预训练大模型的应用技术

蒋芳清

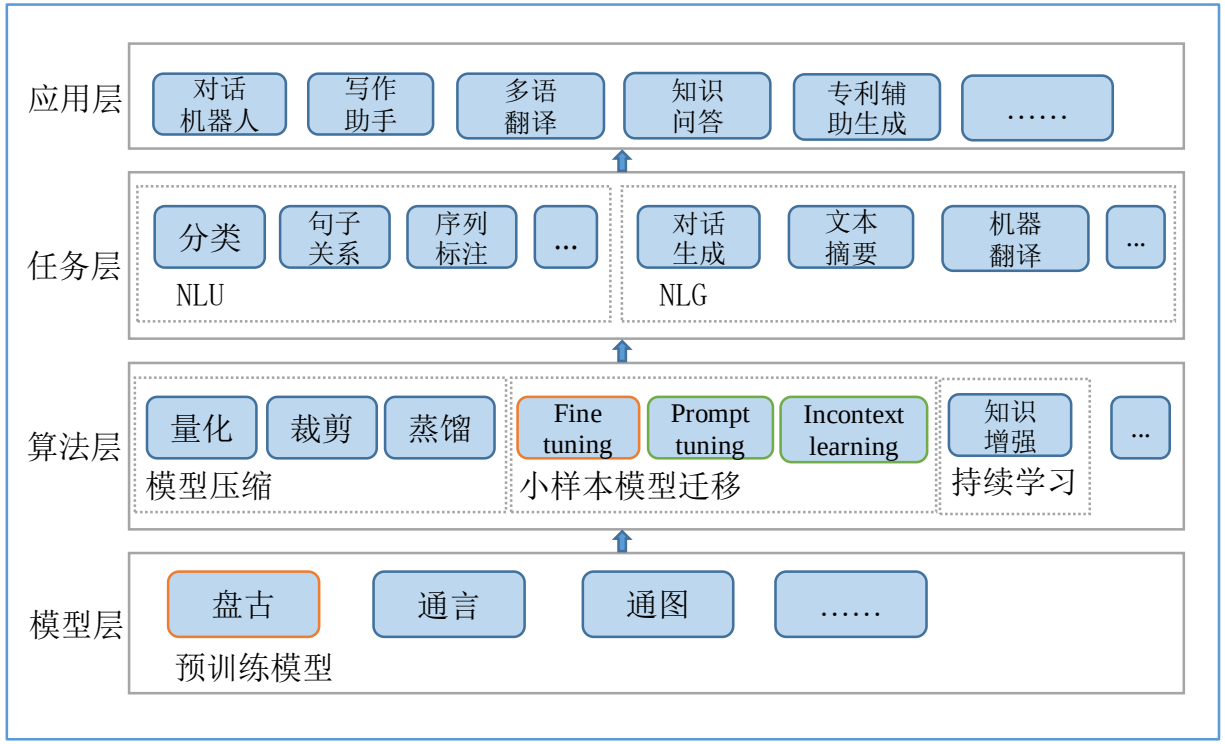
鹏城实验室网络智能研究部开源所

CONTENTS

CONTENTS

模型应用-基本框架

基本框架



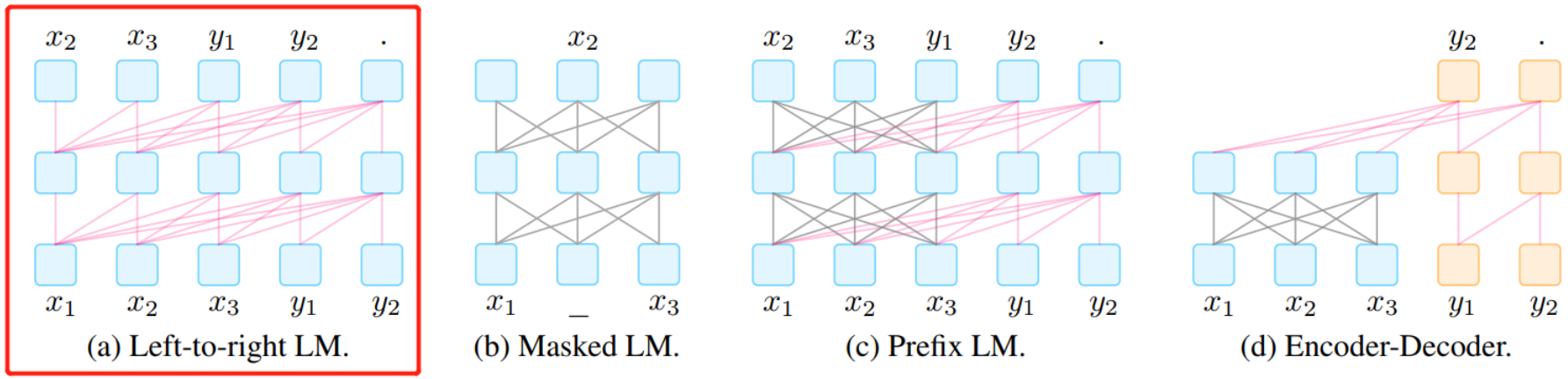
- **模型层**：基于分布训练模架、大数据、高效算法训练大模型，构建文本、多语言、多模型、知识图谱等领域的AI底层基础设施。
- **算法层**：基于预训练模型做模型压缩、小样本模型迁移、持续学习等方面的算法创新，构建基础算法模块，为预训练模型落地应用提供底层算法支撑。
- **任务层**：基于算法层基础算法模块，构建两大基础任务NLU和NLG的实现样例，为上层应用提供底层任务建模支撑。
- **应用层**：基于任务层基础任务，设计对话机器人、写作助手、多语翻译、知识问答、专利辅助生成等应用，为预训练模型落地应用提供示范应用，促进模型的加速落地和深度应用。

：已实现
：正在实现
：规划中

模型应用-基本框架

■ 模型层-盘古模型

网络结构



预训练任务

预训练目标	描述	应用
标准语言模型LM	文本以自回归方式预测，从左到右依次预测序列中的词。	NLG
