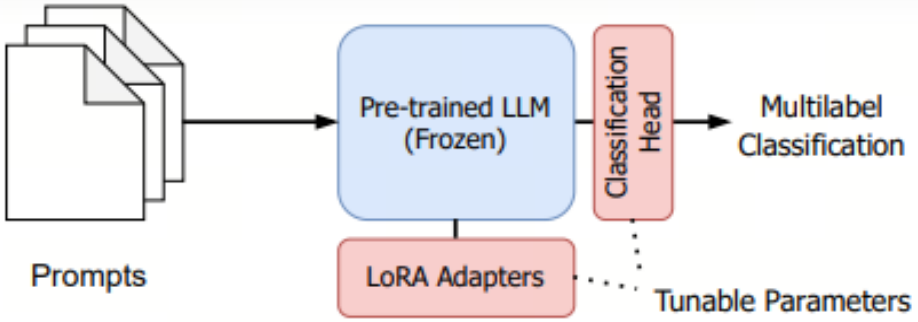
知乎




# 大模型如何个性化，最新论文解密独家方法

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注

6 人赞同了该文章

## Introduction

近年来，个性化语言模型的研究焦点在于如何提升LLMs在对话、推荐及敏感内容处理中的效用。文献如Kocon等（2021 2023）强调个性化在处理争议和冒犯内容中的关键作用，通过模拟用户个人视角提升预测准确性。Kazienko等人（2023）开发了考虑个体差异的深度学习模型，证实其在主观任务上的优异表现。ChatGPT和GPT-4+的实例显示，即使在少量示例下，LLMs也能展现适应个体需求的能力，凸显了其潜在的强大个性化潜力。传统如LSTM和Transformer架构也在个性化调整中占有一席之地，特别在处理敏感内容时，它们能与用户特征紧密匹配。

然而，当前研究正逐步转向理解和利用LLMs的零样本学习能力，以充分利用其通用性和适应性。微调通常优化LLMs以适应特定任务，可能提升性能，但伴随资源消耗增加且可能引发"灾难性遗忘"。大型模型的零样本推理能力则使其能从上下文和提示中自我学习。然而，微调对所有任务的全面效果尚无定论，尤其是对个性化的语料库任务，如情感识别和仇恨语料库检测，这方面研究相对不足。未来需进一步探究微调与零样本推理的权衡，以及如何在多任务环境下保持LLMs的泛化能力。

## Concept of Personalizing LLMs for Subjective Text Perception

在人际沟通中，理解文本的深层次含义不仅受限于语义，还包括个体的主观认知。对于涉及主观判断的任务，如情感识别、极端言论过滤、幽默解析和情感分析，模型需具备个性化，即根据用户特性调整回应。这要求我们在训练或生成阶段，可能需要持续地向模型输入用户的个人偏好和信念信息，或者一次性提供这些数据。

