



深圳技术大学
SHENZHEN TECHNOLOGY UNIVERSITY

本科毕业论文（设计）

题目：大语言模型评估平台
的对比和评测

姓 名	黄思研
学 院	大数据与互联网学院
专 业	计算机科学与技术
学 号	202001020107
指 导 教 师	黄炳顶
职 称	特聘教授
提 交 日 期	2024 年 5 月 17 日

深圳技术大学本科毕业论文（设计）诚信声明

本人郑重声明：所呈交的毕业论文（设计），题目《大语言模型评估平台的对比和评测》是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期： 年 月 日

目录

摘要（关键词）	I
1. 绪论	1
1.1 课题研究背景及意义	1
1.2 课题基础及国内外研究进展	3
1.3 课题主要研究内容	4
2. 大语言模型基本原理与“幻觉”成因	6
2.1 大语言模型基本原理	6
2.1.1 Transformer 的基本原理	6
2.1.2 GPT 的类别与发展	7
2.1.3 GPT 到 ChatGPT	10
2.2 大语言模型“幻觉”原因	10
2.2.1 “幻觉”的定义	10
2.2.2 “幻觉”原因来源分析	11
3. 大语言模型评估数据集与评估方法	13
3.1 大语言模型评估数据集概述	13
3.1.1 大语言模型评估数据集定义	13
3.1.2 大语言模型评估数据集制作常见问题	13
3.1.3 大语言模型制作评估数据集常见形式	13
3.1.4 大语言模型评估数据集题型	14
3.2 大语言模型评估方法概述	16

3.2.1 大语言模型评估方法定义	16
3.2.2 大语言模型评估方法介绍	16
3.3 评估数据集的格式化保存	18
3.3.1 评估数据集常见格式	18
3.3.2 评估数据集整体介绍	19
3.3.3 统一数据集格式	20
3.4 评估数据集重分类和清洗	23
3.4.1 评估数据集重分类目的	23
3.4.2 评估数据集重分类方法与流程	24
3.4.3 评估数据集数据清理	25
4. 常见开源大语言模型部署与评估流程	27
4.1 常见的开源大语言模型	27
4.2 开源大语言模型部署	27
4.3 大语言模型评估流程	28
4.3.1 大语言模型评估实现	28
4.3.2 大语言模型评估流程如下	29
4.4 大语言模型评估代码	30
4.5 评估结果整体介绍	31
4.5.1 评估结果展示与分析	35
5. 总结与展望	41
5.1 全文总结	41

5.2 工作展望	42
参考文献	43
致谢	46

大语言模型评估平台的对比与评测

【摘要】大语言模型近些年如雨后春笋般在国内外涌现，但是大语言模型本身存在着不稳定性和“幻觉”，如何对其进行评估从而验证模型可靠性成为一个重要问题。现阶段主流的评估方案基本都是通过雇佣专家或收集面向人类的考试内容制作评估数据集建立评估平台。但是各个不同的评估平台都有自己独特的格式和评估方式，且每个评估数据集在分类上都不相同，这无疑提高了评估和改进大语言模型的难度。本文理解大语言模型基本原理上，部署常见开源大语言模型。同时收集了六大常见的评估平台 and 对应数据集，提出了可以统一各个评估数据集的 JSON 结构并对各个评估数据集进行重分类，将处理好后的评估数据集保存为 JSON 格式文件。在统一完评估数据集结构后，利用 python 脚本和对应的评估提示词，调用大语言模型 API 实现大语言模型在服务器上的自动化评估并记录评估结果。最后，利用 python 中的 matplotlib.pyplot 库，对评估结果进行可视化并分析评估结果，最终实现高可扩展性的自动化评估多数据集的大语言模型平台。在评估题目的数量上，本文收集和统一的评估题目数量上达到了万级，评估结果具备一定的可靠性，评估结果清晰的表明了不同大语言模型知识面的差距，以及某个大语言模型在学科领域和语言方面等偏好。

【关键词】大语言模型；评估数据集；评估平台；自动化评估脚本；NLP

Comparison and Review of Large Language Model Evaluation Platforms

【Abstract】 Big Language Models (BLMs) have been springing up at home and abroad in recent years, but BLMs are inherently unstable and “illusory”, so how to evaluate them and verify their reliability has become an important issue. At this stage, the mainstream assessment programs are basically to build assessment platforms by hiring experts or collecting human-oriented test contents to make assessment data sets. However, different assessment platforms have their own unique formats and assessment methods, and each assessment dataset is different in classification, which undoubtedly increases the difficulty of evaluating and improving the large language model. In this paper, we understand the basic principles of large language models on deploying common open source large language models. At the same time, we collect six common evaluation platforms and corresponding datasets, propose a JSON structure that can unify each evaluation dataset and reclassify each evaluation dataset, and save the processed evaluation dataset as a JSON format file. After unifying the structure of the evaluation datasets, the python script and the corresponding evaluation prompts are used to call the API of the big language model to realize the automated evaluation of the big language model on the server and record the evaluation results. Finally, using the matplotlib.pyplot library in python, we visualize the assessment results and analyze the assessment results, and finally realize the highly scalable automated assessment of multi-dataset big language model platform. In terms of the number of assessment topics, the number of assessment topics collected and unified in this paper reaches 10,000, and the assessment results have a certain degree of reliability, and the assessment results also clearly show the gap in the knowledge of different big language models, as well as the preference of a certain big language model in terms of subject areas and languages, and so on.

【Key words】 Large Language Model; Assessment Datasets; Assessment Platforms; Automated Assessment Scripts; NLP

1. 绪论

1.1 课题研究背景及意义

大语言模型，英文名 Large Language Model，常缩写为 LLM，是一种基于深度学习的人工智能模型^[1]。这种模型的主要目标是理解和生成人类语言。为了实现这个目标，大语言模型需要在大量的文本数据上进行训练，以学习自然语言的规则、语法和语义。大型语言模型通常使用神经网络来实现，其中最常见的是循环神经网络 RNN^[2]和 Transformer^[3]。

该类模型的训练过程通常分为两个阶段：预训练和微调。在预训练阶段，LLM 使用大量的无标签文本数据进行训练。这个阶段的目标是让模型学习语言的基本规则和模式。预训练的核心是让模型预测文本中的下一个词或字符。在这个过程中，模型会学习到词与词之间的关联性、句子的结构以及语言的语法和语义规则。预训练通常使用的是自监督学习的方法，即模型自己生成训练样本并进行训练。微调阶段，LLM 使用有标签的数据进行进一步的训练。这个阶段的目标是让模型适应特定的任务或领域。微调的核心是让模型根据任务的标签来调整自己的参数，使得模型在这个任务上表现更好。微调通常使用的是监督学习的方法，即模型根据标签进行训练。以上两个过程的目的都是为了使 LLM 可以更好的理解和生成自然语言^[4]。

但是大型语言模型（LLM）在快速发展的同时也面临着一些问题。首先，LLM 存在不稳定性^[5]，即对于相同的提示词，其回答可能存在前后不一致的情况。这主要是由于 LLM 的预测结果受到模型参数和输入数据的影响，而这些因素在不同的情况下可能会发生变化，从而导致模型的预测结果不一致。其次，LLM 仍然具有一定的局限性，会出现“幻觉”事实并犯推理错误。这主要是由于 LLM 的训练数据是基于大量的文本数据，而这些文本数据可能包含错误、不准确或不完整的信息，从而导致 LLM 在预测时出现错误。最后，LLM 在模型训练、线上推理服务、安全测试等方面需要持续加强安全合规能力。这主要是由于 LLM 的应用场景涉及到用户的隐私信息和敏感信息，因此需要采取一系列的安全措施来保护用户的数据安全。同时，市场需求不断变化，LLM 的功能不断地得到深度挖掘，如今

在各个领域的内容创作和行业细分领域深度结合的应用越来越多。市场对于 LLM 的需求也随着用户所在行业的变化而不断变化。因此，LLM 评估测试集和平台对于检测、优化、解决 LLM 的不稳定性与个性化微调 LLM 以满足市场需求的不断变化至关重要。

为了更好地解决大型语言模型（LLM）的“幻觉”问题，同时也为了更好地开发和应用大型模型，就需要有一套完善的大型语言模型评测体系。大型语言模型产品的初期评测阶段，主要基于小规模问题集进行评测^[6]。然而，这种方式也存在一些明显的缺陷。首先，小规模问题集的覆盖范围有限。这种问题集通常只包含少量的样本，无法全面覆盖大型语言模型的各种应用场景和任务类型。由于大型语言模型的训练数据是基于大量的文本数据，而这些文本数据可能包含错误、不准确或不完整的信息，从而导致 LLM 在预测时出现错误。这种评测方式在初期对于了解大型语言模型产品的基本性能，比如知识点全面性和推理速度上具有一定的价值。然而，随着时间的推移和技术的进步，众多厂商开始根据测评题目进行定向优化。通过针对测评数据集特定调整，以获取相对较好的排名。最后，小规模问题集评测可能无法满足大型语言模型的下游任务的需求。大型语言模型的评估不仅仅是一个终点，而是一个连续的过程。因此，小规模问题集评测可能无法满足这一需求。为了解决这些问题，研究人员正在不断改进 LLM 的训练算法和模型结构，以提高模型的效率和性能。同时，研究人员还在研究如何保护用户的数据隐私，以确保 LLM 在应用中的安全性和可信度。

大语言模型评测方法难以跟进目前大语言模型推出的速度，因此市面上的大语言模型评估平台开始逐年增加，鉴于不同平台的评估数据集的侧重点不同以及大语言模型本身存在的不稳定性，对比不同评估平台和对应的评估数据集在不同大语言模型和任务的表现是很有必要的。

大型语言模型（LLM）的评估平台是近期研究和开发的热门话题，市面上不同侧重的评估方法也较为混杂。因此，本文注重于收集和整理大型语言模型评估平台上的数据集，对比不同评估平台所对应的数据集在不同任务上的利弊，以便用户可以自由选择 and 评估大型语言模型，更好地满足用户需求。

1.2 课题基础及国内外研究进展

在大模型方面,据 2023 年 11 月新华社研究院中国企业发展研究中心的大语言模型报告显示,自 2022 年 11 月 30 日 ChatGPT 发布以来,AI 大模型在全球范围内掀起了有史以来规模最大的人工智能浪潮。同时国外仅过了三个月就推出了 GPT4,参数规模大幅增长。但在国内学术和产业界在过去一年也有了实质性的突破。大致可以分为三个阶段,即准备期(ChatGPT 发布后国内产学研迅速形成大模型共识)、成长期(国内大模型数量和质量开始逐渐增长)、爆发期(各行各业开源闭源大模型层出不穷,形成百模大战的竞争态势)。同时,在 2023 年 7 月到 12 月,仅过去半年,国内领军大模型企业实现了大模型代际追赶的奇迹,从 7 月份与 GPT3.5 的 20 分差距,每个月都有稳定、巨大的提升,到 11 月份测评时已经完成总分上对 GPT3.5 的超越。GPT3.5 和 GPT4 在中文上的表现情况基本一致,在 11 月份测评结果中显示,在中文能力都有一定的下滑,而国内头部模型则展现了继续稳健提升的能力。在 12 月份的测评结果中可以看到,国内第一梯队模型与 GPT4 的差距在缩小。但仍有较大的距离需要追赶。百度也在 2023 年 6 月发布了文心一言 3.5,仅四个月后又推出了文心一言 4.0,实现了基础模型的全面升级。其他厂商的产品也在不断升级迭代。

随着国内国外都开始针对大模型进行提升和发展,大模型测评技术也开始逐步被重视。无论是国内还是国外,大型语言模型的评估方法都是在大型语言模型的发展过程中逐步演变,通常可以划分为三个关键阶段:基于规则阶段、基于统计学阶段和基于神经网络阶段^[7]。

在基于规则的阶段,评估主要依赖于对语言模型在特定任务上表现的分析,其中包括语法正确性、词汇使用、句法分析准确性和信息提取效果等方面。早期的评估方法主要基于人工规则和语言学知识,例如 Roche 在 1997 年提出的有限状态自动机方法就是其中之一^[8],它对自然语言处理中的模型进行了评估和比较。而在之后,基于统计学的评估方法逐渐兴起,使得语言模型的评估结果得以量化,并在困惑度、准确率、召回率、语言模型生成、词性标注评估、数据集划分和交叉验证等方面进行了创新和改进。例如,Goodman 在 2001 年提出了在 N-gram 语言模型中应用平滑技术^[9],并将困惑度作为评估模型性能的指标,这一方法使得

评估更具客观性和可量化性。最后到了 GPT 的诞生，随着基于神经网络的语言模型的崛起，评估方法也随之发生了变化。在这一阶段，评估方法主要体现在人工评估文本生成任务中的质量评估以及专家或志愿者对模型生成文本正确性的评估等方面。例如，Radford 在 2018 年提出了对 GPT 模型性能的评估方法，重点关注模型在语言建模任务和下游自然语言处理任务上的表现^[10]，这使得评估更加全面和多维度。在下游任务上选择正确的评估指标可以更直接地体现模型在实际应用中的效果。虽然该篇论文主要集中于展示预训练模型相对于传统方法的性能提升，但其评估方法也对后续的大语言模型开发和评估都展现了不小的启示。但是目前基于神经网络就行大语言模型评估方式可靠性欠佳，因此目前该阶段更多属于探索阶段。