噪声文本重建CTR	输入句子引入噪声，将处理后的文本恢复到未损坏的状态。	NLU
全文本重建FTR	计算整个输入文本的损失来重构文本，无论输入文本是否有噪声。	NLU&NLG

辅助目标	描述	
NSP	下一句预测，二分类	
SOP	句子顺序预测，二分类	

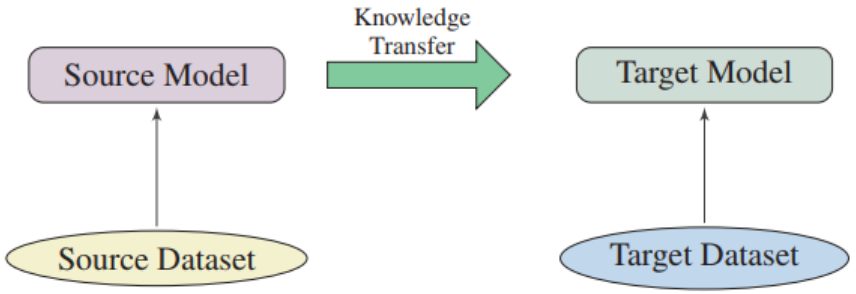
CONTENTS

模型应用-小样本学习

什么是小样本学习

人类非常擅长通过极少量的样本识别一个新物体，比如小孩子只需要书中的一些图片就可以认识什么是“斑马”，什么是“犀牛”。
在人类的快速学习能力的启发下，研究人员希望机器学习模型在学习了一定类别的大量数据后，对于新的类别，只需要少量的样本就能快速学习，这就是小样本要解决的问题。

大规模预训练模型下的小样本学习

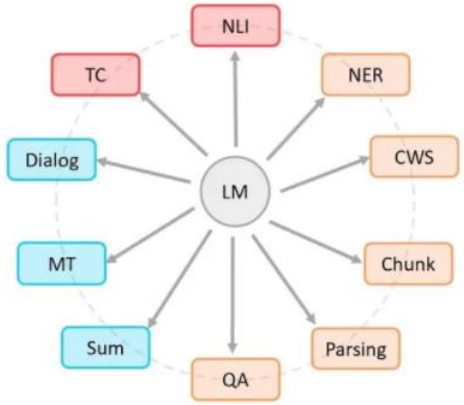


预训练模型通过大量数据训练学习到广泛的通用知识和知识表示，直接将其应用到特定领域任务并不能得到好的性能。

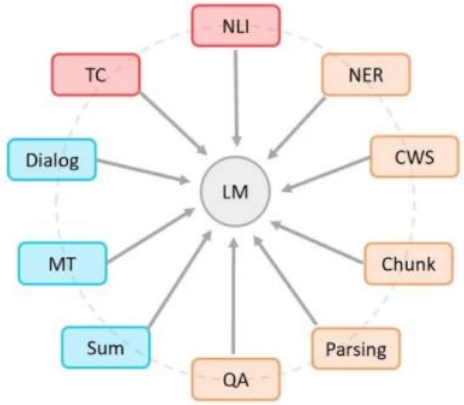
结合预训练语言模型通用和强大的泛化能力基础上，探索通小样本学习将预训练模型的知识和表示迁移到特定领域任务。

