从基于大语言模型的任务机器人中抽取训练对话数据

Shuo Zhang、Junzhou Zhao、Junji Hou、Pinghui Wang、Chenxu Wang、Jing Tao

arXiv:2603.01550v1 [cs.CL] 2 Mar 2026

摘要——**大语言模型（LLM）已被广泛应用于增强面向任务的对话系统（任务型对话系统），通过建模复杂的语言模式并提供符合上下文的回复。然而，这种集成也带来了显著的隐私风险：大语言模型作为软知识库，会将大量训练数据压缩为丰富的知识表示，因此可能无意中记住包含可识别信息（如电话号码）甚至完整对话级事件（如完整旅行日程）的训练对话数据。尽管这一隐私问题至关重要，但开发任务机器人时如何继承大语言模型的记忆能力，仍未得到探索。在这项工作中，我们通过一项系统的定量研究填补这一空白：评估现有的训练数据抽取攻击，分析使现有方法失效的任务型对话建模关键特征，并提出针对基于大语言模型的任务型对话系统的新型攻击技术，以改进响应采样和成员推断。实验结果证明了我们提出的数据提取攻击的有效性。我们的方法能够抽取数千条对话状态训练标签，最佳情况下精确率超过 70%。此外，我们还通过识别并量化关键影响因素，深入分析了基于大语言模型的任务型对话系统中的训练数据记忆，并讨论了有针对性的缓解策略。**

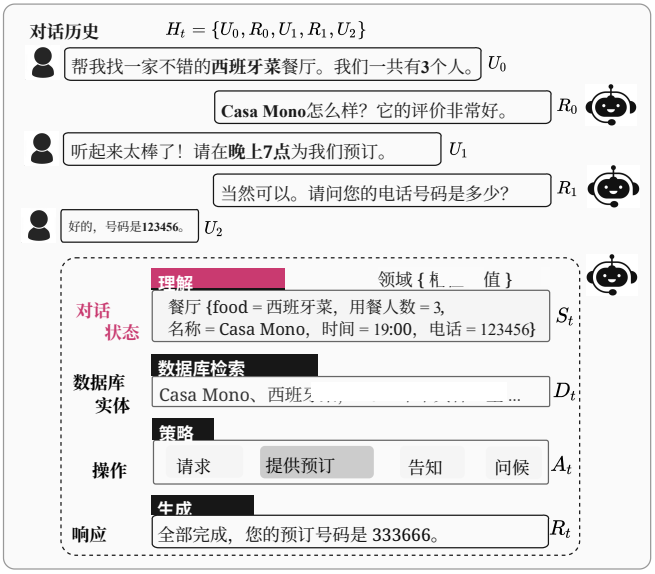
索引术语—**大语言模型，面向任务的对话系统，训练数据抽取攻击，隐私泄露**

I. 引言

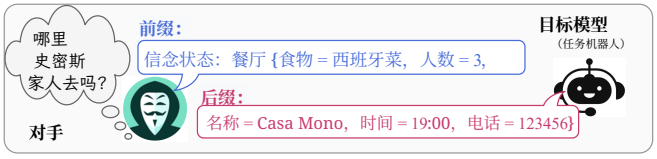
TASK-Oriented 对话系统（任务型对话系统），即任务机器人，在协助用户完成旅行预订和医疗咨询等特定任务方面发挥着至关重要的作用 [1]。大语言模型（大语言模型）在涵盖广泛人类知识的海量数据集上接受训练，通过更准确地理解用户意图并生成相关回复，大幅提升了任务型对话系统的性能 [2]–[7]。尽管取得了重大进展，集成大语言模型仍会引入显著的隐私风险，因为大语言模型具备记忆并在无意中泄露敏感训练数据的能力 [8]–[10]。训练数据记忆通常被归因于为了实现有效的响应泛化而必须保留特定信息，其中模型依赖于从单个训练样本进行复制和插值 [11]。

尽管此前已有关于大语言模型训练数据记忆的研究，但在理解大语言模型针对任务型对话系统进行微调时，如何继承这种记忆方面，仍存在研究空白。

S. Zhang、J. Zhao、J. Hou、P. Wang、C. Wang 和 J. Tao 就职于西安交通大学智能网络与网络安全教育部重点实验室，邮政信箱 1088，咸宁西路 28 号，西安，陕西 710049，中国。电子邮箱：{zs412082986, 15955192}@stu.xjtu.edu.cn,{junzhou.zhao, phwang, cxwang, jtao}@mail.xjtu.edu.cn（通讯作者：Junzhou Zhao。）



(a) 用户与任务机器人之间交互的案例示例。



(b) 基于大语言模型的任务机器人的对话状态抽取攻击。

图 1. (a) 基于大语言模型的任务机器人利用单个神经自回归模型来参数化顺序对话流水线。(b) 我们的对话状态抽取攻击抽取对话状态 (S) 的训练标签，无需 访问相应对话历史的条件文本。

考虑到任务型对话系统在全球现实应用中的广泛部署，这一研究空白尤其令人担忧。这些系统本质上会处理敏感的个人数据，不仅包括电话号码和个人地址等可识别信息，还包括对话层面的事件，其中包含用户详细的个人日程和偏好，例如旅行计划和爱好。请看下面这个例子。

示例 1（餐厅预订助手机器人）。 餐厅预订助手机器人可帮助用户寻找餐厅并进行预订。如图所示。*1*，一段对话由多个轮次的用户—系统话语组成。在每个轮次中，机器人都会根据对话历史跟踪对话状态。对话状态由从先前用户话语中提取的一组用户偏好和约束条件构成，例如食物类型和预订时间。随后，机器人会查询数据库

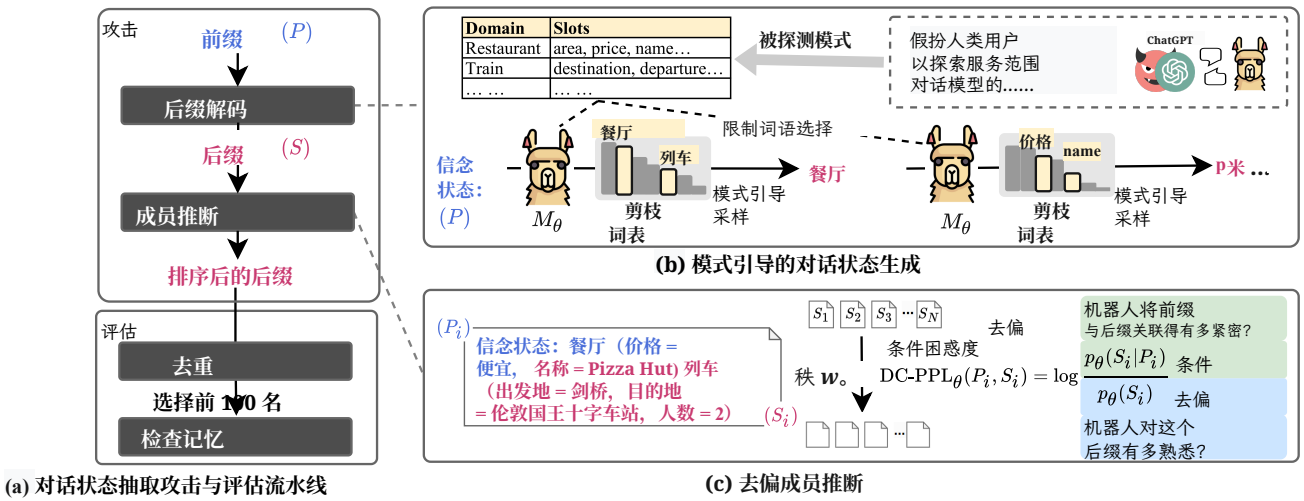


图 2. **（a）对话状态提取攻击的工作流程：**我们首先使用**后缀解码**，通过向模型输入空内容（用于无目标攻击）或部分对话状态前缀（用于目标攻击），从基于大语言模型的任务机器人生成大量样本。随后，使用**成员推断**指标对每次生成进行排名，排名越高，表示其越可能源自训练数据。**在评估时**，我们对生成结果去重，保留排名前 100 的结果，并将每个生成结果与训练数据集中的对话状态进行比较，以标记其是否为记忆内容。**（b）模式引导采样 §IV-B2** 通过使用对话模式（即服务范围）约束采样时领域和槽位的选词，生成更多有效结果；对话模式由 ChatGPT 模拟用户交互自动探索得到。**（c）去偏条件困惑度 §IV-C** 降低了对常见通用对话状态组件的偏向。

根据对话状态预测适当的系统动作，例如提供预订服务，并生成适当的响应。实际上，对话状态是训练数据的一部分。因此，能够访问任务机器人 *API* 的对手可能会恶意诱使机器人输出其对话状态，例如“餐厅（名称=*Casa Mono*，食物=西班牙菜，时间=*19:00*，电话=*123456*，人数=*3*）”，从而泄露用户的隐私信息。

在本研究中，我们通过实施训练数据提取攻击，探究基于大语言模型的任务机器人如何记忆训练对话数据——具体而言，是维护来自对话历史的约束条件的对话状态——[8], [10], [11]。这些攻击旨在通过基于分数的黑盒查询访问，从微调语言模型中恢复训练标签，并且通常遵循两阶段流程。如图2所示，第一阶段在后缀解码步骤中通过各种采样策略提示模型生成大量候选对话状态。第二阶段应用成员推断指标对这些候选进行排序，标记那些最有可能源自训练数据集的候选。

由于任务型对话系统的结构特性，从基于大语言模型的任务机器人中抽取训练对话状态本质上具有挑战性。与开放式对话模型不同，任务型对话系统从未被训练来复现用户输入话语——因此，原始输入不会被记忆，也无法从模型中抽取。相反，该模型被优化为根据先前上下文预测结构化对话状态，从而使抽取只能通过输出空间实现。此外，任务型对话系统必须在多个对话轮次中保持一致性与连贯性 [1]。当前置上下文仅被部分观测或被截断时，恢复正确的后续状态变得极具挑战。最后，即便已知全部上下文，对话回复的一对多特性（即针对相同的前置上下文）进一步增加了数据提取的难度，因为所抽取的回复未必反映原始训练数据。

