

本科毕业论文



论文题目： **基于语言模型的日常任务规划研究**

姓 名： 黄呈松 学 号：19302010004

院 系： 软件学院

专 业： 软件工程

指导教师： 郑骁庆 职 称：副教授

单 位： 复旦大学软件学院

完成日期： 2023 年 5 月 1 日

|  |
| --- |
| **论文撰写人承诺书**  本毕业论文是本人在导师指导下独立完成的，内容真实、可靠。本人在撰写毕业论文过程中不存在请人代写、抄袭或者剽窃他人作品、伪造或者篡改数据以及其他学位论文作假行为。  本人清楚知道学位论文作假行为将会导致行为人受到不授予/撤销学位、开除学籍等处理（处分）决定。本人如果被查证在撰写本毕业论文过程中存在学位论文作假行为，愿意接受学校依法作出的处理（处分）决定。  **承诺人签名：**  **日期： 20 年 月 日** |

|  |  |  |  |
| --- | --- | --- | --- |
| 指导教师对论文学术规范的审查意见：  □ 本人经过尽职审查，未发现毕业论文有学术不端行为。  □ 本人经过尽职审查，发现毕业论文有如下学术不端行为：  **指导教师签名： 日期： 20 年 月 日** | | | |
| 指导教师评语：  **签名**：  **20 年 月 日** | | 答辩委员会（小组）评语：  **签名**：  **20 年 月 日** | |
| **学分** |  | **成绩** |  |
| **备注：** | | | |

目录

[摘要 III](#_Toc134348992)

[ABSTRACT IV](#_Toc134348993)

[第一章 引言 5](#_Toc134348994)

[1.1 研究背景 5](#_Toc134348995)

[1.2 研究目的和意义 5](#_Toc134348996)

[1.2.1 研究目的 5](#_Toc134348997)

[1.2.2 研究意义 6](#_Toc134348998)

[1.3 核心内容和创新点 6](#_Toc134348999)

[1.3.1 核心内容 6](#_Toc134349000)

[1.3.2 创新点 6](#_Toc134349001)

[1.4 相关工作简介 6](#_Toc134349002)

[1.4.1 常识推理 6](#_Toc134349003)

[1.4.2 预训练语言模型与表格预训练 7](#_Toc134349004)

[1.4.3 日常任务智能化 8](#_Toc134349005)

[1.5 文章结构简介 8](#_Toc134349006)

[第二章 任务定义与先导知识 9](#_Toc134349007)

[2.1 背景知识 9](#_Toc134349008)

[2.1.1 置身任务 9](#_Toc134349009)

[2.1.2 语言指令 9](#_Toc134349010)

[2.1.3 任务规划 10](#_Toc134349011)

[2.2 任务详细定义 11](#_Toc134349012)

[2.2.1 任务形式化 11](#_Toc134349013)

[2.2.2 数据形式 12](#_Toc134349014)

[2.3 石器时代 13](#_Toc134349015)

[2.4 工业革命 13](#_Toc134349016)

[2.5 信息革命 13](#_Toc134349017)

[第三章 算法分析与模型建立 14](#_Toc134349018)

[3.1 模型建立 14](#_Toc134349019)

[3.2 算法流程 14](#_Toc134349020)

[3.3 详细设计与实现 14](#_Toc134349021)

[3.3.1 Foo 14](#_Toc134349022)

[3.3.2 Bar 14](#_Toc134349023)

[第四章 实验验证与效果分析 15](#_Toc134349024)

[4.1 数据集 15](#_Toc134349025)

[4.2 进行实验 15](#_Toc134349026)

[4.3 实验结果 15](#_Toc134349027)

[4.4 实验分析 15](#_Toc134349028)

[第五章 结论 16](#_Toc134349029)

[参考文献 17](#_Toc134349030)

[致谢 20](#_Toc134349031)

# 摘要

Test

关键词：关键词1，关键词2，关键词3，关键词4，关键词5

# ABSTRACT

英文摘要与中文摘要对应，只要意思符合即可，无须逐字对照翻译。

Key words：

# 引言

## 研究背景

日常机器人是一个正在发展的领域。随着人口老龄化的加剧，更加需要可以在家中完成一系列日常任务的机器人，来缓解护理人员不足的问题。这类机器人可以完成例如自动化打扫卫生，完成早饭准备等不需要专业知识的工作。而将人工智能应用在这类机器人中也是十分有前景的方向。已有一些工作通过人工智能算法来操控机器人在房间内完成这些工作，这些算法主要通过摄像头看到的图片与文本指令来执行相关动作。

在这一过程中，任务规划是至关重要的。以我们给机器人下令“泡一杯茶”为例，我们期望机器人自动的完成“找到茶杯茶叶，找到热水，泡好茶，端上来”这整个过程，而不是需要用户分四次布置这样的指令。这要求我们可以通过“泡一杯茶”这一全局指令来生成详细的四步规划。这类规划不同于简单的路线规划，不仅需要对于周围环境的认识，同时也需要常识知识的帮助，例如在这个例子中，需要了解泡茶需要热水这一常识性的知识才能做出正确的规划。如何利用这些常识知识进行

预训练语言模型在不同的任务中都取得了不错的性能，例如问题回答和常识推理等。这证明了预训练语言模型拥有一些解决这些问题所需要的常识知识，例如微波炉可以加热食物。更进一步，已有一些工作研究预训练语言模型是否可以将这些知识用于日常任务的规划中。但受限于预训练语言模型没有观察外部世界和与外部世界交互的能力，先前的工作并没有将环境信息纳入考虑。使得这些任务的规划中缺少例如“向左转”类似的导航规划。而这些规划是机器人可以适应新的环境的关键因素。

