已被接收并将在Scientometrics发表

书目网络抓取的提示工程

Manuel Blázquez Ochando. manublaz@ucm.es 马德里康普顿斯大学
https://orcid.org/0000-0002-4108-7531

Juan José Prieto Gutiérrez. jjpg@ucm.es 马德里康普顿斯大学。
https://orcid.org/0000-0002-1730-8621

María Antonia Ovalle Perandones. maovalle@ucm 马德里康普顿斯大学。
https://orcid.org/0000-0002-6149-4724

摘要

书目目录存储着数百万条数据。运用网页抓取等计算机技术，可以高效、准确地提取数据。ChatGPT 的问世促进了适用提示词的开发，使人们能够配置抓取程序，从数据库中识别并提取信息。本文旨在探讨如何高效运用提示词工程，构建合适的数据输入模型，使其能够通过与 ChatGPT-4o 的一次交互，生成一个功能完备、以 PHP语言编写并适用于书目目录的网页抓取程序。本文以西班牙国家图书馆的书目目录为示例，该目录的数据集包含数千条记录。研究结果提出了一种有效的网页抓取程序开发模型，借助人工智能辅助，并将交互次数降至最低。模型所得结果表明，使用提示词与大语言模型（LLM）可以通过理解特定上下文和模式，并适应书目信息的不同格式和呈现风格，提升抓取质量。

关键词。提示词；数据抓取；书目目录；大语言模型；ChatGPT

引言

大语言模型（大语言模型）改变了自然语言的生成与理解方式，并在科学界获得了广泛关注与应用潜力（Zhao 等人，2023）。在取得巨大进展的背景下，了解如何利用人工智能（人工智能）的能力，在所委托的任务中取得最佳结果至关重要。这就是提示词设计，换言之，即用户输入的文本，其中包含用户在查询中提出的指令或问题。提示词设定对话的上下文，告诉大语言模型哪些信息至关重要，以及期望的输出、形式、转换和内容应当是什么（Qi 等人，2023）。歧义、偏见强化、过拟合、缺少上下文以及伦理考量，都是需要重点关注的问题，

信息专业人员必须应对这些挑战，以免得到不完整的、错误、不准确或严重误导的回答。这意味着要生成经过优化、结构清晰的提示词，以准确体现需求、任务、方法或人工智能行动必须经历的工作阶段。事实上，Gao 等人（2023）指出，ChatGPT 的表现高度取决于消息的风格和形式组织，以及语义优化（Zhu, X., Kuang, Z., & Zhang, L. 2023）。

因此，提示词必须按照一种方法来设计，而这套方法仍在科学界的探索发展之中，有些人甚至大胆地称之为“提示工程”。Liu 等人（2023）最初对其进行了研究，并在大语言模型领域推广了这一概念。这说明，我们需要培养提示词创建方面的专门能力，善用 ChatGPT 等大语言模型的功能与特性，从而与这些先进的语言模型展开更有吸引力、影响力更强的互动。这类工程实践对软件开发具有深远影响（Khojah 等人，2024年；Vaillant 等人，2024年），并为上文所述的软件编程提供支持（Xia 和 Zhang，2023）。从这个意义上说，提示工程涉及创建和调整特定指令，以引导语言模型的行为，从而获得更准确、一致的回答。这可能包括零样本提示、少样本提示、思维链（CoT）提示、自动思维链、一致性与连贯性或情感提示等技术：通过提供具体示例，让模型学会妥善应对新任务。

因此，作为一种零样本提示技术，语言模型需要在没有提供特定输入输出示例的情况下执行任务。模型利用训练过程中获得的已有知识，根据给定的指令生成回答。例如，如果要求模型翻译一个句子，却没有事先提供翻译示例，模型就会尝试运用对语言和上下文的理解来完成任务（Kojima 等，2022；Kong 等，2023）。少样本提示是指向模型提供一些输入输出示例，帮助它更好地理解需要执行的任务。这些示例为模型提供指引，从而提升其生成准确且相关回答的能力。例如，如果你想让模型判断一个句子的情感，可以给它几个带有相应情感分类的句子示例，然后要求它对一个新句子进行分类（Reynolds 和 McDonell，2021）。此外还有思维链（CoT）提示，这是一种鼓励语言模型进行逐步推理的技术。

模型并非直接生成答案，而是被引导将问题拆解为更小的步骤，并按顺序逐一解决。针对我们的案例，思维链（CoT）的一个示例如下：“我们需要从大学图书馆的目录中提取书目数据。请按以下步骤作答：1）识别目录卡片的模型。2）识别数据容器标签。3）设计网页抓取策略。4）用 PHP 编写程序。”人工智能模型会获得一系列具体步骤，这些步骤彼此关联，并由逻辑串联起来，从而为解决复杂问题提供清晰的步骤。这有助于让答案更有条理，也更容易解决问题。例如，遇到数学问题时，模型可以详细说明计算的每一步，而不只是给出最终结果（Wei 等，2022）。自动思维链是一种自动生成推理链的方法。该方法旨在生成多条不同的推理链，以提升模型鲁棒性并减少错误。随后会选出最具

适当的推理链。这个自动化流程使模型能够探索多种思路，从而提高答案的准确性。
Huang 等（2024）的研究便是一个例子，其中事件及其时间关系和因果关系被串联起来，作为大语言模型的输入参数。此外，一致性与连贯性技术着重于“对比式思维链”，以提升模型生成答案的一致性与连贯性。一致性指模型在整个对话或任务中保持同一推理思路的能力，而连贯性则指回答的内在逻辑和行文流畅度。这些技术有助于模型生成不仅正确，而且连贯、易于理解的回答（Nye 等，2021）。在情绪方面，情感提示是一种不仅能调节模型回答中的情绪，也能调整其语气，并使之适应不同上下文和需求的技术。这项技术让模型能够以更人性化、更富同理心的方式作答，并根据具体情境调整语气。例如，在客户支持对话中，如果模型察觉用户感到沮丧或不快，就可以采用更体谅、更平和的语气（Ul Huda 等，2024）。

尽管已有关于人工智能脚本生成的研究成果，例如 Lázaro-Rodríguez（2024）的研究，但我们的研究在三个基本方面有所不同：a) 针对书目网络抓取，开发一种结构化的提示工程方法。b) 经过优化，在单次交互中获得可运行代码。c) 使用包含超过 55,000 条记录的数据集进行广泛验证。d) 展示该方法在不同人工智能模型（ChatGPT-4o 和 Claude）之间的互操作性。

研究范围与目标

本文通过将高级提示工程技术应用于大语言模型，提出一种创新方法，革新书目目录网页抓取器的开发方式。主要目标是通过与 ChatGPT-4o 进行单次交互，实现可运行代码生成的完全自动化和高效化，从而无需多次迭代和手动完善。该主要目标将通过以下具体目标实现：

a) 实现面向图书馆、信息科学与文献学领域、尤其是书目上下文的网络爬虫程序编码全自动化。 b) 为人工智能设计一种查询方法，大幅减少获取功能完备的网络爬虫所需的交互次数。众所周知，要获得所需的软件代码，通常需要进行大量修正。这在一定程度上是因为交互不够理想，未能正确界定上下文、目的和预期结果。 c) 通过 OAI-PMH、SRU-39.50 之外的替代方法，让书目数据挖掘更易实现。研究人员并非总能拥有最有利的数据收集条件，而网页抓取可作为实现这一目的的工具。 d) 了解人工智能在使用 PHP 等语言开发网页抓取程序时的行为、逻辑或推理，从而能够轻松地在任何 Web 服务器上部署。