尽管现有的评估平台取得了一定成就，但仍然存在一些挑战和限制。首先，传统的基于规则和统计学的评估方法往往缺乏对模型深层语义理解和生成能力的全面评估。其次，大型语言模型的复杂性和规模使得现有的评估方法往往需要大量的人力和时间成本，且不够高效。此外，大型语言模型的特性使得其评估更加复杂，涉及到模型的语义理解、逻辑推理、文本生成等多个方面，需要更加全面和多维度的评价数据集对其进行评估。

因此，为了更好地评估大型语言模型，需要不断改进和创新评估方法。创建和改进要基于现有的研究而非凭空捏造，因此本毕业设计旨在收集大语言模型的评估平台中的数据集，学习评估原理并整合形成较为全面的评估方法和结果。

1.3 课题主要研究内容

本文的主要研究内容是研究大语言模型的发展历程和基本原理，收集并整合目前大语言模型评估平台使用较多或较为经典的数据集。通过各种评估平台所对应的评估数据集来对市面上较为主流的大语言模型进行评估，测试大语言模型的基本逻辑能力、创造能力、知识的全面性与正确性和上下文联系能力。从而达到利用已标定好的数据测试大语言模型，并且可以针对大语言模型的“胡说”与“幻觉”等问题做集中评估，并有利于使用者将来更好的改进或微调大模型，从而达到建立一套较为完整的大语言模型评估体系的目的。本文的主要研究内容如下：

1、大语言模型原理与发展历史并初步收集评估平台

研究大语言模型的原理，包括大语言模型的基石 Transformer 和 GPT 的发展历史。通过分析大语言模型的原理来研究大语言模型基本框架和结构。根据 GPT 论文推断大语言模型幻觉出现的原因。并初步了解和收集常见的评估平台上的评估数据集的收集、制作和评估方式。初步建立大语言模型评估方案。

2、统一评估平台数据集格式并部署大语言模型

利用 python 中的 Panda、Json 库，针对不同的评估数据集的特点和存储模式，读取并重新统一数据集的格式后再重新进行保存，方便后续编写一个 python 脚本对不同的评估平台数据集进行自动化评估。同时在 Hugging face 中下载 ChatGLM 并根据其中的官方手册对大语言模型进行部署。

3、编写脚本实现自动化评估并记录评估结果

编写 python 脚本调用大语言模型 API，同时编写提示词函数，实现大语言模型的多平台自动化评估，同时记录评估结果方便后续对评估结果进行可视化和分析。

4、评估结果可视化和分析

根据评估结果计算大语言模型在各个评估平台数据集上的正确率。同时根据评估数据绘制直方图、折线图和柱状图等，同时分析 ChatGLM 的一代至三代在评估平台上的表现，观察 ChatGLM 性能方面的变化。

2. 大语言模型基本原理与“幻觉”成因

2.1 大语言模型基本原理

2.1.1 Transformer 的基本原理

在大语言模型出现之前，Transformer 就凭借其自注意力机制、多头注意力机制、前馈神经网络与位置编码技术在 NLP 领域大放异彩。以至于后续的 NLP 模型大多都属于 NLP 的变体，比如 BERT^[11]、XLNet^[12]和今天的主角——GPT。因此在介绍 LLM 之前，需要先对 Transformer 有一个简单了解。

Transformer 属于简单的序列转录模型，仅仅使用纯注意力实现训练速度较快的基础模型。相对于 RNN 的利用隐藏层不断把先前的历史信息 and 当前读取到的词作为输入得到当前输入的隐藏状态并以此往复，从而实现 RNN 处理时序信息的能力^[2]；与 RNN 相比，Transformer 不再使用先前使用的循环神经层，将其替换为自注意力机制，且有很高的并行度。在 Transformer 当中，其架构属于编码器和解码器架构，编码器会将一个句子的每个词 ($x_1, x_2 \cdots x_n$) 编码为 ($z_1, z_2 \cdots z_n$) 但是 z 为对应的 x 的向量表示；而解码器以编码器为输入，输出主表示为 ($y_1 \cdots y_m$)，且输出只能以自回归的方式一个一个生成。在编码器和解码器间用残差网络联系在一起，且子层中都有一个归一化层 (Layer Normalization)。通过编码器的多头注意力和全连接前馈网络和解码器中的线性层与 softmax，将解码器输出转换为预测的下一个词的概率分布，最终实现高效的端到端处理自然语言序列模型^[3]。但是相比于 RNN，Transformer 的 attention 对模型做的假设更少，因此训练 Transformer 需要更多的数据和模型才可达到更好的效果。这可以解释现在的大语言模型为何越来越大的原因。Transformer 基本原理和最初作用如图 2-1 所示：

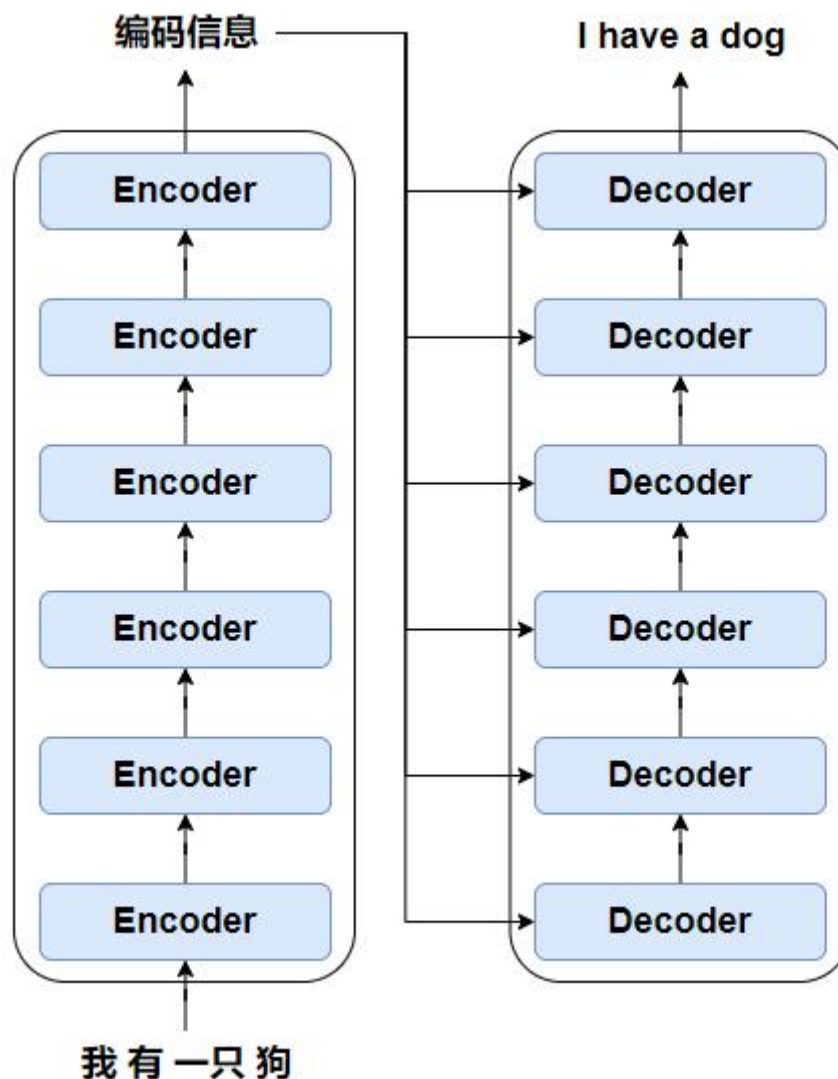


图 2-1: Transformer 基本原理与翻译任务举例

2.1.2 GPT 的类别与发展

从核心技术上来讲，GPT-1（Generative Pre-Training）是将 Transformer 的解码器拿出来，在没有标号的文本上训练一个语言模型作为预训练。并且用他在多个 NLP 子任务上做微调，得到每一个任务所要的分类器，最终统合为 GPT-1 的整体模型^[4]。在 GPT 诞生之初首要就是为了解决 NLP 中大量不同的任务，且拥有很多大量没有标号的文本，但是标号类型的数据是非常少的。如果只是在标好的数据上训练出分辨模型，则会局限于 Transformer 本身局限性会变得非常难以训练。而 GPT 对此的解决方案是现在一个没有进行标号的数据上训练一个训练模型作为预训练模型，之后再在有标号的 NLP 子任务上训练一个分辨的微调模型。

这种模型迁移的方法 在计算机视觉上较为常用，但在 NLP 领域一直没有流行。主要原因是 NLP 领域没有像 ImageNet 规模如此之大且附带标签的数据集，并且在视觉领域也有图片数据增强的概念，此外图片信息量比一个句子要大得多。这种 NLP 本身存在的局限性也导致了 NLP 的发展缓慢的问题，直到现在的 GPT-1 和 BERT 才将这些未标注的句子利用起来。

学习类似视觉领域的迁移学习，如图 2-2，利用起 NLP 中大量的未标注文本数据是 GPT-1 中最大的创新点之一。

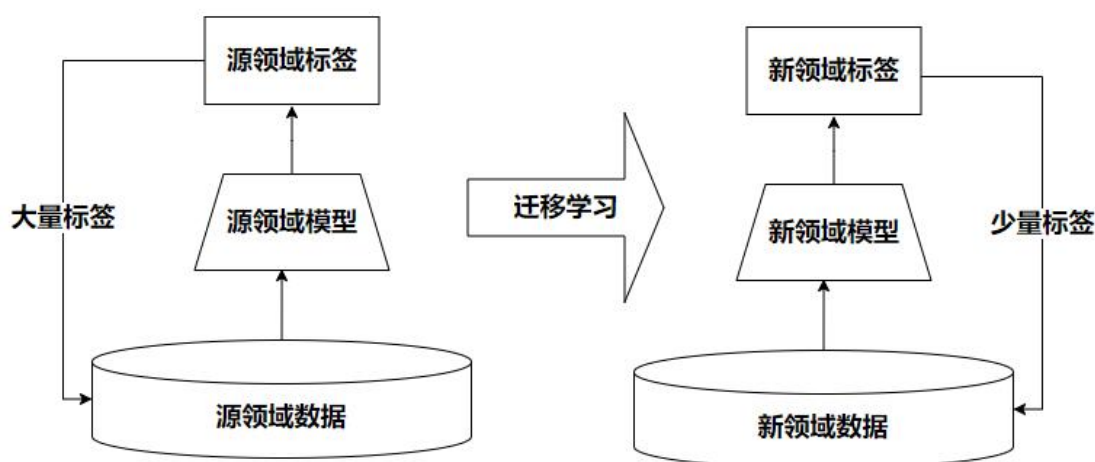


图 2-2: 视觉迁移学习基本原理

但是在对待没有标号的文本中有两大问题，一是没有一种统一的目标函数；二是如何有效的把学习到的文本表示传递到下游的子任务。而 GPT-1 则是利用多个目标函数解决前者，后者的解决方式则是利用无标注数据结合极大似然估计函数，具体公式如下：

$$L(\theta) = \prod_{i=1}^n f(x_i; \theta)$$

其中公式表示对所有观察到的样本数据 x_1, x_2, \dots, x_n 的概率密度函数连乘得到的似然函数。其中， x_i 表示第 i 个观察到的样本数据。因此，在极大似然估计中，我们通过最大化似然函数 $L(\theta)$ 来估计参数 θ ，使得观察到的数据出现的概率最大化。这样得到的估计值 θ 即为极大似然估计值。

与前文提到的 Transformer 解码器给定前 $i-k$ 到第 $i-1$ 个词(k 为窗口大小)，

同时预测第 i 个词出现的概率，这种方式对比于 BERT 的完型填空来说更难，因此也需要一个更为强大的模型；对不同的下游任务进行微调，比如分类、蕴含、相似和问答任务。GPT-1 的方法是将这些任务统一表示为一个序列，如图 2-3 所展示的三大 NLP 子任务。并且输入到一个 Transformer 结构中，最后利用线性层投影到该子类任务所需要的结果，这样就可以保证即使有多种不一样的数据但仍然可以维持 GPT-1 去进行微调，同时这也解释了 GPT-1 为什么能适应如此多的 NLP 任务，且在进行评估时，我们也可以利用类似的方法处理和评估这些子任务表示。

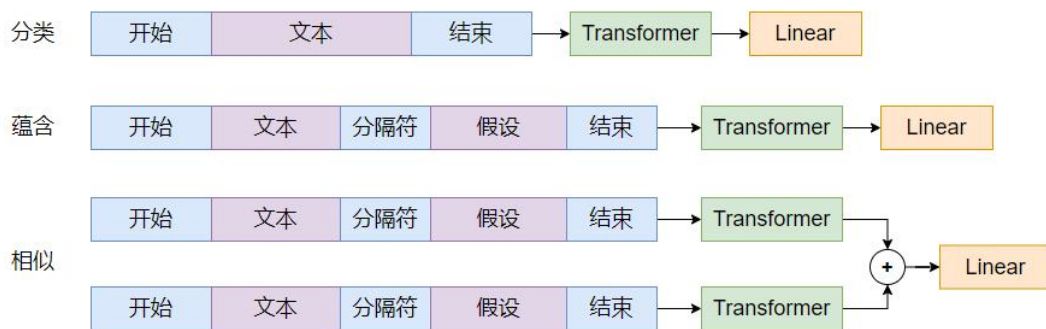


图 2-3: GPT-1 进行多任务微调原理

到了 GPT-2^[13]，他的优化不仅体现在模型的量级，还提出了 zero-shot 这个非常重要的概念。Zero-Shot 的设定主要运用在 GPT-2 的下游模型，其本质概念是指在做到下游任务时，不需要下游任务的标注信息，也不需要训练自身模型。因此在处理下游任务时，需要把输入视为自然语言，以请求和提示词的形式输入到模型当中。即 GPT-2 并没有微调的步骤，其本质是通过大量的无监督学习，让模型具有广泛的知识能力。但是我们仍然可以用小部分已标注数据进行微调来执行特定任务。