## 研究目的和意义

### 研究目的

在已有的ALFRED数据集的基础上，完善一个从任务命令到具体步骤的数据集，对于这个任务提出一个更好的更完善的评价指标。在该基础上实现一个可以完成日常任务规划的语言模型

### 研究意义

规划算法是家政机器人智能化的重要组成部分。通过研究机器人规划算法，可以使家政机器人能够自主的进行决策和规划，以实现更加复杂的任务，如清扫，洗衣，烹饪等。这些规划依赖于大量的常识知识，而已有的预训练语言模型被证明拥有了常识推理的能力，因此本文期望将预训练语言模型用于家政机器人的规划算法。同时完成的新数据集也可以帮助之后的研究者在这一任务上更好的进行研究。

## 核心内容和创新点

### 核心内容

通过已有ALFRED数据集构建一个从任务命令加环境信息到具体步骤的数据集，通过预训练语言模型来完成具体的规划过程，并通过表格的方法将环境信息加入输入，试图使得语言模型可以生成与环境相关的规划。同时提出每次只生成下一个步骤的迭代式规划方法。最后证明这两种方法都可以提升日常任务的规划性能。

### 创新点

本文首次将环境信息通过表格的形式进行编码，使得预训练语言模型有了了解环境信息的能力，进而利用语言模型内涵的常识知识和推理能力进行日常任务的任务规划。本文同时新建了全新的该任务上的数据集和提出了全新的评价方法，给科研社区在该问题的研究打下基础。

## 相关工作简介

### 常识推理

常识推理（commonsense reasoning）是指通过每个人都有的普遍基本的知识或者经验进行推理，来推断出其他未知的信息。这一能力是人类认知过程的核心组成部分，同时也是构建自然语言理解系统或者人工智能系统的核心任务[1][2]。传统方法（深度学习）通过知识表示和知识图谱解决常识推理的相关任务[3]。随着深度学习的发展，通过预训练语言模型来解决常识推理任务成为了主流方法。

不同于人类通过周围环境了解常识信息[4],先前的工作证明了预训练语言模型通过在大规模无标签语言库的训练过程中可以获得常识相关的知识[5]。因此预训练语言模型开始在没有外联知识库的情况下被经常的用于解决常识推理的问题[6]。有许多任务从各个角度考察了与训练语言模型对于尝试知识的应用例如常识抽取[7]，下一状态预测[8]，文化和社会认知[9]等。

而已有一些工作将预训练语言模型用于日常任务规划当中。例如ALFWorld[10]构建了一个文字游戏的场景，在这种场景下检测语言模型在日常任务中的常识推理能力。语言模型可以根据任务目标和当前状态来生成下一步的步骤。ScienceWorld[11]同样通过文字游戏的方式来检验模型的常识推理能力，不同于ALFWorld在日常任务中完成，ScienceWorld主要着眼于小学生科学实验常识例如热胀冷缩等。SayCan[12]通过预训练语言模型生成一系列语义上可行的规划，之后通过强化学习算法来构建价值函数（value function）来选择在当前环境下的操作方法。同样GPT-3等大规模语言模型也被用于从文本中总结计划等[13]。

### 预训练语言模型与表格预训练

预训练加微调是近几年来常用的将深度学习应用在下游任务的方法。预训练模型主要通过大规模语料进行无监督训练。预训练的编码器-解码器模型例如BART[14]和T5[15]在各种自然语言生成任务上都取得了不错的效果。尤其实在一些被认为跟常识推理相关的任务比如CommonsenseQA[16]的优秀效果更是说明了这类预训练语言模型具有很强的常识推理能力，可能在这次的任务上获得不错的效果。

表格是一种数据组织的方式，被广泛应用于记录，分析和呈现数据。可以更好的将大量数据集合在一起进行直观的展示。不同于常规的文本语句，表格带有一些具有语义的结构信息例如标题，行和列等，因此不能简单的将预训练语言模型直接应用在表格型数据上。在这样的背景下有许多先前的工作对于预训练模型进行了更多的表格优化的预训练方法。表格预训练主要目标在两个方向：生成更好的表格表示和将表格作为中间表示。

对于生成更好的表格表示，从TabNet[17]使用采用序列关注（Sequential Attention）的方式，使用预测遮掩特征（Masked Features）作为无监督的与训练任务，得到更好的表格表示。TAPAS[18]在BERT所使用的MLM（masked language model）的基础上加上了判断表格是否符合文本的与训练任务并且加上对于行列以及排名的embedding层来获得更好的表格cell表示。TAPEX[19]通过大量表格结合SQL语句，通过SQL的执行结果进行大规模有监督训练。

### 日常任务智能化

机器人技术的发展使得机器人可以完成日常中的一些任务，然而这类家居机器人可以真正的进入日常生活需要机器人可以通过自然语言和摄像头了解到的知识来完成这样的任务，在这样的基础上AIFRED（Action Learning From Realistic Environments and Directives）数据集[20]应运而生。任务集包括了在多个不同场景下的不同类型的任务。环境可以进行交互式操作等，以此为基础，前任提出了多种不同的解决方案用于操纵机器人在这类环境中的动作。

FILM[21]（Following Instruction in Language with Modular Methods）证明了显式空间记忆（explicit spatial memory）和语义搜索策略（semantic search policy）可以在执行日常任务过程中为环境和指令提供更好的编码方式。同时证明了在没有专家轨迹和具体每一步指令的情况下，模型也可以完成一些任务，这说明了模型具有从高级指令分析出具体的每一步步骤的能力，作为支撑证明了本文的方法有很强的可行性。

