**Learning to Generate Instruction Tuning Datasets for Zero-Shot Task Adaptation**

学习生成用于零样本任务适应的指令调优数据集

**摘要**

本文介绍了Bonito，一个用于条件任务生成的开源模型。条件任务生成：将未注释的文本转换为特定于任务的训练数据集，用于指令调整。研究人员通过将现有的指令调优数据集重新混合到元模板中，创建了一个新的165万个示例的大规模数据集，并在上面训练了Bonito。元模板为数据集生成训练示例，其中输入是未注释的文本和任务属性，输出由指令和响应组成。作者使用Bonito为三种领域生成合成任务——是非问题回答、抽取式问题回答和自然语言推理，并对语言模型进行调整。实验结果表明，Bonito显著提高了预训练模型和指令调优模型的平均性能，超过了事实上的自我监督基准。作者用Bonito进行了额外的实验，以理解领域、训练集的大小和替代合成任务生成器的影响。总体而言，使用合成指令调优数据集进行学习是将语言模型适应到新领域的有效方法。模型、数据集、代码已开源：<https://github.com/BatsResearch/bonito>。

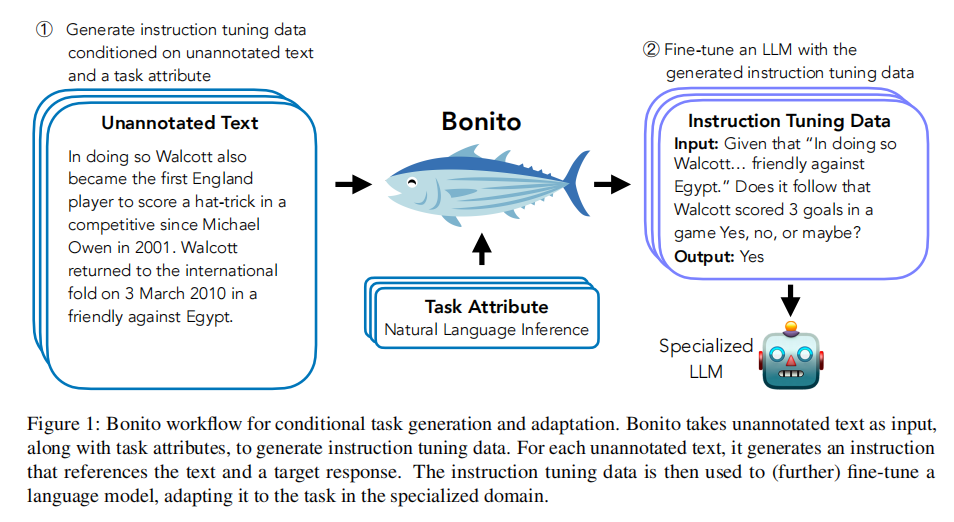
**引言**

大语言模型显示出显著的零样本能力。通过在指令调优数据集上微调模型，模型型通常会提高对不可见的指令的响应能力。指令调优数据集包含许多任务，每个人物都包含一个输入的指令和一个期望的响应。然而，这种泛化仍然受到指令调优数据集的质量的限制。现有的数据集如Public Pool of Prompts (P3)、Natural Instructions、Dolly-v2通常不包含专门的领域知识，如生物医学和法律领域。本文研究如何使模型遵循无标注的特定领域的指令。

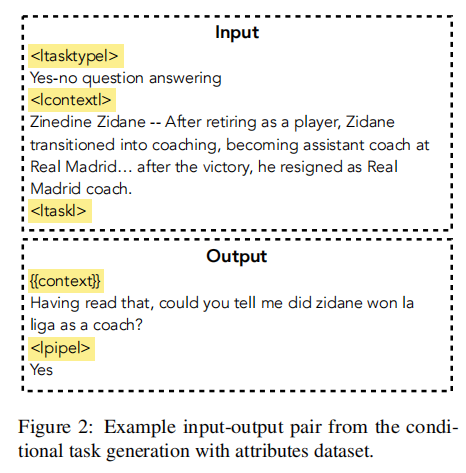
在特定领域中遵循特定任务指令的能力对于将大语言模型推广给更多用户非常重要。近期的评估表明，LLM在需要专业知识的领域上的表现明显不如专门的模型，这促使作者研究为大型语言模型提供领域知识的有效方法。

