Научная статья
УДК 004.386
DOI: https://doi.org/10.18127/j19998554-202506-04

Эволюция методов извлечения и структурирования данных из текста в JSON и XML

Н.Н. Олтян[1]
[1] Финансовый университет при Правительстве Российской Федерации (Москва, Россия)
[1] nikitaoltyan@mail.ru

Аннотация

Постановка проблемы. Современные проблемы анализа данных и машинного обучения требуют преобразования текстовой информации в частично структурированные форматы (JSON, XML). Существующие подходы, включая основанные на правилах, глубоком обучении и больших языковых моделях (LLM), имеют существенные недостатки, такие как высокая стоимость поддержки, потребность в большом объеме разметки и нестабильность вывода.

Цель. Систематизировать подходы к структурированию данных, включая алгоритмы поствалидации, а также разработать концепцию детерминированного конвейера для обеспечения синтаксической и семантической валидности.

Результаты. Проведен анализ современных методов структурирования данных, который показал, что ни один из них не обеспечивает одновременно строгую типизацию, детерминированность вывода, а также синтаксическую и семантическую валидность. Выявлены ключевые ограничения существующих решений. Предложен подход, основанный на использовании детерминируемых конвейеров, который сочетает сильные стороны LLM и механизмов формальной валидации для достижения надежного структурирования данных.

Практическая значимость. Разработанный подход позволяет преодолеть ограничения существующих методов, повысить надежность и качество структурирования текстовых данных. Это облегчит разработку систем анализа данных, требующих точного и предсказуемого преобразования неструктурированной информации.

Ключевые слова

Анализ данных, извлечение информации, структурирование данных, большие языковые модели (LLM), валидация данных, JSON, XML

Для цитирования

Олтян Н.Н. Эволюция методов извлечения и структурирования данных из текста в JSON и XML // Нейрокомпьютеры: разработка, применение. 2025. Т. 27. № 6. С. 37–49. DOI: 10.18127/j19998554-202506-04

A brief version in English is given at the end of the article

Введение

Современные цифровые экосистемы опираются на данные как ключевой ресурс для аналитики. Однако значительная часть этих данных представлена в неструктурированной форме – в виде текстов, документов и записей диалогов. Для их последующего использования необходим перевод информации в форматы с условной, но формально заданной структурой, такие как JSON и XML, которые в дальнейшем можно использовать в интеграции с существующими конвейерами преобразования и использования данных.

Исторически задача извлечения структурированных представлений решалась на основе правил, позднее – с помощью фреймворков информационного извлечения, а затем – методами глубокого обучения. Развитие крупных языковых моделей (LLM) в последние годы дало новые возможности: сделало возможным прототипирование систем извлечения без необходимости ручного написания правил или дорогостоящей разметки. Тем не менее, наряду с преимуществами, такие подходы показали и недостатки: нестабильность результатов, отсутствие детерминизма, синтаксические ошибками и слабые привязки к формальным схемам валидации (например, JSON Schema или XSD).

Цель работы – систематизировать существующие подходы к извлечению и структурированию данных.

Проведённый анализ показывает как плюсы существующих решений, так и их ключевые ограничения, препятствующие созданию надёжных, воспроизводимых и строго типизированных преобразований. Данный обзор фиксирует эволюцию методов, выявляет исследовательские пробелы и формирует требования к следующему шагу развития – созданию детерминируемых конвейеров структурирования данных, способных объединить универсальность LLM и строгие гарантии валидации.

© Олтян Н.Н., 2025

Предпосылки и определения

Задача преобразования текстовой информации в форматы JSON и XML требует строгого определения используемых понятий. Под структурированием текста в рамках данного обзора понимается процесс извлечения информации из неструктурированных источников и её представления в древовидной форме с заданной схемой и типами данных. Форматы JSON и XML, несмотря на различия в синтаксисе, имеют общую цель: обеспечить возможность формальной валидации разрозненных структур с вложениями.

При обсуждении корректности таких структур данных важно различать два уровня проверки её валидности:

первый уровень — синтаксическая валидность, т.е. соответствие вывода правилам грамматики формата (например, корректное использование скобок, кавычек и тегов);

второй уровень — семантическая валидность, проверяющая принадлежность структуры к готовой схеме.

В случае JSON это может быть JSON Schema, а в случае XML – XML Schema Definition. Схемы задают типы данных и обязательность наличия полей, их допустимые значения, а также позволяют формализовать требования к структуре.

Также в задаче преобразования текста необходим детерминизм выхода, под которым понимается способность системы при одинаковых входных данных и фиксированных условиях давать идентичный результат. Такой детерминизм является необходимым, поскольку именно он позволяет интегрировать конвейеры извлечения в промышленные задачи, где требуется надёжность и повторяемость. Противоположностью этому выступают вероятностные механизмы генерации, характерные для LLM, что приводит к вариативности результата и снижает доверие к системе.

Наряду с этим в практике структурирования данных существенна проблема нормализации значений и структур. Даже при синтаксически и семантически валидных представлениях может существовать множество эквивалентных вариантов сериализации представлений, которые надо упорядочить, а данные должны иметь каноничный вид, например, фиксированный порядок ключей, унифицированные форматы чисел и дат. Все это позволяет устранить неоднозначность систем и повысить их стабильность.

Важным аспектом остаётся также работа с неопределённостью: отсутствием данных, нулевыми значениями и частично извлечёнными структурами, что требует доработки – от отклонения некорректных экземпляров до «ремонта» с помощью правок. В современных исследованиях такие методы позиционируются как дополнительные слои валидации, обеспечивающие согласованность данных.

Таким образом, в рамках данного обзора под корректным структурированием текста понимается процесс, который обеспечивает не только синтаксическую правильность, но и соответствие формальной схеме, а также детерминизм и воспроизводимость вывода.

Классические подходы к структурированию текста

Исторически извлечение структур из текстов развивалось в русле правил-ориентированных технологий: регулярных выражений и шаблонов. В таких системах проектировщик самостоятельно формулировал набор детерминированных правил сопоставления (pattern matching), отмечая сущности, отношения и их роли, после чего полученные результаты сериализовывались в целевые форматы (чаще всего XML). Классические обзоры подчёркивают ключевые достоинства этого класса (высокую точность и объяснимость, управляемость изменений и лёгкость аудита) при условии, что доменная область стабильна и покрыта набором правил. Однако также фиксируются и ограничения: высокая стоимость разработки и сопровождения, ограниченная повторяемость и перенос между доменами и снижение полноты на «длинном хвосте» языковых вариаций. Эти свойства делают правила эффективными для узких задач с чёткими шаблонами, но слабо масштабируемыми при расширении домена [23, 28].

Rule-based EI опирается не только на посимвольные регэкспы, но и на грамматики по аннотациям уровня фраз\токенов. Показательный пример – JAPE (Java Annotation Patterns Engine) в составе GATE (General Architecture for Text Engineering): это конечный алгоритм преобразования аннотаций, где левая часть правила задаёт шаблон соответствий (регулярные выражения над метками и их атрибутами), а правая – действия по созданию новых аннотаций и атрибутов. Такой подход систематизирует правила на

уровне лингвистических объектов (предложения, сущности, токены), что делает их удобными для конструирования доменных извлечений. Вместе с тем они сохраняют все базовые ограничения подхода, основанного на правилах правил – стоимость их подготовки и доменную привязку [24, 33].

