DOI: 10.15514/ISPRAS-2025-37(2)-20

Архитектура системы сбора и извлечения информации для интеллектуальной поисково-аналитической системы

[1,2] *Д.С. Серенко, ORCID: 0009-0003-6676-7255 <serenko.d.s@yandex.ru>*
[1,2] *Е.Д. Терентьев, ORCID: 0009-0003-6797-9292 <eterentevd@yandex.ru>*
[2] *Д.В. Зубарев, ORCID: 0000-0002-9687-6650 <zubarev@isa.ru>*
[2,3,4] *И.В. Соченков, ORCID: 0000-0003-3113-3765 <sochenkov@isa.ru>*

[1] *Российский университет дружбы народов имени Патриса Лумумбы, Россия, 117198, г. Москва, ул. Миклухо-Маклая, д. 6.*
[2] *Федеральный исследовательский центр «Информатика и управление» РАН, Россия, 119333, г. Москва, ул. Вавилова, д. 44/2.*
[3] *Институт проблем передачи информации им. А.А. Харкевича РАН, Россия, 127051, г. Москва, Большой Каретный пер., д. 19 стр. 1.*
[4] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. Александра Солженицына, д. 25.*

Аннотация. Данные из интернета служат основой для решения широкого круга задач, от информационного поиска до аналитической обработки. Рост объёмов данных повышает важность эффективного извлечения описательных сведений о документах (метаданные – заголовки, имена авторов, даты публикации и так далее) с научных и образовательных сайтов (веб-ресурсов). Традиционные методы сбора и извлечения информации на основе статических шаблонов малоэффективны при обработке веб-страниц с динамически формируемым содержанием. В работе предложена архитектура адаптивной системы сбора и извлечения информации, сочетающая стандартные методы извлечения данных с технологиями машинного обучения. Система имеет модульную структуру, включающую подсистемы управления заданиями, мониторинга и журналирования, краулинга (робота сбора информации), управления ссылками, извлечения метаданных. Подсистема краулинга обрабатывает как статически, так и динамически формируемое содержание через имитацию работы прикладного программного обеспечения для просмотра веб-страниц. Для извлечения метаданных применяется комбинированный подход, совмещающий структурированные правила и машинное обучение. Эксперименты показали успешное извлечение метаданных из различных веб-ресурсов, включая страницы с динамически формируемым содержанием и сложными структурами. Система обладает высокой точностью и устойчивостью к изменениям форматов данных, при этом строго соблюдаются этические нормы сбора данных, включая обязательное выполнение инструкций и применение разумных интервалов между запросами.

Ключевые слова: интеллектуальные поисково-аналитические системы; система сбора и извлечения информации; извлечение метаданных; веб-краулинг; динамический контент; машинное обучение; автоматизация сбора данных; браузерная эмуляция; MarkupLM.

Для цитирования: Серенко Д.С., Терентьев Е.Д., Зубарев Д.В., Соченков И.В. Архитектура системы сбора и извлечения информации для интеллектуальной поисково-аналитической системы. Труды ИСП РАН, том 37, вып. 2, 2025 г., стр. 263–280. DOI: 10.15514/ISPRAS–2025–37(2)–20.

Architecture of an information collection and extraction system for an intelligent search and analytical platform

[1,2] *D.S. Serenko, ORCID: 0009-0003-6676-7255 <serenko.d.s@yandex.ru>*
[1,2] *E.D. Terentev, ORCID: 0009-0003-6797-9292 <eterentevd@yandex.ru>*
[2] *D.V. Zubarev, ORCID: 0000-0002-9687-6650 <zubarev@isa.ru>*
[2,3,4] *I.V. Sochenkov, ORCID: 0000-0003-3113-3765 <sochenkov@isa.ru>*

[1] *RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russia.*
[2] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44, build. 2, Vavilova St, Moscow, 119333, Russia.*
[3] *A. A. Kharkevich Institute of Information Transmission Problems of the RAS, 19, build.1, Bolshoi Karetny per., Moscow, 127051, Russia.*
[4] *V.P. Ivannikov Institute for System Programming of the Russian Academy of Science, 25, Alexander Solzhenitsyn St., Moscow, 109004, Russia.*

Abstract. Internet data serves as the foundation for a wide range of tasks, from information retrieval to analytical processing. With the rapid growth of data volumes, efficient metadata extraction from dynamic web resources has become critically important. Traditional information collection and extraction methods based on static templates are largely ineffective when processing interactive content. This paper presents the architecture of an adaptive information collection and extraction system that integrates standard data extraction techniques with machine learning technologies. The system has a modular structure comprising the following subsystems: task management, monitoring and logging, crawling, link management, and metadata extraction. The crawling subsystem processes both static and dynamic content through browser emulation. A hybrid approach combining structured rules and machine learning is used for metadata extraction. Experimental results demonstrated successful metadata extraction from various web resources, including pages with dynamic content and complex structures. The system exhibited high accuracy and resilience to changes in data formats while strictly adhering to ethical data collection standards, such as compliance with robots.txt directives and applying reasonable request intervals. Thus, the proposed solution represents a significant step toward the development of universal data collection and extraction systems for modern information environments. The developed software tools have been utilized in populating the index databases of the Neopoisk system.

Keywords: intelligent search and analytical systems; information collection and extraction system; metadata extraction; web crawling; dynamic content; machine learning; automated data collection; browser emulation; MarkupLM.