LEBP[22]（Language Expectation & Binding Policy）首先将每一步的step生成文本上的单个字的表示，之后通过绑定政策将这个表示对应到具体的操作，例如导航到某一点或对于面前的物体进行某种操作。该方法说明了，自然语言的步骤可以有效迁移到未见过的场景中，同时说明了分步骤的指令的重要价值，为本文研究提供了意义。

## 文章结构简介

文章分为todo

# 任务定义与先导知识

本章节将会先介绍任务规划任务任务所需要的先导知识，之后给出任务的详细定义与数据来源。

## 背景知识

### 置身任务

置身任务（embodied task）是指一种任务或者活动，需要执行者再现实世界中进行实际的身体行动，而不仅仅实在虚拟环境或者抽象的思维层面中进行，例如打扫房间，搬运物品，跳舞等都属于置身任务。

而在计算机科学领域中，一般执行置身任务的被称为智能体（agent）。智能体是一种能够自主行动，执行任务或者决策的实体。它们能够接受来自外部环境的输入，然后跟书事先定义的规则或者算法来执行相应的操作或者输出相应的响应。例如聊天机器人，智能家居系统或者自动驾驶汽车都可以被看作是智能体。

AIFRED（Action Learning From Realistic Environments and Directives）[20]数据集是首个关注现实环境下的置身任务的基准评估（benchmark），尽管其中绝大部分人是家政任务。这个数据集旨在测试智能体（agent）或是机器人在现实环境下执行置身任务的能力。同时在这个基准测试中，智能体需要理解自然语言的指示并且在名为AI2-THOR（The House Of inteRactions）[23]的游戏引擎创造的虚拟环境中完成具体给定的任务。

### 语言指令

ALFRED数据集中的置身任务都包括两类语言指令（language instruction），全局指令与细分指令。这两类语言指令也都会作为智能体的输入。智能体的输出是一个可执行的动作序列（action sequence），主要包括转向，移动和与场景中的物体交互等。

全局指令是指如“把茶壶放到架子上”这一任务的全面概括，智能体无法通过简单的规则或是算法将其转变为可执行的动作序列，而细分指令例如“左转后前进到桌子”是简单的可以根据模板等方法简单的转变为动作序列的。

ALFRED数据集中的语言指令均是由人工设计，首先由智能体随机在场景内活动，之后由人工筛选出有意义的事件，并且补充相关的两类指令。

### 任务规划

本文所指的任务规划（task planning）并非是在长时间的任务排布，而是如何在场景内找到可行的动作序列完成这一任务。由于细分指令可以很简单的转变为动作序列，因此我们认为将任务划分为细分指令就可以完成这一任务。考虑到现实中的应用场景，机器人作为智能体在家中完成人类所需要的任务是，也很难获得细分的指令，因此这里的任务规划指的就是将人类所给出的全局指令映射到对应的细分指令上。

这类规划任务往往需要一些潜在的常识性知识来完成。例如当全局指令为“端一份热面包过来”时，显然需要有面包需要通过烤面包机来加热这一常识性知识。由于这些常识性知识的数量巨大导致了这部分知识无法通过硬编码等方式嵌入规则或者代码当中。然而语言模型往往在预训练过程中获得了这些常识性知识，因此通过语言模型来完成规划任务显得理所应当。

先前的工作已经证明了大预言模型（large language model）例如GPT-3可以通过生成任务的方法完成这一任务[24]，然而这些规划缺少了现实环境的基础，因为语言模型没有针对特定环境进行设计。这导致了这些语言模型所生成的计划并不能直接应用于智能体上。例如“将茶壶从炉子移到架子上”这一全局指令，置身的智能体需要对于茶壶位置的信息来使得自己可以到达这一位置。人类可以通过对于环境观察，了解他们现在的位置来形成一个精确的可执行的规划，来提出例如“右转之后前进五米”这类细分指令。

