0507 汇报

朱睿涵　李宸亦

项目三

2022 年 5 月 6 日

目录

标题

PANGU-BOT 部分的 slides

目录

PLATO 系列

PLATO 系列：百度这两年针对 NLP 对话领域提出的一系列预训练的模型。

- PLATO
- PLATO-2
- PLATO-XL

从 PLATO 到 PLATO-XL，使用的数据越来越多，模型大小越来越大，但是在 PLATO-XL 中模型的结构实际上更加简单了。

模型结构

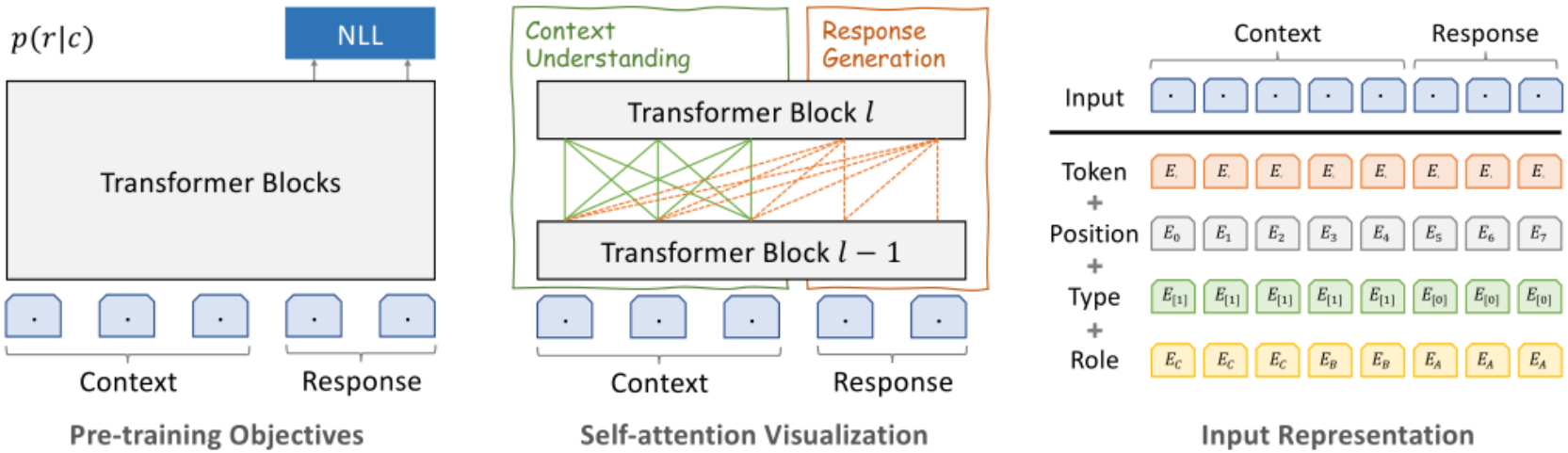


Figure 1: Network overview of PLATO-XL.

- 抛弃了 PLATO 的隐变量和 PLATO-2 的课程学习的方法，直接训练
- 沿袭 PLATO 中使用的 unified transformer
- 只保留了负对数似然（NLL）作为对话生成的预训练目标
- 训练时加入了多角色感知的预训练

为何采用 unified transformer

传统用 Transformer 的对话生成：使用 encoder-decoder 结构。
使用 unified transformer 的两方面好处：

1. 提升训练效率
   - 传统：对话样本长短不一
   - padding 补齐带来大量的无效计算
   - unified transformer：可以对输入样本进行有效的排序
2. 提高参数性价比
   - 可以**同时**进行对话理解和回复生成的联合建模
   - 灵活的注意力机制
   - 对 context 进行了双向编码
   - 对 response 进行了单向解码