For citation: Serenko D.S., Terentev E.D., Zubarev D.V., Sochenkov I.V. Architecture of an information collection and extraction system for an intelligent search and analytical platform. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 2, 2025, pp. 263-280 (in Russian). DOI: 10.15514/ISPRAS-2025-37(2)-20.

1. Введение

В современном информационном пространстве, характеризующемся стремительным ростом объёмов данных [1], интеллектуальные поисково-аналитические системы играют ключевую роль в предоставлении пользователям структурированной и релевантной информации. Эти системы не только обеспечивают доступ к данным, но и способствуют их глубокой аналитической обработке, что становится особенно важным при работе с динамическими и интерактивными веб-ресурсами [2].

Одной из ключевых задач при разработке таких систем является эффективное извлечение сведений о публикациях из различных источников, включая веб-страницы со сложной структурой, динамически обновляемыми разделами и подгрузкой данных без перезагрузки страницы. Традиционные методы сбора и извлечения информации, базирующиеся на статических шаблонах (элементы навигации по структуре документа: CSS-selectors, язык запросов XPath для навигации по XML-документам), часто сталкиваются с ограничениями

при обработке динамических веб-ресурсов, что требует ручной адаптации под каждый конкретный ресурс [3]. Это приводит к значительным временным и трудозатратным издержкам.

Для решения данной проблемы могут применяться методы машинного обучения [4]. Их использование требует разработки архитектуры, обеспечивающей универсальность и гибкость при сборе и извлечении метаданных. В данной работе предложена такая архитектура, объединяющая стандартные методы обработки данных с технологиями машинного обучения. Основной акцент сделан на адаптивность системы, её способность автоматически подстраиваться под особенности взаимодействия с различными веб-ресурсами для эффективного обхода ограничений и извлечения данных.

Ключевым элементом предложенной архитектуры является модульная структура, где каждая функциональная часть системы отвечает за отдельные этапы сбора, извлечения и сохранения метаданных. Такой подход обеспечивает масштабируемость и расширяемость системы, позволяя легко внедрять новые технологии и алгоритмы. В рамках настраиваемого задания, формируемого пользователем, предусмотрена возможность указания сценариев интерактивного взаимодействия подсистемы по сбору данных с динамическими элементами веб-страниц. Для обработки и извлечения метаданных предусмотрена возможность использования методов машинного обучения, которые позволяют автоматизировать извлечение данных и минимизировать необходимость ручной настройки.

Интеграция моделей машинного обучения с традиционными методами извлечения данных предоставляет пользователю возможность гибкого выбора оптимальных инструментов для решения конкретных задач. Это может быть как использование обученной модели для автоматического извлечения метаданных, так и ручное задание правил для обработки специфичных структур [5].

Таким образом, разработка и реализация предложенной архитектуры позволяют обеспечить не только высокую точность и эффективность сбора и извлечения метаданных, но и адаптивность системы к изменениям информационной среды. Это делает её актуальным и востребованным решением в условиях стремительного роста объемов данных и усложнения их структуры.

Основные результаты данной работы заключаются в следующем:

- разработана современная архитектура системы сбора и извлечения информации для интеллектуальной поисково-аналитической системы;
- предложена интеграция моделей машинного обучения в подсистему извлечения метаданных;
- представлены результаты работы системы сбора и извлечения информации из веб-ресурсов, созданной на основе представленной архитектуры.

2. Связанные работы

2.1 Виды систем сбора и извлечения информации

Системы сбора и извлечения информации (web-crawler, система краулинга, или "краулер") можно классифицировать по различным признакам, включая стратегию обхода веб-страниц и архитектуру. Универсальные системы осуществляют полный обход сайтов без учета специфики содержимого, что позволяет охватывать максимально широкий спектр информации, однако зачастую страдают от избыточности получаемых данных. В отличие от них, тематические (фокусированные) системы выбирают только страницы, соответствующие заданной тематике, что приводит к экономии сетевых и вычислительных ресурсов, но может

снижать полноту охвата. Инкрементальные системы сбора и извлечения информации обновляют уже собранные данные, извлекая лишь изменившуюся информацию, что позволяет снизить нагрузку на серверы, в то время как распределенные и параллельные системы ускоряют процесс сбора за счет координации работы нескольких серверов или процессов [6]. В последнее время появляются также специализированные системы сбора и извлечения информации, ориентированные на сбор данных с устройств Интернета вещей (IoT), анализирующие сервисные метаданные подключенных устройств. Таким образом, выбор конкретного типа системы определяется компромиссами между полнотой охвата, актуальностью информации и скоростью обработки.

2.2 Обзор существующих решений

Различные исследования демонстрируют разнообразие архитектурных подходов к реализации систем сбора и извлечения информации. Так, система полнотекстовых электронных библиотек (ПС ПЭБ), предложенная в работе [7], содержит модуль автоматического обхода и сбора данных с веб-сайтов (краулер) для автоматического наполнения коллекций документов, использующий регулярные выражения и шаблоны языка запросов для XML-документов XPath для фильтрации целевых документов. Это позволяет добиться высокой точности извлечения данных, однако требует ручной настройки конфигурационных файлов для каждой коллекции, что снижает гибкость при изменении структуры источников.

Работа [8] посвящена распределенной архитектуре системы автоматического обхода и индексирования веб-страниц, предназначенной для обхода огромного числа страниц. В предложенном решении основное внимание уделяется эффективному управлению очередью URL, оптимизации хранения данных и соблюдению сетевого этикета посредством распределения задач между множеством серверов. Разделение URL по хостам позволяет минимизировать межсерверные коммуникации, однако такие системы всё равно ограничены сетевыми задержками и нагрузкой на центральные сервера.

