

1. 常用大模型介绍

- [DeepSeek](#): 由中国深度求索 (DeepSeek) 公司开发, 目前仅支持文本生成任务, 不支持多模态任务, 支持工具调用。
- [ChatGLM](#): 由智谱AI公司开发, 网页和API接口均支持多模态数据, 支持工具调用。
- [KIMI](#): 由月之暗面 (Moonshot) 公司开发, 支持文本生成和多模态任务, 支持工具调用。
- [通义千问](#): 由阿里云开发, 支持文本生成和多模态任务, 支持工具调用。

2. 大模型常用API接口 (标准 OpenAI API 风格)

本次实验使用的是 DeepSeek 的 deepseek-chat 模型。

- 密钥: sk-0c12a1be7cba41d3af3c598e4d350c1c
- API请求基本地址: <https://api.deepseek.com>

注意: 密钥将于2024年11月13日16:00:00过期, 如需继续使用请自行申请密钥。

[DeepSeek开放平台](#)

[智谱 \(GLM\) 开发平台](#)

2.1 环境搭建

由于本次实验采用的 DeepSeek 模型采用标准OpenAI API风格接口, 故本次实验使用 OpenAI Python SDK, 也可以根据官方API参考文档使用requests或httpx等库发送HTTP请求。

2.1.1 使用 Anaconda 创建虚拟环境

```
# 创建虚拟环境
conda create -n llm_learn python=3.11.10
# 激活虚拟环境
conda activate llm_learn
# 安装必要的包
conda install openai
```

2.1.2 使用 venv 创建虚拟环境

```
# 创建虚拟环境
python3 -m venv llm_learn
# 激活虚拟环境 (windows cmd, 非 powershell)
call llm_learn/Scripts/activate.bat
# 安装必要的包
pip install openai
```

2.1 补全接口 (Completion)

补全接口的功能是续写用户给定的提示词, 即预测并生成后续内容, 通常应用于代码补全, 文本续写等场景。

补全接口并非所有大模型服务提供商都开放，以下补全接口实验均使用 DeepSeek 模型。

2.1.1 补全接口参数介绍

[补全接口文档](#)

- model: 模型名
- prompt: 提示词
- suffix: 被补全内容的后缀
- max_tokens: 最大生成长度
- echo: 是否返回提示词
- temperature: 温度参数，temperature 越小，模型返回的结果越确定。如果调高该参数值，LLM 可能会返回更随机的结果
- top_p: 采样概率，top_p 越小，模型返回的结果越确定。如果调高该参数值，LLM 可能会返回更随机的结果
- logprobs: 是否返回每个 token 的对数概率
- frequency_penalty: 频率惩罚参数，基于某个词在响应和提示中出现的次数惩罚下一个出现该词的概率，即：frequency_penalty 越大，重复词出现的概率越小。
- presence_penalty: 存在惩罚参数，惩罚模型生成重复的词，与 frequency_penalty 不同的是，presence_penalty 对于所有重复词的惩罚是相同的，而 frequency_penalty 是基于词频的惩罚，重复词出现的次数越多，惩罚越大。
- ...

通常 top_p 和 temperature 两个参数指定一个即可，无需同时指定。

[Efficient Training of Language Models to Fill in the Middle](#)

2.1.2 补全接口实验

使用补全接口实现代码补全功能。

例如：利用补全接口补全下列快速排序代码。

```
def quick_sort(arr):  
    ...  
  
arr = [3, 6, 8, 10, 1, 2, 1]  
quick_sort(arr)  
print(arr)
```

实验脚本见 completion.py

2.2 对话接口 (Chat Completion)

对话接口的功能是基于用户输入的对话历史记录，回答用户的问题，通常应用于智能对话助手等场景。该种接口也是应用最多的接口，网页版的智能对话助手通常都是基于对话接口实现的。

2.2.1 对话接口参数介绍

[对话接口文档](#)

- model: 模型名
- messages: 对话历史记录
 - role: 发送者角色

- content: 发送内容
 - ...
- max_tokens: 最大生成长度
- stream: 是否流式对话
- ...