Промышленные фреймворки начала 2000-х годов закрепили эту практику в полноценных конвейерах по трансформации текста. GATE предоставил архитектуру и графическую среду (рис. 1) для сборки NLP приложений из взаимозаменяемых компонентов: токенизаторы, правила JAPE, нормализаторы, а также инфраструктуру аннотирования и экспорта результатов в XML. За счёт повторного использования и богатой экосистемы плагины GATE позволяли быстро собирать прикладные IE-системы для конкретных доменов (вплоть до специализированных отраслей, например, биомедицины). Сильные стороны GATE – зрелость, воспроизводимость и транспарентность пайплайнов; слабые стороны – необходимость значимой ручной работы по написанию и поддержке правил, ограниченная переносимость между доменами и зависимость качества от экспертизы разработчика [7].

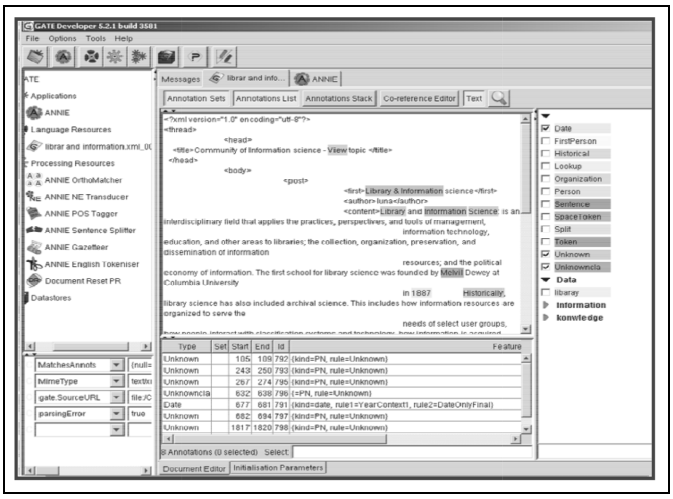

Рис. 1. Концепт аннотации в GATE
Fig. 1. GATE annotation concept

Одновременно с этим была предложена архитектура UIMA (Unstructured Information Management Architecture) – стандартизированная архитектура для разработки и развёртывания конвейеров обработки неструктурированных данных. Базовые элементы UIMA – аннотаторы, общее хранилище аннотаций и дескрипторы компонентов/потоков. Такая модульность и формализация метаданных упростили компоновку, повторное использование и переносимость IE-компонент между проектами. На уровне результатов UIMA поощряла сериализацию в XML/stand-off аннотации и строгую спецификацию интерфейсов между стадиями. Как и в случае GATE, архитектурные гарантии UIMA не решают автоматически проблему доменной масштабируемости: стоимость создания правил, покрытие выбросов и устойчивость к вариативности языка остаются создателем правил внутри конвейера [10, 11].

Прикладные исследования демонстрируют, что сочетание регэкспов/шаблонов и грамматик по аннотациям может дать воспроизводимые результаты в узких областях, где терминология стабильна, а структуры для извлечения хорошо формализованы. В таких сценариях процесс типичен: анализ небольшой обучающей выборки, построение доменных правил, оценка на новых документах и экспорт результатов. Это подтверждает тезис о высокой точности и объяснимости правил при узкоспециализированной настройке и одновременно иллюстрирует трудозатраты и риски деградации при переносе в новый домен [22].

Таким образом, классические подходы закладывают фундаментальные принципы, важные и сегодня [23, 28]:

детерминированность и трассируемость решений;

контракты структуры через XML/схемы;

явная зона ответственности между стадиями конвейера.

Однако их слабые места – стоимость создания/сопровождения, ограниченная переносимость и чувствительность к языковой вариативности – послужили основной мотивацией перехода к обучаемым моделям.

Нейросетевые методы до LLM

Переход от правил-ориентированных систем к обучаемым моделям начался с последовательной разметки на уровне токенов: BiLSTM-CRF дал способ совместно моделировать контекст и последовательные зависимости меток [15]. Классические работы показали, что двунаправленные LSTM (рис. 2) с верхним CRF обеспечивают устойчивый прирост качества для NER без ручных признаков [18]; ключевые вари-

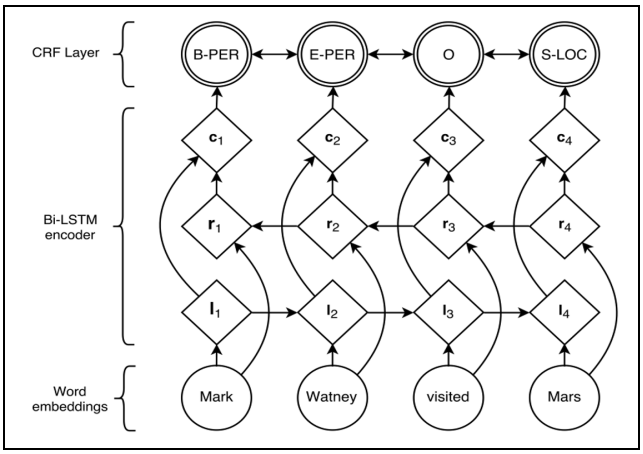

Рис. 2. Основная архитектура двунаправленной LSTM сети
Fig. 2. The basic architecture of a bidirectional LSTM network

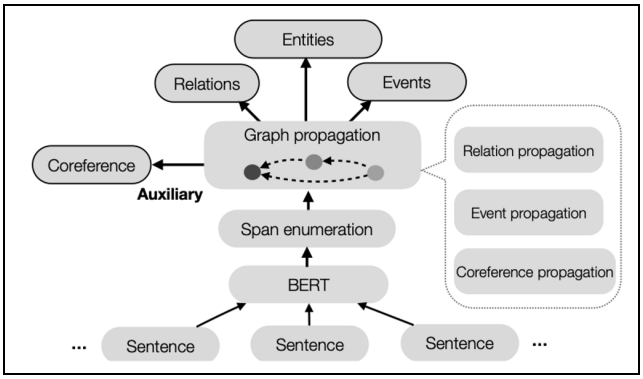

Рис. 3. Представление идеи фреймворка: DYGIE++. Общие представления span создаются путем уточнения контекстуализированных вставок слов с помощью обновлений графа span, а затем передаются в функции оценки для трех задач IE (Relations, Entities, Events)
Fig. 3. Overview of the framework idea: DYGIE++. Common span representations are created by refining contextualized word inserts using updates to the span graph, and then passed to the evaluation functions for the three IE tasks (Relations, Entities, Events)

анты включают в себя BiLSTM-CRF с символьными признаками и/или CNN на уровне символов (для морфологии) и совместную оптимизацию всей цепочки [6, 21]. Эти результаты были тщательно продемонстрированы в работах, которые задали «де-факто» базовую архитектуру для до-LLM эпохи [6, 15, 18, 21].

Дальнейшее развитие привело к объединению подзадач IE и работе за пределами предложения. Span-based и графовые архитектуры строят кандидаты-спаны и совместно предсказывают сущности, отношения и события, распространяя информацию по графу. Репрезентативен DyGIE++: поверх контекстуализированных представлений (BERT) формируются спаны (рис. 3) и выполняется передача сообщения (message passing), что улучшает извлечение, особенно в специализированных доменах и при межфразовой связности. Исследование демонстрирует прирост качества на наборах для сущностей, отношений и событий и подчёркивает пользу совместного обучения [27].