Обзор, проведенный в работе [9], охватывает широкий спектр методов обхода веб-страниц, включая универсальные, тематические и распределенные системы сбора и извлечения информации. Авторы акцентируют внимание на механизмах параллельного выполнения запросов и эффективном управлении очередью URL, описывая при этом методы приоритизации ссылок с использованием очередей с различными уровнями приоритета и алгоритмов обнаружения дубликатов с применением хеш-функций. Несмотря на богатство предлагаемых решений, в обзоре подчеркивается, что выбор стратегии сбора и извлечения информации напрямую зависит от конкретных задач, что затрудняет создание универсального решения.

Еще одним направлением развития являются системы на основе сервисно-ориентированной архитектуры и облачных вычислений. Сервисно-ориентированные архитектуры (SOA) позволяют разделять функциональные компоненты на независимые сервисы, взаимодействующие через стандартные программные интерфейсы (API), что обеспечивает гибкость, отказоустойчивость и масштабируемость. Облачные вычисления, в свою очередь, предоставляют возможность динамического выделения ресурсов для обработки больших объемов данных в распределенных системах. Примером такого подхода является архитектура веб-краулера как облачного сервиса (Crawler as a Service, CaaS), представленная в работе [10]. В данном решении процесс сбора и извлечения информации разделен на независимые микросервисы, что позволяет запускать несколько экземпляров системы параллельно в различных регионах облака и настраивать параметры через интерфейс программирования приложений на основе передачи репрезентативного состояния (REST API). Хотя такой подход позволяет значительно снизить время обхода за счет

масштабирования, он также может сталкиваться с дополнительными задержками, обусловленными передачей данных по сети между серверами распределенной среды.

Также следует отметить подход, основанный на распределенных вычислениях в сети (грид-вычислениях), описанный в работе [11]. Здесь для повышения производительности используется платформа Alchemi, позволяющая распределять процессы между несколькими вычислительными серверами в рамках среды .NET. Применение многопоточного выполнения и централизованного управления очередями URL позволяет добиться линейного уменьшения времени обхода при увеличении числа серверов, хотя эффективность этого подхода может снижаться из-за сетевых ограничений и неоднородности размеров обрабатываемых страниц.

Дополнительно, специализированные системы сбора и извлечения информации, описанные в работе [12], демонстрируют более точные методы извлечения информации за счет семантического анализа содержания. Однако такие системы зачастую требуют значительных вычислительных ресурсов и испытывают трудности при масштабировании в условиях быстро изменяющейся веб-среды.

В свою очередь, аналитический обзор систем для сбора данных с представительских сайтов [13] показывает разнообразие архитектурных подходов – от последовательных до параллельных и распределённых систем. Несмотря на то, что распределённые решения позволяют существенно ускорить процесс обхода, они не решают проблему повторной обработки одних и тех же страниц, что негативно сказывается на общей производительности.

Таким образом, несмотря на существенные достижения в области разработки систем сбора и извлечения информации, в каждом подходе наблюдаются определенные недостатки, что указывает на необходимость дальнейших исследований для создания универсальных архитектур, способных адаптироваться к динамичным изменениям веб-ресурсов и эффективно интегрироваться с внешними аналитическими модулями.

3. Архитектура предложенной системы

В разработанной архитектуре система сбора и извлечения информации реализована как набор независимых подсистем, каждая из которых выполняет свою специализированную задачу и предоставляет функциональность через API. Такой модульный подход позволяет обеспечить гибкость, масштабируемость и отказоустойчивость решения, а также упрощает интеграцию с внешними системами.

3.1 Общая структура системы

В основе системы лежит распределенная микросервисная архитектура (рис. 1), где каждая подсистема реализует специализированную задачу и предоставляет функциональность через API.

Система содержит следующие подсистемы:

- подсистема управления заданиями (Task Management Module);
- подсистема мониторинга и журналирования (Monitoring and Logging Module);
- подсистема краулинга (Crawler Module);
- подсистема управления ссылками (Frontier Module);
- подсистема извлечения метаданных (Parser Module).

Все сервисы взаимодействуют посредством стандартных REST-запросов, что обеспечивает высокую гибкость, масштабируемость и простоту интеграции (рис. 2). Процесс работы системы включает несколько последовательных этапов.

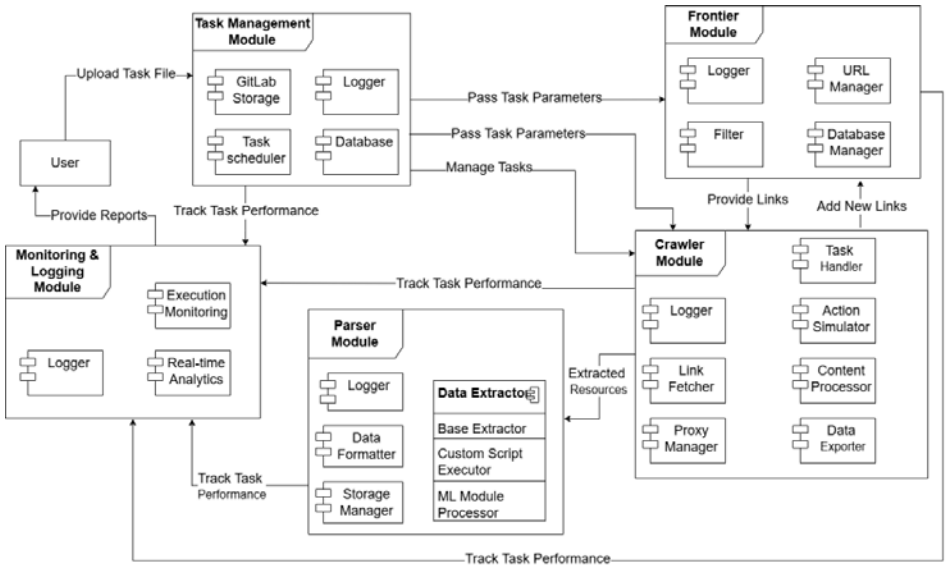

Рис. 1. Разработанная архитектура системы сбора и извлечения информации.
Fig. 1. Developed Architecture of the Information Collection and Extraction System.

