

Google 利用强化学习重组开源主动问题


自然语言理解是 Google AI 研究持续关注的一个重要领域，并已应用于机器翻译、句法和语义分析等更多方面。重要的是，随着会话技术越来越需要直接回答用户问题的能力，我们致力于研究的最活跃领域之一便是问答 (QA)，这是人类对话的基本组成部分。

由于开源代码是可重复研究的关键组成部分，我们发布了 Active Question Answering (ActiveQA) 的 TensorFlow 软件包，此项目研究使用强化学习来训练人工智能代理进行问答。我们在 ICLR 2018 论文“提出正确问题：利用强化学习重组主动问题 (Ask the Right Questions: Active Question Reformulation with Reinforcement Learning)”中首次引入 ActiveQA，此系统使用自然语言与 QA 系统进行交互，以提供更优的答案。

Active Question Answering

在传统 QA 中，监督式学习技术通常与标记数据结合使用，以训练能够回答任意输入问题的系统。这种方法虽然有效，但缺乏人类重组问题、进行多次搜索、评估和汇总回答等处理不确定性的能力。受人类“提出正确问题”能力的启发，ActiveQA 引入一个反复询问 QA 系统的代理。如此一来，代理可以多次重组原始问题，以找到最可能的答案。我们称其为主动性方法，是因为代理会与 QA 系统进行动态交互，以提高所返回答案的质量。

例如，对于“Tesla 何时出生？”这一问题，代理会以两种不同的方式重组这个问题：“何时是 Tesla 的生日”和“Tesla 出生于哪一年”，并从 QA 系统中检索这两个问题的答案。整合所有相关信息之后，它决定返回“1856 年 7 月 10 日”。



When was
Tesla born?

ActiveQA 系统的特点是，它能够学会提出可以给出更好答案的问题。但是，由于训练数据采用由原始问题和更优化变体组成的问题对的形式，并不容易获得，因此，ActiveQA 使用强化学习（与训练代理有关的机器学习方法），以便采取使奖励最大化的行动，同时与环境互动。

学习过程在 ActiveQA 代理与 QA 系统交互时即开始进行，并根据相应答案的质量来评估每个重组问题的优劣，从而形成奖励。如果是优质答案，学习算法便会调整模型参数，以提高再次产生优质答案的重组问题的可能性；如果答案不佳，便会降低相应的可能性。

我们的论文表明，通过提出更好的问题，我们有可能训练这些代理，使其表现优于基础 QA 系统（用于提供重组问题的答案）。这项结果具有重要意义，因为此 QA 系统已通过监督式学习训练，以解决相同的任务。我们的研究还有一个引人注目的发现，即 ActiveQA 代理能够学会一种相当复杂且某种程度上仍然可以解释的重组策略（强化学习中的策略）。此学习策略使用广为人知的信息检索技术，例如 tf-idf 查询术语重新加权法，通过这一过程，我们可以得到比通用术语更多的加权信息性术语，以及词干提取。

构建您自己的 ActiveQA 系统

我们发布的 TensorFlow ActiveQA 包由三个主要组件组成，并包括训练和运行 ActiveQA 代理所需的全部代码。

经过预训练的序列到序列模型，它将问题作为输入并返回重组问题。此任务类似于机器翻译中的英语到英语翻译，而且初始模型确实可用于一般性释义。我们通过使用和自定义 TensorFlow 神经机器翻译教程代码来实现这一目的。我们还使用策略梯度方法调整代码，以支持强化学习训练。*

答案选择模型。原始问题、重组问题和答案三个为一组，而答案选择器会使用卷积神经网络并为其打分。此选择器使用经过预训练且公开可用的单词嵌入 (GloVe)。

问答系统 (环境)。为此，我们使用一种流行的问答系统 BiDAF，详见 [Seo 等人的论文](#) (2017 年)。(https://openreview.net/forum?id=HJ0UKP9ge¬eId=HJ0UKP9ge)

我们也针对所有训练模型提供指向检查点的指针。

Google 的使命是整合全球信息，供大众使用，让人人受益，而我们相信 ActiveQA 是实现这一使命的重要步骤。我们设想这项研究将帮助我们设计能够提供更好和更易于理解的答案的系统，并希望它能帮助其他人开发可使用自然语言与世界互动的系统。

致谢

对这项研究和发布作出贡献的人士包括 Alham Fikri Aji、Christian Buck、Jannis Bulian、Massimiliano Ciaramita、Wojciech Gajewski、Andrea Gesmundo、Alexey Gronskiy、Neil Houlsby、Yannic Kilcher 和 Wei Wang。

* 我们在论文中探讨的系统使用了 TensorFlow 序列到序列代码，而此代码也用于 [Britz 等人的论文](#) (2017 年) (https://arxiv.org/pdf/1703.03906.pdf)。之后，我们将 Google 翻译模型 (GNMT) 的开放源代码版本作为 [教程发布](#) (https://ai.googleblog.com/2017/07/building-your-own-neural-machine.html)。今天发布的 ActiveQA 版本以最近大力开发的实现方式为基础。因此，发布的系统与论文介绍的版本会稍有不同。不过，无论是在质量还是数量上，性能和行为都相差无几。