УДК 004.853

Исследование методов интеграции внешних инструментов в системе автоматического машинного обучения, управляемой БЯМ

Лапин А. А.[1], Чумаков С.В.[1], Марьян П.А. [1]
Научный руководитель – канд. техн. наук, доцент Никитин Н.О.[1]
[1]Университет ИТМО
a.lapin@itmo.ru

Введение

AutoML прошёл путь от традиционных инструментов, таких как AutoGluon [1] и H2O [2], до агентов на основе больших языковых моделей, способных автоматизировать рабочие процессы с помощью естественного языка, таких как AutoKaggle [3] и AIDE [4]. Однако современные системы сталкиваются с проблемой эффективной интеграции специализированных внешних библиотек, что приводит к некорректным вызовам API и неэффективным пайплайнам.

Основная часть

Для решения проблемы эффективной интеграции внешних знаний предлагается AgenticML-Tools — модульный мультагентный фреймворк, использующий двухфазный цикл "Think-Act" и унифицированную архитектуру для интеграции знаний. Analyst отвечает за исследование структуры данных и разведочный анализ. Researcher выполняет глубокий поиск подходящих библиотек через механизм Deep Research и формирует отчет с примерами использования. Manager на основе полученных отчетов создает детальный план решения и спецификацию для кодера. Coder реализует план через итеративную генерацию и выполнение кода с оптимизацией параметров. Debugger изолированно перехватывает и исправляет ошибки, не влияя на основной механизм решения задачи.

Ключевая особенность системы — шесть механизмов интеграции знаний о внешних инструментах. Self-RAG по коду использует векторный поиск с самопроверкой релевантности и детекцией галлюцинаций. GraphRAG по коду на базе фреймворка Cognee [5] комбинирует векторный поиск с графами знаний, сохраняя структурные связи. Deep Search Agent реализует ReAcT-агента с древовидным мышлением для рекурсивного сбора информации и отдельным субагентом для суммаризации длинных текстов. Шаблоны описывают типовые паттерны применения целевых библиотек. LLM-генерируемая документация описывает использование доступных функций и параметров. Программируемый API алгоритмически извлекает документацию на базе исходного кода, предоставляя примеры использования с ранжированием по приоритету.

Выводы

Эксперименты на 12 задачах показали, что мультиагентное планирование критически важно для работы со специализированными библиотеками, а разные методы интеграции эффективны для разных типов задач: Self-RAG — для задач, требующих точного знания API; шаблоны — для структурированных постановок; веб-поиск — для отладки. Абляционный анализ не выявил преимущества специализированных кодовых моделей перед универсальными, что указывает на приоритет широты знаний над узкой оптимизацией. Мультиагентная архитектура также позволила успешно применять модели с ограничениями контекста, в то время как прочие конфигурации терпели неудачу.

Модульность AgenticML-Tools обеспечивает лёгкое расширение новыми инструментами и стратегиями интеграции. Ограничения связаны с зависимостью от качества документации и необходимостью ручного создания шаблонов. Дальнейшие направления включают поддержку мультимодальных задач, автоматическую генерацию шаблонов и адаптивный выбор стратегий под специфику задачи.

Литература

1. Erickson N. et al. Autogluon-tabular: Robust and accurate automl for structured data //arXiv preprint arXiv:2003.06505. – 2020.
2. LeDell E. et al. H2o automl: Scalable automatic machine learning //Proceedings of the AutoML Workshop at ICML. – 2020. – Т. 2020. – С. 24.
3. Li Z. et al. Autokaggle: A multi-agent framework for autonomous data science competitions //arXiv preprint arXiv:2410.20424. – 2024.
4. Jiang Z. et al. Aide: Ai-driven exploration in the space of code //arXiv preprint arXiv:2502.13138. – 2025.
5. Topoteretes. Cognee: Build and manage ai memory. – 2024. – URL: https://github.com/topoteretes/cognee

/_____/ Марьян П.А.

/_____/ Никитин Н.О.