УДК 004.89:004.41

ВЛИЯНИЕ МЕТОДОВ ДООБУЧЕНИЯ НА АДАПТАЦИЮ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА ПРОЦЕССОВ РАЗРАБОТКИ

П.А. Красёв
Студент факультета компьютерных систем и сетей БГУИР
paulkrasev@gmail.com

М.А. Калугина
Доцент кафедры информатики, кандидат физико-математических наук, доцент
marina_kalugina@list.ru

П.А. Красёв

В 2022 году окончил Минское суворовское военное училище. Интересы направлены на глубокое обучение, дообучение больших языковых моделей, а также применение генеративных моделей для анализа данных и автоматизации бизнес-процессов.

М.А. Калугина

Окончила Белорусский государственный университет. Область научных интересов связана с исследованием проблем метрической теории диофантовых приближений зависимых величин и приложений математических методов к нейросетевому анализу

Аннотация. В статье рассматриваются методы адаптации больших языковых моделей для автоматизации анализа процессов разработки программного обеспечения. Акцент сделан на интеграции моделей с данными систем управления проектами. Проведено сравнение различных подходов к дообучению: полный fine-tuning, параметрически эффективные методы (LoRA, QLoRA, P-Tuning, IA3), а также методы оптимизации на основе предпочтений (DPO). Оценка качества выполнена по двум группам показателей: стандартные метрики оценки связности и семантической близости текста (Perplexity, BLEU, ROUGE-L, BERTScore) и предметные метрики – релевантность формируемых аналитических заключений и практическая реализуемость предлагаемых шагов по улучшению процессов. Установлено, что наибольшую эффективность демонстрирует комбинированное применение QLoRA и DPO, обеспечивающее высокое качество генераций при существенно меньших

вычислительных затратах по сравнению с полным fine-tuning. Показана необходимость предварительной стратификации данных по типам задач для корректного анализа потоковых метрик.

Ключевые слова: автоматизация анализа процесса разработки программного обеспечения, языковые модели, дообучение, fine-tuning, LoRA, P-Tuning, анализ процессов, метрики потока, BERTScore, ROUGE, релевантность.

Введение. Большие языковые модели открывают новые возможности для автоматизации анализа процессов разработки программного обеспечения, позволяя извлекать знания из накапливаемых данных и формировать рекомендации по его улучшению [1]. Однако применение универсальных моделей в этой предметной области затруднено из-за специфики профессиональной лексики и необходимости интерпретации количественных показателей выполнения работ.

Целью исследования являлся сравнительный анализ методов дообучения языковых моделей для генерации аналитических заключений и практических рекомендаций на основе данных, характеризующих ход разработки [2]. Поэтому ключевой особенностью работы является ориентация на комплексные данные из систем управления проектами: временные ряды метрик потока в сочетании с основными атрибутами задач. Это обусловливает следующие ограничения:

1 Семантический разрыв между текстами общего назначения и предметной областью управления разработкой.

2 Ограниченный объем эталонных аналитических выводов, требующий эффективных методов дообучения при малом количестве размеченных данных.

3 Необходимость совместной обработки числовых временных рядов и генерации связных текстовых заключений.

В работе сравниваются полный fine-tuning, параметрически эффективные методы (LoRA, QLoRA, P-Tuning, IA3) и оптимизация на основе предпочтений (DPO) [3]. Оценка проводится по стандартным NLP-метрикам (Perplexity, BLEU, ROUGE-L, BERTScore) и специальным предметным показателям релевантности выводов и реализуемости рекомендаций.

Описание алгоритма дообучения. В качестве источника информации о задачах разработки используются открытые датасеты, содержащие данные из различных систем управления проектами: текстовые описания, временные метки изменения задач, а также информация о типах задач. Такой подход обеспечивает воспроизводимость экспериментов и исключает зависимость от конкретных коммерческих систем управления проектами.

На основе временных меток для каждой задачи вычисляются различные метрики потока, характеризующие эффективность процесса работы. Вычисление этих метрик выполняется специализированным модулем предобработки, который также осуществляет нормализацию числовых значений и их преобразование в текстовый формат, пригодный для подачи в языковую модель.

Каждый пример включает входной контекст и целевой вывод – аналитическое заключение с рекомендациями. Для балансировки выборки применяется стратификация по типам задач.