Для обеспечения высокой производительности и масштабируемости подсистемы извлечения метаданных и автоматического обхода веб-страниц (краулинга) поддерживают множественные экземпляры, которые могут динамически масштабироваться в зависимости от нагрузки и объема обрабатываемых данных. В то же время другие подсистемы являются единственными экземплярами в системе, так как они выполняют централизованные функции управления заданиями, ведения журнала событий (логгирование) и координации ссылок, обеспечивая консистентность данных и контроль над процессами.

На первом этапе подсистема управления заданиями осуществляет загрузку файлов формата YAML, определяющих параметры обхода и извлечения данных. Далее происходит сбор данных, при котором подсистема краулинга обрабатывает веб-страницы, включая динамически формируемое содержание, используя при необходимости механизмы имитации работы браузера. Подсистема управления ссылками координирует процесс обхода сайтов. При обработке новых ссылок данная подсистема проверяет их наличие в базе данных, исключая дублирование, и управляет их статусами. Необработанные ссылки передаются подсистеме краулинга, а после их загрузки обновляются их статусы в базе данных, сохраняя родительско-дочерние связи между страницами.

Подсистема извлечения метаданных применяет комбинированные методы, сочетающие традиционное извлечение по шаблонам XPaths или CSS-selectors и модели машинного обучения, для обработки данных. Завершающим этапом является сохранение и валидация данных в унифицированных форматах с проверкой на соответствие заданным шаблонам.

Для контроля работы системы и диагностики ошибок используется подсистема мониторинга и логирования. Она собирает и анализирует метрики выполнения заданий, фиксирует временные характеристики этапов обработки данных и регистрирует ошибки. Логирование охватывает как общие события работы системы, так и детализированные записи о событиях по каждой подсистеме, что позволяет оперативно выявлять и устранять возможные неисправности.

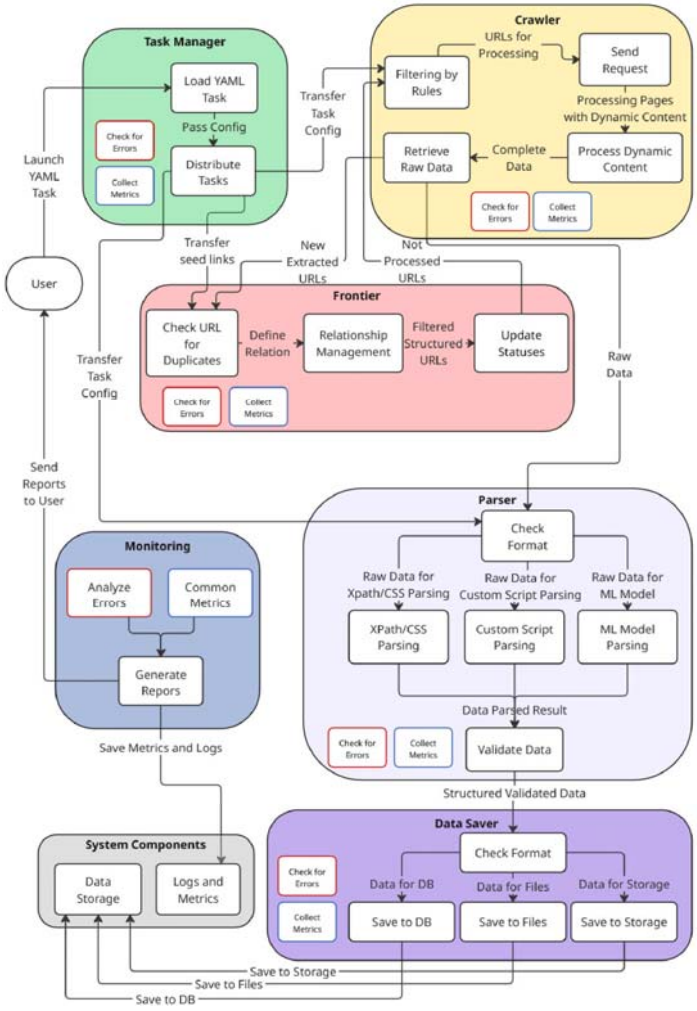

Рис. 2. Диаграмма потоков данных.
Fig. 2. Data Flow diagram.

3.2 Подсистема управления заданиями (Task Management Module)

Подсистема обработки заданий управляет созданием, хранением, планированием и распределением задач в системе сбора и извлечения информации. Она отвечает за обработку файлов формата YAML с параметрами заданий, их хранение и передачу в соответствующие подсистемы для выполнения.