e) 提供一种方法，使研究人员能够快速定制并调整提示词，以用于任何替代计量学和文献计量学研究。这样，研究人员便能够创建并开发自己的数据提取工具。

方法论

要借助人工智能在 PHP 中正确开发网页抓取程序，必须使用定义明确的提示词。

该方法论基于“角色提示”和“少样本提示”的结合，这两种技术因其在代码生成任务中的已证实有效性而被选用（Yang 等，2023；Nguyen 等，2023）。角色提示能够为任务所需的专业能力提供上下文，而少样本提示则通过书目领域中的具体示例促进学习。事实证明，在专业编程任务中，这种组合优于零样本提示等其他方法（Kong 等，2023）。我们的方法力求通过尽可能充分地利用 ChatGPT 的注意力层，最大限度地减少与人工智能的交互（Vaswani 等，2017）。需要注意的是，随着与人工智能的对话不断延长，以便更好地刻画任务，注意力层可能会丢失原始任务的关键细节，从而导致产生“幻觉”的风险，并因此造成不理想的结果（Huang 等，2023；Duan、Yang 和 Tam，2024；Yehuda 等，2024；Verma 等，2023）。这迫使用户开启新的对话，并使提问流程变得复杂，最终浪费时间和精力，难以实现最初目标。人工智能必须能够识别创建程序所需的指令和任务，使程序具备良好性能，并进一步适应每个案例的具体特点和需求。为此，一些作者认为，可以通过 Markdown 对输入提示词或消息进行结构化（Atlas，2023；Pividori 和 Greene，2023；Greshake 等，2023），从而提高人工智能注意力层对任务进行解释和执行的有效性。注意力层至关重要，因为它能够让人工智能聚焦于输入中的相关部分，优先处理关键信息，并过滤掉多余内容。这种差异化的注意力能力改善了文本理解和处理，使系统能够更准确、更相关地回应提示词中的指令。正如 Vaswani 等（2017）所指出的，注意力架构对于处理数据流中的长期依赖关系至关重要，而这对于生成和理解复杂指令必不可少，例如网页抓取编程所需的指令。

在本研究中，我们将使用人工智能系统 ChatGPT-4o 和 Claude Sonnet 3.5，并按照以下阶段开展工作：

1) 网页抓取目标的选择。datos.bne.es 目录隶属于西班牙国家图书馆，因其所提供数据的独特性而被选中，即其语义范围，以及通过结构化和半结构化访问在网页上呈现数据的方式。此外，根据最新的机构报告（BNE，2021），该目录总计包含 1,740 万条书目记录、规范记录、馆藏和馆藏位置。这为展示提示词设计的重要性以及提升人工智能在此环境中的表现，提供了合适的测试平台。所述方法可应用于任何书目目录，不受国家或内容限制。

2) 控制提示词的定义：基于直接查询，编写用于抓取目标分析书目目录的网页抓取程序。这里展示一个简单查询，其形式为直接的句子或短语，并包含程序生成指令。

此提示词将用于比较随后获得的结果与高级提示词生成的结果。

3）根据软件开发领域科学文献的最新进展设计高级提示词，以获取用于指定书目目录的 PHP 网页抓取器代码库。在此阶段，目标是详细说明人工智能需要完成的任务，以便针对该案例创建最优的抓取程序。

4）定量 与定性比较分析。通过以下指标评估结果： :

- 生成可运行代码的成功率
- 获得可运行代码所需的交互次数
- 书目数据提取的准确性。
- 生成代码的执行时间与性能
- 对两种提示词（对照版和进阶版）生成的代码进行比较分析。评估基于西班牙国家图书馆目录中的 55,473 条记录样本进行，这些记录约占可用记录总数的 0.32 %。

5) 整合并验证人工智能生成的网页抓取器。本阶段旨在对设计的软件进行压力测试，目标是抓取西班牙国家图书馆目录中的前 50,000 条记录。这将有助于发现任何运行故障，以便进行讨论和分析。测试期间，依次执行 ChatGPT-4o 开发的代码，并自动调整记录标识变量。收集到的数据会被记录在 MySQL 数据库中，构成测试数据集。为便于编写循环及所需的额外数据插入指令，将使用针对 Claude Sonnet 3.5 人工智能的专用提示词，这进一步证明了本文所述提示词设计方法的有效性，以及该方法与其他人工智能的互操作性。这种方法确保任何研究人员都能复现并评估这项工作，从而确保程序的可靠性和准确性。

开发

提示词，或称人工智能输入，是获取 PHP 网页抓取器代码库的关键环节。与人工智能的互动不佳，可能导致生成的代码无法满足需求，也无法适应现有的技术支持。这种情况通常发生在大多数情况下：提示词设计简单，只包含直接提问，且没有经过规划或预先设定结构（Sahoo 等，2024；Chen, B.; Zhang, Z.; Langrené, N.; Zhu, S. 2023a）。这会让人误以为人工智能造成的失败比实际更多。根据本研究采用的方法，用于西班牙国家图书馆数据门户 datos.bne.es 案例、生成网页抓取器的控制提示词（代表典型的简单直接查询）如下：

表 1. 控制提示词，代表一个用于解决 datos.bne 网页抓取器问题的简单查询。
https://github.com/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test1-en.txt

创建数据	一个 PHP 网页抓取程序，能够从以下记录中提取书目数据 o n 西班牙国家图书馆门户网站： https://datos.bne.es/edicion/bimo0001291967.html

在这类查询中，用户默认了许多人工智能并不了解的细节。虽然用户说明了要使用的编程语言、要开发的软件类型以及目标，却没有提供足够的信息来说明上下文、数据提取方法，以及西班牙国家图书馆书目目录的具体情况。人工智能只能发挥想象，推断 PHP 应用程序将运行在本地的测试环境中，使用 Apache 类型的 Web 服务器。它还必须推断，用户提到“提取书目数据”时，意指选择标题、作者、版本、出版日期、出版者、丛书、主题等典型的书目元数据。然而，用户并未要求访问链接，分析记录“bimo0001291967”中可能使用的元数据；也没有说明可能包含哪些描述字段及其结构，因此可以假设这些信息以表格、都柏林核心元数据、XML、MARC21 或其他格式呈现。直接提出过于开放的问题，也会让人工智能拥有过大的发挥空间。这意味着它可以自行选择认为最适合从目录记录中提取所需信息的方法，而该方法未必符合用户的要求。在本例中，表2所列的控制提示词响应表明，ChatGPT 选择了“simple_html_dom.php”库，而没有采用“cURL”或“file_get_contents()”等其他常用且推荐的选项。从这个意义上说，这种自主选择的效果直接取决于知识库、示例、训练，以及注意力层在控制提示词中能够理解的内容。面对不明确的要求，人工智能往往会提供一套代码组合，这些代码针对一般的网页抓取情况进行了最优权衡。

表2. ChatGPT 针对控制提示词生成的回复。
https://github.com/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test1-en-results.php

```
<?php
// 引入 simple_html_dom 库
需要简单_HTML_文档对象模型.php';


// 西班牙国家图书馆记录的 URL
$url = 'https://datos.bne.es/edicion/bimo0001291967.html';


// 创建一个简单_HTML_文档对象模型实例
$html = 文件_获取_html($url);


if(!$html) {
    die('无法访问该网页');
}


// 提取书目数据
$标题 = $html->find('h1[property="dc:title"]', 0);
$作者 = $html->find('span[property="dc:creator"]', 0);
$date = $html->find('span[property="dc:date"]', 0);
$出版者 = $html->find('span[property="dc:publisher"]', 0);
$description = $html->find('div[属性="dc:description"]', 0);


echo "标题：" . ($title ? $title->plaintext : '不可用') . "\n";
echo "作者：" . ($author ? $author->plaintext : '不可用') . "\n";
echo "日期：" . ($date ? $date->plaintext : '暂无信息') . "\n";
echo "出版者：" . ($publisher ? $publisher->plaintext : '暂无信息') . "\n";
echo "描述：" . ($description ? $description->plaintext : '暂无信息') . "\n";
?>
```