知乎

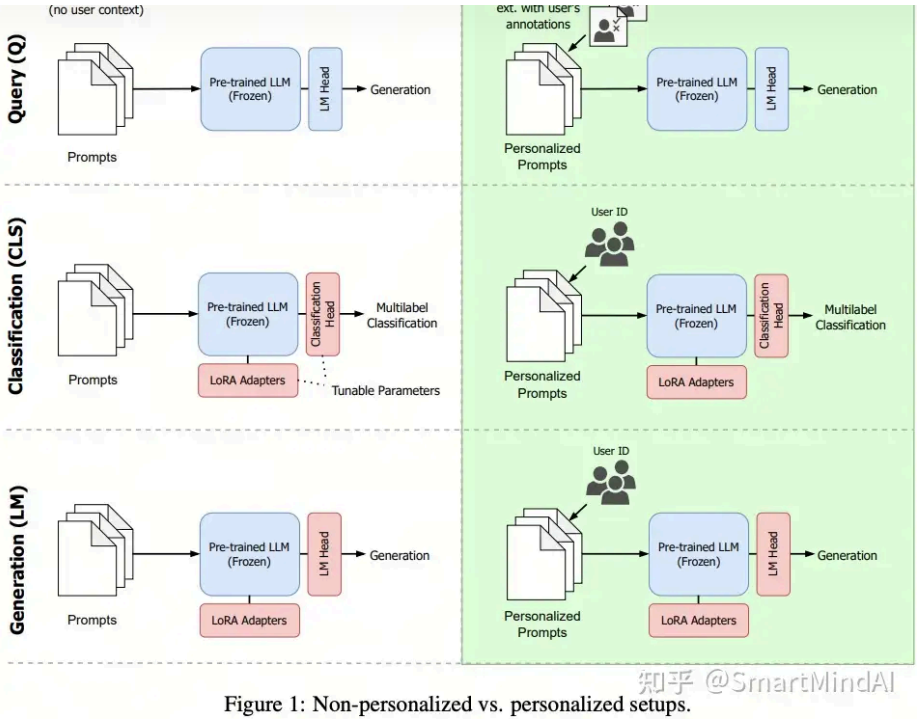



Figure 1: Non-personalized vs. personalized setups.

为此，我们研究了个性化LLMs（图示概念）与非个性化的基线方法，两者之间的比较旨在揭示如何优化LLMs以更好地适应个体需求。然而，目前对于如何在微调和零样本推理策略中平衡个性化和模型的泛化能力，以及如何利用用户信息的时机和方式，还有待深入探讨。我们已建立一个研究资源库来支持相关实验和研究的复现。

### Problem Definition

$\hat{Y}_u$代表基于用户$u$的预测结果$T$是文本输入$u$代表用户的信息（个人偏好或信念），而$f$是根据用户信息和文本内容的模型函数，其参数$\theta$控制模型*的行为和决策。这个模型在训练过程中通过学习用户特定的文本数据，使其学会理解和适应用户的交互风格。这个过程通常在训练阶段进行，目的是让模型能精准预测和回应用户的主观需求。$\hat{Y}_u = f(T, C_u)$在这个模型中$C_u$代表用户$u$的个性化上下文，它包含了用户独有的信息，如偏好和历史行为。

通过将$C_u$融入到$f$中，即$f(T, C_u; \theta)$，我们期望能获得更个性化的预测结果$\hat{Y}_u$。在训练阶段，这个过程通过学习用户数据，让LLM理解并适应$C_u$，从而生成针对用户特定需求的文本响应。这样，即使在零样本或少量样本的情况下，LLM也能展现出较强的适应性和理解力。

### Personalized Text Classification

我们提出一种利用用户特定信息$C_u$个性化LLM的方法，通过调整模型参数$\theta$以适应用户需求，或通过修改输入以包含个性化提示。优化目标是找到个性化模型，其预测误差最小化，同时保持对用户信息使用程度的平衡。具体公式如下：

$$L_{\text{个性化}}(C_u, \theta) = \min_\theta \mathbb{E}_{(X,Y)}[d(f(X;\theta), Y) - d(f(X;\theta)_{C_u}, Y) + \lambda R(C_u)]$$

这里$d$度量模型预测与真实输出的差距$f(X;\theta)_{C_u}$是在考虑用户信息后的预测$R(C_u)$是用于控制隐私保护的惩罚项$\lambda$是个调节参数。这个函数的目标是优化模型性能，同时确保用户信息的有效且适度利用。

$$\min_\theta \mathcal{L}(\theta; \hat{Y}_u, Y_u, T, C_u)$$

在这个语境中$\mathcal{L}$是衡量生成响应与用户预期目标$Y_u$相符性的指标。个性化可以通过两种方式进行：微调或上下文学习。微调是通过改变模型参数$\theta$以适应用户的具体需求；上下文学习则是直接将用户特征$C_u$融入到输入$T$中，使得模型在生成响应时参考这些信息。这两种方法都是为了提高生成的响应$\hat{Y}_u$与用户理想响应的匹配程度。

知乎

为了评估个性化对LLMs效果的贡献，我们构建了三个非个性化基线模型：

1. 原始指令优化模型：保持标准训练，不进行个性化的微调，直接应用通用模型处理所有任务。
2. 新嵌入分类模型：通过新增分类层，观察模型在不考虑用户特性的条件下处理主观任务的方式。
3. 生成式微调模型：虽然未针对特定用户，但通过对模型进行微调以适应任务，间接评估其在非个性化的环境下表现。