Компоненты подсистемы:

- Task Manager – принимает задания через API или файловую систему, обрабатывает и сохраняет их в базе данных, распределяет параметры между соответствующими подсистемами;

- Task Scheduler – управляет выполнением заданий, обеспечивая их запуск и повторное выполнение согласно расписанию;
- GitLab Storage – отвечает за хранение файлов заданий в системе управления версиями GitLab [14], предоставляя централизованный доступ к ним;
- Database – хранит время добавления/ изменения заданий, их служебные параметры и состояние выполнения;
- Logger – регистрирует события выполнения заданий, включая их запуск, завершение и возможные ошибки.

3.3 Подсистема мониторинга и журналирования (Monitoring and Logging Module)

Для контроля работы системы сбора и извлечения информации реализована подсистема мониторинга и логирования.

Данная подсистема включает следующие компоненты:

- Logger: ведёт регистрацию всех событий и ошибок, возникающих в процессе работы системы, что позволяет проводить детальный анализ работы и выявлять неисправности;
- Execution Monitoring: отслеживает время выполнения задач, количество обработанных страниц и другие ключевые показатели эффективности, позволяя оперативно реагировать на сбои или отклонения от нормальной работы;
- Real-time Analytics: предоставляет возможность анализа в реальном времени, что способствует своевременному принятию управленческих решений.

3.4 Подсистема краулинга (Crawler Module)

Подсистема краулинга реализует двухуровневый подход к обработке веб-страниц. Для статически формируемого содержания применяются стандартные HTTP-запросы, в то время как для содержимого, формируемого с помощью языка программирования JavaScript, задействуется имитация работы браузера на базе библиотеки для автоматизации браузера Playwright [15], включающая имитацию действий пользователя.

Одной из ключевых особенностей подсистемы является динамическое переключения режимов работы. При обнаружении ограничений, направленных на предотвращение автоматического сбора данных (например, система проверки "человек или робот" (капча) или блокировка), подсистема переходит на альтернативный режим работы, строго соблюдая указания, заданные в файлах robots.txt, и устанавливая увеличенные интервалы между запросами для снижения нагрузки на сервер. Компоненты подсистемы:

- LinkFetcher – получает и загружает страницы через HTTP или браузерный эмулятор;
- ContentProcessor – анализирует загруженную информацию, извлекает ссылки, фильтрует их и определяет релевантность страницы;
- ProxyManager – управляет прокси-серверами и настройками подключения;
- ActionSimulator – выполняет пользовательские действия (прокрутка страницы, нажатие кнопок и т. д.) в программе имитации браузера для обработки динамически формируемого содержимого;
- TaskHandler – управляет параметрами заданий, переданными в подсистему;
- DataExporter – сохраняет целевые страницы и передаёт их в подсистему извлечения метаданных;

- Logger – регистрирует ключевые события процесса краулинга, фиксируя успешные и ошибочные запросы, время загрузки страниц и возможные сбои.

Эта подсистема обеспечивает адаптивный сбор веб-данных, поддерживая обработку как статически, так и динамически формируемого содержимого с соблюдением этического подхода к краулингу.

3.5 Подсистема управления ссылками (Frontier Module)

Подсистема управления ссылками отвечает за координацию процесса краулинга, хранение информации о посещённых и непосещённых URL, а также за оптимизацию повторного обхода веб-ресурсов. Она обеспечивает хранение структуры связей между страницами, контроль статусов обработки и исключение дублирующихся ссылок.

В основе подсистемы лежит реляционная база данных PostgreSQL [16], где центральной структурой является таблица UrlsTable. Она содержит уникальный идентификатор URL, полный адрес страницы, ссылку на родительский URL и текущий статус обработки. Такой подход позволяет эффективно управлять процессом сбора данных, анализировать взаимосвязи между страницами и оптимизировать стратегию обхода.

Для поддержки периодического обновления система использует метки времени, что позволяет запускать повторный обход страниц через заданные интервалы, обеспечивая актуальность данных без избыточного дублирования.

Компоненты подсистемы:

- UrlManager – отвечает за добавление, обновление и удаление URL, а также за их распределение между подсистемами краулинга;
- DatabaseManager – управляет хранением информации о ссылках в PostgreSQL, поддерживает связи между страницами и выполняет аналитические запросы;
- Filter – исключает хранение дублирующихся ссылок в базе данных;
- Logger – фиксирует изменения статусов ссылок, операции обновления и удаления URL, а также регистрирует ошибки и метрики работы подсистемы.

Подсистема играет важную роль в управлении процессом обхода сайтов, обеспечивая его эффективность, целенаправленность и структурированность.

3.6 Подсистема извлечения метаданных (Parser Module)

Подсистема извлечения метаданных отвечает за обработку загруженных веб-страниц и выделение из них структурированной информации. В её основе лежит комбинированный подход, сочетающий традиционные методы извлечения данных (XPath, CSS-selectors, регулярные выражения) с современными алгоритмами машинного обучения. Это позволяет эффективно обрабатывать как статические, так и сложные динамические структуры веб-документов.

Для повышения качества извлеченных данных реализована автоматическая валидация ряда метаполей (ISSN, EISSN, ISBN, EISBN, DOI [17]) на основе заранее определённых шаблонов. Такой подход гарантирует соответствие данных установленным требованиям и снижает количество ошибок в процессе обработки.

Компоненты подсистемы:

- DataExtractors – набор методов для извлечения данных из веб-страниц:
  - BaseExpExtractor – использует регулярные выражения, XPath и CSS-selectors для извлечения информации;
  - CustomScriptExecutor – загружает и выполняет пользовательские Python-скрипты для специфических задач извлечения;

  - MLModelProcessor – применяет предобученные модели машинного обучения для определения и извлечения сущностей в тексте.
- DataFormatter – унифицирует извлечённые данные в единый формат;
- StorageManager – отвечает за сохранение обработанных и исходных данных во внешние системы хранения;
- Logger – регистрирует события процесса извлечения, фиксируя успешные операции и возможные ошибки. Подсистема играет важную роль в управлении процессом обхода сайтов, обеспечивая его эффективность, целенаправленность и структурированность.