在同一张表2中，我们可以看到，书目数据的提取是基于对西班牙国家图书馆记录 HTML 源代码的“假设性”解读进行的。代码中使用了带有“property”属性的 和 <h1> 标签，其属性值包含都柏林核心元数据的前缀和名称，用于表示书目描述。这一参数假设同样源于提示词中缺少相关信息。ChatGPT 认为，都柏林核心元数据是国家图书馆书目目录中的常见格式，但它并不确切知道该记录的实际源代码是否确实如此。基于这一判断，它给出的回答只是一种近似方案，与实际情况并不相符，因而会让用户觉得人工智能不够准确或能力不足。因此，运行控制提示词所生成的代码时，无法获取目录记录中的数据，最终会出现“错误”屏幕或内容不可用的情况。

关于使 ChatGPT 能够创建测试程序或软件所需的提示词结构，已有共识（Brown 等，2020；Yang 等，2023；Nguyen 等，2023）。应重点强调以下五个过程：a) 应设置目的和上下文部分，清楚描述问题、开发环境以及所需的详细程度。b) 应明确输入与约束，以清楚确定用户将提供哪些信息，以及在数据或内容处理过程中需要遵循哪些规则。c) 提供示例和预期结果，以简单明了的方式展示预期内容

其工作内容或所提供的指令。d) 方法论、执行人工智能所给指令或任务的步骤和流程，应准确说明并进行必要的细分，明确程序代码需要完成的工作及其工作流程。e) 定义程序输出或待准备数据的类型，或规定其输出规范。

考虑到这些基础，针对网页抓取器开发案例，高级提示词的结构如下面的表3所示：

表3。 用于生成 datos.bne 网页抓取器的高级提示词。
https://github.com/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test2-en.txt

```
**角色**
你是一名软件开发和文献学领域的研究人员。你的专长是网页抓取和PHP语言，以及 cURL 和 file{v1}get{v2}contents 函数。
抓取”和“PHP语言”，以及“cURL”和“file_get_contents”函数。


**背景与目的**
# *问题* 开发一个网页抓取器，从以下网页提取数据：
https://datos.bne.es/edicion/bimo0001291967.html
# *开发环境* 网页抓取器必须使用“PHP语言”编写，并运行在“Apache
环境”中，同时支持“MySQL”数据库。
# *详细程度* 我需要网页抓取器程序能够下载网页的 HTML 代码，并从其中包含的表格中提取
信息（标题、出版地、出版者、出版日期、形态描述或
载体形态、其他物理特征、尺寸、资料类型、符号、位置、总部），并将其存储
在 PHP 数组 "$arrayPHP" 中。


**输入与约束 **
# 编写网页抓取器时，使用以下函数“curl_init”、“curl_exec”、“curl_setopt”。
# 使用“XPath”提取标题、出版地、出版者、出版日期、物理描述或范围、其他
物理特征、尺寸、资料类型、索书号、位置和机构的数据。
# 你可能需要使用函数 "preg_match" 检查检索到的数据行是否包含我们要查找的元数据标题，
其中 <strong>XXXX</strong>  中的 XXXX 是标识我们要
提取的信息的元数据名称（标题、出版地、出版者、出版日期、物理描述或范围、其他物理特征、
尺寸、资料类型、索书号、位置、机构）。


**输入和输出示例**


# 输入示例
<tr><td class="标签行"><strong>标题</strong></td> <td>《“深邃的 Isaac”；Isaac 档案中的未刊文献》
佩拉尔·卡瓦列罗；其子安东尼奥汇编的事迹与文献；</td></tr>
 <tr><td class="标签行"><strong>出版地</strong></td><td>马德里</td></tr> ...


# 输出示例


$title="《深邃的艾萨克| Isaac 档案中的未刊文献 佩拉尔·卡瓦列罗| 事迹与
文献，由其子安东尼奥编纂》；$publishPlace="马德里";


echo " <li>$标题 - $lugarPublicacion</li> ";


$arrayPHP[]=array("title"=>'深邃的艾萨克 | 艾萨克·佩拉尔·卡瓦列罗档案中的未公开文件 | 汇编，
由其子安东尼奥整理的事实与文件', "publishPlace"=>'马德里');


**详细步骤**
```

1. 处理 $url="https://datos.bne.es/edicion/bimo0001291967.html";2. 使用 cURL 获取 HTML 代码。3. 使用 XPath 获取表格中的数据行。4. 使用 preg_match 区分元数据，并将信息添加到 PHP 数组中。5. 在屏幕上打印结果。

首先，会说明人工智能的角色，包括其表现方式以及具体技能和知识。本节的目的是让你选择与这些关键词相关的内容，用于开发任务。在本例中，指定了“软件开发”，以表明你是一名程序员，专门使用所需的特定语言“PHP”编写“网页抓取”代码。

出于具体的技术原因，明确要求熟悉“cURL”和“file_get_contents”函数：

cURL 在书目网络抓取方面具有重要优势：

- 对 HTTP 连接进行更精细的控制
- 高级 Cookie 和会话管理
- 能够处理重定向和 SSL 证书
- 更好的错误处理和超时
- 支持身份验证和自定义请求头

file_get_contents 虽然更简单，但之所以提到它，是因为：

- 它是基本抓取脚本中的常见替代方案。
- 使人工智能能够理解所需的复杂程度
- 便于与更强大的方法进行比较
- 这有助于引导人工智能找到最合适的解决方案。

在提示词中刻意提及这两个函数，能够起到隐性的技术指导作用，引导人工智能使用 cURL，因为它更适合系统性地提取大规模书目数据。随后，提示词中的具体约束进一步强化了这一偏好。

在“背景与目的”部分，我们旨在界定三个关键方面，即：a) 问题，b) 开发环境，c) 详细程度。首先是人工智能需要解决的“问题”。在本例中，问题是开发一个网页抓取器，用于获取特定网页的数据。目标网站的 URL 并非必需，但它可以提供部分所需的上下文，因为程序能够识别域名以及内容的扩展名或格式。例如，

通过处理“*datos.bne*”可以推断出这是西班牙国家图书馆的目录，这意味着抓取器的应用背景聚焦于包含书目描述元数据的目录。在“开发环境”（也称为工作环境）中，会明确说明程序的运行环境；在本例中，环境由 Apache HTTP 服务器、PHP 编译器和 MySQL 组成，也就是任何 Web 服务器的经典配置。这进一步限定了程序的开发框架。至于“详细程度”，则会说明所需的提取详细程度、需要从目标网页获取的 HTML 源代码中提取哪些元数据，以及数据的存储方式。在这一开发阶段，我们选择将数据记录为数组。

在“输入与限制”部分，列出了在代码开发过程中必须遵守的限制或要求。例如，必须使用“curl_init”、“curl_exec”、“curl_setopt”函数，以确保提取方法完全采用 cURL，而不是“file_get_contents”。这一限制是有效的，因为尽管角色也了解这种替代方法，但明确要求其使用 cURL。事实上，在获得的结果中，它舍弃了“file_get_contents”，转而使用 cURL。此外，还加入了 XPath 约束，将其作为选择包含所需元数据节点的首选方法；这些节点按照其在目标网页上出现的顺序列出。最后，还添加了一项“可选”限制，以便自由判断是否适合使用该函数来实现目标。这里指的是“preg_match”函数，用于检查或评估正则表达式模式。由于这些数据在 HTML 代码中并未通过属性或类来标识，因此需要获取包含这些数据的表格行，并检测其中一个单元格内是否出现关键词；该单元格在 XXXX 标签之间包含元数据名称，其中 XXXX 表示元数据的位置。通过这种方式，我们在人工智能中创建了一种识别模式，其中 XXXX 必须替换为向其提供的某一项元数据（标题、出版地、出版者、出版日期、物理描述或范围、其他物理特征、尺寸、资料类型、符号、位置、场所）。我们可以将这种技术称为“未知替换技术”，并认为它是人工智能更广泛推理链中的一项关键操作。正如所见，这种技术 consiste en 用具体数值替换占位符；尽管其本身是一项简单操作，却可以成为更复杂推理过程的一部分，在这些过程中，需要执行多个逻辑步骤并进行多次推导，才能达到最终目标。

