语言模型中的实体追踪

Najoung Kim*
语言学系波士顿大学
najoung@bu.edu

Sebastian Schuster*
语言科学与技术系　萨尔大学
seschust@lst.uni-saarland.de

arXiv:2305.02363v2 [cs.CL] 8 Sep 2023

摘要

随着文本或对话的展开，持续追踪实体状态的变化是实现话语理解的关键前提。然而，针对大语言模型（LLM）追踪话语实体的能力，目前还缺乏系统性研究。在这项工作中，我们提出了一项探测任务：给定实体初始状态的英语描述以及一系列改变状态的操作，考察语言模型在多大程度上能够推断出实体的最终状态。我们首先利用这项任务探究 Flan-T5、GPT-3 和 GPT-3.5 能否追踪实体状态，发现只有在大量代码上预训练过的 GPT-3.5 模型展现出了这一能力。随后，我们通过在多个训练集/评估集划分上对 T5 进行微调，探究主要在文本上预训练的较小模型能否学会追踪实体。尽管在更复杂的划分上，模型性能有所下降，但我们发现，即使评估时使用的实体与训练集中的不同，或操作序列更长，经过微调的模型仍能进行一定程度的实体追踪。综合来看，这些结果表明，语言模型能够学会追踪实体，但仅在文本语料库上预训练并不会让这一能力显现出来。

1 引言

实现长上下文理解和生成连贯文本的一个关键前提，是能够随着话语的展开，准确地表征实体（Karttunen，1976；Groenendijk 和 Stokhof，1991；Heim，2002；Nieuwland 和 Van Berkum，2006年；Kamp 等人，2011，等等）。例如，考虑以下食谱语境中的例子：

（1）将鸡蛋、糖、面粉和泡打粉放入碗中，搅拌成轻盈的面糊。确保最终的面糊中没有面粉或糖结块。

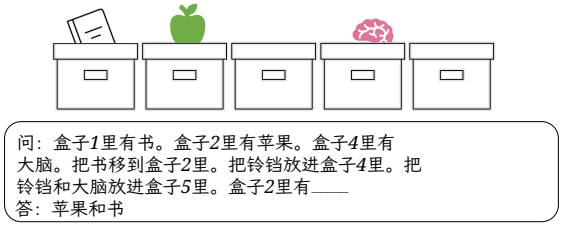


图 1：我们的实体追踪任务示意图。

要理解这条指令，需要具备几种不同的能力：

新的话语实体识别：识别何时引入新的话语实体。例如，一个碗 引入了一个新的话语实体，但最终的面糊 或任何……结块 则没有。

共指消解：将指代表达与话语实体关联起来。例如，一名轻型面糊 和最后一名面糊 指的是同一个实体。

话语实体追踪：追踪每个话语实体的状态变化。例如，鸡蛋被放入碗中，并与其他配料混合。

存在许多旨在评估这些能力的数据集（例如，Walker 等人，2006年；Pradhan 等人，2012；Rahman 和 Ng，2012；Weston 等人，2015年；Chen 等人，2018；Bamman 等人，2020；Uryupina 等人，2020），以及许多旨在解决这些任务的自然语言处理模型（例如，Haghighi 和 Klein，2010；Lee 等人，2011；Hill 等人，2016；Henaff 等人，2017；Ji 等人，2017；Lee 等人，2017；Bosselut 等人，2018；Gupta 和 Durrett，2019a,b；Aina 等人，2019；Toshniwal 等人，2020；Wu 等人，2020）。在大语言模型（LLM）的背景下，Tenney 等人（2019）、Clark 等人（2019）和 Sorodoc 等人（2020）发现，长短期记忆网络和基于 Transformer 的模型（如 BERT（Devlin 等人，2019））的表征确实捕捉到了共指关系。Loáiciga 等人（2022）以及 Schuster 和 Linzen（2022）发现，预训练模型能够

*共同贡献。作者排序通过公平的方式确定：让 Cookie 这只猫挑选一个彩色球。

检测名词短语是否引入话语实体，尽管这一过程并不完全系统化。

然而，关于大语言模型是否能够追踪话语实体的状态，主要还是通过间接评估来考察。Toshniwal 等人(2022年) 表明，GPT-2（Radford 等人，2019年）能够根据先前走法的简洁、非语言描述，学会预测有效的国际象棋走法。类似地，Li 等人(2023年) 表明，在黑白棋数据上训练的 GPT 模型能够预测有效的下一步走法，而且这些预测与模型对棋盘状态的内部表征相关联。尽管如此，这些结果并未说明大语言模型是否会追踪自然语言话语中表达的状态变化。最相关的评估来自Li 等人(2021年)，他们测试了模型表征是否编码了自然文本中描述的实体状态。他们使用探测分类器，发现可以从 T5（Raffel 等人，2020）和 BART（Lewis 等人，2020）中以较高准确率解码出这些状态。然而，正如我们在重新分析其结果时所展示的（第2节），这些结果并未为实体追踪提供确凿证据。因此，大语言模型能否在处理自然语言话语时追踪实体，仍是一个悬而未决的问题。-vesm玩大发快三然而，关于大语言模型是否能够追踪话语实体的状态，主要还是通过间接方式进行评估。Toshniwal 等人(2022年) 表明，GPT-2（Radford 等人，2019年）能够根据先前走法的简洁、非语言描述，学会预测有效的国际象棋走法。类似地，Li 等人(2023年) 表明，在黑白棋数据上训练的 GPT 模型能够预测有效的下一步走法，而且这些预测与模型对棋盘状态的内部表征相关联。尽管如此，这些结果并未说明大语言模型是否会追踪自然语言话语中表达的状态变化。最相关的评估来自Li 等人(2021年)，他们测试了模型表征是否编码了自然文本中描述的实体状态。他们使用探测分类器，发现可以从 T5（Raffel 等人，2020）和 BART（Lewis 等人，2020）中以较高准确率解码出这些状态。然而，正如我们在重新分析其结果时所展示的（第2节），这些结果并未为实体追踪提供确凿证据。因此，大语言模型能否在处理自然语言话语时追踪实体，仍是一个悬而未决的问题。xəвəр然而，关于大语言模型是否能够追踪话语实体的状态，主要还是通过间接方式进行评估。Toshniwal 等人(2022年) 表明，GPT-2（Radford 等人，2019年）能够根据先前走法的简洁、非语言描述，学会预测有效的国际象棋走法。类似地，Li 等人(2023年) 表明，在黑白棋数据上训练的 GPT 模型能够预测有效的下一步走法，而且这些预测与模型对棋盘状态的内部表征相关联。尽管如此，这些结果并未说明大语言模型是否会追踪自然语言话语中表达的状态变化。最相关的评估来自Li 等人(2021年)，他们测试了模型表征是否编码了自然文本中描述的实体状态。他们使用探测分类器，发现可以从 T5（Raffel 等人，2020）和 BART（Lewis 等人，2020）中以较高准确率解码出这些状态。然而，正如我们在重新分析其结果时所展示的（第2节），这些结果并未为实体追踪提供确凿证据。浩然而，关于大语言模型是否能够追踪话语实体的状态，主要还是通过间接方式进行评估。Toshniwal 等人(2022年) 表明，GPT-2（Radford 等人，2019年）能够根据先前走法的简洁、非语言描述，学会预测有效的国际象棋走法。类似地，Li 等人(2023年) 表明，在黑白棋数据上训练的 GPT 模型能够预测有效的下一步走法，而且这些预测与模型对棋盘状态的内部表征相关联。尽管如此，这些结果并未说明大语言模型是否会追踪自然语言话语中表达的状态变化。最相关的评估来自Li 等人(2021年)，他们测试了模型表征是否编码了自然文本中描述的实体状态。他们使用探测分类器，发现可以从 T5（Raffel 等人，2020）和 BART（Lewis 等人，2020）中以较高准确率解码出这些状态。然而，正如我们在重新分析其结果时所展示的（第2节），这些结果并未为实体追踪提供确凿证据。因此，大语言模型能否在处理自然语言话语时追踪实体，仍是一个悬而未决的问题。

贡献 本研究尝试通过设计一项任务来回答这个问题，该任务旨在评估语言模型追踪话语实体状态变化的能力（如图 1所示）。我们使用这项新任务评估 GPT-3（Brown 等人，2020）、GPT-3.5和 Flan-T5（Chung 等人，2022年），且不进行任何微调。我们发现，只有 GPT 3.5 系列中的模型能够进行实质性的实体追踪；这些模型既接受过文本训练，也接受过代码训练。随后，我们展示了较小的语言模型（T5）能够学会进行实质性的实体追踪，并且在泛化到包含更多操作或词汇重叠较低的状态描述方面也具备一定能力。我们的结果表明，语言模型能够学会追踪实体，但仅凭文本语料库进行预训练并不足以显现这种能力。更广泛地说，我们的任务也可作为研究语言模型中涌现的世界模型的实用工具（例如，Li 等人，2023年；Tsai 等人，2023年）。[1]

2 重新分析Li 等人(2021年)

我们首先考察Li 等人(2021年)，这是与我们最相关的工作。他们改编了两个现有的数据集 Alchemy（Long 等人，2016）和 TextWorld（Côté 等人，2019年），用于测试模型跟踪实体状态变化的能力。模型的输入是对初始世界状态的文本描述，后面接着状态变更指令。根据这些描述，模型需要识别每个实体的正确最终状态。例如，在 Alchemy 中，模型会接收到对 7 个烧杯及其所盛不同数量有色液体的公式化描述，后面接着操控其内容物的指令，例如将液体从一个烧杯倒入另一个烧杯，或排空一个烧杯。给定类似(2)这样的输入，模型需要识别出第一个烧杯中有 4 单位棕色液体，第二个烧杯中有 2 单位红色液体，而第三个烧杯是空的。