到了 GPT-3^[14]上，由于 GPT-3 模型本身存在大量参数，它所追求的是在微调上不做梯度更新，因为模型本身比较大，如果针对每个新任务都进行微调那么模型的应用难度就会变得比较高。另外 GPT-3 所用的数据集本身就足够大，子任务数据集可能被训练数据集所包含，若再进行梯度更新则可能导致过拟合，并没有很好的体现模型的泛化能力。在 GPT-3 中可以分为三种评估方法：Zero-shot、One-Shot、Few-shot。在 Zero-shot 评估中，模型被要求在没有接触过测试类别的情况下进行分类或预测。评估过程通常包括将模型应用于新类别的样本，以检

查其对未见过类别的泛化能力。零样本评估侧重于模型对先验知识和属性的利用，以推断新类别的特征。One-shot 评估要求模型在只有一个样本的情况下学习新类别或任务，并进行分类或预测。评估过程类似于 One-shot Learning 中的任务，模型需要根据极少量的样本推断出类别特征。一样本评估通常用于测试模型的快速学习和泛化能力。Few-shot 评估是指模型在只有少量训练样本的情况下学习新类别或任务，并进行分类或预测。评估过程旨在测试模型在面对数据稀缺情况下的性能和泛化能力。少样本评估要求模型能够从有限的样本中有效学习并适应新任务。

2.1.3 GPT 到 ChatGPT

ChatGPT 则是对于 Instruct-GPT 的一种应用方式，而 Instruct-GPT^[15]最大的创新点是在 GPT 的基础上利用人类强化学习（RLHF）提高了文本数据的有效性，并增加了对话的安全性。在 GPT 中，作者认为出现幻视的主要原因是目标函数的设置，因为其数据集来自互联网，所以目标函数的目的也是预测网络文本一段话的下一个词的概率；但是网络文本质量无法保证且交错杂糅。因此会出现 GPT 进行胡说的现象。而人类强化学习则让 GPT 生成四个答案并让人类来对 GPT 生成的答案的优劣进行排序。并将此排序训练一个奖励模型，并将此模型运用在训练 GPT 上，从而使强化学习可以被运用于微调 GPT 中。这种训练方式出来的 GPT 模型也叫做 Instruct-GPT。在 Instruct-GPT 大体解决生成模型的有效性和安全性后，ChatGPT 在此基础上进行对话的上下文链接，使得先前的输入和输出都作为当前的输入从而实现对话的效果，并最终形成了现在爆火的 ChatGPT3.5。

2.2 大语言模型“幻觉”原因

2.2.1 “幻觉”的定义

大语言模型的“幻觉”，按照哈工大与华为共同编写的论文^[16]中可以将其定义为指模型的生成内容与现实世界或用户输入不一致的现象，用大白话描述就是“胡说八道”。且幻觉也可以分之为事实性幻觉和忠实性幻觉；事实性幻觉顾名思义是指模型生成的内容与可验证的现实世界事实不一致，其中的不一致又有事

实不一致和事实捏造。忠实性幻觉则指模型生成的内容与用户的指令或上下文不一致；忠实性幻觉也可以细分，分为指令不一致、上下文不一致和逻辑不一致三类。指令不一致上主要指用户提供的指令或问题与模型生成的回答或输出不一致，上下文不一致是指在对话或文本生成过程中，模型在处理前后文本时出现了矛盾或不连贯的情况。逻辑不一致则主要指大语言模型输出的内容与现实逻辑存在较大偏差。如图 2-4 展示了指令不一致的情况。



图 2-4：大语言模型指令不一致图例

2.2.2 “幻觉” 原因来源分析

根据本文 2.1 对 Transformer 和 GPT 系列模型论文的阅读，可以把模型产生幻觉的来源分为三类：数据源、训练过程和推理。

数据源方面，最大的原因在于 GPT 数据来自互联网。且 GPT 也是采用无监督训练处理这些网络文本，其数据有效性和利用率较低。虽然进行了大量的数据清理和在 instructgpt 上采用人工反馈强化学习，但是由于 GPT 本身的灵活性较高，预处理无法覆盖 NLP 和用户请求的全部方面。并且数据清理也无法消除数据中的全部偏见和错误信息，而数据中也存在着知识边界、知识缺陷和过时的事实知识。在这些文章当中，大模型可能会过度的依赖数据中的一些框架，比如中国广东省和山东省都是中国的一个省，两者在语义相似度上也较近。但是深圳是位于广东省而非山东省。但是也有些文章在对比省间经济时候会将深圳与其他省份进行对比，如果训练数据中频繁共同出现山东省和深圳市，那么大模型可能错误的将深圳归属于山东省的一座城市。

训练过程方面，如 2.1 详细介绍可知，目前效果最好的 Instruct-GPT 主要分为两大阶段——预训练阶段（无监督训练）、微调阶段（包括人类强化学习）。

预训练阶段可能会存在架构缺陷，虽然 Transformer 改进了 RNN，更加注重上下文，但是这种方式也有极限，自注意力机制仍然会不可避免地随着 token 长度增加而不同位置的注意力会被稀释。另外是 GPT 的模型本身参数越来越大，出现越来越多反直觉的训练方式，比如增大 Batchsize，降低数据噪声并不会像之前的模型训练经验那样出现训练的过拟合问题。此外，在 GPT 文章也有提到，他在模型推理的时候会依赖自己生成的词进行后续的预测，所以容易出现联级错误。

在推理阶段，如果出现 GPT 不曾掌握的知识，但是 GPT 仍然会根据相似性进行预测，从而放大出现胡说的风险。另外，推理和训练所运用的人工强化学习输出会更加迎合某类人群的偏好，从而牺牲信息真实性。并且在推理阶段，大语言模型会进行固有的随机抽样，生成内容存在随机性。

3. 大语言模型评估数据集与评估方法

3.1 大语言模型评估数据集概述

3.1.1 大语言模型评估数据集定义

大语言模型评估数据集是用来测试和比较不同语言在各种任务上性能的数据样本。这些数据集的目的是通过数据集来评估庞大的语言模型，以提高在各种自然语言处理任务的表现，同时也可以将评估数据集制作成微调数据集方便对大语言模型进行微调。这些任务包括自然语言理解、推理、生成、多语言任务和自然语言真实性等。

3.1.2 大语言模型评估数据集制作常见问题

大语言模型的评估数据集的常见问题有以下几点：1、数据泄露：测试数据集的信息泄露到训练集中，主要原因是大语言模型训练数据集来源互联网。2、数据集样本的覆盖率：评估数据集往往无法全面覆盖特定任务的各种评估方式。可能导致准确性问题、样本大小问题或鲁棒性问题。3、测试评估样本与任务无关问题：存在很多不相关或重复的测试样本，若一个评估数据集大量重复就会出现评估大语言模型过于乐观的情况。4、评估结果无法复现与分析：大语言模型输出本身存在随机性，仅记录评估准确率无法针对性分析评估结果。

3.1.3 大语言模型制作评估数据集常见形式

大语言模型数据集的制作主要包含以下几种常见形式：预整理的数据集，该类数据集大多先于大语言模型而存在，这些测试主要为人类而非模型。并且这类数据集可能具有基于记忆的问题，这类问题并没有很好的考察大语言模型的逻辑推理。利用爬虫等制作的数据集，该类数据集虽然庞大，但是评估集的样本可能已经存在基础的训练集中，因此仅仅取互联网作为评估不可取。人工制作数据集，这类通常用来防止数据泄露，但是缺点是规模小，数据集覆盖范围小。模糊化的评估集，这是现有评估数据集的扩展版本，目的是测试模型在

面对各种变化的鲁棒性。人工筛选数据集，这类数据集由专业人士进行选取，较人工制作来说制作速度略有上升，但是其最大的问题是可能会收到一定偏见的影响，通常情况下，评估者也需要掌握问题的解决方法，对人的要求较高。

3.1.4 大语言模型评估数据集题型

常见的评估数据集题型与评估方式主要有以下几种：

1. 选择题形式

选择题形式是当前较为主流的评估数据集形式，这种形式的数据集主要优势在于选择题有明确答案，评估过程可以更加标准化，减少了主观性。并且评估起来较为高效。以选择题作为载体也可以覆盖多种领域，包括教育、科研等。劣势则选择题大多都过度依赖记忆性，忽视了深层次的理解与推理；其次是难以评估模型的创新能力；且如果仅以这种形式的题目作为考核标准可能会引导模型过拟合。

选择题的评估大语言模型的方法主要源自 GPT 的论文，大体可以分为 Zero-shot、Five-shot（也叫 Few-shot）。在进行 Five-shot 也分为仅预测答案的 prompt 和进行思维链的 prompt^[17]。但是进行思维链模式一般对模型本身参数大小有较大的要求，而 Zero-shot 则最好是进行了人工反馈强化学习后，属于 Instruction-GPT 后再进行 Zero-shot 评估效果会更具有参考性。Zero-shot 的评估结果可以用于面向用户的参考，因为用户没精力编写 prompt，而 Few-shot 是面向开发者的，因为在构造基于 LLM 的应用的时候，开发者总是希望用 prompt engineering 的方法进一步提升模型的效果。利用该类数据集评估模型时，往往都是统计模型对题目的准确性来评估大模型知识的可靠性。图 3-1 展示了 Few-shot 和 Few-shot COT 两者提示词之间的区别。

以下是单项选择题，请选出其中的正确答案。	以下是单项选择题，请选出其中的正确答案。
{题目1}	{题目1}
A. {选项A}	A. {选项A}
B. {选项B}	B. {选项B}
C. {选项C}	C. {选项C}
D. {选项D}	D. {选项D}
答案：A	答案：让我们一步一步思考，
	1. {解析过程步骤1}
	2. {解析过程步骤2}
	3. {解析过程步骤3}
	所以答案是A。
{测试题目}	{测试题目}
A. {选项A}	A. {选项A}
B. {选项B}	B. {选项B}
C. {选项C}	C. {选项C}
D. {选项D}	D. {选项D}
答案：	答案：让我们一步一步思考，
	1.

图 3-1：Few-shot 和 Few-shot CoT 提示词对比

2. 问答题形式

问答题形式回答较为多样，为了更好的评估模型输出的准确性，尤其是在一些数学的问答题上。在此基础上，使用的较多的方法的基本原理源自 GPT-3，主要使用 few-shot CoT^[17]方法，这种方法结合了“Few-shot learning”和“CoT (Chain-of-Thought)”两个概念。Few-shot 前文已陈诉，而 CoT 则是代表着一种填充模板的形式，模板中关键回答都会有特殊符号标明，而大模型的思维链可以引导大语言模型将复杂问题分步骤解决，同时增加其过程的可解释性和可控性，也方便在评估过程中分析大语言模型存在的问题，使得我们可以更好的判断大模型在求解当前问题上究竟是如何工作的。CoT 已经被证明在数学应用题、常识推理和符号操作上经常使用，缺点就是会增加输出输入的正确性检测难度。如果是 Zero-shot CoT 则可以直接在问题的结尾加上“Let’s think step by step”这几个词即可。

利用 CoT 方法检测准确性主要依靠特殊字符的标识，可以仅检测大语言模型回答的最终答案，作为当前样例的输出。也可以选择将每个步骤都作为得分点之一，但是步骤的权重没有最终权重要高，但是这会增加评分的难度。

3. 其他题型

像是其他的机器翻译、文本分类、序列标注、抽取式摘要、信息检索、表格解读、数理推论、常识推理等题型也都可以采用 few-shot CoT 方法，如果模型经过 RLHF 后，且达到了一定的参数规模，则可以尝试采用 zero-shot CoT 方法进一步从用户的角度进行测试。其评测方法类似于问答题，但是排序和分类等评测方法则可以引导大模型转为类似选择题的评估方式。

本设计主要跟随大语言模型评估主流，主要采用选择题这种客观且较好评估的数据集对大语言模型进行多角度评估。问答题仅提供评估结果以供参考。

3.2 大语言模型评估方法概述

3.2.1 大语言模型评估方法定义

大语言模型的评估方法主要有两种形式，人工评估和自动评估。基于人工评估：人工评估是指由人类专家对模型生成的文本进行主观评价，如准确度、流畅度、可读性等。这种评估方法比较可靠，但成本较高，时间较长。基于自动评估：自动评估是指使用客观的评估指标对模型生成的文本进行评价，如 BLEU^[18]、ROUGE^[19]、METEOR^[20]等，当然以上的评估方法都是需要基于一定的评估数据集之上，得到评估结构后才可以进行对应的评估。但是大语言模型的评估通常会结合使用多种方法，包括但不限于直接评估指标、基于辅助模型的评估、基于模型的评估以及人工评估等，这样可以多个角度全面评估大模型的性能，同时可以避免单一评估方法可能存在一些局限性。

