基于大语言模型的网页内容提取方法应用研究

吕孝忠[1] 张雨荷[1] 吴其林[1]
LǙ Xiaozhong ZHANG Yuhe WU Qilin

摘 要 针对现有网页内容提取方法在准确性与泛化能力方面存在的不足，文章提出了一种基于大语言模型的网页内容提取方法。该方法融合大模型的语义理解能力，结合分层线索与页面类型适配机制，实现对多类型网页的高效结构化信息提取。在涵盖新闻、电商、博客等 7 大类共 4 100 份网页的数据集上进行实验，结果表明，所提方法在内容提取任务中的 F_1 值达到 91.8%，较基准方法提升 5.3%。尤其在跨站点场景下仍表现出优异的泛化能力，为网页信息提取提供了新的思路和方法，且更具应用价值，可为信息检索、舆情分析和知识图谱构建提供支撑。

关键词 大语言模型；提示工程；跨网站泛化；信息抽取；网页挖掘

doi：10.3969/j.issn.1672-9528.2026.04.001

0 引言

随着互联网信息的快速增长，网页内容自动提取成为信息检索、文本分析与知识发现的关键技术。传统基于规则、DOM 树结构或文本密度的方法在处理结构复杂、动态加载及多源异构网页时存在显著局限性 [1-2]。近年来，以 ChatGPT、LLaMA 等为代表的大语言模型展现出强大的语义理解与上下文推理能力，为网页内容提取提供了新的解决路径。

尽管国内外已有研究尝试结合大模型与传统方法，国内更关注轻量化与场景适配，国外则侧重于结构语义建模与伦理治理，然而在多模态融合、动态自适应与可解释性方面仍存在技术瓶颈 [3]。本文围绕非结构化与半结构化网页数据的联合建模，通过引入分层提示机制与页面类型自适应策略，提升提取精度与泛化性能，推动智能化网页内容提取技术的发展。

1 相关技术基础

网页通常以 HTML 为基础，通过 DOM 树组织内容，并借助 CSS 与 JavaScript 实现样式与交互。语义上，网页可划分为标题、导航、正文、侧边栏及页脚等模块。不同类别网页（如新闻、电商、论坛等）具有各自的内容结构与特征分布，例如新闻网页注重标题、正文与发布时间，电商网页则突出商品信息与价格参数。深入理解网页结构特性是实现高效内容提取的前提 [4]。

近年来，大规模预训练语言模型在自然语言处理任务中表现出色。大语言模型基于海量语料进行预训练，具备较强的语义理解、知识存储与上下文推理能力。主流模型如 GPT[5]、LLaMA[6] 等采用 Transformer 架构，通过自注意力机制捕捉长程依赖关系。在任务适配方面，提示工程与微调是两种主要策略，其中提示工程因灵活性强、资源消耗低而更受青睐。

传统网页内容提取方法包括基于规则的正则匹配、基于统计特征的文本密度计算，以及基于机器学习的 DOM 路径分析等 [7]。不同网站的 HTML 结构差异较大，导致模型跨站点迁移困难 [8]、泛化性较差。因人工标注成本高昂，弱监督和远程监督方法得到广泛关注 [9]。最新的结合大模型的提取方法通过指令引导与少样本学习显著提升了对复杂网页的语义理解能力。网页内容提取方法的评估指标通常涵盖准确率、召回率、F_1 值等经典度量，以及跨网站泛化性能等。

2 方法设计与实现

2.1 总体架构

本文提出的方法分为网页获取、预处理、大模型适配、后处理与结果验证等步骤，具体如图 1 所示。

2.2 网页数据预处理模块

在预处理阶段完成 HTML 清洗、噪声过滤、语义分块与结构编码。通过正则表达式与 DOM 解析器去除冗余标签，依据文本密度与视觉特征进行内容块划分，最终生成带结构标记的文本序列以供大模型处理，具体如图 2 所示。

1. 巢湖学院计算机与人工智能学院 安徽合肥 238001

[基金项目] 国家自然科学基金联合研究项目“复杂网络社会下风险舆情的发现与调控关键技术”（U23B2031）；校级科研启动基金项目“基于专利数据的中国新能源汽车产业技术合作网络分析”（KYQD-2024035）；科研创新团队项目“模式识别与智能系统”（2024AH010022）