(2) 第一个烧杯中有*1*个绿色物体，第二个烧杯中有*2*个红色物体，第三个烧杯中有*3*个红色物体。将最后一个装有红色物体的烧杯倒入烧杯*1*中。混合。

使用此类描述，Li 等人（2021年）发现，以 T5 或 BART 对这些描述的编码作为输入的探测分类器能够正确预测 75–76% 实体的状态，表明实体追踪取得了一定程度的成功。

然而，当进一步仔细审查数据集和结果时，这一结论就变得值得质疑。具体而言，我们对 Alchemy 实验中的成功案例进行了细粒度分析。在该实验中，每条导致状态变化的指令执行后，都会对每个烧杯的状态进行探测。由于每条指令至多涉及两个烧杯（例如，将 X 倒入 Y），而总共有 7 个烧杯，因此，在数据集中，实际发生状态变化的烧杯被探测到的案例表示非常稀疏。事实上，所有被探测的烧杯状态中，有 62.7% 与初始状态相同，这意味着，一个始终预测初始状态的简单基线就已经能达到 62.7% 的准确率（Li 等人也指出了这一点）。另一个潜在的捷径是空的最终状态比例很高（32.4%）。[2] 对于这些案例，由于存在诸如排空第四个烧杯这样的清空指令，初始状态通常可以完全忽略。仅凭这条指令，就足以在不考虑第四个烧杯初始状态的情况下预测其最终状态。因此，这类示例同样

[1] <escape>[1]</escape>代码和数据可在以下地址获取: https://github.com/ sebschu/entity-tracking-lms。

[2] 该百分比还包括烧杯在初始世界状态描述中已为空的情况。在初始描述中烧杯为空但含有一些液体的情况占24.9%。

并不适合全面评估实体追踪。鉴于这两种简单场景的比例很高（合计 87.6%），只有 12.4 % 的数据点可以被视为真正评估了话语语境中逐步展开的状态变化。如果分别计算简单案例和非简单案例的准确率，探测分类器在简单案例上的准确率为 86.8%，而在非简单案例上仅为 3.1%，这表明报告中的大部分成功都来自简单案例。

总而言之，我们重新分析后发现，Li 等人（2021年）的结果并未提供确凿证据，证明语言模型具备实质性的状态跟踪能力。[3] 然而，目前尚不清楚这是由于实验设置存在问题，还是因为模型确实缺乏实体追踪能力。为此，我们提出了一种新的行为评估方法。

3 任务设计与数据集

3.1 设计要求

实体追踪能力在很大程度上应当不受特定语言形式的影响。对于能够正确追踪实体的模型而言，讨论的是烧杯还是食谱，或使用哪种特定句法结构，都不应影响追踪结果。因此，在评估意义是否以及如何得到表征时，这种能力很值得研究，因为至少经典语言模型只接受形式方面的训练（Bender 和 Koller，2020）。与此同时，真正的实体追踪应具备这种形式与实体状态相互独立的特性，而这给评估带来了挑战，因为必须确保无法根据单个词汇项或短语（例如 Alchemy 数据集中讨论的第*2*节中的单词排空）预测实体状态。此外，在大型语料库上预训练的语言模型可能已经学会了实体的常见状态，例如鸡蛋通常会被放进碗里。基于这些原因，任何用于评估实体追踪能力的任务都应符合以下四项设计要求：

1. 实体被探测的状态不应呈现与预训练数据中可能存在的状态相似的分布模式（另见 Linzen，2020）。

2. 不应仅凭单个词语或短语本身，在不考虑之前按顺序排列的话语的情况下，预测实体的状态。

3. 如果使用任何数据进行演示、微调或训练，那么训练集和评估数据之间应几乎没有词汇重叠。

4. 如果使用任何数据进行演示、微调或训练，那么任务不应能通过基于已观测数据点填充槽位来解决。

这些属性无法通过食谱等自然风格的数据集（Dalvi 等人，2015年）、科学文本（Dalvi 等人，2019年）或 Alchemy 和 TextWorld 数据集来保证；这些数据集此前曾用于评估实体追踪能力。因此，我们以程序化方式生成了满足这些属性的数据集。

3.2 数据集

在数据设计方面，我们借鉴了 Winograd (1971) 和 Li 等人（2021）的研究。我们的数据集由特定世界状态的文本描述及其后续的一系列变化组成。这些世界包含可以放置物体的盒子。物体可以被放入盒子、从盒子中取出，或从一个盒子移动到另一个盒子。我们将一个世界 $\mathcal{W}$ 定义为 $\mathcal{W} = (O, n, m, e)$，其中 O 是一个物体集合，n 是盒子的数量，m 是单个盒子最多能够容纳的物体数量，e 是初始世界状态下每个盒子中物体数量的期望值。对于我们的数据集，我们使用了 $n = 7$、$m = 3$、$e = 2$，并采用了一组表示可能放入盒子中的物品的名词（例如，书、石头、大脑；$|O| = 100$），这些名词选自英国国家语料库（BNC；Leech et al. 2001）中词频高于27的词汇列表。

一个数据集由多个不同的场景组成。一个场景由初始状态以及一组应用于该初始状态的操作构成。我们将每个场景中的操作数量（NumOps）固定为 12。我们随机抽取了 2200 个场景，其中初始状态和 12 个操作均为随机抽取。抽样过程经过设计，确保只有在当前世界状态下有效的操作才会被抽取。初始状态和操作使用预定义模板转换为自然描述。

与 desideratum 1 的关系 我们选择了在盒子之间移动物体的任务，因为这是一个实体的词汇内容无法提供线索来预测状态变化结果的领域（desideratum 1）。我们没有包括清空一个盒子的操作，因为该操作允许根据之前的

[3] Li 等人（2021年）还展示了另外两组实验。关于其他实验如何呈现类似问题的详情，请参阅附录A 。

将被舍弃的话语（期望条件2）。对于期望条件2，我们还进一步考虑了使用上下文依赖性更强的操作描述进行的实验。具体而言，我们测试了包含额外将框 N 中的内容移动到框 M 中操作的场景，该操作并未明确提及物体名称；此外，还测试了这样一些场景：只有了解某个框的当前状态，才能对操作中的物体描述进行完整消歧。对于要求 3，我们考虑了示范/微调与评估阶段的状态和操作措辞完全不同的实验（参见表2）。最后，对于所有实验，我们都计算了每个初始状态的“特征签名”，用于表示每个框中所包含的物体数量。[4] 随后，我们确保不存在两个初始描述完全相同（忽略物体名称）的示例，分别出现在训练集和评估集中的情况。这样可以防止该任务通过槽位填充（期望条件4）来解决。[5]

3.3 任务

我们将实体追踪任务定义如下：给定一段自然语言描述，其中先介绍世界的初始状态，随后包含 0–12 个改变状态的操作，必须正确识别描述结束时每个盒子中的内容。为了评估这一点，我们为每个场景中的每个操作之后的每个盒子都创建了一个测试示例。这相当于$n \times (\text{NumOps} + 1)$ 个示例（我们的数据集中共有 91 个示例）。每个示例都采用完形填空测试的形式。也就是说，输入描述初始状态及随后的一系列操作，并以框 N 中包含__结尾。预期输出是根据前缀描述，框 N 中的正确物体集合。示例见附录B 。

4 实验1：上下文内示范

在第一组实验中，我们使用少量实体追踪任务的上下文示例评估了预训练语言模型。这种方法既能在无需为该任务提供大量监督的情况下探测模型，也能引导模型以一致的格式输出最终状态，以便自动评估。[6]

4.1 模型

我们使用了已知能够通过上下文示例适应任务的模型：GPT- 3 175B（davinci：Brown 等人，2020）、GPT-3.5（text-davinci-003[7]）以及 Flan-T5（基础版和 XL 版：Chung 等人，2022年）。OpenAI 透露的关于其模型的有限信息[8] 表明，davinci是一种自回归语言模型，主要在文本语料库上训练。text-davinci-003在文本与代码混合数据上以语言建模目标进行训练，此外还通过强化学习，利用人类反馈进行微调。Flan-T5 基于 T5，这是一种采用去噪目标训练的序列到序列模型，并进一步在一系列任务上进行了指令微调。研究表明，这有助于模型更好地响应指令，无论是否提供示例（Chung 等人，2022年）。我们通过 OpenAI API 评估 GPT 模型，并使用 HuggingFace 库评估 Flan-T5（Wolf 等人，2020）。模型概述见表 1 ，实现细节见附录 C 。

我们将这些模型与一个基线进行了比较：该基线会从与待考察框出现在同一子句中的物体集合里随机输出 0 到 $m = 3$ 个物体。请注意，这个基线远强于完全随机的基线，后者会从所有提及的物体中选择输出。

4.2 提示与示例

我们的提示包括：(a) 任务的一般说明，(b) 两个任务示例，用于展示预期格式，(c) 初始状态描述及其后的一系列操作，以及 (d) 一个不完整的句子框 N 包含___ 模型应补全该句子（完整提示见附录E ）。我们使用的示例会一次性输出所有盒子的状态。然而，在早期实验中，即使上下文中的示例包含所有箱子状态的最终描述，Flan-T5 也经常只输出第一个箱子的状态。