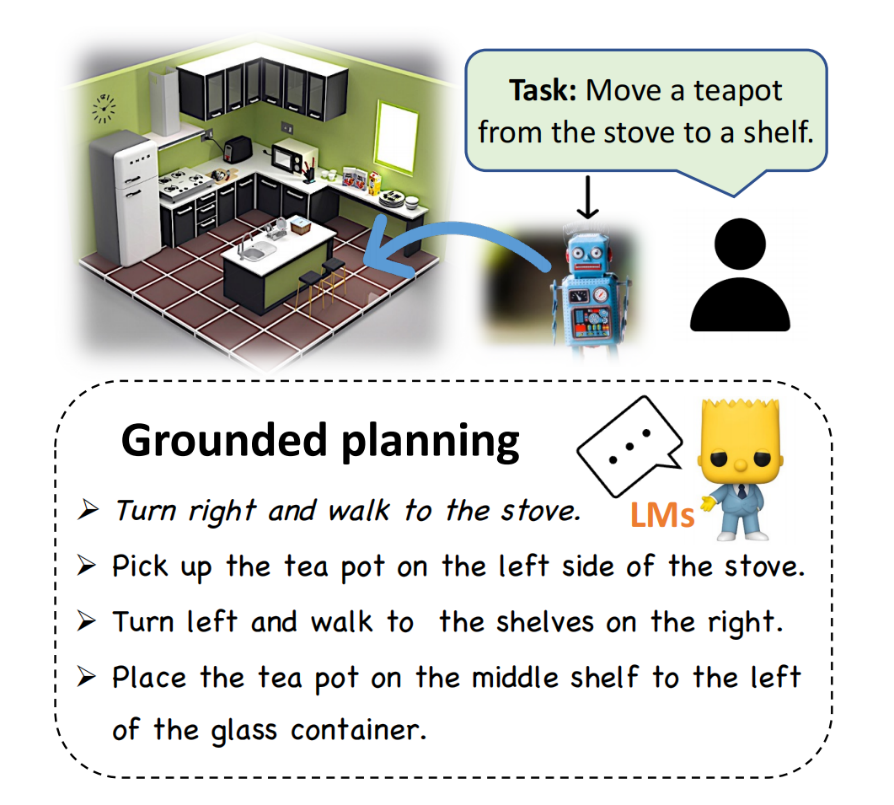


图 1：任务规划样例

## 任务详细定义

图1中给出了一个任务规划任务的数据样例，本节后续介绍了任务规划任务的详细定义，给出了该任务的目标和相关细节，并且对于所有后续所使用的变量给出统一的规范定义，使得任务表述更加规范。因此这个任务是为了完成基于环境的置身任务规划G-PlanET（Grounded planning for embodied tasks）。

### 任务形式化

在整个任务规划的问题中，智能体需要找到映射，在环境中根据全局指令生成一系列细分指令

其中全局指令和细分指令都是自然语言形式的，而环境可以由其中的对象集合表示，即

物体的信息也可以通过其个属性（attribute）来表示，即

最终整个任务期望找到一个映射f使得

### 数据形式

为了构建一个用于研究G-PlanET任务的大规模数据集，本人重复使用了ALFRED的目标和计划，并从AI2THOR中提取对象信息以对齐环境。ALFRED数据集使用AI2-THOR引擎为具有自我中心视觉的代理提供交互环境来执行操作。然而，该数据集不包含有关环境中对象的显式数据（例如，它们之间的关系、旋转和空间关系）。

本人开发了一套转换程序，利用AI2THOR重新使用ALFRED基准来评估本文中所使用的方法。本人成功获得了一个结构化数据表来描述ALFRED数据集中每个任务的环境。本人探索了AI2THOR引擎并编写转换程序，以便我们可以获得所有对象的完整观测数据：属性（可移动、可打开等）、位置（三维坐标和旋转）、大小和空间关系（例如，对象A在对象B的顶部）。

本人相信我们的ALFRED数据集变体将成为社区研究该任务和基于实地推理的未来方向的重要资源。

由于输入长度限制等原因，对于每一个环境中的对象仅保留了特定的几个属性作为对象的表征。筛选属性选用了和该任务最相关的几个属性：

1. 对象编号：区分对象的唯一标识。
2. 对象名字：自然语言表示的对象名称，用于引入与该对象相关的常识知识。
3. 对象位置：通过对象的三维坐标表示，用于智能体找到该对象。
4. 对象旋转：同样通过三维坐标表示，例如柜子方向等信息，表达某些对象仅能在某些方向与之进行交互。
5. 对象父节点编号：例如放在橱柜中的碗，其父节点就是橱柜，这一属性为橱柜的节点编号。

其中表示环境的对象除了房间中的物体之外还包括两种特殊的对象：

1. 智能体初始位置：由于智能体的初始位置会影响后续的细分指令，因此智能体的初始位置是该任务下的重要信息，因此我们将智能体的初始位置作为对象编号0的对象纳入环境表征当中。
2. 墙：由于整个任务在一个有限空间完成，因此需要一个对于xy坐标的限制，这一限制通过建造四座虚拟的墙来完成，通过虚拟墙这一方式来表达对于xy坐标的尺度（scale）和相对位置。

## 文本表示

G-PlanET任务中的语言指令包括环境信息中的对象名称均使用自然语言，然而计算机无法理解自然语言，因此需要将文本信息转化成为计算机可以理解的编码方式，这个过程被称为文本表示，它能够帮助计算机理解自然语言，随着技术的发展文本表示经历了多个不同的阶段。

### 基于计数的文本表示方法

#### 独热编码（One-Hot Encoding）

独热编码是最简单的文本表示方法，将每个词或者字与一个整数对应，这样每一个文本可以用一个稀疏向量进行表示，向量中的不同位置表示某个词或者字的出现或者缺失。这种编码方式的问题在于无法表示词之间的关系。

#### 词袋模型（Bag of Words）

词袋模型将文本编码为一个词汇表中所有单词或汉字的频率向量，这种方法解决了不同词之间的关系但没有办法处理词序信息，即没有办法给“前进后右转”和“右转后前进”不同的文本表示。

#### N元语法模型（N-gram）

N-gram[25]是一种考虑了语序的文本表示方法，通过文本相邻的N的单词建立一个统计模型，用于预测下一个词出现的概率，虽然可以编码上下文信息，但是随着N的增加会使得向量维度快速增加，引发维度灾难。

### 基于预训练的文本表示方法

除了基于计数的文本表示方法以外，还有基于大规模无标签语料库的预训练 文本表示方法，其中主要将字词映射为高维空间中的连续向量，这种分布式表示可以分为上下文无关的文本表示和上下文相关的文本表示。

#### 上下文无关的文本表示

当使用基于预训练模型的方法来进行文本表示时，Word2vec[26]是一种常用的方法。Word2vec提出了两种模型：跳字模型（Skip-gram）和连续词袋模型（continuous bag of words CBOW）。这两种模型都可以用于从大规模文本语料中学习单词的分布式表示，这些表示将单词映射为向量，并具有捕捉单词语义信息的能力。

在跳字模型中，我们固定一个中心词，并预测在其周围的上下文单词。相反，在连续词袋模型中，我们使用中心词周围的单词来预测中心词。这两种模型都是基于单词在上下文中出现的频率和位置来学习单词的表示，这使得Word2vec可以捕捉到单词的语义和上下文信息。