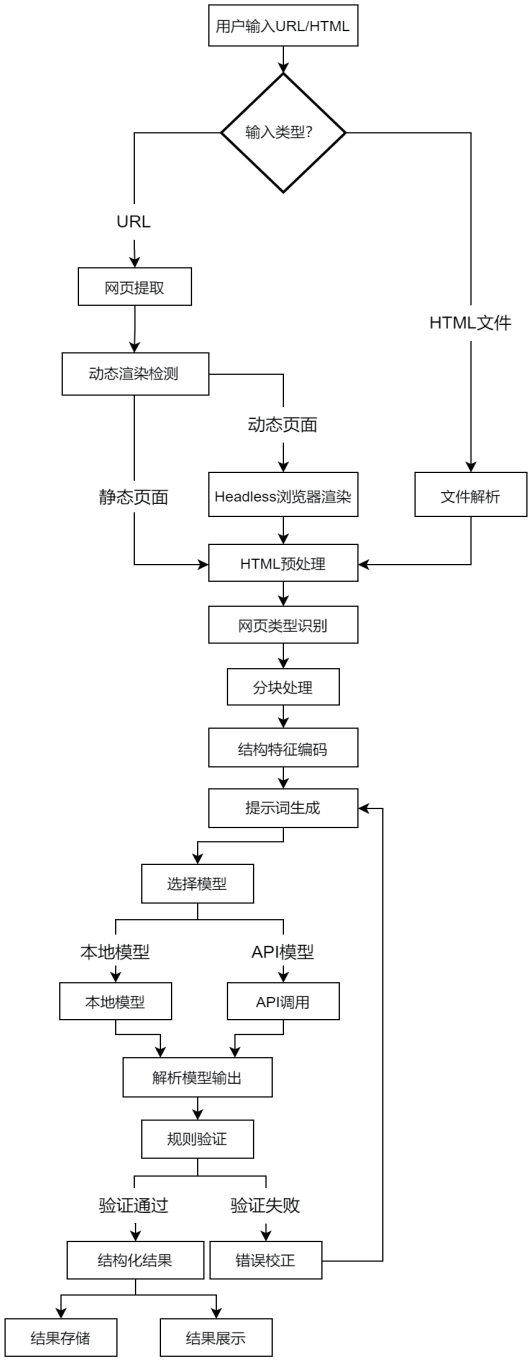


图 1 方法流程图

2.3 大模型适配与优化

针对不同网页类型设计专用提示模板，融合少样本示例与分层提示策略，优化模型对关键信息的聚焦能力。本模块的架构设计如图 3 所示。大模型适配过程中的关键技术和参数如表 1 所示。