这些基线为评估LLMs在非个性化环境中的行为提供了参照，通过比较它们与个性化方法，我们可以量化个性化带来的性能提升。

### Querying Instruction-tuned Language Models (Q)

这种方法利用预训练LLM，它在指令优化下具备泛化能力，但未针对特定用户进行个性化。给定文本$T$，模型$\hat{Y}$通过固定的

$$\theta_{\text{instruction-tuned}}$$

生成响应，这是基于预先训练模型的函数$f$。这种策略保持通用知识，但忽略了用户的个性化需求和上下文，因此是中性处理。通过分析非个性化基线，如原始指令优化模型、新嵌入分类模型和生成式微调模型，我们能更精确地评估这种中立LLM在处理主观文本时的效果。$\hat{Y} = f_\theta(T)$ $f_\theta$代表预训练的LLM，由参数$\theta$驱动，它基于指令进行文本理解和生成。输入的文本$T$是其生成响应的源头。这种方法关注的是模型在不考虑用户个人特性和上下文信息的情况下，仅按照训练时学习的指令准则工作的性能。通过比较不同的非个性化基线，我们能了解这种通用模型在处理主观文本任务时可能存在的局限性及有效性。

### Classification Head and Model Fine-tuning (CLS)

在分类任务中，我们向LLM（由参数$\theta$定义的预训练模型）添加了一个附加的嵌入层，然后对模型进行微调。微调的目标函数是：

$$L_{\text{分类微调}}(T, y) = \min_\theta \mathcal{L}(f_\theta(T), y) + \Omega(\theta)$$

这里$\mathcal{L}$是衡量模型预测与真实标签$y$偏差的损失函数，比如交叉熵⁺ $f_\theta(T)$是经过新嵌入层处理的输入文本的预测结果。$\Omega(\theta)$是针对任务特性的正则化项，以防止过拟合⁺。我们的目标是找到使分类准确性的最佳$\theta$，同时保持对原模型结构的约束，利用LLM的泛化能力。这种基于新嵌入头的微调策略旨在增强模型在特定分类任务上的表现。

$$\min_{\theta'} \mathcal{L}_{CLS}(\theta'; \hat{Y}, Y_u, T)$$

在该情景中$\mathcal{L}_{CLS}$代表分类任务的损失$\theta'$代表经过额外分类头层微调后的模型参数。输入是文本$T$，目标是用户标记的标签$Y_u$。通过这种方式，我们试图改进LLM在特定任务上的表现，但没有直接针对用户个性，以保持模型的广泛适用性。这种设置是为了在保持泛化能力的同时，提升模型在分类任务上的准确度。

### Generative Fine-tuning via Language Modeling (LM)

$$\hat{Y}_u = g(T, P; \theta')$$

其中$g$是经过重新训练的模型$P$是指导性的提示$\theta'$是更新后的参数集，包含额外的模型结构变化。这种方法认为模型能理解并生成与输入文本相关的个性化标签，即使未针对每个用户单独优化。这种方法允许模型在保持泛化能力的同时，学会根据特定输入和提示生成符合用户期望的分类输出。

$$\min_{\theta''} \mathcal{L}_{LM}(\theta''; \hat{L}, L_u, T)$$

在这个语境中$\mathcal{L}_{LM}$是用于文本生成任务的损失，通常计算的是各位置的交叉熵损失之和。经过生成式微调后的参数集是$\theta''$。输入的文本是$T$，而非直接预测的标签，而是通过模型生成。与分类不同，生成式微调不是直接在输出中定位标签，而是让模型在生成整个文本的过程中理解和学习用户意图，从而生成更贴近用户预期的分类标签。

知乎

少量示例个性化、个性化分类和个性化语言建模，它们通过利用用户特定数据来提升模型在主观文本处理任务中的相关性和准确性。

### Few-shot Personalization (Q-NS)

少量示例个性化通过极少量用户特定示例（如$N$条文本$E_1, E_2, \ldots, E_N \in E_u$）来调整模型，使其更贴近用户独特的解读。这种方法在上下文学习中操作，将用户上下文$C_u$融入输入文本$T$，形成带有个人特色的上下文。模型以此为基础生成个性化的响应$\hat{Y_u}$，通过这种方式学习和模仿用户的独特交互方式。

$$C_u = (E_1, E_2, \ldots, E_N)$$

$$\hat{Y_u} = f_\theta(T, C_u)$$

在这个场景里$f_\theta$代表预训练LLM$\theta$是其参数。初始输入文本是$T$，附加的用户注释样例$E_i = (T_i', Y_i')$包含用户独特看法的文本$T_i'$，而非简单的原始文本，但$T_i'$并非等于$T$。这些示例$E_i$构建了用户观点的上下文$C_u$。目标是利用这些例子引导模型去理解和生成与用户个人见解相符的响应$\hat{Y_u}$。通过这种方式，模型调整其初始输入以更好地理解和生成个性化的输出。