Эта подсистема обеспечивает высокую точность и универсальность процесса извлечения метаданных, позволяя адаптировать методы обработки под различные структуры веб-страниц.

3.7 Интеграция с внешними системами хранения данных

Система поддерживает интеграцию с различными внешними хранилищами данных, что обеспечивает гибкость в организации хранения как исходных веб-страниц, так и структурированной информации, извлечённой в процессе обработки. В зависимости от типа данных могут использоваться различные системы хранения.

Например, для хранения исходных веб-страниц может использоваться облачное объектное хранилище, совместимое со стандартом Simple Storage Service (S3-совместимое хранилище), позволяющее загружать и управлять неструктурированными данными в распределённой среде.

Структурированные данные, извлечённые подсистемой извлечения метаданных, могут сохраняться, например, в графовые и реляционные базы данных, такие как ArangoDB [18] или PostgreSQL, позволяя эффективно управлять взаимосвязями между сущностями, выполнять сложные аналитические запросы и обеспечивать целостность данных. Дополнительно может осуществляться экспорт данных в форматах JSON или CSV, упрощая интеграцию с внешними аналитическими и поисковыми системами.

4. Интеграция моделей машинного обучения в подсистему извлечения данных

Современные интеллектуальные поисково-аналитические системы требуют эффективных механизмов сбора и обработки данных из различных источников, включая веб-документы с динамической структурой. Традиционные методы извлечения информации, основанные на статических правилах, таких как XPath и CSS-selectors, обладают ограниченной гибкостью и требуют значительных затрат на настройку. В связи с этим возникает необходимость в интеграции методов машинного обучения, позволяющих адаптивно анализировать и извлекать данные без предварительной настройки под конкретную структуру страницы.

Одним из перспективных решений в данной области является применение моделей, предобученных на структурированных данных, таких как HTML-документы. В качестве одного из таких решений рассматривается модель MarkupLM [19], предложенная для обработки документов с разметкой. В отличие от классических моделей обработки естественного языка (NLP-моделей), MarkupLM использует информацию о структуре документа через встроенные векторные представления путей XPath (XPath-эмбеддинги), что делает её особенно эффективной для задач извлечения информации из HTML-страниц.

Преимущества интеграции машинного обучения в процесс извлечения данных:

- обработка сложных зависимостей в данных. Многие веб-документы содержат сложные взаимосвязи между элементами (например, вложенные таблицы, списки,

интерактивные элементы). Глубокие модели, такие как MarkupLM, способны учитывать иерархическую структуру документа, улучшая точность извлечения информации;

- снижение затрат на поддержку. В отличие от традиционных методов, требующих постоянного обновления правил обхода и извлечения данных, машинное обучение позволяет автоматизировать процесс адаптации к новым источникам данных, снижая временные и трудозатраты на настройку системы.

4.1 MarkupLM как основа для специализированного извлечения данных

MarkupLM — это предобученная модель, ориентированная на обработку документов с разметкой, таких как HTML и XML [20].

Её архитектура включает несколько ключевых компонентов:

- текстовые векторные представления (эмбеддинги) — числовое представление текста, аналогичное традиционным языковым моделям;
- XPath-эмбеддинги — кодирование структуры документа в виде пути к элементу в дереве документа, что позволяет модели учитывать вложенность элементов.

Хотя MarkupLM уже показывает хорошие результаты в анализе веб-страниц, её эффективность можно значительно повысить с помощью дообучения (тонкой настройки, fine-tuning) на специализированных наборах данных. Например, дообучение модели на данных о статьях и книгах (метаданные, такие как ISSN, DOI, авторы, названия) позволит адаптировать её к задачам извлечения структурированной информации в конкретных предметных областях. Такой подход обеспечит более точные и надёжные результаты, что делает MarkupLM отличной основой для создания специализированных решений в области автоматического анализа веб-документов.

5. Апробация и результаты

Был отобран набор из 21 веб-сайта, представляющих как иностранные, так и российские веб-ресурсы. Данные веб-ресурсы условно разделены на две группы в зависимости от особенностей обработки – статические и динамические. Ресурсы динамического типа требуют применения расширенных методов, таких как имитация работы браузера, управление файлами, сохраняющими информацию о пользователе (cookies), и автоматизация нажатий, что обеспечивает полноценный обход страниц с динамически формируемым содержанием. Статические ресурсы обрабатываются с использованием стандартных HTTP-запросов, что обеспечивает высокую скорость и эффективность сбора данных.