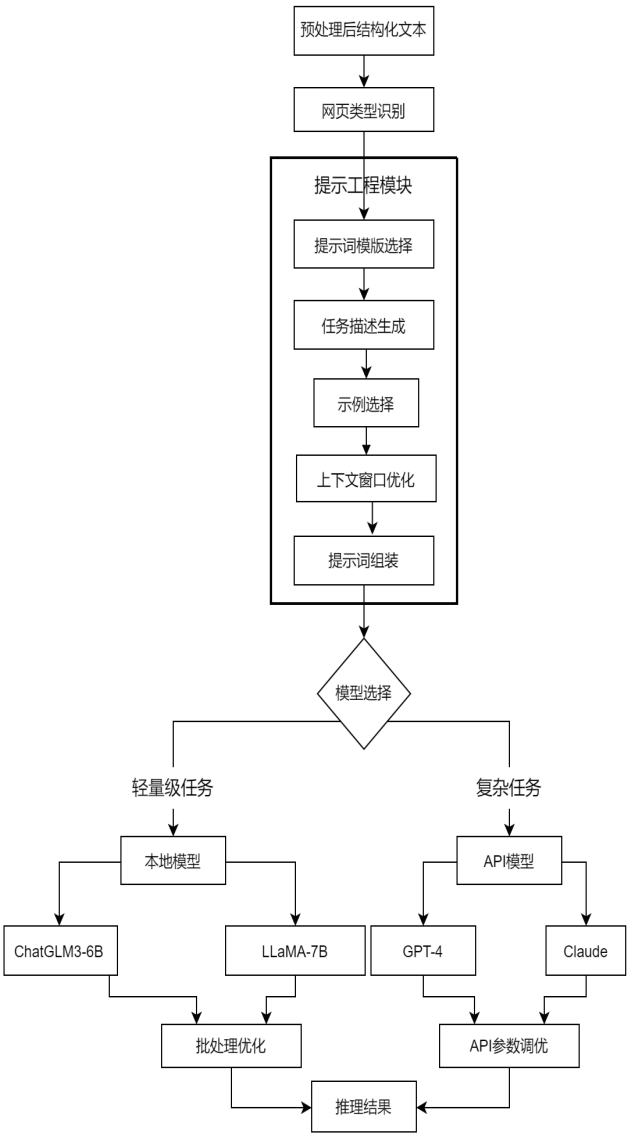


图 3 大模型适配与优化架构

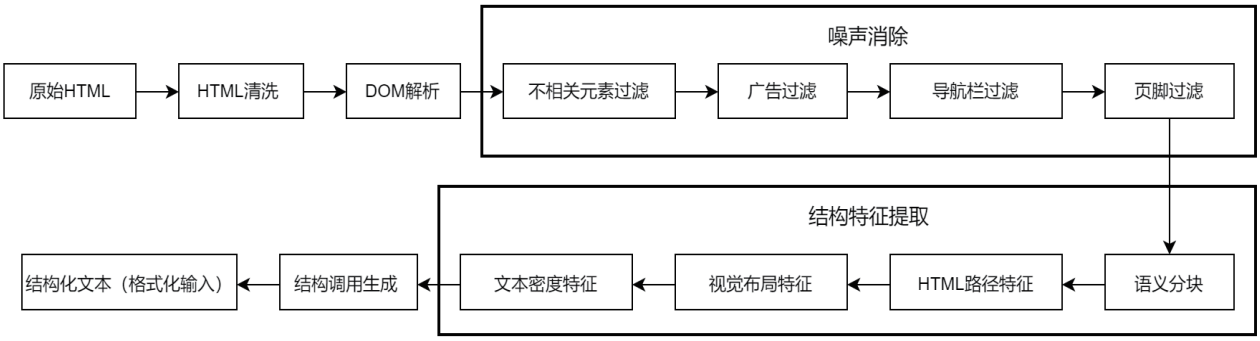


图 2 网页数据预处理流程图

表 1 大模型适配关键技术与参数配置

适配环节	技术方法	关键参数	优化目标	评估指标
网页类型识别	基于规则 + 轻量级分类器	特征集：DOM、URL，分类阈值：0.75	准确率 >90%	分类 F_1 值
提示词模板	不同网页类型的模板库	模板数量：7 种网页类型，模板长度：200~500 字符	覆盖主流网页类型	模板适配度
示例选择	相似度匹配 + 少样本学习	示例库大小：100 条，Top-K 选择：K=3	找到最相关示例	示例相关度
上下文窗口	滑动窗口 + 重要性排序	窗口大小：4 000 tokens，重叠率：30%	最大化信息利用	召回率
模型选择	任务复杂度自适应	复杂度阈值：0.65，延迟阈值：2 s	平衡效果与效率	延迟 - 准确率曲线
批处理优化	动态批次大小调整	最小批次：4，最大批次：16	提高吞吐量	每秒处理页面数
API 参数调优	温度、Top-P 等参数优化	温度：0.1~0.3，Top-P：0.85~0.95	提高输出稳定性	输出一致性

2.4 结果后处理与验证

通过规则校验、格式规范化与错误修复机制提升输出结果的准确性与一致性。针对发布时间、价格等特定字段设计验证规则，结合正则表达式与结构约束实现自动校正，具体流程如图 4 所示。