3.2.2 大语言模型评估方法介绍

1. 大语言模型自我评估：大语言模型自我评估分为两种，一种是利用另一个评价模型针对问答题相关的评估数据集做出评价，但是这种评价并没有一个统一的标准。评价结果虽然来自同一个大语言模型，并且即使这个评价模型有经过相应的微调，但是由于大语言模型本身就存在一定的随机性，所以大语言模型自我评估通常都难有一个统一的标准进行评估。而另外一种则是专门训练一个大语言模型，利用大语言模型自主产生数据集，自主追问被评估的大语言模型从而从多角度自主评估大语言模型的创造性、准确度等。但是目前该

技术评估结果的认可度和可靠性上还有待加强，所以在主流任务上还是以评估数据集为主。

2. 利用评价指标反应大语言模型评估，目前评价指标主要包含以下几类：

(1) 准确度 (Accuracy)：衡量模型预测结果与实际结果的一致性。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP 表示真正例 (True Positives)、TN 表示真负例 (True Negatives)、FP 表示假正例 (False Positives)、FN 表示假负例 (False Negatives)。

(2) 困惑度 (Perplexity) [21]：衡量模型对测试数据的预测能力，常用于语言模型评价。

$$Perplexity = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2 P(x_i)}$$

其中，N 表示数据集中的总词数， x_i 表示第 i 个词， $P(x_i)$ 表示模型对第 i 个词的预测概率。

(3) BLEU 分数：衡量模型生成的文本与参考文本的相似度，常用于机器翻译任务。

$$BLEU = BP \times \exp \left(\sum_{n=1}^N w_n \cdot \log p_n \right)$$

BP 表示短句子惩罚因子，exp 中表示 n -gram 精确度的加权求和， w_n 表示 n -gram 的权重， p_n 表示 n -gram 的精确度。

(4) Distinct：衡量生成文本的多样性，Distinct 值越高，表示生成文本的多样性越高。

$$Distinct - 1 = \frac{N_{unigrams}}{N_{tokens}}$$

其中， $N_{unigrams}$ 表示生成文本中不重复的单词数量， N_{tokens} 表示生成文本的总词数。

(5) F1 分数：是准确率和召回率的调和平均数，常用于信息检索和文本分类任务。

(6) MCC (Matthews Correlation Coefficient) [22]：是二分类任务的一个评价指标，取值范围为 $[-1, 1]$ ，1 表示完全预测正确，-1 表示完全预测错误。

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3. 人工评估大语言模型：人工评估是利用人类专家针对大语言模型生成的文本进行主观性评价，在利用人工评估时，通常都会重点评估以下几个能力：通用能力，主要是评估模型在各种任务和场景下的表现。其中包括自然语言处理任务、推理、生成、多语言任务和自然语言真实性等。泛化能力，主要评估模型对未见过的数据的处理能力，即大语言模型是否能够在实验人员精心设计甚至错误的数据集上表现良好。鲁棒性，人工评估模型对输入的微小变化的敏感度，以及模型在面对不同数据分布、噪声或输入变化时的性能表现。跨域性能：评估模型在不同领域的任务上的表现，例如在自然科学、社会科学、医疗应用等领域的表现。多语言能力和评估模型解题步骤的可解释性。安全性：评估模型是否存在可能导致物理、情感或财务伤害的行为。以上是人工评估大语言模型在通常情况下会关注的方面，但是这些方面并非孤立，而是相互关联，共同构成对大语言模型全面的评估。

3.3 评估数据集的格式化保存

3.3.1 评估数据集常见格式

当前的评估数据集的保存格式多样且不固定，目前并没有一个统一的格式保存，并且大多数数据集并没有一个统一的学科分类。这不仅增加了评估程序和评估流程的编写，同时多类型没有统一的分类也给微调大语言模型带来了不小的麻烦。在大语言模型评估数据集上，常见的保存数据集的形式为 CSV (Comma-Separated Values)：这是一种常见的表格数据格式，每行代表一个数据样本，每列代表一个特征。这种格式简单且易于理解，但可能不适合复杂的数据结构。

JSON (JavaScript Object Notation): JSON 是一种轻量级的数据交换格式,易于人阅读和编写,同时也易于机器解析和生成。它基于 JavaScript Programming Language 的一个子集。XML (eXtensible Markup Language): XML 是一种标记语言,它的设计目标是传输和存储数据。它被设计为具有自我描述性,并且能够定义自己的标签。HDF5 (Hierarchical Data Format): 这是一种用于存储大量数据的文件格式,包括大型数组和表格。它支持复杂的数据类型和分层数据组织。但是最为常用的还是 CSV 与 JSON 对评估数据集进行保存。

3.3.2 评估数据集整体介绍

1. Ceval^[23]: Ceval 数据集本身都是基于选择题的形式,具体的保存形式为 csv,并且 Ceval 上是分多个学科,以文件名的形式记录多类问题,在这个数据集的 subject_mapping.json 文件上也记录了各个文件的四大分类,分别是 STEM、Social_Sciencv、Humanities、Other 四大类,在进行 Ceval 数据集评估时,所用的评估指标为准确率作为唯一的评估指标。

2. M3KE^[24]: M3KE 数据集形式为选择题,保存格式为 json,保存和记录类别的方式和 Ceval 数据集类似,在数据集的 subject_cluster.mapping.json 文件上记录每个文件对应的四大类,分别是 Natural Sciences、Social_Sciencv、Humanities、Other。

3. GAOKAO-Bench^[25]: GAOKAO-Bench 是一个以中国高考题目为数据集,测评大模型语言理解能力、逻辑推理能力的测评框架。其中包含 2010-2022 年全国高考卷的题目,其中包括 1781 道客观题和 1030 道主观题,构建起 GAOKAO-Bench 的数据部分。在分类上该数据集主要分为不同题型——主观题与客观题以及不同的高考学科。

以上数据集都为测试中文数据集重在测试大语言模型理解中文的准确性。

4. MMLU^[26]: MMLU 评估数据集形式和 Ceval 的学科分类和问题形式以及评估指标都类似,但是 MMLU 属于英文的评估数据集,主要评估大语言模型在英文的理解和阅读能力。

5. GSM8K^[27]: GSM8K 是一个基于问答题相关的高质量多样化的小学数学问题的数据集, 主要采取问答的方式。在检测方式上他采取的方式和 GPT-1 的评估原理, 对关键的计算式等采用特殊符号进行标记, 在不限制大语言模型的自由度的前提下更好的检测结果准确性。同时也要求模型一定要有 COT 的能力。其主要的存储方式为 json, 依靠大语言模型生成的特殊符中的计算式判断该题是否正确。

6. BIG-Bench Hard^[28]: BIG-Bench Hard 也是属于一个多样的问答式的大语言模型评估数据集, 但是他针对每个数据集都按照 GPT-2 的 few-shot 提供了几个样例让大语言模型明白当前所要评估的任务对象和目标。数据集的存储方式为 json, 评估方法为准确率。

3.3.3 统一数据集格式

由于每个评估数据集的存储格式和类别都有所不同, 这对后续自动化评估带来不小的困扰, 因此需要将每个数据集都转为统一的格式, 本设计采用 JSON 统一保存数据, 主要因为 JSON 为一种轻量级的数据交换格式, 具有易读性和易编写性的特点。其易于解析和生成: JSON 格式的数据易于解析和生成, 适合机器处理。各种编程语言都提供了 JSON 的解析和生成库, 可以方便地读取和操作 JSON 格式的数据集。且有良好的可读性与灵活性。另一方面在统一好数据集后可以便于数据交换和共享, 同时简化评估程序和流程, 促进模型性能比较的同时降低数据处理成本。因此, 统一数据集格式对于评估数据集的保存、交换、分析和利用具有重要意义, 有助于推动研究领域的发展和进步。

经过之前对每个不同评估数据集的了解, 本文针对不同数据集的类型、提问方式、语言、问题存储方式和类别等进行统一了解, 同时将多种不同的数据集统一存储为统一格式的 JSON 格式, 具体的程序流程如图 3-2 所示:

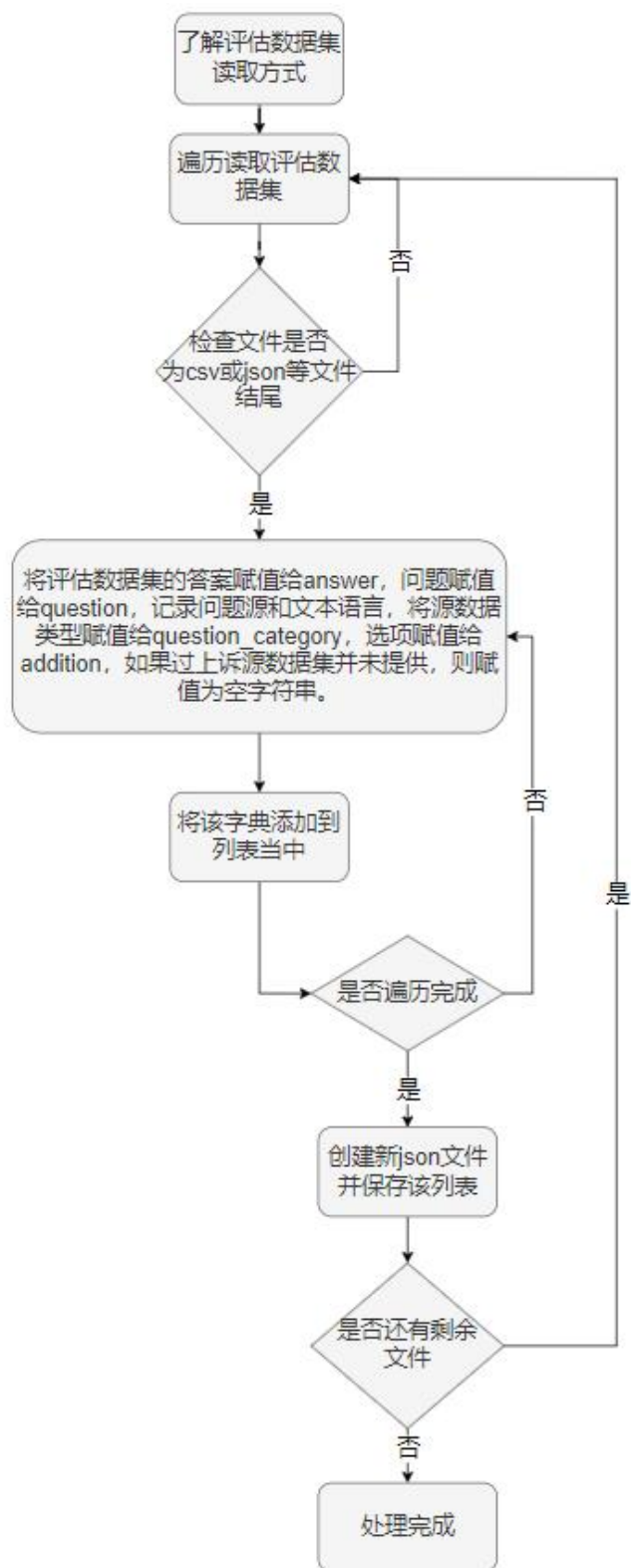


图 3-2: 统一评估平台数据集脚本流程

具体字段与目的如表 3-1 所示:

表 3-1：统一格式后的 JSON 字段与对应目的

字段名	目的
id	数据条目的唯一标识符
ch/en	语言标识符（例如中文或英文）
question	问题文本
source	数据来源（数据集名称或来源）
category	问题类别（例如客观题或主观题）
question_category	问题的具体类别
question_category_s_ch	问题的具体类别（中文描述）
question_category_s_en	问题的具体类别（英文描述）
addition	附加信息（可选）
prompt	提示信息（可选）
answer	问题的答案
explan	答案解释或说明（可选）
other	其他信息或字段（可选）
question_category_new	保存重定义统一后的类别
Same_questions	保存相同的问题文件名与 id（可选）

在统一处理完六大大语言模型评估数据集后，一共获得累计 168060 条涉及多种领域的大语言模型评估数据集。其中附带正确答案的数据集条目共有 135596。评估数据集中主观题的数量为 16333，客观题的数量为 168060，客观题的类别为大语言模型评估数据集常见的选择题形式。且语言为中文的评估数据集数量为 37236，英文评估数据集为 130824。而在经过统一格式化处理后的 Json 文件如图 3-3 所示：