2.2.2 对话接口实验

利用对话接口实现支持流式对话和多轮对话的智能助手。

实验脚本见 chat_completion.py

3. 提示工程

3.1 思维链 (Chain of Thought) 实验

Chain-of-Thought(CoT)是一种改进的提示工程技术，目的在于提升大模型LLMs在**复杂推理任务**上的表现，对于复杂问题尤其是复杂的数学题大模型很难直接给出正确答案。如算术推理、常识推理、符号推理等。CoT通过要求模型在输出最终答案之前，显式输出中间逐步的推理步骤这一方法来增强大模型的算数、常识和推理能力。

- 问题1：单词 Strawberry 中一共有多少个 r？
- 问题2：小明早上 8:00 离开家，以每小时 40 公里的速度行驶。早上 8:30，他停下来 10 分钟加油。然后他以每小时 60 公里的速度行驶。如果目的地距离 120 公里，他什么时候到达目的地？

[Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#)

[Large Language Models are Zero-Shot Reasoners](#)

2.1 标准提示词 (无思维链)

单词 Strawberry 中一共有多少个 r？

- 系统提示词：You are a helpful assistant.
- 用户提示词：单词 Strawberry 中一共有多少个 r？
- 模型回答：

单词 "Strawberry" 中一共有 **2** 个 "r"。

显然这个回答是错误的。

实验脚本见 cot/standard.py

2.2 Zero-Shot-CoT 提示词

Zero-Shot-CoT (Zero-Shot Chain-of-Thought) 是一种在无需示例的情况下，通过提示大型语言模型逐步推理以解决复杂问题的技术。具体而言，只需在问题后添加提示语句，如“让我们一步一步地思考”，模型即可生成详细的推理过程，从而提高对复杂任务的理解和解决能力。

模型参数量小的情况下，Zero-Shot-CoT 效果可能不佳。

问题1

单词 Strawberry 中一共有多少个 r?

- 系统提示词: You are a helpful assistant.
- 用户提示词: 单词 Strawberry 中一共有多少个 r? Let's step by step.
- 模型回答:

让我们逐步分析单词 "Strawberry" 中的字母:

1. 首先, 写出单词 "Strawberry":

s t r a w b e r r y

2. 然后, 逐个检查每个字母, 看看有多少个 "r":

s t r a w b e r r y
 ^ ^ ^

3. 可以看到, 单词 "Strawberry" 中有三个 "r"。

因此, 单词 "Strawberry" 中一共有 **3** 个 "r"。

这个回答是正确的。

问题2

小明早上 8:00 离开家, 以每小时 40 公里的速度行驶。早上 8:30, 他停下来 10 分钟加油。然后他以每小时 60 公里的速度行驶。如果目的地距离 120 公里, 他什么时候到达目的地?

- 系统提示词: You are a helpful assistant.
- 用户提示词: 小明早上 8:00 离开家, 以每小时 40 公里的速度行驶。早上 8:30, 他停下来 10 分钟加油。然后他以每小时 60 公里的速度行驶。如果目的地距离 120 公里, 他什么时候到达目的地? Let's step by step.
- 模型回答:

好的, 让我们一步一步来解决这个问题。

第一步: 计算小明在第一个半小时内行驶的距离

小明在早上 **8:00** 离开家, 以每小时 **40** 公里的速度行驶。他在 **8:30** 停下来加油。

- 时间: **0.5** 小时 (从 **8:00** 到 **8:30**)
- 速度: **40** 公里/小时

行驶的距离 = 速度 × 时间 = **40** 公里/小时 × **0.5** 小时 = **20** 公里

第二步: 计算小明在加油后行驶的剩余距离

目的地距离 **120** 公里, 小明已经行驶了 **20** 公里。

剩余距离 = **120** 公里 - **20** 公里 = **100** 公里

第三步: 计算小明在加油后行驶的时间

小明在加油后以每小时 **60** 公里的速度行驶。

- 剩余距离: **100** 公里

- 速度: 60 公里/小时

行驶时间 = 剩余距离 ÷ 速度 = 100 公里 ÷ 60 公里/小时 = 1.67 小时 (约 1 小时 40 分钟)

