Методы и средства автоматизированного построения онтологий при помощи open-source библиотек с применением больших языковых моделей

Основные подходы

Область применения больших языковых моделей (LLM) для автоматизации процессов инженерии онтологий (Knowledge Graph and Ontology Engineering, KGOE) быстро развивается, однако она далека от наличия единого стандарта или парадигмы. Анализ предоставленных источников позволяет выделить четыре доминирующие архитектурные и методологические стратегии:

- Fine-Tuning
- Zero-Shot/Few-Shot Prompting
- Использование подготовленных схем
- Human-in-the-Loop

Fine-Tuning

Fine-Tuning, представляет собой наиболее ресурсоемкий, но потенциально самый точный подход. Его суть заключается не в использовании готовой модели "из коробки", а в дополнительном обучении (fine-tuning) существующей LLM на специально подготовленном корпусе данных для решения узкой задачи извлечения онтологии. Представителем этого направления является метод OLLM (Ontology Learning with Large Language Models), который был разработан для построения таксономической основы онтологии — то есть иерархии "is-a" отношений — с нуля. Исследователи используют базовую модель Mistral 7B v0.2 и применяют к ней технику Low-Rank Adaptation (LoRA) для эффективного обучения. Ключевой особенностью OLLM является использование кастомной функции регуляризации, которая маскирует вклад в функцию потерь часто встречающихся отношений во время обучения. Это позволяет модели лучше обобщаться и избегать переобучения на частые концепты, которые могут дестабилизировать процесс обучения. После обучения модель генерирует подграфы целевой онтологии,

которые затем агрегируются и очищаются в постобработке. Преимущество этого подхода заключается в достижении большой семантической точности и структурной целостности, особенно в рамках узкоспециализированной задачи построения таксономий. Эксперименты показали, что OLLM превосходит другие методы, такие как паттерны Hearst или простой промптинг (Fuzzy F1 0.915 на Wikipedia). OLLM демонстрирует отличную способность к переносу знаний между доменами: после обучения на 2048 примерах из Wikipedia, модель смогла успешно адаптироваться к новому, совершенно иному домену arXiv, построив там логичную и семантически правильную таксономию. Однако этот подход имеет существенные недостатки. Он требует значительных вычислительных мощностей. Обучение на одном GPU A100 заняло 12 часов, а полный цикл работы потребовал 19 часов вычислений. Кроме того, он является крайне специализированным: OLLM сконцентрирован исключительно на таксономических отношениях и не способен извлекать неразветвленные связи или логические аксиомы. В-третьих, воспроизведение результатов может быть сложным из-за неизвестного состава данных, на которых была предварительно обучена базовая модель Mistral.

Zero-Shot/Few-Shot Prompting

Подход Zero-Shot/Few-Shot Prompting, является наиболее распространенным и доступным. Он использует готовые, широко доступные LLM без необходимости в дополнительном обучении. Его успех опирается на промпт-инжиниринг — формулирование запросов к модели, для достижения желаемого результата. Этот подход характеризуется своей гибкостью и быстротой внедрения. Например, LLMs4OL систематически оценивает различные LLM (от BERT до Flan-T5-XL) на трех стандартных задачах: Term Typing (определение типа термина), Type Taxonomy Discovery (поиск иерархии классов) и Type Non-Taxonomic Relation Extraction (извлечение связей). Результаты показывают, что модели без дообучения могут быть недостаточно надежны для сложных инженерных задач. Другим примером является NeOn-GPT, который использует GPT-3.5 с использованием методологии NeOn. Этот процесс включает несколько этапов: генерацию черновика онтологии с помощью промптов, синтаксическую проверку с помощью RDFLib, проверку логической согласованности с помощью HermiT ризонера и

исправление ошибок с помощью OOPS API. Этот пример демонстрирует, как LLM может стать центральным элементом более крупной, управляемой системы, где его способности дополняются традиционными инструментами Semantic Web. ConExion — еще один проект, использующий zero-shot prompting для извлечения всех понятий из документа, а не только ключевых слов, и сравнивающий производительность различных LLM, таких как Llama-2 и GPT-3.5 Turbo. Главное преимущество этого подхода — минимальные требования к вычислительным ресурсам и возможность быстро начать работу. Однако его качество сильно зависит от качества промптов, и он часто страдает от проблем несогласованности, плоских иерархий, а также галлюцинаций. Для академической работы этот подход является наиболее доступным и быстрым для прототипирования и тестирования гипотез.