На уровне глобальной согласованности решений показателен OneIE: модель формирует единый граф предсказаний (entities, relations, event triggers/arguments) и применяет beam-декодирование с глобальными признаками, выбирая согласованный результат, а не набор локальных решений (рис. 4). Это снижает каскадные ошибки и обеспечивает прирост метрик в сравнении с поузловыми классификаторами [19].

Эти методы задали стандарт качества и переносимости внутри домена, уменьшив зависимость от правил и ручных признаков. Их сильные стороны:

высокое качество на уровне токенов (BiLSTM-CRF) и в RE (CNN/PCNN);

совместное моделирование структур (DyGIE++, OneIE).

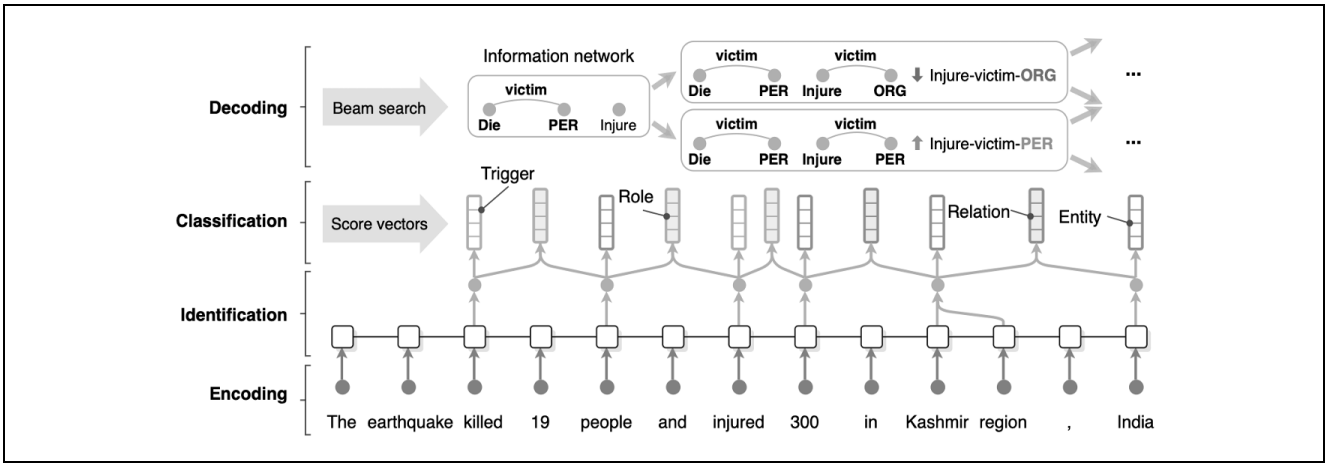

Рис. 4. Представление комплексной платформы совместного извлечения информации OneIE
Fig. 4. An illustration of the integrated OneIE information sharing platform

Однако ограничения остаются существенными:

высокая потребность в размеченных данных;

стоимость обучения и адаптации;

отсутствие жёстких гарантий соответствия схемам JSON/XSD и детерминизма вывода на уровне конвейера.

Это мотивирует переход к LLM-ориентированным методам и к интеграции с механизмами grammar-/schema-constrained decoding и пост-валидацией [9, 32].

LLM для структурированного извлечения информации

Появление LLM изменило парадигму информационного извлечения. С развитием генеративных моделей стало возможным формулировать задачи извлечения единым запросом (рис. 5) [20].

Одним из первых шагов в этом направлении стала модель UIE (Unified Information Extraction), которая предложила рассматривать извлечение как задачу генерации структур по схеме, представленной на рис. 6, унифицировав NER, отношения и события в общий формат. Эксперименты на множестве датасетов показали, что генеративная постановка позволяет достигать высокой переносимости между задачами и доменами (supervised, low-resource, few-shot), а это заложило основу для дальнейших исследований [20].

Следующим этапом развития стали методы инструкционного извлечения, где спецификация задачи формулируется в человеко-читаемом виде. Так, модель InstructUIE показала, что LLM можно дообучать на множестве IE-задач с инструкциями и примерами, покрывающими более 30 разнородных датасетов [29]. Такой подход позволил задавать новые схемы извлечения без отдельного обучения моделей под каждую из задач, обеспечивая гибкость и универсальность (рис. 7). Развитие этой идеи привело к парадигме ODIE (On-Demand Information Extraction), где пользователь в запросе формулирует как целевую схему, так и формат представления (например, таблицу с заданными заголовками) [16]. ODIE демонстрирует устойчивые результаты при adhoc-извлечении, что особенно важно для прикладных сценариев, требующих детерминированности.

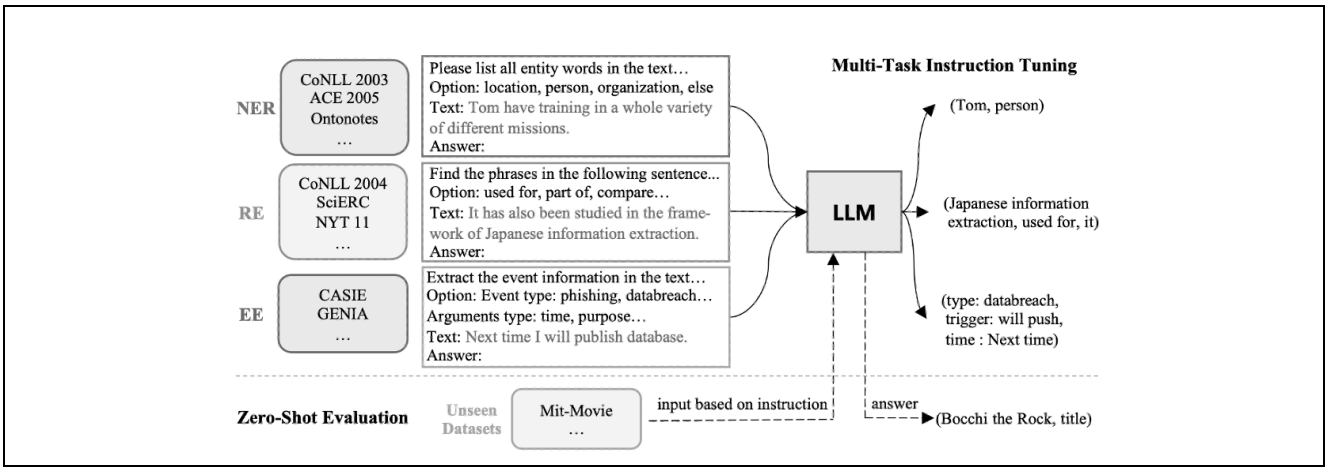

Рис. 5. Переход от специализированных IE для различных задач, структур, схем к универсальному IE через генерацию структуры
Fig. 5. Transition from specialized IE for various tasks, structures, schemes universal IE through structure generation