第四步: 计算小明到达目的地的时间

小明在 8:30 开始加油, 加油花了 10 分钟, 然后继续行驶。

- 加油时间: 10 分钟 (0.17 小时)

- 行驶时间: 1.67 小时

总时间 = 0.17 小时 + 1.67 小时 = 1.84 小时 (约 1 小时 50 分钟)

小明在 8:30 开始加油, 1.84 小时后到达目的地。

到达时间 = 8:30 + 1 小时 50 分钟 = 10:20

最终答案

小明在 10:20 到达目的地。

这个回答是正确的。

实验脚本见 `cot/zero_shot_cot.py`

2.3 Few-Shot-CoT 提示词

Few-Shot CoT (Few-Shot Chain-of-Thought) 是一种在提示大型语言模型时, 通过提供少量示例, 展示逐步推理过程, 以提升模型在复杂任务上的表现的方法。与 Zero-Shot CoT 不同, Few-Shot CoT 通过在提示中加入示例, 指导模型生成类似的推理步骤, 从而提高其在复杂任务上的准确性。

问题1

单词 Strawberry 中一共有多少个 r?

- 系统提示词: You are a helpful assistant.
- 用户提示词:

问题: 单词 Banana 中一共有多少个 n?

答案: 单词 "Banana" 中一共有多少个 "n", 我们可以逐步分析:

1. 首先, 写出单词 "Banana":

- B
- a
- n
- a
- n
- a

2. 然后, 数一数有多少个 "n":

- 第三个字母是 "n"
- 第五个字母是 "n"

3. 总结一下, 单词 "Banana" 中一共有 2 个 "n"。

问题: 单词 Strawberry 中一共有多少个 r?

答案:

- 模型回答：

单词 "Strawberry" 中一共有多少个 "r"，我们可以逐步分析：

1. 首先，写出单词 "Strawberry"：

- s
- t
- r
- a
- w
- b
- e
- r
- r
- y

2. 然后，数一数有多少个 "r"：

- 第三个字母是 "r"
- 第八个字母是 "r"
- 第九个字母是 "r"

3. 总结一下，单词 "Strawberry" 中一共有 3 个 "r"。

这个回答是正确的。

问题2

小明早上 8:00 离开家，以每小时 40 公里的速度行驶。早上 8:30，他停下来 10 分钟加油。然后他以每小时 60 公里的速度行驶。如果目的地距离 120 公里，他什么时候到达目的地？

- 系统提示词：You are a helpful assistant.
- 用户提示词：

问题：小明有 5 个网球，他又买了 2 罐网球，每罐有 3 个网球。请问现在小明一共有多少个网球？

答案：小明一开始有 5 个网球。2 罐网球一共有 $2 \times 3 = 6$ 个网球。所以小明现在一共有 $5 + 6 = 11$ 个网球。

问题：小明早上 8:00 离开家，以每小时 40 公里的速度行驶。早上 8:30，他停下来 10 分钟加油。然后他以每小时 60 公里的速度行驶。如果目的地距离 120 公里，他什么时候到达目的地？

答案：

- 模型回答：

小明早上 8:00 离开家，8:30 停下来加油，行驶了 30 分钟，即 0.5 小时。在这段时间内，他以每小时 40 公里的速度行驶，因此行驶了 $40 \times 0.5 = 20$ 公里。

加油后，他以每小时 60 公里的速度行驶。目的地距离 120 公里，他已经行驶了 20 公里，因此还需要行驶 $120 - 20 = 100$ 公里。

以每小时 60 公里的速度行驶 100 公里需要 $100 \div 60 = 1.67$ 小时，即 1 小时 40 分钟。

小明在 8:30 停下来加油 10 分钟，因此他重新出发的时间是 8:40。加上 1 小时 40 分钟的行驶时间，他到达目的地的时间是 $8:40 + 1 \text{ 小时 } 40 \text{ 分钟} = 10:20$ 。