Динамические ресурсы:

- Institute of Electrical and Electronics Engineers - https://ieeexplore.ieee.org/
- Association for Computing Machinery - https://dl.acm.org/
- World Scientific - https://www.worldscientific.com/
- Society of Photo-Optical Instrumentation Engineers - https://www.spiedigitallibrary.org/
- SAE International - https://www.sae.org/
- Sage Publishing - https://www.sagepub.com/
- Duke University Press - https://www.dukeupress.edu/
- EDP Sciences - https://www.edpsciences.org/
- Emerald Publishing - https://www.emerald.com/insight/
- Institute of Physics - https://iopscience.iop.org/
- European Mathematical Society - https://ems.press/

Статические ресурсы:

- Репозиторий Белорусского государственного педагогического университета им. М. Танка - https://elib.bspu.by/
- Репозиторий Гомельского государственного университета имени Франциска Скорины -https://elib.gsu.by/
- Репозиторий Белорусского государственного технологического университета - https://elib.belstu.by/
- Euromonitor - https://www.euromonitor.com/
- CyberLeninka - https://cyberleninka.ru/
- ProQuest (BlackFreedom) - https://blackfreedom.proquest.com/
- KSF Lebedev - https://ksf.lebedev.ru/
- Cambridge - https://www.cambridge.org/core/
- Успехи химии - https://www.uspkhim.ru/
- АНО Редакция журнала «УФН» - https://ufn.ru/

В исследовании учитывались как иностранные, так и российские ресурсы, демонстрирующие различную степень сложности при обработке. Экспериментальная оценка показала, что 10 из 21 (примерно 48%) сайтов успешно обрабатываются с помощью стандартных HTTP-запросов без необходимости применения дополнительных механизмов. Однако для 11 из 21 (около 52%) ресурсов требовались расширенные подходы, включающие имитацию работы браузера, управление cookies и автоматизацию взаимодействия с элементами страницы, такими как кнопки нумерации страниц, динамическая подгрузка содержания и обработка множественных выпадающих окон.

Полученные результаты подтверждают, что стандартные методы краулинга применимы лишь к части веб-ресурсов, тогда как остальные требуют интеграции современных технологий для полноценного обхода и высокоточного извлечения данных. Таким образом, экспериментальные данные подчеркивают важность комбинированного подхода в разработке систем сбора и извлечения информации, адаптирующихся к специфике различных типов веб-ресурсов.

Для извлечения метаданных использовались два подхода: BaseExpExtractor, который применяет регулярные выражения и язык XPath для выделения информации, и CustomScriptExecutor, позволяющий загружать и выполнять пользовательские Python-скрипты для специфических задач извлечения. Это обеспечивало адаптивность системы к различным структурам веб-ресурсов и повышало полноту сбора данных.

Разработанная система сбора и извлечения информации использовалась для наполнения индексных баз системы Неопоиск [21]. На основе полученных данных была проведена детальная оценка полноты извлечения метаданных (рис. 3). В общей сложности было собрано 8830424 документа, для которых проведён анализ полноты заполнения ключевых метаполей.

Результаты показали высокую полноту извлечения ключевых метаданных, таких как заголовки, авторы, даты публикации и названия журналов. Поля, связанные с идентификацией издательства и периодических изданий, также демонстрируют высокую степень заполненности. Вместе с тем, извлечение цифровых идентификаторов и ключевых слов оказалось менее полным, что подчёркивает необходимость дальнейшей оптимизации методов обработки данных. В некоторых случаях неполнота данных была обусловлена их отсутствием на исходных веб-ресурсах, что накладывает естественные ограничения на полноту извлечения.

Эти результаты подтверждают практическую применимость предложенного решения для обеспечения детальной информационной поддержки. В то же время они указывают на необходимость дальнейшей оптимизации методов обработки динамических элементов веб-страниц, а также разработки более эффективных подходов к извлечению слабоструктурированных метаполей.

Для более глубокого анализа работы системы были сформированы четыре таблицы, демонстрирующие результаты извлечения метаданных из журналов. Табл. 1 и табл. 2 содержат показатели, характеризующие долю и абсолютное количество извлечённых метаданных в пяти журналах, где система показала наилучшие результаты, тогда как табл. 3 и табл. 4 отражают соответствующие показатели для пяти журналов с наименьшей полнотой извлечения. Сопоставление этих данных позволяет сделать вывод о различиях в эффективности обработки различных ресурсов и подчёркивает необходимость применения специализированных методов для обеспечения высокого качества извлечения метаданных в зависимости от специфики веб-ресурсов.

В таблицах представлены данные по следующим научным изданиям:

- IEEE MGWL — IEEE Microwave and Guided Wave Letters;
- IEEE PTL — IEEE Photonics Technology Letters;
- Demography — Demography Journal;
- IEEE TBE — IEEE Transactions on Biomedical Engineering;
- Int. J. STD & AIDS — International Journal of STD & AIDS;
- J. Trop. Ped. — Journal of Tropical Pediatrics;
- Ind. J. Pub. Adm. — Indian Journal of Public Administration;
- Theater — Theater;
- Mod. Lang. Q. — Modern Language Quarterly;
- S. Atl. Q. — South Atlantic Quarterly.

Для удобства анализа в таблицах использованы следующие сокращенные обозначения полей метаданных:

- Title — заголовок публикации;
- Authors — авторы;
- Date — дата публикации;
- Doi — цифровой идентификатор публикации (Digital Object Identifier);
- Publ — издательство;
- Abstr — аннотация;
- Issn — международный стандартный серийный номер (International Standard Serial Number);
- Kwds — ключевые слова;
- Issue — номер выпуска журнала;
- Vol — том журнала;
- Pages — страницы публикации.

6. Заключение

В данной работе показано, что представленная архитектура демонстрирует практическую применимость при извлечении метаданных с широкого спектра веб-ресурсов, включая динамические и структурно сложные страницы. Отдельное внимание было уделено анализу эффективности при работе с ресурсами, где применяются механизмы защиты от автоматического сбора информации, а также внедрению модели машинного обучения для извлечения метаданных. Таким образом, разработанное решение представляет собой шаг в развитии универсальных систем сбора и анализа данных для современных информационных сред. Представленные программные средства были использованы при наполнении

индексных баз системы Неопоиск.

