- 单样本学习（One-shot Learning）：在模型输入中提供任务描述、一条训练样例和测试样例输入，得到测试样例输出；
- 少样本学习（Few-shot Learning）：在模型输入中提供任务描述、少量训练样例和测试样例输入，得到测试样例输出。

GPT-3 的论文的实验结果中，表明 GPT-3 这三种情况均取得了一定的效果，这里不再展开了，有兴趣的同学可以看看 GPT-3 的论文。

## MetalCL 介绍

了解了语境学习之后，相信大家都会被这种通过推理就能学习一个新的任务的，并且无需任何参数更新的学习方式所吸引。然而利用语境学习让模型学会处理一个新的任务的方法效果往往比不上传统的微调，预测的结果也有很大的方差，并且如何把多种多样的任务转化为语境的模板（Prompt）是很难被设计出来的。基于这些挑战，也就是如何让模型在语境学习中所学习到的新的任务效果更好，MetalCL 应运而生。

MetalCL，全称 Meta-training for In-context Learning，其中先对模型进行元训练（Meta-training），让模型学会如何根据数据集自动调整语境学习策略，然后再进行语境学习。在 Meta-training 的过程中，论文的作者使用了数十个不同领域的 NLP 的任务作为元训练任务（Meta-training Task）对于模型进行训练，来提升模型进行少样本学习的能力。在这一训练过程中，MetalCL 的优点在于不需要在输入中提供任务描述模板，只需提供训练样例和目标输入：论文作者希望通过在元训练任务上进行多任务学习的策略，让模型能够自动学习到如何通过输入中的少量训练样本重构任务信息，从而省去人工设计模板的麻烦。

MetalCL 的具体的训练过程是：对于每一个元训练任务，利用 K-shot Learning，将元训练任务中随机抽取出的任务中的  $k$  个样本的  $x$  和  $y$ ，和第  $k+1$  个样本的  $x$  连接起来作为一个输入，输入到模型中，预测对应的第  $k+1$  个样本的  $y$ 。其中，前  $k$  个样本相当于提供了对于任务的描述信息。MetalCL 推测的过程也采用了与训练过程相同的输入方式来处理预测的样本：不需要任务描述，只需将该任务的  $k$  个训练样本与目标输入相拼接即可。

总结一下 MetalCL 的流程如下：

- Meta-training
  - 从准备好的大量的元训练任务中，随机抽取出一个任务  $i \in [1, C]$ ；
  - 从任务  $i$  中抽取  $k+1$  个样例  $(x_1, y_1), (x_2, y_2), \dots, (x_{k+1}, y_{k+1})$ ；
  - 利用模型最大化概率  $P(y_{k+1}|x_{k+1}, x_1, y_1, \dots, x_k, y_k)$ ，loss 损失函数为交叉熵。
- Inference
  - 对于一个 meta-training 中没有见过的任务，获取  $k$  个样例  $(x_1, y_1), \dots, (x_k, y_k)$  和预测的输入  $x$ ；
  - 取  $\text{argmax}_{c \in C} P(c|x_1, y_1, \dots, y_k, x)$ ，其中  $C$  是输出  $y$  所有可能的结果的集合。

除此之外，论文的作者还借鉴了在少样本的文本分类任务中的一种噪声通道[1]的方式，将元训练过程中的目标函数变换为  $p(x|y)$ ，即在噪声通道模型中， $P(y|x) = \frac{P(y)p(P(y))}{P(x)} \propto P(x|y)P(y)$ ，将  $P(y)$  定义为  $\frac{1}{|C|}$ ，那么目标函数则变换为最大化概率  $P(x|y)$ ，即最大化概率  $P(x_{k+1}|y_{k+1}, x_1, y_1, \dots, x_k, y_k)$ ，对应推理阶段模型则是计算  $\text{argmax}_{c \in C} P(c|y_1, x_1, \dots, y_k, x, c)$ 。

## 实验结果

论文中作者使用 GPT-2 large 进行 MetalCL 训练，元训练过程中一共使用了 142 个元训练数据集，这 142 个数据集包含了文本分类、QA、NLI 和 paraphrase detection 等任务。论文作者在实验过程中对于 Meta-training 采取了 7 种设置，对 Target 采取了 5 种设置，并且在 Meta-training 和 Target 所使用的任务是没有交集的，具体的设置如下：