多角色感知的预训练

目的：

- 改善对话模型有时候自相矛盾的问题
- 提升多轮对话上的一致性

产生矛盾的原因：

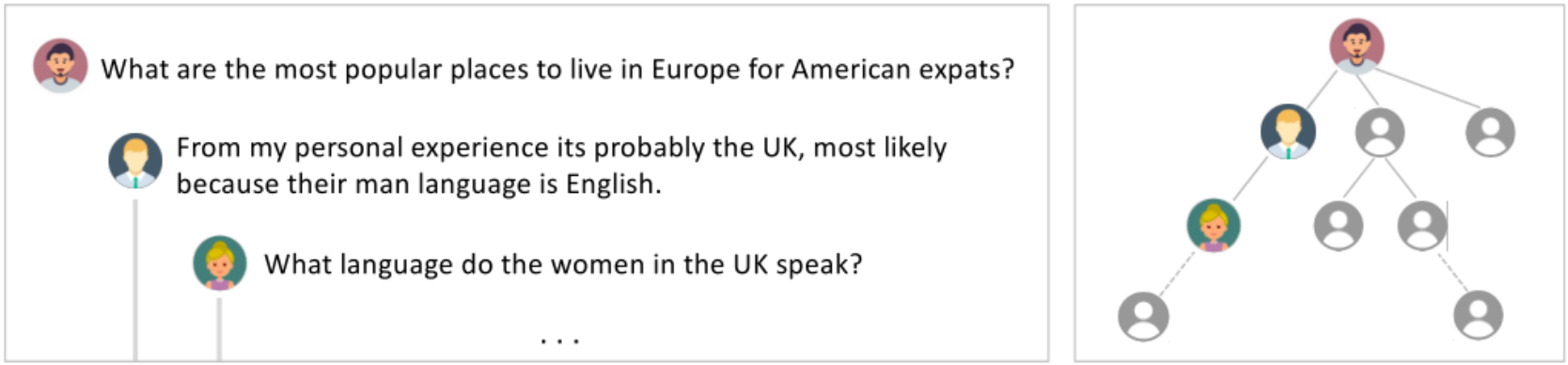


Figure 2: Left: one toy example to illustrate social media conversations. Right: corresponding message tree.

对话模型所用的预训练语料：社交媒体对话

- 特点：通常有多个用户参与
- 训练时模型较难区分对话上文中不同角度的观点和信息
- 容易产生一些自相矛盾的回复

实验结果

评估方法：采用了两个模型针对开放域进行相互对话（self-chat）的形式，然后再通过人工来评估效果。

- PLATO-XL 与 Facebook Blender、微软 DialoGPT、清华 EVA 模型、PLATO-2 相比，取得了更优异的效果
- PLATO-XL 也显著超越了目前主流的商用聊天机器人
- 除了开放域闲聊对话，模型也可以很好的支持知识型对话和任务型对话，在多种对话任务上效果全面领先
- 模型规模扩大对于效果提升也有显著作用，呈现较稳定的正相关关系

目录

Facebook Blender

这篇论文融合了 Facebook Blender 这个组近些年来在 open-domain chatbot 方向的诸多相关工作，读起来有些吃力。
论文中提出了三个模型：