模型应用-小样本学习

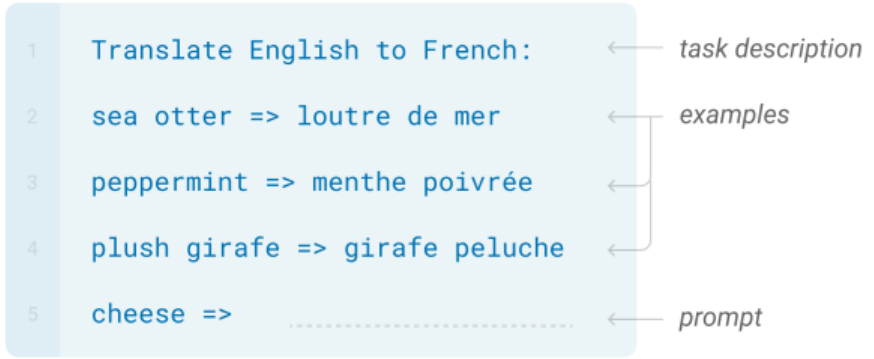
- 技术方向



Fine-tune



Prompt-tune



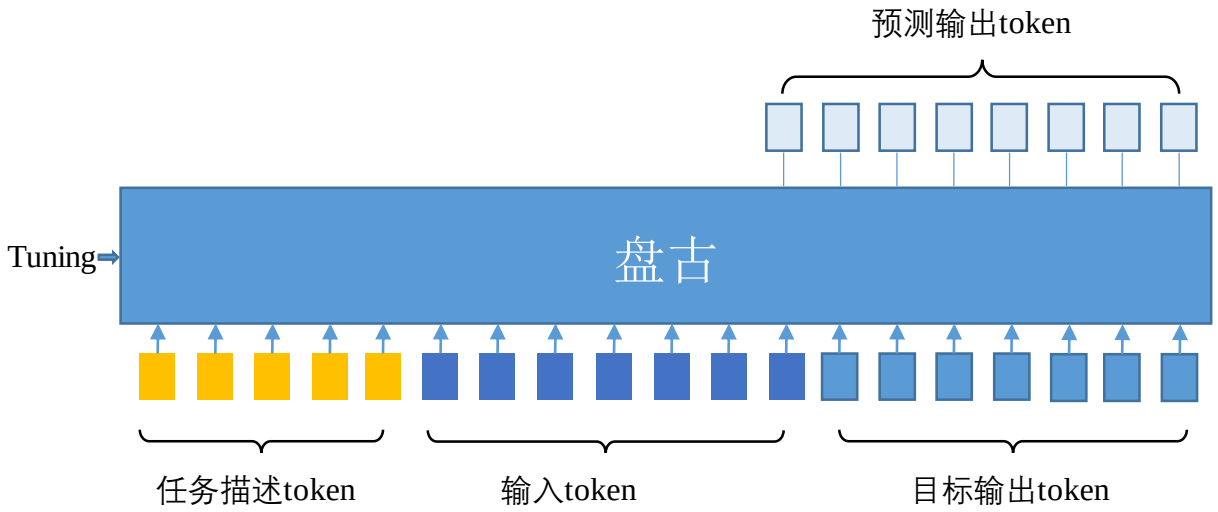
In-context learning

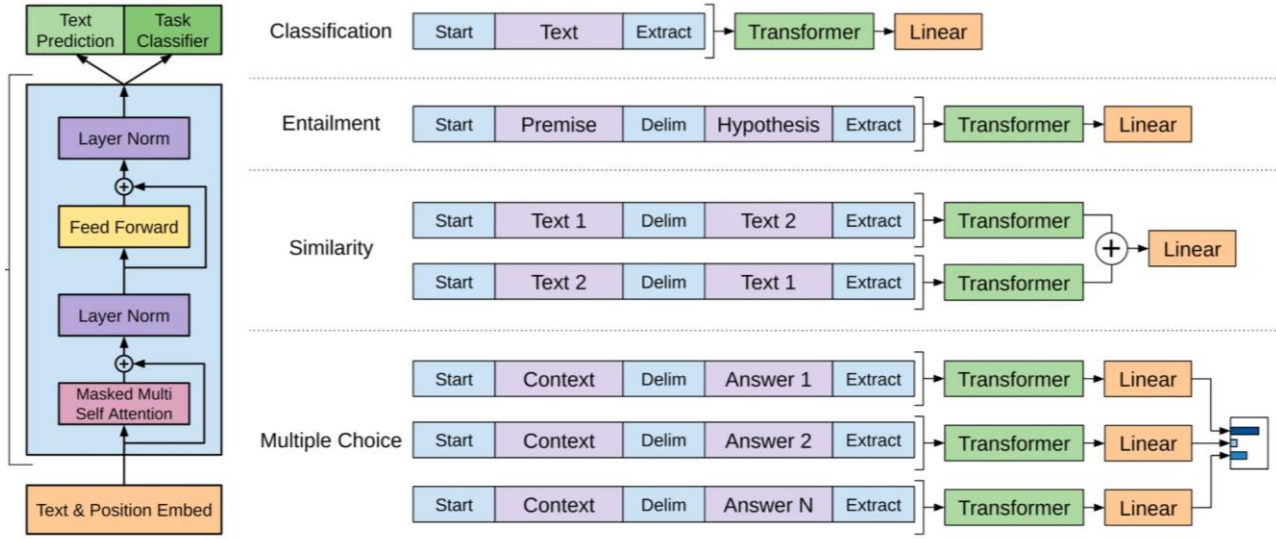
类别	工程	描述
Fine-tune	Objective Tuning	预训练微调范式，通过引入较多额外的参数并使用特定任务的目标函数对预训练模型进行微调，将预训练语言模型适应于不同的下游任务。
Prompt-tune	Prompt + Tuning	prompt-tune即预训练加提示的范式，这种范式不是让预训练的语言模型适应下游任务，而是形式化下游任务使之在提示(prompt)的帮助更适合预训练模型的原始任务来解决问题。
In-context learning	Prompt + Examples	上下文学习，通过给模型输入任务提示和相关示例使用预训练语言模型，无需重新训练模型和增加额外参数，更接近人类解决新任务的机制。

模型应用-小样本学习

盘古模型—Fune-tune

预训练阶段：采用文本下一词预测作为语言模型训练任务；
Fune-tune：NLG任务采用与预训练一致的方式进行处理，NLU任务加一层Linear Project来完成分类/相似度计算等任务。

