

## **COOLPROMPT: БИБЛИОТЕКА АВТОМАТИЧЕСКОЙ ОПТИМИЗАЦИИ ПРОМПТОВ**

**Кулин Н.И.<sup>1</sup>, Журавлев В. Н.<sup>2</sup>, Хайруллин А.Р.<sup>3</sup>**

**Научный руководитель – канд. техн. наук, доцент ИПКН, Муравьев С. Б.<sup>4</sup>**

Университет ИТМО

<sup>1</sup>242106@niuitmo.ru <sup>2</sup>334857@niuitmo.ru <sup>3</sup>arkhairullin@itmo.ru <sup>4</sup>smuravyov@itmo.ru

Работа выполнена в рамках темы НИР №625135 «Разработка программных модулей для эффективного обучения и применения моделей глубокого обучения».

### **Введение**

На сегодняшний день большие языковые модели (LLM) являются распространенными и эффективными моделями для решения задач обработки естественного языка (NLP) от генерации текста до решения сложных логических задач. Эффективность использования LLM сильно зависит от составления текстовой входной инструкции – промпта. Оптимизация промпта – это практика разработки входных инструкций для определения желаемого поведения LLM. На текущий момент задача оптимизации промптов является ручной и экспертной, и она напрямую зависит от знаний доменной области, понимания и опыта работы с языковыми моделями. Одно из наиболее значительных направлений исследования — это разработка методов автоматической оптимизации промптов (автопроммптинга), которая включает в себя различные алгоритмы и стратегии оптимизации для автоматизации создания такие, как подходы на основе фундаментальных моделей (FM) и планирования [1], обучение с подкреплением (RL) [2], эволюционные алгоритмы [3] и на основе градиентной оптимизации [4]. Несмотря на достижения, текущие методы обладают рядом недостатков. Во-первых, методы автопроммптинга на основе градиентной оптимизации и на основе RL требовательны к дообучению самой LLM для каждой новой задачи, а также уступают по эффективности качества на ряде датасетов методам на основе эволюционных алгоритмов и на основе FM, что делает их перспективными направлениями для развития исследования. Во-вторых, методы на основе эволюционных алгоритмов и на основе FM имеют риск застревания в локальном оптимуме – из-за ограниченности по количеству операторов изменений промптов и из-за появления “галлюцинаций” LLM, вследствие чего могут потерять контекст первоначальной постановки задачи. В-третьих, перенос одних промптов на другие задачи, данные и большие языковые модели не гарантирует аналогичных высоких результатов, снижая масштабируемость и воспроизводимость и требуя дополнительное время на исправление инструкций. В-четвертых, отсутствие данных и разметки для валидации качества ответов по оптимизированным промптам является “узким местом” для этапа оптимизации промптов, замедляя весь этап оптимизации.

### **Основная часть**

С целью устранения упомянутых ранее недостатков в данной работе предлагается библиотека автоматической оптимизации промптов CoolPrompt, которая включает следующие решения задач.

1) Предлагаемый метод автопроммптинга на основе эволюционных алгоритмов с помощью техники рефлексивной эволюции с целью сохранения контекста первоначальной задачи и поддержания разнообразия популяции промптов – ReflectivePrompt. В отличие от классического эволюционного алгоритма с использованием операции мутации и кроссовера в рефлексивной эволюции происходят этапы рефлексии, разбиваемые на краткосрочную, которая формирует подсказку улучшения промпта из популяции в сравнении с наилучшим, и на долгосрочную, которая

запоминает наилучшую краткосрочную рефлексию на протяжении всего эволюционного процесса, благодаря чему сохраняется первоначальный контекст задачи.

2) Предлагаемый метод автопромтинга на основе фундаментальных моделей с помощью техники дистилляции знаний с целью отсутствия необходимости дообучения LLM и сохранения контекста первоначальной задачи – DistillPrompt. Отличие метода заключается в том, что внедрение и агрегация знаний происходит не на уровне параметров модели, а на уровне промпта, тем самым сохраняя его интерпретируемость и переносимость на другие LLM в сравнении с аналогами.

3) Предлагаемый метод на основе фундаментальных моделей с помощью техники мета-промтинга с целью контроля над изменением промптов для уменьшения “галлюцинаций” в ответах LLM – НуРЕ.

4) Предлагаемый модуль генерации синтетических данных для устранения необходимости в реальных валидационных данных, состоящий из трех этапов. На первом этапе по исходному запросу пользователя с помощью LLM выделяется формальное описание проблемы задачи. На втором этапе задается промпт-инструкция с генерацией базового набора данных формата вход-выход по ранее выделенному описанию проблемы задачи. На третьем этапе задается инструкция для LLM проанализировать базовый набор и описание задачи для расширения синтетических данных пограничными и сложными примерами.

В экспериментальном исследовании были использованы наборы данных на различные типы задач: SQuADv2, GSM8K, CommonGen, AG News, XSum. В качестве метрик оценки использовались F1-macro, Exact Match, BertScore. В качестве моделей использовались различные LLM, включая gpt-3.5-turbo и gpt-4o-mini. В работе демонстрируются следующие результаты. Во-первых, эффективность качества предлагаемых методов автопромтинга на ряде датасетов в сравнении с аналогами. Во-вторых, эффективность в качестве генерации синтетических данных в сравнении с аналогичными модулями других библиотек автопромтинга. В-третьих, высокую масштабируемость переноса промптов, полученных по предлагаемым методам, в сравнении с аналогами.

### **Выводы**

В рамках работы были предложены методы автопромтинга как часть библиотеки CoolPrompt, продемонстрировавшие качество выше и сопоставимо в сравнении с аналогами, также был предложен модуль генерации синтетических данных, продемонстрировавший сопоставимое качество в сравнении с оригинальными данными. Результаты работы внедрены в проект по разработке мета-агентной системы для генерации агентов в ИЦ «Сильный искусственный интеллект в промышленности» Университета ИТМО. Также планируется внедрение предлагаемых методов автопромтинга в индустриальную промышленность.

### **Литература**

1. Zhou Y. et al. Large language models are human-level prompt engineers //The eleventh international conference on learning representations. – 2022.
2. Deng M. et al. Rlprompt: Optimizing discrete text prompts with reinforcement learning //Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. – 2022. – С. 3369-3391.
3. Guo Q. et al. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers //arXiv preprint arXiv:2309.08532. – 2023.
4. Li L., Zhang Y., Chen L. Prompt distillation for efficient llm-based recommendation //Proceedings of the 32nd ACM international conference on information and knowledge management. – 2023. – С. 1348-1357.