

AI 辅助语言学研究

盛金标

jim.sheng@gmail.com

摘要： 本文探讨了人工智能辅助语言学研究的多种方法，特别是提示工程（Prompt Engineering）在大语言模型中的应用。提示工程通过结构化输入和角色扮演等方法，提升了模型的准确性和效用。文章还介绍了 AI 代码生成、智能学术团队的构建以及大语言模型的本地化应用。通过具体案例，如《上海地方志》的标点和翻译，展示了 AI 在语言学研究中的潜力和应用前景。

关键词： 提示工程、大语言模型、AI 代码生成、智能学术团队、本地化应用、GPT4All、LM Studio、检索增强生成（RAG）

一、提示工程

关于人工智能辅助语言学研究，如何高效利用大语言模型，提升模型输出的准确性，最基本的方法是**提示词设计**，或者叫**提示工程**(Prompt Engineering)。提示工程是指对输入提示进行系统化、结构化设计，以引导大型语言模型的响应，确保生成输出的准确性、相关性和连贯性(Chen 等, 2024)。

大语言模型本身是一个处于休眠状态的全能手，需要用户激活并分派任务。因此，提示工程最基本的方法是**角色扮演** (Role Playing) (Chen 等, 2024)。比如古汉语处理最基础的工作是标点和翻译。笔者长期断断续续地手工校对、标点戴侗的《六书故》，chatGPT 出来时大约完成了一半，于是尝试让 chatGPT 进行对话式标点和翻译。后来调用应用程序接口（API）自动处理《上海地方志》共一万多个文本文件。这个时候，给 AI 分配的角色是古汉语专家。图表 1 的代码，第一行大概指定你期望 AI 做什么事情。第 4 行分配给 AI 角色（“role”）的是“帮助用户处理古文的助手”。第 5 行指定具体需要完成的工作。接下来一行是输出格式。`\n\n{text}`的意思是两个分行符加一个变量，变量就是每次发送给大语言模型处理的文本材料。

图表 1 的代码用来跟 API 互动。chatGPT 等应用都有**自定义设置**，你可以在自定义中做同样的配置。

通过 GPT 处理文本，例如添加标点或翻译

```
"""
messages = [
    {"role": "system", "content": "你是一个帮助用户处理古文的助手。"},
    {"role": "user", "content": f"请为以下古文添加标点。\\n\\n{text}"},
    {"role": "user", "content": f"请为以下古文添加标点并翻译成现代汉语。输出格式如下：\\n\\n 1. 添加标点的古文：\\n[标点后的古文]\\n\\n 2. 现代汉语翻译：\\n[翻译内容]\\n\\n 古文：\\n{text}"}
```

图表 1 角色分配和格式化输入、输出

chatGPT 自定义如图 2，左边“名称”框，让你给自定义 GPT 取一个名字，在“描述、指令、对话开场白”等空格，分别指定了 AI 的角色、需要 AI 完成工作以后返回给你的内容，每一次对话你都下达同样的指令，让 AI 知道需要完成的具体工作。右边显示每次只需要把待处理的古文文

ChatGPT

古汉语标点翻译

· 上线 · @ 知道链接的任何人

创建配置

名称

古汉语标点翻译

描述

你是一个帮助用户处理古文的高手。

指令

输出格式如下：\n\n1. 添加标点的古文：\n[标点后的古文]\n\n2. 现代汉语翻译：\n[翻译内容]

对话开场白

请为以下古文添加标点并翻译成现代汉语。

知识

如果您在“知识库”上传文件，则与您的 GPT 的对话中可能会包含文件内容。在启用代码解释器后，方可下载文件

上传文件

功能

☒ 网页搜索

☒ 画布

☐ DALL·E 图片生成

☐ 代码解释器和数据分析

操作

创建新操作

预览

南汇民風樸素未分縣以前風土之紀人物之志闕如也自雍正四年析上海東南境為縣長與欽首宰是邑因朱氏鶴沙志備蔣氏分縣志稿成分建南匯縣志十六卷厥後李感胡君志能重加編輯為新志十五卷距欽志之作已六十年私家著述楊普光稿文獻之徵君子楊瑞同治十三年春秀水金君福官權知是邑則距胡氏修志已八十年上溯欽志已百四十年歷時彌久又涉遭兵亂舊聞逸典缺遺甚懼紀述之無徵思典型之尚在乃請於大府有事修縣志張塘山先生總司其事張先生古文虎本邑人善德南學於鄉里風見尤極談治金君亦好古不怠網羅塊斷緒漸與未幾金君謝事去而新興縣君思賢實來君久官江蘇或置聞官宰寧縣與南匯接壤張先生昔年亦通雅故相與踵事增訂實折衷同會大中丞吳公有續修通志之舉統飭各屬先成縣志復以松江一府命太守錢君壽圖重其成。錢君奄有三長大雅宏達君子也於是官師皆一時之選閱歲未久