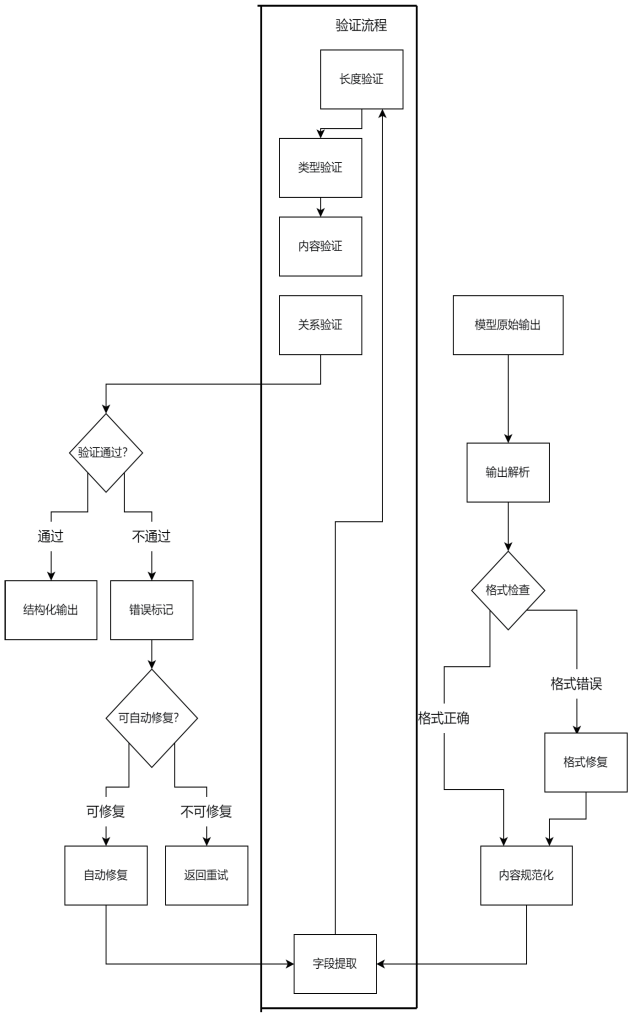


图 4 结果后处理与验证流程

后处理模块应用多层验证规则，针对不同类型网页的特定字段设置验证标准，具体如表 2 和图 5 所示。

表 2 内容验证规则详情

网页类型	字段	验证规则	阈值 / 格式	自动修复策略
新闻网页	标题	镂空、长度适中	5~100 字符	从 meta 标签提取
	正文	非空、最小长度、关键词密度	>200 字符、含 3+ 关键词	重选内容块
	发布时间	符合日期格式、时间范围合理	ISO 格式、近 5 年	正则修复、从 URL 提取
	作者	格式合理、长度适中	2~30 字符	可选字段，无需修复
电商网页	商品名称	镂空、长度适中	5~200 字符	从 title 标签提取
	价格	格式匹配、数值合理	货币符号 + 数字	正则提取、数值标准化
	参数描述	结构化、键值对匹配	>5 对参数	表格重构
论坛网页	帖子标题	镂空、长度适中	5~100 字符	从 breadcrumb 提取
	回复内容	嵌套结构正确	层级一致性	重构嵌套关系
	用户信息	格式一致	用户名 + 时间格式	模式匹配修复

3 实验与评估

3.1 实验环境与数据集

实验基于配置 AMD Ryzen 9 CPU、NVIDIA RTX 3090 GPU 的工作站，使用 PHP、PyTorch 等工具实现，具体如表 3 所示。为全面评估系统性能，构建了覆盖不同类型网站的综合数据集，数据集包含 7 类共 4 830 个网页，涵盖中英双语，均经过人工标注与交叉验证。详情如表 4 所示。

数据集采样策略以网站多样性和内容复杂度为主要考量因素，确保覆盖主流网页类型和不同结构特征。标注过程采用双人交叉审核机制，保证标注质量，最终标注一致性达到 94.8%。如图 5 所示。

表 3 实验环境配置详情

类别	组件	详细配置	说明
硬件环境	CPU	AMD Ryzen 95900X （12 核 24 线程）	负责前端处理与测量模型处理
	内存	64 GB DDR4 3 600 MHz	支持大规模网页批处理
	GPU	NVIDIA RTX 3090 (24 GB VRAM)	用于本地大模型推理
	存储	2 TB NVMe SSD	高速数据读写
软件环境	操作系统	Windows 10 Pro （64 位）	主要开发环境
	开发工具	VSCode 1.76.0	主要 IDE
	编程语言	PHP	主要开发语言
	深度学习框架	PyTorch 2.0.1，Transformers 4.30.2	大模型部署框架
	网页处理库	BeautifulSoup 4.11.2，lxml 4.9.2	HTML 解析工具
	API 框架	FastAPI 0.95.1	后端服务框架
	前端框架	React 18.2.0，Tailwind CSS 3.3.2	用户界面实现
网络环境	本地测试	千兆以太网	本地开发环境
	云端部署	AWS EC2 t3.2xlarge	生产环境模拟
	API 访问	5 Mbit/s 带宽（限制）	模拟实际 API 调用环境

表 4 实验数据集组成

数据集名称	网页类型	样本数量	网站数	语言分布	标注内容	动态页面比例 /%
NewsCorp	新闻网页	1 200	35	中文 60%，英文 40%	标题、正文、时间、作者	25
E-commerce	电商网页	850	20	中文 80%，英文 20%	商品名、价格、描述、参数	60
BlogSphere	博客网页	680	28	中文 50%，英文 50%	标题、正文、时间、分类	35
ForumData	论坛帖子	750	15	中文 70%，英文 30%	标题、内容、回复、用户信息	40
SocialMedia	社交媒体	520	12	中文 40%，英文 60%	用户、内容、互动、时间	75
AcademicPub	学术网页	350	8	中文 20%，英文 80%	标题、摘要、作者、引用	10
TechDocs	技术文档	480	10	中文 30%，英文 70%	标题、正文、代码块、链接	15
TotalDataset	综合数据	4 830	128	中文 54%，英文 46%	全部字段	38