为应对这些挑战，我们做出两项主要贡献。首先，我们观察到，不完整的前置上下文通常会导致现有后缀解码方法生成不连贯或无意义的对话状态。为缓解这一问题，我们提出一种模式引导采样方法，该方法利用对话模式（即服务范围）来约束词语选择，从而确保生成输出的多样性与有效性。其次，我们发现，对话的一对多特性会导致现有成员推断指标出现模型偏差，这些指标往往偏向信息贫乏的通用对话片段（例如问候）。为克服这一问题，我们引入去偏条件困惑度指标，该指标在评估输入—输出蕴涵的同时，减轻模型对通用片段的过度熟悉。

实验结果表明，我们的方法能够有效抽取数百至数千个训练对话状态。我们考虑了两种抽取设置，即定向提取和非定向抽取。定向提取会向任务机器人提供部分对话状态作为提示，而非定向抽取则不会。在非定向抽取设置下，我们发现，对话状态中的值（例如名称）具有相对较高的隐私泄露风险，最高抽取精确率可达 67%；而完整对话状态引发的担忧较少，最高精确率为 26%。在定向提取设置下，我们的方法精确率显著提高：单个值的最佳精确率达到 100%，事件级状态的精确率则超过 70%，这表明其隐私泄露风险大幅增加。此外，我们还确定了影响任务机器人记忆的关键因素，即轮次级对话训练数据中固有的子串重复增多，这会增强记忆；以及对话的一对多特征，即多个回复可能对应

同一个提示词，这可能会减弱记忆。基于我们的发现，我们进一步讨论了切实可行的缓解策略，包括采用对话级建模来解决子串重复问题，以及采用值复制机制来减轻上下文相关性。

我们的贡献总结如下：

- 据我们所知，这是首项研究基于大语言模型的任务机器人训练数据记忆问题的工作。在这类系统中，隐私泄露发生在结构化信念状态和跨槽位任务语义层面，而非孤立的话语片段。
- 我们首次开展了系统性研究，包括：(i) 建立任务型对话系统的数据提取威胁模型；(ii) 在这一新背景下对现有技术进行系统性基准测试；(iii) 识别领域特定偏差与挑战；(iv) 提出两种针对该任务的方法调整：一种模式引导的采样策略，通过利用任务模式约束，高效生成有效的对话状态候选项；以及一种去偏条件困惑度指标，用于校正现有基于困惑度的成员推断方法中的模式频率偏差。
- 在各种配置下进行的大量实验表明，我们的攻击技术会造成严重且切实的影响，并凸显了显著的隐私泄露风险。我们还识别了影响基于大语言模型的任务机器人记忆的因素，并讨论了有针对性的缓解策略，以指导未来的实现。

II. 预备知识与相关工作

在本节中，我们首先阐述基于大语言模型的任务机器人及微调过程的一些背景，然后介绍现有的训练数据提取攻击。

A. 基于大语言模型的任务机器人

面向任务的对话系统，也称任务机器人，旨在通过多轮对话帮助用户完成特定任务，例如预约和医疗咨询 [1]。近年来，大语言模型的进展彻底改变了任务机器人，使预训练大语言模型成为其实现的基础 [2]–[6]。这些方法涉及使用面向任务的对话数据，对预训练的因果变换器模型（如 GPT 系列）进行微调。SimpleTOD [5] 是一种基础方法，它将面向任务的对话重新定义为条件语言建模任务。具体而言，该方法对预训练的 GPT-2 模型进行微调，使其根据对话上下文和从数据库检索的结果，生成中间标签（如对话状态和对话动作）以及最终响应。SOLOIST [3] 通过省略对话动作预测以减少错误传播，并引入对比目标来增强响应生成，从而改进了上述方法。后续改进包括 AuGPT [2], ，它添加了用户意图和系统动作分类等辅助任务；以及 UBAR [4], ，它通过将对话状态和动作纳入历史记录，将方法扩展到会话级示例。虽然基于编码器—解码器大语言模型的任务机器人（如 T5 [6]）也取得了令人瞩目的成果，但本研究仍聚焦于仅解码器大语言模型，这是该研究方向的基础步骤。在实际应用中，任务机器人通常会持续或定期更新，以便反复将近期对话数据吸收到其参数中。

B. 面向任务的对话的因果语言建模

如图1所示，从高层来看，基于大语言模型的任务机器人本质上被训练来建模联合概率分布 $p(S_t, D_t, A_t, R_t|H_t)$。此处 $H_t \triangleq \{U_0, R_0, \ldots, U_{t-1}, R_{t-1}, U_t\}$ 表示对话历史，其中包含用户话语 $\{U_i\}_{0\le i\le t}$ 和系统响应 $\{R_j\}_{0\le j<t}$。给定 H_t，模型首先预测对话状态 S_t ，该状态跟踪先前聊天中提及的一组用户约束。

考虑以下文本形式的对话状态示例：“信念状态：餐厅 {食物 = 意大利菜，区域 =中心}酒店 {电话 = 12345}”。每个对话状态由一个或多个领域–槽位–值三元组组成，例如（餐厅，食物，意大利菜）或（酒店，电话，12345）。此处，领域 指定主题（例如，餐厅），每个槽位表示一个属性（例如，食物），每个值表示用户偏好或所提供的信息。随着对话的进行，每当用户添加、修改或移除约束时，对话状态都会动态更新。

随后，利用对话状态从领域数据库中检索相关数据 D_t ，以满足用户约束条件。然后，模型结合利用对话历史、对话状态和检索到的数据来预测操作 A_t，其中包含意图及相关槽位值。例如，操作“inform(domain=hotel,price=expensive)”的意图是 告知，表示用户告知系统将搜索范围约束为高价酒店。最后，系统根据对话历史、对话状态、检索到的数据和操作生成系统响应 R_t 。因此，联合概率可以分解为

$$p(S_t, D_t, A_t, R_t|H_t) = \underbrace{p(S_t|H_t)}_{\text{state prediction}}\ \underbrace{p(A_t|H_t, S_t, D_t)}_{\text{action prediction}}\ \underbrace{p(R_t|H_t, S_t, D_t, A_t)}_{\text{response generation}}$$

其中， $p(D_t|H_t, S_t)$ 被省略，因为检索到的数据 D_t 是通过确定性约束匹配获得的，因此$p(D_t|H_t, S_t) = 1$。这种分解将语言建模任务划分为三个子任务，即状态预测、动作预测和响应生成。

微调期间，大语言模型以端到端的方式针对这三项任务进行优化。也就是说，给定一个训练序列 $\mathbf{x}^t_{\text{train}} = \{(H_t, S_t, D_t, A_t, R_t)\}$，以 θ 为参数的大语言模型经过训练，以最小化负对数似然损失

$$L_\theta(\mathbf{x}^t_{\text{train}}) = -\log p_\theta(S_t|H_t) - \log p_\theta(A_t|H_t, S_t, D_t) - \log p_\theta(R_t|H_t, S_t, D_t, A_t).$$

C. 训练数据提取攻击

大语言模型由于参数数量庞大，可能会记忆训练数据。Carlini 等人 [8]率先探索了模型知识提取，并定义了

κ-情景记忆，证明了通过有针对性的查询从 GPT-2 中恢复单个训练样本的可行性。在 Carlini 等人 [8], 的研究中，互联网上下文被用作前缀，以引出可能反映训练数据的后缀。随后，通过成员推断攻击检查这些示例是否来自训练数据集。后续研究 [9], [12] 发现，数据重复、上下文长度和模型规模会显著影响这一记忆现象。Yu 等人 [10] 通过改进文本解码和成员推断策略，提高了从预训练语言模型中提取数据的效能。在黑盒设定下，针对 ChatGPT 等经过对齐的生产环境中的大语言模型， [13] 提出了一种新型攻击：迫使语言模型无休止地重复“poem”一词，从而破坏聊天机器人式输出，并在数百次迭代后导致输出偏离。Carlini 等人 [11] 将类似技术应用于扩散模型，而 Qi 等人 [14] 则设计了通过提示注入从检索增强生成系统中提取数据库信息的方法。Zeng 等人 [15]进一步研究微调模型是否会记忆输入序列，以及能否通过提示使其复现这些序列。

然而，以上研究的目标要么是预训练记忆，要么是开放式生成。这些模型经过优化，能够自由复现输入或输出文本，因此可以逐字恢复内容。相比之下，面向任务的对话系统（任务型对话系统）采用条件语言建模目标进行训练，以预测结构化对话状态，而非原始话语。在任务型对话系统中，即使某个上下文在训练期间频繁出现，它也仅用作条件输入，并未针对生成进行优化，因此输入序列不会被记忆。潜在的信息泄露反而存在于模型输出中出现的结构化槽位–值对。这一区别至关重要：开放式对话记忆涉及自由形式文本复现，而任务型对话系统的记忆则涉及话语与结构化状态之间受模式约束、类似复制且高度依赖上下文的映射。因此，为开放式抽取开发的方法无法直接应用于任务型对话系统，因为缺乏适当的模式落地时，这些方法无法诱导模型生成有效的结构化输出。

因此，我们的研究填补了这一空白。我们没有考察预训练记忆或开放式记忆，而是首次对基于大语言模型的任务机器人中的训练数据提取开展系统性研究，揭示记忆如何体现在结构化的条件生成过程中。通过全面实验，我们量化了现有解码和成员推断方法在模式约束下失效的情形，并提出有针对性的任务感知型调整，以更好地衡量和缓解这一风险。

III. 威胁模型

A. 记忆的定义

在这项工作中，我们探究从基于大语言模型的任务机器人中抽取训练对话状态的能力。这类机器人学习根据相应的对话历史作为上下文来预测对话状态。与现有攻击不同，我们假设在进行对话状态提取时无法访问训练对话历史。因此，我们将对话状态记忆定义为仅将前缀限制为包含前一个对话状态，如下所示。

定义 1（任务机器人的对话状态记忆）。 若存在长度为 l 的部分对话状态前缀 x ， 使得拼接 $x||s$ 出现在训练数据中，并且 f_θ在仅以 x 作为提示、使用贪心解码、且不以相应对话历史为条件时生成 s ， 则带有由 l 个领域-槽-值三元组组成的前缀的对话状态字符串 s 被视为被模型f_θ记忆。

例如，如果训练数据包含以下序列。

用户：我想找一家提供中餐的高档餐厅。**系统**：好的，您偏好哪个区域？ **用户**：城镇北部。信念状态：餐厅(价格_范围=昂贵,食物=中餐,区域=北部)