1. 添加标点的古文：

南汇民風樸素，未分縣以前，風土之紀、人物之志闕如也。自雍正四年析上海東南境為縣長與欽首宰是邑，因朱氏《鶴沙志》舊，蔣氏《分縣志稿》成，分建《南匯縣志》十六卷。厥後，李感胡君志能重加編輯，為《新志》十五卷，距欽志之作已六十年。私家著述楊普光稿，文獻之徵，君子楊瑞，同治十三年春，秀水金君福官權知是邑，則距胡氏修志已八十年，上溯欽志且百四十年。歷時弥久，又涉遭兵亂，旧聞弘典，故軼遺甚，俱紀述之無徵，思典型之尚在，乃請于大府，有事修編，延張塘山先生總司其事。張先生名文虎，本邑人，善德南學，于乡里闻見尤殷。談治。金君亦好古不怠，网罗塊斷，端緒漸兴。未几，金君謝事去，而新興縣君思賢實來。君久官江蘇，咸丰间嘗宰寶慶，與南匯接壤。張先生昔年亦通雅故，相與踵事增訂，實折異同。會大中丞吳公有續修《通志》之舉，既飭各屬先成縣志，復以松江一府命太守錢君壽圖重其成。錢君奄有三長，大雅宏達君子也。于是官師皆一時之選，閱歲未久。

2. 现代汉语翻译：

南汇的民风质朴，在未设立县治之前，没有关于风土和人物的记载。雍正四年（1726年），从上海东南部划出设县，长兴的钦县成为第一任知县，借助朱氏的《鹤沙志》和蒋氏的《分县志稿》，他编了《南汇县志》共十六卷。后来，李感胡君志能重加编辑，为《新志》十五卷，距钦县志之作已六十年。私家著述杨普光稿，他完善了《南汇县志》并上之。后，李感胡君志能重加编辑，为《新志》十五卷，距钦县志之作已六十年。私家著述杨普光稿，文獻之徵，君子楊瑞，同治十三年春，秀水金君福官權知是邑，則距胡氏修志已八十年，上溯欽志且百四十年。歷時弥久，又涉遭兵亂，旧聞弘典，故軼遺甚，俱紀述之無徵，思典型之尚在，乃請于大府，有事修編，延張塘山先生總司其事。張先生名文虎，本邑人，善德南學，于乡里闻见尤殷。談治。金君亦好古不怠，网罗塊斷，端緒漸兴。未几，金君謝事去，而新興縣君思賢實來。君久官江蘇，咸丰间嘗宰寶慶，與南匯接壤。張先生昔年亦通雅故，相與踵事增訂，實折異同。會大中丞吳公有續修《通志》之舉，既飭各屬先成縣志，復以松江一府命太守錢君壽圖重其成。錢君奄有三長，大雅宏達君子也。于是官師皆一時之選，閱歲未久。

给“古汉语标点翻译”发送消息

笔者尝试将潘悟云的《汉语历史音韵学》(潘悟云, 2000)和论文翻译成英语, 音韵学术语的翻译是一个问题, 比如韵部、摄、韵、等、开合口、圆唇这些术语每次输出不统一, 另外还有人名、地名等可能需要在括号里注明汉字、拼音甚至直译, 从而增强可读性, 方便母语为非汉语的读者理解。韵部、韵摄和等, 在翻译行文中经常使用 `division`, 经过多次询问对话以后, `chatGPT` 可能会使用 `category`, `family`, `group`, `class`, `grade` 等等, 韵部、摄、韵的名称翻译有时候使用拼音, 有时候直译, 比如鱼部甚至可能被翻译成“Fish group”。音韵学术语翻译不统一甚至有点随意, 原因可能是英文音韵学论著中也并不非常严格, 为了行文统一你可以根据自己的偏好或者某个汉学家的使用偏好, 比如根据沙加尔的偏好指定“等”统一使用 `division-I`、`II`、`III`、`IV` (sìděng 四等)(Baxter & Sagart, 2014), “摄”使用 `rhyme group` 如蒲立本的偏好(Pulleyblank, 1984)。这个时候, 你给 AI 的指令使用“示例提示”(Example Prompt), 具体来说, 它结合了任务说明(Instructional Prompt)和少量示例(Few-shot Examples), 来明确传达你的需求并提供具体参考, 以确保生成内容符合预期。

