**Report of Deep Learning for Natural Langauge Processing**

Zengchang Qin

[zengchang.qin@gmail.com](mailto:zengchang.qin@gmail.com)

# 摘要

本实验旨在利用微软的 Phi-3-mini-128k-instruct 模型生成静态 HTML 着陆页，探索大型语言模型（LLMs）在网页设计中的应用潜力。通过一个 Python 脚本，用户可以输入网页的主题和基调（如“AI 编程助手”与“清新极简”），模型自动生成包含 hero banner、features（三个）、call-to-action 和 footer 等部分的 HTML 代码。本报告详细描述了实验的设计、执行过程、生成结果的分析，以及讨论模型的性能和局限性。研究结果表明，Phi-3-mini-128k-instruct 模型能够根据自然语言提示生成功能性网页，但存在一些小问题，如图像链接缺失或样式不一致，需手动调整。未来工作可包括模型微调、集成更多工具以提升生成质量，以及扩展到更复杂的动态网页设计任务。

# 简介

微软的 Phi-3 系列模型是当前 LLMs 领域的重要成果之一，专为指令跟随和安全性设计。Phi-3-mini-128k-instruct 模型具有 38 亿参数，支持 128K 令牌的上下文长度，适用于多种语言任务，包括代码生成。其训练数据包括高质量的合成数据和过滤后的网络数据，专注于推理密集型任务的表现。本实验选择该模型，旨在验证其在生成静态 HTML 着陆页中的效能，探讨 AI 在网页设计中的潜力。

# 理论方法

There are models of my research.

## M1: Phi-3-mini-128k-instruct

本实验的核心是利用 Phi-3-mini-128k-instruct 模型生成 HTML 代码。模型基于 Transformer 解码器架构，拥有 32 层，词汇表大小为 32,064 令牌，训练数据规模达 3.3 万亿令牌。模型经过监督微调（SFT）和直接偏好优化（DPO），确保其遵循指令并具备安全性。其长上下文能力（128K 令牌）特别适合处理复杂的提示生成任务。

TailwindCSS 作为实用优先的 CSS 框架，通过内联类名直接在 HTML 中定义样式，无需编写自定义 CSS。这与模型生成代码的特性高度契合，能够快速构建响应式设计。脚本中使用的提示工程设计，确保模型输出符合 HTML5 标准，并包含指定的部分（如 hero banner、features 等）。

# 实验过程

实验使用 Python 脚本“auto\_webpage.py”，其主要功能包括模型加载、提示构建、HTML 生成和文件保存。以下是详细步骤：

1. **环境配置**：
2. 确保安装了必要的库，包括 torch、transformers 和 webbrowser。脚本检测硬件环境，若有 CUDA 可用则使用 GPU（例如 NVIDIA 3090），否则使用 CPU。
3. **模型加载**：通过 AutoTokenizer.from\_pretrained 和 AutoModelForCausalLM.from\_pretrained 加载 Phi-3-mini-128k-instruct 模型，设置 device\_map="auto" 以自动分配设备。
4. **用户输入**：脚本提示用户输入网页主题（如“AI 编程助手”）和基调（如“清新极简”）。
5. **提示构建**：使用 build\_prompt 函数生成提示，系统指令要求模型作为高级网页设计师，生成完整的 HTML5 着陆页，包含内联 TailwindCSS CDN 和最少 JS。用户指令指定主题、基调和部分（如 hero banner、features（3）、call-to-action、footer）。
6. **HTML 生成**：通过 generate\_page 函数调用模型的 generate 方法，设置 max\_new\_tokens=7000、temperature=0.7、top\_p=0.9 和 do\_sample=True，使用 TextStreamer 实时打印生成进度。
7. **保存与打开**：将生成的 HTML 代码保存到“index.html”，并使用 webbrowser.open 在默认浏览器中打开。