Процесс дообучения выполняется на GPU: модель последовательно обучается на примерах с использованием различных методов (полный fine-tuning, LoRA, QLoRA, P-Tuning, IA3, DPO). Контроль качества осуществляется на валидационной выборке по стандартным NLP-метрикам (Perplexity, BLEU, ROUGE-L, BERTScore) и предметным показателям релевантности и реализуемости [4].

Для предотвращения переобучения при малом объеме размеченных данных применяются регуляризация, ранняя остановка на основе метрик валидационной выборки, а также смешанное семплирование, обеспечивающее равномерное представление различных категорий данных в каждом батче.

Влияние выбранного метода на результаты дообучения. В ходе исследования был проведен сравнительный анализ различных методов дообучения языковых моделей на задаче генерации аналитических заключений и рекомендаций по улучшению процессов разработки. Оценивалось их поведение в условиях ограниченного объема размеченных данных и необходимости интеграции числовых метрик потока с текстовым контекстом.

Первым был использован подход полного fine-tuning (full fine-tuning) на базовой модели. Начальные эксперименты проводились на упрощенном подмножестве данных: использовались только текстовые описания задач без временных метрик. Модель обучалась предсказывать аналитическое заключение по заданному набору заголовков задач за интервал. В этой упрощенной постановке full fine-tuning продемонстрировал приемлемое качество: значения ROUGE-L достигали 0.35–0.40, а оценки релевантности были положительными для простых случаев (например, когда требовалось лишь обобщить типы задач).

Успех в базовом текстовом сценарии мотивировал переход к полной постановке задачи – включению метрик потока. Для этого потребовалось модифицировать входной формат: числовые показатели нормализовались и преобразовывались в текстовые описания, которые конкатенировались с текстовым контекстом. Эксперименты показали, что full fine-tuning на таких данных приводит к быстрому переобучению: уже после 3–4 эпох метрики на валидационной выборке начинали ухудшаться, а сгенерированные рекомендации становились шаблонными и не учитывали специфику числовых показателей. Для борьбы с этим применялись стандартные методы регуляризации (dropout 0.1, weight decay), а также ранняя остановка. Однако даже при этом модель демонстрировала высокую чувствительность к начальным значениям весов и требовала тщательного подбора гиперпараметров (скорости обучения, размера батча).

Поскольку полный fine-tuning оказался ресурсоемким и склонным к переобучению при малом объеме данных, следующим шагом стало исследование параметрически эффективных методов (PEFT) [5]. Был использован метод LoRA с ранговой размерностью r=8 и α=16. LoRA позволила дообучать модель, замораживая основные веса и добавляя обучаемые адаптеры [6]. Это дало два ключевых преимущества: во-первых, объем обучаемых параметров сократился с сотен миллионов до нескольких миллионов, что ускорило эксперименты; во-вторых, метод оказался более устойчивым к переобучению. На валидационной выборке LoRA достигла сопоставимых с full fine-tuning значений ROUGE-L (0.38–0.42), но при этом кривые обучения были глаже, а разброс между запусками меньше (рисунок 1).