图表 3 中的示例提示方法，其实采用了从少量例子中学习（Few-Shot Learning）的学习方法 (Parnami & Lee, 2022)，意思是仅从几个训练样本中学习数据中的基本模式的问题。仅仅使用几个示例（Few-shot Examples）向模型提供“上下文记忆”，让模型学会模仿示例中的逻辑

和输出格式。如果你觉得例子不足以覆盖任务的复杂性，可以提供更多 Few-shot 示例，如再加入 2~3 个不同的条目，涵盖更多类型（如古代著作、学者名、地名）。

以上关于角色扮演、结构化输出指令和示例指令等提示工程的方法，可以极大简化跟 AI 的对话，并且每次都能得到预期的返回结果。另外还有比如思维链（chain of thought, COT），一步一步的顺藤摸瓜式的，从根部开始，到枝枝叶叶结束(Tree of thought, TOT)，任何跟人类专家对话的思维方式，都可以有效地用于 AI 对话中。

二、AI 代码生成

黄仁勋曾经建议大家不要学编程，然后再次被问到这个问题，改口说：作为一种爱好，你想学当然可以。言外之意，AI 完全可以取代程序员，文理科隔行不再如隔山，人类只需要自然语言作为通用编程语言，跟机器对话，因此文理科的隔阂将逐渐消失。在编程竞赛中，

```
# 构造 Prompt，明确翻译规则
prompt = f"""
你是一名历史语言学和汉语音韵学专家。请将以下文本翻译成英文，并按照以下格式处理
专有名词：
- 音韵学术语、学者名字、书名和地名，翻译时需按照“英文名（拼音 + 汉字）”的格式。
- 如果某些术语没有标准的英文翻译，请使用拼音转写，并加括号注明汉字。

示例：
1. 四等 -> Division-IV (sìděng 四等)
2. 王力 -> Wang Li (Wáng Lì 王力)
3. 切韵 -> Qieyun (Qièyùn 切韵)
4. 长安 -> Chang'an (Cháng'ān 长安)

待翻译文本：
{input_text}
"""
```

图表 3 示例提示

Codeforces 是一个广泛使用的在线评测平台，参赛者的表现通常基于解题正确率和提交的时间进行评分。这个指标用于衡量大语言模型在算法和编程问题上的能力。比如最近推出的开源 DeepSeek 模型，在 Codeforces 竞赛问题上的表现出色，其得分相比于人类参赛者超过了 51.6% 的参赛者(DeepSeek-AI 等, 2024)。做人文科学的，接受多年的教育和训练，逻辑思维能力应该不亚于理工科，文科生可以使用自然语言进行编程，而不需要人工写代码，那么文理科的界线开始模糊起来了。

一些日常工作，需要编程处理，大都只需要只需要告诉 AI 你想要做什么，比如给一个目录中的 1000 个文本文件删除多余的空行。

笔者采用“语保工程”温州地区十一个方言点的记录数据作为实验语言材料，首先使用历史比较语言学工具 LingPy 测算各方言点之间的距离(List, 2014)，然后采用 AI 的建议，使用列文斯泰因距离测算(Levenshtein, 1966)，并且将列文斯泰因编辑距离映射到一个时间轴上，从而大致推算方言分化的时间点，由 AI 读取方言原始数据表并生成代码，自动完成计算过程，然后生成方言距离数据矩阵的热图和谱系树。这两种方法分析语言同源关系以及亲疏距离所得到的结果类似，AI 计算结果更符合个人直观语感(Sheng, 2024a)。

第一节提到调用大语言模型的应用程序接口（API）批处理标点和翻译，笔者处理《上海地方志》的标点和现代汉语翻译。不再使用复制粘贴进行对话，而是调用大语言模型 API 批处理，让 AI 二十四小时运行干活（代码见附件）。

现在的开发环境也非常智能化，语言学家只需要具备 Python 和命令行操作的基本知识，知道使用 pip 安装 python 库，熟悉 Visual Studio Code 界面。Python 代码简单易懂，调试和出错基本上也只需要把错误提示复制粘贴给 AI，一般能很快找到问题所在。VS Studio 整合 Copilot，或者谷歌的 Colab 整合 Gemini 等等，根据自然语言提示编程，调试和出错结果替换修改错误代码。