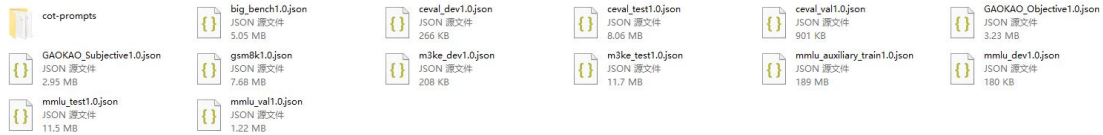


图 3-3：格式化后文件名称

3.4 评估数据集重分类和清洗

3.4.1 评估数据集重分类目的

将收集到的评估数据集重新分为 STEM、Humanities、Social Science、Other 四类有以下好处：1、组织性和清晰性：将数据集按照类别进行分组可以使整个数据集更有组织性和清晰性。通过将数据集分为四类，可以更容易地管理和维护数据，使其更易于理解和使用。2、方便性和效率性：分类可以提高数据的查找和访问效率。当数据按照类别进行分组时，用户可以更快速地找到他们需要的数据，而不必浪费时间在整個数据集中进行搜索。3、方便分析和比较：将数据集分为四类可以方便进行分析和比较。通过对每个类别的数据进行分析，可以更好地了解数据集的特点和趋势，从而更好地进行数据挖掘和分析工作。4、更好的进行标准化和规范化：将数据集分为四类可以帮助标准化和规范化数据集。通过对数据进行分类，可以确保数据集中的每个类别都具有相似的特征和属性，从而提高数据的质量和可靠性。同时统一分类数据集可以更容易地确定每个类别的微调目标和策略。通过对每个类别的数据进行微调，可以更好地训练模型以适应不同类别的数据特征，从而提高模型在各个类别上的性能表现。

3.4.2 评估数据集重分类方法与流程

先对四大类的关键词进行扩写和定义，STEM 为理工科的缩写，代表科学、技术工程和数学等。而 Humanities 代表人类文化，主要包括价值观念、历史、语言、文学等。Social Science 代表社会科学，主要研究人类社会、社会关系、社会行为和社会制度等，也包含社会、心理和政治学等。其余不属于以上三类则归为其他类。

每个评估数据集的分类都具有其独特的特征和特点，因此无法简单的利用字典映射将各个评估数据集原有的类别映射到以上四大类。所以本设计将重分类主要归为以下几个步骤：文本预处理：首先需要对文本数据进行预处理，包括去除停用词、标点符号，进行词干提取或词形还原等操作，以减少噪音和提取文本特征。检查该问题是否属于四大类本身，如果不是则利用 spaCy 将四大类和评估数据集的类别转化为词向量或句子向量表示，并通过计算两个文本之间的相似度。这个相似度是基于向量表示的，通过计算文本向量之间的余弦相似度或其他相似度指标来衡量文本之间的相似程度。如果相似度并没有超过阈值，则归为其他类。否则选出最相似的类别作为该数据的新类别，并添加到新的 question_category_new 字段中并保存到新的 JSON 文件当中。具体流程图如图 3-4 所示：

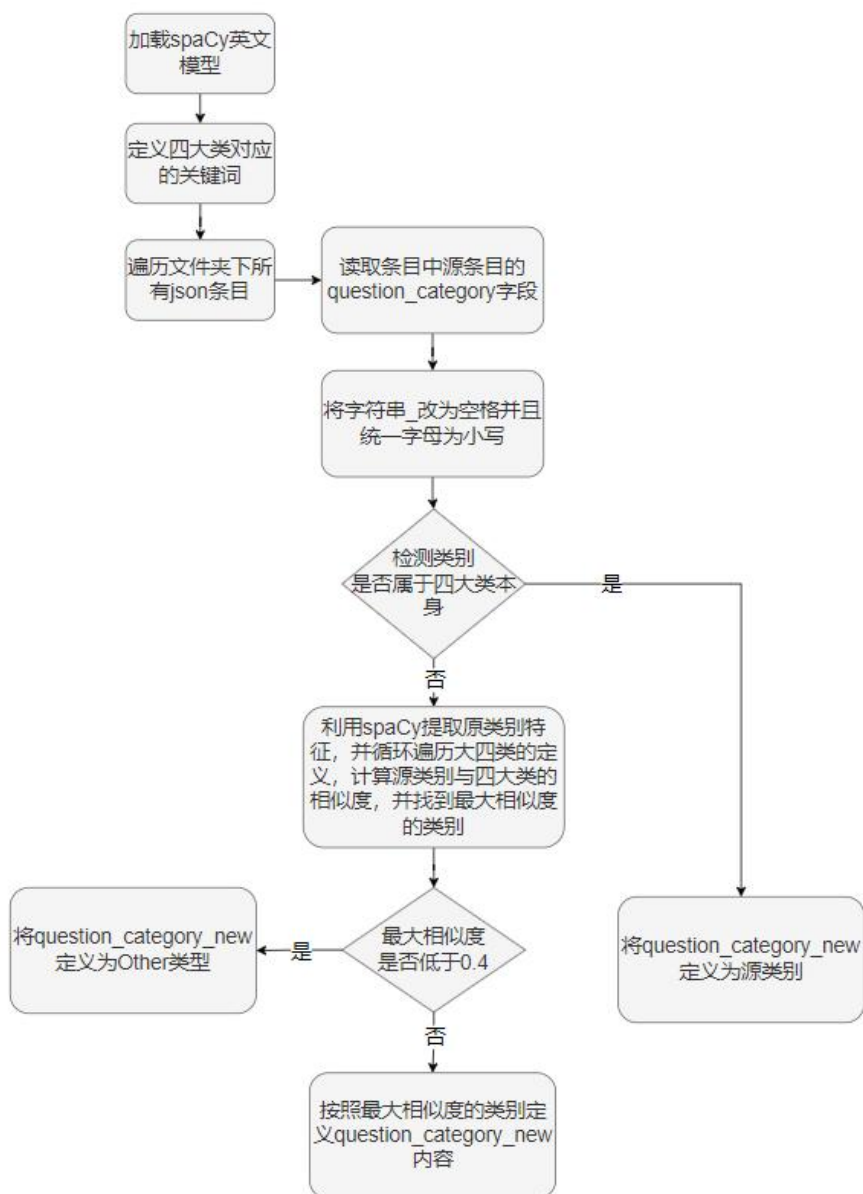


图 3-4：重分类脚本程序流程

在具体的代码中也利用了字典来保存已得到源类型的具体四大分类，用来避免多次相似度匹配，提高流程重分类效率。

3.4.3 评估数据集数据清理

传统数据集的数据清理主要针对缺失值和异常值处理，并且需要针对数据进行去重。但是评估数据集最重要的数据为问题和答案（非测试集），并且由于大语言模型本身就存在随机性等诸多问题，因此在评估数据集中是允许出现少量相同的数据对大语言模型随机性进行测试。因此数据清理方面本设计主要

编写代码，遍历 json 文件并统计出现相同 question 和 addition 字段，并将相同的问题以文件名和 id 的形式添加到 Same_question 字段当中。同时将没有答案的 json 文件和附带答案的 json 文件分离，方便后续对大语言模型进行评估。代码流程如图 3-5 所示：

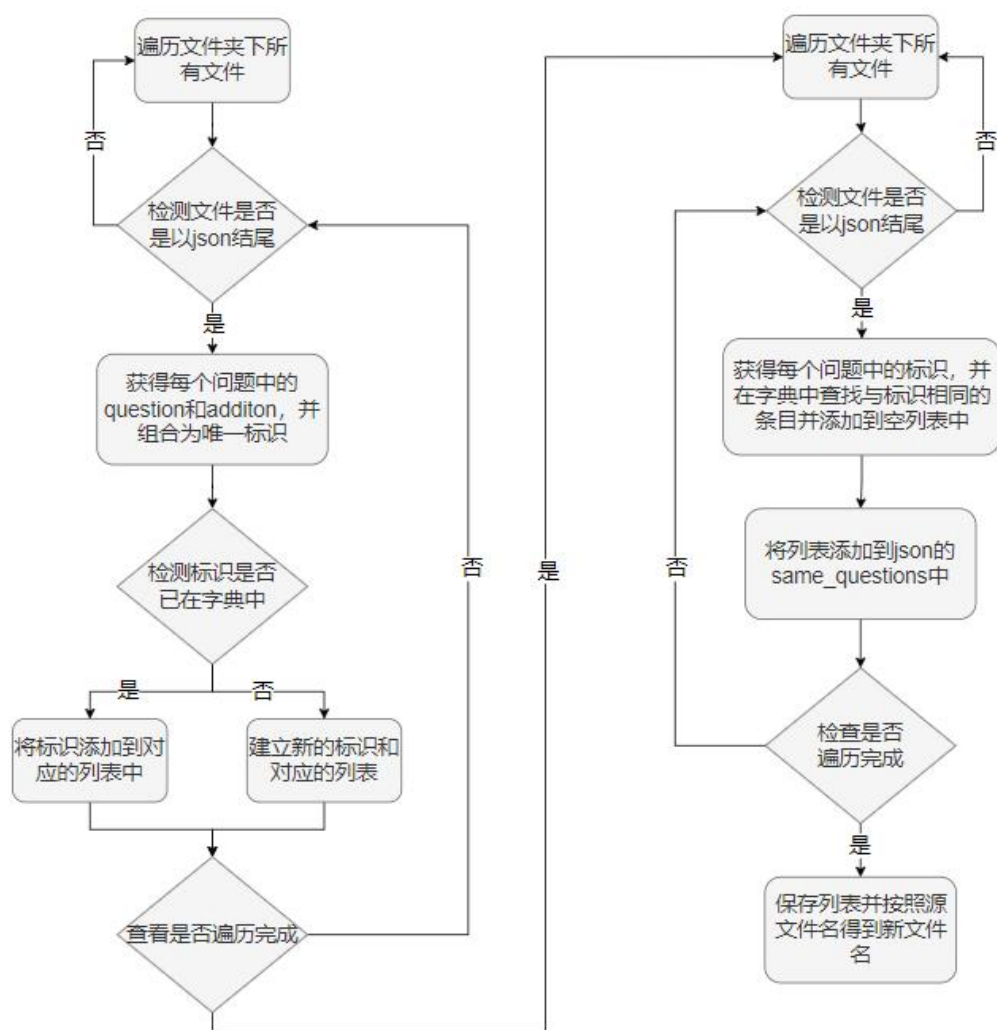


图 3-5：大语言模型评估脚本流程

4. 常见开源大语言模型部署与评估流程

4.1 常见的开源大语言模型

在国内较为常见的开源大语言模型为清华大学发布的 ChatGLM，其为一个开源的、支持中英双语的对话型语言模型，其本身的参数达到亿级别，但是在 INT4 量化级别下最低只需要 6GB 的显存即可。并且也使用过类似 Instruct-GPT 类似的技术，针对中文回答和对话进行优化。经过人类反馈强化学习的加持，ChatGLM 已经能正常的与人类对话，并可以生成相当符合人类偏好的回答。

目前的 ChatGLM 已经开源了 3 代，ChatGLM1 是基于 GLM 的经过 1T 标识符的中英无监督训练，再辅以监督微调、反馈自助、人类反馈强化学习等技术的加持，最终训练出来的大语言模型。而二代在保留初代模型对话流畅、部署门槛较低等众多优秀特性的基础之上，还利用更多的中英文标识符进行预训练，获得更强大的性能、更长的上下文和更加高效的推理。而第三代则是由智谱 AI 和清华大学 KEG 实验室联合发布的对话预训练模型。对比先前的几代，3 代在语文、数学和知识等有了更多的提升，且 ChatGLM3-6B-Base 具有在 10B 以下的基础模型中最强的性能。并且在 ChatGLM3 上采用全新设计的 Prompt 格式，且支持工具调用、代码执行和 Agent 任务等复杂场景。

4.2 开源大语言模型部署

本设计大语言模型的部署主要基于 Anaconda 创建虚拟 Python 环境，使用其中的环境管理工具 Conda，可以创建、管理和切换不同的 Python 环境。这对于不同项目或不同模型的部署非常有用，可以避免版本冲突和依赖问题。依据本地的 CUDA 和 NVCC 版本进行安装对应的 Pytorch，并依据 ChatGLM 中的工程文件，对 requirements.txt 文件安装模型的依赖项。配置好大体环境后在 hugging 上下载模型权重并加载模型权重，同时对模型进行简单测试。

部署与调用结果图如图 4-1 所示：



图 4-1：大语言模型部署结果图

4.3 大语言模型评估流程

4.3.1 大语言模型评估实现

在处理好大语言模型评估数据集的格式化问题，同时了解完将要评估的大语言模型和对应的 api 调用就可以开始编写大语言模型自动化评估的 python 脚本。但是由于大语言模型本身存在不稳定性 and 不确定性，每次评估的结果都不尽相同，因此需要有庞大的评估数据集才可尽可能的从多角度对大语言模型进行测评。因此在编写大语言模型评估数据集脚本时候具体需要聚焦于以下几个问题：

- 1、大语言模型的加载方式，如何将大语言模型载入到正确的显卡上并成功运行。
- 2、如何建立提示词文本，用于生成模型的输入，同时提示词文本也会一定程度上影响大语言模型的评估结果。
- 3、如何建立提取评估答案的提示词或方法。
- 4、在编码之后如何保存大语言模型的输出，方便后续对评估结果进行分析和可视化。