然而，Word2vec的模型存在一个固定大小的上下文窗口，这限制了它在学习单词表示时能够利用其他包含广泛上下文信息的语料位置。这种限制可以导致Word2vec在处理某些类型的语言问题时表现不佳。

为了解决这一问题，GloVe[27]通过共现矩阵，调整了损失函数

其中，为训练得到的词表示，为偏置，是共现矩阵中的值，代表单词出现在单词的上下文中的频次。

#### 上下文相关的文本表示

上下文相关的文本表示（Contextualized Text Representations）是一种文本表示方法，它可以根据上下文动态地生成不同的表示。与传统的固定词向量不同，上下文相关的文本表示可以根据输入文本的上下文、语法和语义信息来动态地生成不同的表示，从而更好地捕捉文本的语义和上下文信息。同时上下文相关的文本表示可以很好的处理上下文歧义或是单词的多义性以此提高文本表示的准确性和鲁棒性（robustness）。

ELMo（Embeddings from Language Models）[28]使用双向长短期记忆神经网络（LSTM，Long-Short Term Memory）[29]来构建，是最早的上下文相关的文本表示。通过优化损失函数

通过ELMo就可以对于一个文本获得其表示作为下游任务的输入。

本文采用的其他语言模型有着各自的编码方式将在后续章节详细介绍，其中的文本表示方法均为上下文相关的文本表示。

## 本章小结

todo

# 算法分析与模型建立

## 概述

预训练语言模型（Pretrained Language Models, PLMs）已经成为自然语言处理领域研究的重要方向之一。它通过无监督方式在大规模语料库上训练，可以产生具有良好表示能力的词向量和句向量，为许多自然语言处理任务提供强大的基础支持。其中，编码-解码模型（Encoder-Decoder Models）由于其优秀的生成能力，已经成为当前最流行的PLMs之一。

除了在生成式任务中表现出色，这些大型语言模型还表现出在常识推理任务中具有一定的通用常识。例如，CommonsenseQA[30]任务要求对给定的问题和场景进行常识推理，这些预训练的encoder-decoder语言模型也展现出了很好的性能。这表明，这些模型在大量的训练数据中学到了大量的常识知识，并能够将这些知识应用于不同的任务中。

鉴于在生成式任务和常识推理任务中的表现，我们希望将这些预训练的encoder-decoder语言模型作为开发进一步规划方法的基础，并希望它们所学到的通用常识能够落地到具体的实际应用场景中。由于G-PlanET也可以被视为一个文本生成问题，因此使用这些大型语言模型作为我们发展规划方法的骨干，并希望它们所具备的常识能够与现实世界中的具体情境相结合，为具体的智能体任务提供更加强大和高效的规划能力。

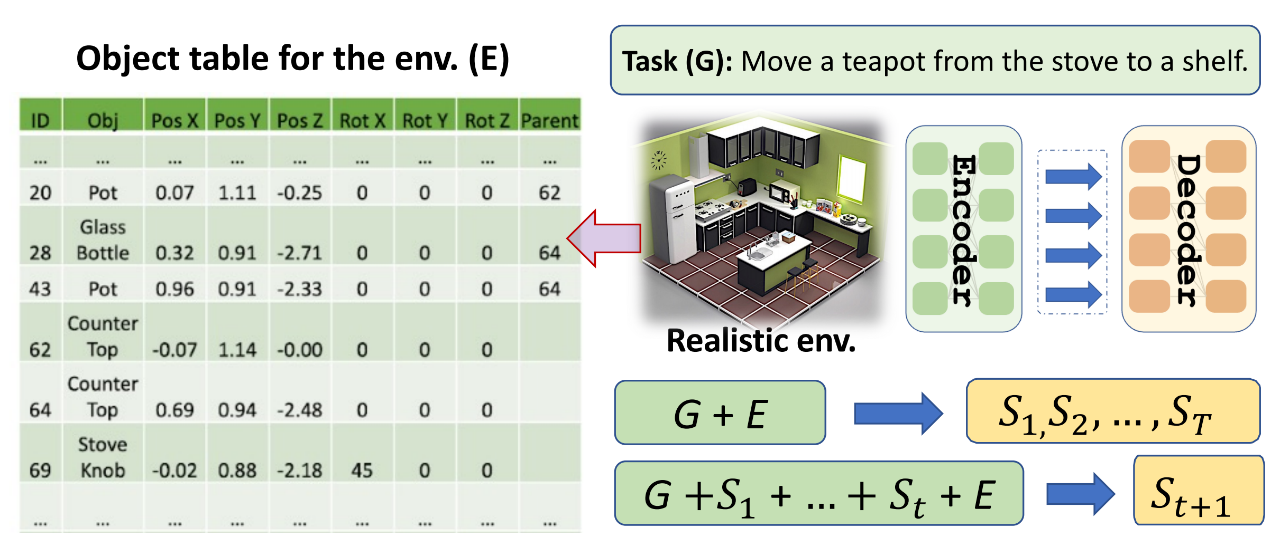
总之，这些预训练的encoder-decoder语言模型具有在生成式任务和常识推理任务中表现出色的优秀性能，展现出在自然语言处理领域的广泛应用前景。我们将使用这些模型作为我们发展规划方法的基础，以期望它们所具备的通用常识能够为实际应用场景中的具体智能体任务提供更加强大和高效的规划能力。

图 2：主要方法图示

上图表示了其中的大概流程，后文将详细介绍其中所使用的模型和实现细节。

## 基础模型

预训练语言模型在大量无监督的语料库上进行预训练，可以学习到自然语言的内在规律和特征，从而使得在各种自然语言处理任务上都能表现出色。针对不同的任务和应用场景，也涌现出了许多不同类型的预训练语言模型，这些语言模型都在各自的领域内具有卓越的表现和广泛的应用。现代常用的序列到序列（sequence to sequence）生成式模型包括T5[31]和BART[32]，由于G-PlanET也可以看作是文本生成问题，因此我们使用这些大型语言模型作为开发进一步规划方法的支撑，希望它们的常识能够在真实世界的情境中为具体任务提供支持。