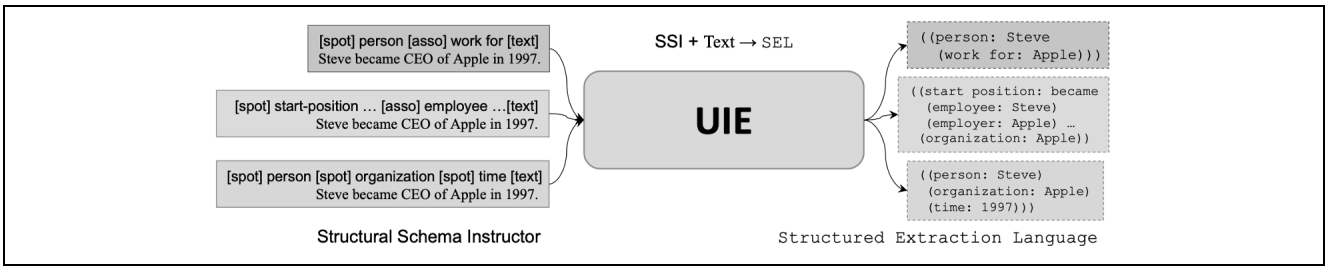

Рис. 6. Общая структура UIE
Fig. 6. The general structure of the UIE

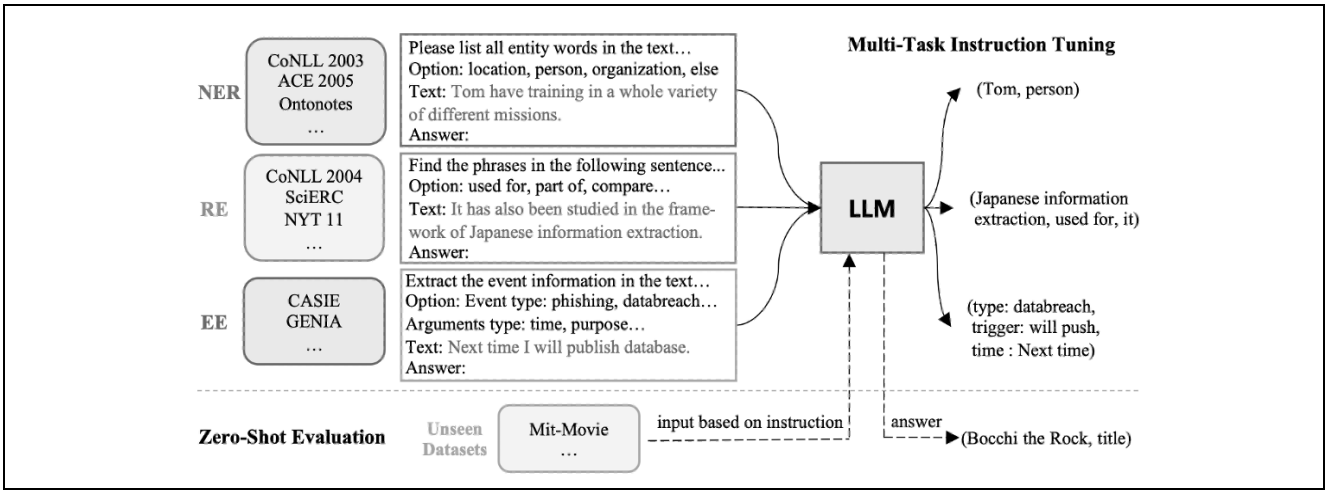

Рис. 7. Обзорная структура InstructUIE (входные данные состоят из инструкций по выполнению задач, опций и текста, на выходе получается более понятное предложение, преобразованное из исходных структур меток)
Fig. 7. The overview structure of InstructUIE. The input data consists of instructions for completing tasks, options, and text. The output is a more understandable sentence transformed from the original label structures

Применимость этих подходов подтверждается исследованиями в прикладных областях. Так, эксперименты показали, что LLM способны извлекать структурированные факты из научных публикаций по материаловедению, формируя базы знаний о составе и свойствах материалов, что открыло путь к ускорению открытий в этой области [8]. В области химии успешно исследовали возможность использования LLM для выделения ключевых элементов реакций (реагентов, катализаторов, условий) в zero-shot режиме [25], а также продемонстрировали, что при дообучении модели способны приводить данные к строго структурированному виду [3, 25]. В медицине исследования показали применимость LLM, таких как GPT-4, для выполнения извлечение клинических сущностей и признаков из медицинских записей, достигая конкурентного качества в zero-shot и few-shot режимах. Однако эти работы фиксируют также серьёзные ограничения: чувствительность к формулировке инструкций, нестабильность вывода и необходимость пост-валидации для использования в настолько регламентированных и критичных к ошибкам средах [2, 14].

Обобщающие обзоры последних лет систематизируют результаты и подчёркивают двойственность применения LLM в IE [30]. С одной стороны, универсальность и возможность быстро формулировать новые запросы для извлечения без размеченных данных делают LLM мощным инструментом для прототипирования, а с другой – всё так же сохраняются фундаментальные проблемы: отсутствие гарантий синтаксической и семантической валидности, вариативность вывода, несоответствие схемам и слабая типизация [30].

Таким образом, LLM позволили переосмыслить информационное извлечение в задачу генерации с инструкциями, чья применимость подтверждается как академическими экспериментами, так и доменными примерами. Однако их внедрение в производственные конвейеры ограничивается отсутствием строгого контроля формата и детерминизма вывода. Это обстоятельство напрямую подводит к необходимости изучения методов grammar-constrained decoding, schema-guided prompting и пост-валидации, которые будут рассмотрены далее.

Гарантии формата: Grammar-Constrained и Schema-Guided Decoding

Одним из главных ограничений применения LLM для информационного извлечения остаётся отсутствие формальных гарантий корректности вывода. Даже при точной настройке подсказок модели склонны производить синтаксически некорректный JSON или XML, нарушать схему или выдавать ответы, отличные от ожидаемой структуры. Для преодоления этих проблем в последние годы активно развиваются методы ограниченного декодирования (grammar-constrained decoding) и схемо-ориентированного вывода (schema-guided prompting/decoding).

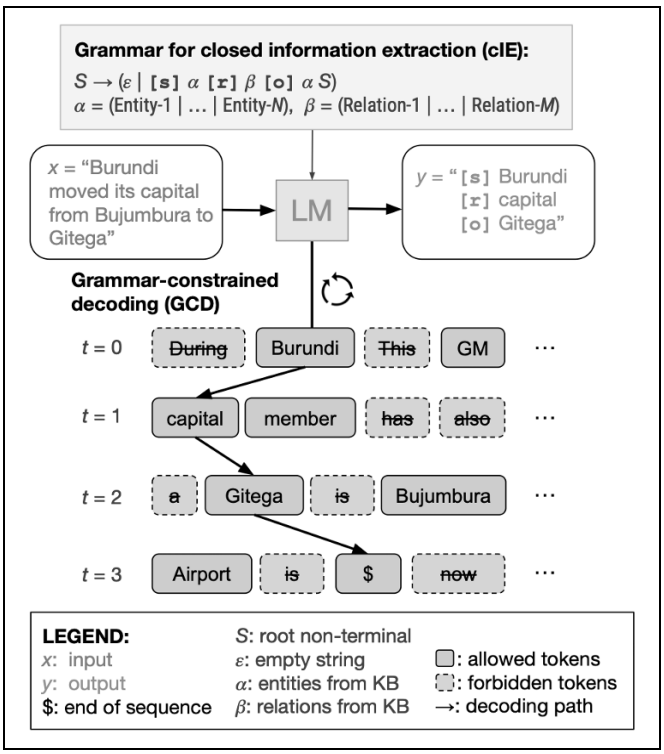

Рис. 8. Декодирование с грамматическими ограничениями (GCD), применяемое к задаче извлечения информации, целью которой является извлечение списка y триплетов «субъект-отношение-объект» из входного текста x

Fig. 8. Grammar-constrained decoding (GCD), applied to an information extraction task, the purpose of which is to extract a list y of subject-relation-object triplets from the input text x