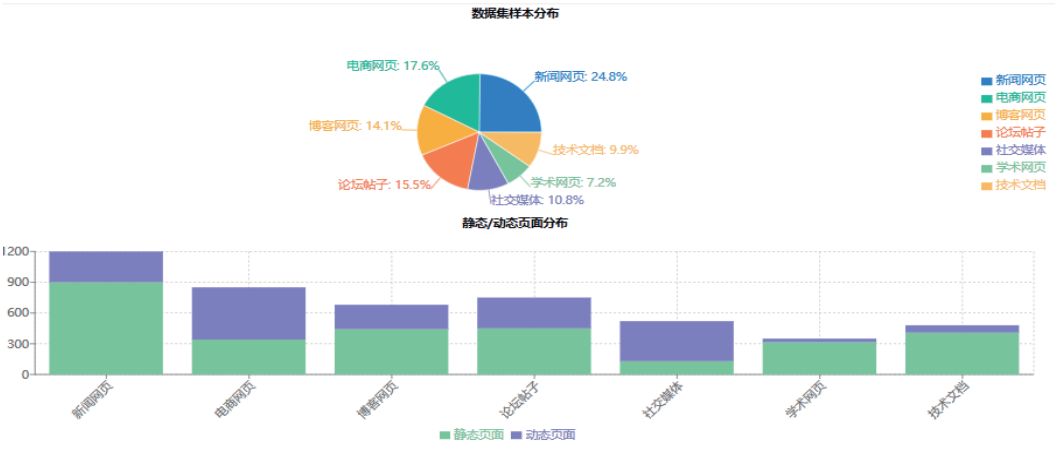


图 5 实验数据集分布

3.2 评估指标

为全面评估本文所提方法的性能，采用表 5 所示评估指标。

3.3 预处理效果

预处理模块的核心步骤包括 HTML 清洗、噪声过滤和结构编码三部分。各阶段的处理技术如表 6 所示。

在实验分析中，对不同类型网页的预处理效果进行了量化评估，具体结果如图 6 所示。该方法在不同类型网页上均取得较高的噪声清除率与结构保留率，验证了其普适性。新闻网页与博客网页，因其结构相对规则，算法能兼顾高精度与高效率。社交媒体和论坛网页，由于结构复杂、动态内容多，虽然效果依旧良好，但处理时间显著上升，需要进一步优化算法效率。

表 5 评估指标体系

指标类别	具体指标	计算公式	说明
准确性指标	精确率（Precision）	TP/(TP+FP)	提取内容的准确度
	召回率（Recall）	TP/(TP+FN)	提取内容的完整度
	F_1 值	2×(*P*×*R*)/(*P*+*R*)	精确率和召回率的调和平均
	ROUGE-L	LCS（提取，参考）/ 参考长度	最长公共子序列评估
效率指标	处理时间	s/ 页面	单个网页的平均处理时间
	内存消耗	MB	处理过程中的内存峰值
	API 调用成本	请求数 / 页面	API 调用效率
泛化性指标	跨网站准确率	见准确性指标	在未见过网站上的表现
	零样本性能	见准确性指标	无领域示例时的表现
	适应能力	随时间准确率变化率	网页结构变化下的适应性
用户体验指标	任务完成时间	秒	用户完成提取任务所需时间
	满意度评分	1~10 分	用户主观评价
	错误率	错误操作次数 / 总操作数	用户操作错误频率

表 6 预处理阶段关键技术与效果分析

处理阶段	关键技术	处理前数据	处理后数据	效果提升	计算复杂度
HTML 清洗	正则表达式替换 HTML 解析器	原始 HTML	规范化 HTML	减少 40% 冗余代码 修复 95% 错误标签	$O(n)$
噪声过滤	基于 CSS 选择器的过滤 广告模式识别	完整 DOM 树	核心内容 DOM 子树	去除 75% 非核心内容 提高内容密度 3 倍	$O(n\log n)$
语义分块	文本密度计算 标题检测	连续 DOM 结构	语义块列表	准确识别 90% 主要内容区块 改善段落边界 85%	$O(n)$
结构编码	DOM 路径提取 视觉特征编码 文本特征标注	语义块	编码后结构文本	保留 95% 结构信息 适配性提升 40%	$O(n^2)$

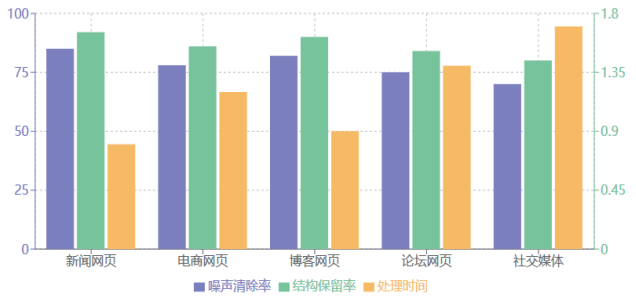


图 6 不同类型网页预处理效果对比

3.4 不同网页类型的提取效果

本文对比了内容提取方法在不同类型网页上的提取效果，如表 7 所示。

在新闻、电商、博客等网页上均取得较高提取精度，其中标题提取 F_1 值达 96.1%，正文提取为 92.1%。结构化程度高的网页（如新闻、学术页面）表现优于社交媒体与论坛类网页。

从图 7 结果可以看出，结构化程度高的网页（新闻、学术、技术文档）提取效果普遍高于结构复杂的网页（社交媒体、论坛）。标题提取效果最佳（平均 F_1 值 96.1%），其他元数据提取最具挑战性（平均 F_1 值 86.8%）。动态页面与静态页面相比，F_1 值平均下降 5.8 个百分点，社交媒体网页差异最明显。