在目标语料库上以预测下一个token的形式进行的自监督，是一种教授语言模型新领域知识的简单方法。然而，这种方法需要大量的训练才能获得强大的表现。此外，在本文的工作中，研究人员发现自监督会抵消指令调优的好处。另外，模型的持续指令调优已经被证明可以提高在特定领域数据集上的性能。但是，这些工作都消耗了大量时间和人力来对特定领域的数据集进行标注。在本文工作中，作者的目标是自动创建特定领域的指令调优数据集。

研究人员创建了Bonito，一个开源模型，用于将未注释的文本从特定的领域转换为特定任务的训练数据集，以用于指令调优（如图1）。作者称这个问题为条件任务生成。



这里的关键思想是，可以利用现有的数据集为指令调优创建一个新的训练集。首先在P3数据集中选择一个模板子集，该模板从上下文中创建任务，上下文是响应指令所需的文本片段。例如，上下文可以是包含事实或包含问题答案的段落。研究人员还用任务属性注释了这些模板，然后使用这些模板来创建用于训练一个新的语言模型的元模板（如图2）。



每个元模板都生成训练示例，其中输入是上下文和一个任务属性，输出是一个整个任务：指令（包括上下文）和所需的响应。通过这种方式，可以轻松地创建丰富、多样的条件任务生成示例。然后，可以在合成的数据集上训练语言模型，使其适应目标领域的期望任务。

Bonito对预训练和指令调整模型的零样本任务适应的提升显著优于自监督方法。作者使用Bonito为三种任务类型的七个数据集生成指令调优数据——是非问题回答（PubMedQA和Privacy Policy QA）、抽取式问题回答（SQuADShifts-NYT,Amazon,和Reddit）和自然语言推理（ContractNLI 和 Vitamin C），并对语言模型进行调整。

实验结果显示，相较于自监督的基线，在Bonito将Mistral-7提高了34.7的F1值，将Llama 2 7B提高了31.6的F1值。作者还考虑了一个更实际的设置，其中进一步训练Mistral-7B-Instruct-v0.2和Mistral-7B和Llama 2 7B的指令调整变体。结果显示，在所有模型中，Bonito的性能平均都比强的零样本基线性能高出22.1的F1值。另一方面，研究人员发现**自监督破坏了指令调优的一些好处**，它会造成灾难性的遗忘，导致所有模型的性能平均下降0.8 F1值。对Bonito的分析表明，即使是针对特定任务的模型，也可以通过简单地学习Bonito生成的任务来进一步改进。研究人员还发现，与其他数据集相比，在PubMedQA和Vitamin C数据集上使用更多的合成指令进行训练对模型性能的提高最大。

最后，研究人员通过prompt现成的开源模型，如Zephyr-7B-β，Mistral-7B-Instruct-v0.2和GPT-4来生成任务，并发现它们通常可以改进预训练模型，但当模型被指令调整时，仍然难以进一步提高模型性能。

本文的贡献可以概括如下：

·提出了Bonito，一个用于条件任务生成的开源模型，它可以将用户的未注释文本转换为特定任务的指令调优数据集。

·在三种任务类型的7个数据集上进行的零样本任务适应实验表明，Bonito在预训练模型上平均比自监督基线提高了33.1 F1值，在指令调整模型上平均提高了22.9 F1值。

·分析了领域、训练规模和替代任务生成器的选择的影响，强调了Bonito的好处和局限性。

**零样本任务适应**

此部分描述了零样本任务适应问题。给定一个语言模型，要么通过自监督进行预训练，要么在类似P3的数据集以及来自目标域的无注释语料库上进行微调。同时也知道目标任务的类型，如抽取式问题回答和自然语言推理等。如果目标任务类型有一组固定标签，那么假设可以访问它们。本文的目标便是使语言模型适应于目标领域的任务指令，而不需要人工注释，即实现零样本任务适应。

**相关工作**

**指令调优**

多任务指令调优显著提高了语言模型遵循指令和概括到新的不可见任务的能力。通常，预训练模型在大规模的混合训练集上进行训练，如P3和FLAN，以遵循指令。在这项工作中，研究人员使用P3来创建元模板，并训练Bonito在特定的领域中生成NLP任务。

**领域适应**

一些工作表明，在目标域语料库上对预训练模型进行自监督或持续进行预训练可以提高下游性能。在本工作中，研究人员发现自监督提高了预训练模型的性能，但损害了指令调优模型的性能。

最近的工作通过在大规模领域内数据集或一些特定领域任务的样例上进行训练来调整语言模型。在实践中，为新领域注释训练数据集需要大量的人力和成本。本工作专注于为任务生成训练数据，并将语言模型适应到没有注释的特定领域。