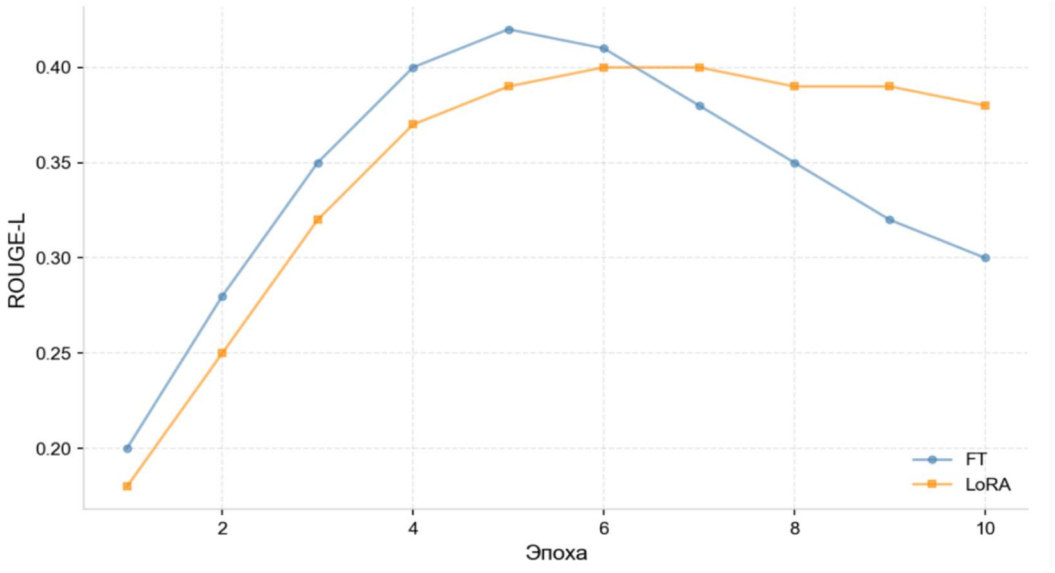

Рисунок 1. Сравнение full fine-tuning и LoRA

Для дальнейшего снижения потребления памяти был применен метод QLoRA, использующий 4-битное квантование базовой модели. QLoRA позволила дообучать модель на GPU с 8 ГБ памяти, тогда как full fine-tuning требовал не менее 24 ГБ [7]. Качество при этом практически не ухудшилось: метрики BERTScore остались на уровне 0.86–0.88 против 0.87–0.89 у LoRA. Однако при работе с QLoRA потребовалась дополнительная настройка гиперпараметров оптимизатора (использовался paged AdamW) для предотвращения нестабильности градиентов.

Параллельно тестировались методы на основе мягких промптов: P-Tuning и IA3 (рисунок 2) [8]. P-Tuning, обучающий непрерывные векторы-промпты, показал более низкую эффективность в рассматриваемой задаче: ROUGE-L не превышал 0.30, а генерируемые тексты часто были несвязными. Вероятно, это связано с тем, что мягкие промпты хуже справляются с интеграцией числовой информации, требующей точного отражения в тексте. IA3, напротив, продемонстрировал результаты, близкие к LoRA, но с еще меньшим числом обучаемых параметров. Однако при малом объеме данных IA3 иногда приводил к нестабильности, выражавшейся в резких скачках функции потерь [9].

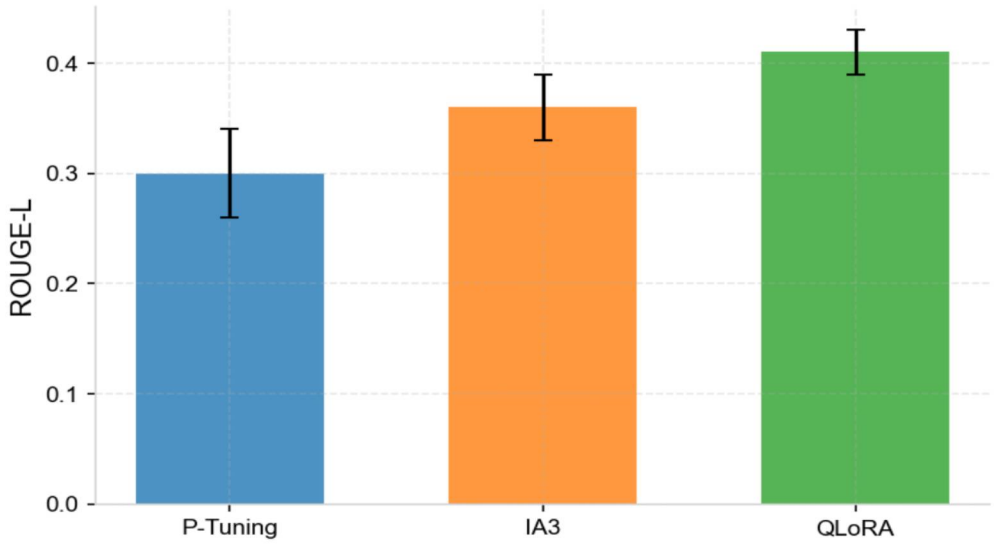

Рисунок 2. Сравнение PEFT методов

После того как модели, дообученные с помощью LoRA и QLoRA, научились генерировать связные заключения, возникла проблема: рекомендации часто были формально правильными, но не учитывали тонкие нюансы, важные с практической точки зрения (например, предлагали увеличить ресурсы, хотя реальной причиной задержек была блокировка зависимостями). Для решения этой проблемы был применен метод DPO [10]. На основе ответов модели были сформированы пары предпочтений: для каждого входного контекста выбирался лучший (релевантный и реализуемый) и худший (формальный или нереалистичный) ответы из числа сгенерированных моделью.