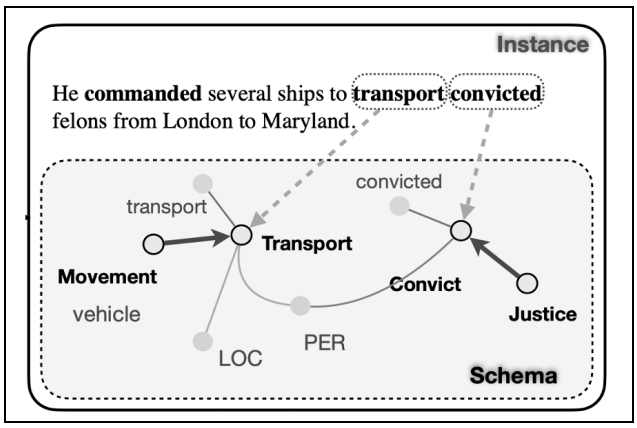

Рис. 9. Ссылка на схему в виде подсказки: создается гибридное хранилище ссылок на экземпляры схемы, из которого извлекаются связанные знания в виде промптов для дата-эффективного обучения

Fig. 9. A link to the diagram in the form of a hint. A hybrid repository of links to schema instances is being created, from which related knowledge is extracted in the form of prompt for data-efficient learning

В подходе grammar-constrained decoding выход модели разрешается только в пределах формальной грамматики целевого языка вывода (рис. 8). Грамматика и инкрементальный парсер накладываются поверх LLM, демонстрируя, что такая декодировка обеспечивает грамматически корректные структуры без дообучения модели. Это делает поведение генерации предсказуемым на уровне синтаксиса [12].

Что касается формальной спецификации заданной схемы данных, то ограничения могут быть добавлены на уровне подсказок. В работе Schema-aware Reference as Prompt извлекаются «ссылочные» элементы схемы и помещаются в подсказку как RAP-контекст (рис. 9). Это улучшает эффективность извлечения структурированных фактов и построения знаний в доменах с маленьким количеством данных, хотя не предоставляет жёстких гарантий формального соответствия на уровне валидатора [31].

В совокупности эти результаты показывают следующее: грамматические и схемо-ориентированные ограничения делают вывод синтаксически и структурно корректным и пригодным для формальной проверки. При этом вопросы более «смысловой» (доменно-семантической) согласованности лежат вне непосредственного фокуса указанных методов и требуют дополнительных слоёв в конвейере.

Обсуждение и проблемы в исследованиях

Даже при ограниченном декодировании LLM остаются случаи, когда итоговые JSON/XML-экземпляры синтаксически- или схемонекорректны, поэтому на «выходе» конвейера применяются валидаторы (рис. 10) и алгоритмы минимального ремонта. Для JSON-семейства ключевая практика – проверка экземпляра по JSON Schema с последующим итеративным исправлением нарушений (обязательные поля и типы). Современные валидаторы, такие как Blaze, компилируют схему во внутреннее представление и на порядок ускоряют многократные проверки, делая циклы «проверка → правка → повторная проверка» приемлемыми в продакшене [26]. Авторы также показывают, что некоторые популярные валидаторы дают некорректные результаты на части тестов, тогда как Blaze сохраняет строгое соответствие спецификации. Это важно для доверенного пост-контроля перед записью в хранилище.

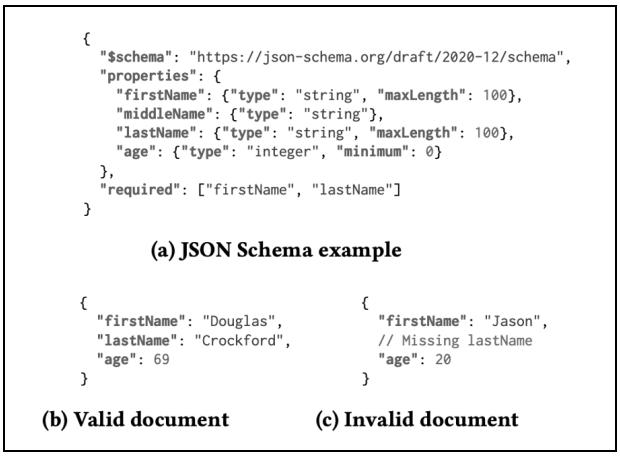

Рис. 10. Пример быстрой проверки по схеме JSON
Fig. 10. An example of a quick check according to the scheme JSON

При этом существуют и другие методы оценки работы LLM модели и получившейся схемы. Так, рассматриваются гибридные методы оценки качества генерации LLM, где сочетаются экспертная разметка, автоматические метрики (BLEU, ROUGE, BERT-Score) и новые подходы LLM-as-a-Judge (LLM как судья). Идеи гибридной оценки напрямую применимы к задаче верификации качества извлечённых структур. В частности, предложенная логика «LLM как судьи» дает дополнительный уровень семантической валидации JSON/XML-структур: если классические валидаторы проверяют только синтаксис и типы, то LLM-оценщик способен анализировать содержательную точность и соответствие исходному тексту. Это создает предпосылки для автоматизированных схем пост-валидации, где роль эксперта-человека частично выполняет модель, калиброванная по эталонной разметке [1].

Ремонт обычно начинается с устранения синтаксических ошибок (пропущенные запятые/скобки, неверные кавычки) и только затем переходит к нарушениям схемы. Для первого шага применяются утилиты «JSON repair», которые автоматически восстанавливают well-formed JSON. Они известны именно как средства пост-обработки LLM-выводов и устраняют распространённые синтаксические ошибки:

отсутствие кавычек вокруг ключей;

одиночные кавычки;

запятые на конце;

неправильные булевы значения [34].

Далее вступают методы минимального структурного ремонта относительно схемы: классический подход для полуструктурированных данных формулирует правки как операции над деревом (вставка, удаление, замена) с минимизацией стоимости. Для XML эта постановка исследована на уровне исправления документов к заданному DTD/схеме, где предлагаются алгоритмы поиска ближайшего «правильного» дерева. Эти работы задают общий каркас для схемно-осмысленного ремонта: исправлять не «как угодно», а ровно настолько, чтобы удовлетворить формальным ограничениям [4, 17].

При этом современный формальный анализ JSON Schema показывает, что с усложнением «современных» конструкций (динамические ссылки и др.) проверка и, следовательно, навигация по пространству правок становятся нетривиальными. Однако при фиксированной схеме проверка остаётся полиномиальной по данным. Это уточняет границы вычислимости для практик авто-ремонта «поверх» валидатора [5].

Наконец, специализированные бенчмарки устойчивости к схемам (JSONSchemaBench) демонстрируют, что даже при ограниченном декодировании часть моделей нарушает типы и пропускает обязательные поля. Это эмпирически подтверждает необходимость пост-валидации и ремонта в конвейере структурирования [13]. Иными словами, auto-repair – не замена ограниченному декодированию, а обязательный завершающий слой, повышающий долю валидных экземпляров без повторного запуска генерации.

Сравнительный анализ подходов

В обзоре рассмотрена эволюция методов структурирования данных: от правил и фреймворков к нейросетевым моделям, LLM и методам ограниченного декодирования с пост-валидацией. Показано, что каждый класс решений решает лишь часть задачи:

правила обеспечивают контроль и детерминизм, но не масштабируются;

нейросети повышают качество, но требуют больших ресурсов и не гарантируют строгой валидации;

LLM универсальны и гибки, но их вывод нестабилен и плохо воспроизводим;

ограничения и авто-ремонт улучшают синтаксис и схемное соответствие, но не устраняют риски семантической несогласованности.