零样本任务适应与无监督领域适应密切相关。在无监督域适应中，使用模型为目标未标记数据生成伪标签，然后在这些标签上进行训练。在本文工作中，朴素的伪标签并不适用，因为本文考虑了诸如问题回答和自然语言推理等任务，即在预测标签之前需要一个问题或一个假设。此外，在无监督领域适应中使用的流行技术，如“选择top-K自信类”，不能轻易地适应可能没有明确的类概念的NLP任务。

人们对使用检索增强生成（RAG）进行开放领域问题回答的兴趣越来越大。在RAG工作流程中，给定一个问题，在使用语言模型准确生成答案之前需要检索最相关的文档。本文工作支持了RAG，因为本文假设可以访问来自专门领域的文档或段落，并提高了语言模型回答问题的能力。

**任务生成**

任务生成是一个快速增长的研究领域，用于调整大语言模型遵循指令。模型将GPT或其自身设置为一组种子任务演示，并生成新的任务。然而，以未注释文本为条件的任务生成大多被现有工作忽略了。此外，使用基于API的模型生成是昂贵的，并且不能用于专有或私有的研究数据。相比之下，Bonito是一个开源模型，可以使用用户的未注释文本创建任务，而不需要额外的API成本。

**知识蒸馏**

知识蒸馏是一个研究广泛的领域。通常，较小的模型会从较大模型的输出中学习。最近，基于API的模型已经被用来使用生成任务，并提炼成更小的模型，以模拟基于API的模型的能力。在本文工作中，研究人员使用Bonito根据用户的上下文生成任务，并将它们提炼成相同大小的预训练和指令调整模型，用于零样本任务适应。

**问题生成**

多年来，人们提出了一些关于问题生成的工作。它们经常使用启发式方法，如模板、命名实体识别和语义图。在本文的工作中，研究人员训练了一个没有特定于任务的启发式的语言模型Bonito。Bonito可以生成提取式问题回答之外的任务，并通过Llama 2 7B和Mistral-7B等大型模型在几种任务类型上实现零样本任务适应。

**Bonito：学习生成任务**

本节描述了使用属性数据集创建条件任务生成来训练Bonito的步骤。然后，简要描述了为未注释的目标文本创建合成任务以调整语言模型的过程。

**关键特性**

希望的任务生成模型的关键特性：

(1)给定一个包含文章和段落的语料库，模型应该把文本作为输入并生成高质量的任务，需要最小的清洗或后处理。

(2)模型应该坚持任务类型，如抽取式问题回答或自然语言推理任务。

(3)对于不同任务，模型应该以不同风格生成精确文本。

**具有属性的条件任务生成（CTGA）**

为了创建满足关键属性的模型，我们首先创建一个新的训练数据集：具有属性的条件任务生成（CTGA）。该数据集包含来自P3的165万个样例，通过注释39个具有16种任务类型的数据集的323个提示模板来生成。

提示模板用于创建元模板，而元模板又依次生成训练样本。元模板的输入有一个任务类型（<|tasktype|>）作为属性，后面是未注释的文本或上下文（<|context|>）。元模板的输出包括属性任务、提示或任务描述和上下文（{context}），后面跟着一个符号（<|pipe|>）和任务的解决方案。使用<|pipe|>符号来分离用于适应下游模型的指令对和响应对。图2显示了一个使用元模板生成的CTGA数据集的输入输出示例。

**构建数据集**

该数据集是通过识别需要上下文来完成任务的数据集来构建的。例如，SQuAD需要一个上下文来完成抽取式问题回答的任务，而CommonSenseQA在不提供任何相关文本的情况下询问多项选择题。本文确定了39个数据集。在选择带有上下文的相关数据集之后，使用任务类型注释数据集中的所有提示。本文用16种任务类型注释了总共323个提示。然后，重新构造提示模板以创建元模板。最后，将元模板应用于数据集中的所有示例。如果数据集有多个元模板，那么在每个示例中统一使用一个元模板。将每个数据集的示例总数限制在10万个。最后的训练数据集用于训练Bonito。

**训练Bonito模型**

在CTGA数据集上训练Mistral-7B。从训练集中统一抽取10,000个例子作为验证集来监测损失，数据集的其余部分用于训练Bonito。使用Q-LoRA，通过优化输出的交叉熵损失来训练模型。该模型被训练了10万步。在4个gpu上进行的训练大约需要4天完成。