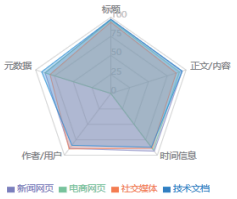
表 7 不同网页类型提取效果评估（准确率 / 召回率 /F_1 值，%）

网页类型	标题	正文	时间信息	作者 / 用户	其他元数据	整体 F_1 值
新闻网页	96.8 98.2 97.5	93.5 96.4 94.9	94.2 92.8 93.5	90.5 87.6 89.0	88.3 86.5 87.4	94.2
电商网页	95.3 97.0 96.1	90.2 92.5 91.3	92.8 90.6 91.7	不适用	89.5 86.2 87.8	91.8

表 7（续）

网页类型	标题	正文	时间信息	作者 / 用户	其他元数据	整体 F_1 值
博客网页	95.8	91.8	93.5	92.1	87.6	92.7
	97.5	94.3	91.2	88.7	84.5	
	96.6	93.0	92.3	90.4	86.0	
论坛帖子	94.2	88.7	89.4	90.8	84.2	89.5
	96.3	91.6	87.2	86.5	81.8	
	95.2	90.1	88.3	88.6	83.0	
社交媒体	92.5	85.6	90.2	91.5	82.3	87.2
	94.8	88.2	87.5	88.3	79.5	
	93.6	86.9	88.8	89.9	80.9	
学术网页	95.6	92.5	93.8	94.2	91.7	93.8
	97.8	94.8	91.5	92.6	89.2	
	96.7	93.6	92.6	93.4	90.4	
技术文档	96.2	94.3	88.5	85.6	93.2	93.2
	98.5	95.7	86.2	82.4	90.8	
	97.3	95.0	87.3	84.0	92.0	
平均	95.2	90.9	91.8	90.8	88.1	91.8
	97.2	93.4	89.6	87.7	85.5	
	96.1	92.1	90.6	89.2	86.8	

（a）网页类型整体提取效果（F_1 值 /%）

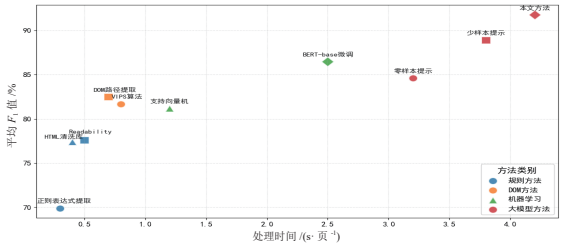


（b）不同内容类型的提取效果（F_1 值 /%）

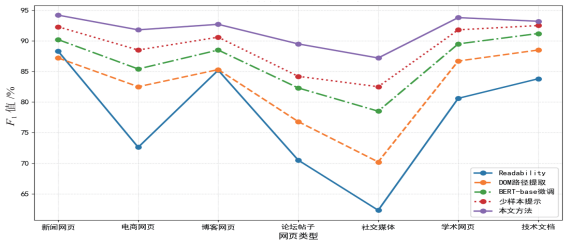
图 7 不同网页类型提取效果对比

3.5 与传统方法的对比

为评估大模型方法的优势，将其与主流传统方法进行了对比实验，结果如表 8 和图 8 所示，部分数据来自文献 [10]。



(a) 方法性能对比 (准确率 vs 处理时间)



(b) 主要方法在不同网页上的表现

图 8 不同提取方法性能对比

本文方法在平均 F_1 值上达到 91.8%，较最佳传统方法（BERT 微调）提升 5.3%，且在跨网站场景下泛化能力显著优于规则与 DOM-based 方法。研究结果表明，该方法在各种类型的网页页面上都获得了较好的结果，其 F_1 均值为 91.8%，较传统的最优算法（BERT 修正）提高了 5.3%。大型模型的效能比传统的模式有显著的准确性，尽管运算的耗时更多。对于不需要实时处理的场合，大规模建模方式有着显著的优点，如离线分析、批量爬取等场景仍具实用性。与传统算法相比，本文方法在社会媒体及网络环境下的应用表现出了更加明显的优越性，表现出了对非结构化信息的更好理解。在交叉站点的实验中，大型模型算法的性能降低最少（2.8%），而基于规则与 DOM 的算法降低了 15%。

表 8 大模型与传统方法对比（F_1 值，%）

方法	新闻网页	电商网页	博客网页	论坛帖子	社交媒体	学术网页	技术文档	平均 F_1 值	处理时间（s/ 页）
基于规则的方法									
正则表达式提取	76.2	68.5	72.4	63.8	55.6	78.5	74.3	69.9	0.3
Readability	88.3	72.6	85.2	70.5	62.3	80.6	83.8	77.6	0.5
HTML 清洗库	82.5	75.8	80.6	73.2	65.7	82.3	81.5	77.4	0.4
基于 DOM 的方法									
VIPS 算法	85.6	80.3	83.5	78.6	72.4	84.9	86.3	81.7	0.8
DOM 路径提取	87.2	82.5	85.3	76.8	70.2	86.7	88.5	82.5	0.7
基于机器学习的方法									
支持向量机	84.5	81.2	82.6	77.5	73.8	83.6	85.2	81.2	1.2
BERT-base 微调	90.2	85.4	88.5	82.3	78.5	89.5	91.2	86.5	2.5
基于大模型的方法									
零样本提示	88.6	83.2	86.7	80.5	76.3	87.3	89.6	84.6	3.2
少样本提示	92.3	88.5	90.6	84.2	82.5	91.8	92.5	88.9	3.8
本文方法	94.2	91.8	92.7	89.5	87.2	93.8	93.2	91.8	4.2