[4]例如，初始状态的特征表示为：第一个盒子中有两个物体，其余盒子中各有一个物体，对应的表示为 2111111。[5]此外，与 Alchemy 设置相比，我们的设置还有一个优点：所需的额外推理能力较少。Alchemy 中的烧杯领域要求模型进行计数和简单算术运算（例如，先加入一单位液体，再加入两单位液体，结果就是三单位液体）。此外，Alchemy 中的一些操作还要求模型了解颜色如何混合（例如，红色液体和绿色液体混合会变成棕色液体）。盒子领域则无需这些能力。

[6]对于 GPT-3.5，我们能够指示模型以统一的格式输出预测，而无需任何上下文示例，因此这构成了一项真正的零样本实验。我们将在附录中讨论这项实验的结果D。

[7]https://beta.openai.com/docs/models/gpt-3[8] https://beta.openai.com/docs/model-index-for-researchers

模型	Size	代码?	额外训练
GPT-3 davinci	175B	✗	-
GPT-3 davinci-instruct-beta	175B	✗	人工示范（微调）
GPT-3 文本-davinci-001	175B	✗	人工示范 + 评分很高的模型输出（微调）
GPT-3.5 code-davinci-002	?	✓	-
GPT-3.5 文本-davinci-002	?	✓	人工示范 + 高评分的模型输出（微调）
GPT-3.5 text-davinci-003	?	✓	基于人类反馈的强化学习
Flan-T5 基础版	250M	✗	1.6K 个任务 + 指令（微调）
Flan-T5 XL	3B	✗	1.6K 个任务 + 指令（微调）

表 1：用于上下文内示例实验的模型概述。

因此，对于 Flan-T5，我们调整了提示，使其逐个输出每个框。[9]

形式—含义解耦的示例/测试不匹配（替代形式）如第3.1–3.2节所述，我们还尝试了一种设置，其中示例与测试样例在状态和操作描述的形式上并不匹配。在这种设置下，我们使用一组与示例中不同的物体名称，以及不同措辞的状态和操作描述来评估模型（见表2）。除了限定词*the* 和介词*into*之外，两组之间没有共同的词语（不过，根据分词方式，可能会有共同的子词）。

上下文依赖性更强的场景（MoveContents、AmbiRef）正如第 3.1–3.2 节所述，我们还对两个上下文依赖性更强的附加场景进行了实验。第一个场景（MoveContents）引入了一个额外操作 将框 N 中的内容移至框 M，该操作会将框 N 中的所有物体移至框 M，但不会明确列出被移动的物体。这要求模型依赖前文描述来确定被移动的物体集合，进一步减少了使用启发式方法、无需按时间顺序整合初始描述和操作即可预测最终状态的空间。第二个场景（AmbiRef）在物体名称中加入了形容词修饰（例如 大脑 和 小脑），

[9]为确认这种任务格式差异不会低估 GPT 模型的准确率，我们还进行了一项实验，提示 GPT 每次只输出一个框的状态。我们发现，与 Flan-T5 模型相反，提示 GPT 输出单个框时的准确率更低，而不是提示它输出所有盒子，因此我们可以排除任务格式上的这一差异低估 GPT 表现的可能性。

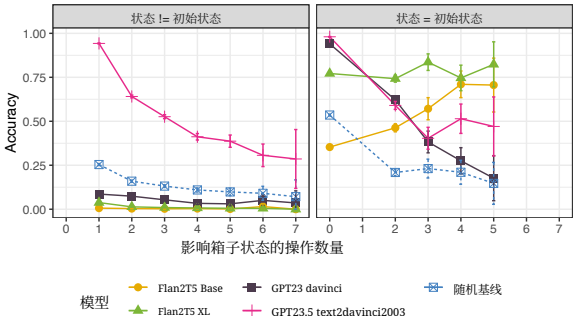


图2：在进行影响特定盒子的 n 次操作后，状态预测的准确率。左：内容不同于初始状态的盒子的预测；右：内容与初始状态相同的盒子的预测。误差线表示95%置信区间。

根据所描述的框的状态，在某些操作中可以省略形容词。具体而言，如果某个框中某一特定类型的物体只有一个，则省略修饰语（例如，将大脑从盒子*1*移至盒子*2* 如果盒子1中只有一个大脑）。在预测某个框中的内容时，模型会被要求输出带有正确形容词的物体类型，以充分消除对该物体指称表达的歧义。这同样要求按照时间顺序，将初始描述与各项操作进行组合，从而正确理解原本存在歧义的物体指称。（示例场景和操作请参见附录B 。）

4.3 评估

我们通过计算每次操作后预测每个框内容的准确率，来估计模型的实体追踪能力。鉴于我们依靠任意长度的完形填空来预测内容，因此必须对不受约束的生成结果进行评分。虽然我们只将输出提及给定框中所有物体（且不包含其他物体）的实例视为正确，但我们的评估允许回答形式存在细微偏差。具体而言，我们允许物体以任意顺序出现，物体名称之间可以用逗号或和分隔；带有限定词的完整名词短语（例如*thefurby*）和裸名词（例如，*furby*）均视为正确。

随着对特定框执行的操作越来越多，任务本身会变得更加困难，因为需要将初始状态描述与之后的所有操作依次结合起来。此外，与我们在第 2 节中的观察类似，每次操作最多会改变两个框的状态。

操作	Base	替代形式
Move	将汽车从盒子*1*移到盒子*3*。	拿起容器 *A* 中的毛绒玩具，并将其放入容器 *C*。
移除	从盒子*1*中移除汽车。	将毛绒玩具 *Furby* 从容器 *A* 中取出。
Put	把汽车放进盒子*1*。	将 *Furby* 放入容器 *A* 中。

表 2：AltForms 评估设置下状态变更操作的不同措辞

这意味着，对应较少操作的数据点数量远多于对应较多操作的数据点数量。基于这些原因，我们报告准确率时，会根据影响特定框的操作数量来展示，而不是报告总体准确率；所有结果均附有 95% 置信区间。

4.4 结果

图 2 展示了受到不同数量操作影响的框的预测准确率（例如，3 表示在初始状态之后，有三次操作改变了框中的内容）。左侧面板展示了探测状态与初始状态不同的实例；右侧面板展示了探测状态与初始状态相同的实例。正如左侧面板所示，只有 GPT-3.5`text-davinci-003` 始终优于（表现强劲的）随机基线。尽管不出所料，该模型的准确率也会随着操作次数的增加而下降，但即使在经过 7 次操作后，它仍能在超过 25% 的情况下正确预测框中的全部内容。另一方面，Flan-T5 模型似乎忽略了这些操作，主要预测初始状态描述。这一点可以从右侧面板中最终状态与初始状态一致时始终保持较高的准确率，以及左侧面板中最终状态偏离初始状态时始终保持较低的准确率看出。GPT-3 `davinci` 也主要重复初始状态，但正如右侧面板中准确率的急剧下降所表明的那样，即使是在重复初始状态时，它也会受到中间操作的干扰。

形式—意义解耦我们还评估了 `text-davinci-003`——这是第一组结果中唯一表现出非平凡实体追踪能力的模型——在第4.2节所述的替代形式设置下进行的表现；在该设置中，示例中的示范与测试示例之间词汇重叠较低。图3展示了`text-davinci-003`在我们数据的一个代表性子样本[10] 上的预测准确率。蓝线表示每

[10] 对于每种操作数量，抽取了 100 个状态，其中至少有一个示例包含 n 个操作。

当示范示例和测试示例中的描述词汇重叠较低时的表现。与原始结果（红线）相比可以看出，当影响一个框的操作超过两项时，使用不重叠的示范示例确实会导致表现略有下降。尽管如此，text-davinci-003 在许多情况下仍能预测实体的正确状态，进一步支持了其具备非平凡实体追踪能力这一结论。

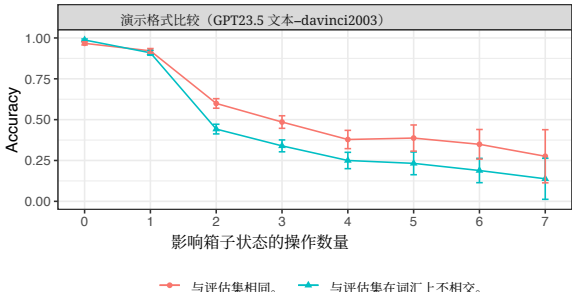


图 3：在示范示例与测试示例之间词汇重叠较低的情况下，text-davinci-003 的实体追踪准确率（替代形式）。

更强的上下文依赖性 最后，我们评估了 `text-davinci-003` 在第 4.2 节所述的另外两个数据集上的表现：AmbiRef 数据集，其中形容词修饰可以根据当前上下文省略；以及 Move-Contents 数据集，其中包含一个新操作将框 N 的内容移动到 M，该操作要求根据上下文识别框 N 的内容。图4展示了 `text-davinci-003` 在这两个数据集上的表现，并与我们的随机基线进行比较。由于