Использование подготовленных схем

Подход и использованием подготовленных схем возник как прямой ответ на проблемы несогласованности и хаотичности, свойственные zero-shot подходу. Вместо того чтобы позволять LLM свободно извлекать любые факты, этот метод заставляет модель заполнять заранее определенную схему или онтологию. Это обеспечивает высокую степень семантической согласованности и предсказуемости результатов, что является критически важным для создания практически применимых систем. LangExtract от Google является одним из самых продвинутых инструментов в этой категории. Он позволяет определить схему извлечения с помощью Pydantic моделей, а затем использует few-shot примеры для того, чтобы гарантировать, что LLM будет выводить данные, точно соответствующие этой схеме. LangExtract поддерживает широкий спектр моделей, включая Gemini и OpenAI, и предоставляет продвинутые функции, такие как чанкинг для обработки длинных документов, параллельную обработку для повышения скорости и интерактивная HTML-визуализация для анализа результатов. Другой пример — OmEGa, который использует Task-Centric Ontology (TCO) для извлечения информации из технических документов, применяя schema-based data augmentation для повышения производительности в условиях малых данных. Бенчмарк Text2KGBench был специально создан для оценки LLM на основе их способности генерировать графы, которые точно соответствуют

заданной входной онтологии, что делает его идеальным инструментом для исследования именно этого подхода. Преимущество этого подхода заключается в том, что он напрямую решает проблему контроля и консистентности. Недостатком является необходимость предварительного создания детальной схемы, что само по себе является трудоемкой задачей инженерии онтологий.

Human-in-the-Loop

Метод Human-in-the-Loop (с участием человека в цикле) признает, что текущие LLM не являются панацеей, и эффективно использует их в паре с человеком-экспертом для итеративного улучшения результатов. Этот подход распределяет труд: LLM выполняет рутинные, трудоемкие задачи, а человек-эксперт занимается контролем качества, принятием решений и верификацией. SCHEMA-MINERpro является ярким примером такого многоступенчатого рабочего процесса. Он начинается с использования LLM для первоначальной генерации схемы из текстовых описаний, но затем включает в себя несколько итераций, где эксперт вносит корректировки, использует более широкий корпус научных статей для уточнения и, в конечном итоге, помогает в привязке сгенерированных схем к формальным онтологиям, таким как QUDT. ODKE+ (Open-Domain Knowledge Extraction) добавляет критически важный модуль Grounding, где второй, более легкий LLM используется для проверки того, основан ли каждый факт, извлеченный первым LLM, на предоставленном контексте. Эта двойная проверка значительно снижает количество галлюцинаций и повышает фактическую точность. OntoChat — это среда, которая помогает экспертам формулировать пользовательские истории и извлекать вопросы компетенций (CQs), которые служат основой для онтологии, тем самым упрощая и ускоряя процесс ее проектирования. Преимущество этого подхода заключается в возможности создания высококачественных, профессиональных онтологий, которые были бы невозможны при полностью автоматическом подходе.

А нализ open-source библиотек для извлечения онтологий

OntoGPT

OntoGPT является одной из наиболее известных и активно развивающихся библиотек, ориентированных на извлечение структурированной информации с помощью LLM. Её технологическим ядром является метод SPIRES (Structured Prompt Interrogation and Recursive Extraction of Semantics), который работает в режиме zero-shot и предназначен для заполнения заранее определенных структур, описанных в LinkML схеме. OntoGPT поддерживает широкий спектр LLM-провайдеров, включая OpenAI (GPT-3.5, GPT-4), Anthropic (Claude), Azure и локальные модели через ollama (например, llama3), что делает его гибким инструментом. Он доступен для установки через pip (pip install ontogpt) и предполагает взвимодействие через командную строку (CLI), так и веб-интерфейс. Проект поддерживается Monarch Initiative и имеет сильную связь с биомедицинским сообществом, предлагает готовые шаблоны для извлечения данных из научной литературы. Однако у OntoGPT есть и серьезные недостатки. Одной из главных проблем является его производительность при работе с большими онтологиями. Процесс "grounding" (сопоставления извлеченных сущностей с терминами в онтологии) является основным узким местом. Когда время исполнения может измеряться часами. Переход к "slim" версиям онтологий может ускорить процесс в 6 раз, но это достигается ценой потери покрытия (50% потерянных сопоставлений). Кроме того, пользователи отмечаютнизкую степень извлечения отношений в OntoGPT (лишь ~700 отношений из 1000 документов). Также существует зависимость от API OpenAI, хотя теоретически поддерживаются и другие модели, их интеграция не всегда гладкая из-за специфического парсинга ответов, оптимизированного под OpenAI. Библиотека активно используется и развивается.

