УДК 004.89

МЕТОД ИЗВЛЕЧЕНИЯ И АВТОМАТИЧЕСКОЙ ФОРМАЛИЗАЦИИ ПРОЕКТНЫХ ЗНАНИЙ ДЛЯ ЯЗЫКОВЫХ МОДЕЛЕЙ ИИ, ПУТЕМ ГЕНЕРАЦИИ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЯЗЫКОВ

Ковальчук М. А.[1], Новикова А. С.
Научный руководитель – канд. техн. наук, доцент Насонов Д. А.[1]
[1]Университет ИТМО
mlhakov2011@gmail.com

Введение

В инженерной отрасли планирование проектов опирается на строгие модели, такие как RCPSP и многочисленные расширения, требующие четкого определения задач, ресурсов, графиков и логических ограничений [1]. В реальных проектах эти знания часто представлены в виде разрозненной документации на естественном языке (технические требования, контракты, отчеты, правила и корпоративные методологии), что делает ручную формализацию дорогостоящей и подверженной ошибкам. Большие языковые модели (LLM) обладают способностью извлекать и обобщать знания, но свободная форма запроса плохо согласуется с формальной природой ограничений, что затрудняет формулирование задач воспроизводимым образом. Цель данной работы — предложить метод, который автоматически извлекает знания о проекте из неструктурированных источников и преобразует их в проверяемое представление, подходящее для последующей оптимизации и надежного взаимодействия с LLM.

Основная часть

Предлагается гибридный рабочий процесс, в котором LLM используется в качестве компонента извлечения знаний, а формальные грамматики — в качестве механизма для сбора и проверки знаний. Рабочий процесс основан на предметно-ориентированном языке описания проекта (DSL), который обеспечивает единый интерфейс между людьми, LLM и алгоритмами оптимизации [2].

Рабочий процесс состоит из пяти этапов.

1) Онтология предметной области и промежуточное представление (IR). Определяются типы сущностей, такие как работа (деятельность), рабочие пакеты, ресурсы (возобновляемые/невозобновляемые), календари, зависящие от времени, временные задержки и атрибуты (длительность, объем, стоимость и квалификация). В IR также хранится регистрация фактов (ссылка на фрагмент документа).

2) Извлечение фактов. Корпус документации делится на фрагменты, и LLM извлекает структурированные записи IR из каждого фрагмента, используя схему (например, JSON), которая минимизирует вариативность формул и облегчает последующую проверку.

3) Нормализация и слияние. Вы определяете разрешение сущностей (разрешение объектов), нормализацию объектов и обнаружение измерений и пробелов (например, зависимости без идентификаторов работ или ресурсы без пропускной способности).

4) Синтез DSL. На основе онтологии и накопленного IR (а) автоматически генерируется грамматика DSL (EBNF), (б) генерируются типичные шаблонные

структуры и (в) генерируется экземпляр DSL, специфичный для проекта, вместе с объявлениями и ограничениями. DSL может быть скомпилирован как в формулировки RCPSP, так и в языки планирования (например, PDDL) для внешних решателей [3].

5) Валидация и завершение цикла. Парсер DSL обеспечивает синтаксическую корректность, а семантические проверки (цикличность графа предшествования, непревышающие пропускные способности ресурсов и корректность календарей) приводят к более раннему обнаружению ошибок. Диагностика ошибок используется для генерации целевых запросов уточнения для документов или эксперта, что сокращает объем ручной работы. Как формат запроса, так и ограничение генерации (ограниченное декодирование) используются для взаимодействия с LLM, что позволяет получить корректный результат построения [4].

Выводы

Предложенный метод автоматической формализации проектных знаний обеспечивает переход от неоднозначных текстовых артефактов к проверяемому и воспроизводимому представлению. Использование автоматически синтезированного DSL:

- уменьшает неоднозначность спецификации ресурсов и ограничений за счет типизированных конструкций;

- повышает полноту описания за счет проверок целостности и контролируемого цикла уточнения;

- создает единый интерфейс для последующей оптимизации и безопасного применения LLM в задачах планирования.

Полученные результаты могут быть использованы при построении корпоративных систем планирования, где необходимо быстро интегрировать LLM в существующие процессы без ручного переписывания документации в математические модели.

Литература

1. Hartmann S., Briskorn D. A survey of variants and extensions of the resource-constrained project scheduling problem // European Journal of Operational Research. 2010. Vol. 207, no. 1. P. 1–14.

2. Fowler M. Domain-Specific Languages. Boston: Addison-Wesley, 2010. 640 p.

3. McDermott D. et al. PDDL – The Planning Domain Definition Language. Technical Report. 1998.

4. Scholak T., Schucher N., Bahdanau D., de Vries H. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models // Proceedings of EMNLP. 2021.