当仅提供长度为1的部分对话状态、而不以对话历史作为前缀时，例如“信念状态：餐厅(价格_范围=昂贵”，模型最有可能生成“食物=中餐,区域=北部)”作为输出。这部分对话状态可被视为可抽取或已被记忆。请注意，前缀还可以进一步简化为仅仅“信念状态：”。我们将这种本质上更具挑战性的情形称为非定向抽取，而纳入部分状态前缀的做法则称为定向提取。

B. 目标与对手

1) 对手的目标： 对手的目标是从基于大语言模型的任务机器人中抽取其记忆的训练对话数据。乍看之下，人们可能会考虑直接通过提示大语言模型来抽取对话历史 H_t。然而，这不可行，因为模型经过专门训练，给定 §II-B 所述内容时，预测的是对话状态和后续响应，而非对话历史。相反，对话历史仅在训练任务机器人时用作条件上下文。因此，对话历史并未被记忆，无法直接抽取对话历史数据。因此，我们旨在抽取用于概括对话历史的对话状态（见图 1），这与现有关于从大语言模型中抽取训练数据的工作有很大不同 [8]。

具体而言，我们在基于得分的黑盒假设下攻击任务型对话系统（TODS）：我们只能访问由以下参数化的基于大语言模型的任务机器人 θ的输入和输出。我们通过向任务机器人提供对话状态起始短语 P，例如“信念状态：”和“[BOS]”[1], , 获取下一词预测的概率分布 $p_\theta(x_i|P,\ x_1,\ .....,\ x_{i-1})$，并据此采样生成文本序列，再将其中的对话状态部分作为我们的抽取候选。接下来，我们使用多种成员推断度量对这些候选进行排序，以确定这些抽取结果是否源自

[1]对话状态通常会以“信念状态：”等短语或“[BOS]”等特殊词元标记，以便模型区分不同任务。攻击者可以轻松生成一个样本，以确定使用了哪些词元或短语。

训练数据集。在此，我们追求两个对抗目标，即训练数据中的对话状态抽取和成员推断。

- **对话状态提取**：对手旨在促使任务机器人生成大量且多样化的独特对话状态，从而最大限度地覆盖训练数据集。
- **成员推断**：对手试图准确判断某个特定的对话状态、模式或值是否存在于训练数据集中，以便有效排除噪声候选项。

2）对手能力：在实际部署中，ChatGPT 等大语言模型（LLM）不会公开其底层参数。因此，本文关注对手能够以基于分数的黑盒方式访问由大语言模型驱动的任务机器人的场景。具体而言，对手可以查询系统，获取任意序列的概率（得分）以及下一词预测，但严格禁止访问任何内部权重或隐藏表示，包括注意力矩阵。一些平台可能会进一步加强安全防护，只提供文本输出，并向用户隐藏所有概率信息。虽然我们完全认同严格的黑盒设置十分重要，但我们强调，部分状态泄露在实际部署中既有可能发生，也会产生重大影响。

在实践中，对话状态可能通过多种渠道部分暴露。例如，内部工程师或第三方服务提供商可能会访问包含部分对话状态的调试日志；平台 API 可能会开放用于监控或集成的影子端点；终端用户本身也可能在跨应用同步时无意间泄露部分状态。为具体说明这种风险，可以设想这样一种情形：恶意行为者通过旁信息获知用户旅行计划的部分内容，随后查询对话系统，以推断行程的更多细节。这类情况并不需要访问完整的训练语料库，但仍表明，对话状态轨迹的有限泄露也可能造成实质性的隐私风险。

本研究聚焦于这种部分访问威胁模型，将其视为一种介于中间但在实践中切实可行的情形。它反映了这样一种情况：对手能够观察到部分对话状态，却无法直接访问底层模型权重或训练数据。研究这种中间情形，使我们能够在较弱但现实的假设下系统评估潜在的隐私泄露，并为未来适用于完全黑盒系统的防御措施设计提供参考。

3) 目标模型：在本研究中，我们以最先进的 Llama2（7B）为例，[16] 将其用作骨干大语言模型，构建用于攻击的目标任务机器人。请注意，我们的技术可以轻松扩展到其他大语言模型。该大语言模型采用 [17]§II-B节所述的方法，在 MultiWOZ 基准V-A节中进行了微调。有关 MultiWOZ 数据集和模型微调的更多详情，请参阅

IV. 对话状态提取方法

本节首先介绍一种稻草人方法，该方法利用现有的、为从大语言模型中抽取预训练数据而设计的方法，并讨论其在用于从基于大语言模型的任务机器人中抽取训练对话状态时的局限性。随后，我们设计了新方法来解决这些局限性。

A. 对话状态提取的稻草人方法

请注意，先前的工作已探索通用训练数据提取攻击 [8]–[10]。接着，一种稻草人方法是直接利用现有的文本生成方法来提取训练对话状态。该稻草人方法包括两个步骤，即候选对话状态生成和成员推断。

1) 候选对话状态生成：对于对话状态生成，我们遵循无目标抽取攻击设定，即不提供训练对话历史或对话状态信息来引导生成。具体而言，我们只需向任务机器人输入提示“信念状态：”，然后以自回归方式从模型中采样词元，使模型仅根据这一最简提示生成回复。我们的目标是生成模型认为可能性很高的序列，并假设这些高概率序列体现了记忆文本。

然而，我们观察到，任务机器人倾向于生成空的对话状态，从而给出诸如“我能为您做些什么？”这样的通用回复。这种行为源于训练与测试（抽取）设置之间的差异。具体而言，在训练期间，任务机器人针对条件语言建模任务进行微调，并可使用对话历史和部分对话状态等上下文信息（见 §II-B）。相比之下，在训练数据抽取期间，任务机器人缺少此类上下文。当通过选择概率最高的词来生成内容时，任务机器人会遇到“状态结束符”或“[EOS]”词元[2]具有最高似然度的情况，导致对话状态内容生成过早终止。为促进对话状态生成，我们探索了各种采样策略，以鼓励模型生成更多样化的词元。

- **温度采样** [18] 从扁平化的下一词概率分布中采样词元。具体而言，原始的 softmax 概率分布 $\text{softmax}(z)$ 被替换为 $\text{softmax}(z/t)$，其中 t 是温度，而 $z = f_\theta(x_1, \ldots, x_{i-1})$ 表示语言模型的输出，即词表上的预测概率分布。较高的温度 t 会使模型输出的置信度降低，但更多样化。
- **衰减温度采样** [8] 通过在初始阶段采用较高温度，并在每个生成步骤中逐渐降低温度，进一步改进了温度采样。在早期阶段，这种方法使模型能够探索多种前缀，从而增强生成结果的多样性。随着生成的推进，温度逐渐降低，模型更有可能专注于生成置信度较高、随机性较低的路径，从而使输出更加有效。
- **束搜索** [19]会在每个时间步评估排名前B 的假设，并最终选出整个序列总体概率最高的假设。这样做的优点是能够识别出概率较高的

[2]“状态结束符”或“[EOS]”词元是一种特殊词元，用于表示对话状态的结束。

序列，即使它们以概率较低的初始词元开头，而这些序列原本会被贪心搜索忽略。

- **分组束搜索** [20]将束分组，并引入组内多样性项以提高多样性。
- **集束采样** [21] 将 top-B 替换为多项式采样以提高多样性。我们还使用温度采样进一步增强这种多样性。

2）成员推断：完成上一步后，我们可以生成大量对话状态候选项；然而，其中许多可能根本不在原始训练数据集中。因此，我们需要执行成员推断，以确定每个候选项是否确实属于训练数据集。成员推断攻击利用了这样一种观察：模型倾向于对训练过程中见过的样本赋予更高的置信度 [8]。具体而言，我们根据以往研究中广泛使用的两项指标，对提取出的对话状态候选项进行排序；每个候选项都被视为一个词元序列 [8], [10]。

- **困惑度（PPL）** [22]用于衡量语言模型生成文本的可预测程度。困惑度越低，表示模型越不觉得该序列令人意外，也就是说，它很可能为序列中的每个词元赋予更高的平均概率。形式上，对于序列$\mathbf{x} = (x_1, \ldots, x_n)$和一个大语言模型 f_θ，我们有

$$\mathrm{PPL}(\mathbf{x}) \triangleq \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log f_\theta(x_i|x_1,\ldots,x_{i-1})\right).$$

- **PPL-zlib** 通过以所生成序列的信息熵对一次生成的困惑度进行缩放来加以调整，其中熵通过该序列的 zlib 压缩进行量化 [23]。该方法缓解了 PPL 偏向于偏好信息量较低的平凡生成次数的偏差，例如含有重复子序列的文本，如“123123123”。

3) 稻草人方法的局限性：稻草人方法存在若干缺陷。对于候选对话状态生成，现有方法存在探测常见且多样性较低的对话状态的问题。例如，我们观察到“餐厅(名称=yippee noodle bar)”在我们的实验中重复了数百次。对于成员推断，现有度量容易产生假阳性，不准确地为未被记忆的样本赋予较高似然。该问题在长序列以及重复子串中尤为明显，例如“餐厅(名称=yippee noodle bar) 景点(名称=yippee noodle bar)”。

频繁生成琐碎或泛化内容，可归因于面向任务的对话的固有特征。在实践中，用户往往以常见请求开启对话，例如“请帮我在城里预订一家餐厅”。这些初始交互通常会使对话扩展，并根据用户不断变化的偏好纳入多种话题。因此，早期交互更容易出现重复。结果，使用传统方法提取的训练数据往往集中于这些早期轮次，而这些轮次通常较为表面，缺乏实质性细节。此外，由于对话具有连贯性，常见的情况是同一值会在多个轮次中重复出现，例如出租车的到达地址与餐厅地址相同。在不以对话历史为条件的情况下，任务机器人很容易采用这种值复制机制，并在逻辑上并不适用的地方错误地复制值。

B. 模式引导的定向对话状态生成

稻草人方法的候选对话状态生成存在两个主要问题：1）生成的对话状态多样性有限；2）生成的对话状态不合法，要么包含超出范围的领域、槽位和值，要么不符合合法格式。为解决这些问题，我们首先提出一种定向提取设置，利用部分对话状态作为上下文。然后，我们介绍一种模式引导采样方法，通过探查任务机器人的服务范围，在采样过程中限制词语选择范围，从而从任务机器人中提取出更相关且合法的对话状态。