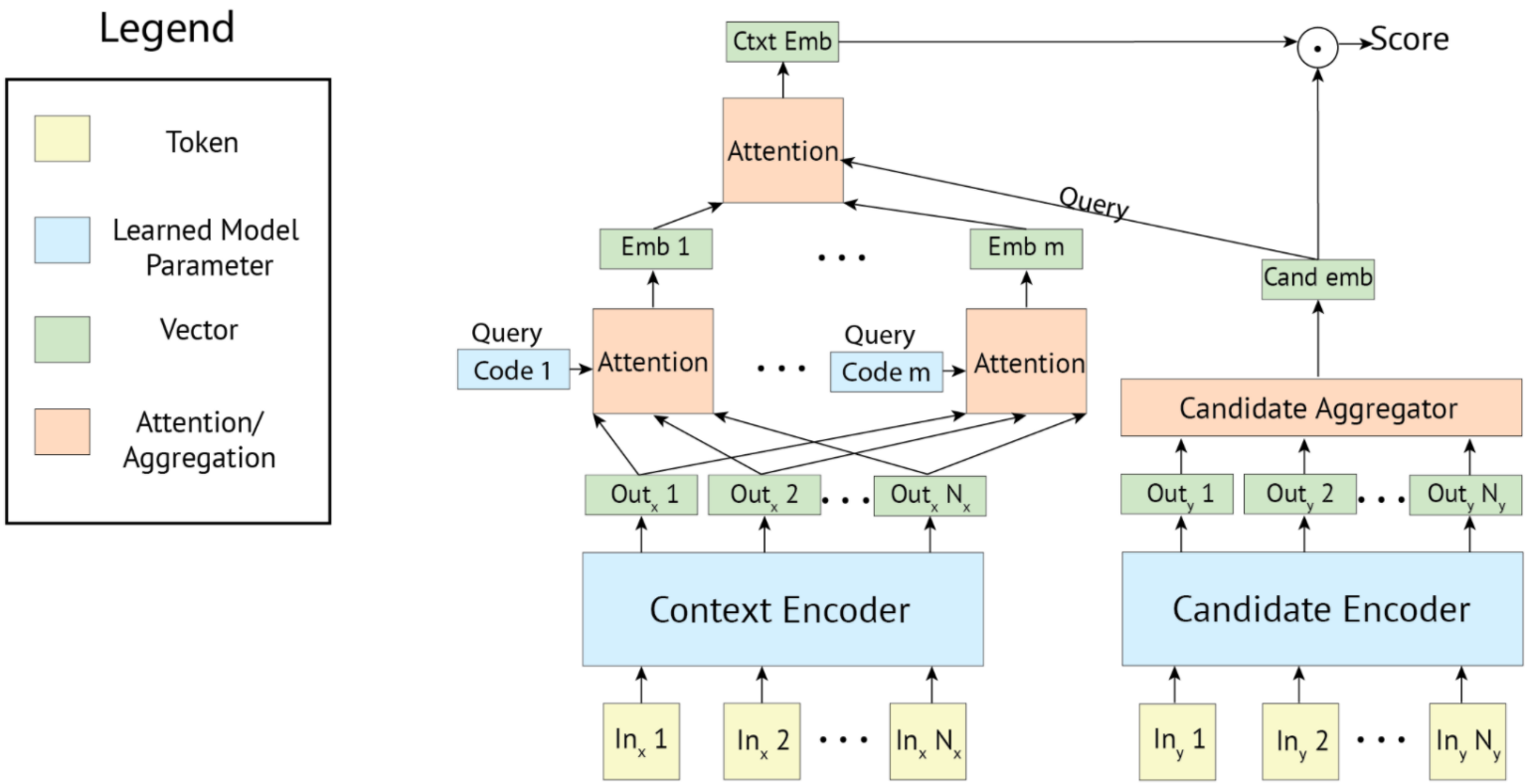
- 检索模型（Retriever）
- 生成模型（Generator）
- 检索 + 生成（Retrieve and Refine）

检索模型（Retriever）

从候选集中选取最合适的句子作为 bot 当前的答复。

- 训练时，候选集只有给定的一句 response
- 推断时，候选集由训练集中的所有 response 组成

打分 / 排序模型使用他们在之前的论文中提出的 Poly-encoder 模型。



实验表明：m 越大效果越好，当然模型打分也越耗时。

生成模型（Generator）

模型结构：

- 标准的 seq2seq 结构，只是用了标准的 Transformer 层
- encoder 层数少，decoder 层数多的设计

Blender 使用 beam search，但是加入了一些限制方法：

- 限制生成 response 的最小长度：
  - Minimum length
  - Predictive length
- 屏蔽重复的子序列

beam search 的这些限制方法实际上都是锦上添花，如果模型本身的质量不高，上述的方法作用也不大。

生成模型（Generator）

训练方法：

- Seq2seq 模型标准的训练方法 MLE
- Unlikelihood Loss

Unlikelihood Loss（UL）是作者在之前的论文中提出的一种损失函数，在提高正确的 token 概率的同时，降低其他 token 的概率。

UL 的关键

如何选取这些被打压的负 token。

- 作者选的是那些容易组合成常见 n-grams 的 tokens。
- 目的：期望降低生成无意义 response 的比例。

检索 + 生成（Retrieve and Refine）

Retrieve and Refine (RetNRef) 融合了检索和生成两种方法，是作者在 18 年的论文中提出的。

RetNRef

1. 先利用检索模型检索出一个结果
2. 把检索出的结果拼接到 context 后面，用一个特殊的分割符和 context 分隔
3. 整体作为 generator 模型的输入