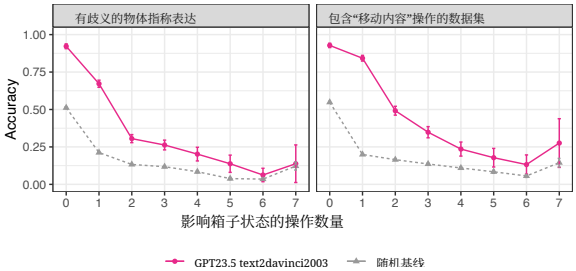


图 4：text-davinci-003 在 AmbiRef（左）和 MoveContents（右）数据集上的实体追踪准确率。

这些图表显示，即使在这些更具挑战性的情况下，我们仍观察到了不可忽视的实体追踪能力。然而，随着操作数量增加，性能会迅速接近随机基线，这表明需要联合考虑的状态变化越多，实体追踪就越容易失效。

4.5 讨论

我们的结果表明，在我们评估的模型中，只有 GPT-3.5 text-davinci-003 表现出了实质性的实体追踪行为。尽管随着操作数量增加，其表现有所下降，但即使在经历六到七轮操作序列后，该模型仍能做出许多准确的预测。此外，通过 AltForms 实验（其中示例与测试集之间的词汇重叠较低），我们也排除了以下可能性：示例教会了模型这项任务，或者模型主要依赖表面的槽位填充启发式。因此，我们得出结论，text-davinci-003 确实具备一定能力，能够在以语言表达的语境中追踪话语实体。在很大程度上依赖语境的 AmbiRef 和 MoveContents 实验中，我们也在较小程度上观察到了这种能力。这进一步证明，我们观察到的表现并非源于表面启发式；同时，这也凸显了即使是最新模型也会遇到困难的场景。

另一方面，GPT-3 davinci（规模可能与 GPT-3.5 text-davinci-003 相近）并未表现出实体追踪行为；该模型主要在文本语料库上以预测下一个词为目标进行预训练。对于经过微调、在结合指令和示例的多种任务上训练的去噪模型，情况也同样如此：Flan-T5 模型在非平凡示例上的准确率也接近于零。

总体而言，我们的结果表明，存在一种语言模型能够在一定程度上执行实体追踪，但并非所有在大型语料库上训练的足够大型的模型都必然具备这种能力。那么，哪些因素导致了这种差异？鉴于`davinci` 和`text-davinci-003` 至少在两个方面有所不同（`text-davinci-003` 基于一个在代码上训练的模型，并且经过了额外的人类反馈训练（Ouyang 等人，2022年）；见表1），我们的初步结果并未揭示究竟是什么导致了这种差异。因此我们

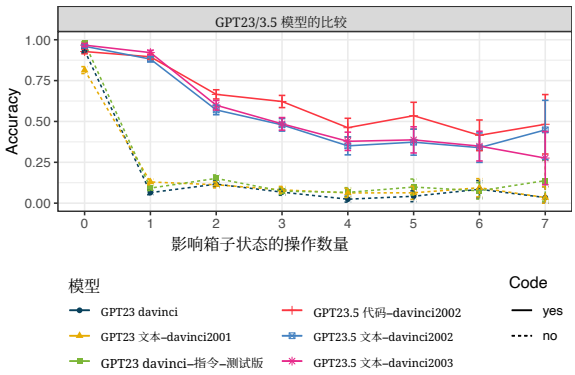


图 5：不同 GPT-3 模型的状态预测准确率。实线表示在代码和文本上训练的模型，虚线表示主要在文本上训练的模型。

我们进行了一项后续实验，比较了多种 GPT-3 和 GPT-3.5 模型，以确定导致 davinci 和 text-davinci-003 之间存在显著差异的因素。[11]

在代码上训练会促使模型表现出实体追踪行为

如表1所示，不同模型之间的两个关键差异维度是是否额外接受人类反馈训练，以及是否在代码上进行预训练。如果额外的人类反馈训练赋予语言模型追踪实体的能力，那么除 GPT-3 `davinci` 和 GPT-3.5`code-davinci-002` 之外的所有模型都应能够追踪实体。另一方面，如果在代码上进行预训练能提升实体追踪能力，我们预计所有 GPT-3.5 模型在我们的任务中都优于 GPT-3。如图5所示，在代码上训练过的 GPT-3.5 模型的表现始终优于 GPT-3 模型，包括未经过人类反馈训练的 `code-davinci-002` 。这表明，预训练数据中包含大量代码，有助于展现语言模型的实体追踪能力。

迄今为止，我们的结果尚未回答一个问题：模型大小在多大程度上会产生影响？与 GPT-3.5 模型规模相当的模型是否也能表现出有一定意义的实体追踪行为？由于不存在使用与 GPT-3.5 模型相同的目标和训练数据训练的小型模型，我们通过对 T5 进行微调实验来探究这个问题。

[11]为限制推理成本，我们使用了与 AltForms 实验中相同的数据子样本。

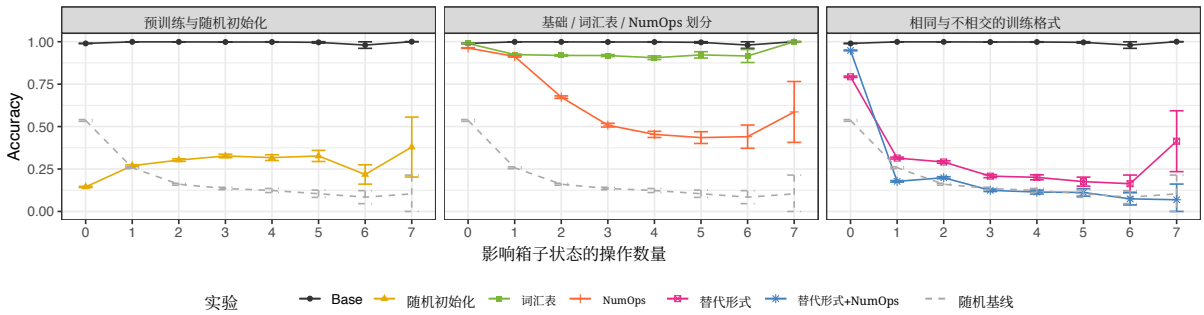


图 6：微调后的 T5 模型的结果。

5 实验2：微调

我们通过一系列监督训练实验，探究 T5 规模的小型模型是否能够学习追踪实体状态。

5.1 训练/测试划分

如第 3.1 节所述，评估实体追踪能力的一项挑战，是将这种能力与模板化槽位填充等简单启发式方法区分开来。因此，我们设计了多种训练/评估不匹配，以阻断以下所述的几种可能的捷径。

基础划分 在此，我们对训练和评估示例使用相同的格式。训练集和评估集中的初始状态各不相同，以阻止简单的槽位填充启发式方法，如第 3.2 节所述。

操作数划分 操作数划分将训练集中单个示例所含操作的最大数量限制为 2，但评估集中的操作数量最多可达 12。此划分旨在测试经过微调的模型能否泛化到比微调期间见过的更长的操作序列。

词汇/替代表达划分 词汇划分用于测试：训练期间未使用过的物体是否也能被充分追踪。我们整理了一份相对罕见的物体名称列表（例如柚子、福比玩偶、 *Flav-R-Straw* ；不见于 BNC），并使用两组完全不相交的物体名称来抽取训练集和测试集。训练集使用罕见物体列表，测试集则使用原始物体列表。替代形式划分采用第 4.2 节所述的设计。这些划分旨在厘清模型究竟是学会了将特定词语/短语与操作关联起来，还是微调使其获得了更具泛化性的实体追踪能力。

歧义指代/移动内容划分 AmbiRef 和 MoveContents 划分遵循第4.2节所述的设计。这些划分旨在测试模型是否能够在不考虑受影响盒子的当前状态的情况下，学会解读描述不充分的操作。对于这些划分，训练示例和测试示例采用相同的格式。

5.2 模型

我们对 Li 等人的(2021年）中表现最佳的模型 T5-base 进行了评估，方法是在上述每个数据集上对其进行微调。作为额外的基线，我们将其与直接在我们的数据集上训练、参数随机初始化的 T5 进行了比较。

5.3 结果与讨论

预训练 T5 能够学会执行实体追踪如图 6 （左）所示，对 T5 进行微调后，其在基础划分上的准确率接近完美。这表明模型有能力学会这项任务。使用随机初始化的 T5 进行训练却没有取得同样的结果：随机权重训练出的模型准确率低得多，因为模型几乎只会预测盒子是空的。这两项结果表明，预训练对于模型学会这项任务至关重要。此外，模型的实体追踪能力能够适应测试时出现的新物体名称，准确率仅略有下降（图 6，中）。仅使用最大长度为 2 的操作序列进行训练（操作数划分），会导致模型在较长操作序列上的性能下降更多；但即使面对较长的操作序列，模型仍能在超过 45% 的情况下推断出正确的最终状态。最后，当训练示例与测试示例的词汇重合度较低时，模型性能会显著下降（图 6，右）。不过，当模型使用最多包含 12 个操作的序列进行训练时（粉色线），其性能仍高于随机基线。如果我们只使用最多包含两个

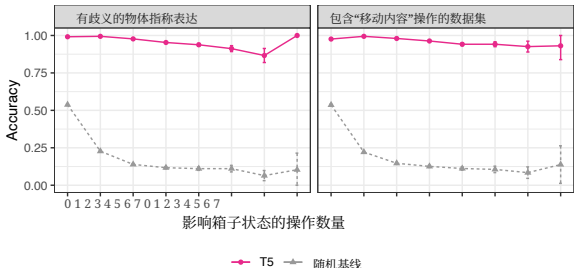


图 7：针对 AmbiRef（左）和 MoveContents（右）数据集微调的 T5 的准确率。

个操作进行训练（蓝色线），性能下降则会进一步加剧，且不再高于随机基线。这些结果表明，在实体追踪任务上进行微调，确实能使模型获得可泛化至许多具有挑战性的场景的实体追踪能力。然而，与此同时，随着微调集与评估集之间的差异越来越大，模型性能会迅速下降。微调若仅限于箱子环境这类有限领域，能在多大程度上迁移为更普遍的实体追踪能力，并应用于更自然的话语中，仍是一个有待解答的问题。