3.6 提示策略与模型选择分析

为了优化方法性能，对不同大模型和提示策略进行了对比实验，结果如表 9 所示。

表 9 不同模型与提示策略对比实验（F_1 值，%）

模型	普通提示	结构化提示	示例增强	层次化提示	领域适配	平均提升
开源模型						
BERT-base	86.5	—	—	—	88.2	—
ChatGLM3-6B	84.6	87.3	89.5	90.2	91.8	+7.2
LLaMA-7B	83.2	86.5	88.7	89.3	90.6	+7.4
API 模型						
GPT-3.5	87.5	90.2	92.3	93.5	94.8	+7.3
GPT-4	89.2	92.5	93.8	95.2	96.5	+7.3
Claude	88.6	91.8	93.2	94.6	95.9	+7.3

注：普通提示：基本的工作说明，例如“提取网页页面的题目和内容”；结构化提示：增加结构化的输出形式需求，例如“以 JSON 形式回传”；示例增强：增加 1~3 个带有注释的例子；层次化提示：把一项工作分成多个阶段，例如“首先确定重点领域，然后提炼特定领域”；领域适配：专用的特殊页面样式模板。

实验表明，分层提示与领域自适应策略对性能提升贡献最大。GPT-4 等 API 模型效果最优，而 ChatGLM3-6B 等开源模型在资源受限环境下仍具实用价值。如图 9 所示，为深入分析提示策略的影响，对 GPT-4 模型进行了详细的参数敏感性分析，结果如表 10 所示。

图 9 不同提示策略对模型性能的影响

从以上结果可以看出，总体来说，API 模型的总体表现要好于开放源码模型，但是出于开销的考量，ChatGLM3-6B 更适合于资源受限的场合。从图 9 可以看出，各模型都显示出对线索战略改善的显著反应，其平均提高 7.3 个百分点，这表明线索设计对于绩效是非常关键的。

在诸如社会化网络的复杂页面上，领域适应能力的增强幅度最大，达到了 10.4%，说明有针对性地改进可以很好地处理困难的问题。温度与样本数目是主要的影响因素，只要稍微降温，并且给出 2~3 个高品质的例子，就可以明显提高结果。

3.7 讨论与局限性

基于前述实验，对方法的综合性能进行评估，结果如表 11 所示。根据实验结果，认为大语言模型在网络信息提取方面具有独特的优越性，特别是对于结构复杂的网页页面。F_1 的平均值较常规处理提高了 5.3%~14.5%。提示的设置对于整个方法的整体表现有很大的作用，通过调整可以使 F_1 提高 7.2%~10.4%。分层线索与域适应策略是最好的。基于 API 模型（以 GPT-4 为例）的算法性能较好，但基于开销和资源约束，基于局部开放源码模型（例如 ChatGLM3-6B）进行均衡的取舍。本方法在多站点、多数据环境下，F_1 值降低不超过 3%，具有较强的泛化能力。大型模型下的逻辑运算速度仍然是一个关键的问题，并且在每页面 4.2 s 的情况下，还存在着进一步的改进余地。方法在以下场景仍存在局限性。如表 12 所示。

当前方法在处理高度动态加载内容、多主题混合页面及表格数据时仍存在一定局限。未来工作将聚焦多模态特征融合、轻量化推理与动态自适应机制。

4 总结

本文以大语言模型为基础，通过对网页预处理、大数据模型自适应、结果验证以及人机交互等方面的研究，构建面向不同类型网页的网页内容提取方法。试验证明，对于各种页面，本文提出的算法获得了 91.8% 的 F_1，比常规算法提高了 5.3%~14.5%，尤其在处理复杂或多站点的情况下，表现出了明显的优越性。研究表明，特征分层设计是决定整个方法性能的关键因素，通过分层线索与领域适应，可以将预测精度提高 7.2%~10.4%。本研究将为网页内容提取技术的发展和完善奠定理论基础。

当然，当前方法主要依赖文本特征，未来应结合视觉和 DOM 结构特征，构建多模态融合模型，提升复杂网页理解能力。探索模型压缩、知识蒸馏等技术，优化模型结构，提高运算速度，降低资源消耗，使其更适合实际应用。此外，本文还将研究动态自适应技术，使模型能够更好地适应网页