针对上诉问题，首先是加载方式。在 python 的脚本中大语言模型通过 `AutoModel.from_pretrained()` 方法加载预训练模型，并使用 `.cuda()` 将模型放置在 GPU 上进行计算。这样可以充分利用计算机的显卡性能，达到评估速率最大化。提示词方面，在进行选择题测评时本设计采用较为基础的一问一答的 `zero-shot` 模型，因为选择题大模型的灵活性相对没有主观题那么大，所以不需要采用 `few-shot` 固定其输出。在提取评估答案的设计上，先设计了新的提示词，在每个问题及其回答后面补上一句“综上所述，ABCD 中正确的选项是：”同时提取大语言模型的输出结果（`logits`），并选择最高概率的选项作为预测答案，将模型输出的 `logits` 转换为预测的答案，并最终记录。在得到大语言模型的预测结果后，会将大语言模型的解释部分和最终答案以及问题的来源于 `id` 都保存到对应的字典，并最终保存为 `json` 文件，方便后续的可视化和结果分析。保存的 `json` 具体格式如表 4-1 所示：

表 4-1：保存结果的 JSON 的字段名与目的

字段名	目的
<code>Source_question</code>	方便找到源问题文件
<code>id</code>	方便在问题文件中对问题进行定位
<code>explan</code>	保存大语言模型的解释
<code>answer</code>	方便对大语言模型结果进行统计

4.3.2 大语言模型评估流程如下

先加载模型和数据，通过 `AutoTokenizer` 和 `AutoModel` 加载预训练模型和分词器，定义答案选项和相应的标签映射。后进行数据处理和准备，遍历指定目录下的所有 `.json` 文件，读取数据并将每个数据点的问题和附加信息合并为一个新字段。创建空字典来存储每个数据点的 `ID` 和模型生成的答案。模型推理

部分，对每个数据点进行模型推理，先生成模型输入的提示文本，使用 tokenizer 对输入进行编码，并转换为 PyTorch 张量。生成模型输出并解码为文本。构建包含答案提示文本的完整文本。对完整文本进行编码，并转换为 PyTorch 张量。获取模型输出的 logits 并预测最可能的答案。计算匹配数量和准确率。在得到最终答案后将结果存储，将预测的答案和相关信息存储到字典中，同时将每个数据集的预测结果存储为新的 .json 文件。同时在评估的同时也计算每个数据集的准确率，并且打印在 linux 界面上。具体流程如图 4-2 所示：

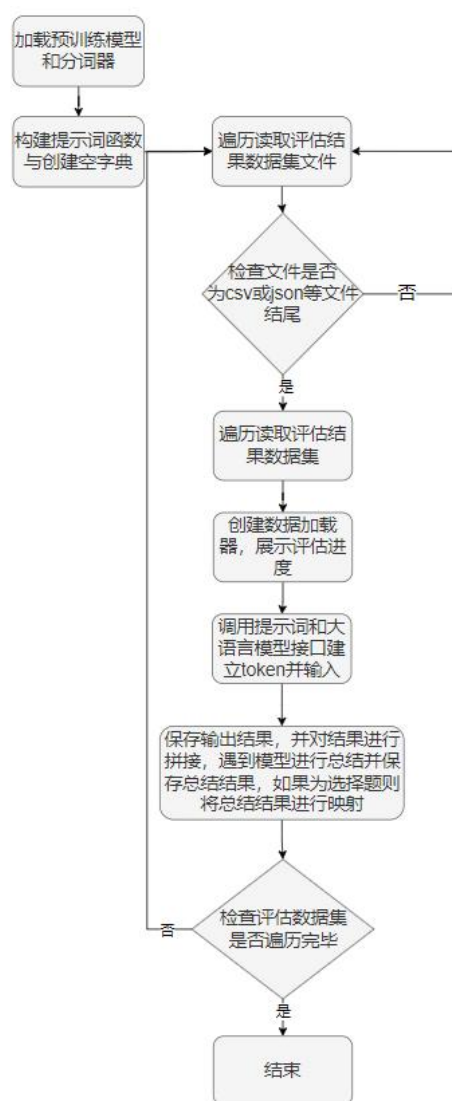


图 4-2：大语言模型评估流程

4.4 大语言模型评估代码

具体附带详细注释的代码请参考下方 GitHub 链接：<https://github.com/B>

[orfish/llm_evaluate](#)（由于 gihub 文件大小限制，本链接仅包含代码和部分数据集）

4.5 评估输入整体介绍

在客观题方面，大语言模型的评估数据集大多都为选择题的形式。本次大语言模型评估的客观题总数为 17708 条，客观题具体数目如表 4-2 与图 4-3：

表 4-2：客观题评估数据集数量

评估数据集名称	客观题数量
ceval_dev	260
ceval_val	1346
m3ke_dev	355
Mmlu_dev	228
Mmlu_test	13985
Mmlu_val	1474
总数	17648

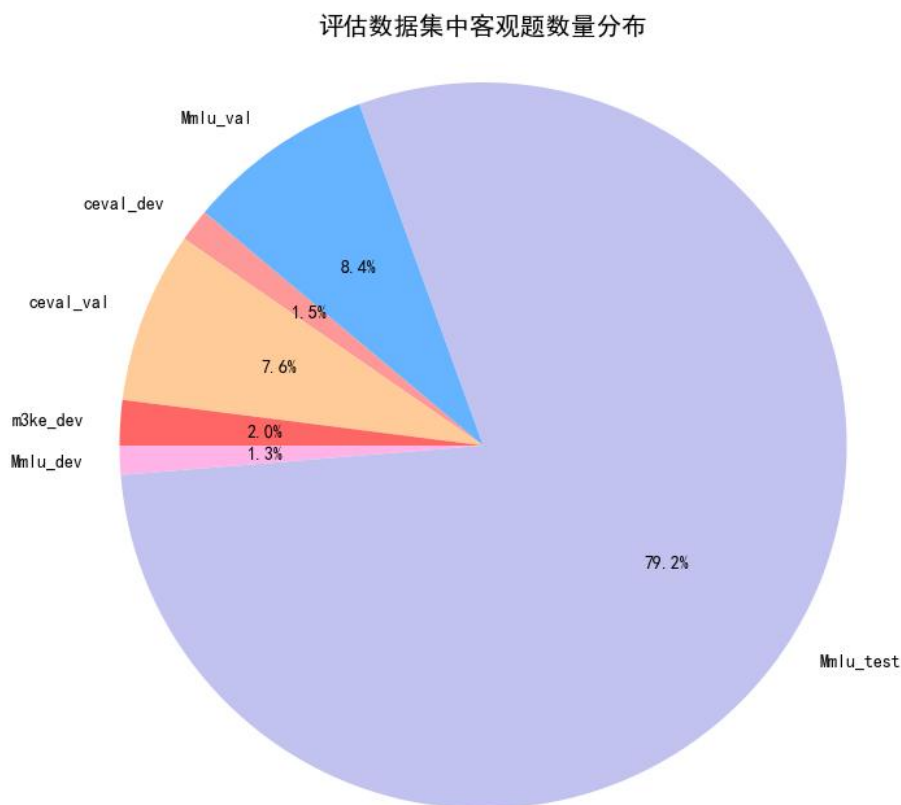


图 4-3：客观题评估数据集数量占比

主观题方面如表 4-3 与图 4-4，其总数为 16333：

表 4-3：主观题评估数据集数量

评估数据集名称	主观题数量
Big_bench	6511
GAOKAO	1030
Gsm8k	8792
总数	16333

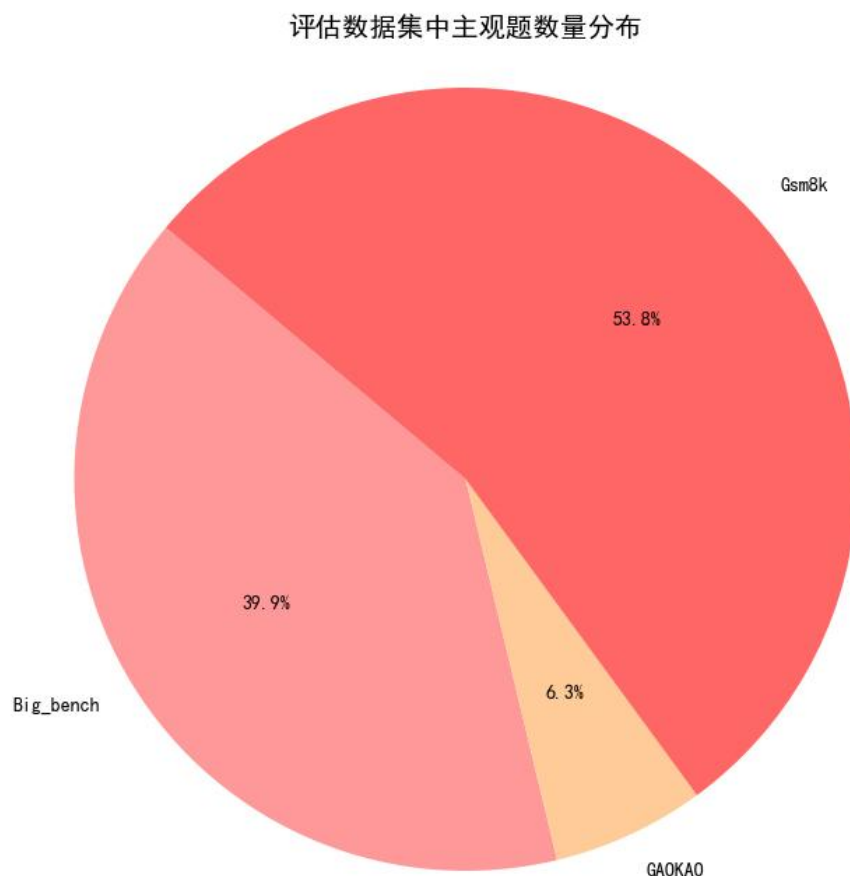


图 4-4：主观题评估数据集数量占比

其中选择题的语言方面分布如下，其中英文所占的比例较大，主要原因是英语的评估数据集其测试集也附带了答案。但是中文评估数据集中的测试集是不附带答案的。虽然也对该数据集进行了评估，但是无法验证评估的正确性，也无法展示，所以不计入本次评估中。具体客观题分布如表 4-4 与图 4-5：

表 4-4：数据集中英语语言数量分布

语言	英语	中文
客观题数量	15687	1961
主观题数量	15303	1030

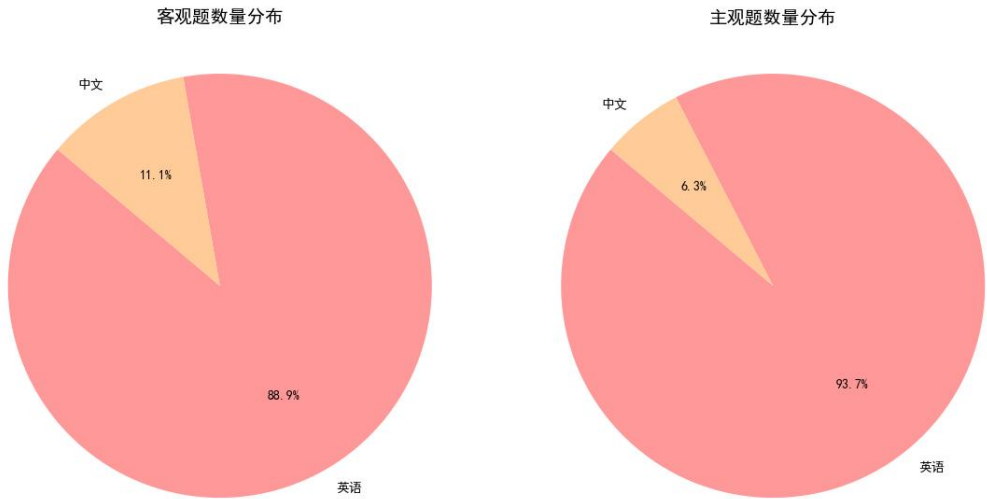


图 4-5：数据集中英语语言数量占比

类别方面，客观题评估数据集中主要集中于理工科，其次是人文学科和社会科学，最后其他类。主观题则是人文科学的比例较大，具体数量如表 4-5 与图 4-6：

表 4-5：评估数据集类别分布

类别	客观题数量	主观题数量
STEM	12606	5339
Humanities	3284	9607
Social Science	1248	250
Other	510	1137