### Personalized classification (CLS-P)

个性化分类通过将用户标识$I_u$整合到模型训练中，让LLM针对特定用户进行调整。通过优化函数：

$$L_{\text{个性化分类}}(T, Y_u, \theta) = \min_\theta \sum_{i=1}^{n} \mathcal{L}(f_\theta(T, I_u), Y_i) + \Omega(\theta)$$

其中$T$是输入文本$Y_u$是用户独有的标签$I_u$是用户特征$n$是样本数$\mathcal{L}$衡量预测与实际标签的差异。$f_\theta(T, I_u)$是考虑用户信息的模型预测$\Omega(\theta)$是防止过拟合的正则化项。目标是找到使预测尽可能接近用户实际标签的$\theta$，以提升分类的精确性和个性化水平。

$$\min_{\theta'} \mathcal{L}_{CLS-P}(\theta'; \hat{Y_u}, Y_u, T, C_u)$$

在这个语境中$\mathcal{L}_{CLS-P}$代表个性化分类的损失函数，它涉及到使用个性化的模型参数$\theta'$。输入是文本$T$，用户目标是标签$Y_u$，附加的用户特定上下文信息是$C_u$，比如用户ID。这个方法通过融合用户特征，使得模型预测更精确且具备个性化，因为它直接从用户特有的角度影响模型的决策过程。这样，模型能够更深入地理解并满足每个用户的特定需求。

### Personalized Languge Modeling (LM-P)

$$\min_{\theta'} \mathcal{L}_{LM-P}(T, Y_u'; C_u) = -\sum_t \log p_{\theta'}(t|T, Y_u'; C_u)$$

其中

$$p_{\theta'}(t|T, Y_u'; C_u)$$

是基于$T$、用户提供的个性化标签$Y_u'$和用户特定上下文$C_u$的每个词的生成概率。通过最大化这个概率，模型在生成文本时会考虑用户的具体需求和习惯，从而达到个性化的目的。

$$\min_{\theta''} \mathcal{L}_{LM-P}(\theta''; \hat{L_u}, L_u, T, C_u)$$

在该场景中$\mathcal{L}_{LM-P}$代表个性化语言建模的损失$\theta''$是处理过的模型参数，用于生成更贴合用户偏好的文本。输入文本是$T$，目标输出是用户期望的文本$L_u$，通过用户特定的上下文$C_u$（如用户标识）来体现。通过微调，模型能理解和适应用户需求，生成定制化的文本，以满足用户个性化要求。优化函数旨在通过最大化生成文本与用户期望的匹配度来实现这一目标。

### Datasets

实验中，我们利用了两个英语数据集：GoEmotions和Unhealthy Conversations。GoEmotions包含58,000个Reddit评论，由82个标注者共标注了211,000个情感类别，经过筛选，我们保留了

练集有168,000条，验证和测试各20,000条。在数据处理阶段，我们特别注意了异常标注者（低标注频率）的排除，以确保数据的准确性和完整性。

### Models

我们实验中使用了两种大型语言模型✦：稳定性AI的3B和7B参数无解码语言模型，它们通过在多种对话和指令数据上进行细致的微调，以增强其在聊天场景中的应用。这些模型基于NeoX变压器架构，专为此目的设计。此外，我们还采用了EleutherAI提供的20B参数的GPT-Neox模型。我们特别提到，30亿参数的模型来自HuggingFace模型库。 对于ChatGPT，我们利用了OpenAI团队的GPT-3.5和GPT-4模型。GPT-4在零样本推理等多元任务上展现出卓越性能。

### Experimental setting

实验部分分为两部分，分别从数据集和模型两个维度展开。我们关注的是探究个性化微调和在上下文中的学习如何影响不同方法在具体任务上的效果。我们主要使用Transformer架构的无编码解码✦语言模型，如稳定性AI的3B和7B，以及EleutherAI的GPT-Neox，它们相较于带编码器-解码器的Flan-T5在微调上资源需求较低。

对于语言建模任务，我们仅在无编码器模型间进行了对比，没有包含Flan-T5的编码器解码器版本。然而，对于其他个性化方法，我们与Mistral的7B无编码器模型进行了较量，这个模型的规模比Flan-T5大两倍以上，以此来评估不同模型的有效性。