После сведения воедино рассмотренных поколений методов и оценки их сильных и слабых сторон сравнительный анализ подтверждает, что ни один из подходов не обеспечивает сразу всех требований: синтаксической и семантической валидности, строгой типизации и воспроизводимости при разумной стоимости (таблица).

Таблица. ***Сравнительный анализ подходов к структурированию данных***

Подход	Гарантии синтаксиса	Гарантии типов/схемы	Воспроизводимость	Стоимость внедрения	Типичные сбои
Правила и шаблоны	Полные	Частично	Высокая	Высокая	Низкая полнота, чувствительность
Фреймворки (GATE, UIMA)	Полные	Частично	Высокая	Средняя	Ограниченная переносимость
Нейросети до LLM	Нет	Ограниченные	Средняя	Высокая	Ошибки структуры и домена
LLM (UIE, InstructUIE)	Нет	Слабые	Низкая	Низкая (прототип), высокая (продакшн)	Форматные ошибки, пропуски
Constrained decoding	Полные	Частично	Средняя	Средняя	Ошибки типов и значений
Пост-валидация и ремонт	После коррекции	Полные	Средняя	Средняя/Высокая	Риск искажения смысла

Правила и фреймворки дают жёсткий контроль, но не масштабируются. Нейросети и LLM универсальны, но не гарантируют корректность. Ограниченное декодирование и пост-валидация решают отдельные слои проблемы (синтаксис, частично – типы), однако в отрыве от конвейера не обеспечивают полного решения. Таким образом, необходим новый уровень – детерминируемые конвейеры структурирования данных, объединяющие сильные стороны разных подходов и минимизирующие их слабости.

Ключевые ограничения текущих подходов

Проведённый обзор показывает, что развитие методов структурирования текста прошло несколько этапов: от правил и фреймворков к нейросетевым моделям, далее – к LLM и методам ограниченного декодирования с пост-валидацией. Каждый из классов решений внёс вклад в повышение качества и универсальности извлечения, однако ни один из них не обеспечивает всех ключевых требований одновременно:

гарантированной синтаксической и семантической валидности;

строгой типизации;

детерминизма;

воспроизводимости.

Правила и фреймворки задали основу детерминизма и контроля формата, но плохо масштабируются и требуют значительных ручных усилий. Нейросетевые модели показали рост качества и способность к обобщению, но остались зависимыми от больших размеченных корпусов и не решают задачу формальной валидации. LLM обеспечили универсальность и гибкость постановки задач, открыв возможности быстрого прототипирования и применения в реальных доменах (медицина, химия, материаловедение), однако их вывод нестабилен и подвержен вариативности. Методы grammar-constrained и schema-guided decoding решают задачу синтаксиса и, частично, схемы, но не обеспечивают контроль смысловой согласованности. Пост-валидация и авто-ремонт структур повышают долю валидных экземпляров, однако рискуют искажать исходное содержание и увеличивают вычислительные затраты.

Ключевыми пробелами исследований являются:

отсутствие целостного конвейера – существующие работы решают лишь отдельные аспекты задачи, не формируя end-to-end решение;

недостаток метрик на уровне конвейера – большая часть исследований ограничивается оценкой точности IE, в то время как метрики воспроизводимости, устойчивости и полноты соответствия схемам остаются нерегламентированными;

семантическая валидность – пока не существует методов, которые бы систематически проверяли и гарантировали доменно-специфические ограничения поверх формальных схем;

интеграция с индустриальными требованиями – практические сценарии требуют не только точности, но и строгого контроля типов, версионирования схем, трассируемости и формальной сертифицируемости.

Таким образом, дальнейшее развитие исследований должно быть направлено на проектирование детерминируемых конвейеров структурирования данных, которые объединят сильные стороны LLM (универсальность, способность к генерации структур в новых доменах) с формальными механизмами контроля (ограничения по грамматике и схеме, валидация, ремонт), дополнив их метриками качества на уровне всего процесса. Именно такой подход можно рассматривать как следующий эволюционный шаг исследований в данной области.

Заключение

В работе рассмотрена эволюция методов структурирования данных: от правил и фреймворков к нейросетевым моделям, LLM и методам ограниченного декодирования с пост-валидацией. Показано, что каждый класс решений решает лишь часть задачи:

правила обеспечивают контроль и детерминизм, но не масштабируются;

нейросети повышают качество, но требуют больших ресурсов и не гарантируют строгой валидации;

LLM универсальны и гибки, но их вывод нестабилен и плохо воспроизводим;

ограничения и авто-ремонт улучшают синтаксис и схемное соответствие, но не устраняют риски семантической несогласованности.

Таким образом, в современном состоянии исследований отсутствует единый подход, который одновременно обеспечивал бы строгую типизацию, гарантированную валидность и детерминизм вывода. Следующим шагом развития исследований должны стать детерминируемые конвейеры структурирования данных, интегрирующие сильные стороны LLM с формальными механизмами контроля и воспроизводимости. Именно такая комбинация открывает путь к надёжному и промышленно применимому решению задачи.

Список источников

1. *Тетеревенков Д.Л.* Экспертно-ориентированные методы оценки качества текстовой генерации больших языковых моделей // Мягкие измерения и вычисления. 2025. № 5. Т. 90. С. 30–37; https://doi.org/10.36871/26189976.2025.05.003
2. *Adam H., Lin J., Keenan H., Wilson A. & Ghassemi M*. Clinical Information Extraction with Large Language Models: A Case Study on Organ Procurement. AMIA Annu Symp Proc (2025): 115–123.
3. *Ai, Qianxiang, Meng, Fanwang, Shi, Jiale, Pelkie, Brenden, & Coley, Connor W*. Extracting structured data from organic synthesis procedures using a fine-tuned large language model. Digital Discovery, 3(9) (2024): 1822–1831.
4. *Amavi, Joshua, Bouchou, Béatrice, & Savary, Agata*. On correcting XML documents with respect to a schema. The Computer Journal, 57(5) (2014): 639–674.
5. *Attouche, Lyes, Baazizi, Mohamed-Amine, Colazzo, Dario, Ghelli, Giorgio, Sartiani, Carlo, & Scherzinger, Stefanie*. Validation of modern JSON schema: formalization and complexity. Proceedings of the ACM on Programming Languages, 8(POPL) (2024): 1451–1481.
6. *Chiu, Jason P.C., & Nichols, Eric*. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4 (2016): 357–370.
7. *Cunningham, Hamish, Maynard, Diana, & Bontcheva, Kalina*. Text Processing with GATE. Gateway Press CA, 2011.
8. *Dagdelen, John, Dunn, Alexander, Lee, Sanghoon, Walker, Nicholas, Rosen, Andrew S., Ceder, Gerbrand, Persson, Kristin A., & Jain, Anubhav*. Structured information extraction from scientific text with large language models. Nature Communications, 15(1) (2024): 1418.
9. *Delaunay, Julien, Tran, Hanh Thi Hong, González-Gallardo, Carlos-Emiliano, Bordea, Georgeta, Sidere, Nicolas, & Doucet, Antoine*. A comprehensive survey of document-level relation extraction (2016–2023). arXiv preprint arXiv:2309.16396 (2023).
10. *Ferrucci, David, & Lally, Adam*. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering, 10(3–4) (2004): 327–348.
11. *Ferrucci, David, & Lally, Adam*. Building an example application with the Unstructured Information Management Architecture. IBM Systems Journal, 43(3) (2004): 455–475.