第一节还提到 AI 能从少量例子中学习，那么需要编程，同样给 AI 提供少量例子处理结构化很强的文本。比如，谢思泽的《因音求字》是一部清代永嘉方言词典(谢思泽, 2013)，需要转换成数据库格式的话，我们可以提供给 AI 少量词条，并且指令输出的结果，这里不转贴代码和输出结果，有兴趣的可以自己测试生成：

我想把以下文本文件转换成数据库格式：[十一]表示声母。罷：表示代表字。然後“過罷、欲~不能”是例詞或詞義解釋。“上聲，又去聲”是這個字的聲調。（蟹）是這個字所屬韻部。增加一個備注字段，把不能歸類的放進備注裏。把代碼寫成讀取文件，轉換成數據庫，保存數據庫文件。依此類推。

[十一]罷：過罷、欲~不能，上聲，又去聲（蟹）。𩚑：部板切，比目魚。

[十二]買：買賣。挽：~回。輓：~歌，送死者。

[十三]斬：殺也（謙）。琖：大杯（潛）。盞：同上。□：~借。鄭：四里地，又去聲（旱）。醴：酒~（潛）。

[十四]剗：~頭。剗：~度，削平也（潛）。羴：羊相廁（潛）。𩚑：炙肉器（潛）。

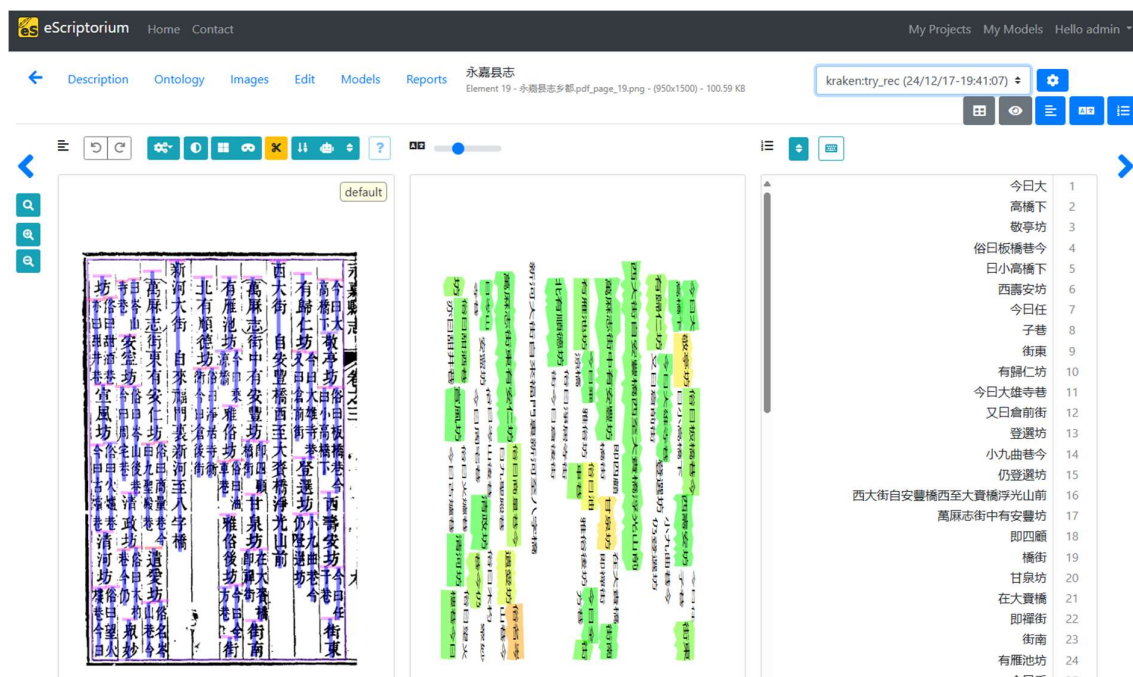
[十五]賺：本音遣，俗作~折之~，音棧。𩚑：平聲，同。湛：~露之~，平聲；~恩之~，上聲。𩚑：同豸（蟹）。嶼：~岩，又平聲（賺）。𩚑：魚~，浮沉貌，又水聲（賺）。