在个性化微调实验中（LM-P, CLS-P），我们通过在提示插入 `'### 用户ID:<用户ID> '` 来关联用户身份。在查询任务（Q-0S, Q-1S, Q-2S）和语言建模（LM, LM-P）中，明确要求模型给出标签列表，提示包含'请将你的回复作为选择的标签列表，用逗号分隔。

Q-1S和Q-2S还额外提供了示例文本和正确答案，每个用特定模板组织，新行分隔，示例编号用 `<N>` 表示。在Q-1S中， `<N>` 为空；而在Q-2S中， `<N>` 会依据例子编号变化。

### Results and discussion

我们定义了一个名为'增益'的度量，用于量化个性化模型（LM-P, CLS-P）相对于非个性化基线模型的质量提升百分比。这个指标通过比较个性化处理后的结果和未处理的基线结果来评估模型✦性能的改进。

$$gain = \left( \frac{\text{personalized} - \text{baseline}}{\text{baseline}} \right) \times 100\%$$

知乎

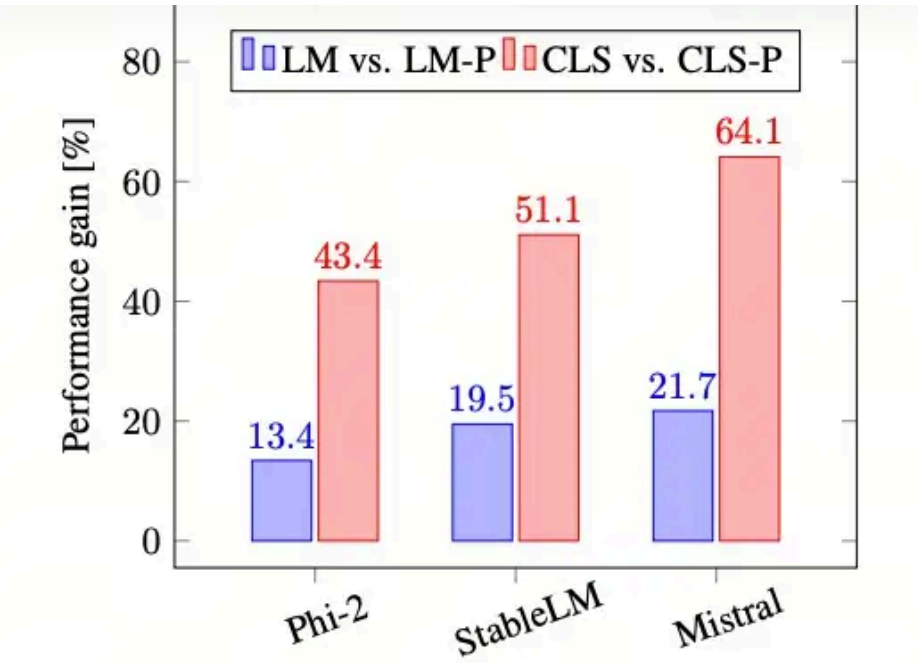



Figure 2: Performance gains of personalized vs. non-personalized methods on the GoEmotions dataset.

实验结果表明，个性化处理对主观任务分类显著提升了模型性能，无论是通过个性化微调（CLS-P和LM-P）还是与无个性化基线（如Q-0S和Q-1S, Q-2S）对比。在GoEmotions数据集上，虽然效果相对较强，但在Unhealthy Conversations数据集上，这种提升更为显著。这符合之前研究，强调了针对任务特性的定制化提示（Q-1S和Q-2S）在某些情况下可能是最优策略。

| Model \ Setting | | LM | LM-P | CLS | CLS-P |
|---|---|---|---|---|---|
| | | **GoEmotions** | | | |
| Phi-2 | 2.7B | 28.99 | 32.87 | 30.03 | 43.07 |
| StableLM | 3B | 26.55 | 31.72 | 27.42 | 41.44 |
| Mistral | 7B | 28.36 | 34.52 | 26.77 | 43.94 |
| | | **Unhealthy Conversations** | | | |
| Phi-2 | 2.7B | 34.97 | 45.89 | 31.91 | 48.26 |
| StableLM | 3B | 29.61 | 48.54 | 16.92 | 44.68 |
| Mistral | 7B | 34.29 | 51.65 | 23.10 | 52.83 |

