

Парсинг логов с использованием LLM с самогенерированным обучением в контексте и самокоррекцией

Дата: 2025-02-03 00:00:00

Ссылка на исследование: <https://arxiv.org/pdf/2406.03376>

Рейтинг: 75

Адаптивность: 85

Ключевые выводы:

Исследование представляет AdaParser - эффективный и адаптивный фреймворк для парсинга логов с использованием больших языковых моделей (LLM). Основная цель - преодолеть ограничения существующих парсеров логов, которые плохо справляются с эволюционирующими логами. AdaParser превосходит современные методы по всем метрикам точности, даже в сценариях без предварительных данных (zero-shot), благодаря использованию самогенерируемого обучения в контексте (SG-ICL) и механизма самокоррекции.

Объяснение метода:

Исследование предлагает адаптивный фреймворк для парсинга логов с использованием LLM, демонстрируя инновационные подходы к самокоррекции и обучению в контексте. Методы могут быть адаптированы для различных LLM и применены в других задачах с эволюционирующими данными. Концепции самогенерируемого обучения и самокоррекции имеют широкий потенциал применения, хотя требуют некоторой адаптации для широкой аудитории.

Ключевые аспекты исследования: 1. **AdaParser** - адаптивный фреймворк для парсинга логов, использующий большие языковые модели (LLM) с самогенерируемым обучением в контексте (SG-ICL) и самокоррекцией для эффективной обработки эволюционирующих логов.

Самогенерируемое обучение в контексте (SG-ICL) - компонент, который поддерживает динамический набор кандидатов из ранее сгенерированных шаблонов и выбирает демонстрации из этого набора для создания более точных запросов к LLM.

Корректор шаблонов - новый компонент, который использует LLM для исправления потенциальных ошибок парсинга в шаблонах, которые она генерирует, улучшая точность парсинга.

Древовидный парсер - компонент, использующий дерево парсинга для хранения шаблонов, сгенерированных LLM, и быстрого сопоставления шаблонов для новых сообщений логов, повышая эффективность.

Адаптивность к эволюционирующим логам - способность фреймворка работать с меняющимися логам без необходимости в обширных исторических данных, что делает его применимым в реальных сценариях.

Дополнение: Исследование AdaParser не требует дообучения или специального API для реализации его основных концепций. Хотя авторы использовали API для ChatGPT в своих экспериментах, они также продемонстрировали, что фреймворк работает с различными LLM, включая локальные открытые модели (DeepSeek-v2-chat и Qwen-1.5-72B-chat), которые могут быть развернуты локально.

Основные концепции и подходы, которые можно применить в стандартном чате:

Самогенерируемое обучение в контексте (SG-ICL) - пользователи могут сохранять успешные примеры взаимодействия с LLM и использовать их как демонстрации в будущих запросах. Например, если LLM успешно структурировала какой-то текст, этот пример можно включить в следующий запрос на структурирование похожего текста.

Стратегии самокоррекции - пользователи могут реализовать двухэтапный подход к запросам: сначала получить ответ, а затем задать уточняющий вопрос, указывающий на возможные ошибки. Например: "Проверь, правильно ли ты структурировал этот текст. Обрати внимание на X и Y."

Верификация результатов - пользователи могут запрашивать LLM проверить собственные ответы, задавая конкретные критерии проверки.

Динамический набор примеров - пользователи могут поддерживать и обновлять библиотеку успешных примеров взаимодействия с LLM для разных типов задач.

Результаты от применения этих концепций: - Повышение точности ответов LLM - Лучшая адаптация к специфическим задачам пользователя - Снижение необходимости в сложном промптинге для каждого запроса - Более эффективное использование контекстного окна LLM - Возможность работы с эволюционирующими данными и требованиями без необходимости в переобучении

Эти подходы особенно полезны в сценариях, когда требуется высокая точность или структурированный вывод, например при анализе документов, извлечении структурированной информации из текста или формализации знаний.