解读依赖上下文的操作仍然具有挑战性如果我们将包含含糊指称表达的操作（图7，左）或移动内容 操作（图7，右）纳入微调和评估数据中，就会发现性能略有下降，而且模型不再能达到近乎完美的准确率，尽管训练和评估所用的示例格式相同。在这两种情况下，模型几乎所有错误都出现在需要解读依赖上下文的操作的示例中。这表明，即使模型接受了直接的训练信号，依赖上下文的操作仍会加剧实体追踪的难度。

6 结论

我们旨在研究预训练大语言模型是否表现出实体追踪行为。我们设计了一项任务，使我们能够评估大语言模型是否能够根据实体的初始状态描述以及作用于该实体的操作来预测其状态。在第一组实验中，我们发现，GPT-3 davinci（一种原始预训练语言模型）和 Flan-T5（一种指令微调语言模型）无法完成这项任务，只是简单地重复初始状态描述。另一方面，GPT-3.5 模型（与上述模型最核心的区别在于预训练语料库中包含代码）则表现出表现出有限但非平凡的实体追踪行为。即使实体状态经历了许多次操作，它们的表现仍始终优于强随机基线。在第二组实验中，我们展示了较小的模型（如 T5）也能在很大程度上学会这种行为。针对这一任务对模型进行微调后，它还能够对与训练数据在多个维度上存在差异的示例进行实体追踪。综合来看，我们的结果表明：(a) 主要使用文本数据训练的普通语言模型，开箱即不具备实体追踪能力；(b) 代码和文本数据预训练能显著提升这种能力；[12] (c) 针对这一任务进行微调，也能使主要使用文本数据训练的较小模型展现出这种行为，不过，微调后较小模型的这项能力究竟有多强的泛化性，仍是一个悬而未决的问题。

同时使用文本和代码进行训练为何有效？要生成正确的代码，追踪变量的状态很重要。因此，我们推测，相较于纯文本数据，这类预训练数据可能为模型追踪实体提供更强的信号。也可能如先前所推测的那样（Potts，2020；Mer-rill et al.，2021年，等等），代码预训练能提供额外的落地。

当前结果还凸显了预训练数据组成的重要性。我们的结果表明，在大规模语料库上以语言建模目标进行训练的模型，在实体追踪方面的表现会因训练数据是否包含大量代码而大不相同。鉴于此，未来的研究可以更广泛地探究代码预训练的影响，包括其对模型表征的影响，以及它在多大程度上促进语言模型中世界模型的涌现。

最后，我们还提出了评估状态跟踪能力时应遵循的若干原则。除了这些具体原则之外，我们还提出一个更普遍的观点：在评估与意义相关的能力时，需要考虑模型可能采用哪些策略来完成任务，并确保测试示例不会模仿训练集或微调数据的分布模式。只有这样，我们才能恰当地评估

[12]这也与Sap 等人（2022年）的观察相符：GPT-3.5 在回答有关物体位置变化的事实性问题时，表现明显优于 GPT-3。此外，对于代码预训练似乎有益的其他任务，请参见Madaan 等人（2022年）。

语言模型的意义相关能力。

局限性

本研究的一个局限在于，我们仅考察了行为数据，因此难以确立实体追踪能力与在本任务中取得高性能之间完全的因果联系。实体追踪是一种高层次的语言行为，而要在本任务中达到高准确率，还需要许多其他能力。因此，我们无法排除这样的可能性：我们在不同模型间观察到的行为差异，主要是由其他某种能力的差异所驱动的，例如组合式地理解句子的能力（参见Bogin 2022年 和 Bogin et al. 2022年，其中有证据表明，GPT-3 和 GPT-3.5 模型在组合泛化行为上存在差异）。

对我们的设置的一种可能批评是，它要求的短期记忆能力超出了大多数人（甚至可能是所有人）的记忆能力。也就是说，如果给人类呈现与模型相同的输入，由于记忆限制，我们不会预期他们能够追踪所有 7 个盒子的内容。因此，我们可能是在期待模型实现超越人类水平的实体追踪；其他语言能力的模型评估也曾因采用这种设置而受到批评（Lampinen，2022年）。尽管如此，鉴于受评估模型的架构，我们认为这项任务是合理的。基于 Transformer 的模型可以在其上下文窗口内回看整个输入序列中的任意词元，因此，要在人类和模型之间进行恰当的比较，应当以书面形式向人类呈现完整描述，并在提示他们说出某个框内的内容后，让他们重新阅读描述。虽然我们没有在更大规模的人群中正式评估人类是否具备这种能力，但就我们个人而言，在能够查阅书面描述时，追踪盒子中的内容并不困难。

此外，我们设计这项任务时，确保完整文本能够放入预训练语言模型的上下文窗口中。然而，正如我们在引言中提到的，实体追踪是理解长上下文的一项重要能力。由于上下文窗口有限，我们的结果并不适用于长度超过模型上下文窗口的文本；要对更长的文本进行恰当的实体追踪，可能需要采用不同的模型架构。

此外，尽管我们发现 GPT-3.5 模型以及微调后的 T5 模型在我们的任务中追踪实体的准确率高于强大的随机基线，但我们的结果也表明，一旦对某个实体执行了多个操作，这种行为就不太稳定。因此，我们的结果不应被视为在需要高准确率的关键应用中使用这些模型的依据。

最后，本研究仅评估了英语模型。鉴于我们发现，即使训练示例与评估示例之间的词汇重叠度不高，模型仍能在一定程度上追踪实体，因此我们的结果似乎也适用于其他语言。不过，这是否确实如此，仍有待进一步探讨。

致谢

我们感谢 Jacob Andreas、Ellie Pavlick、Allyson Ettinger、Tal Linzen、Will Merrill 以及纽约大学计算与心理语言学实验室的成员参与讨论，也感谢 Belinda Li 分享模型输出，并介绍其数据准备流程和实验的相关细节。我们感谢 Cookie 参与作者署名决策并给予情感支持。本研究部分依托纽约大学 IT 高性能计算资源、服务及工作人员的专业支持开展，并获得美国国家科学基金会（NSF）通过授予计算研究协会的 #2030859 号拨款（CIFellows 项目）以及欧洲研究理事会（ERC）通过欧盟“地平线 2020”研究与创新计划提供的资助（拨款协议编号 #948878）。本文中表达的任何观点、研究结果、结论或建议均属于作者，不一定反映美国国家科学基金会或计算研究协会的观点。

参考资料

Laura Aina、Carina Silberer、Ionut-Teodor Sorodoc、Matthijs Westera 和 Gemma Boleda。2019年。以实体为中心的模型学到了什么？来自多方对话实体链接的见解。载于*2019*年北美计算语言学协会年会论文集：人类语言技术，第 *1* 卷（长篇和短篇论文），第 3772–3783 页，美国明尼苏达州明尼阿波利斯。计算语言学协会。

David Bamman、Olivia Lewke 和 Anya Mansoor。2020。一份关于英语文学中指代消解的标注数据集。载于第十二届语言资源与评估会议论文集，第 44–54 页，

法国马赛。欧洲语言资源协会。

Emily M. Bender 和 Alexander Koller。2020 年。迈向自然语言理解：数据时代的意义、形式与理解。载于计算语言学协会第*58*届年会论文集，第5185–5198页，线上。计算语言学协会。

Ben Bogin。2022年。GPT-3 在使用上下文学习时，在 COVR-10splits 上的表现如何？Twitter 讨论串。

Ben Bogin、Shivanshu Gupta 和 Jonathan Berant。2022年。未观测到的局部结构使组合泛化变得困难。载于*2022*年自然语言处理经验方法会议论文集（*EMNLP 2022*年），阿布扎比，阿拉伯联合酋长国。计算语言学协会。

Antoine Bosselut、Omer Levy、Ari Holtzman、Corin En- nis、Dieter Fox 和 Yejin Choi。2018 年。使用神经过程网络模拟动作动态。载于 国际学习表征会议（*ICLR 2018*）。

Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D. Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Ma- teusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。2020。语言模型是少样本学习者。载于 神经信息处理系统进展，第 33 卷，第 1877–1901 页。Curran Associates, Inc.