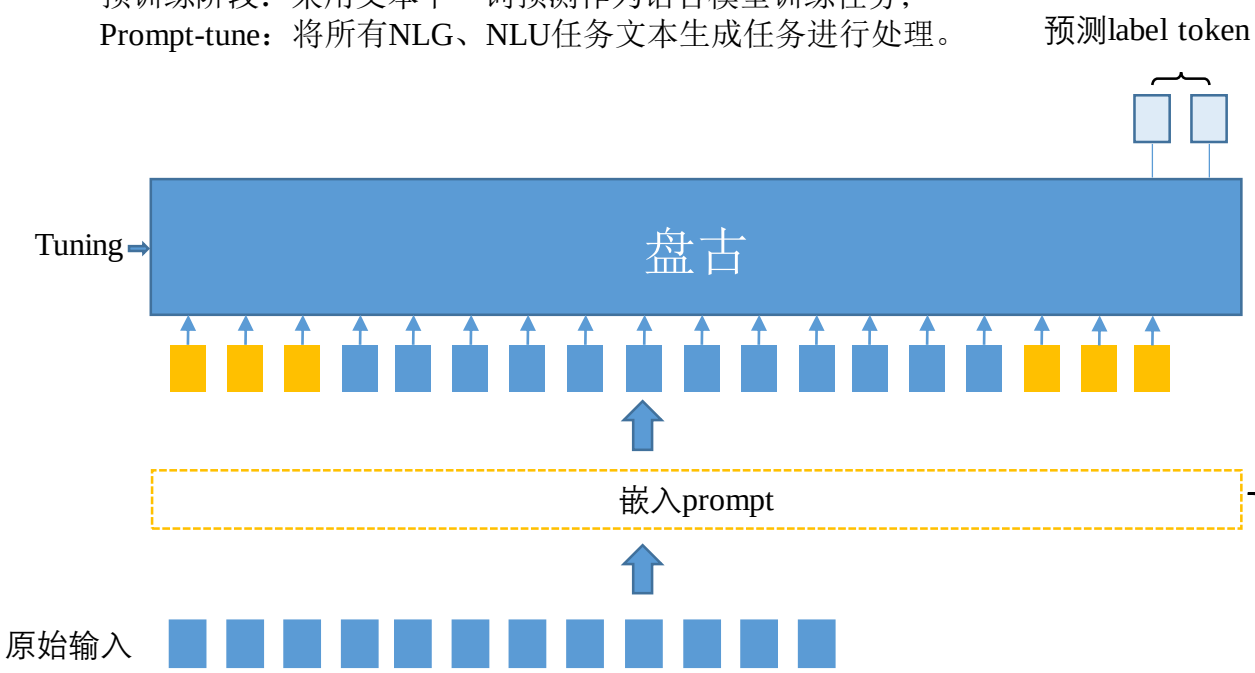




模型应用--小样本学习

盘古模型—Prompt-tune

预训练阶段：采用文本下一词预测作为语言模型训练任务；
Prompt-tune：将所有NLG、NLU任务文本生成任务进行处理。



将输入嵌入到prompt中，从而将任务转换成生成任务。
NLU任务的label映射到特定的token或词上。
NLG任务无需做label的映射。

手工prompt+tuning：PET…
自动搜索prompt+tuning： Auto-prompt…
自动生成soft prompt+tuning ： Ptuning…

PET示例

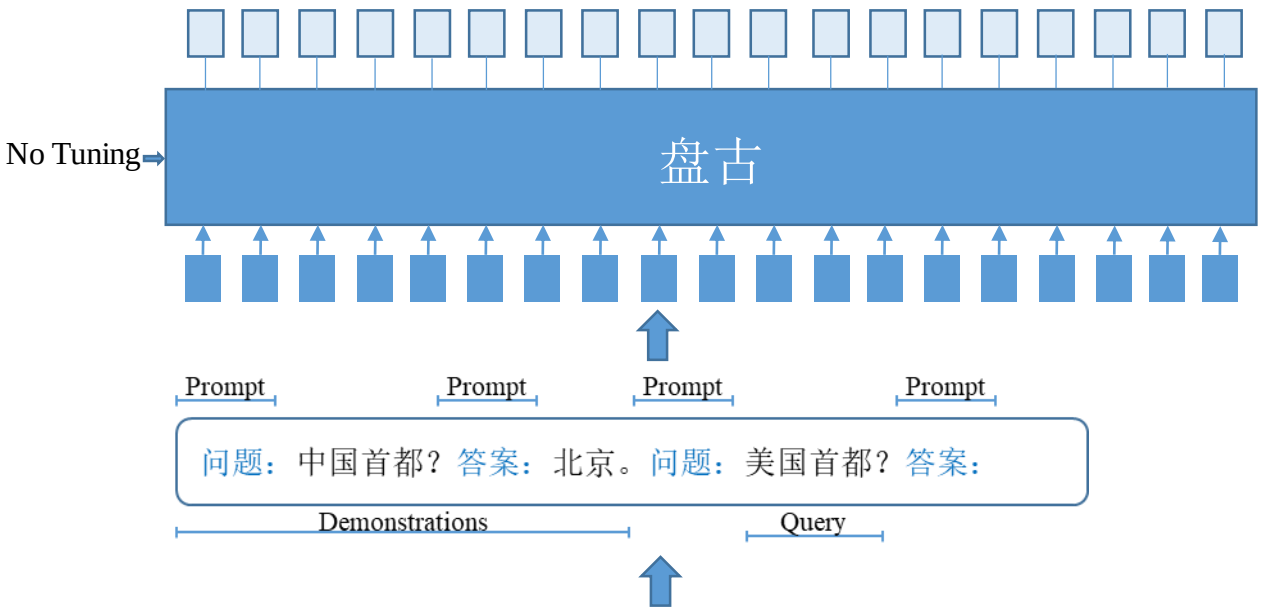
pattern P: 原始输入到完形填空式问题的映射。

verbalizer v: 输出映射到label。

模型应用-小样本学习

盘古模型—Incontext learning

预训练阶段：采用文本下一词预测作为语言模型训练任务；
Incontext few shot learning：将输入嵌入到prompt中同时加上示例输入给模型，类似人类解决新任务的机制。