1）定向对话状态提取：在没有任何条件上下文的情况下采样对话状态，往往只会生成训练数据集中出现过的通用文本模式。例如，由于训练数据中多次出现“name=Pizza Hut”，生成的所有餐厅名称都是“Pizza Hut”。为解决这个问题，我们探索将部分对话状态用作提示词，这些状态可以来自稻草人攻击，也可以由攻击者编写，以生成更详细、更具针对性的样本。例如，如果攻击者知道某人的部分旅行计划，就可以提示模型透露其余部分。

在这项工作中，我们使用任务机器人的 origintraining 数据集中的示例进行定量概念验证分析。为了避免我们的方法被轻易滥用，我们不会模拟真实攻击者来生成触发线索。具体而言，我们将训练数据集中的部分对话状态用作长度受控的条件上下文，例如“信念状态：餐厅（名称=pizza hut”，作为长度为一的前缀，即一个领域-槽位-值三元组。

2) 模式引导采样：在稻草人方法中，生成的对话状态有很大一部分要么超出服务范围，要么格式杂乱。为了解决这些问题，我们借鉴了约束解码方法 [24]，提出首先提取任务机器人的服务范围，然后用它来剪枝探索范围。具体而言，我们先从任务机器人中提取对话模式，其中包括定义机器人服务范围的领域和槽位[3] 。随后，在提取对话状态的过程中，采样领域和槽位时，我们严格将词元选择限制在提取出的模式范围内。这样可以确保生成的候选对话状态既与服务范围相关，也符合所需的结构化格式。

为了从任务机器人中抽取对话模式，我们首先收集各种可能含有错误的对话状态，然后使用正则表达式从这些状态中提取领域和槽位。在此过程中，优先保证召回率至关重要，因为遗漏任何元素都意味着在状态提取攻击期间不会生成相应的领域或槽位。为此，

[3]我们不会预先定义值的范围（例如，“12345”是槽位“电话”的一个值），因为它们高度依赖于用户话语。

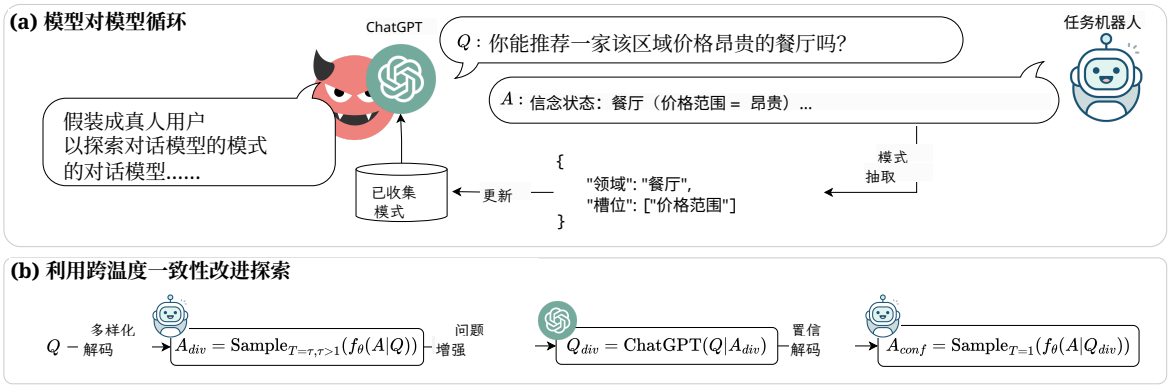


图3. 使用 ChatGPT 进行模式提取。在**(a) 模型对模型循环**中，基于已收集的模式（首次循环时为空），我们提示 ChatGPT 模拟用户向任务机器人提问，以发现更多领域和槽位。为提高精确率并加强探索，我们进一步提出使用**(b) 跨温度一致性**来改进基本循环。与(a)中一样，在解码任务机器人的回答时，我们使用高温度设置生成多样的信念状态，其中可能包含错误的领域和槽位。随后，我们通过验证过程筛除这些错误：提示 ChatGPT 针对抽取出的（可能不正确的）状态提出相应问题，再让任务机器人在低温度设置下解码回答，生成置信度高且不包含非法领域和槽位的答案。

我们首先使用高温度进行温度采样，以最大限度地探索并从提取出的状态中收集模式。然而，实验结果表明，尽管该方法在提高召回率方面表现出色，但由于会生成包含无关领域和槽位的损坏输出，其精确率往往过低。此外，要实现高召回率需要生成数千次任务机器人输出，计算效率低下。

为克服这一局限，我们提出一种启发式模型对模型方法，将 ChatGPT 伪装成用户，与任务机器人交互并探查其服务范围。以这些 ChatGPT 提出的问题为条件，可确保任务机器人给出置信度高的回复。该方法不仅能在保持高召回率的同时确保更高的精确率，还能减少所需的任务机器人生成次数。具体而言，我们通过将先前已收集的领域和槽位纳入提示词，改进这一迭代式提问阶段。如图3所示，该过程从 ChatGPT 根据已探索的模式（若有）提出问题开始。我们设计了一个问题生成协议，并使用表I中的指令提示大语言模型。

尽管这种方法有效，但仅使用 ChatGPT 进行探索效率低下，因为它缺乏对任务机器人的先验知识，而且很容易提出超出范围的问题。为了解决这一问题，我们提出了一种变体，通过跨温度一致性增强探索能力。该方法首先使用高温采样获取多样化的对话状态，然后将其转换回问题（指令很简单："请尝试根据以下对话状态生成问题："），再通过低温采样探查置信度较高的输出。将在高温和低温采样中未能一致出现的领域和槽位过滤掉。这样可以在不损失精确率的情况下进一步提升探索能力。

提取对话模式后，我们使用已识别的模式来约束特定领域和槽位的采样。具体而言，我们采用硬约束采样方法，将采样所使用的词表限制为仅包含所提取的领域和槽位。给定一个由 C 表示的词语或短语候选集，我们首先计算

表I 用于生成模式探测问题的大语言模型提示

按照以下步骤，根据现有模式为对话系统生成问题：审查现有模式：仔细检查所提供的已验证并确认的模式。确保你充分理解其中列出的每个领域、槽位和值。构造模拟的面向任务的查询：基于每个现有领域，创建一系列与已知范围略有偏离的问题。这些查询应有助于探查这些领域中潜在的新槽位或值。例如，如果有一个天气领域和一个城市槽位，你可以询问："明天适合户外活动的天气怎么样？"如果有一个餐厅领域，你可以说："你能推荐东京附近的纯素餐厅吗？"实施跨领域探查：结合已知的槽位和值，尝试构造跨领域查询。例如，你可以询问："伦敦下雨预报期间有音乐活动吗？"或者："我能从北京烤鸭乘出租车前往最近的电影院吗？"发起随机的面向任务的查询：设计并提出一系列随机的面向任务的问题，旨在发现当前模式未涵盖的潜在领域、槽位或值。生成的问题应与现有模式完全**无关**，并基于广泛的领域。已验证并确认的模式：{模式： {领域：餐厅，槽位：[价格_范围，名称]}}你的问题：

给定前缀时每个候选项的概率，从而形成一个概率分布。我们从该分布中采样一个候选项，以确定领域和槽位。形式化表示为$w = \text{Sample}(P(w'|\text{前缀})\ w' \in C)$，其中 w 是从 C中选出的词或短语。

C. 去偏成员推断

稻草人方法中的成员推断方法存在较高的假阳性率，尤其是在处理重复子串和长序列时。这一问题主要归因于训练与验证之间的不一致。通常，任务机器人会根据给定的对话历史学习生成对话状态。然而，困惑度及其变体（例如 zlib 缩放 [8], [10]）等度量将对话状态视为没有条件上下文的独立序列。更合理的方法则会考虑

序列的条件性质。具体而言，如果在给定对话历史的情况下某个序列是熟悉的，那么它更有可能是训练样本。

鉴于无法获取训练对话历史，我们提出一种条件困惑度（C-PPL）的近似度量，该度量仅关注序列后缀的概率，而非整个序列，即：

$$\text{C-PPL}(\mathbf{x}) \triangleq \exp\left(-\frac{1}{L_s}\sum_{i=L_p+1}^{L_p+L_s}\log f_\theta(x_i|x_1,\ldots,x_{i-1})\right)$$

其中，$\mathbf{x}=(x_1,\ldots,x_{L_p},x_{L_p+1},\ldots,x_{L_p+L_s})$，$L_p$ 表示前缀的长度，L_s 表示后缀。对于非定向设置，该前缀就是“信念状态：”。

此外，为了应对假阳性问题——后缀而非整个序列对任务机器人而言过于熟悉——我们提出了 C-PPL 的去偏版本，称为去偏条件困惑度（DC-PPL），通过后缀本身的困惑度对 C-PPL 进行缩放，即：

$$\begin{aligned}\text{DC-PPL}(\mathbf{x}) &\triangleq \frac{\log(\text{C-PPL}(\mathbf{x}))}{\log(\text{PPL}(\mathbf{x}_{\text{suffix}}))}\\ &= \frac{\sum_{i=L_p+1}^{L_p+L_s}\log f_\theta(x_i|x_1,\ldots,x_{i-1})}{\sum_{i=L_p+1}^{L_p+L_s}\log f_\theta(x_i|x_{L_p+1},\ldots,x_{i-1})}\end{aligned}$$

其中，$\text{PPL}(\mathbf{x}_{\text{suffix}})$ 按照常规方式计算，即计算后缀作为独立序列时的困惑度。我们对困惑度进行对数变换，使其分布更接近正态分布。

V. 评估

本节评估不同训练数据提取方法在一个大规模、公开可用的面向任务的对话数据集上的表现。

A. 数据集和评估指标

1) 数据集：我们在 MultiWOZ 数据集上评估不同的训练数据提取方法 [17], 该数据集是目前使用最广泛的黄金标准大规模面向任务的对话数据集。MultiWOZ 数据集包含 8,438 段对话，共计 113,556 轮。平均而言，每段对话有 13.46 轮，每轮包含 13.13 个词元。总体而言，该数据集涵盖 7 个领域、24 个槽位和 4,510 个不同的值。