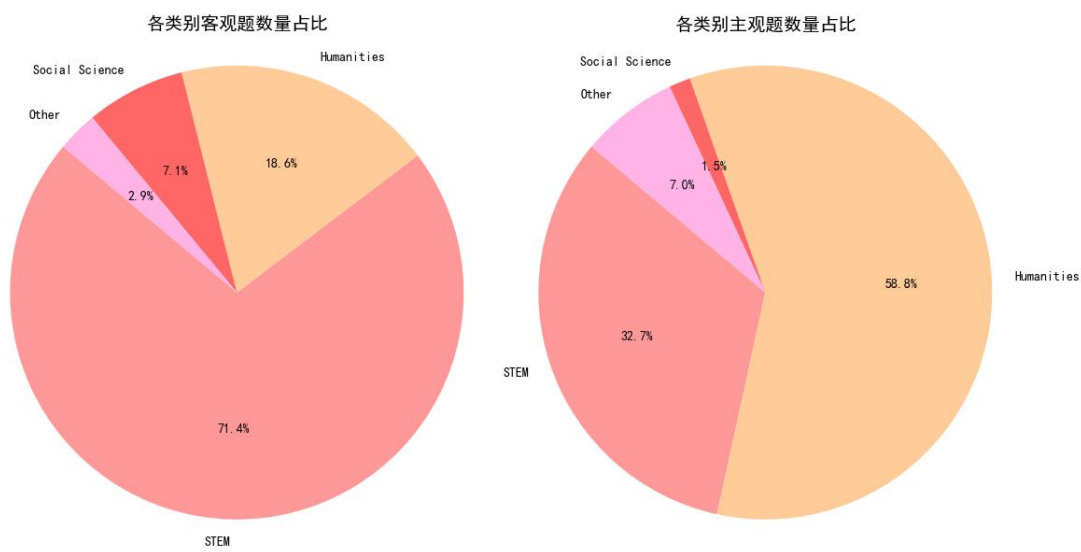


图 4-6: 评估数据集类别占比

4.5.1 评估结果展示与分析

统计得到客观题评估数据集上的答案，其中大语言模型评估的正确率如表 4-6 所示：

表 4-6：客观题评估结果正确率统计

模型名称	正确数量	正确率
ChatGLM1	6834	38.72%
ChatGLM2	7642	43.30%
ChatGLM3	9350	68.51%
总量：	17648	100%

从上表的表格可以很清楚的观察到，每一代模型的正确答案数量分别为 6834、7642、9350，总问题数为 17648。通过这些数据，可以看出随着每一代的更新，模型在正确答案数量上有所增加。图 4-7 为本次评估的整体直方图和折线图，可以更进一步的观察 ChatGLM 每一代的进步：

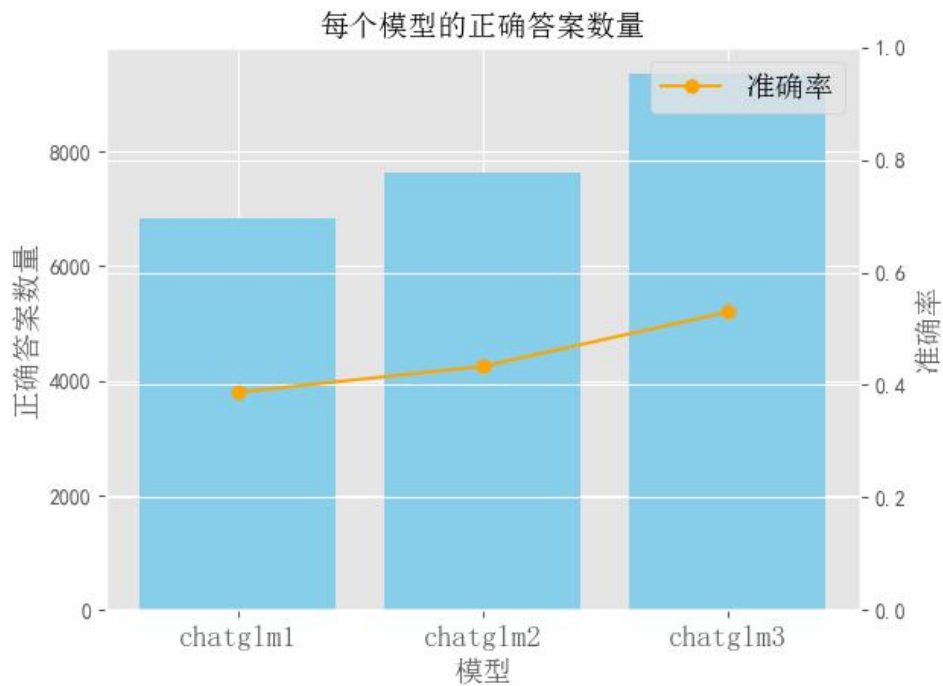


图 4-7：大语言模型整体评估结果展示

同时，本文也统计了大语言模型在理工科、人文科学、社会科学和其他领域的表现情况，具体可以参考表 4-7，表内数据代表正确率：

表 4-7：客观题不同类别正确率展示

模型	STEM	Humanities	Social Science	Other
ChatGLM1	37.27%	41.29%	49.44%	31.96%
ChatGLM2	40.11%	49.48%	57.21%	48.43%
ChatGLM3	50.37%	57.25%	68.11%	52.94

同时为了更好的比较不同组别的数据，对于以上数据也针对性的绘制了多组并排柱状图，主要是为了比较不同组别数据，使得不同组别之间的比较更加直观。通过在同一图表中显示多组数据，可以方便地比较它们的差异和关联性。同时该图强调组内比较：多组并排柱状图可以突出显示不同组内部的数据差异。观察每组内不同柱状图的高度，可以直观地了解不同类别或条件下的数据变化，帮助识别趋势和模式。同时可以更加清楚的观测到组内数据。具体柱状图如图

4-8:

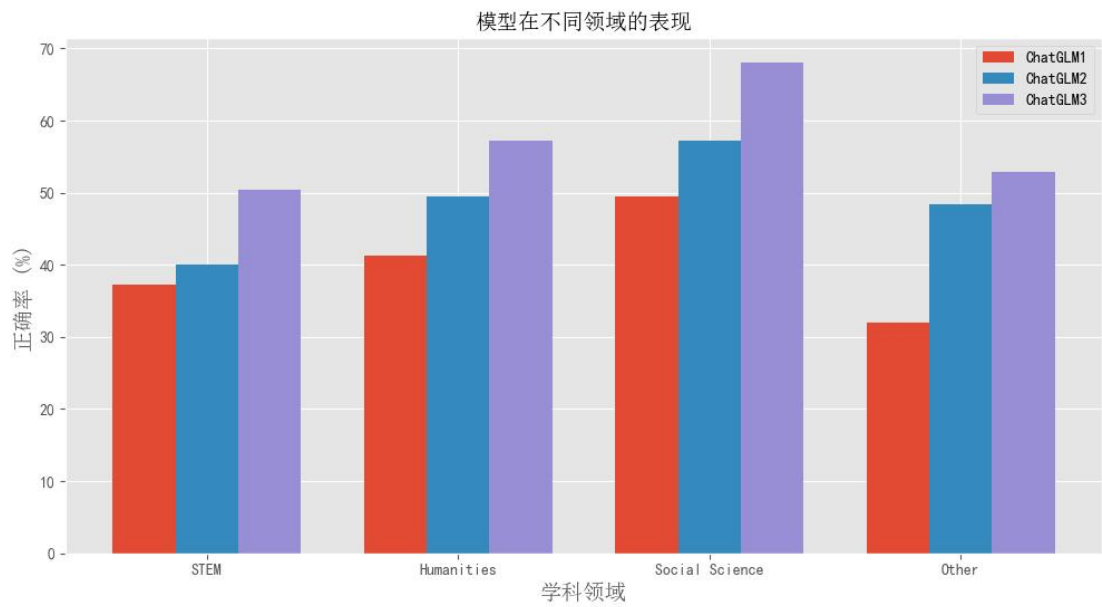


图 4-8：三代大语言模型再不同类问题下的表现

在语言方面，如下表所示，这个表格展示了三个不同模型在英语和中文方面的语言偏向情况。从数据可以看出，虽然每个模型在不同语言上的表现存在一定差异，但整体来说，差别并不是很大。尽管 ChatGLM1 在两种语言上的表现相对较低，但仍然保持一定水平。如表 4-8 所示：

表 4-8：客观题中英文正确率表示

模型名称	English	Chinese
ChatGLM1	38.83%	37.79%
ChatGLM2	42.52%	49.57%
ChatGLM3	52.75%	54.85%

图 4-9 直观反映了本次评估在多个评估平台和对应的数据集上的表现：

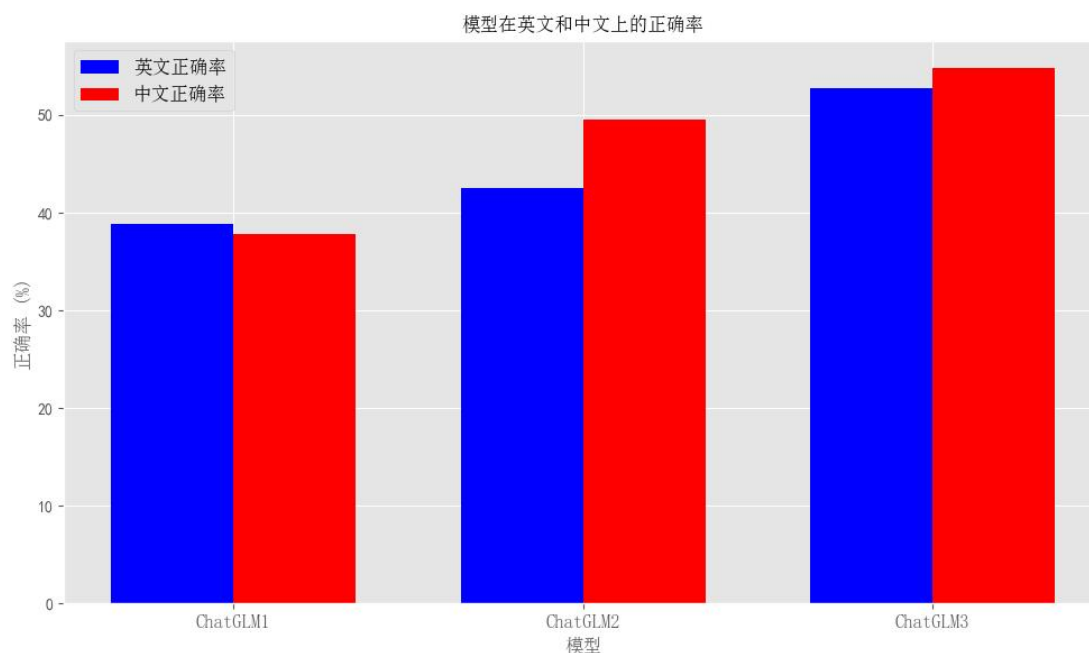


图 4-9: 模型在中文和英文上的正确率

同时 ChatGLM 本身也有利用 C-Eval 评估数据集评估的评估结果，但是他并没有公开其评估 MMLU、GSM8K 的评估方法，以下为 ChatGLM 本身的评估数据和本次实验得到的评估数据正确率对比，其中附带 (base) 的项表示该大模型并未进行类似 Instruct-GPT 那样的利用人类强化学习 (RLHF) 提高了文本数据的有效性。具体数据如表 4-9，空白部分表明并没有官方正确率：

表 4-9: 评估结果与官方结果对比

模型名称	数据集名称	本次实验正确率	官方正确率
ChatGLM1	C-Eval	36.92%	38.90%
ChatGLM2	C-Eval	49.88%	50.10%
ChatGLM3	C-Eval	54.67%	69.00% (base)
ChatGLM1	Mmlu	38.83%	40.63%
ChatGLM2	Mmlu	42.52%	45.46%

ChatGLM3	Mmlu	52.76%	61.40% (base)
ChatGLM1	M3KE	41.97%	23.60% (base)
ChatGLM2	M3KE	48.17%	
ChatGLM3	M3KE	55.21%	

图 4-10 直观反映了本次评估在多个评估平台和对应的数据集上的表现：

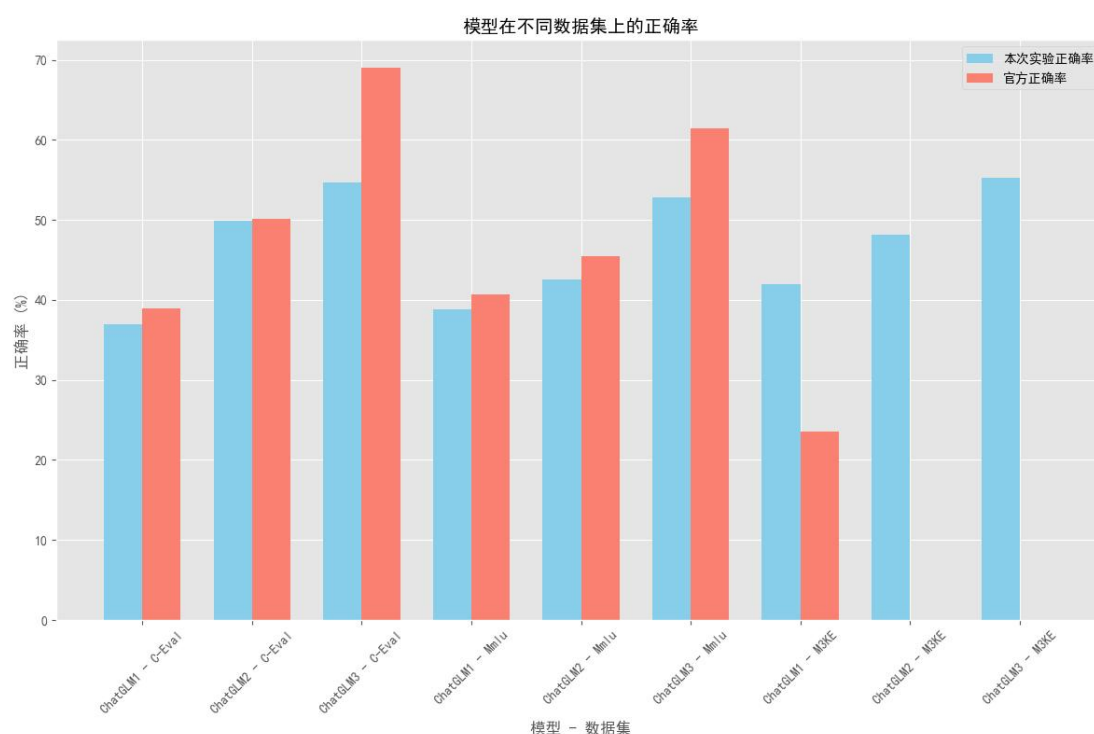


图 4-10：不同模型在评估数据集上和官方正确率比较

主观题方面，因为主观题答案上通常涉及到人类主观感受、情感、态度等方面，评估需要能够理解和包容不同的观点和答案。现阶段的评估方式难以保证评分的一致性和客观性。因此本设计仅展示并记录由大语言模型生成的主观题答案。如图 4-11、4-12、4-13 所示：

```
问题: not ( True ) and ( True ) is
源问题解释:
源问题答案: True
chatglm答案: not not ( not ( False ) ) = not not ( A ) = not not ( True ) = not not False = True
not ( True ) and ( True ) = A and B = True and True = True
chatglm解释: Let's think step by step.
Remember that (i) expressions inside brackets are always evaluated first and that (ii) the order of operations from highest priority to lowest priority is "not", "and", "or", respectively.
We first simplify this expression "Z" as follows: "Z = not ( True ) and ( True ) = A and B".
Let's evaluate A: A = not True = not False = True.
Let's evaluate B: B = True.
Plugging in A and B, we get: Z = A and B = True. So the answer is True.
```