目的：期望生成模型能学习到在合适的时候从检索结果中 copy 词或词组。

检索 + 生成（Retrieve and Refine）

两种检索方法：

Dialogue Retrieval 从训练数据中检索出得分最高的 response 作为结果

Knowledge Retrieval 从外部的大知识库如 Wiki 中检索

- 对于 Knowledge Retrieval，把检索出的结果直接追加到 context 后面，然后利用标准的 MLE 训练即可
- 对于 Dialogue Retrieval，直接利用 MLE 训练会有问题。训练出来的模型很容易直接忽略掉追加的检索部分

α -blending

训练时以 $\alpha\%$ 的概率把检索结果替换为实际 response。
这样模型就会被吸引去关注检索部分了。

数据集

数据集公认的标准：

- 对话要个性有趣
- 对话要包含知识
- 对话要富有同理心

作者在他之前的论文中发现：**在具有某些特性的数据上训练出的模型也会拥有这些特性。**

训练使用的数据集：

- http://pushshift.io Reddit
- ConvAI2
- Empathetic Dialogues (ED)
- Wizard of Wikipedia (WoW)
- Blended Skill Talk (BST)

模型训练流程：

1. 在 http://pushshift.io Reddit 上进行预训练
2. 在 ConvAI2、ED、WoW 上多任务精调
3. 在 BST 上精调

在优质数据上训练模型，也能降低模型产生不好的 response 的概率。

评估方法

自动评估：

- 生成模型采用 Perplexity(PPL)
- 检索打分模型采用 Hits@1/K

人工评估：

- ACUTE-Eval：
  1. 每次给两个对话 session
  2. 让人来评判哪个 speaker 聊的更好
  3. ACUTE-Eval 可以给出两个 speaker 各自的胜率
- Self-Chat ACUTE-Eval：和 ACUTE-Eval 的做法类似，只是评估时用的是自己跟自己聊的 session。

评估结果

PPL：

- 模型越大 PPL 越低
- RetNRef 相比于同等规模的生成模型，PPL 会略有提升

Self-Chat ACUTE-Eval：

- 相同尺寸，Generator 和 RetNRef 模型都未使用最小长度约束时：Retrieval 比 RetNRef 略好，二者都远远好于 Generative。
- Beam Search 中加入限制方法：
  - 限制最小生成长度效果显著，最小长度限制设为 20 效果最好
  - 子序列屏蔽有点用，但不显著
- 精调后的模型比只进行预训练的模型效果好很多
- Unlikelihood Loss 比 MLE 效果好一点点，但不显著

评估结果

ACUTE-Eval：

- 相同尺寸 Generator 和 RetNRef 模型使用的 beam search 都加入了最小长度 20 的限制：Generator 和 RetNRef 显著优于 Retrieval，RetNRef 略优于 Generative，但不显著。
- Blender 显著优于 Meena，胜率高达 70%
- Blender 最好的模型和人 PK 的胜率已经到了 49%，与人类的差距仅有 1%

模型存在的问题

- 倾向于使用高频词
- 倾向于生成重复信息
- 模型生成的答复可能前后冲突
- 其他问题：
  - 无法针对某个话题做深度对话，不会倾向于使用更多知识进行深聊
  - 模型无法深度理解，无法通过对话真正教会模型理解某个概念
  - 当前的评测针对的是 14 轮长度的对话，分析更长的对话肯定会发现其他问题

Blender 总结

- 大模型、大数据集对实验结果提升极其显著
- 训练数据的特性决定模型特性
- 评估方法使用 ACUTE-Eval、Self-Chat ACUTE-Eval
- 对 decoding 过程加入控制，比如控制生成句子长度，效果会更佳
- Blender 在轮次少的情况下已经接近人类的水平，但目前还有很多问题待解决