正如许多论文所展示的那样，当模型参数量相似时，BART和T5在许多生成任务上表现出可比性。因此，我们选择了BART-base和BART-large两个预训练模型以及其变体进行评估。后续将

### BART

BART (Bidirectional and Auto-Regressive Transformers)是由Facebook AI Research于2020年提出的一种预训练的编码器-解码器模型它被设计为一个去噪自编码器，将一个被破坏的文档映射到其原始文档。BART模型的编码器采用双向的Transformer[33]结构，对破坏后的文本进行编码，解码器采用单向的Transformer结构，对原始文档进行解码。在预训练阶段，BART采用最大似然估计来优化模型，即通过最小化原始文档的负对数似然来学习模型参数。

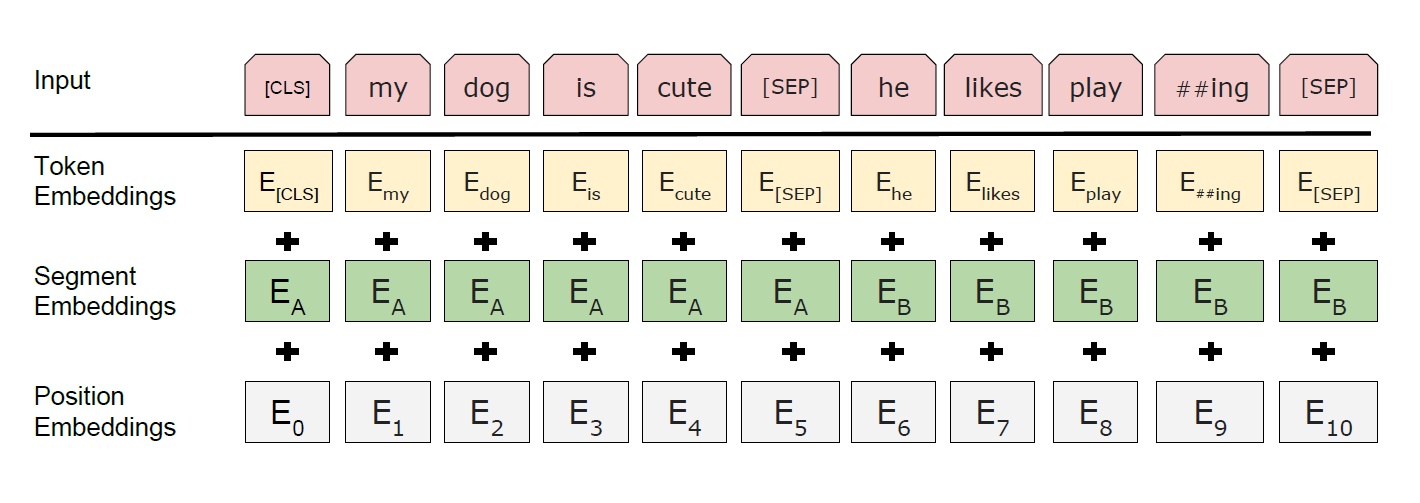
相较于其他的语言模型，BART采用了多种噪声引入方式，例如单词掩码，单词删除，句子排列变换，文档旋转变换和文本填充等。

图 3：BART对输入文本的分布式表示

在进行序列到序列的生成过程中BART需要经过如下步骤

1. **输入编码**：如图3所示，BART的文本表示采用由三种不同的表示相加得到，分别是词嵌入向量（Token Embedding），分段嵌入向量（Segment Embedding）和可学习的相对位置向量（Position Embedding）。
2. **编码器（encoder）：**通过多个相同的块，每个块包括一个自注意力层和一个前馈神经网络，前者用于计算各个位置相互影响后的表示，后者总结总体特征。
3. **解码器（decoder）：**与编码器类似，但输入包括了前一时间步的输出和编码器最后一层的表示。
4. **输出编码：**解码器的输出会被送到输出编码器中，由嵌入和前向神经网络模块组成，最终输出一个概率分布，代表目标序列中下一个符号的概率。

### TAPEX

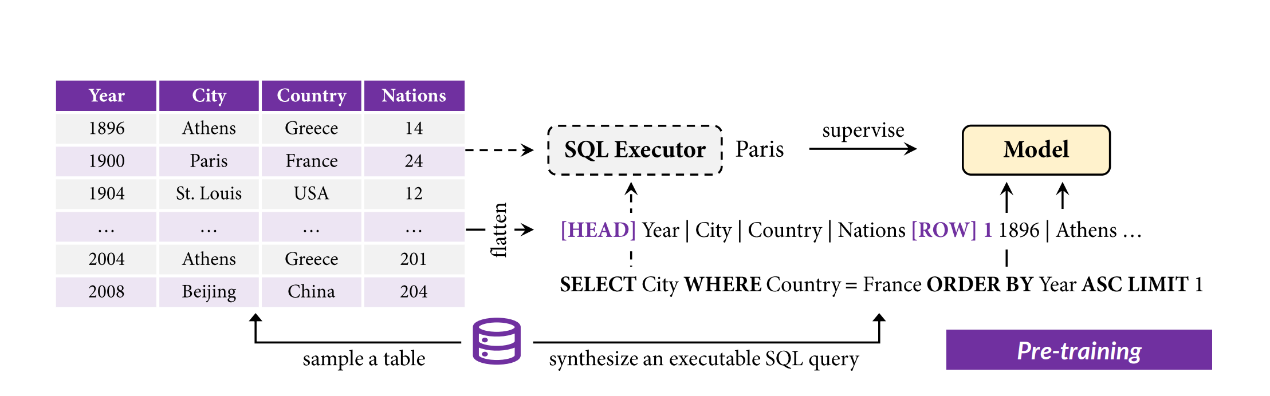
TAPEX（Table Pretraining via Execution）[34]是一种在原有预训练的BART基础上进行二次预训练之后的版本，以增加模型在表格类型数据上的性能。