2) 模型微调：我们使用 MultiWOZ 数据集对 Llama2 (7B) 模型进行微调，微调后的 Llama2 (7B) 将作为我们要攻击的任务机器人。然而，由于数据稀疏性，直接使用 MultiWOZ 数据集微调大语言模型很容易导致过拟合，因为模型会记忆数据，而不是学习如何泛化，因此容易遭受数据提取攻击。为了更有说服力地评估攻击的有效性，我们使用低秩适配 (LoRA) [25],对目标模型进行微调。LoRA 是一种参数高效的大语言模型微调技术，它通过限制可训练参数来减少过拟合，同时确保性能与全参数微调相当。

经过微调的任务机器人如今更能抵御攻击，对数据提取攻击构成了更大的挑战，从而使我们能够更好地评估攻击策略的效率。由于我们专注于抽取训练对话状态，我们评估该任务机器人在对话状态跟踪中的表现，即评估其根据对话历史预测对话状态的能力。微调后的任务机器人在精确率、召回率和F1分数上分别达到91.43%、71.52%和80.26%，展现了其有效性。

3）评估指标：我们评估被记忆示例的数量，以及这些示例在整体提取的对话状态和其中包含的具体值（例如电话号码）中的精确率。请注意，即使整个对话状态并不完全匹配，仍可能从训练数据集中准确提取出单个值。指标如下。

- **提取的对话状态数量（#States）**。攻击所提取的对话状态总数。每个对话状态都表示为一组领域–槽位–值三元组，例如 Restaurant{名称 =pizzahut，电话 = 12345}。如果一个对话状态在三元组分解后与训练集中的任一真实状态**完全匹配**或是其**子集**，则视为成功提取。

- **对话状态提取精确率（SP）**。正确出现在训练数据集中的提取对话状态所占比例：

$$\text{SP} = \frac{N_{\text{correct states}}}{N_{\text{extracted states}}}.$$

该指标在状态层面评估准确性，并对相同状态去重。

- **每个提取状态的领域–槽位–值三元组平均数量（#Triplets）**。每个成功提取的状态中包含的领域–槽位–值三元组（例如，餐厅-名称-pizzahut）的平均数量：

$$\#\text{Triplets} = \frac{\sum_{i=1}^{N_{\text{correct states}}}|T_i|}{N_{\text{correct states}}},$$

其中，T_i 表示第 i个正确对话状态中的三元组集合。该指标反映了每个重构的对话状态的信息丰富度。

- **提取的槽位值数量（#Values）**。在所有提取的对话状态中出现的唯一槽位值总数，经过跨状态去重。例如，如果“pizzahut”出现在多个状态中，则只计数一次。

- **槽位值提取精确率（VP）**。提取的槽位值存在于训练数据集中的比例：

$$\text{VP} = \frac{N_{\text{correct values}}}{N_{\text{extracted values}}}.$$

该指标评估值级恢复的保真度，与状态结构无关。

B. 稻草人方法的评估结果 *hod*

1) 设置：对于每种对话状态生成方法，我们使用任务机器人生成 10，000 个样本。随后，我们对抽取结果去重，并使用成员推断指标对后缀进行排名，仅保留排名前 100 的样本以供进一步

表 II 无目标对话状态提取攻击的成员推断结果。状态少于 100 的方法被排除，因为排名后仅保留排名前 100 的提取结果。

提取方法	状态精确率 (%)			#三元组			值精确率 (%)		
	None	PPL%	PPL-zlib	None	PPL%	PPL-zlib	None	PPL%	PPL-zlib
普通采样	**14.37**	**26.00**	**26.00**	1.60	2.27	2.00	**15.00**	60.52	54.9
温度采样	4.59	19.00	11.00	1.29	1.53	1.45	5.76	38.56	33.00
集束采样	11.95	23.00	21.00	**1.95**	**2.52**	**2.71**	10.73	**66.97**	**68.1**

表 III 使用稻草人方法进行非目标提取

方法	#状态	SP%	#三元组	#值	VP%
普通采样	336	14.37	1.60	231	15.00
温度采样	207	4.59	1.29	234	5.76
衰减温度	75	3.32	1.51	223	17.13
束搜索	5	**100**	1.2	4	**100**
分组束搜索	15	65.22	1.47	16	64.00
集束采样	**715**	11.95	**1.95**	**315**	10.73

分析。我们进行了广泛的超参数搜索，以确定 §IV-A1中介绍的每种生成方法的最佳设置。所选配置如下：1) 温度采样：温度设为 2；2) 衰减温度采样：以较高的温度 5 启动流程，以促进初始输出的多样性，然后在后续每一步将温度降低 2 ，直至稳定在 1；3) 束搜索：采用较大的集束大小 50 ，并保留所有束，以提升输出多样性并最大限度地提高训练数据覆盖率；4) 分组束搜索：组数设为与束数量相同，并施加 1 的多样性惩罚；5) 集束采样：集束大小设为5，温度设为 2。

2) 对话状态提取性能：我们发现了各种训练对话状态，即使不以相应的对话历史为条件也能抽取出来。例如，“餐厅（价格_范围=昂贵）列车（出发地=剑桥，目的地=伦敦国王十字车站，人数=2）”描述了一段对话：有人计划先在剑桥享用一顿昂贵的餐食，然后与另一人乘坐列车前往伦敦国王十字车站。我们还发现了单领域预订的示例，例如“列车（日期=星期五，人数=2）”。这些示例表明，即使在无目标攻击场景中，也有可能从任务机器人中抽取隐私信息。

接下来，我们在表III中给出定量结果。我们发现，稻草人方法的所有变体正确抽取的对话状态数量都明显偏低，在所有生成次数中占比不到 3.36% 。尽管束采样方法看起来可能生成了更多输出，但它实际生成的输出数量是基本采样方法的集束大小倍。因此，正确抽取的对话状态的实际占比甚至更低。此外，在我们的上下文中，温度采样及其衰减变体均未奏效，这与先前的研究相悖 [8], [10]。我们认为，这种差异源于对话状态高度结构化的特性，而这些方法会破坏这种结构，导致输出无效。我们还观察到，束搜索和组束搜索的精确率都很高，这归因于它们的确定性，但由于计算约束限制了集束大小，它们未能生成多样化的输出：尽管进行了 10，000 次迭代，输出多样性仍受限于波束数量。束采样通过结合束搜索与温度采样，能够更广泛地探索对话状态，但发现的对话状态数量仍然有限。此外，正确对话状态提取的平均长度（#三元组）明显较短，通常少于 2 个，最多为 4 个。这些内容往往只是关于餐厅或列车时刻表的简单细节，这些信息很可能出现在对话开头，并在训练数据集中多次出现，因此不太敏感。

3) 成员推断性能：如表II所示，结果凸显了成员推断在提高抽取精确率和延长所抽取状态方面的有效性。然而，尽管有所改进，整体隐私风险仍处于中等水平。尽管个别值的风险相对较高，最高精确率达到67%，事件级状态引发的担忧则要小得多，其最高精确率为26%。

4) 稻草人方法的局限性：我们的实验结果揭示了稻草人方法的若干局限性。首先，该方法容易探测到常见且多样性较低的对话状态。例如，“餐厅(名称=yippee noodle bar)”会重复出现数百次。此外，成员推断度量容易产生假阳性，错误地将较高的可能性赋予未被记忆的样例。对于长序列和重复子串，这一问题尤为明显，例如：“餐厅(名称=yippee noodle bar) 景点(名称=yippee noodle bar)”。在对话中，由于对话连贯，某个值多次出现很常见，例如出租车的到达地址与餐厅地址相同。然而，若不以对话历史为条件，任务机器人可能会错误地沿用这种模式，在逻辑上不适用的地方重复使用这些值。

C. 对话模式提取结果

由于我们提出的目标模式引导式对话状态提取方法的性能取决于提取出的对话模式，因此我们先介绍模式提取结果，再讨论对话状态提取结果。

1) 设置：本节评估我们提出的模型对抗模型对话模式探测策略（见 §IV-B2）。我们评估了五项主要指标。在准确性分析中，我们分别考察领域和槽位的精确率与召回率。在成本分析中，我们考察 ChatGPT 为实现覆盖率所需提出的问题数量。

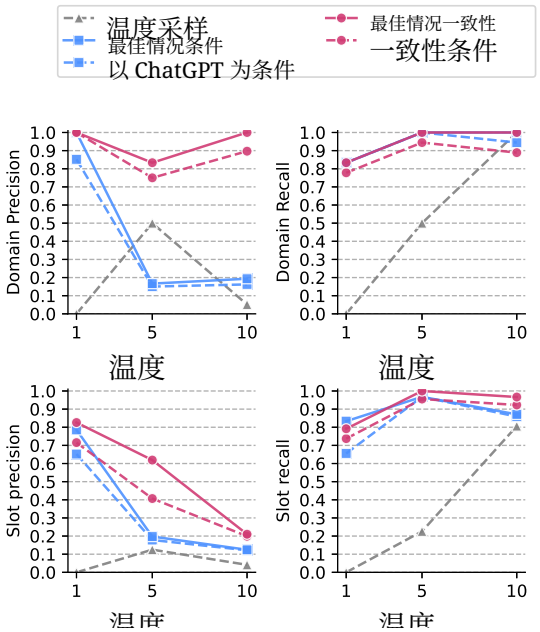


图 4. 不同温度下的模式提取性能。我们报告每种方法三次运行的平均值以及最佳结果。

2) 抽取性能：如图 4所示，采用所提出的模型对抗模型方法时，温度为 5 的最佳情况下，对话模式的覆盖率达到完整的 100% ，领域精确率超过82% ，槽位精确率超过60%。与温度采样相比，我们的方法显著提高了精确率，同时也提升了召回率。事实证明，跨温度一致性能够进一步提高召回率，且精确率损失极小。

我们进一步深入错误分析，发现主要的假阳性槽位包括：1）槽位的共指或替代表达，例如“people”与“number of people”；2）来自其他领域的槽位及其变体。然而，我们的下一项实验将表明，这些问题在增强对话状态提取时不会导致严重的性能下降。