Hong Chen、Zhenhua Fan、Hao Lu、Alan Yuille 和 Shu Rong。2018 年。PreCo：用于共指消解的学前词汇大规模数据集载于*2018* 年自然语言处理实证方法会议论文集，第 172–181 页，比利时布鲁塞尔。计算语言学协会。

Hyung Won Chung、Le Hou、Shayne Longpre、Barret Zoph、Yi Tay、William Fedus、Yunxuan Li、Xuezhi Wang、Mostafa Dehghani、Siddhartha Brahma、Al- bert Webson、Shixiang Shane Gu、Zhuyun Dai、Mirac Suzgun、Xinyun Chen、Aakanksha Chowdh- ery、Alex Castro-Ros、Marie Pellat、Kevin Robinson、DashaValter、Sharan Narang、GauravMishra、Adams Yu、Vincent Zhao、Yanping Huang、Andrew Dai、Hongkun Yu、Slav Petrov、Ed H. Chi、Jeff Dean、Ja- cob Devlin、Adam Roberts、Denny Zhou、Quoc V. Le 和 Jason Wei。2022年。扩展指令微调语言模型。arXiv:2210.11416。

Kevin Clark、Urvashi Khandelwal、Omer Levy 和 ChristopherD.Manning。2019年。 BERT 关注什么？对 BERT 注意力机制的分析。载于 *2019*年*ACL*研讨会 *BlackboxNLP*：自然语言处理神经网络的分析与解释，第 276–286 页，意大利佛罗伦萨。计算语言学协会。

Marc-Alexandre Côté、Ákos Kádár、Xingdi Yuan、Ben Kybartas、Tavian Barnes、Emery Fine、James Moore、Matthew Hausknecht、Layla El Asri、Mahmoud Adada、Wendy Tay 和 Adam Trischler。2019年。Textworld：一个用于文本游戏的学习环境。载于计算机游戏，第41–75页，沙姆。施普林格国际出版公司。

Bhavana Dalvi、Niket Tandon、Antoine Bosselut、Wen- tau Yih 和 Peter Clark。2019 年。万事皆有因：在程序性文本中发现行动的目的。载于《*2019*年自然语言处理经验方法会议暨第九届自然语言处理国际联合会议论文集（*EMNLP-IJCNLP*）》，第4496–4505页，中国香港。计算语言学协会。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：用于语言理解的深层双向变换器预训练。载于《计算语言学协会北美分会*2019*年会论文集：人类语言技术，第*1*卷（长篇与短篇论文）》，第4171–4186页，美国明尼苏达州明尼阿波利斯。计算语言学协会。

Jeroen Groenendijk 和 Martin Stokhof。1991 年。动态谓词逻辑。语言学与哲学，14：39–100。

Aditya Gupta 和 Greg Durrett。2019a。有效利用变换器网络进行实体追踪。载于*2019*年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集，第759–769 页，中国香港。计算语言学协会。

Aditya Gupta 和 Greg Durrett。2019年b。跟踪离散和连续的实体状态以理解过程。载于第三届自然语言处理结构化预测研讨会论文集，第 7–12 页，明尼阿波利斯，明尼苏达州。计算语言学协会。

Aria Haghighi 和 Dan Klein。2010年。 模块化以实体为中心的模型中的共指消解。载于人类语言技术：北美计算语言学协会 *2010* 年年会，第 385–393 页，美国加利福尼亚州洛杉矶。计算语言学协会。

Irene Heim。2002年。文件变更语义学与定指的熟悉性理论。载于《形式语义学：必读文选》，第9章，第223–248页。牛津：Blackwell 出版社。

Mikael Henaff、Jason Weston、Arthur Szlam、Antoine Bordes 和 Yann LeCun。2017年。利用循环实体网络追踪世界状态。载于国际学习表征会议（*ICLR 2017*）。

Felix Hill、Antoine Bordes、Sumit Chopra 和 Jason Weston。2016 年。金发姑娘原则：运用显式记忆表示阅读儿童书籍。载于国际学习表征会议（*ICLR 2016*）。

Yangfeng Ji、Chenhao Tan、Sebastian Martschat、Yejin Choi 和 Noah A. Smith。2017 年。神经语言模型中的动态实体表示。载于 *2017* 年自然语言处理实证方法会议论文集，第 1830–1839 页，丹麦哥本哈根。计算语言学协会。

Hans Kamp、Josef Van Genabith 和 Uwe Reyle。2011年。话语表征理论，第125–394页。SpringerNetherlands，Dordrecht。

Lauri Karttunen。1976。话语指称项。载于 J. D. McCawley 编，《句法与语义学，第 7 卷》，第 363–386 页。Academic Press。

Chloé Kiddon、Ganesa Thandavam Ponnuraj、Luke Zettlemoyer 和 Yejin Choi。2015年。Mise en place：对食谱说明进行无监督解读。收录于*2015*年自然语言处理经验方法会议论文集，第 982–992 页，葡萄牙里斯本。计算语言学协会。

Andrew Kyle Lampinen。2022年。语言模型能否处理递归嵌套的语法结构？一项比较模型与人类的案例研究。arXiv:2210.15303。Heeyoung Lee、Yves Peirsman、Angel Chang、Nathanael Chambers、Mihai Surdeanu 和 Dan Jurafsky。2011。斯坦福大学在 CoNLL-2011 共享任务中采用的多轮筛选共指消解系统。载于第十五届计算自然语言学习会议：共享任务论文集，第28–34页，美国俄勒冈州波特兰。计算语言学协会。Kenton Lee、Luheng He、Mike Lewis 和 Luke Zettlemoyer。2017。端到端神经共指消解。载于*2017*年自然语言处理实证方法会议论文集，第188–197页，丹麦哥本哈根。计算语言学协会。Geoffrey Leech、Paul Rayson 和 Andrew Wilson。2001。英语书面语和口语中的词频：基于英国国家语料库。朗文。Mike Lewis、Yinhan Liu、Naman Goyal、Marjan Ghazvininejad、Abdelrahman Mohamed、Omer Levy、Veselin Stoyanov 和 Luke Zettlemoyer。2020。BART：用于自然语言生成、翻译和理解的去噪序列到序列预训练用于自然语言生成、翻译和理解。载于第*58*届计算语言学协会年会论文集，第7871–7880页，线上。计算语言学协会。

Belinda Z. Li、Maxwell Nye 和 Jacob Andreas。2021年。神经语言模型中的意义的隐式表征。收录于计算语言学协会第 *59* 届年会暨自然语言处理国际联合会议第 *11* 届会议论文集（第 *1* 卷：长篇论文），第 1813–1827 页，线上。计算语言学协会。

Kenneth Li、Aspen K Hopkins、David Bau、Fernanda Viégas、Hanspeter Pfister 和 Martin Wattenberg。2023年。涌现的世界表征：探索一个在合成任务上训练的序列模型。收录于国际学习表征会议（*ICLR 2023*）。

Tal Linzen。2020年。我们如何加速迈向类人语言泛化的进程？载于计算语言学协会第*58*届年会论文集，第5210–5217页，线上。计算语言学协会。

Sharid Loáiciga、Anne Beyer 和 David Schlangen。2022年。新的还是旧的？探索预训练语言模型如何表示话语实体。载于第*29*届国际计算语言学会议论文集，第875–886页，庆州，大韩民国。国际计算语言学委员会。Reginald Long、Panupong Pasupat 和 Percy Liang。2016年。通过模型投影实现更简单的依赖上下文的逻辑形式。载于第*54*届计算语言学协会年会论文集（第*1*卷：长篇论文），第1456–1465页，柏林，德国。计算语言学协会。Aman Madaan、Shuyan Zhou、Uri Alon、Yiming Yang 和 Graham Neubig。2022年。代码语言模型是少样本常识学习者。载于*2022*年自然语言处理实证方法会议论文集，第1384–1403页，阿布扎比，阿拉伯联合酋长国。计算语言学协会。William Merrill、Yoav Goldberg、Roy Schwartz 和 Noah A. Smith。2021年。从无根基形式中获取意义的可证明局限：未来的语言模型能理解什么？计算语言学协会汇刊，9:1047–1060。Mante S. Nieuwland 和 Jos J. A. Van Berkum。2006年。当花生坠入爱河：话语力量的 N400 证据。认知神经科学杂志，18(7):1098–1111。

OpenAI。2023年。 GPT-4 技术报告。arXiv:2303.08774。

Long Ouyang、Jeff Wu、Xu Jiang、Diogo Almeida、Car- roll L Wainwright、Pamela Mishkin、Chong Zhang、Sandhini Agarwal、Katarina Slama、Alex Ray 等。2022年。训练语言模型，使其能够遵循指令并利用人类反馈。arXiv:2203.02155。

Christopher Potts。2020年。语言模型是否有可能实现语言理解？Medium 博客文章。

Sameer Pradhan、Alessandro Moschitti、Nianwen Xue、Olga Uryupina 和 Yuchen Zhang。2012。CoNLL-2012 共享任务：在 OntoNotes 中对多语言不受限共指进行建模。载于自然语言处理经验方法会议与 *CoNLL* 联合会议——共享任务，第 1–40 页，韩国济州岛。计算语言学协会。

Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei 和 Ilya Sutskever。2019年。语言模型是无监督多任务学习者。

Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J Liu。2020。《探索统一文本到文本转换器的迁移学习极限》。机器学习研究期刊，21：1–67。

Altaf Rahman 和 Vincent Ng。2012。解决确定代词的复杂案例：Winograd 模式挑战。载于*2012*年自然语言处理经验方法联合会议暨计算自然语言学习会议论文集，第777–789页，韩国济州岛。计算语言学协会。

Maarten Sap、Ronan Le Bras、Daniel Fried 和 Yejin Choi。2022年。神经心智理论？大型语言模型社会智能的局限。载于*2022*年自然语言处理经验方法会议论文集，第3762–3780页，阿布扎比，阿拉伯联合酋长国。计算语言学协会。

Sebastian Schuster 和 Tal Linzen。2022年。当句子未引入话语实体时，基于 Transformer 的模型有时仍会提及它。载于《*2022*年计算语言学协会北美分会年会论文集：人类语言技术》，第 969–982 页，美国西雅图。计算语言学协会。