12. *Geng, Saibo, Cooper, Hudson, Moskal, Michał, Jenkins, Samuel, Berman, Julian, Ranchin, Nathan, West, Robert, Horvitz, Eric, & Nori, Harsha*. Generating structured outputs from language models: benchmark and studies. arXiv preprint arXiv:2501 (2025).
13. *Geng, Saibo, Josifoski, Martin, Peyrard, Maxime, & West, Robert*. Grammar-constrained decoding for structured NLP tasks without finetuning. arXiv preprint arXiv:2305.13971 (2023).
14. *Hu, Yan, Chen, Qingyu, Du, Jingcheng, Peng, Xueqing, Keloth, Vipina Kuttichi, Zuo, Xu, Zhou, Yujia* et al. Improving large language models for clinical named entity recognition via prompt engineering. Journal of the American Medical Informatics Association, 31(9) (2024): 1812–1820.
15. *Huang, Zhiheng, Xu, Wei, & Yu, Kai*. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015).
16. *Jiao, Yizhu, Zhong, Ming, Li, Sha, Zhao, Ruining, Ouyang, Siru, Ji, Heng, & Han, Jiawei*. Instruct and extract: instruction tuning for on-demand information extraction. arXiv preprint arXiv:2310.16040 (2023).
17. *Korn, Flip, Saha, Barna, Srivastava, Divesh, & Ying, Shanshan*. On repairing structural problems in semi-structured data. Proceedings of the VLDB Endowment, 6(9) (2013): 601–612.
18. *Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, & Dyer, Chris*. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016).
19. *Lin, Ying, Ji, Heng, Huang, Fei, & Wu, Lingfei*. A joint neural model for information extraction with global features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7999–8009 (2020).
20. *Lu, Yaojie, Liu, Qing, Dai, Dai, Xiao, Xinyan, Lin, Hongyu, Han, Xianpei, Sun, Le, & Wu, Hua*. Unified structure generation for universal information extraction. arXiv preprint arXiv:2203.12277 (2022).
21. *Ma, Xuezhe, & Hovy, Eduard*. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354 (2016).
22. *Rosier, Arnaud, Burgun, Anita, & Mabo, Philippe*. Using regular expressions to extract information on pacemaker implantation procedures from clinical reports. AMIA Annual Symposium Proceedings, vol. 2008, p. 81 (2008).
23. *Sarawagi, Sunita*. Information extraction. Foundations and Trends in Databases, 1(3) (2008): 261–377.
24. *Sawsaa, Ahlam, & Lu, Joan*. Extracting information science concepts based on JAPE regular expression. In Proceedings of the International Conference on Internet Computing (ICOMP), p. 1. The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), (2011).
25. *Vangala, S. R., Krishnan, S. R., & Bung, N*. et al. Suitability of large language models for extraction of high-quality chemical reaction dataset from patent literature. J Cheminform, 16, 131 (2024).
26. *Viotti, Juan Cruz, & Mior, Michael J*. Blaze: compiling JSON Schema for 10x faster validation. arXiv preprint arXiv:2503.02770 (2025).
27. *Wadden, David, Wennberg, Ulme, Luan, Yi, & Hajishirzi, Hannaneh*. Entity, relation, and event extraction with contextualized span representations. arXiv preprint arXiv:1909.03546 (2019).
28. *Waltl, Bernhard, Bonczek, Georg, & Matthes, Florian*. Rule-based information extraction: advantages, limitations, and perspectives. Jusletter IT, 4 (2018).
29. *Wang, Xiao, Zhou, Weikang, Zu, Can, Xia, Han, Chen, Tianze, Zhang, Yuansen, Zheng, Rui*, et al. Instructuie: multi-task instruction tuning for unified information extraction. arXiv preprint arXiv:2304.08085 (2023).
30. *Xu, Derong, Chen, Wei, Peng, Wenjun, Zhang, Chao, Xu, Tong, Zhao, Xiangyu, Wu, Xian, Zheng, Yefeng, Wang, Yang, & Chen, Enhong*. Large language models for generative information extraction: a survey. Frontiers of Computer Science, 18(6) (2024).
31. *Yao, Yunzhi, Mao, Shengyu, Zhang, Ningyu, Chen, Xiang, Deng, Shumin, Chen, Xi, & Chen, Huajun*. Schema-aware reference as prompt improves data-efficient knowledge graph construction. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 911–921 (2023).
32. *Zhao, Xiaoyan, Deng, Yang, Yang, Min, Wang, Lingzhi, Zhang, Rui, Cheng, Hong, Lam, Wai, Shen, Ying, & Xu, Ruifeng*. A comprehensive survey on relation extraction: recent advances and new frontiers. ACM Computing Surveys, 56(11) (2024): 1–39.
33. GATE Documentation. JAPE: Regular Expressions over Annotations [Электронный ресурс]. Офиц. сайт GATE. Режим доступа: https://gate.ac.uk/releases/gate-5.0-build3244-ALL/doc/tao/splitch7.html (дата обращения: 16.10.2025).
34. JSON Repair Tool. Online utility for fixing malformed JSON (well-formedness repair) [Электронный ресурс]. Режим доступа: https://jsonrepair.com (дата обращения: 16.10.2025).

Информация об авторе

Никита Николаевич Олтян – аспирант
SPIN-код: 9730-0962

Статья поступила в редакцию 10.10.2025
Одобрена после рецензирования 22.10.2025
Принята к публикации 30.10.2025

Original article

Evolution of methods for extracting and structuring data from text to JSON and XML

N.N. Oltyan[1]
[1] Financial university under the Government of the Russian Federation (Moscow, Russia)
[1] nikitaoltyan@mail.ru

Abstract

In modern data analysis and machine learning, a common problem is the need to transform textual information into semi-structured formats like JSON and XML. Existing approaches, including those based on rules, deep learning, and Large Language Models (LLMs), have significant drawbacks, such as high maintenance costs, the need for large amounts of labeled data, and unstable output.
The goal is to systematize data structuring approaches, including post-validation algorithms, and to develop a concept for a deterministic pipeline to ensure both syntactic and semantic validity.
An analysis of modern data structuring methods was conducted, which showed that none of them simultaneously provide strict typing, deterministic output, and syntactic and semantic validity. Key limitations of existing solutions were identified. An approach based on using deterministic pipelines is proposed, which combines the strengths of LLMs with formal validation mechanisms to achieve reliable data structuring.
The developed approach helps to overcome the limitations of existing methods and to improve the reliability and quality of structuring textual data. This will facilitate the development of data analysis systems that require precise and predictable transformation of unstructured information.

Keywords

Data analysis, information extraction, data structuring, large language models (LLM), data validation, JSON, XML

For citation

Oltyan N.N. Evolution of methods for extracting and structuring data from text to JSON and XML. Neurocomputers. 2025. V. 27. № 6. P. 37–49. DOI: 10.18127/j19997493-202506-04 (in Russian).

