Применение принципов визуальных интерфейсов в больших языковых моделях как альтернатива текстовому взаимодействию

Е.А. Трофимова[1], Р.С. Савчук[2]

ВШЭ, Научно-учебная лаборатория методов анализа больших данных[1],
ВШЭ ФКН[2]

На данный момент, бóльшая часть взаимодействия между пользователем и моделью происходит при помощи текстового интерфейса. Данный вид взаимодействия с LLM является наиболее интуитивным как и с точки зрения архитектуры модели, так и с точки зрения использования. В то же время, неформатированный вывод в текстовом формате является неудобным для восприятия комплексной информации. Язык разметки *Markdown*, использующийся повсеместно в чат-ботах, позволил минимально форматировать текст. Несмотря на успешный опыт внедрения методов форматирования текста, взаимодействие с LLM не получило существенного развития. В *SearchGPT* [1] были представлены виджеты, встроенные в текст, но наполнение данных виджетов генерируется без участия LLM. В рамках исследования предлагается изучить возможные подходы к взаимодействию с LLM, выявить их ограничения и предложить более интуитивные методы визуального взаимодействия.

Такие языки разметки как *Markdown* являются простыми для генерации, так как не имеют вложенности и поддерживают лишь примитивные *теги*, такие как заголовки и списки. Генерация более сложных языков разметки долгое время являлась затруднительной из-за множества ошибок в выводе, которые не решались увеличением числа параметров модели. Данная проблема была решена [2] путем ограничения возможных токенов в выводе, причем метод Grammar-Constrained Decoding применим для любых *бесконтекстных грамматик*. Данное решение было усовершенствовано [3] и добавлено в исходный код LLM пакетов, таких как llama.cpp [4].

Предложенный подход: представление диалогов с пользователем в виде двунаправленного графа: вершины представляют собой логические блоки информации, а ребра – гиперссылки. Каждая вершина содержит набор виджетов наподобие HTML. Таким образом, данный граф возможно конвертировать в произвольный язык разметки. Для того, чтобы найти наилучший язык разметки для конкретной LLM, был разработан инструментарий для генерации и оценки бесконтекстных грамматик, в том числе SR(1) парсер-генератора.

Литература

1. OpenAI, Introducing ChatGPT Search.
   URL: https://openai.com/index/introducing-chatgpt-search/.

2. Geng S., Josifoski M., Peyrard M., West R. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023. Association for Computational Linguistics, 2023. P. 10932–10952.
   DOI: 10.18653/V1/2023.EMNLP-MAIN.674.

3. Shorten C., Pierse C., Smith T.B. et al. StructuredRAG: JSON Response Formatting with Large Language Models // CoRR. 2024. Vol. abs/2408.11061. arXiv: 2408.11061. URL: https://arxiv.org/abs/2408.11061.

4. llama.cpp: LLM inference in C/C++. URL: https://github.com/ggerganov/llama.cpp.