Анализ практической применимости: 1. **AdaParser как фреймворк для парсинга логов - Прямая применимость:** Высокая. Пользователи, работающие с логам, могут использовать AdaParser для более точного парсинга и анализа логов в своих

системах, включая сценарии с ограниченными историческими данными. - **Концептуальная ценность:** Значительная. Демонстрирует возможности LLM для структурирования неструктурированных данных и адаптации к изменениям без переобучения. - **Потенциал для адаптации:** Высокий. Фреймворк совместим с различными LLM, включая локальные открытые модели, что решает проблемы конфиденциальности.

Самогенерируемое обучение в контексте (SG-ICL) **Прямая применимость:** Средняя. Техника может быть адаптирована для других задач, где требуется обучение LLM на ограниченных данных. **Концептуальная ценность:** Высокая. Показывает, как LLM могут адаптироваться к новым задачам без обширных демонстраций. **Потенциал для адаптации:** Высокий. Подход применим в различных сценариях, где данные постоянно эволюционируют.

Корректор шаблонов

Прямая применимость: Средняя. Концепция самокоррекции LLM может быть применена в других областях для повышения точности. **Концептуальная ценность:** Высокая. Демонстрирует, как можно преодолеть ограничения LLM через направленную самокоррекцию. **Потенциал для адаптации:** Высокий. Стратегии самокоррекции могут быть адаптированы для других задач, требующих высокой точности.

Древовидный парсер

Прямая применимость: Средняя. Техника может быть применена для оптимизации использования LLM в других задачах. **Концептуальная ценность:** Средняя. Показывает, как сочетать традиционные алгоритмы с LLM для повышения эффективности. **Потенциал для адаптации:** Средний. Подход может быть адаптирован для других задач, где требуется эффективное кэширование результатов LLM.

Адаптивность к эволюционирующим данным

Прямая применимость: Высокая. Методология может быть применена к любым задачам с эволюционирующими данными. **Концептуальная ценность:** Высокая. Демонстрирует подход к работе с меняющимися данными без необходимости переобучения модели. **Потенциал для адаптации:** Высокий. Концепция применима в широком спектре задач машинного обучения и анализа данных.

Prompt:

Использование знаний из исследования AdaParser в промптах для GPT ##
Ключевые знания из исследования

Исследование AdaParser демонстрирует эффективный подход к парсингу логов с использованием: - Самогенерируемого обучения в контексте (SG-ICL) - Механизма самокоррекции - Древовидного парсера для эффективной обработки

Пример промпта для парсинга логов с использованием принципов AdaParser

[=====] Действуй как продвинутый парсер логов с функциями самообучения и самокоррекции, основанный на методологии AdaParser.

Вот набор логов, которые нужно проанализировать: [ВСТАВИТЬ ЛОГИ ЗДЕСЬ]

Выполни следующие шаги: 1. Сгенерируй шаблоны для каждого уникального типа лог-сообщения, абстрагируя переменные части (IP-адреса, временные метки, ID и т.д.) 2. Сгруппируй логи по этим шаблонам 3. Проведи самокоррекцию шаблонов, проверяя: - Нет ли слишком широких шаблонов, объединяющих разные типы сообщений - Нет ли слишком специфичных шаблонов, разделяющих однотипные сообщения 4. Для каждого шаблона извлеки ключевые переменные и их значения

Представь результаты в структурированном формате: - Список шаблонов с количеством соответствующих им сообщений - Для каждого шаблона: пример исходного логa и извлеченные переменные - Статистика распределения сообщений по шаблонам [=====]

Как работают знания из исследования в этом промпте

Применение SG-ICL: Промпт инструктирует модель генерировать шаблоны на основе примеров логов и использовать их для дальнейшего анализа, что имитирует самогенерируемое обучение в контексте.

Механизм самокоррекции: Включен явный шаг проверки и исправления ошибок парсинга, фокусируясь на двух типах ошибок, выявленных в исследовании: слишком широкие шаблоны и неточные шаблоны.

Древовидный подход: Хотя GPT не может создать настоящую древовидную структуру, промпт направляет модель на группировку подобных логов, что концептуально соответствует древовидному парсеру AdaParser.

Адаптивность к новым форматам: Промпт не предполагает предварительных знаний о форматах логов, что позволяет модели адаптироваться к новым и эволюционирующим логам, как это делает AdaParser в режиме zero-shot.

Такой промпт позволяет максимально использовать способности GPT для анализа логов, применяя научно обоснованные подходы из исследования AdaParser.