Ionut-Teodor Sorodoc、Kristina Gulordava 和 Gemma Boleda。2020。 探究语言模型中的指称信息。载于 第*58*届计算语言学协会年会论文集，第4177–4189页，线上。计算语言学协会。

Ian Tenney、Dipanjan Das 和 Ellie Pavlick。2019年。BERT重新发现了经典的自然语言处理流程。载于第*57*届计算语言学协会年会论文集，第4593–4601页，意大利佛罗伦萨。计算语言学协会。

Shubham Toshniwal、Sam Wiseman、Allyson Ettinger、Karen Livescu 和 Kevin Gimpel。2020。学习忽略：使用有界记忆神经网络进行长文档共指。载于 *2020*年自然语言处理经验方法会议论文集（自然语言处理经验方法会议），第 8519–8526 页，线上。计算语言学协会。

Shubham Toshniwal、Sam Wiseman、Karen Livescu 和 Kevin Gimpel。2022年。将国际象棋作为语言模型状态跟踪的测试平台。载于 *AAAI*人工智能会议论文集，第36卷，第11385–11393页。

Chen Feng Tsai、Xiaochen Zhou、Sierra S Liu、Jing Li、Mo Yu 和 Hongyuan Mei。2023年。大语言模型能否玩好文本游戏？当前最先进水平与开放问题。arXiv:2304.02868。

Olga Uryupina、Ron Artstein、Antonella Bristot、Feder- ica Cavicchio、Francesca Delogu、Kepa J Rodriguez 和 Massimo Poesio。2020 年。在多种体裁中标注广泛的回指现象：ARRAU语料库。《自然语言工程》，26(1):95–128。

ChristopherWalker、StephanieStrassel、Julie Medero 和 Kazuaki Maeda。2006年。《ACE 2005 多语言训练语料库》。语言数据联盟，费城：*LDC2006T06*。

Jason Weston、Antoine Bordes、Sumit Chopra、Alexan- der M Rush、Bart Van Merriënboer、Armand Joulin 和 Tomas Mikolov。2015年。迈向AI完备问答：一组先决玩具任务。arXiv:1502.05698。

Terry Winograd。1971年。用于理解自然语言的计算机程序中数据的表征方式。博士论文，麻省理工学院。

Thomas Wolf、Lysandre Debut、Victor Sanh、J ulien Chaumond、Clement Delangue、Anthony Moi、Pier- ric Cistac、Tim Rault、Remi Louf、Morgan Funtow- icz、Joe Davison、Sam Shleifer、Patrick von Platen、Clara Ma、Yacine Jernite、Julien Plu、Canwen Xu、Teven Le Scao、Sylvain Gugger、Mariama Drame、Quentin Lhoest 和 Alexander Rush。2020年。Transformers：自然语言处理领域的最新进展。载于 *2020*年自然语言处理经验方法会议：系统演示论文集，第38–45页，线上。计算语言学协会。

Wei Wu、Fei Wang、Arianna Yuan、Fei Wu 和 Ji wei Li。2020 年。CorefQA：将共指消解作为基于查询的跨度预测。载于第*58*届会议论文集

计算语言学协会年会，第6953–6963页，线上。计算语言学协会。

A Li 等人（2021年）研究结果的补充分析

如脚注3所述，Li 等人 （2021年）又进行了两项实验，乍看之下，这些实验为语言模型中的隐式意义表征和状态跟踪能力提供了更多证据。然而，这两项实验的数据和设置也很可能高估了模型的能力。

在第二组探测分类器实验中，Li 等人 (2021年) 使用 TextWorld 引擎生成的数据（Côté 等人，2019年）。在这种设置中，文本游戏里有对多个实体的文本描述（例如，一扇木门），以及玩家执行的动作（例如，打开木门）。他们的探测分类器接收某个实体及该实体某个属性的表征（例如，木门是关着的），或者两个实体及它们之间某种关系的表征（例如，特大号床位于卧室里），并根据这些表征预测：结合初始描述和玩家执行的一系列动作，给定命题是真是假。只有涉及已被提及实体的命题才会被探测。这种设置的问题在于，在训练集和评估集划分中，有许多命题始终为真（例如，在所有游戏模拟中，抽屉柜都在卧室里，因此无论之前的上下文如何，探测分类器对这一输入都应始终返回真）。此外，即使对于训练数据和评估数据中并非始终为真的实体—属性命题和实体—关系—实体命题，数据也存在严重偏差；一个不考虑初始描述或用户动作、仅预测训练数据中最常见答案的基线（违反要求 3，见第 3.1节）已经能达到 88.5% 的准确率，这让报告的探测分类器准确率 96.9% 更具参考意义。

此外，Li 等人（2021年）还展示了一项实验，他们操控了 Alchemy 数据集合成版本中的特定实体表征。在这项实验中，他们首先使用一个经过微调、用于预测下一步操作的语言模型，对初始描述 D 和一项操作 O 进行编码；该操作会影响烧杯 X，从而得到表征 R_1。然后，他们对相同的初始表征 D 以及一项操作进行编码，操作形式为

Drain n from beaker Y，其中烧杯 Y始终不同于烧杯 X，而 n 是根据初始描述，烧杯 Y 中所含液体的量。由此得到表征 R_2。接着，他们从表征 R_2 中提取烧杯 Y的初始描述表征，并将 R_1 中烧杯Y的表征替换为R_2 中相应的表征，从而得到 R_{mixed}。然后，他们将 R_{mixed} 作为 T5 的输入，并表明，与根据 R_1 或 R_2 预测下一步操作相比，根据 R_{mixed} 预测出的下一步操作与两项操作（ R_1 中编码的操作和 R_2中编码的操作）均相符的情况要多得多。他们据此认为，初始描述的词元表征编码了执行操作后的实体状态。这个实验的问题在于，用来计算 R_2 的操作始终是从第 Y 个烧杯中排空 n 的形式，因此烧杯 Y的最终状态总是空的。因此，这个实验主要表明，词元“排空”影响了初始状态的表征（进而影响了对下一步操作的预测），但并未更普遍地证明，被操控烧杯 Y的实际状态完全编码在其初始状态描述中。要回答这个问题，就必须用更复杂、不会泄露最终状态的操作重复这项实验（这违反了期望条件2，见第 3.1 节）。

总之，Li 等人（2021年）中的额外实验也违反了我们在第3.1节提出的一些期望，因此很难从这些实验中得出有关状态跟踪能力的结论。

B 示例输入-输出对

(3) 展示了我们基础数据集（NumOps 在框 6 = 2）中的一个输入-输出示例对。

(3) a. 输入： 盒子*0*里有画作，*Box1*里有铃铛，盒子*2*里有吉他，盒子*3*里有鸡蛋、镜子和床单，盒子*4*里有化学品，盒子*5*里有磁盘和电线，盒子*6*里有玻璃杯和刀。把玻璃杯从盒子*6*移到盒子*4*。把礼物放进盒子*5*。把吉他从盒子*2*移到盒子*6*。把牛奶放进盒子*4*。从盒子*3*中取出镜子和床单。盒子*3*。盒子*6*__输出： 里有吉他和刀。

(4)展示了数据集中一个输入-输出示例，其中的操作包含指代表达，若不考虑初始状态和先前操作（Nu-mOps，关于框 6 = 2），这些表达就会产生歧义。请注意，要正确理解 把吉他从盒子*2*移到盒子*6*，模型必须考虑到蓝色吉他（而非红色吉他）在该操作之前位于盒子2这一信息。因此，脱离上下文来看，这个操作存在歧义。

（4）a. 输入： 盒子*0*包含黄色书籍、绿色花朵和红色吉他，盒子*1*包含小炸弹、小书和蓝色骨头，盒子*2*包含蓝色吉他，盒子*3*包含蓝色铃铛，盒子*4*包含绿色纸张、黄色便签和黄色电视机，盒子*5*包含黄色铃铛，盒子*6*为空。将吉他从盒子*2*移到盒子*6*。将蓝色电线和大电视机放入盒子*5*。将花朵从盒子*0*移到盒子*6*。盒子*6*__b. 输出： 包含蓝色吉他和绿色花朵。

(5)展示了数据集中的一个输入-输出示例，其中包含将内容从操作（Nu- mOps 对框 6 = 2）。请注意，为了正确理解将盒子*2*中的内容移至盒子*6*，模型必须考虑到操作发生前，茶（而不是红色吉他）位于盒子2中的信息。

(5) a. 输入： 盒子*0*包含风扇、礼物和信件，盒子*1*包含啤酒、镜子和领带，盒子*2*包含茶，盒子*3*包含靴子，盒子*4*包含外套、盘子和衬衫，盒子*5*包含瓶子，盒子*6*是空的。将盒子*2*中的物品移到盒子*6*。将连衣裙和画作放入盒子*5*。将信件从盒子*0*移到盒子*6*。盒子 *6* __ b. 输出： 包含信件和茶。