3) 成本分析：我们使用收敛所需问题数 （# Questions）来评估计算成本，该指标计算为获得最优模式提取性能而必须查询任务机器人的最少次数。对攻击者而言，更少的交互意味着更高的效率。如图5所示，所提出的模型对抗模型方法需要约3 至 5 倍更少的交互（与传统温度采样相比）。此外，采用跨温度一致性的变体进一步提高了领域提取和槽位提取的效率。另外，我们发现在更高温度下成本可能会增加。原因在于，多样化采样会通过连续发现逐步揭示难以检测的边缘槽位以及槽位的共指或替代表达，从而需要更多交互才能实现收敛。

D. 模式引导的定向对话状态提取结果

*1)*设置： 我们将训练数据集中的部分对话状态作为条件输入任务机器人，以进行概念验证，开展定量分析，

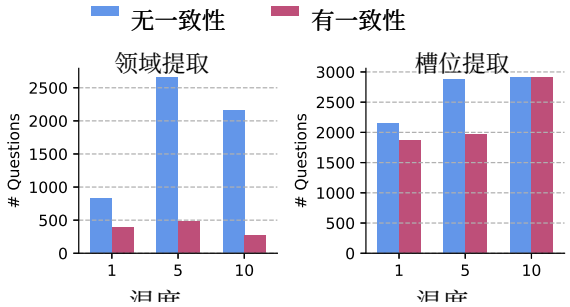


图 5. ChatGPT 在模式提取中达到性能上界所需的最少问题数。对于温度采样，生成 10，000 个样本以达到报告的性能。

研究定向对话状态提取（参见 §IV-B1）。具体而言，我们控制部分信念状态中的领域-槽-值三元组数量（前缀长度），以抽取其余后缀。这些输入部分对话状态通过截断训练数据集中超出所控长度的对话状态获得。例如，生成的前缀“信念状态：餐厅（名称=Pizza Hut”用于促使任务机器人生成后缀。请注意，前缀不包含训练对话历史，仅包含部分对话状态。由于不同对话片段可能具有相同的对话状态——例如，它们都涉及餐厅预订——我们会对前缀去重以避免冗余，从而得到若干唯一前缀。抽取完成后，我们会进一步去重，再计算评估指标。

请参阅 §IV-A1和 §IV-B2，了解基线方法和我们提出的模式引导后缀解码方法的详细说明。请注意，我们在此使用的模式是通过模型对抗模型方法结合跨温度一致性获得的。

2) 定向对话状态提取结果：如图 6所示，在以部分对话状态作为种子时，我们的方法能够从任务机器人中抽取数千个训练对话状态。与所有基线相比，所提出的模式引导采样方法表现出显著优势，且性能仅略低于黄金模式上界，这证明了通过限制采样空间来探测任务机器人的有效性。令人惊讶的是，就有效状态数量和状态精确率而言，贪心搜索仍然是一个强有力的基线，其表现优于束搜索和温度衰减。原因在于，在给定条件上下文时，所有方法生成的输出数量相同；然而，增强多样性的技术会产生更多样的值（这体现在 #Values 和值精确率结果中）。将这些多样化的输出组合成状态时，它们更有可能无效。相比之下，贪心搜索始终生成一组固定的值，这些值与训练数据高度吻合，因此最终得到的有效状态总体上更多。

如图左侧的灰色柱所示，6，部分对话状态的去重前缀数量存在差异，因此不同前缀长度对应的抽取总数也有所不同。因此，我们进一步评估了各方法的精确率。如图6所示，模式引导的

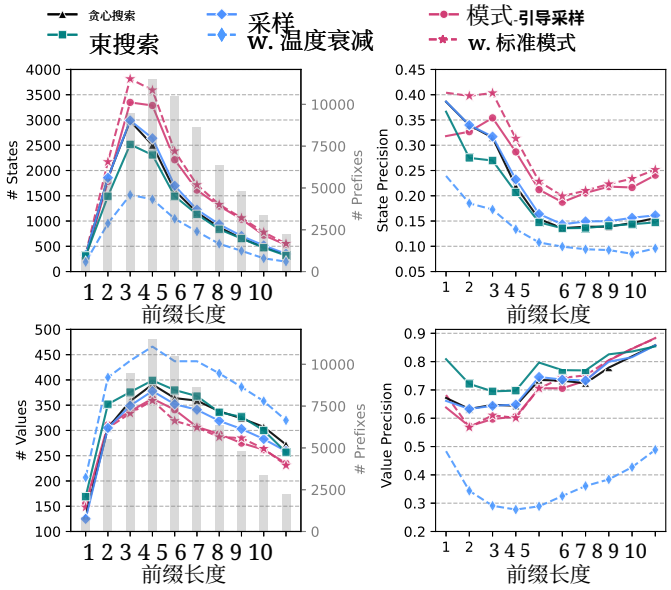


图 6. 定向状态抽取结果。我们报告成功抽取的训练对话状态总数、这些状态中的记忆值，以及抽取精确率。灰色柱表示去重前缀的数量。

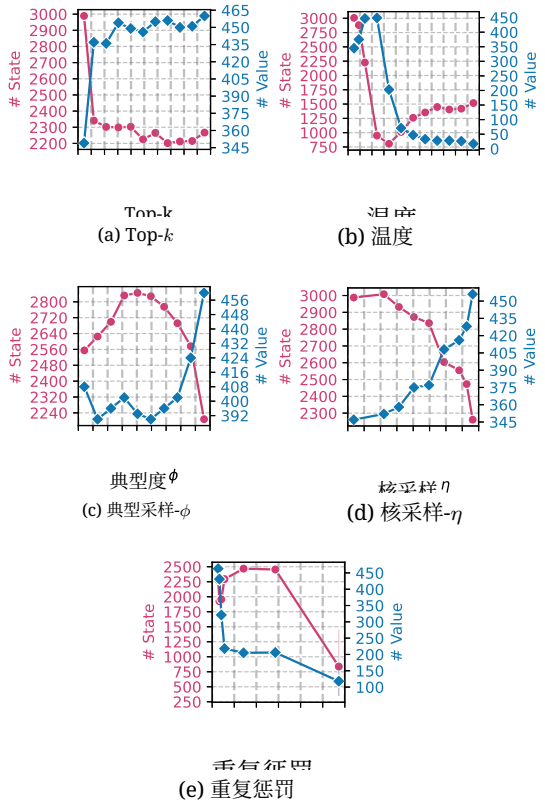


图 7. 在不同采样配置下抽取的训练对话状态和值的数量。结果基于前缀长度为 3 的设置报告，并采用最大样本量，以确保结果具有统计显著性。

采样仍优于其他方法。有趣的是，我们观察到，随着前缀长度增加，精确率反而下降。这一发现与先前的研究 [9]，相矛盾；先前研究认为，更多上下文会带来更高的精确率。我们将在 §V-D4中进一步讨论这一现象。

3）采样配置的影响分析：本节考察调整采样配置（即超参数）如何影响对话状态提取结果。结果见图7。

Top-k [26] 用于将词表限制在概率最高的 Top-k个词元，从而使生成集中于最可能的结果。如图7(a)所示，较大的 k会增加值的多样性，但会减少有效对话状态的数量。

温度（§IV-A1）用于调整词元的概率分布。从图7(b)可以看出，较低的温度会产生更多有效但多样性较低的对话状态。温度相对较高时，我们观察到提取出的有效状态更多，但值更少。这可能是因为，尽管大多数情况无效，但某些槽位仍能成功提取，而置信度较高且被牢牢记住的值，即使在较高温度下也会被频繁采样。

典型度ϕ [27] 旨在建议选择一个词元，使其信息量接近给定先前上下文时的预期信息量，从而增强句子一致性并减少退化重复。如图7(c)所示，性能会随不同 ϕ 值的变化呈非单调变化，这与 [10]一致。

核采样-η [28]设置为确保输出从最有可能的词元中进行采样，且这些词元构成的最小集合的总概率等于或大于 η。在图7(d)中，我们观察到，较低的 η 值能够提升状态提取效果，这与上文讨论的 Top-k方法类似。

重复惩罚 [29] 设置为根据每个词元是否重复先前词元，调整其生成概率。重复词元的 logit 在经过 soft max 层处理前，会先除以惩罚值。图7(e)所示的结果表明，适当的重复惩罚对对话状态提取有积极影响。减少重复有助于防止任务机器人错误地将前缀中的值直接复制到后缀中，例如“餐厅(名称=Pizza Hut,区域=Pizza Hut)”。

4) 原因分析：为什么更长的上下文会导致状态抽取精确率降低：在 §V-D2中，我们发现，随着输入的部分对话状态前缀长度增加，状态抽取精确率会下降。这一趋势会持续下去，直到前缀达到一定长度（在我们的分析中大于 6），此时精确率才开始回升。这一发现与先前的研究相矛盾 [9], 该研究指出，增加上下文词元数量能够提升抽取记忆文本的能力。为了解其中原因，我们探究了微调后的任务机器人的独特特性，重点考察：1）对话数据在训练数据集中的重复情况；2）对话的一对多特性，即单个对话历史可能对应多个有效回复。

关于训练对话数据的重复： 使用面向任务的对话数据训练任务机器人时，对话会被拆分为若干轮，每轮包含对话历史、对话状态等（见图 1）。通过这种拆分，在对话早期出现的对话状态会被重复

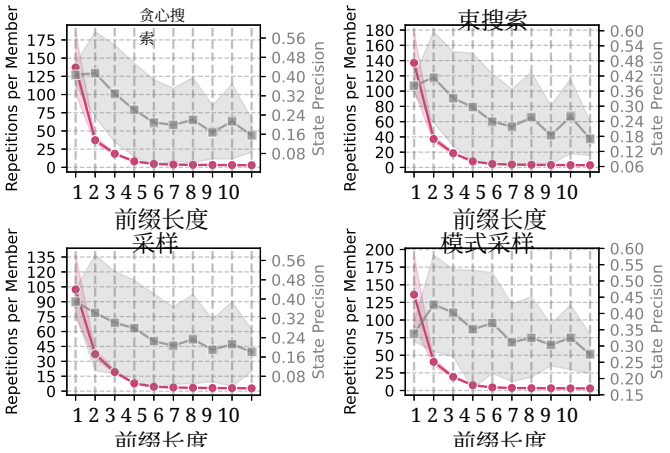


图 8. 不同前缀长度下每个成员的重复次数：此指标衡量成功抽取的训练对话状态在训练数据集中重复出现的次数。结果以 99% 置信区间报告。此外，我们还展示了不同提取方法的状态精确率（灰线）。