尽管如此，仍有必要提供“输入和输出示例”，以便你判断需要设计哪种代码来编写网页抓取器。因此，我们提供了一个输入示例，其中应包含目标网页 HTML 源代码的样本或片段，以及相关的数据和元数据。如图所示，<粗体>标签会出现，由此可以推断或判断这是包含所需内容的行。接下来，我们针对同一输入提供输出示例：创建一个以元数据命名的变量，其中包含所需数据；显示该变量的屏幕截图；以及前文提到的数组或列表，其中存储了整条记录的信息。最后，为提示词总结代码的工作方法，也就是“详细步骤”部分，其中会详细或简略地列出程序流程必须经历的各个阶段。结果见下方的表 4。

表 4。 ChatGPT 针对高级提示词生成的响应。
https://github.co
m/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test2-en- results.php

```
<?php
// 要抓取的网页 URL
$url = "https://datos.bne.es/edicion/bimo0001291967.html";


// 使用 cURL 获取网页的 HTML 内容的函数
函数 getHtmlContent($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    $html = curl_exec($ch);
    curl_关闭($ch);
    返回 $html;
}


// 获取网页的 HTML 内容
$html = 获取HTML内容($url);


// 创建一个 新的 DOMDocument 对象
$dom = new DOMDocument();
@$dom->loadHTML($html);


// 创建一个新的 DOMXPath对象
$xpath = new DOMXPath($dom);


// 用于选择表格行的 XPath 表达式
$rows = $xpath->查询("//tr");


// 初始化用于存储数据的数组
$data = [];


// 用于清理和规范化文本的函数
函数 cleanText($text) {
    返回 trim(preg_replace('/\s+/', ' ', $text));
}


// 使用 preg_match 遍历各行并提取数据
foreach ($rows as $row) {
    $标签 = $XPath->查询("td[@class='标签行']/粗体", $row)->item(0);
    $value = $xpath->query("td[2]", $row)->item(0);

    if ($label && $value) {
        $labelText = cleanText($label->nodeValue);
        $valueText = cleanText($value->节点值);


        switch ($labelText) {
            case '标题':
                $数据['标题'] = $数值文本;
                中断;
            情况 '出版地':
                $数据['出版地'] = $数值文本;
                中断;
            情况 '出版社':
                $数据['出版者'] = $数值文本;
```

```
                break;
            case '出版日期:
                $data['出版日期'] = $数值文本;
                break;
            case '物理描述或篇幅:
                $data['物理描述'] = $数值文本;
                break;
            case '其他物理特征':
                $data['其他物理特征'] = $数值Text;
                break;
            case '尺寸':
                $data['尺寸'] = $valueText;
                break;
            case '材料类型':
                $data['材料类型'] = $valueText;
                break;
            case '索书号':
                $data['签名'] = $数值Text;
                break;
            case '位置':
                $data['位置'] = $valueText;
                break;
            case '总部':
                $data['总部'] = $valueText;
                break;
        }
    }
}

// 在屏幕上输出结果
foreach ($数据 as $key => $数值) {
    echo "<li><粗体>$key:</strong> $value</li>";
}

// 输出数组以验证数据
echo '<pre>';
打印_r($数据);
回显 '</pre>';
?>
```

如我们所见，ChatGPT 创建了一个函数，用于获取目标网页的 HTML 代码，并遵循了针对必须使用的函数所提出的所有限制。随后，它使用文档对象模型函数，并通过 XPath 获取包含相关元数据和数据的行的路径。它甚至自行考虑加入文本清理和规范化函数，这对于数据处理而言相当恰当（Chai，2023）。然而，更有意思的是 foreach 循环的编写：它会分析从数据表中提取的行，并使用 XPath 选取元数据标签及其在相邻单元格中对应的数据。不过，为了区分文本和元数据标签，它没有使用“preg_match”，而是使用“switch() {case}”函数来区分不同情况。这完全符合提示词的要求，因为它可以自行决定是否使用该函数。正如我们所见，它认为这种方法更好；我们可以推断，这是因为它实现起来更简单。这不禁让人思考人工智能在应对各种情境时自主做出最佳决策的能力；这也令人担忧，因为重复该操作并增加强制使用该函数的限制后，结果确实……

最终使用“preg_match”函数。然而，第一个版本完全满足了所表达的要求和需求。

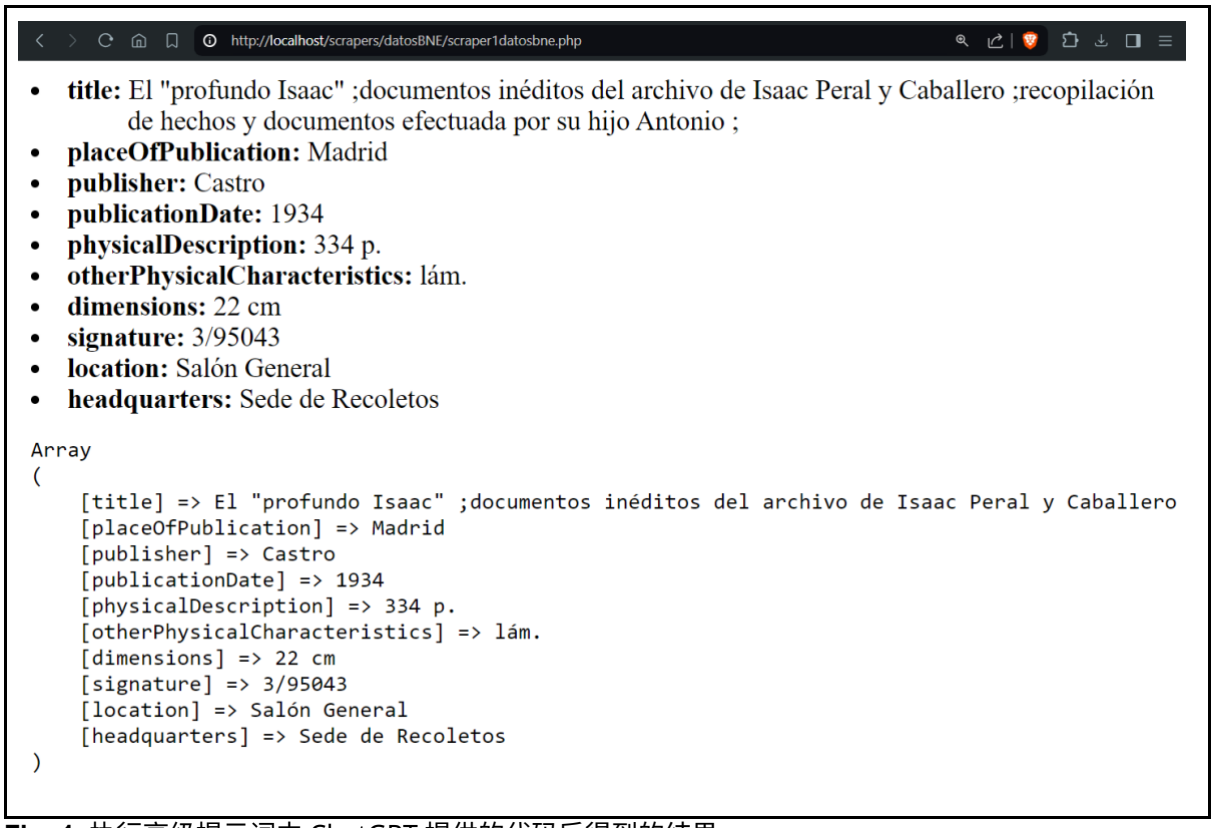


Fig. 1. 执行高级提示词中 ChatGPT 提供的代码后得到的结果 .