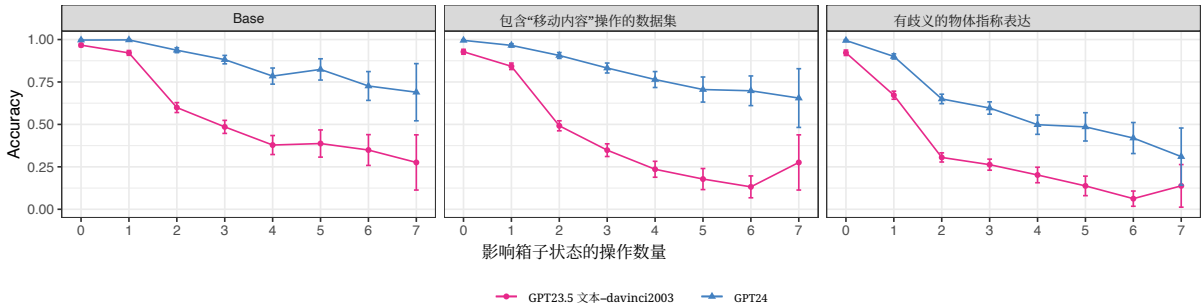


图 8：在两样本上下文学习实验中，text-davinci-003 和 GPT-4 的实体追踪准确率。

C 实现细节

上下文学习 对于 GPT-3，我们使用了 OpenAI API。我们采用贪心解码，将温度参数设为 0，将目标生成长度的上限设为 150，并将换行符用作额外的停止词元。GPT-3 的推理时间和模型大小数据不可用，但我们总共生成了约 1600 万个词元。本文报告的所有 GPT-3 实验均于 2023年1月进行。

对于 Flan-T5，我们使用了大小为 3 的束搜索，并将目标生成长度上限设为 256。其他超参数均采用 HuggingFace 库中 T5ConditionalGeneration 的默认值。在单张 A100 GPU 上，T5-XL 的推理耗时约为 5 小时，T5 基础版约为 2 小时。

微调我们使用批次大小为 8、学习率为 1×10^{-4}，对 T5 进行了单轮训练。在初步探索中，增加微调训练轮次并未显著提升开发集性能，有时甚至会产生负面影响。在单张 RTX8000 GPU 上，训练和推理耗时约 3 小时。

D 附加结果

D.1 GPT-3.5 零样本结果

如正文所述，GPT-3.5 无需任何上下文示例，就能够按照正确的格式输出盒子的内容（提示模板见表 5）。

图 9 比较了`text-davinci-003` 在2样本设置和零样本设置下的性能。在没有示例的情况下，模型性能略有下降，这表明示例确实有助于引导模型正确完成任务。尽管如此，我们发现在零样本设置下模型仍表现出不可忽视的性能，这进一步印证了GPT 3.5展现出非平凡的实体追踪能力这一结论。

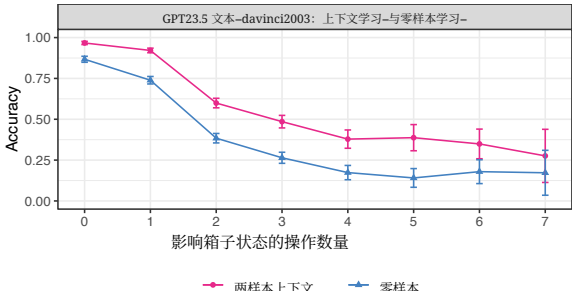


图 9：GPT 3.5 在上下文学习（2-shot）和零样本设置下的实体追踪准确率。

D.2 GPT-4 结果

我们还通过 OpenAI API 测试了近期发布的 GPT-4 模型（OpenAI，2023年）。图 8 比较了 GPT-3.5`text-davinci-003` 和 GPT-4 在 base 数据集（左）以及第 4.2 节中讨论的更具挑战性的 MoveContents（中）和 AmbiRef（右）数据集上的表现。在所有数据集上，GPT-4 的表现都显著优于 GPT-3.5。但与 GPT-3.5 类似，随着影响一个框的操作增多，其性能会下降。鉴于目前没有关于 GPT-4 的训练数据、训练过程或架构细节的公开信息，我们无法对 GPT-4 相较于之前模型性能提升的原因得出任何结论。

E 提示

表格3 展示了上下文实验中使用的 2-shot 提示。表格4 展示了 AltForms 实验中使用的 2-shot 提示，其中示例包含的描述与测试示例之间的词汇重叠程度较低。表格5 展示了附录 D中讨论的零样本提示。所有提示均可在以下网址获取：`https://`

github.com/sebschu/entity-tracking-l ms.

F 数据集统计信息与许可证信息

描述性统计数据请参见表格6 和7 。所有数据集均依据 GNU 通用公共许可证 v3.0 发布。

一次性查询所有框的 2-shot 提示（GPT-3 实验）

根据“Description:”之后的描述，在“Statement:”之后写出一个关于所有框及其内容的真实陈述。

描述：盒子0包含汽车，盒子1包含十字架，盒子2包含包和机器，盒子3包含纸和绳子，盒子4包含账单，盒子5包含苹果、现金和玻璃杯，盒子6包含瓶子和地图。

陈述：盒子0里有汽车，盒子1里有十字架，盒子2里有包，而
机器，盒子3里装着纸和绳子，盒子4里装着账单，盒子5里装着 the
苹果和现金以及玻璃杯，盒子6包含瓶子和地图。

描述：盒子0中有汽车，盒子1中有十字架，盒子2中有包和机器，盒子3中有纸和绳子，盒子4中有账单，盒子5中有苹果、现金和玻璃杯，盒子6中有瓶子和地图。从盒子0中移除汽车。从盒子3中移除纸和绳子。将飞机放入盒子0。将地图从盒子6移至盒子2。从盒子4中移除账单。将外套放入盒子3。

陈述：盒子0装有飞机，盒子1装有十字架，盒子2装有包、机器和地图，盒子3装有外套，盒子4是空的，盒子5装有苹果、现金和玻璃杯，盒子6装有瓶子。

描述：{description} 陈述：
盒子0装有

逐个查询盒子的 2-shot 提示（T5 实验）

根据“描述：”之后的描述，按照“陈述：”之后的描述，写出一条关于某个盒子及其所装物品的真实陈述。

描述：盒子0装有汽车，盒子1装有十字架，盒子2装有包和机器，盒子3装有纸和绳子，盒子4装有账单，盒子5装有苹果、现金和玻璃杯，盒子6装有瓶子和地图。陈述：盒子1装有十字架。

描述：盒子0装有汽车，盒子1装有十字架，盒子2装有包和机器，盒子3装有纸和绳子，盒子4装有账单，盒子5装有苹果、现金和玻璃杯，盒子6装有瓶子和地图。从盒子0中取出汽车。从盒子3中取出纸和绳子。将飞机放入盒子0。将地图从盒子6移到盒子2。从盒子4中取出账单。将外套放入盒子3。陈述：盒子2装有包、机器和地图。

描述：{description} 陈述：框
{boxnum} 包含

表3：包含两样本上下文示范的提示。

来自测试示例的具有不相交表面形式的 2-shot 提示

根据“Description:”后的描述，写出一个关于所有容器或框及其内容的正确陈述，填入“Statement:”后。

描述：意式脆饼在容器 A 中，冰柱在容器 B 中，花岗岩和机器在容器 C 中，对开本和百科全书在容器 D 中，账单在容器 E 中，叉勺、折叠刀和星冰乐在容器 F 中，剪刀和瓢虫在容器 G 中。

陈述：容器 A 包含意式脆饼，容器 B 包含冰柱，容器 C 包含花岗岩和机器，容器 D 包含对开本和百科全书，容-器 E 包含账单，容器 F 包含叉勺、折叠刀和星冰乐，容器 G 包含剪刀和瓢虫。

描述：意式脆饼在容器 A 中，冰柱在容器 B 中，花岗岩和机器在容器 C 中，对开本和百科全书在容器 D 中，账单在容器 E 中，叉勺、折叠刀和星冰乐在容器 F 中，剪刀和瓢虫在容器 G 中。将意式脆饼从容器 A 中取出。将对开本和百科全书从容器 D 中取出。将四足动物放入容器 A 中。拿起容器 G 中的瓢虫，并将其放入容器 C 中。将账单从容器 E 中取出。将口香糖球放入容器 D 中。

陈述：容器 A 包含四足动物，容器 B 包含冰柱，容器 C 包含花岗岩、机器和瓢虫，容器 D 包含口香糖球，容器 E 不包含任何物品，容器 F 包含叉勺、折叠刀和星冰乐，容器 G 包含剪刀。

描述：{description} 陈述：
盒子0包含

表 4：具有不相交表面形式的两样本上下文示范提示。

零样本提示

根据“Description:”后面的描述，按照“Statement:”后面的描述，写出一个关于所有盒子及其内容的真实陈述。陈述格式为：Box0 包含 A，盒子1 包含 B，盒子2 包含 C，盒子3 包含 D，盒子4 包含 E，盒子5 包含 F，盒子6 包含 G。A、B、C、D、E、F、G 是各个盒子内物品的占位符。

描述：{description}
陈述：盒子0包含

表 5：零样本实验的提示模板。

数据集	场景		示例	
	演示	Test	演示	Test
完成	1	990	14	90,090
子样本	1	491	14	5,012
AmbiRef.	1	474	14	4,991
MoveContents	1	451	14	4,956

表 6：实验 1 的数据集统计。

数据集	场景			示例		
	训练集	Dev	Test	训练集	Dev	Test
Base	990	220	990	90,090	20,020	90,090
NumOps	990	220	990	20,790	20,020	90,090
词汇表	990	220	990	90,090	20,020	90,090
替代形式	990	220	990	90,090	20,020	90,090
替代形式+NumOps	990	220	990	20,790	20,020	90,090
AmbiRef	990	220	990	90,090	20,020	90,090
MoveContents	990	220	990	90,090	20,020	90,090

表 7：实验 2 的数据集统计。