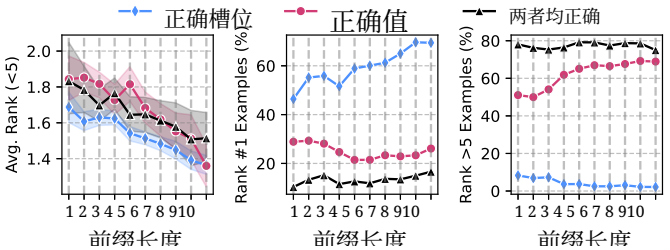


图 9. 基于任务机器人的解码概率对标准答案进行排名。我们报告槽位、值以及组合对的结果。**平均排名 (<5)** 指排名位于前 5 名以内的示例的平均排名。排名越低，表示任务机器人对正确答案的解码置信度越高。**排名第 1** 和**排名 >5** 分别表示排名第 1 和未进入前 5 名的真实示例所占的百分比。排名第 1 的比例越高、排名 >5 的比例越低，表明任务机器人在解码正确答案时越有把握。

更频繁地出现在训练数据中。例如，在一段包含 10 轮的对话中，第一轮的对话状态（例如“价格=便宜”）也会出现在后续9 轮中，除非用户请求修改价格范围。先前的研究表明，重复可以强化记忆 [8], [9]。因此，尽管较长的状态包含更多上下文，却可能更难提取。

为验证这一点，我们统计了所抽取的真正例在训练数据集中重复出现的次数。如图8所示，每个成员的重复次数会随着前缀长度增加而按对数规律下降。当提供的上下文不足时，重复次数的急剧下降可能会抵消上下文增加带来的益处，使状态提取更具挑战性，并导致状态提取精度降低。然而，当上下文达到足够长度后，优势便转向更长的上下文，从而使精确率逐步提升。

关于面向任务的对话的一对多特性：图中的状态提取结果6 表明，状态精确率下降的同时，值精确率却在上升。这说明，尽管我们可以抽取训练数据集中出现的值，但这些值并不总是构成真正的蕴涵当前前缀的结果。我们推测，这是由面向任务的对话固有的一对多特性导致的：相同的上下文可能使不同用户给出不同但有效的回复，从而使同一前缀对应多个可能的值。为进一步探究这一点，我们分析了在不同前缀长度下，金标准槽位、金标准值及其组合按照解码概率的排名情况。如图9所示，平均排名会随着前缀长度增加而下降。然而，金标准槽位排名第一的实例显著增加，从40% 增至70%，表明其对上下文有很强的依赖性。相比之下，金标准值排名第一的比例仍然较低，仅在20% 至30%之间波动，这表明其对上下文的依赖性较弱。尽管金标准值很少排名第一，但它们越来越多地出现在前五位，证实了对话上下文具有一对多特性，即单个对话上下文可能产生多种不同的潜在回复。

E. 去偏成员推断结果

1) 设置：我们根据多种成员推断度量对抽取结果进行排序，并保留排名前100 个示例用于评估。有关成员推断方法的详细介绍，请参阅 §IV-A2和 §IV-C 。

2) 定向对话状态提取结果：给定使用各种方法提取的对话状态，我们将成员推断度量应用于这些对话状态，并报告最终获得的对话状态和槽值的精确率。如图10所示，所提出的条件困惑度和去偏条件困惑度均能小幅提升对话状态精确率，并在槽值精确率方面表现领先。值得注意的是，去偏条件困惑度表现最佳，其最佳情况下的对话状态精确率超过70%，槽值精确率达到100% 。我们还注意到，PPL 和 PPL-zlib 在对话状态精确率方面的表现逊于随机排序，但在值精确率方面表现良好。这种差异是因为 PPL 和 PPL-zlib 会评估完整的前缀-后缀序列。然而，在定向提取中，前缀直接来自训练数据，因此对话模型天然熟悉这些前缀。因此，即使搭配了错误的后缀，这些熟悉的前缀仍往往会获得较高排名，从而造成有损性能的偏差。

为进一步研究不同成员推断度量区分抽取文本的效果，我们报告了以下对象的分数分布：真实的成员和非成员。图11 显示，对于 PPL 和 PPL-zlib，两者的分布完全重叠，因此无法区分成员和非成员。相比之下，尽管条件困惑度和去偏条件困惑度存在一定重叠，但仍可在较低得分区间识别出成员，证明了该方法的有效性。

3）无目标对话状态提取的结果：定向提取使用训练数据集中的部分黄金前缀，这可能会使 PPL 和 PPL-zlib 处于不利地位。因此，我们还报告了无目标抽取攻击设置下的结果，其中不包含部分对话状态。在这种设置下，条件 PPL 及其变体的上下文仅为“信念状态：”，如 §IV-A1 所述。

表 IV 不同成员推断度量下无目标对话状态提取的状态与值精确率。“D”和“C”分别表示“去偏”和“条件”。

提取方法	状态精确率（%）					值精确率（%）				
	None	PPL	PPL-zlib	C-PPL	DC-PPL	None	PPL	PPL-zlib	C-PPL	DC-PPL
普通采样	14.37	26.00	26.00	24.00	**42.00**	15.00	60.52	54.90	55.00	**70.27**
温度采样	4.59	19.00	11.00	14.00	**23.00**	5.76	38.56	33.00	36.95	**46.45**
集束采样	11.95	23.00	21.00	29.00	**40.00**	10.73	66.97	68.10	70.19	**76.34**

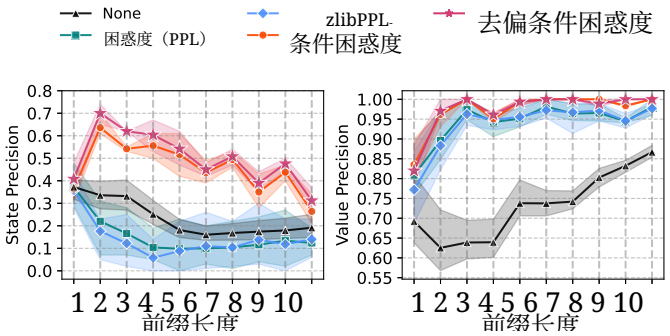


图 10. 成员推断后定向对话状态提取的对话状态和槽值精确率。我们报告了 §V-D 中详述的各种提取方法的平均结果。阴影区域表示从最佳到最差提取方法的性能范围。

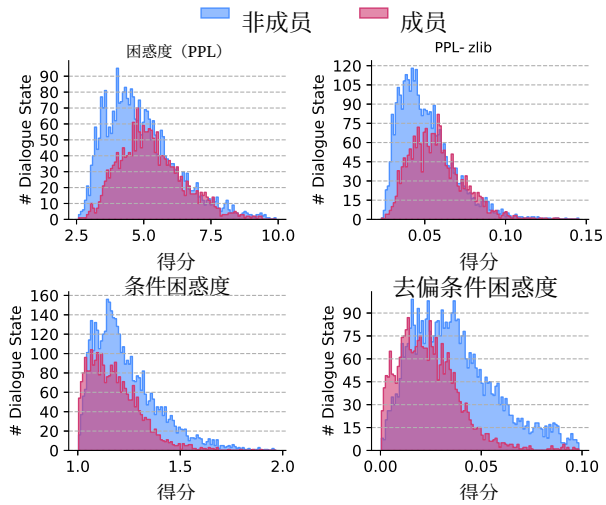


图 11. 抽取的对话状态的成员推断分数分布。我们比较了出现在训练数据集中的对话状态（成员）与未出现在其中的对话状态（非成员）。

如表 IV 所示，我们提出的去偏条件困惑度在对话状态和槽值精确率两方面，均显著优于基线方法，展现了其检测训练对话状态的有效性。然而，条件困惑度的表现并不理想，因为在没有部分对话状态的情况下，它会夸大困惑度对重复序列或较长序列固有的偏好。

F. 隐私级别分析结果

为定量评估所提取信息的隐私敏感性，我们进行了细粒度隐私级别分析。本实验旨在验证我们的核心论点：隐私风险不仅源于直接的个人身份信息（例如名称、电话号码），也源于组合个人身份信息，

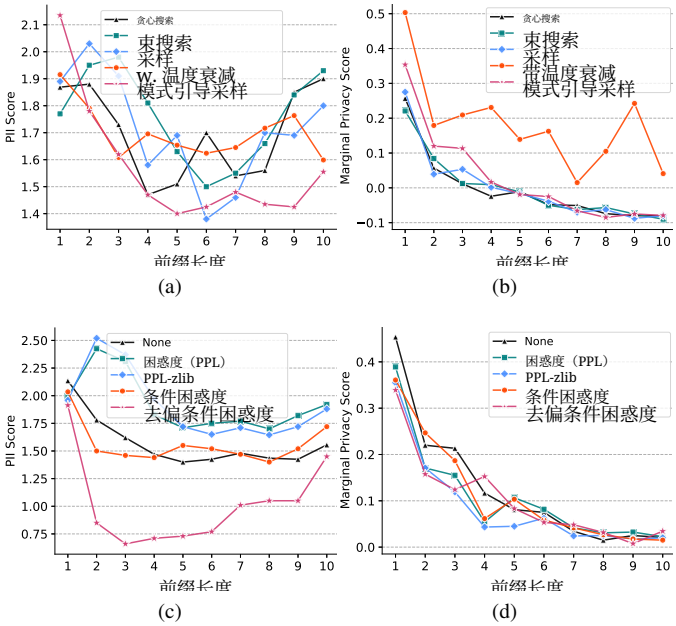


图12. 不同训练数据抽取 (a)(b) 与成员推断方法 (c) (d) 下的个人身份信息分数和边际隐私分数。

看似无害的数据点组合在一起时也可能泄露敏感信息。为此，我们采用基于提示词的 deepseek-rl-671B 模型作为专家评估者，按照严格的规范评估每个提取出的信念状态（事件级）的隐私程度。模型给出 0（无需担忧）至 5（极高）的总体隐私得分，并遵循一项关键的单调性规则：较长的信念状态（超集）所得分数不得低于其任何子集，确保新增信息不会降低感知到的敏感性。