最后，代码输出了每项元数据及其数据，并生成了一个数组。这个数组被命名为“$data”，而不是“$arrayPHP”，但这并不算严重错误。因此，可以将这次体验总结为令人满意、非常准确且没有错误，正如图 1 中的代码执行结果所示。

接下来进行了压力测试，以巩固并验证使用 ChatGPT 获得的网页抓取器是否正常运行。测试旨在收集数据目录中的前 50,000 条记录，以检测代码可能存在的错误、遗漏或运行故障。然而，为了使网络爬取程序能够循环运行，并遍历每个目录记录标识符，需要进行以下修改：(a) 添加数据库连接设置代码，创建并验证连接；(b) 添加一个处理空值或空值的函数，以防信息来源未提供数据；(c) 创建一个主循环，遍历 BNE 数据目录中的记录；(d) 添加 SQL 插入代码，并针对与网页抓取器所收集数据类似的表结构运行该代码；(e) 每次插入后暂停 3 秒，以免服务器出现并发问题。为高效完成这些修改并确保提示方法正常运行，我们沿用了提示词的原始结构，并添加了网络爬取程序所需的修改和要求。这次与上次不同，提示词将由另一种人工智能处理，具体而言，是 Claude Sonnet 3.5。此项变更对于评估提示方法的

互操作性能力，即无论使用何种人工智能模型，都能被正确解读的能力至关重要。调整后的提示词见表 5。

表 5。在 Claude Sonnet 3.5 中执行的提示词，用于添加循环函数并将记录插入数据库，同时重复使用 ChatGPT 生成的网络爬取程序。
https://github.com/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test3-es.txt

```
**角色**
你是一名从事“软件开发”和“文献学”领域的研究人员，专长是
“网页抓取”和“PHP语言”。


**背景与目的**
# *问题* 重复使用所提供的网页抓取程序，并添加必要的 PHP 代码，使其能够
在循环中执行，以收集目录中的书目记录。
# *开发环境* 该软件使用‘PHP语言’编写，并运行在‘Apache环境’中，且
支持‘MySQL’数据库。


**条目和限制**。
- 遵循提供的代码。
- 添加数据库连接设置代码，创建并验证连接。
- 创建一个主循环，遍历 BNE 数据目录中的记录。
- 创建变量 $number，用前导零格式化主循环的计数器变量，以组成一个 10 位
前导零数字，即 https://datos.bne.es/edicion/bimo$number.html，使用 ‘str_pad’ 函数。
- 引入一个函数来处理空值或空字符串，以防数据源未提供数据。
- MySQL 表名为 ‘datosBNE’，其结构如下：‘url, 作者, 标题, 出版地, 出版者,
出版日期, 物理描述, 其他物理特征, 尺寸, 材料类型, 签名, 位置,
总部’。
- 添加 SQL 插入代码，并针对指定的表结构执行该代码，加载从网页
抓取过程中收集的信息。
- 每次插入后暂停 3 秒，以免在服务器上造成并发问题。
- 如果 $data[‘标题’] 的数值未指定或为空，则跳转到循环中的下一个 URL。

**详细步骤**
1. 在代码中添加数据库连接。
2. 创建一个主循环，遍历目录记录。
3. 管理 SQL 查询的准备和执行，以便将记录插入 MySQL。
```

可以看出，保留了“角色”“背景与目的”“输入与约束”以及“详细步骤”这一主要结构，同时删除了“输入和输出示例”部分。在本例中，最相关的是用于设定模式和必须满足条件的限制条件。首先是“遵循所提供的代码”，因为随此提示词附上了 ChatGPT 使用 PHP语言编写的网页抓取程序文件。目的是让人工智能遵循这段代码，并据此编写一个循环，以便遍历图书馆目录记录。另一方面，前一段中提到的变更也在此处说明，包括与数据库的连接、循环计数器的控制（它会修改 $url 变量，以便逐条检查目录中的记录），以及记录编号为十位数这一特殊情况。本例中，人工智能必须使用“str_pad”函数，并用双引号标明注意力层所需的关键文本或数据。另一方面，由于程序必须处理遇到空字段的情况，因此提示词指出，程序需要考虑使用某种方法或函数来处理空值。这一点很重要，因为这是一种链式推理，其关键条件是记录

数据库中的记录，即变量'$data[标题]'中获取的主字段不为空或空值。这样的串联有助于人工智能关联这两项指导，并认识到，对空值的处理决定了是否登记记录，或继续循环处理目录中的下一条记录。此外，提示词还提供了数据库表的详细信息及其结构，以便生成相应的 SQL 插入查询。此外，虽然无需特意说明，但通过指出应将网页抓取程序收集的信息插入该表并执行，进一步强化了这一 SQL 构造。这样，人工智能便能推断出，所收集的数据与数据库表中变量和字段的名称相对应。此提示词在 Claude Sonnet 3.5 人工智能中执行，用于测试本文所述的提示词构建方法是否具有互操作性。

结果再令人满意不过了。表 6 展示了 Claude 返回的源代码。可以看出，它遵循了编写数据库连接代码的准则：创建连接变量和一个 PDO 对象，以便执行与数据库的连接。另一方面，它也遵循了 ChatGPT 编写的网页抓取代码，这表明它正确理解了其中的含义，并在新代码中实现了该代码，同时加入了所述的补充内容。此外，空值或空值的处理方法也值得关注；该方法正确地整合了抓取程序的数据存储数组"$data[]"。另一方面，它遵循了有关暂停的指令，以及根据记录标题是否为空值或空值来向数据库插入记录的条件。整个代码集被正确整合在一个循环中，最多处理十位数字。