References

1. *Teterevenkov D.L.* E`kspertno-orientirovanny`e metody` ocenki kachestva tekstovoj generacii bol`shix yazy`kovy`x modelej. Myagkie izmereniya i vy`chisleniya. 2025. № 5. T. 90. S. 30–37. https://doi.org/10.36871/26189976.2025.05.003
2. *Adam H., Lin J., Keenan H., Wilson A. & Ghassemi M.* Clinical Information Extraction with Large Language Models: A Case Study on Organ Procurement. AMIA Annu Symp Proc (2025): 115–123.
3. *Ai, Qianxiang, Meng, Fanwang, Shi, Jiale, Pelkie, Brenden, & Coley, Connor W.* Extracting structured data from organic synthesis procedures using a fine-tuned large language model. Digital Discovery, 3(9) (2024): 1822–1831.
4. *Amavi, Joshua, Bouchou, Béatrice, & Savary, Agata*. On correcting XML documents with respect to a schema. The Computer Journal, 57(5) (2014): 639–674.
5. *Attouche, Lyes, Baazizi, Mohamed-Amine, Colazzo, Dario, Ghelli, Giorgio, Sartiani, Carlo, & Scherzinger, Stefanie*. Validation of modern JSON schema: formalization and complexity. Proceedings of the ACM on Programming Languages, 8(POPL) (2024): 1451–1481.
6. *Chiu, Jason P.C., & Nichols, Eric*. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4 (2016): 357–370.
7. *Cunningham, Hamish, Maynard, Diana, & Bontcheva, Kalina*. Text Processing with GATE. Gateway Press CA, 2011.
8. *Dagdelen, John, Dunn, Alexander, Lee, Sanghoon, Walker, Nicholas, Rosen, Andrew S., Ceder, Gerbrand, Persson, Kristin A., & Jain, Anubhav*. Structured information extraction from scientific text with large language models. Nature Communications, 15(1) (2024): 1418.
9. *Delaunay, Julien, Tran, Hanh Thi Hong, González-Gallardo, Carlos-Emiliano, Bordea, Georgeta, Sidere, Nicolas, & Doucet, Antoine*. A comprehensive survey of document-level relation extraction (2016–2023). arXiv preprint arXiv:2309.16396 (2023).
10. *Ferrucci, David, & Lally, Adam*. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering, 10(3–4) (2004): 327–348.
11. *Ferrucci, David, & Lally, Adam*. Building an example application with the Unstructured Information Management Architecture. IBM Systems Journal, 43(3) (2004): 455–475.
12. *Geng, Saibo, Cooper, Hudson, Moskal, Michał, Jenkins, Samuel, Berman, Julian, Ranchin, Nathan, West, Robert, Horvitz, Eric, & Nori, Harsha*. Generating structured outputs from language models: benchmark and studies. arXiv preprint arXiv:2501 (2025).
13. *Geng, Saibo, Josifoski, Martin, Peyrard, Maxime, & West, Robert*. Grammar-constrained decoding for structured NLP tasks without finetuning. arXiv preprint arXiv:2305.13971 (2023).
14. *Hu, Yan, Chen, Qingyu, Du, Jingcheng, Peng, Xueqing, Keloth, Vipina Kuttichi, Zuo, Xu, Zhou, Yujia* et al. Improving large language models for clinical named entity recognition via prompt engineering. Journal of the American Medical Informatics Association, 31(9) (2024): 1812–1820.
15. *Huang, Zhiheng, Xu, Wei, & Yu, Kai*. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015).
16. *Jiao, Yizhu, Zhong, Ming, Li, Sha, Zhao, Ruining, Ouyang, Siru, Ji, Heng, & Han, Jiawei*. Instruct and extract: instruction tuning for on-demand information extraction. arXiv preprint arXiv:2310.16040 (2023).
17. *Korn, Flip, Saha, Barna, Srivastava, Divesh, & Ying, Shanshan*. On repairing structural problems in semi-structured data. Proceedings of the VLDB Endowment, 6(9) (2013): 601–612.
18. *Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, & Dyer, Chris*. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016).

19. *Lin, Ying, Ji, Heng, Huang, Fei, & Wu, Lingfei*. A joint neural model for information extraction with global features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7999–8009 (2020).
20. *Lu, Yaojie, Liu, Qing, Dai, Dai, Xiao, Xinyan, Lin, Hongyu, Han, Xianpei, Sun, Le, & Wu, Hua*. Unified structure generation for universal information extraction. arXiv preprint arXiv:2203.12277 (2022).
21. *Ma, Xuezhe, & Hovy, Eduard*. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354 (2016).
22. *Rosier, Arnaud, Burgun, Anita, & Mabo, Philippe*. Using regular expressions to extract information on pacemaker implantation procedures from clinical reports. AMIA Annual Symposium Proceedings, vol. 2008, p. 81 (2008).
23. *Sarawagi, Sunita*. Information extraction. Foundations and Trends in Databases, 1(3) (2008): 261–377.
24. *Sawsaa, Ahlam, & Lu, Joan*. Extracting information science concepts based on JAPE regular expression. In Proceedings of the International Conference on Internet Computing (ICOMP), p. 1. The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), (2011).
25. *Vangala, S. R., Krishnan, S. R., & Bung, N.* et al. Suitability of large language models for extraction of high-quality chemical reaction dataset from patent literature. J Cheminform, 16, 131 (2024).
26. *Viotti, Juan Cruz, & Mior, Michael J*. Blaze: compiling JSON Schema for 10x faster validation. arXiv preprint arXiv:2503.02770 (2025).
27. *Wadden, David, Wennberg, Ulme, Luan, Yi, & Hajishirzi, Hannaneh*. Entity, relation, and event extraction with contextualized span representations. arXiv preprint arXiv:1909.03546 (2019).
28. *Waltl, Bernhard, Bonczek, Georg, & Matthes, Florian*. Rule-based information extraction: advantages, limitations, and perspectives. Jusletter IT, 4 (2018).
29. *Wang, Xiao, Zhou, Weikang, Zu, Can, Xia, Han, Chen, Tianze, Zhang, Yuansen, Zheng, Rui*, et al. Instructuie: multi-task instruction tuning for unified information extraction. arXiv preprint arXiv:2304.08085 (2023).
30. *Xu, Derong, Chen, Wei, Peng, Wenjun, Zhang, Chao, Xu, Tong, Zhao, Xiangyu, Wu, Xian, Zheng, Yefeng, Wang, Yang, & Chen, Enhong*. Large language models for generative information extraction: a survey. Frontiers of Computer Science, 18(6) (2024).
31. *Yao, Yunzhi, Mao, Shengyu, Zhang, Ningyu, Chen, Xiang, Deng, Shumin, Chen, Xi, & Chen, Huajun*. Schema-aware reference as prompt improves data-efficient knowledge graph construction. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 911–921 (2023).
32. *Zhao, Xiaoyan, Deng, Yang, Yang, Min, Wang, Lingzhi, Zhang, Rui, Cheng, Hong, Lam, Wai, Shen, Ying, & Xu, Ruifeng*. A comprehensive survey on relation extraction: recent advances and new frontiers. ACM Computing Surveys, 56(11) (2024): 1–39.
33. GATE Documentation. JAPE: Regular Expressions over Annotations. https://gate.ac.uk/releases/gate-5.0-build3244-ALL/doc/tao/splitch7.html (data obrashcheniya: 16.10.2025).
34. JSON Repair Tool. Online utility for fixing malformed JSON (well-formedness repair). https://jsonrepair.com (data obrashcheniya: 16.10.2025).

Information about the author

Nikita N. Oltyan – Post-graduate Student

The article was submitted 10.10.2025
Approved after reviewing 22.10.2025
Accepted for publication 30.10.2025