

Report of Deep Learning for Natural Language Processing

Shixiang Li BY2203043

lishixiang@buaa.edu.cn

Abstract

With the rapid advancement of generative artificial intelligence, general-purpose large language models (such as ChatGPT[1], Gemini[2], Deepseek[3], and Qwen[4]) have been widely applied in areas such as content creation, code generation, and multimodal interaction. This paper explores the feasibility and effectiveness of using general-purpose large models to assist in web design. A travel-themed web page (taking Milan as an example) was selected as the experimental subject, and the Gemini large model was employed for design and generation. The report covers the model usage methodology, webpage structure planning, content generation, style design, and debugging process. Experimental results indicate that large models offer significant efficiency advantages in the early stages of web design, particularly in structural layout and initial draft generation. However, challenges remain in fine-grained style control and consistency of generated content. This study validates the potential of general-purpose large models in web design and proposes directions for future improvement.

Introduction

Web design is a comprehensive task that integrates content planning, aesthetic design, and front-end development. Traditionally, it requires close collaboration between designers and developers, encompassing a lengthy process from requirement analysis and interface conception to code implementation and content population. In recent years, General-purpose Large Language Models (GLLMs), with their powerful capabilities in language understanding and code generation, have emerged as new assistive tools in the web design workflow.

This report aims to explore the complete process of using general-purpose large models to assist in web design, with a particular focus on their performance in structural design, content generation, and code output. To contextualize the study, a travel-themed web page was selected as the experimental case, using the city of Milan as the content reference. It should be noted that

Milan serves merely as illustrative material for the experiment; the focus of this study lies in the methods and effectiveness of using large models to assist in web design, rather than on the tourism information of any specific city.

Methodology

M1: Tool Selection

- Large Language Model Used: Gemini (developed by Google)
- Interaction Method: Prompt-based natural language input
- Web Development Technologies: HTML + CSS + limited JavaScript
- Editing Environment: VS Code (for code editing and local debugging)

M2: Design Workflow

The overall web design process is divided into the following stages:

1. Requirement Description: Using natural language to describe to Gemini the desired type of webpage (travel-themed), page structure (including navigation bar, introduction, attraction display, recommended activities, contact information, etc.).
2. Content Generation: Prompting the model to generate textual content and image suggestions for each module of the webpage, with semantic completion or refinement applied to specific sections.
3. Structural Code Generation: Instructing the model to output the HTML code for the page structure, including the hierarchical organization of elements and the use of semantic tags.
4. Style and Layout Design: Prompting the model to generate CSS styles, covering aspects such as color schemes, layout design, and responsive design.
5. Debugging and Modification: Running the model-generated code locally to identify functional or stylistic issues, then requesting modification suggestions from the model through additional prompts.
6. Optimization and Summary: Making fine adjustments to the overall page and proposing directions for optimization based on user experience and visual style.

Experimental Studies

E1: Initial Design Generation

A prompt was given to Gemini: "Help me design a webpage that provides information for tourists visiting a city, including weather forecasts, transportation information, dining and accommodation, and attractions. The page should be aesthetically pleasing and feature dynamically updated information." The model generated a complete HTML page structure, along with basic CSS styling and JavaScript-based interactive functionalities.

E2: Content Customization and Module Refinement

Through a series of prompts, the city name was replaced with "Milan," and additional content modules such as "Local Cuisine" and "Festivals and Culture" were requested. The model responded promptly and expanded the content with a high degree of semantic coherence and logical structure.

E3: Style Enhancement Attempts

Gemini was prompted to add features such as gradient backgrounds, card-style layouts for attractions, and responsive design. The model-generated CSS generally met the requirements, although some details (e.g., font selection and spacing control) still required manual adjustment.

E4: Addition of Interactive Elements

Further prompts requested simple JavaScript-based interactive features such as scroll animations and a "back to top" button. The model generated basic functional scripts, but compatibility issues in event binding required adjustments based on the actual browser environment.

E5: Iterative Optimization

Based on debugging results, feedback was provided to Gemini regarding style inconsistencies and image loading failures. The model was able to revise the code accordingly, demonstrating a certain degree of contextual awareness.



Figure 1 Webpage Display Example 1

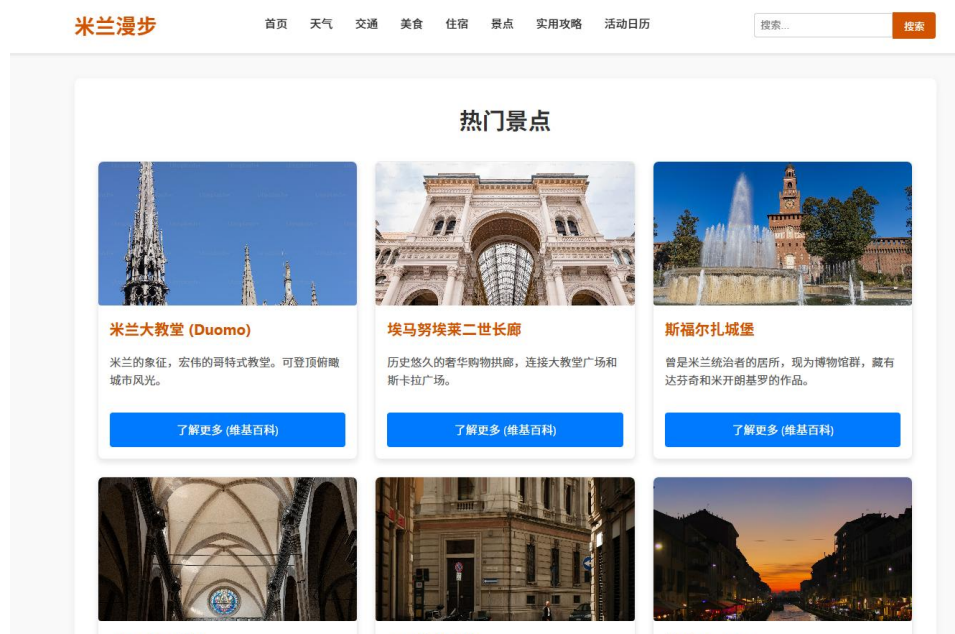


Figure 2 Webpage Display Example 2

Conclusion

This experiment validates the effectiveness of using general-purpose large models for web design. Gemini demonstrated strong capabilities in structure generation, content filling, and style

assistance, making it particularly suitable for quickly building web prototypes and implementing basic functionality. However, the following limitations were observed:

Limited Style Precision: For complex or detailed design requirements, the generated styles often require manual adjustments.

Simplified Interaction Logic: JS functionality support is limited, and complex interactions need to be supplemented by developers later.

Generation Consistency Issues: In multi-round interactions, the model sometimes "forgets" previous structures or styles, affecting overall consistency.

Overall, general-purpose large models offer an efficient and low-barrier approach to web design, especially suitable for prototyping, educational experiments, and rapid iteration scenarios. Future work could explore combining professional web frameworks and multimodal inputs (such as image sketches) to further improve web generation quality.

References

- [1] OpenAI. (2024). GPT-4 technical report. Retrieved from <https://arxiv.org/abs/2303.08774>
- [2] DeepMind. (2024). Gemini 1.5 series: Multimodal and efficient language models. Retrieved from <https://arxiv.org/abs/2403.05530>
- [3] DeepSeek AI. (2024). DeepSeek LLM: Expanding capabilities in large language models. Retrieved from <https://arxiv.org/abs/2401.02954>
- [4] Alibaba Cloud. (2024). Qwen2.5: Reinforcement learning for long-text generation. Retrieved from <https://arxiv.org/abs/2412.15115>