https://deepwiki.com/monarch-initiative/ontogpt
https://github.com/monarch-initiative/ontogpt

LLMs4OL

LLMs4OL представляет собой другой класс инструментов — это не монолитная библиотека, а проект и репозиторий (2023 год). Его цель — оценить роль LLM в трех ключевых задачах инженерии онтологий: Term Typing, Type Taxonomy Discovery и Type Non-Taxonomic Relation Extraction. Основное преимущество LLMs4OL заключается в его академическом признании. Однако для практического применения он менее удобен.

https://github.com/HamedBabaei/LLMs4OL
https://arxiv.org/abs/2307.16648

LangExtract

LangExtract, разработанный Google, является современным инструментом, сфокусированным на подходе с использованием подготовленных схема. Его ключевая особенность — это механизм (enforced output schema), который гарантирует, что LLM будет генерировать структурированный JSON, соответствующий предопределенному Pydantic-классу. Это решает одну из главных проблем чистого промптинга — несогласованность и неструктурированность вывода. LangExtract поддерживает широкий спектр LLM-провайдеров, включая Gemini, OpenAI и локальные модели через Ollama, и предоставляет множество продвинутых функций, которые критически важны для обработки реальных данных.

Для обработки длинных документов используется чанкинг, параллельная обработка для повышения скорости, визуализация результатов в виде интерактивных HTML-страниц, а также поддержка кастомных провайдеров. Библиотека находится в активной разработке. Его недостатком является необходимость явного определения схемы, что требует усилий по проектированию. Кроме того, для некоторых моделей, таких как OpenAI, некоторые механизмы контроля вывода пока не поддерживаются.

https://github.com/google/langextract
https://pypi.org/project/langextract/
https://langextract.net/

NeOn-GPT

NeOn-GPT представляет собой гибридный подход, который сочетает в себе использование LLM и структурированность традиционных методологий инженерии онтологий. Он вызывает LLM для извлечения фактов, а интегрирует его в четырехэтапный рабочий процесс, основанный на методологии NeOn:

1 генерация черновика онтологии

2 синтаксическая проверка с помощью RDFLib

3 проверка логической согласованности с помощью HermiT reasoner

4 исправление ошибок с помощью OOPS API

Этот подход позволяет создавать онтологии, которые не только семантически богаты, но и формально корректны. Исследования показали, что такой workflow значительно превосходит zero-shot промптинг по качеству генерируемых онтологий. Однако у NeOn-GPT есть и ограничения. Во-первых, он зависит от одного типа LLM — GPT-3.5, который показал лучшие результаты в этом конкретном контексте. Во-вторых, он склонен генерировать слишком простые и плоские иерархии и не использует более сложные конструкции OWL, такие как кардинальность или универсальные ограничения. Это делает его хорошим инструментом для быстрого прототипирования, но не для создания сложных, формализованных онтологий.

https://github.com/andreamust/NEON-GPT

https://2024.eswc-conferences.org/wp-content/uploads/2024/05/77770034.pdf

OLLM

OLLM (Ontology Learning with Large Language Models), представленный на NeurIPS 2024, использует fine-tuning для построения онтологий. Его основной фокус — на построении таксономической основы, и он достигает в этом высокой точности благодаря кастомной функции потерь, которая борется с переобучением. Важным преимуществом OLLM является наличие собственного набора данных и репозитория с исходным кодом на GitHub, что обеспечивает полную воспроизводимость исследований. Однако этот подход имеет очень высокие требования к вычислительным ресурсам и узкую специализацию, что делает его менее доступным для широкого круга исследователей и менее

универсальным по сравнению с подходами онованными на промптинге.

https://deepwiki.com/andylolu2/ollm/2-getting-started

https://arxiv.org/abs/2410.23584