表 6。使用 ChatGPT 代码和 Claude 3.5 提供的代码生成的程序

https://github.com/manublaz/promptAI/blob/main/webScraping-biblioCatalog-test3-es-results.php

```php
<?php
// 数据库连接配置
$host = '本地主机';
$dbname = 'DB_name';
$username = 'your_数据库_用户名';
$password = 'your_数据库_密码';


// 创建连接
try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "成功连接到数据库。<br>";
} catch(PDO 异常 $e) {
    die("连接错误：" . $e->getMessage());
}


// 使用 cURL 获取网页的 HTML 内容的函数
函数 getHtmlContent($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    $html = curl_exec($ch);
    curl_关闭($ch);
    返回 $html;
}


// 用于清理和规范化文本的函数
函数 cleanText($text) {
    返回 trim(preg_replace('/\s+/', ' ', $text));
}


// 用于处理空值或空值的函数
函数 handleNullValue($value) {
    返回 ($value !== null && $value !== '') ? $value : null;
}


// 准备 SQL 插入查询
$sql = "INSERT INTO datosBNE (URL, 作者, 标题, 出版地, 出版者, 出版日期, 物理描述,
其他物理特征, 尺寸, 材料类型, 签名, 位置, 总部) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?)";
$stmt = $pdo->prepare($sql);


// 主循环以迭代记录
for ($i = 1; $i <= 9999999999; $i++) {
    $number = str_pad($i, 10, '0', STR_PAD_LEFT);
    $url = "https://datos.bne.es/edicion/bimo{$number}.html";


    // 获取网页的 HTML 内容
    $html = getHtmlContent($url);


    // 创建一个新的 DOMDocument 对象
    $dom = new DOMDocument();
    @$文档对象模型->loadHTML($html);
```

```
// 创建一个新的 DOMXPath 对象
$xpath = new DOMXPath($dom);

// 用于选择表格行的 XPath 表达式
$行 = $xpath->查询("//tr");

// 初始化用于存储数据的数组
$数据 = [
    'URL' => $url,
    '作者' => null   ,
    '标题' => null   ,
    '出版地' => null,
    '出版者' => null,
    '出版日期' => null,
    '物理描述' => null,
    '其他物理特征' => null,
    '尺寸' => null         ,
    '材料类型' => null,
    '签名' => null,
    '位置' => null    ,
    '总部' => null
];

// 遍历行并提取数据
遍历（$行 作为 $行）{
    $标签 = $XPath->查询("td[@class='标签行']/粗体", $row)->item(0);
    $value = $xpath->query("td[2]", $row)->item(0);

    if ($标签 && $数值) {
        $labelText = cleanText($label->节点值);
        $valueText = cleanText($value->节点值);

        switch ($标签Text) {
            case '标题':
                $data['标题'] = $valueText;

                break;
            case 'Lugar de publicación - 出版地':
                $数据['出版地'] = $valueText;
                break;
            case 'Editorial - 出版者':
                $数据['出版者'] = $数值Text;
                break;
            case '出版日期:
                $数据['出版日期'] = $valueText;
                break;
            case '物理描述:
                $data['physicalDescription'] = $valueText;
                break;
            case '其他物理特征':
                $data['其他物理特征'] = $valueText;
                break;
            case '尺寸':
                $data['尺寸'] = $valueText;

                break;
            case '材料类型 - 材料类型':
                $数据['材料类型'] = $valueText;
                break;
            case '索书号 - 签名':
                $数据['签名'] = $valueText;
                break;
            case '位置 - 位置':
                $数据['位置'] = $数值Text;
                break;
```

```
                case '总部':
                    $data['总部'] = $valueText;

                    break;
            }
        }
    }

    // 如果标题为空或未指定，则跳转到下一个 URL
    if (handleNullValue($data['标题']) === null) {
        continue;
    }

    // 准备要插入的数值，并处理空值
    $values = 数组_map('handleNullValue', array_values($data));

    // 执行 SQL 查询
    try {
        $stmt->execute($values);
        echo "记录插入成功：{$url}<br>";
    } 捕获(PDO 异常 $e) {
        echo "插入记录时出错：" . $e->getMessage() . "<br>";
    }

    // 暂停 3 秒
    sleep(3);
}

// 关闭连接
$pdo = 空值;
?>
```

基于这段代码，为了验证程序在真实运行条件下能否正常工作，我们进行了一项测试。为此，我们在 datosBNE 目录中设定了数据收集目标，直至达到 50,000 条书目记录。该测试的结果可在数据集https://github.com/manublaz/datasets/blob/master/dataset-datosbne-50k_2024-07-19.sql对应用于西班牙国家图书馆书目目录（datosBNE）的网页抓取程序进行性能分析，揭示了其效率的若干重要方面，同时考虑了为保护服务器完整性而实施的限制。表 6 中的程序检查了 62,786 个链接，其中 7,313 个（11.65%）返回了 404 错误。总计收集了 55,473 条记录，整个过程耗时 63 小时 3 分 12 秒，其中包括每次插入之间安排的 3 秒暂停。有效抓取时间（不包括这些暂停）约为 13.85 小时。总时间与有效时间之间的显著差异凸显了为保护服务器而采取的缓解措施所产生的影响。有效提取速率为每分钟 66.76 条记录，即每小时 4,005.60 条记录。表 7 汇总的数据反映了爬虫的实际性能，已扣除定时休息时间。

表7。 ChatGPT 和 Claude 生成的网页抓取程序的性能结果。此处所列计算所用的 SQL 查询。https://github.com/manublaz/datasets/blob/master/dataset-datosbne-50k_2024-07-19_SQL queries-es.txt

指标	数值
已审查链接总数	62.786
来自 theBNE 数据目录的 404 个链接	7.313
收集的记录总数	55.473
开始日期	2024年-07-16 15:47:57
结束日期	2024年-07-19 11:16:57
总时长	67:29:00
与该程序无关的目录服务异常数量 程序	1
异常持续时间	4:25:48
扣除异常后的总时长	（63 小时、3 分钟和 12 秒）= 227.592 秒
计划休息时间	每条记录3秒
计划休息总时长	46:13:39
有效抓取时间	（16 小时 49 分 33 秒）= 60,573 秒
执行抓取过程的理论理想时间	55,473 条记录 * 3 秒 = 166,419 秒

共检测到 5,407 次记录编号跳跃，平均每 2,3521 条记录出现一次跳跃。这些跳跃的范围为 2 到 81 条记录，标准差为 2.5433，表明 BNE 数据库的结构存在相当大的差异。除定时暂停外，插入操作之间额外的等待时间相对少见。大多数操作都在预期时间内完成，只有 14 次耗时超过 20 秒，这可能是由目录记录编号跳跃造成的。每条记录的平均完成率为 85.4301%，反映了 BNE 数据中信息的详细程度。该数值表明，平均而言，记录包含大量信息，但数据完整性仍有提升空间。

对网页抓取器效率的比较分析显示，其有效性能（计算方式为理论理想时间与有效抓取时间之比，即 166,419 / 60,573 秒）为 274.74%，而实际性能（采用类似方法计算，但使用不含异常的总时间，即 166,419 / 227,592 秒）为 73.12%。这一差异量化了所实施缓解措施的影响，主要是插入操作之间设置的 3 秒定时暂停。较高的有效效率表明，抓取器能够快速处理数据，超过预期基准速度的 2.7 倍。另一方面，73.12% 的实际效率反映了性能与服务器保护之间的权衡，说明由于有意设置的暂停，总执行时间长于理论理想时间，但仍能实现有意义且可持续的数据收集。

局限性

本研究存在一些应予考虑的局限性：

- 领域特异性：该方法主要通过西班牙国家图书馆的目录进行验证，但其原则应适用于其他书目目录。

- 技术依赖性：生成的代码仅限于 PHP 以及指定的技术栈（Apache、MySQL）。

- 大语言模型的演进：结果与所用模型的具体版本（ChatGPT-4o 和 Claude）相关。

- 性能约束：尽管 3 秒的定时暂停对于尊重服务器的承载能力是必要的，但会影响总执行时间。

讨论

本研究的发现具有重要的实践意义，显著改变了研究人员、图书馆员和文献中心的工作方式。在文献计量学和文献研究领域，研究人员过去在收集大型书目数据集时常常面临重大障碍（Robinson-Garcia 等，2017）。本研究开发的方法消除了对高级编程技能的需求，使缺乏深厚技术专长的研究人员也能开发自己的数据挖掘工具。例如，研究人工智能出版物演变的研究人员，可以轻松调整生成的代码，从国家图书馆目录中提取并分析数千条记录，只需数小时即可完成；而这项工作过去通常需要数周的软件开发，或借助昂贵的商业工具。

对于图书馆员和文献工作者而言，这对他们日常工作的影响同样重大。系统间的数据迁移是一项至关重要但往往问题重重的任务，而这一方法大大简化了该任务（Dula & Ye，2012年）。一个典型案例是大学图书馆需要将其目录与国家或国际知识库同步。所提出的方法可快速为此任务开发定制脚本，从而减少错误并

缩短处理时间。此外，它还有助于发现书目记录中的不一致之处；例如，在我们使用 BNE 目录进行测试时，系统会自动识别作者和题名著录方式的差异，从而更高效地实现数据标准化。