表 10 GPT-4 参数敏感性分析

参数	取值范围	最佳值	性能影响	稳定性影响
温度	0.0~1.0	0.2	低温提高准确性（+3.5%）	低温提高一致性（+8.2%）
Top-P	0.1~1.0	0.92	较大值略微提高召回率（+1.2%）	中等值提高稳定性（+4.5%）
最大令牌数	256~2 048	1 024	适中值平衡效果与效率	较大值提高完整性（+5.8%）
示例数量	0~5	2~3	2~3 个示例效果最佳（+6.5%）	过多示例降低性能（-2.3%）
提示词长度	50~500 词	200~250 词	中等长度最佳（+4.8%）	过长提示降低关注度（-3.6%）

表 11 方法综合性能评估

评估维度	性能指标	指标值	与目标差距	主要影响因素
准确性	平均 F_1 值	91.8%	+6.8%	大模型理解能力、提示优化
	标题提取 F_1 值	96.1%	+11.1%	网页结构特征、模型上下文理解
	正文提取 F_1 值	92.1%	+7.1%	文本密度特征、内容边界识别
	时间信息 F_1 值	90.6%	+5.6%	格式多样性、规则验证
效率	平均处理时间	4.2 s/ 页	-0.8 s	大模型推理速度、API 延迟
	内存峰值	6.8 GB	可接受	模型大小、批处理优化
	API 成本	约 0.05 元 / 页	可优化	提示长度、模型选择
泛化性	跨网站准确率	88.5%	+13.5%	提示策略、示例选择
	零样本性能	84.6%	+9.6%	大模型基础能力、指令理解
	适应能力评分	8.5/10	+1.5	动态网页处理、错误恢复
用户体验	任务完成时间	45 s	-15 s	UI 设计、操作流程
	满意度评分	8.7/10	+1.7	结果展示方式、响应速度
	错误率	5%	-10%	引导设计、错误提示

表 12 失败案例分析

失败类型	案例描述	失败原因	改进方向
结构识别错误	微博热搜嵌套内容错误识别	高度嵌套 DOM 导致语义混淆	增强结构特征编码，优化层次提示词
内容边界模糊	论坛回复与引用内容混淆	视觉上分离，但语义上关联的内容	结合视觉特征与语义理解
多主题干扰	新闻多篇报道混合提取	缺乏明确主次内容区分能力	增加主题识别，分段处理
动态内容缺失	社交媒体评论漏提取	渲染时机识别不准确	改进动态内容检测机制
特殊格式处理	表格数据结构化提取不完整	对表格语义理解有限	专门设计表格提取提示策略

结构的变化，并提高模型的可解释性，为用户提供更直观的提取依据和结果解释。

参考文献：

[1]AGUN H V. WebCollectives: a light regular expression based web content extractor in Java[EB/OL]. [2026-04-07].https://ui.adsabs.harvard.edu/abs/2023SoftX.2401569A/abstract.

[2]JEYARAJ A. Teaching tip scaffolding in business analytics education: using python for web scraping[J]. Journal of information systems education, 2024, 35(4): 438-450.

[3]WHITEHOUSE C X, VANIA C, AJI A F, et al. WebIE: faithful and robust information extraction on the web[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics.Brussels: ACL, 2023:7734-7755.

[4]ZHANG J, QIANG J P, ZHOU C Q. New horizons in web search, web data mining, and web-based applications[J]. Applied sciences, 2024, 14(2): 530.

[5]BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems. NewYork: ACM, 2020:1877-1901.

[6]TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: open and efficient foundation language models[EB/OL]. (2023-02-27)[2026-04-07].https://doi.org/10.48550/arXiv.2302.13971.

[7]SWAMI S A.Web data extraction and alignment tools: a survey[EB/OL].[2026-04-07].https://www.semanticscholar.org/paper/Web-Data-Extraction-and-Alignment-Tools%3A-A-Survey-Swami/6913a6e190d1b2102e2f01270889cf9bfcf7b524.

[8]AGIRRE E. Few-shot information extraction is here[C]//SIGIR'22: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. NewYork: ACM, 2022:2.

[9]ZHOU Z H. A brief introduction to weakly supervised learning[J]. National science review, 2018, 5(1): 44-53.

[10] 魏建兵 . 基于 DOM 树和混合文本密度的网页信息提取方法研究 [J]. 信息与电脑 (理论版),2023,35(10):52-54.

【作者简介】

吕孝忠（1988—），男，安徽霍邱人，博士，助理研究员，研究方向：大数据与大模型应用。

张雨荷（2003—），女，安徽和县人，本科，研究方向：数据科学与大数据技术研究。

吴其林（1975—），男，安徽肥东人，博士，教授，研究方向：计算机与人工智能。

（收稿日期：2025-09-25 修回日期：2026-04-10）