- HR-LR：从大数据集迁移至小数据集，在 meta-training 过程中使用 HR（High resource）即大于等于 10000 训练样本的数据集，在 target 的过程中使用 LR（Low resource）即小于 10000 的数据集。
- Classification/QA：元训练与预测使用相似任务，在 meta-training 过程中使用 Classification 和 QA 的数据集，在 target 的过程中也使用 Classification 和 QA 的数据集。
- NLI/Paraphrase：元训练与预测使用不相似任务，在 meta-training 过程中使用 Non-Classification、Non-QA、Non-NLI、Non-Paraphrase 的数据集，在 target 过程中使用 Classification、QA、NLI、Paraphrase 的数据集。

Meta-train 和 Target 所采取的设置总结如下，见表格：

Meta-train			Target		
Setting	# tasks	# examples	Setting	# tasks	
HR	61	819,200	LR	26	
Classification	43	384,022	Classification	20	
Non-Classification	37	368,768			
QA	37	486,143	QA	22	
Non-QA	33	521,342			
Non-NLI	55	463,579	NLI	8	
Non-Paraphrase	59	496,106	Paraphrase	4	

最终实验结果里作者将 MetalCL 和 Channel MetalCL 同其他 baseline 做了比较，实验结果如下：

Method	HR→LR	Class→Class	non-Class→Class	QA→QA	non-QA→QA	non-NLI→NLI	non-Para→Para
0-shot	34.9	34.0	34.0	39.9	39.9	25.7	36.5
PMI 0-shot	36.1	34.9	34.9	37.7	37.7	36.6	35.0
Channel 0-shot	40.0	42.4	42.4	40.4	40.4	31.4	37.3
In-context	36.5/34.7	36.0/33.2	36.0/33.2	39.6/38.4	39.6/38.4	26.4/25.6	33.1/33.1
PMI In-context	32.4/32.0	32.4/23.0	32.4/23.0	37.6/36.4	37.6/36.4	32.6/27.8	34.0/32.9
Channel In-context	42.0/36.9	45.2/38.4	45.2/38.4	40.2/37.6	40.2/37.6	39.4/33.3	44.4/41.7
Multi-task 0-shot	41.9	37.4	36.9	45.3	35.6	42.4	36.7
Channel Multi-task 0-shot	38.9	42.6	42.7	41.4	35.8	39.0	47.2
MetalCL	45.6/43.1	43.7/40.1	38.1/33.7	43.4/41.7	38.5/37.0	51.4/48.1	35.1/33.2
Channel MetalCL	47.8/43.0	47.1/42.9	45.8/40.9	41.2/38.5	40.3/37.5	50.7/44.3	51.3/47.9
Oracle	46.4/40.0	50.7/44.0	50.7/44.0	41.8/39.1	41.8/39.1	44.3/32.8	54.7/48.9
Oracle w/ meta-train	52.0/47.9	53.5/48.5	51.2/44.9	46.7/44.5	41.8/39.5	57.0/44.6	53.7/46.9
Target tasks in unseen domains							
0-shot	33.9	33.9	33.9	44.7	44.7	34.9	47.3
PMI 0-shot	24.5	24.5	24.5	22.8	22.8	49.7	37.1
Channel 0-shot	31.0	31.0	31.0	44.1	44.1	32.9	34.6
In-context	29.8/26.9	29.8/26.9	29.8/26.9	44.4/42.5	44.4/42.5	33.3/33.5	34.1/34.1
PMI In-context	27.8/21.1	27.8/21.1	27.8/21.1	22.8/22.8	22.8/22.8	44.8/36.1	33.1/32.6
Channel In-context	37.5/31.3	37.5/31.3	37.5/31.3	45.4/40.0	45.4/40.0	40.2/35.7	45.4/40.7
Multi-task 0-shot	33.4	31.5	27.9	65.9	29.1	34.6	46.0
Channel Multi-task 0-shot	32.1	27.7	33.3	51.6	43.8	59.4	53.5
MetalCL	41.1/37.4	40.0/36.5	33.6/28.1	58.7/56.2	38.1/36.6	80.3/77.7	42.0/34.4
Channel MetalCL	40.2/34.1	41.3/36.7	41.4/38.1	50.8/49.1	45.1/41.6	56.7/41.5	48.2/43.2
Oracle	44.9/37.6	44.9/37.6	44.9/37.6	43.6/39.1	43.6/39.1	56.3/33.4	56.6/51.6
Oracle w/ meta-train	53.3/43.2	53.2/43.7	46.1/36.9	67.9/66.2	44.5/42.8	71.8/58.2	65.6/61.4