Результаты применения DPO оказались наиболее значимыми. Предметная метрика релевантности выросла с 3.2 до 4.1, а индекс практической реализуемости – с 2.8 до 3.9. При этом стандартные NLP-метрики изменились незначительно (ROUGE-L вырос на 0.02), что подтверждает, что DPO улучшает именно содержательное качество, а не просто текстовое сходство.

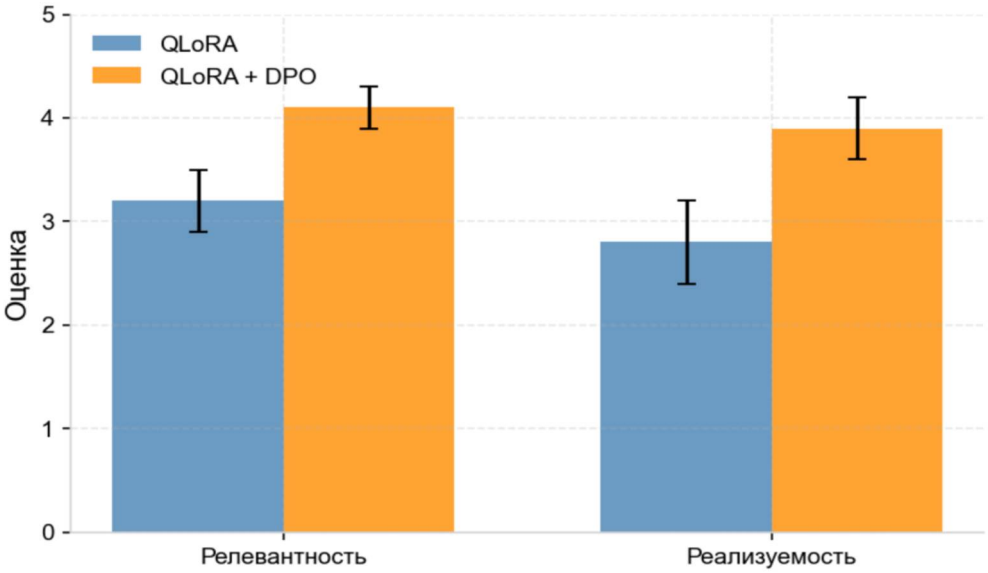

Рисунок 3. Результаты применения DPO

Комбинированное применение QLoRA и DPO позволило достичь наилучшего баланса между вычислительной эффективностью и качеством. QLoRA обеспечила экономию памяти и устойчивость к переобучению, а DPO скорректировала модель в соответствии с сформированными предпочтениями, повысив практическую ценность.

Заключение. В ходе исследования сформирована и экспериментально апробирована методика сравнительного анализа известных методов дообучения языковых моделей применительно к задаче автоматизации анализа процессов разработки программного обеспечения. Реализованный подход включает формирование обучающих примеров на основе открытых датасетов, вычисление метрик потока и их интеграцию с текстовым контекстом, а также применение современных методов адаптации моделей к предметной области. Проведённый сравнительный анализ методов дообучения подтвердил, что выбор алгоритма адаптации определяющим образом влияет на качество генерируемых аналитических заключений и рекомендаций в условиях ограниченного объёма размеченных данных:

1 Полный fine-tuning продемонстрировал быстрое достижение высоких значений на обучающей выборке, однако уже после 4–5 эпох начиналось переобучение, выражавшееся в падении метрик на валидации и шаблонности формируемых рекомендаций. Кроме того, метод предъявляет высокие требования к вычислительным ресурсам, что затрудняет его многократное применение.

2 Параметрически эффективные методы (PEFT) показали существенно более устойчивые результаты. LoRA и QLoRA обеспечили стабильный рост метрик без признаков переобучения. Метод P-Tuning оказался малоэффективным для задач, требующих точной интерпретации числовых показателей, а генерируемые тексты часто теряли связность. IA3 показал результаты, близкие к LoRA, однако при малом объёме данных наблюдались эпизодические выбросы функции потерь, что снижает его надёжность.

3 Оптимизация на основе предпочтений (DPO) позволила качественно улучшить содержательную сторону генераций. Комбинированное применение QLoRA и DPO привело к росту предметных метрик: релевантности аналитических выводов и индекса практической реализуемости рекомендаций. При этом стандартные NLP-метрики изменились незначительно, что подтверждает способность DPO корректировать модель в соответствии с заданными предпочтениями без ухудшения формального качества текста.

Таким образом, проведённое сравнение показало, что для создания эффективных аналитических систем на основе языковых моделей в предметной области управления разработкой недостаточно применения базовых методов дообучения.