在无法访问或只能有限访问 OAI-PMH 等专用应用程序接口或协议的情况下，图书馆机构尤其能从中受益。研究期间，我们发现许多中小型图书馆缺乏实施商业数据交换解决方案所需的资源。我们的方法提供了一种可行且经济高效的替代方案；正如我们通过成功修改原始脚本以适用于 BNE 所展示的那样，生成的代码可以轻松调整，以适配不同的目录系统。

在文献服务的开发方面，这一方法对图书馆服务创新尤具发展前景。例如，大学图书馆可以使用经过调整的系统版本，创建自动化书目提醒服务。该服务能够每天监测各类国家目录中的新增馆藏采购，并向研究人员推送其感兴趣领域的相关出版物。由于技术和资源方面的约束，这类增值服务过去难以实现。

要实际应用这一方法论，只需一台运行 PHP 的基础 Web 服务器，这是大多数机构都已具备的最低技术条件。在测试中，平均实施时间不到一天，其中包括代码调整。凭借这种技术易用性，以及通过与人工智能进行一次交互即可生成可运行代码的能力，该方法在推动先进文献管理工具普及方面实现了重大进展。

本研究的定量结果支持以下实践意义：每分钟提取 66.76 条记录，平均完成率为 85.43%，这表明该系统确实可用于大规模实际应用。这些数据意味着，系统能够在几天内处理完整的中型目录，相较于传统的书目数据提取方法有了显著提升。

通过网页抓取自动提取书目数据，需要仔细考虑伦理和法律问题（Krotov 等，2020年）。在本研究中，我们实施了具体的保障措施，以确保负责任地使用图书馆资源。请求之间按计划暂停 3 秒并非随意安排，而是体现了我们避免给图书馆机构的服务器造成过重负担的伦理承诺。此外，我们的方法遵循书目目录的合理使用政策，限制提取频率，并避免干扰其他用户的正常访问。实施这一方法论的研究人员和从业者务必考虑三项基本原则：1）查阅并遵守各机构的数据使用政策；2）实施适当的访问频率控制机制；3）仅将提取的数据用于研究或图书馆发展。透明地开展提取工作，并记录数据收集的方法论和目的，对于维护图书馆机构与研究社群之间的信任至关重要。

结论

本文介绍了一种有效的工作方法，可在人工智能辅助下，以尽可能少的交互开发网页抓取程序。研究表明，配置正确的提示词能够从首次交互起便生成经过调试的代码，满足数据提取与处理的需求；其起点是给定的网络资源，在本例中则是一个书目目录。人工智能能够理解指令并把握任务的细微之处，在定义明确的方法论辅助下规划工作流程。这表明，采用清晰的方法和结构来定义提示词十分重要，因为人工智能能够识别这些方法和结构，并可能因此将注意力集中在它需要完成的工作的关键数据和流程上。

提示词是与人工智能交互时不可或缺的要素。在软件开发，尤其是网络爬取程序开发的背景下，建议采用以下部分：a) 角色，b) 背景与目的，c) 输入与约束，d) 输入和输出示例，e) 详细步骤。这印证了科学界的相关观察（Brown 等，2020；Yang 等，2023；Nguyen 等，2023）。

约束对于让人工智能按照操作者指定的方向运行至关重要。观察发现，人工智能会遵循所有指导原则；如果提供的是可选指导原则，它也能判断是否采纳。这表明，人工智能给出的结果受既定规则以及其他原则的影响，这些原则可能是最小化工作量、简洁性，或代码开发中的简约性。在此处的具体案例中，为避免使用正则表达式以及由此导致的大量条件结构，选择了“switch”函数，而没有使用“preg_match”。因此，在没有进一步确认或实验的情况下，人工智能或许掌握了代码设计中有关效率和有效性的规则，这对所有研究人员而言都可能是非常有用的特性。这意味着，无须精通编程，也能创建满足特定需求（或具体需求）的程序；这类需求是科学开发的典型需求，在本案例中则属于文本与数据挖掘。

提示词设计中另一个至关重要的方面是输入和输出部分。人工智能能够将上下文中提供的细节，以及需要执行的任务或指令，与提供给它的源代码联系起来。因此，如果向它展示如何处理相关信息或数据的示例，它给出正确解决方案的可能性就会更高。从理论上说，这相当于对它进行训练，使其能够解决针对特定情境定制的网络爬虫代码问题。尽管对于输入和输出示例的数量尚无共识，但人们普遍认为，更多的输入和输出示例有助于提升学习效果。不过，用户需要注意，并非所有人工智能都拥有足够大的上下文窗口，能够保留所有示例的全部细节（Chen, S.; Wong, S.; Chen, L.; Tian, Y. 2023b；Dong 等，2024）。因此，建议提供足够的示例以帮助其充分理解，数量可能在一到两个之间。本例中，一个示例就足够了。此外，示例中呈现的各种案例或例外情况也值得关注，它们往往会增加人工智能训练的难度。在这种情况下，最好在提示词的方法论或上下文指令中考虑这些因素。

提示词中的方法概述部分至关重要，因为它能确保从头到尾始终保留任务的上下文、指令和目的（Giray，2023；

Ye 等人，2023）。这能确保人工智能的注意力层在整个交互过程中保留最重要的基本原理、流程和步骤。结构良好的方法概述不仅能为当前任务提供清晰连贯的框架，还能帮助人工智能始终聚焦于关键要素，避免偏离主题或遗漏关键信息。为人工智能系统设计提示词时，必须考虑如何呈现信息，以及如何在整个过程中保持关注重点。通过加入方法概述，可以创建一种引导模型注意力聚焦于待处理要点的“锚点”。在复杂或多阶段任务中，这一点尤为重要，因为清晰性和一致性是成功执行任务的关键。此外，这种方法有助于减少潜在的困惑或错误，避免人工智能忽略任务中的重要组成部分。通过持续强化上下文和指令，它能提高回复的准确性和效率。

人工智能生成的网页抓取器已证明是用于书目数据提取的强大而高效的工具。它成功处理了 55,473 条记录，耗时仅为理论理想时间的 73.12%，并高效处理了 7,313 个 404 错误和 5,407 条记录编号跳号。每分钟提取 66.76 条记录，且每条记录的平均完成率为 85.4301%，充分体现了其良好性能。在处理速度与遵守服务器限制之间取得的平衡表明，ChatGPT 和 Claude 生成的代码兼顾了简洁性、性能和数据提取效果。这些结果彰显了人工智能解决方案在开发面向大型书目数据集的网页抓取工具方面的潜力，此类工具能够在研究人员指出的技术和伦理约束范围内实现最佳运行。

参考文献

Atlas, S.（2023）。《面向高等教育与专业发展的 *ChatGPT*：对话式人工智能指南》。https://digitalcommons.uri.edu/cba_facpubs/548

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., & Amodei, D.（2020年）。《语言模型是少样本学习者》。《神经信息处理系统进展》，*33*，1877-1901。https://dl.acm.org/doi/abs/10.5555/3495724.3495883Chai, C.P.（2023）。文本预处理方法比较。《自然语言工程》，*29*（3），509-553。https://doi.org/10.1017/S1351324922000213Chen, B., Zhang, Z., Langrené, N., & Zhu, S.（2023a）。释放大语言模型中提示工程的潜力：全面综述。arXiv预印本 arXiv:2310.14735。https://doi.org/10.48550/arXiv.2310.14735Chen, S., Wong, S., Chen, L., & Tian, Y.（2023b）。通过位置插值扩展大语言模型的上下文窗口。arXiv预印本 arXiv:2306.15595。https://doi.org/10.48550/arXiv.2306.15595