表格中各个 method 的意义如下：

- 0-shot：使用预训练语言模型直接做零样本推断
- In-context：使用预训练语言模型进行 few-shot 的推断
- PMI 0-shot，PMI In-context：使用 PMI [2] 方式的零样本和 few-shot 的推断
- Channel 0-shot，Channel In-context：使用通道模型的零样本推断和使用通道模型的 few-shot 推断
- Multi-task 0-shot：使用 meta-training 对于预训练模型进行训练然后进行零样本推断
- Channel Multi-task 0-shot：使用通道模型的 Multi-task 0-shot
- Oracle：基于任务对于预训练模型进行 fine-tuning
- Oracle w/meta-train：对预训练语言模型进行 meta-training，然后再进行 fine-tuning

从表格中我们可以得到如下结论：

对于模型来说，在元训练/目标任务相似，即不进行任务迁移的情况下：

- 使用 Meta-training 训练模型然后进行 0 样本的任务学习（Multi-task 0-shot）的效果比普通的语言模型直接进行语境学习的效果更好，说明 Meta-training 的确是对于模型是有帮助的。
- 使用 MetalCL 训练的模型，不论是普通的语言模型还是 Channel 语言模型均取得了不错的效果，尤其是 Channel MetalCL 在大部分的实验设置的情况下效果是最好的。

在需要进行任务迁移的情况下：

- 使用 MetalCL 训练的模型的效果并没有收到太大的影响，并且仍然在大部分的实验设置的情况下效果是最好的。
- 而将 MetalCL 训练的模型同 Fine-tuning 的模型比较时：
- 使用 MetalCL 的方法有时候能够取得超过直接在目标任务进行 Fine-tune 的方法，并且 Oracle w/meta-train 的结果还说明了 meta-training 对于 Fine-tuning 也有帮助。

除了上面的结论，作者还将使用 MetalCL 训练的 GPT-2 模型同 GPT-J 进行了比较，实验结果如下：

Method	Diverse	No Diverse
0-shot		34.9
PMI 0-shot		36.1
Channel 0-shot		37.7
In-context		36.5/34.7
PMI In-context		36.3/30.4
Channel In-context		42.0/37.3
Multi-task 0-shot	39.2	35.2
Channel Multi-task 0-shot	40.1	37.2
MetalCL	44.6/41.7	37.6/34.3
Channel MetalCL	44.3/40.5	42.1/38.0

- 分析了任务描述是否对于 MetalCL 有帮助：虽然 MetalCL 不需要任务描述，论文的作者依然使用了在 MetalCL 中使用任务描述是否会带来进一步的性能提升。作者在 HR->LR 实验中使用了 32 个 meta-training 任务和 12 个目标任务，这些任务均是提供了人工撰写的任务描述，可以看出加入任务描述后 MetalCL 效果有着进一步的提升。

Method	w/o Instruct	w/ Instruct
# instruct/task	0	1 8.3
All target tasks		
0-shot	33.3	34.7
PMI 0-shot	33.1	38.5
Channel 0-shot	33.2	30.2
In-context	32.7/30.3	39.7/36.9
PMI In-context	34.4/29.9	42.6/34.8
Channel In-context	37.8/34.2	41.0/37.0
MT 0-shot	39.1	37.2
Channel MT 0-shot	35.9	32.8
MetalCL	37.0/34.2	43.2/40.0
Channel MetalCL	38.5/35.8	43.1/39.1
Target tasks in unseen domains		
0-shot	33.9	29.5
PMI 0-shot	24.5	32.8
Channel 0-shot	31.0	30.3
In-context	29.8/26.9	39.4/35.7
PMI In-context	27.8/21.1	46.3/29.1
Channel In-context	37.5/31.3	41.2/34.9
MT 0-shot	32.3	31.6
Channel MT 0-shot	31.0	27.1
MetalCL	31.5/26.8	47.0/43.2
Channel MetalCL	39.7/36.6	45.4/37.2