**在合成数据集上训练语言模型**

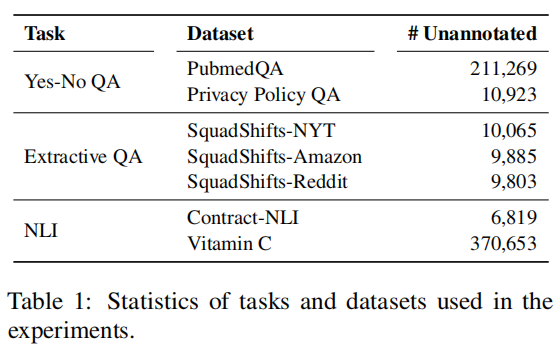
经过训练的Bonito模型在目标任务的目标未注释文本上生成合成任务。对于每个未注释的文本，生成一个指令和响应对，然后用它来训练下游语言模型，优化输出token上的交叉熵损失。

**实验**

**实验设置**

**目标任务和数据集**

在这项工作中，考虑了三个目标任务：是非问题回答（YNQA）、抽取式问题回答（ExQA）和自然语言推理（NLI）。表1显示了实验中的7个数据集以及三种任务类型的未注释文本的数量。



**基线**

本文考虑了两个关键的基线：零样本和自监督基线。对于零样本基线，只需提示模型并进行评估，而不使用目标任务中的任何未带注释的文本（None）。对于自监督的基线，使用任务自适应预训练（TAPT），学习目标是继续实现对下游数据集中未注释文本的预训练目标。在实验中，使用预测下一个单词的目标来微调Mistral-7B和Llama2 7B模型。

**合成任务生成**

这里描述了用Bonito生成合成任务的过程。如第4节所述，给定一个任务类型，使用未注释的文本和任务类型提示Bonito生成指令调优数据。使用vLLM框架下的核采样，最高P值为0.95，温度为0.5，最大序列长度为256。

生成的任务被处理成标准化的指令响应格式，用于指令调优。在每一轮迭代中，用实际的未注释文本替换{context}。如果生成的输出由于缺少<|pipe|>而无法解析，则过滤掉。

**模型**

采用两种预先训练好的LLM：Mistral-7B和Llama2 7B。

还考虑了一个更实际的设置，以进一步调整指令调整模型以适应目标任务。首先考虑一个现成的指令调整模型：Mistral-7B-Instruct-v0.2。此外，在P3数据集的T0切片上对Mistal-7B和Llama 2模型进行训练，并使它们适应目标任务。将这些模型称为Mistral-7BP3和Llama 2P3。

**训练详情**

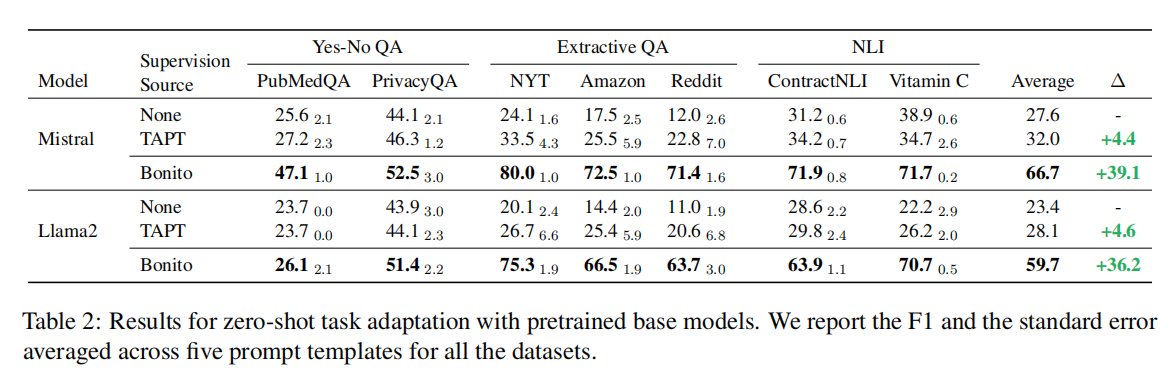
使用Q-LoRA，在监督源TAPT和Bonito上对语言模型进行了微调。当进一步调整Mistral-7BP3和Llama 2 7BP3时，在监督源上调整相同的Q-LoRA适配器，而不是合并和重新初始化适配器。所有的模型都训练1个epoch。如果数据集的大小大于16万个示例，那么将训练10,000步。为了避免额外的超参数调优，使用了Dettmers等（2023）的相同的超参数值。