图表 4 从少量示例中学习，将结构化较强的文本转换成数据库

三、智能学术团队

笔者有幸参与潘悟云带领的《中华地方志深度开发项目》，意识到 AI 可以辅助、增强甚至取代传统学术团队成员，AI 完全有潜力进行传统学术团队智能化转型。一项发表于 2024 年的研究表明，生成式 AI 提升团队绩效并减少对传统团队的需求。该文探讨了生成式 AI 在团队协作中的作用。研究发现，配备生成式 AI 的团队在整体质量、创新性和实用性等关键指标上，表现优于纯人类团队。此外，个体与 AI 的配对在某些情况下能够达到与传统团队相当的绩效，这表明在特定情境下，AI 可以部分替代传统团队的功能(Li 等, 2024)。

中华地方志深度开发，最基础的工作是数字化，将扫描的古籍图像进行 OCR 文本识别，笔者在 chatGPT 指点下，测试比较了 Tesseract，谷歌云视觉，百度飞桨 OCR，Kraken，Escriptorium 等 OCR 工具。最终决定采用 Escriptorium 进行图片预处理、图片分割、文本识别和模型训练(Kiessling 等, 2019)。古代地方志文本识别的一个难点是竖排双行排版方式和手写体，这项工作正在进行之中，初步的少量手工标注和模型训练结果显示，竖排双行和手写体文本识别率得到显著提高。



图表 5 Escriptorium 文本光学识别（OCR）工具

古代地方志中，舆地志往往有手绘地图，这种地图缺乏科学测量，在 chatGPT 指点下，笔者使用 QGIS 软件将古代手绘地图根据谷歌地图经纬度映射重叠，并且尝试叠加方志地理知识库层。浙江省永嘉县（包括部分温州市区）在光绪年代的建制单位是“乡、都”，现在唯一保留的地址是“七都岛”。笔者在整理永嘉方言字典《因音求字》的时候，注意到作者谢思泽都根据“乡、都”描述地方口音的差别，他的故乡在五十都，与笔者家乡相同(Sheng, 2024b)。有这些建置材料，并且跟现代经纬度确定的地图映射重叠，就能一目了然。QGIS 软件的界面复杂，非专业第一次使用，没有任何头绪。但是在 chatGPT 指点下很快就做出来了。（图表 6）

在第一节中还提到调用 OpenAI 的应用程序接口（API），自动标点和翻译《上海县志》总共二十七种一万多个文本文件，每个文件大约八百字左右，总共在一个星期内完成，包括让 AI 写代码、调试和运行。目前中华地方志深度开发项目收集了九千多种县志，手工处理将会是一个浩大的工程。这项工作正在进之中，但是从经验角度，我们可以预见到配备 AI 的智能学术团队，可以大幅度缩减传统团队成员，极大加速项目进度。



图表 6 永嘉地方志手绘地图乡都建置

四、大语言模型本地化

最先进的 LLMs 需要昂贵的基础设施，往往通过限速、地理锁定和固定的网络界面访问，比如用户必须支付昂贵的月费，中国大陆用户无法直接访问 chatGPT，无法检索大量本地文档并生成对话，无法满足需要高准确性且能够动态回答复杂问题的场景。《上海地方志》共有 11300 个文本文件，如何进行本地嵌入（embeddings）和调用，检索增强生成（Retrieval-Augmented Generation, RAG）技术整合本地数据库，并生成这种知识密集型和特定领域的智能对话，已成为一种很有前途的解决方案。(Gao 等, 2024)

整合本地文档并生成对话的一个工具是 GPT4ALL(Anand 等, 2023)。GPT4ALL 是开源的本地模型，可以处理大量的本地文档，生成嵌入向量数据库，GPT4All 支持超过 35 种不同的模型，还提供了不同编程语言（如 Python）的高级 API 接口，方便开发者使用这些模型，比如使用编程批处理古文标点和翻译等任务。

用上海县志未经预处理的文本测试，效果并不是很理想。也尝试结巴分词（Jieba）、SpaCy

```
# 从 gpt4all 模块导入 GPT4All 类
from gpt4all import GPT4All

# 初始化模型，此处以 Meta-Llama-3-8B-Instruct 为例，下载模型文件，大小约 4.66GB，设备为 GPU。
model = GPT4All("Meta-Llama-3-8B-Instruct.Q4_0.gguf", device = 'gpu') # downloads / loads a 4.66GB LLM

# 生成文本，要求将古文翻译成现代汉语普通话。
def translate_text(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        classical_text = file.read()

    prompt = f"请将以下古文翻译成现代汉语普通话：\n\n{classical_text}"

    with model.chat_session():
        modern_translation = model.generate(prompt, max_tokens=1024) # 生成文本，最大长度为 1024 个 token

    return modern_translation

# 测试翻译功能，读取文件内容并翻译。
file_path = '测试南汇 1-10/1.txt'
translation = translate_text(file_path)
print(translation)
```