所以，小明在 10:20 到达目的地。

这个回答是正确的。

实验脚本见 `cot/few_shot_cot.py`

3. Prompt/Agentic Workflow

在与LLM交互时，单次对话通常难以完成复杂任务。因此，开发者需要精心设计工作流，使模型按照预定流程生成满足特定需求的内容。以语言翻译为例，下列实验展示了如何设计工作流，使得模型利用反思机制更好地完成翻译任务。

本示例使用三步完成翻译任务：

- 初步转译：为模型赋予翻译专家的身份，要求模型从源语言翻译为目标语言。
- 反思结果：为模型赋予翻译专家的身份，要求模型对初步转译结果进行反思，针对流畅性、准确性、风格、术语等方面提出改进建议。
- 改进结果：为模型赋予翻译专家的身份，要求模型根据反思结果对初步转译结果进行修改，以提高翻译质量。

[Reflexion: Language Agents with Verbal Reinforcement Learning](#)

[Translation Agent](#)

[Dify](#)

3.1 初步转译

- 系统提示词

```
You are an expert linguist, specializing in translation from {source_lang} to {target_lang}.
```

- 用户提示词

```
This is an {source_lang} to {target_lang} translation, please provide the {target_lang} translation for this text.
Do not provide any explanations or text apart from the translation.
{source_lang}: {source_text}

{target_lang}:
```

3.2 反思初步转译

- 系统提示词

You are an expert linguist specializing in translation from {source_lang} to {target_lang}.

You will be provided with a source text and its translation and your goal is to improve the translation.

- 用户提示词

Your task is to carefully read a source text and a translation from {source_lang} to {target_lang}, and then give constructive criticism and helpful suggestions to improve the translation.

The final style and tone of the translation should match the style of {target_lang} colloquially spoken in {country}.

The source text and initial translation, delimited by XML tags <SOURCE_TEXT> </SOURCE_TEXT> and <TRANSLATION></TRANSLATION>, are as follows:

```
<SOURCE_TEXT>
{source_text}
</SOURCE_TEXT>

<TRANSLATION>
{initial_translation}
</TRANSLATION>
```

When writing suggestions, pay attention to whether there are ways to improve the translation's:

- (i) accuracy (by correcting errors of addition, mistranslation, omission, or untranslated text);
- (ii) fluency (by applying {target_lang} grammar, spelling and punctuation rules, and ensuring there are no unnecessary repetitions);
- (iii) style (by ensuring the translations reflect the style of the source text and take into account any cultural context);
- (iv) terminology (by ensuring terminology use is consistent and reflects the source text domain; and by only ensuring you use equivalent idioms {target_lang}).

Write a list of specific, helpful and constructive suggestions for improving the translation.

Each suggestion should address one specific part of the translation.

Output only the suggestions and nothing else.

3.3 改进初步转译

- 系统提示词

You are an expert linguist, specializing in translation editing from {source_lang} to {target_lang}.

- 用户提示词

Your task is to carefully read, then edit, a translation from {source_lang} to {target_lang}, taking into account a list of expert suggestions and constructive criticisms.

The source text, the initial translation, and the expert linguist suggestions are delimited by XML tags <SOURCE_TEXT></SOURCE_TEXT>, <TRANSLATION></TRANSLATION> and <EXPERT_SUGGESTIONS></EXPERT_SUGGESTIONS> as follows:

```
<SOURCE_TEXT>
{source_text}
</SOURCE_TEXT>
```

```
<TRANSLATION>
{initial_translation}
</TRANSLATION>
```

```
<EXPERT_SUGGESTIONS>
{reflection}
</EXPERT_SUGGESTIONS>
```

Please take into account the expert suggestions when editing the translation. Edit the translation by ensuring:

- (i) accuracy (by correcting errors of addition, mistranslation, omission, or untranslated text),
- (ii) fluency (by applying {target_lang} grammar, spelling and punctuation rules and ensuring there are no unnecessary repetitions),
- (iii) style (by ensuring the translations reflect the style of the source text),
- (iv) terminology (inappropriate for context, inconsistent use), or
- (v) other errors.