**评估**

在目标数据集的测试集上评估了模型的性能。为了防止“prompt hacking”，根据Sanh等（2022）的做法，首先为目标数据集编写5个提示模板，然后对模型性能进行基准测试。根据Radford等（2019），为了评估是非问题回答和NLI，使用排序分类，即生成所有选择的对数似然，并选择对数似然最高的序列作为预测。根据Rajpurkar等（2016），通过计算生成输出的SQuAD F1分数来评估抽取式问题回答的性能。在评估过程中，使用贪心解码来生成模型的输出，然后计算数据集的SQuAD F1得分。

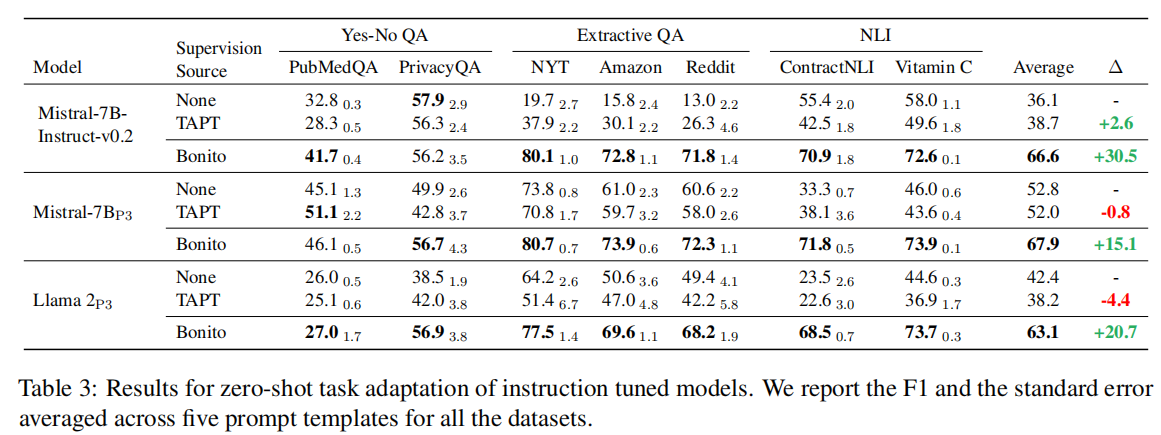
**调整预训练模型**

表2显示，使用Bonito生成的合成指令调优数据调整的预训练模型的性能明显优于零样本和TAPT。与Llama 2 7B相比，Mistral-7B表现出了更大的性能改善，这表明更强的预训练模型可能对合成指令反应更好。



**调整指令调优模型**

表3显示，Bonito将指令调优模型平均提高了22.1 F1值，而TAPT将平均性能降低了0.8 F1值。这是因为使用TAPT的自我监督干扰了先前的指令调优，并导致灾难性的遗忘。相比之下，使用Bonito生成的任务来调整指令调优模型，进一步提高了在特定领域中执行任务的性能。例如，Bonito显著提高了Mistral-7B-Instruct-v0.2在抽取式问题回答方面的表现，因为它通常会对问题产生类似聊天的回答。最后，发现Mistral-7B和Llama 2 7B的指令调整变体比预训练的模型获得了更高的F1分数（见表2）。



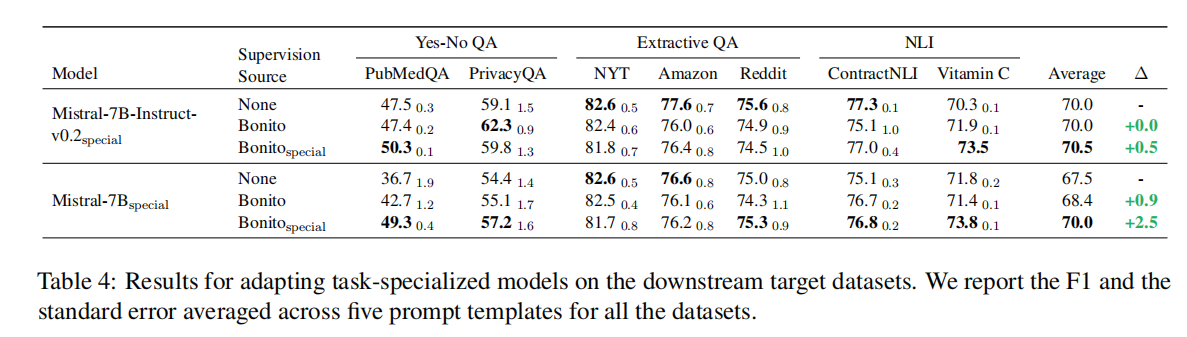
**分析**

**领域知识的影响**

这里提出了一个关键问题：本工作是通过学习特定领域来改进语言模型，还是从强模型到弱模型中提取指令调优数据？为了回答这个问题，训练了专门针对任务的指令调整模型，然后进一步根据Bonito从未注释文本生成的合成任务训练它们。