所有的任务分为两类：分类类型和生成类型。
分类类型任务：将任务分解为困惑度比较任务。

16大下游任务的prompt

Task	Dataset	Input&Prompt
完型填空与补全	WPLC	/
	CHID	/
	PD&CFT	/
	CMRC2017	/
	CMRC2019	/
阅读理解	CMRC2018	阅读文章：$Document\n 问：$Question\n 答：
	DRCD	阅读文章：$Document\n 问：$Question\n 答：
	DuReader	阅读文章：$Document\n 问：$Question\n 答：
闭卷问答	WebQA	问：$Question\n 答：
指代消解	CLUEWSC2020	/
常识推理	C^3	问: $Question\n 答:$Choice\n 该答案来自对话: $Passage
自然语言推理	CMNLI	$S1?对/或许/错，$S2
	OCNLI	$S1?对/或许/错，$S2
文本分类	TNEWS	这是关于$label 的文章：$passage
	IFLYTEK	这是关于$label 的应用程序：$passage
	AFQMC	下面两个句子语义相同/不同：$S1。$S2
	CSL	摘要：$passage，关键词：$keyword 是/不是真实关键词

OCNLI的few shot

<句子 $^1_1$>，对?<句子 $^1_2$>	<Sentence$^1_1$>,Yes?<Sentence$^1_2$>
<句子 $^2_1$>，或许?<句子 $^2_2$>	<Sentence$^2_1$>,Maybe?<Sentence$^2_2$>
<句子 $^3_1$>，错?<句子 $^3_2$>	<Sentence$^3_1$>,No?<Sentence$^3_2$>
...	...
<句子 $^{k-2}_1$>，对?<句子 $^{k-2}_2$>	<Sentence$^{k-2}_1$>,Yes?<Sentence$^{k-2}_2$>
<句子 $^{k-1}_1$>，或许?<句子 $^{k-1}_2$>	<Sentence$^{k-1}_1$>,Maybe?<Sentence$^{k-1}_2$>
<句子 $^k_1$>，错?<句子 $^k_2$>	<Sentence$^k_1$>,No?<Sentence$^k_2$>
<测试句子 $_1$>，<标签>?<测试句子 $_2$>	<Test-sentence$_1$>,<Label>?<Test-sentence$_2$>

CONTENTS

01 基本框架

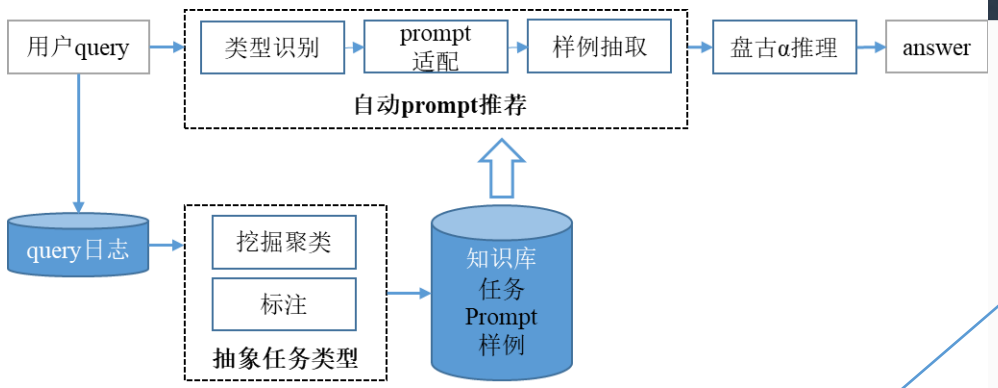
02 小样本学习

03 开源

模型应用-开源

盘古模型—在线推理服务自动提示推荐

基于盘古α在线推理服务的历史数据，进行数据挖掘分类，抽象出任务类型，构造知识库。
抽象出的任务并构建知识库，自动生成prompt和demonstration examples推荐，提升模型推理性能。



当前版本已做到自动判定用户输入任务类型，推荐相应手工prompt和topk/topp参数，后续计算实现自动生成prompt、topk/topp参数和demonstration examples。