计算了两个具体指标：

- **个人身份信息分数：**单个后缀的隐私分数，先对其构成的领域-槽-值三元组 逐一评分，再对这些分数取平均得到。该指标反映在单独看待各组成部分时，后缀固有的敏感程度。
- **边际隐私分数：** 该指标量化将抽取得到的后缀纳入现有上下文后所引入的隐私风险相对增幅。其计算方式为隐私分数增加 量与前缀原始分数之比，形式如下：

$$\text{Marginal Privacy Score} = \frac{S_{\text{prefix + suffix}} - S_{\text{prefix}}}{S_{\text{prefix}}}$$

其中， S_{prefix} 和 $S_{\text{prefix + suffix}}$ 分别是前缀和组合状态的整体隐私得分。较高的

值表示隐私敏感度放大程度更高，从而确定生成组合个人身份信息的具体情形。

如图 12所示，结果揭示了不同提取方法之间的显著差异。传统解码和成员推断方法通常会生成个人身份信息分数较高的后缀，这表明它们倾向于孤立地提取局部敏感信息。相比之下，我们的模式采样和去偏成员推理方法虽然未能取得最高的个人身份信息分数，却获得了相当甚至更高的边际隐私分数。我们还通过人工评估验证了自动评估器的可靠性，发现模型得分与人工判断高度一致，几何平均 Cohen'sκ为 0。858 (边际隐私分数）和 0。761 (个人身份信息分数）。这一发现意义重大，因为它表明，我们的方法能够更有效地从整体上下文中揭示有意义的事件级信息，具体而言，就是揭示组合个人身份信息。更重要的是，我们的方法在显著增加有效事件抽取数量、并大幅提高成员推断准确率的同时，展现出这种揭示数据组合所导致隐私风险的卓越能力，从而为潜在隐私风险确立了更保守的下限。

VI. 面向防御与隐私保护

如 §V-D4所述，我们发现有两个问题会显著影响抽取结果。1）轮次级建模导致的数据重复，即第一轮中的对话状态会出现在所有后续轮次中，从而增强记忆。2）对话具有一对多的特性：给定的对话历史可能对应多个有效的用户选择，因此不存在唯一确定的标准答案，这会降低记忆。鉴于这些特点，我们提出两种潜在的隐私增强方法：

1）对话级建模：为减少轮次级建模造成的数据重复，我们可以在对话级别对对话进行建模，每个训练实例都输入完整对话。具体而言，给定一个由多轮组成的完整对话，$H_t = \{U_0, R_0, \ldots, U_{t-1}, R_{t-1}, U_t\}$，其中包含用户话语和先前的系统响应。当前的训练方法采用轮次级数据，表示为$\{U_0, R_0, \ldots, R_{t-1}, U_t, S_t, D_t, A_t, R_t\}$，其中 t 的取值范围为 1 至 T。我们提出将其转换为对话级模型输入：$\{U_0, S'_0, D_0, A_0, R_0, \ldots, U_T, S'_T, D_T, A_T, R_T\}$。这里，$S'$ 表示当前轮次的对话状态，而非完整的对话历史，从而避免先前轮次的对话状态重复出现。以这种方式进行训练，需要对对话状态、操作和响应执行条件语言建模与损失计算，以更新模型。请注意，尽管这种方法减少了重复，但也需要更多 GPU显存和更多数据。

2）值复制机制：鉴于面向任务的对话具有一对多的特性，任务机器人的一种更注重隐私保护的变体可以采用值复制机制：槽位值不通过上下文建模重新生成，而是直接从对话历史中复制。在这种机制下，如果没有对话历史，值生成将产生空白，从而避免意外复现记忆值。

在实践中，可以通过反事实数据增强来鼓励此类行为：在训练期间，随机替换或删除对话历史及其对应信念状态中的槽位值。此过程能够丰富值分布，同时减弱模型对特定值的依赖，并保持对话的语义结构完整。由于面向任务的对话性能主要取决于准确的模式跟踪和轮次级推理，而非对具体槽位值的记忆，这种增强可以正则化值生成，而不会显著降低性能。从本质上说，该方法鼓励模型对槽位模式进行泛化，而不是记忆用户提供的特定值，从而提升鲁棒性和隐私安全。

VII. 结论

本研究探讨了大语言模型在针对面向任务的对话系统（即任务机器人）这一下游应用进行微调时，如何记忆和再生成训练数据。通过实施训练数据提取攻击，我们首先表明，即使没有任何上下文作为条件，基于大语言模型的任务机器人确实会记忆并无意中泄露对话状态的训练标签。结果表明，电话号码等值更容易被提取，最高精确率可达 67%；而完整对话状态带来的风险则明显较低，最高精确率仅为 26%。我们进一步研究了定向提取设置：以部分对话状态作为条件上下文输入任务机器人，从而提取其余内容，并针对后缀解码和成员推断作出多项改进。实证结果显示，提取效果显著提升；在最佳情况下，值的抽取精确率达到 100%，对话状态的最高精确率超过 70%。我们还研究了影响记忆的因素，并提出了若干潜在的缓解策略，以指导基于大语言模型的任务机器人的进一步发展。

致谢

本工作部分得到中国国家重点研发计划（2023YFC3306100）和国家自然科学基金委员会（62272372）的资助。

参考文献

[1] A. Algherairy 和 M. Ahmed，《对话系统综述：当前趋势与未来方向》，*Neural Computing and Applications*，第 1–27 页，2023 年。[2] J. Kulh´anek、V. Hude˘cek、T. Nekvinda 和 O. Du˘sek，《AuGPT：利用辅助任务和数据增强，基于预训练语言模型实现端到端对话》，arXiv:2102.05126，2021 年。[3] B. Peng、C. Li、J. Li、S. Shayandeh、L. Liden 和 J. Gao，《SOLOIST：利用迁移学习和机器教学大规模构建任务机器人》，计算语言学协会会刊，第 9 卷，第 807–824 页，2021 年。[4] Y. Yang、Y. Li 和 X. Quan，《UBAR：基于 GPT-2 构建完全端到端的面向任务的对话系统》，载于*AAAI*人工智能会议论文集，2021 年。[5] E. Hosseini-Asl、B. McCann、C.-S. Wu、S. Yavuz 和 R. Socher，《一种用于面向任务对话的简单语言模型》，载于神经信息处理系统进展，第 33 卷，第 20 179–20 191 页，2020 年。[6] H. Sun、J. Bao、Y. Wu 和 X. He，《MARS：通过对比学习建模上下文和状态表示，实现端到端面向任务的对话》，arXiv:2210.08917，2022 年。

[7] W. He 等人，《Galaxy：一种结合半监督学习与显式策略注入的面向任务对话生成式预训练模型》，载于AAAI人工智能会议论文集，第 36 卷，第 10 期，2022 年，第 10 749–10 757 页。[8] N. Carlini 等人，《从大语言模型中抽取训练数据》，载于第 30 届 USENIX安全研讨会论文集，2021 年，第 2633–2650 页。[9] N. Carlini、D. Ippolito 等人，《量化神经语言模型中的记忆》，arXiv:2202.07646，2022 年。[10] W. Yu、T. Pang、Q. Liu、C. Du、B. Kang、Y. Huang、M. Lin 和 S. Yan，《从语言模型中抽取训练数据的实用技巧集》，载于国际机器学习会议，2023 年，第 40 306–40 320 页。[11] N. Carlini 等人，《从扩散模型中抽取训练数据》，载于 USENIX安全研讨会论文集，2023 年，第 5253–5270 页。[12] N. Kandpal、E. Wallace 和 C. Raffel，《对训练数据去重可降低语言模型的隐私风险》，载于国际机器学习会议，2022 年，第 10 697–10 707 页。[13] M. Nasr 等人，《从（生产环境中的）语言模型中可扩展地抽取训练数据》，arXiv:2311.17035，2023 年。[14] Z. Qi、H. Zhang、E. Xing、S. Kakade 和 H. Lakkaraju，《遵循我的指令并泄露秘密：从检索增强生成系统中可扩展地提取数据》，arXiv:2402.17840，2024 年。[15] S. Zeng 等人，《探索微调语言模型中的记忆》，载于第 62 届计算语言学协会年会论文集，2024 年，第 3917–3948 页。[16] H. Touvron 等人，《Llama 2：开放基础模型与微调聊天模型》，arXiv:2307.09288，2023 年。[17] M. Eric 等人，《MultiWOZ 2.1：整合多领域对话数据集、状态修正与状态跟踪基线》，arXiv:1907.01669，2019 年。[18] G. Hinton、O. Vinyals 和 J. Dean，《蒸馏神经网络中的知识》，arXiv:1503.02531，2015 年。[19] I. Sutskever、O. Vinyals 和 Q. V. Le，《使用神经网络进行序列到序列学习》，神经信息处理系统进展，第 27 卷，2014 年。[20] A. K. Vijayakumar 等人，《多样化束搜索：从神经序列模型中解码多样化方案》，arXiv:1610.02424，2016 年。[21] D. Ippolito、R. Kriz、M. Kustikova、J. Sedoc 和 C. Callison-Burch，《条件语言模型的多种解码方法比较》，arXiv:1906.06362，2019 年。[22] F. Jelinek、R. L. Mercer、L. R. Bahl 和 J. K. Baker，《困惑度——衡量语音识别任务难度的指标》，《美国声学学会杂志》，第 62 卷，第 1 期，第 S63–S63 页，1977 年。[23] J.-l. Gailly 和 M. Adler。（检索于 2024 年）Zlib压缩库。[在线]。网址：https://www.zlib.net/ [24] X. Lu 等人，《NeuroLogic a*风格解码：利用前瞻启发式方法进行受约束文本生成》，载于北美计算语言学协会会议：人类语言技术论文集，2022 年，第 780–799 页。[25] E. J. Hu 等人，《LoRA：大语言模型的低秩适配》，arXiv:2106.09685，2021 年。[26] A. Fan、M. Lewis 和 Y. Dauphin，《分层神经故事生成》，arXiv:1805.04833，2018 年。[27] C. Meister、T. Pimentel、G. Wiher 和 R. Cotterell，《局部典型采样》，计算语言学协会会刊，第 11 卷，第 102–121 页，2023 年。[28] A. Holtzman、J. Buys、L. Du、M. Forbes 和 Y. Choi，《神经文本退化的奇特案例》，arXiv:1904.09751，2019 年。[29] B. Hilprecht、M. H¨arterich 和 D. Bernau，《针对生成模型的蒙特卡罗与重构成员推断攻击》，《隐私增强技术论文集》，2019 年。