图 4：TAPEX预训练流程图

图4展示了TAPEX二次预训练的过程，在自然语言预训练任务中，常见的方法是在通过掩码语言模型（MLM）任务后，使用表格重构任务进行表格预训练。这些任务通常将损坏的表格和自然语言句子作为输入，并尝试恢复损坏的部分，以加强自然语言句子和表格之间的联系。虽然这些预训练任务表现良好，但它们往往效率较低，因为它们通常需要极大的预训练语料库。

为了设计更有效率的表格预训练任务，TAPEX作者认为关键在于表格的可执行性。也就是说，结构化表格使我们可以通过编程语言（如SQL查询）对它们执行离散操作，而非结构化文本则不行。TAPEX采用SQL执行作为唯一的预训练任务。如图4所示，TAPEX的预训练过程类似于预训练模型的生成微调过程。给定一个可执行的SQL查询和一个表T，TAPEX首先将SQL查询和扁平化的表T\*拼接起来，然后将其输入模型编码器。然后，TAPEX通过现成的SQL执行器（例如MySQL）获得查询的执行结果，作为模型解码器的期望输出。直观地说，预训练的过程就是要鼓励语言模型成为一个神经SQL执行器。TAPEX作者相信，如果一个语言模型能够被训练成能够忠实地“执行”SQL查询并生成正确结果，那么它应该对表格有深刻的理解。TAPEX的表格来自于WikiTableQuestion[35]数据集，sql查询语句（query）根据规则自动生成。

### GPT-J和In-context Learning

GPT（Generative Pre-trained Transformer）是一种预训练的transformer模型，其中数据量较大的GPT会获得少样本（few-shot）推理能力，其中最显著的就是上下文学习（ICL，In-Context Learning）。ICL指的是语言模型学习少量演示输入-标签对的能力，并利用这些知识在没有额外参数更新的情况下预测未见过的输入。这种能力特别引人注目，因为它使语言模型能够在少量标记数据可用的情况下表现出色。例如，仅给出几个摘要文本的示例，语言模型就可以学习高准确性地摘要未见过的文本。

GPT-J 是一种大规模预训练语言模型，由 EleutherAI 社区开发。它是建立在 GPT-3 模型上的变种，由于GPT-3尚未开源且成本较高，本文采用同样具有ICL能力的GPT-J进行测试。

## 详细设计与实现

### 表格化环境

为了使语言模型能够感知环境，我们需要对第2.2.2节中描述的对象表进行编码。参考先前在基于表格的自然语言处理任务中的工作，我们按行将表格展开为文本序列，从而创建对象表的线性化版本。然后，我们将展开后的表格附加到目标后面，形成完整的输入序列。这样，编码器-解码器的输入端最终具有生成基于环境的计划所需的环境信息。

考虑到序列长度的限制，我们只选择通过对象的类型、位置、旋转和容器父级来进行编码。对象类型不仅告诉我们对象的是什么，还隐含着通识的可行性（例如，微波炉可以加热物体，刀子可以切割物体），这对规划非常重要。这一常识再预训练过程中被交给预训练语言模型，位置信息对于智能体导航和寻找物体至关重要，因此在规划中发挥着重要的作用。旋转对于某些只能以特定方向使用的物体也很有用（例如，只有当智能体在冰箱前面时才能打开冰箱）。物体的容器和本身有着密切的空间联系（例如，笔在桌子上，苹果在冰箱里）。每个对象都有一个唯一的标识符，因此当它们成为其他对象的容器时，可以精确地引用相同类型的对象。此外，智能体被表示为一个特殊的对象。

### 迭代式生成

我们将物体信息的展平表格加入输入序列确实提高了LM在感知真实环境方面的表现，这是基于有根规划的基础。然而，思考过程仍然受限于传统的seq2seq学习框架，该框架假设LM应该通过单次解码输出完整的计划。我们认为，周密的规划过程应该仔细处理每个步骤的连贯性，否则错误将累积并导致计划失败。

因此，我们提出了一种简单而有效的解码策略，学习逐步生成计划。具体而言，我们将先前生成的步骤附加到当前步骤$t$的输入序列中（即\texttt{Input} = $[G + S\_1 + \cdots + S\_t (+ E)]$），以生成下一步（即\texttt{Output} = $S\_{t+1}$）。这个迭代解码过程将一直持续到LM生成特殊标记\texttt{END}。在训练阶段，我们使用$S\_{\le t}$的真实参考；在推理阶段，我们没有这样的参考，因此我们使用模型预测作为$S\_{\le t}$。

值得注意的是，与传统的seq2seq学习过程不同，迭代解码策略需要运行编码器-解码器模型$N+1$次才能生成具有$N$个步骤的计划。重新编码的额外计算成本是值得的。想象一下当我们人类在房间中计划任务时。我们很自然地逐步提出计划，很可能生成不同步骤最有用的信息是关于不同的对象。因此，一种时间动态的注意机制在LM中进行规划是有利的。我们的迭代解码策略鼓励编码器-解码器架构学习这种能力。