模型应用-开源

- 盘古模型—应用技术开源

https://git.openi.org.cn/PCL-Platform.Intelligence/PanGu-Alpha-Application

鹏程·盘古α介绍

「鹏程·盘古α」由以鹏城实验
动混合并行模式实现在2048卡
预训练生成语言模型。鹏程·盘
理解等文本生成领域表现突出

[技术报告]
[模型在线推理]
[模型下载]
[模型压缩]
[模型应用]
[GPU推理、Finetune]
[小样本学习]
[megatron中文预训练模型]
[语料数据收集及处理]
[评测数据集下载]
[serving展示视频下载]
[FAQ]
[MindSpore官网]
[加入微信交流群]
[许可证]

PanGu-Alpha-Application

简介

本项目旨在为鹏程系列超大规模预训练模型提供从算法层到
设施，加速大模型的应用技术创新和应用生态构建 。

内容导引

	项目	描述
预训练模型	[鹏程盘古]	鹏程系列预训练模型
算法层&任务层	[模型迁移] [模型压缩]	算法层与任务层位置
应用层	[应用]	应用层项目位置

进展

- **2021.08.18**
  - 小样本模型迁移第一版baseline发布。
  - 任务层：支持盘古模型nli任务，baseline：cmnli。
  - 算法层：支持盘古模型fine-tune，prompt-tune，

..		
distill	cmnli baseline with finetune, prompt tune and zeroshot on pangu.	1 天前
finetune	update readme.	11 分钟前
generate	cmnli baseline with finetune, prompt tune and zeroshot on pangu.	1 天前
incontext	update readme.	11 分钟前
increment	cmnli baseline with finetune, prompt tune and zeroshot on pangu.	1 天前
kgenhance	cmnli baseline with finetune, prompt tune and zeroshot on pangu.	1 天前
prompttune	update readme.	11 分钟前
README.md	更新 'method/README.md'	8 分钟前
__init__.py	cmnli baseline with finetune, prompt tune and zeroshot on pangu.	1 天前

README.md

算法层&任务层

简介

此目录包含算法层&任务层的实现，应用算法常以一个任务做为实现实例，因此算法层和任务层在实现上归为一层。

模型应用-开源

- 问题与挑战

  - 超大规模预训练研究较多，应用技术算法研究较少，无参考。
  - 分布式训练问题，问题定位难。
  - 代码规范问题。

```
predict: [0, 1, 0, 0, 0, 0, 0, 0].
loss_reduced: {'lm loss': tensor(1.5820, device='cuda:0')}.
target: [0, 2, 0, 1, 1, 1, 0, 2].
predict: [1, 0, 0, 0, 1, 1, 1, 0].
loss_reduced: {'lm loss': tensor(1.1885, device='cuda:0')}.
target: [0, 0, 2, 0, 2, 0, 1, 0].
predict: [0, 2, 1, 1, 2, 0, 0, 1].
loss_reduced: {'lm loss': tensor(1.1621, device='cuda:0')}.
Traceback (most recent call last):
  File "/opt/conda/lib/python3.6/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/opt/conda/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/opt/conda/lib/python3.6/site-packages/torch/distributed/launch.py", line 263, in <module>
    main()
  File "/opt/conda/lib/python3.6/site-packages/torch/distributed/launch.py", line 259, in main
    cmd=cmd)
subprocess.CalledProcessError: Command '['/opt/conda/bin/python', '-u', '/userhome/gpt3/PanGu-Alpha-GPU/panguAlpha
--num-attention-heads', '32', '--batch-size', '8', '--seq-length', '1024', '--max-position-embeddings', '1024', '-
oot/pangu/model/pangu_fp16_4mp_2b6', '--data-path', '/root/pangu/data/cmnli_public', '--vocab-file', '/userhome/gp
 'mmap', '--split', '949,50,1', '--distributed-backend', 'nccl', '--lr', '0.00005', '--lr-decay-style', 'cosine',
erval', '100', '--eval-interval', '100', '--eval-iters', '1000', '--attention-dropout', '0.1', '--hidden-dropout',
--fp16-lm-cross-entropy', '--use-cpu-initialization']' died with <Signals.SIGSEGV: 11>.
root@2734b4c399a7:/userhome/gpt3/PanGu-Alpha-GPU/panguAlpha_pytorch/examples#
```

感谢聆听！