图表 7 GPT4ALL 本地 API 编程

等工具进行分词以后翻译，也没有取得明显的效果。但是先使用 chatGPT 古文标点，再调用本地 Meta-Llama 模型进行翻译，效果接近于理想，由此猜测大语言模型翻译古文，可能也依赖于对古文的标点。本地模型如果要达到 chatGPT-4o 那样的处理能力和效果，可能还需要等待一段时间，原因可能是 GPT4ALL 支持的模型不多，模型更新也比较慢。

GPT4ALL 的一个强项是对本地文件构建嵌入索引，并用于对话。本地采用 Nomic-text-embed-1.5 嵌入工具(Nussbaum 等, 2024)。从图表 7 我们可以看到，本地文档集《上海县志》总共索引了 22600 个文件（包括原文和 chatGPT 标点翻译生成的文件），有 357749 个词（也就是 tokens）。对话中调用本地文件，罗列出上海本地贞节牌坊的地点和贞节烈女的名字。并且指出这些材料来自于那几个文件，可以点开来阅读，如果想进一步了解，下面还有后续提问的建议。



图表 8 《上海县志》本地嵌入并构建索引，用于 LLM 对话

LM Studio 是另一个本地大语言模型应用平台，可以部署在本地笔记本电脑上运行大语言模型，完全离线。结合本地文档聊天，有一个聊天界面或一个兼容 OpenAI 的本地服务器来使用模型，跟 GPT4ALL 相比，有更多的模型可以选择，因为你可以从 Hugging Face 存储库下载任何兼容的模型文件，比如最近非常热门的通义千问（Qwen）和深度搜索（DeepSeek），这两个模型在处理古汉语方面比较有优势。

在测试 LM Studio 的时候，注意到给古文标点和翻译时，**文本预处理**极大影响返回的结果。《上海地方志》文本材料的开头和结尾都有许多空行，AI 自动翻译一个目录下的文件返回的处理结果中，有文件没有标点，有的有标点，似乎是随机的。但删除空行以后，能够整齐划一地返回古文标点和翻译内容。

同样的模型，API 参数设置也会极大影响模型的返回结果，比如温度（temperature）参数控制文本生成的随机性。较低的温度（例如 0.4）会使输出更加确定，较高的温度（例如 0.8）会使输出更加多样化。在古文标点和翻译的时候，一般要求翻译做到信、达、雅，当温度参数设置比较保守，那么模型在自信度不高的情况下，可能把原文输出，宁缺毋滥。在测试的时候，温度参数设置为 0.5，返回的结果有比较多的文本没有经过标点，也有翻译结果把古文原文输出。改成 0.8 甚至 0.9，标点和翻译效果明显增强，尽管有时候可能有出错或有“大语言模型幻觉”的风险。

另外还有一些参数，Top_K 采样 (Top K Sampling) 限制了每个步骤的词汇选择。通常，较高的 K 值可以提高生成文本的多样性，但对于古文翻译，我们更倾向于生成精准和流畅的翻译。推荐设置：top_k = 40 - 60

避免重复 (Repeat Penalty) 参数用于惩罚重复出现的词汇或短语, 但是对于地方志翻译, 惩罚力度过高可能效果适得其反, 原因是许多情况下只是罗列一些地名、人名、物产, 或平铺直叙的历史事件。