## 总结

本文介绍了一种 MetalCL 的训练方法，通过这种训练方法可以让模型通过元训练更加简单地进行语境学习。同时笔者认为 MetalCL 也算是一种大力出奇迹的方法吧。这其实可以给予我们一些启发：

- 对于一个语言模型，不论要使用这个语言模型进行什么样的任务，当让模型尽量多的见识各种各样的任务之后（这里什么任务能够对应模型提升比较大也是值得继续去研究的），即使对于没见过任务，不论是零样本学习还是语境学习，模型均能取得不错的效果。
- MetalCL 算是一种基于大规模语言模型的一种训练方法，对于容量较小的模型，笔者认为 MetalCL 是否还能起到很好的效果也是值得去探究的。
- 论文中还提出对于一个语言模型进行 MetalCL 之后，再进行 Fine-tuning，最终模型的效果是比直接对于模型进行 Fine-tuning 要好的，在拥有大量不同任务的数据集时，可以考虑在 Fine-tuning 之前先对模型进行 MetalCL 训练，或许可以进一步提升模型效果。

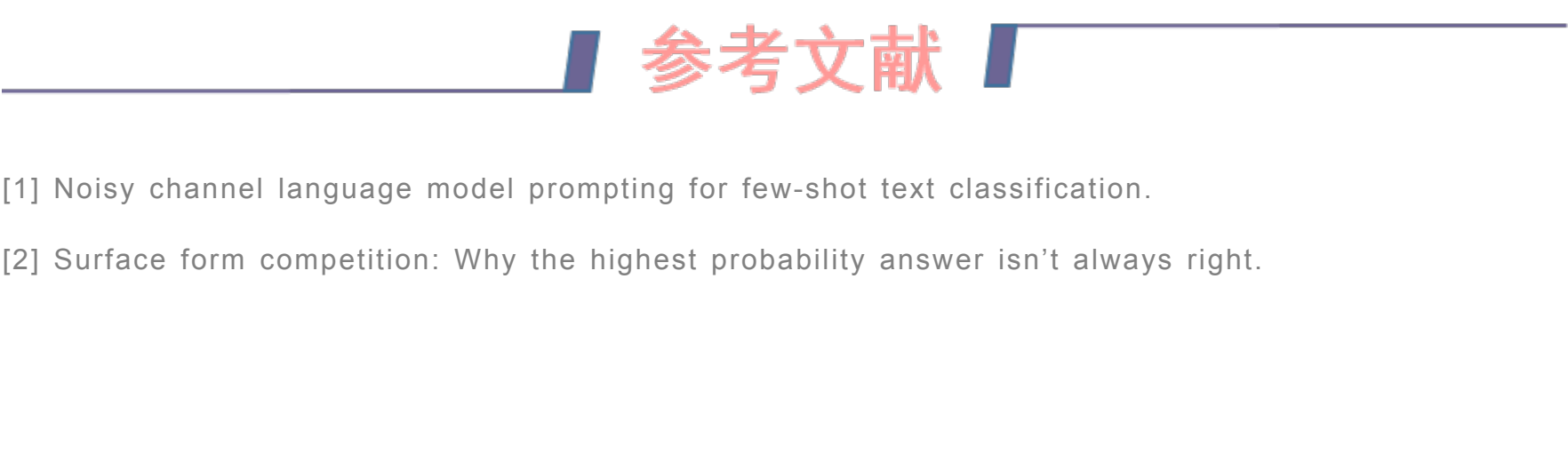


后台回复关键词【入群】

加入卖萌屋 NLP/IR/Roc 与求职讨论群

后台回复关键词【约会】

获取 ACL、CIKM 等各大顶会论文集！



## 参考文献

- Noisy channel language model prompting for few-shot text classification.
- Surface form competition: Why the highest probability answer isn't always right.

喜欢此内容的人还喜欢

新春特辑 | 机器学习在化生相关领域的应用  
王初课题组

如何提高暂态稳定评估模型在拓扑变化场景下的泛化能力？  
中国电机工程学报

推荐 | Pair，医学图像标注神器  
小白学视觉