实验运行了多次，输入包括但不限于以下组合：

* 主题：“AI 编程助手”，基调：“清新极简”
* 主题：“在线教育平台”，基调：“活力四射”
* 主题：“电子商务网站”，基调：“专业可靠”  
  每次运行记录生成时间和输出结果，分析模型的生成一致性和质量。

**结果分析**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>AI Programming Assistant</title>

<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<!-- Hero Banner -->

<section class="bg-gray-100 py-20">

<div class="container mx-auto text-center">

<h1 class="text-5xl font-bold">Revolutionize Your Coding with AI</h1>

<p class="mt-4 text-gray-600">Your personal AI assistant for coding tasks</p>

<a href="#features" class="mt-8 inline-block bg-blue-500 text-white py-2 px-4 rounded">Learn More</a>

</div>

</section>

<!-- Features -->

<section id="features" class="py-20">

<div class="container mx-auto">

<h2 class="text-3xl font-bold text-center">Features</h2>

<div class="mt-10 grid grid-cols-1 md:grid-cols-3 gap-8">

<div class="text-center">

<img src="icon1.png" alt="Code Completion" class="mx-auto">

<h3 class="mt-4 text-xl font-semibold">Code Completion</h3>

<p class="mt-2 text-gray-600">Get instant code suggestions as you type.</p>

</div>

<div class="text-center">

<img src="icon2.png" alt="Bug Detection" class="mx-auto">

<h3 class="mt-4 text-xl font-semibold">Bug Detection</h3>

<p class="mt-2 text-gray-600">Identify and fix bugs quickly with AI-powered tools.</p>

</div>

<div class="text-center">

<img src="icon3.png" alt="Documentation Generation" class="mx-auto">

<h3 class="mt-4 text-xl font-semibold">Documentation Generation</h3>

<p class="mt-2 text-gray-600">Automatically generate documentation for your code.</p>

</div>

</div>

</div>

</section>

<!-- Call to Action -->

<section class="bg-blue-500 py-20 text-white text-center">

<h2 class="text-3xl font-bold">Ready to Get Started?</h2>

<p class="mt-4">Sign up now and take your coding to the next level.</p>

<a href="#signup" class="mt-8 inline-block bg-white text-blue-500 py-2 px-4 rounded">Sign Up</a>

</section>

<!-- Footer -->

<footer class="bg-gray-800 py-4 text-white text-center">

<p>&copy; 2025 AI Programming Assistant. All rights reserved.</p>

</footer>

</body>

</html>

上述代码使用 TailwindCSS 的 utility classes（如 bg-gray-100、text-5xl）定义样式，确保页面响应式和美观。实验中，所有生成的页面均包含 hero banner、三个 features 卡片、call-to-action 部分和 footer，符合提示要求。

生成时间因硬件环境而异，在配备 NVIDIA 3090 GPU 的机器上，通常在几秒内完成；使用 CPU 时可能稍长，但仍保持合理效率。

# 结论

本实验成功验证了 Phi-3-mini-128k-instruct 模型生成静态 HTML 着陆页的可行性。通过简单的自然语言提示，用户可以快速获得基础网页模板，显著提高了开发效率。生成的页面结构清晰，样式通过 TailwindCSS 实现，易于维护和修改。然而，存在一些小问题，如图像链接缺失或样式不一致，需手动调整。未来工作可包括对模型进行微调，扩展训练数据以覆盖更多网页设计趋势，或集成更多工具以支持动态内容生成。

# References

1. Zenchang Qin and Lao Wang (2023)，How to learn deep learning? Journal of Paper Writing, Vol. 3: 23: pp. 1-12.
2. [LDA主题模型及Python实现\_python lda-CSDN博客](https://blog.csdn.net/qq_36128101/article/details/137471372?ops_request_misc=%257B%2522request%255Fid%2522%253A%25220b780646348b0033e5cdfb5f41839575%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=0b780646348b0033e5cdfb5f41839575&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-137471372-null-null.142%5ev102%5epc_search_result_base4&utm_term=LDA&spm=1018.2226.3001.4187)