Dong, Z.,Li, J., Men, X., Zhao, W.X., Wang, B., Tian, Z., & Wen, J. R.（2024年）。探索大语言模型的上下文窗口：基于分解位置向量。arXiv预印本 arXiv:2405.18009。https://doi.org/10.48550/arXiv.2405.18009Duan, H., Yang, Y., & Tam, K. Y.（2024年）。大语言模型了解幻觉吗？对大语言模型隐藏状态的实证调查。arXiv预印本 arXiv:2402.09733。https://doi.org/10.48550/arXiv.2402.09733Dula, M. W., & Ye, G.（2012年）。案例研究：佩珀代因大学图书馆向OCLC的WorldShare迁移。《网络图书馆学杂志》，6（2），125-132。https://doi.org/10.1080/19322909.2012.677296Gao, J., Zhao, H., Yu, C., & Xu, R.（2023）。探索ChatGPT进行事件抽取的可行性。arXiv预印本 arXiv:2303.03836。https://doi.org/10.48550/arXiv.2303.03836Giray, L.（2023）。使用ChatGPT进行提示工程：面向学术写作者的指南。《生物医学工程年鉴》，*51*（12），2629-2633。https://doi.org/10.1007/s10439-023-03272-4Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T.,, & Fritz, M.（2023）。并非你所注册的服务：通过间接提示注入破坏现实世界中集成大语言模型的应用。在《第*16*届*ACM*人工智能与安全研讨会论文集》（第79-90页）。 https://doi.org/10.1145/3605764.3623985Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., & Liu, T.（2023）。大语言模型中的幻觉调查：原则、分类体系、挑战与开放性问题。arXiv预印本 arXiv:2311.05232。https://doi.org/10.48550/arXiv.2311.05232Huang, F. 等（2024年）。使用大语言模型进行事件间关系抽取的三阶段框架。载于：Luo, B., Cheng, L., Wu, ZG., Li, H., Li, C.（编）。神经信息处理。ICONIP 2023。《计算机与信息科学通讯》，第1968卷。施普林格，新加坡。https://doi.org/10.1007/978-981-99-8181-6_33Khojah, R., Mohamad, M., Leitner, P., & Neto, F.G.D.O.（2024年）。超越代码生成：ChatGPT在软件工程实践中的观察性研究。《ACM软件工程论文集》，第1卷，第FSE期。https://doi.org/10.1145/3660788Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., & Iwasawa, Y.（2022年）。大语言模型是零样本推理器。《神经信息处理系统进展》，*35*，22199-22213。https://doi.org/10.48550/arXiv.2205.11916Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., & Zhou, X.（2023）。通过角色扮演提示实现更好的零样本推理。arXiv预印本 arXiv:2308.07702。https://doi.org/10.48550/arXiv.2308.07702Krotov, V., Johnson, L., & Silva, L.（2020年）。教程：网络爬取的合法性与伦理。https://doi.org/10.17705/1CAIS.04724Lázaro-Rodríguez, P.（2024年）。PyDataBibPub：使用ChatGPT 3.5开发的Python脚本，用于自动下载西班牙公共图书馆数据。*Infonomy*，*2*（3）。https://doi.org/10.3145/infonomy.24.042

Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., & Liu, Y. (2023). 通过提示工程越狱 ChatGPT：一项实证研究。arXiv 预印本 arXiv:2305.13860。
https://doi.org/10.48550/arXiv.2305.13860西班牙国家图书馆（2021）。西班牙国家图书馆 *2021* 年报告。https://www.bne.es/sites/default/files/repositorio-archivos/memoria_BNE_2021_0.pdfNguyen Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., & Rafiq, U.（2023）。面向软件工程的生成式人工智能——研究议程。*SSRN*。
http://dx.doi.org/10.2139/ssrn.4622517Nye, M., Tessler, M., Tenenbaum, J., & Lake, B.M.（2021）。通过双系统神经符号推理提升神经序列模型的连贯性与一致性。《神经信息处理系统进展》，*34*，25192-25204。https://doi.org/10.48550/arXiv.2107.02794Pividori, M., & Greene, C.S.（2023）。人工智能辅助学术写作的出版基础设施。BioRxiv。
https://doi.org/10.1101%2F2023.01.21.525030Qi, S., Cao, Z., Rao, J., Wang, L., Xiao, J., & Wang, X.（2023）。多模态大语言模型的局限是什么？通过提示探测深入探究多模态大语言模型。《信息处理与管理》，60（6），103510。
https://doi.org/10.1016/j.ipm.2023.103510

Reynolds, L., & McDonell, K. (2021年5月). 大语言模型的提示词编程：超越少样本范式。载于 *2021*年计算机系统中的人为因素大会扩展摘要集（第 1-7 页）。
https://doi.org/10.1145/3411763.3451760

Robinson-Garcia, N., Mongeon, P., Jeng, W., & Costas, R. (2017). DataCite 作为一种新型文献计量学数据源：覆盖范围、优势与局限。《计量学杂志》，11(3)，841-854。
https://doi.org/10.1016/j.joi.2017.07.003

Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., & Chadha, A.（2024年）。大语言模型中的提示工程系统综述：技术与应用。arXiv 预印本 arXiv:2402.07927。
https://doi.org/10.48550/arXiv.2402.07927Ul Huda, N., Sahito, S.F., Gilal, A.R., Abro, A., Alshanqiti, A., Alsughayyir, A., & Palli, A.S.（2024年）。不同提示技术下细微情绪线索相互矛盾对大语言模型的影响。《国际高级计算机科学与应用杂志》，*15*（4）。
https://dx.doi.org/10.14569/IJACSA.2024.0150442Vaillant, T.S., de Almeida, F.D., Neto, P.A., Gao, C., Bosch, J., & de Almeida, E.S.（2024年）。开发者对 ChatGPT 在软件开发中影响的看法：一项调查。arXiv 预印本 arXiv:2405.12195。
https://doi.org/10.48550/arXiv.2405.12195Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I.（2017年）。注意力就是你所需要的一切。《神经信息处理系统进展》，*30*。https://doi.org/10.48550/arXiv.1706.03762Verma, S., Tran, K., Ali, Y., & Min, G.（2023）。使用认知神经网络减少大语言模型幻觉。arXiv 预印本 arXiv:2312.15576。https://doi.org/10.48550/arXiv.2312.15576

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q., & Zhou, D.（2022年）。思维链提示能够引发大语言模型的推理。《神经信息处理系统进展》，*35*，24824-24837。https://doi.org/10.48550/arXiv.2201.11903Xia, C.S., & Zhang, L.（2023）。让对话继续：使用 ChatGPT 修复 337 个漏洞中的 162 个，每个仅需 0.42 美元。arXiv 预印本 arXiv:2304.00385。https://doi.org/10.48550/arXiv.2304.00385Yang, Z., Chen, S., Gao, C., Li, Z., Li, G., & Lv, R.（2023）。基于深度学习的代码生成方法：文献综述。https://doi.org/10.48550/arXiv.2303.01056Ye, Q., Axmed, M., Pryzant, R., & Khani, F.（2023）。提示工程化提示工程师。arXiv 预印本 arXiv:2311.05661。https://doi.org/10.48550/arXiv.2311.05661Yehuda, Y., Malkiel, I., Barkan, O., Weill, J., Ronen, R., & Koenigstein, N.（2024年）。寻找真相：一种幻觉检测的审讯方法。arXiv 预印本 arXiv:2403.02889。https://doi.org/10.48550/arXiv.2403.02889Zhao, Z., Song, S., Duah, B., Macbeth, J., Carter, S., Van, M. P., 等（2023年6月）。比人类更像人类：大语言模型生成的叙事优于人类与大语言模型交错叙事。载于第 *15* 届创造力与认知会议论文集（第 368-370 页）。https://doi.org/10.1145/3591196.3596612Zhu, X., Kuang, Z., & Zhang, L.（2023）。结合语文优化的方面级情感分析提示模型。《信息处理与管理》，60（5），103462。https://doi.org/10.1016/j.ipm.2023.103462