实验表明，模型对少量样本的学习能力在GPT系列[+]，尤其是GPT-3.5和GPT-4，展现出一致性，这可能源于其对扩展用户上下文的利用。然而，Mistral模型在Q-1S和Q-2S设置下未能充分利用这一点。

Phi-2[+]模型由于未经历指令训练和对齐过程，其性能在个性化任务中较差。 在LM和CLS任务之间，个性化的性能在不同数据集上显示出差异。在GoEmotions，CLS-P在个性化设置中占优，因为数据集情感标签丰富，对语言建模精度要求高。但LM-P在Unhealthy Conversations数据集上有时表现更好，可能缘于其更适合处理这类任务的复杂性和细微情感表达。具体差异源于任务特性对模型适应度的不同影响。

知乎

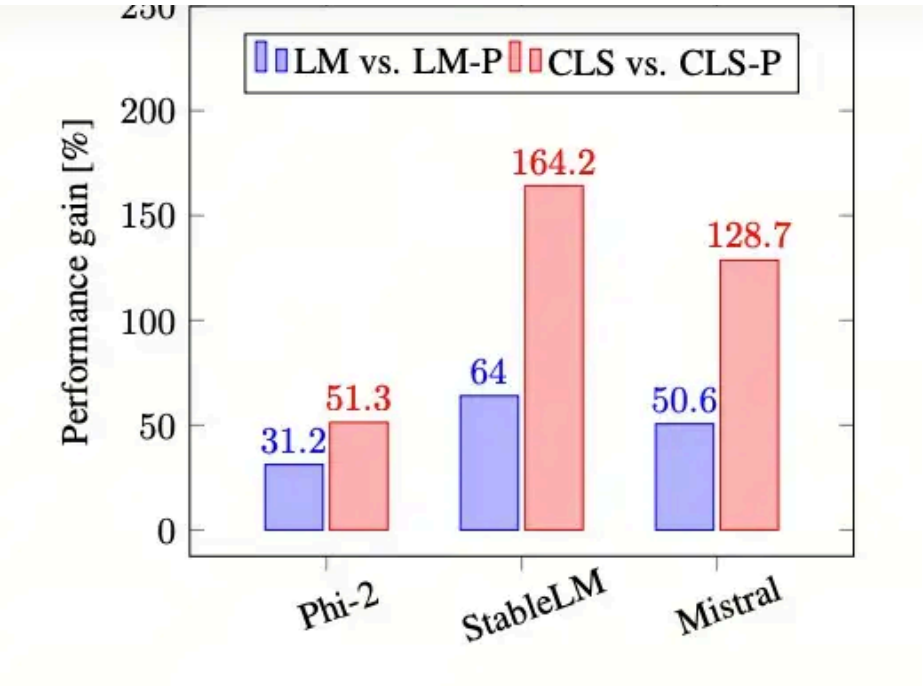



Figure 3: Performance gains of personalized vs. non-personalized methods on the Unhealthy Conversations.

在Unhealthy Conversations数据集上，LM-P和CLS-P的性能差距不如GoEmotions显著，这可能是因为Unhealthy Conversations的标签数量较少，这突显了标签复杂性对个性化微调策略效能的关键作用。任务特定的微调方法显得尤为重要。数据集特性对模型优化至关重要。对比7B纯解码器模型Mistral（比Flan-T5大两倍以上）和参数为3B的编码器-解码器Flan-T5，尽管纯解码器理论上可能有更高的增益，但在分类任务的个性化设置（CLS vs CLS-P）中，编码器-解码结构的性能更优。

| Model \ Setting | | CLS (F1-macro) | CLS-P (F1-macro) | Gain [%] |
|---|---|---|---|---|
| | | **GoEmotions** | | |
| Flan-T5 | 3B | 32.64 | 45.68 | 39.95 |
| Mistral | 7B | 26.77 | 43.94 | 64.14 |
| | | **Unhealthy Conversations** | | |
| Flan-T5 | 3B | 38.57 | 59.42 | 54.06 |
| Mistral | 7B | 23.10 | 52.83 | 128.70 |

**原文《Personalized Large Language Models》**

发布于 2024-04-28 11:10 · IP 属地北京

个性化 大模型 大语言模型

▲ 赞同 6 ▼ 添加评论 分享 喜欢 收藏 申请转载 ···

理性发言，友善互动