Полный fine-tuning оказался неприменим из-за склонности к переобучению и высоких ресурсных затрат. Среди PEFT-методов наилучшее сочетание стабильности, качества и вычислительной эффективности продемонстрировали LoRA и QLoRA, при этом QLoRA предпочтительнее при ограниченных ресурсах.

Дополнительное применение DPO позволило значительно повысить практическую значимость генерируемых рекомендаций, что является ключевым для задач аналитической поддержки.

Наиболее эффективной признана комбинация QLoRA и DPO, обеспечивающая высокое качество как по стандартным NLP-метрикам, так и по предметным показателям релевантности и реализуемости.

В дальнейшей работе планируется расширение набора анализируемых метрик (включение показателей качества кода, данных из систем непрерывной интеграции), исследование возможности генерации не только текстовых заключений, но и визуальных аналитических материалов, а также применение методов мультиагентного подхода для моделирования взаимодействия различных ролей в процессе разработки.

Список литературы

[1] Hou X., Zhao Y., Liu Y., и др. Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology. 2024;33(8):1-79. DOI: 10.1145/3695988.

[2] Vaswani A., Shazeer N., Parmar N., и др. Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS); 2017;30:5998-6008.

[3] Devlin J., Chang M.-W., Lee K., и др. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL); 2019:4171-4186. DOI: 10.18653/v1/N19-1423.

[4] Touvron H., Lavril T., Izcard G., и др. LLaMA: Open and Efficient Foundation Language Models. arXiv; 2023. DOI: 10.48550/arXiv.2302.13971.

[5] Ding N., Qin Y., Yang G., и др. Parameter-Efficient Fine-Tuning of Large-Scale Pre-Trained Language Models. Nature Machine Intelligence. 2023;5:220-235. DOI: 10.1038/s42256-023-00626-4.

[6] Hu E.J., Shen Y., Wallis P., и др. LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations (ICLR); 2022. DOI: 10.48550/arXiv.2106.09685.

[7] Dettmers T., Pagnoni A., Holtzman A., и др. QLoRA: Efficient Finetuning of Quantized Large Language Models. Advances in Neural Information Processing Systems (NeurIPS); 2023;36. DOI: 10.48550/arXiv.2305.14314.

[8] Liu X., Ji K., Fu Y., и др. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL); 2022:61-68. DOI: 10.18653/v1/2022.acl-short.8.

[9] Liu H., Tam D., Muqeeth M., и др. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. Advances in Neural Information Processing Systems (NeurIPS); 2022;35:1950-1965.

[10] Rafailov R., Sharma A., Mitchell E., и др. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. Advances in Neural Information Processing Systems (NeurIPS); 2023;36. DOI: 10.48550/arXiv.2305.18290.

Авторский вклад

Красёв Павел Александрович – выбор задачи исследования, реализация архитектуры экспериментального стенда с модулем предобработки данных, проведение экспериментов со сбором стандартных NLP-метрик и предметных показателей, анализ полученных результатов.

Калугина Марина Алексеевна – постановка проблемы, научное руководство исследованием методов адаптации языковых моделей для анализа процессов разработки, консультации по математическому аппарату алгоритмов.

THE IMPACT OF FINE-TUNING METHODS ON THE ADAPTATION OF LANGUAGE MODELS FOR SOFTWARE DEVELOPMENT PROCESS ANALYSIS

P.A. Krasyov
Student of BSUIR

M.A. Kalugina
Associate Professor of Informatics Department of the BSUIR

Abstract. This article examines methods for adapting large language models to automate analytical support in software development processes. The focus is on integrating models with project management system data. Various fine-tuning approaches are compared: full fine-tuning, parameter-efficient methods (LoRA, QLoRA, P-Tuning, IA3), and preference optimization methods (DPO). Quality assessment is performed using two groups of metrics: standard metrics for evaluating text coherence and semantic similarity (Perplexity, BLEU, ROUGE-L, BERTScore) and task-specific metrics–the relevance of generated analytical conclusions and the practical feasibility of proposed process improvement steps. The combined application of QLoRA and DPO demonstrates the highest effectiveness, providing high-quality generations with substantially lower computational costs compared to full fine-tuning. The necessity of preliminary data stratification by task types for correct analysis of flow metrics is shown.

Keywords: software development analysis automation, language models, fine-tuning, LoRA, P-Tuning, process analysis, flow metrics, BERTScore, ROUGE, relevance.