Output only the new translation and nothing else.

3.4 模型响应

Source: According to our evaluations using BLEU score on traditional translation datasets, this workflow is sometimes competitive with, but also sometimes worse than, leading commercial offerings. However, we've also occasionally gotten fantastic results (superior to commercial offerings) with this approach. We think this is just a starting point for agentic translations, and that this is a promising direction for translation, with significant headroom for further improvement, which is why we're releasing this demonstration to encourage more discussion, experimentation, research and open-source contributions.

----- Initial Translation -----

Initial Translation: 根据我们在传统翻译数据集上使用BLEU评分进行的评估，这种工作流程有时能与领先的商业产品竞争，但有时也表现不如它们。然而，我们也偶尔通过这种方法获得了非常出色的结果（优于商业产品）。我们认为这只是代理翻译的一个起点，并且这是一个有前途的翻译方向，有显著的改进空间，这就是为什么我们发布这个演示，以鼓励更多的讨论、实验、研究和开源贡献。

----- Reflection -----

Reflection: 1. 将“根据我们在传统翻译数据集上使用BLEU评分进行的评估”改为“根据我们在传统翻译数据集上使用BLEU评分进行的评估结果”，以提高句子完整性和准确性。

2. 将“这种工作流程有时能与领先的商业产品竞争，但有时也表现不如它们”改为“这种工作流程有时能与领先的商业产品相媲美，但有时也表现不如它们”，以更准确地表达原文的竞争性。

3. 将“我们也偶尔通过这种方法获得了非常出色的结果（优于商业产品）”改为“我们也偶尔通过这种方法获得了非常出色的结果（优于商业产品）”，以避免重复“通过这种方法”。

4. 将“我们认为这只是代理翻译的一个起点”改为“我们认为这只是代理翻译的起点”，以简化句子结构，使其更符合中文表达习惯。

5. 将“并且这是一个有前途的翻译方向，有显著的改进空间”改为“这是一个有前途的翻译方向，具有显著的改进空间”，以提高语句的流畅性和准确性。

6. 将“这就是为什么我们发布这个演示，以鼓励更多的讨论、实验、研究和开源贡献”改为“因此，我们发布这个演示，以鼓励更多的讨论、实验、研究和开源贡献”，以更自然地表达因果关系。

----- Improved Translation -----

Improved Translation: 根据我们在传统翻译数据集上使用BLEU评分进行的评估结果，这种工作流程有时能与领先的商业产品相媲美，但有时也表现不如它们。然而，我们也偶尔获得了非常出色的结果（优于商业产品）。我们认为这只是代理翻译的起点，这是一个有前途的翻译方向，具有显著的改进空间。因此，我们发布这个演示，以鼓励更多的讨论、实验、研究和开源贡献。

实验脚本见 `translation_agent.py`

4. 工具/函数调用 API

工具调用（Function Calling）功能允许 LLM 在生成响应时调用外部函数或API，以获取实时数据或执行特定操作。通过定义函数的名称、参数和描述，模型可以根据用户输入智能地决定是否调用相应的函数，并生成符合函数签名的 JSON 对象

下列实验展示了如何使用 LLM 调用查询天气的工具来获取天气信息。

4.1 了解工具/函数调用参数

对于支持工具调用的模型，通常在对话补全接口中传入 `tools` 参数，该参数包含所有可用的工具定义（参考下列 JSON Schema）。

工具调用能力需要通过微调模型来实现，所以可能某些模型不支持或工具调用能力较弱，可以参考 [ReAct 框架实现工具调用](#)。

[ReAct: Synergizing Reasoning and Acting in Language Models](#)

[API 文档](#)

4.2 定义天气查询工具

```
{
  "type": "function",
  "function": {
    "name": "get_weather",
    "description": "Get weather of an location, the user shoud supply a
location first",
    "parameters": {
      "type": "object",
      "properties": {
        "location": {
          "type": "string",
          "description": "The city and state, e.g. San Francisco, CA",
        }
      },
      "required": ["location"]
    },
  },
}
```

实验脚本见 `function_call.py`