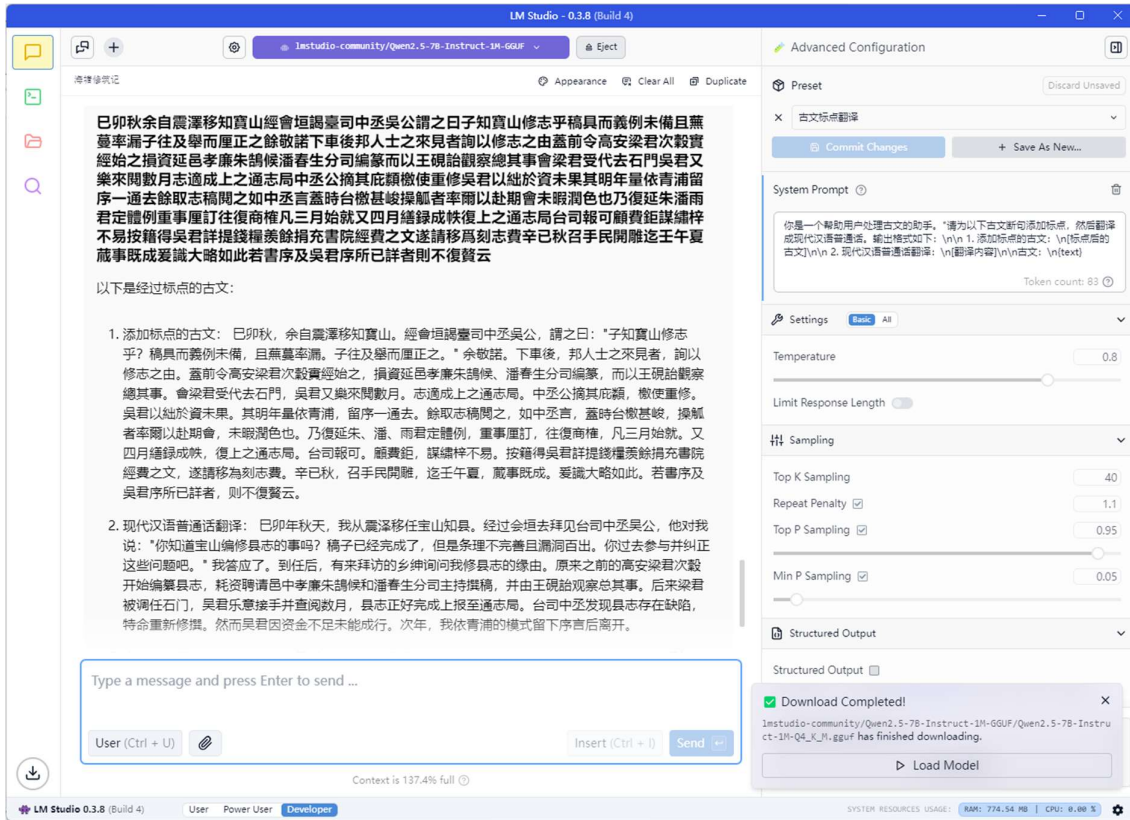
```
from openai import OpenAI # 导入 OpenAI 客户端

# 初始化 OpenAI 客户端
client = OpenAI(
    base_url="http://localhost:1234/v1", # 指向本地 LM Studio 服务器
    api_key="lm-studio" # 本地 LM Studio 的 API key
)

# 下面这个函数用于通过 GPT 处理文本, 包括指定使用的模型,
# 最大 token 数量, 输入文本较长, 需要返回标点和翻译, 超过最大 token 数量, 会返回错误。
# 温度参数控制生成文本的随机性。
# 较低的温度 (接近 0) 会使模型生成的文本更加确定和保守,
# 较高的温度 (接近 1) 会使模型生成的文本更加随机和多样化,
# 设置为 0.5, 这意味着生成的文本在确定性和随机性之间取得了平衡。
def process_text_with_gpt(text):
    """
    通过 GPT 处理文本, 例如添加标点或翻译
    """
    messages = [
        {"role": "system", "content": "你是一个帮助用户处理古文的助手。"},
        # {"role": "user", "content": f"请为以下古文断句添加标点, 然后翻译成现代汉语普通话。\\n\\n{text}"}
        {"role": "user", "content": f"请为以下古文添加标点, 确保输出的文本带有现代标点。然后翻译成现代汉语普通话。输出格式如下: \\n\\n 1. 给古文添加标点: \\n[标点后的古文]\\n\\n 2. 现代汉语翻译: \\n[翻译内容]\\n\\n 古文: \\n{text}"}
    ]
    response = client.chat.completions.create(
        model="your-model", # 使用 GPT 模型
        messages=messages,
        max_tokens=2048,
        temperature=0.8, # 控制生成文本的随机性, 较高的值提高古文标点和翻译质量
    )
    return response.choices[0].message.content # 返回处理后的文本
```