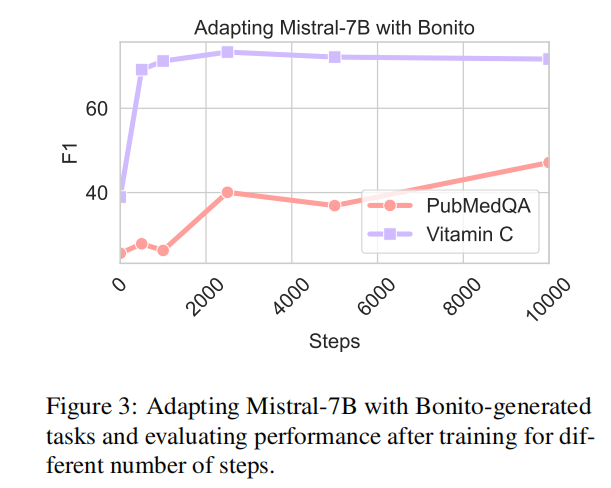
通过在目标任务类型的数据集中选择prompt，来创建特定任务的训练数据集。训练了两个专门的任务模型：Mistral-7B-Instruct-v0.2\_special和Mistral-7B\_special。从同一数据集创建元模板来训练一个针对特定任务的Bonito\_special。

表4显示，对合成指令的进一步训练可以进一步提高性能，这表明该模型受益于来自特定领域的未标记文本。模型在抽取式问答的性能往往降低。作者怀疑，由于在任务专门化的训练数据集中存在SQuAD，模型的性能已经饱和。最后，几乎总是能提高 Vitamin C和PubMedQA数据集的性能，这强调了在更多的任务样本上进行训练的重要性。



**训练数据集大小的影响**

这里研究了训练数据集的大小的影响。特别研究了在PubMedQA和Vitamin C中训练不同数量的合成指令调整数据时，Mistral- 7B的性能如何变化。图3显示，对更多步的训练通常会提高性能。作者建议使用验证集来选择性能最好的模型权重。



**附加实验**

简要描述了附录B和c中包含的其他实验。

附录B中，通过提示Mistral-7B-Instruct-v0.2和Zephyr-7B-β来生成合成任务。研究结果表明，合成任务虽然提高了Mistral-7B的平均性能，但在适应MistralP3时，其平均性能显著降低。这表明，仅仅生成合成任务是不够的，需要高质量的合成任务来提高强指令调优模型的性能。

附录C中，使用GPT-4为 Privacy Policy QA、SQuADShifts Reddit和ContractNLI生成合成任务。结果显示，GPT- 4改善了MistralP3在Privacy Policy QA和ContractNLI上的新能，但略微降低了SQuADShifts Reddit上的性能。分析生成的任务，并确定开源模型和GPT-4中两个常见的问题：标签空间的分布和“聊天”响应，是性能下降的潜在原因。

**结论**

本文提出了Bonito，一个用于条件任务生成的开源模型，以将未注释的文本转换为指令调优数据集。实验表明，在专门领域的合成指令调优数据集并进行训练是一个强大的替代自监督的方法。Bonito生成的指令提升了预训练模型和指令调优模型的零样本任务适应能力。总体来说，Bonito使从业者能够在不需要注释的情况下使大型语言模型适应其数据上的任务。

**局限性**

本文的工作依赖于大量未带注释的文本。如果只有少量未注释文本，目标语言模型在适应后可能会出现性能下降。虽然本文展示了对预训练和指令调整模型的正向改进，但本文的观察仅限于在实验中考虑的三种任务类型。

**潜在风险**

Bonito的潜在风险类似于任何大型语言模型。例如，模型在特定的领域中可能生成不符合事实的数据集。本文模型可以显示出基本模型，Mistral-7B的偏差和刻板印象。最后，本文模型不包括安全培训，可能会产生有害的内容。