图 4-11: BIG-bench 评估结果展示

```
问题: 7. ( 9分) 将两种大鼠分为 A、B两组, A组大鼠除去淋巴细胞后, 产生抗体的能力丧失; 从B组大鼠中获得淋巴细胞并转移到 A组大鼠后, 发现 A组大鼠能够重新获得产生抗体的能力。请回答:
(1) 上述实验可以说明 是免疫反应所需的细胞。
(2) 为了证明接受了淋巴细胞的 A组大鼠重新获得了产生抗体的能力, 需要给 A组大鼠注射 , 然后检测相应的抗体。
(3) 动物体内能产生特异性抗体的细胞称为 。在抗体、溶菌酶、淋巴因子和编码抗体的基因这四种物质中不属于免疫活性物质是 。在吞噬细胞、淋巴细胞和红细胞这三类细胞中不属于免疫细胞的 。

源问题解释: 解: ( 1) 除去淋巴细胞不能产生抗体, 移入淋巴细胞后重新获得产生抗体的能力, 说明淋巴细胞是机体进行免疫反应所需的细胞。
(2) 只有在抗原的刺激下, 通过体液免疫形成的浆细胞才能产生抗体。
(3) 在体液免疫中, 抗体是由浆细胞( 效应 B淋巴细胞 ) 产生并分泌的。免疫系统的组成中, 免疫活性物质包括抗体、溶菌酶、淋巴因子等, 免疫细胞包括吞噬细胞和淋巴细胞。编码抗体的基因和红细胞不属于免疫系统的组成。

源问题答案: (1) 淋巴细胞( 或B淋巴细胞)
(2) 抗原
(3) 浆细胞( 或效应 B淋巴细胞) 编码抗体的基因 红细胞

chatglm答案:
(1) 淋巴细胞是免疫反应所需的细胞。
(2) 为了证明接受了淋巴细胞的A组大鼠重新获得了产生抗体的能力, 需要给A组大鼠注射抗体, 然后检测相应的抗体。
(3) 动物体内能产生特异性抗体的细胞称为浆细胞。在抗体、溶菌酶、淋巴因子和编码抗体的基因这四种物质中不属于免疫活性物质是编码抗体的基因。在吞噬细胞、淋巴细胞和红细胞这三类细胞中不属于免疫细胞的是红细胞。
chatglm解释: (1) 根据题意, A组大鼠在除去淋巴细胞后, 产生抗体能力丧失, 说明淋巴细胞是免疫反应所需的细胞。

(2) 为了证明接受了淋巴细胞的A组大鼠重新获得了产生抗体的能力, 需要给A组大鼠注射抗体, 然后检测相应的抗体。

(3) 动物体内能产生特异性抗体的细胞称为浆细胞。在抗体、溶菌酶、淋巴因子和编码抗体的基因这四种物质中不属于免疫活性物质是编码抗体的基因。在吞噬细胞、淋巴细胞和红细胞这三类细胞中不属于免疫细胞的是红细胞。
```

图 4-12: GAOKAO 主观题结果展示

```
问题: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?
源问题解释: Janet sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day. She makes 9 * 2 = $<<9*2=18>>18 every day at the farmer's market.
### 18
源问题答案: 18
chatglm答案: 22.
chatglm解释: First find the number of eggs Janet buys for her friends: 3 eggs / person \* 4 people = 12 eggs. Then subtract the number of eggs she eats and sells to find the number of eggs she has left: 16 eggs - 12 eggs - 3 eggs = 11 eggs. Then multiply the number of eggs by the price per egg to find the total amount of money she earns selling the eggs at the farmers' market: 11 eggs \* $2 / egg = $<<11\*2=22>>22.
```

图 4-13: GSM8K 主观题结果展示

5. 总结与展望

5.1 全文总结

本文的主要研究对象是大语言模型评估平台的对比与评测，其中主要收集和利用不同平台对应的评估数据集制作统一评估数据集，实现多平台的共同测评。为此研究了大语言模型的基石——Transformer 和 GPT 的原理与发展历程，以及大语言模型“幻觉”产生的定义和根本原因。同时收集大语言模型评估数据集并编写 python 脚本，针对各个评估数据集的特征将评估数据集格式统一化。并利用 NLP 计算类别相似度对每条评估数据进行重分类。在完成开源大语言模型部署后，开发 python 程序在服务器上实现多个评估数据集的自动化评估并将结果记录保存。并最终将结果进行可视化和对比，可辅助大语言模型开发人员针对大语言模型进行评测，提高开发的效率和针对性。

本文完成的工作主要有以下方面：

- 1、对大语言模型本身和“幻觉”产生的原理进行了研究。

初步研究了大语言模型基本原理、改进原理和发展历史，然后基于大语言模型根本原理，针对大语言模型存在的指令、上下文和逻辑不一致三类幻觉进行研究，为评估数据集制作做了铺垫。

- 2、研究了常见的评估数据集的评估流程和格式，并统一了数据集格式并进行数据集重分类。

本文调研了国内外常见的评估平台和对应的评估数据集，编写 python 脚本读取各个数据集并针对每个数据集的特性统一数据集格式，同时利用 spaCy 将评估数据集的类别重新分为四大类，方便后续对大语言模型的评估结果进行针对性分析。

- 3、对开源大语言模型进行部署，研究大语言模型评估流程，编写 python 脚本统一代码。

本文研究了大语言模型部署和评估流程，经过调研决定以 ChatGLM 的三代作为主要研究对象，着重研究其 api 和评估方法，针对本文设计

的统一的数据集格式进行读取、评估并统计结果，最终在服务器上部署算法并进行自动化评估和记录。

4、开发大语言模型评估结果可视化程序。

结合评估记录的结果，计算相应的评估指标，并利用 python 中的 `Json`、`matplotlib.pyplot` 库绘制统计图。同时利用绘制结果对开源大语言模型的多个方向与领域进行分析。程序主要绘制了直方图、柱状图和折线图等方面对结果进行分析，达成本设计最终目的。

5.2 工作展望

总的来说，本文设计的 Python 代码自动化评估工具和评估平台达到了预期的评估大语言模型的效果，但是依然存在值得改进的地方。

首先，本自动化评估工具的数据集格式略微冗余，所造成的额外的空间开销上可以进行优化，目前在对每个源评估数据集进行统一化之后，评估数据集的存储空间会膨胀 1-2 倍。后续可以优化 `Json` 文件的结构，减少文件大小提高大语言模型评估效率。

其次，可以增加评估大语言模型的评估指标。本文的题目指标主要集中于客观题中常用的正确率，虽然这种评估指标是最能直接的反映出大语言模型的性能。且有着较好评估和容易编写自动化代码等优点。但是这种指标难以对大语言模型的创造性等进行评估，同时也无法评估主观题的答案，虽然本设计有编写利用另一个大模型对主观题进行打分的脚本，但是结果可靠性欠佳。

最后，虽然本文得到了大语言模型的评估结果，但是在结果展示上可以采用更多的展示指标。利用更多的评估指标来从多个不同角度反应大语言模型能力，也方便对大语言模型进行评估分析。虽然本文提供了大语言模型回答错误的题号，并且记录了大语言模型在错题上的思路，但是并没有展示出其中的错误原因，望将来能进一步的研究如何提高大语言模型的解题思路。

参考文献

- [1] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., & Mian, A.S. (2023). A Comprehensive Overview of Large Language Models. ArXiv, abs/2307.06435.
- [2] Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent Neural Network Regularization. ArXiv, abs/1409.2329.
- [3] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Neural Information Processing Systems.
- [4] Yenduri, Gokul et al. "GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions." IEEE Access 12 (2023): 54608-54649.
- [5] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., & Wen, J. (2023). A Survey of Large Language Models. ArXiv, abs/2303.18223.
- [6] Huang, Y., Bai, Y., Zhu, Z., Zhang, J., Zhang, J., Su, T., Liu, J., Lv, C., Zhang, Y., Lei, J., Fu, Y., Sun, M., & He, J. (2023). C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. In Advances in Neural Information Processing Systems.
- [7] Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and Applications of Large Language Models. ArXiv, abs/2307.10169.
- [8] YLI-JYRÄ, A., KORNAI, A., & SAKAROVITCH, J. (2011). Finite-state methods and models in natural language processing. Natural Language Engineering, 17(2), 141-144. doi: 10.1017/S1351324911000015
- [9] Joshua T. Goodman, A bit of progress in language modeling, Computer Speech & Language, Volume 15, Issue 4, 2001, Pages 403-434, ISSN 0885-2308, <https://doi.org/10.1006/csla.2001.0174>.
- [10] Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.
- [11] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.
- [12] Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., & Le, Q.V. (2019).

XLNet: Generalized Autoregressive Pretraining for Language Understanding. Neural Information Processing Systems.

[13] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

[14] Brown, Tom B. et al. "Language Models are Few-Shot Learners." ArXiv abs/2005.14165 (2020): n.pag.

[15] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L.E., Simens, M., Aspell, A., Welinder, P., Christiano, P.F., Leike, J., & Lowe, R.J. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

[16] Feng, Zhangyin et al. "Trends in Integration of Knowledge and Large Language Models: A Survey and Taxonomy of Methods, Benchmarks, and Applications." ArXiv abs/2311.05876 (2023): n.pag.

[17] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E.H., Xia, F., Le, Q., & Zhou, D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. ArXiv, abs/2201.11903.

[18] Papineni, Kishore et al. "Bleu: a Method for Automatic Evaluation of Machine Translation." Annual Meeting of the Association for Computational Linguistics (2002).

[19] Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries." Annual Meeting of the Association for Computational Linguistics (2004).

[20] Banerjee, Satanjeev and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments." IEEvaluation@ACL (2005).

[21] Roh, Jihyeon et al. "Unigram-Normalized Perplexity as a Language Model Performance Measure with Different Vocabulary Sizes." ArXiv abs/2011.13220 (2020): n.pag.

[22] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21.

[23] Huang, Yuzhen et al. "C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models." ArXiv abs/2305.08322 (2023): n.pag.

[24] Liu, Chuang et al. "M3KE: A Massive Multi-Level Multi-Subject Knowledge Evaluation Benchmark for Chinese Large Language Models." ArXiv abs/2305.10263 (2023): n.pag.

[25] Zhang, X., Li, C., Zong, Y., Ying, Z., He, L., & Qiu, X. (2023). Evaluating the Performance of Large Language Models on GAOKAO Benchmark. ArXiv, abs/2305.12474.

- [26] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D.X., & Steinhardt, J. (2020). Measuring Massive Multitask Language Understanding. ArXiv, abs/2009.03300.
- [27] Cobbe, Karl et al. "Training Verifiers to Solve Math Word Problems." ArXiv abs/2110.14168 (2021): n.pag.
- [28] Srivastava, Aarohi et al. "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models." ArXiv abs/2206.04615 (2022): n.pag.

致谢

本论文的工作是在我的导师黄炳顶的悉心指导下完成的，本文工作中黄炳顶老师严谨的治学态度和科学的工作方法给了我极大的影响，同时黄炳顶老师的耐心和给予我的信任、帮助与鼓励，在论文开题及后续的各个阶段都提供了指导和交流，在此衷心感谢黄炳顶老师对我的关心和指导。

在编写论文期间，感谢实验室师兄们，在开发项目中提出中肯的建议和讨论，帮助我拓展了思考和学习方式，学会了从不同的角度去理解看待问题。

感谢身边的朋友们，始终以乐观积极的心态感染着我，让我成为更好的自己。因为有你们的陪伴和关心，给予了我的勇气和力量。

最后再次感谢能有机会跟随黄老师继续一年半的实习、项目实践与论文。黄炳顶老师的耐心也非常值得我学习。

感谢家人，没有你们的支持、理解和爱，就不会有我的成长与蜕变。