图表 9 本地 LLM Studio 服务器 API 批量标点和翻译古文

大语言模型生成许多候选词汇, 最终输出结果可能从顶部采样 (Top_P Sampling, P 是指 probability, 概率), Top_P 采样通过选择具有累积概率的最可能词汇来生成文本。较低的 Top_P 值可以保证翻译的精度, 比如 top_p = 0.9, 较高的 P 值可以增加多样性, 但对于翻译, 保持一定的控制, 以确保模型不会偏离原文的意思。相反的, Min_P 采样通常用于设置生成词汇的最小概率。如果设置为较低值, 模型可能会产生非常低概率的词汇。



图表 10 LLM Studio 本地部署大语言模型

图表 10 中可以看到，大语言模型的参数设置可以格式化输出，并且能够提高返回的质量。其中系统提示（system Prompt）设置，让模型每次都只做古文标点和翻译这一件事，而输出结果分成两部分，并且给每一部分添加带序号的标签。

五、讨论与展望

编程语言是早期人机交互的主要接口，在人工智能时代自然语言是跟大语言模型交流的主要工具。同样的，个人电脑取代大型主机（Main Frame 电脑），甚至一部手机也比过去的大型主机功能更加强大。大语言模型采用微调（Fine Tuning）、蒸馏（Distillation）等手段将越来越小型化，同时生成更加高效和精准内容。智能学术团队成员配备本地大语言模型，可以大幅度的缩减甚至取代传统学术团队，同时提高项目进度。以至于可以设想科幻级别的《中华地方志》项目的自动化流程，组合多个工具和平台，构建一个完整的系统。比如使用 Kraken OCR 进行文本识别，调用 OpenAI 或本地大语言模型进行标点、翻译，利用 SpaCy 等自然语言处理工具进行实体和关系的识别，将结果保存到 Neo4j 或 PostgreSQL 数据库，并通过 Rasa 或 Haystack 构建智能问答系统。使用 Apache Airflow 或 Celery 管理整个流程，通过 Webhook 触发各个步骤的执行，确保流程自动化、无缝连接。

潘悟云. (2000). *汉语历史音韵学*. 上海教育出版社.

谢思泽. (2013). 因音求字. 收入 *温州方言文献集成: 卷二卷本和四卷本* (第一辑). 浙江人民出版社.

Anand, Y., Nussbaum, Z., Treat, A., Miller, A., Guo, R., Schmidt, B., Community, G., Duderstadt, B., & Mulyar, A. (2023). *GPT4All: An Ecosystem of Open Source Compressed Language Models*. <https://arxiv.org/abs/2311.04931>

Baxter, W. H., & Sagart, L. (2014). *Old chinese: A new reconstruction* (No. old-chinese-a-new-reconstruction_compres; 页 16). Oxford University Press.

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2024). *Unleashing the potential of prompt engineering in Large Language Models: A comprehensive review*. <https://arxiv.org/abs/2310.14735>

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., ... Pan, Z. (2024). *DeepSeek-V3 technical report*. <https://arxiv.org/abs/2412.19437>

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). *Retrieval-augmented generation for large language models: A survey*. <https://arxiv.org/abs/2312.10997>

Kiessling, B., Tissot, R., Stokes, P., & Stökl Ben Ezra, D. (2019). eScriptorium: An open source platform for historical document analysis. *2019 international conference on document analysis and recognition workshops (ICDARW)*, 2, 19–19. <https://doi.org/10.1109/ICDARW.2019.10032>

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*.

- Li, N., Zhou, H., & Mikel-Hong, K. (2024). *Generative AI enhances team performance and reduces need for traditional teams*. <https://arxiv.org/abs/2405.17924>
- List, J.-M. (2014). *Sequence Comparison in Historical Linguistics*. Düsseldorf University Press. <https://doi.org/10.5281/zenodo.11879>
- Nussbaum, Z., Morris, J. X., Duderstadt, B., & Mulyar, A. (2024). *Nomic embed: Training a reproducible long context text embedder*. <https://arxiv.org/abs/2402.01613>
- Parnami, A., & Lee, M. (2022). *Learning from few examples: A summary of approaches to few-shot learning*. <https://arxiv.org/abs/2203.04291>
- Pulleyblank, E. G. (1984). *Middle chinese: A study in historical phonology* (1 本). University of British Columbia Press.
- Sheng, J. (2024a, 八月). *汉语方言的距离*. Zenodo. <https://doi.org/10.5281/zenodo.13256573>
- Sheng, J. (2024b, 八月). *《因音求字》再版导言和编后记*. Zenodo. <https://doi.org/10.5281/zenodo.13282890>