# 实验验证与效果分析

## 数据集

## 评价标准

## 实验结果

## 实验分析

# 结论

哈哈哈哈哈[6]

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

表 1哈哈哈哈

# 参考文献

1. Bhargava, P., & Ng, V. (2022). Commonsense Knowledge Reasoning and Generation with Pre-trained Language Models: A Survey. AAAI Conference on Artificial Intelligence.
2. Liu, H., & Singh, P. (2004). Commonsense Reasoning in and Over Natural Language. International Conference on Knowledge-Based Intelligent Information & Engineering Systems.
3. Zang, L., Cao, C., Cao, Y., Wu, Y., & Cao, C. (2013). A Survey of Commonsense Knowledge Acquisition. Journal of Computer Science and Technology, 28, 689 - 719
4. Tincoff, R., & Jusczyk, P.W. (1999). Some Beginnings of Word Comprehension in 6-Month-Olds. *Psychological Science, 10*, 172 - 175.
5. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H., & Riedel, S. (2019). Language Models as Knowledge Bases? *ArXiv, abs/1909.01066*.
6. Rajani, N., McCann, B., Xiong, C., & Socher, R. (2019). Explain Yourself! Leveraging Language Models for Commonsense Reasoning. Annual Meeting of the Association for Computational Linguistics.
7. Xu, F.F., Lin, B., & Zhu, K.Q. (2017). Automatic Extraction of Commonsense LocatedNear Knowledge. Annual Meeting of the Association for Computational Linguistics.
8. Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. Conference on Empirical Methods in Natural Language Processing.
9. Lin, B., Xu, F.F., Zhu, K.Q., & Hwang, S. (2018). Mining Cross-Cultural Differences and Similarities in Social Media. Annual Meeting of the Association for Computational Linguistics.
10. Shridhar, M., Yuan, X., Côté, M., Bisk, Y., Trischler, A., & Hausknecht, M.J. (2020). ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. ArXiv, abs/2010.03768
11. Wang, R., Jansen, P.A., Côté, M., & Ammanabrolu, P. (2022). ScienceWorld: Is your Agent Smarter than a 5th Grader? Conference on Empirical Methods in Natural Language Processing.
12. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R.J., Jeffrey, K., Jesmonth, S., Joshi, N.J., Julian, R.C., Kalashnikov, D., Kuang, Y., Lee, K., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D.M., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., & Yan, M. (2022). Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. ArXiv, abs/2204.01691.
13. Olmo, A., Sreedharan, S., & Kambhampati, S. (2021). GPT3-to-plan: Extracting plans from text using GPT-3. ArXiv, abs/2106.07131.
14. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Annual Meeting of the Association for Computational Linguistics
15. Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ArXiv, abs/1910.10683.
16. Talmor, A., Herzig, J., Lourie, N., & Berant, J. (2019). CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. ArXiv, abs/1811.00937
17. Arik, S.Ö., & Pfister, T. (2019). TabNet: Attentive Interpretable Tabular Learning. ArXiv, abs/1908.07442.
18. Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., & Eisenschlos, J.M. (2020). TaPas: Weakly Supervised Table Parsing via Pre-training. ArXiv, abs/2004.02349.
19. Liu, Q., Chen, B., Guo, J., Lin, Z., & Lou, J. (2021). TAPEX: Table Pre-training via Learning a Neural SQL Executor. ArXiv, abs/2107.07653.
20. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., & Fox, D. (2019). ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10737-10746.
21. Min, S., Chaplot, D.S., Ravikumar, P., Bisk, Y., & Salakhutdinov, R. (2021). FILM: Following Instructions in Language with Modular Methods. ArXiv, abs/2110.07342
22. Liu, H., Liu, Y., He, H., & Yang, H. (2022). LEBP - Language Expectation & Binding Policy: A Two-Stream Framework for Embodied Vision-and-Language Interaction Task Learning Agents. ArXiv, abs/2203.04637.
23. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A.K., & Farhadi, A. (2017). AI2-THOR: An Interactive 3D Environment for Visual AI. *ArXiv, abs/1712.05474*.
24. Huang, W., Abbeel, P., Pathak, D., & Mordatch, I. (2022). Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. *ArXiv, abs/2201.07207*.
25. Brown, P.F., Pietra, V.J., Souza, P.V., Lai, J.C., & Mercer, R.L. (1992). Class-Based n-gram Models of Natural Language. *Comput. Linguistics, 18*, 467-479.
26. Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*.
27. Pennington, J., Socher, R., & Manning, C.D. (2014). GloVe: Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing*.
28. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *North American Chapter of the Association for Computational Linguistics*.
29. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*, 1735-1780.
30. Talmor, A., Herzig, J., Lourie, N., & Berant, J. (2019). CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. *ArXiv, abs/1811.00937*.
31. Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv, abs/1910.10683*.
32. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *Annual Meeting of the Association for Computational Linguistics*.
33. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *NIPS*.
34. Liu, Q., Chen, B., Guo, J., Lin, Z., & Lou, J. (2021). TAPEX: Table Pre-training via Learning a Neural SQL Executor. *ArXiv, abs/2107.07653*.
35. Pasupat, P., & Liang, P. (2015). Compositional Semantic Parsing on Semi-Structured Tables. *Annual Meeting of the Association for Computational Linguistics*.

# 致谢

本文的是在李四导师的指导下完成的，从开题到构思、包括最后提炼，都得到了李老师的细心指点和提携。李老师对写作和科研的认真与执着给我留下了十分深刻的印象，同时也给予了我莫大的帮助，仅在此献上我最真挚的敬佩与感激。

同时也要在此感谢帮助和支持我的父母，他们辛苦了。