Несмотря на достигнутые успехи, существует потенциал для дальнейшего совершенствования отдельных компонентов системы. В частности, оптимизация моделей машинного обучения позволит добиться ещё более точного извлечения информации, что обеспечит высокую адаптивность решения в условиях постоянно меняющихся веб-ресурсов.

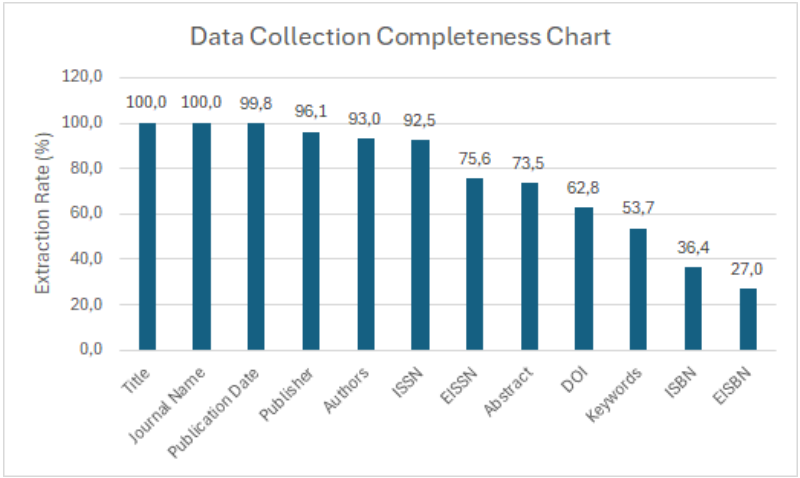

Рис. 3. Статистика полноты сбора метаданных.
Fig. 3. Metadata Collection Completeness Statistics.

Табл. 1. Доля извлечённых метаданных в лучших журналах (%).
Table 1. Share of Extracted Metadata in Top Journals (%).

Journals	Metadata Fields										
	Title	Authors	Date	Doi	Publ	Abstr	Issn	Kwds	Issue	Vol	Pages
J. Trop. Ped.	100	100	100	100	100	100	100	100	100	100	100
Ind. J. Pub. Adm.	100	99	100	100	100	100	100	99	100	100	100
Theater	100	99	100	100	100	94	100	96	100	100	100
Mod. Lang. Q.	100	95	100	100	100	96	100	96	100	100	100
S. Atl. Q.	100	99	100	100	100	90	100	94	100	100	100

Табл. 2. Количество извлечённых метаданных в лучших журналах.
Table 2. Number of Extracted Metadata in Top Journals.

Journals	Metadata Fields										
	Title	Authors	Date	Doi	Publ	Abstr	Issn	Kwds	Issue	Vol	Pages
J. Trop. Ped.	1218	1218	1218	1218	1218	1218	1218	1218	1218	1218	1218
Ind. J. Pub. Adm.	19697	19485	19697	19697	19697	19643	19697	19470	19697	19697	19697
Theater	4088	4036	4088	4088	4088	3835	4088	3938	4088	4088	4088
Mod. Lang. Q.	12176	11529	12176	12176	12176	11678	12176	11704	12176	12176	12175
S. Atl. Q.	3516	3470	3516	3516	3516	3156	3516	3300	3516	3516	3516

Табл. 3. Доля извлечённых метаданных в худших журналах (%).
Table 3. Share of Extracted Metadata in Worst Journals (%).

Journals	Metadata Fields										
	Title	Authors	Date	Doi	Publ	Abstr	Issn	Kwds	Issue	Vol	Pages
J. Trop. Ped.	100	94	100	100	100	57	1	25	100	100	93
Ind. J. Pub. Adm.	100	78	100	100	100	1	100	1	100	100	99
Theater	100	86	100	100	100	6	100	2	100	100	100
Mod. Lang. Q.	100	96	100	100	100	10	100	3	100	100	100
S. Atl. Q.	100	95	100	100	100	10	100	6	100	100	100

Табл. 4. Количество извлечённых метаданных в худших журналах.
Table 4. Number of Extracted Metadata in Worst Journals.

Journals	Metadata Fields										
	Title	Authors	Date	Doi	Publ	Abstr	Issn	Kwds	Issue	Vol	Pages
J. Trop. Ped.	5496	5141	5496	5496	5496	3127	35	1366	5496	5496	5098
Ind. J. Pub. Adm.	1832	1432	1832	1832	1832	13	1832	10	1832	1832	1807
Theater	2203	1886	2203	2198	2203	130	2203	53	2203	2203	2203
Mod. Lang. Q.	4853	4642	4853	4853	4853	495	4853	157	4853	4853	4853
S. Atl. Q.	8295	7857	8295	8295	8295	807	8295	470	8295	8295	8295

Список литературы / References

[1]. Jin, X., Wah, B. W., Cheng, X., & Wang, Y. (2015). Significance and Challenges of Big Data Research. Big Data Research, 2(2), 59–64. doi:10.1016/j.bdr.2015.01.006.

[2]. Китаев, Е. Л., & Скорнякова, Р. Ю. (2019). StructScraper--инструмент для динамического включения в контент веб-страницы семантических данных внешних веб-ресурсов. Научный Сервис в Сети Интернет, 21, 424–431.

[3]. Weichselbraun, A., Brasoveanu, A. M. P., Waldvogel, R., & Odoni, F. (2020). Harvest - An Open Source Toolkit for Extracting Posts and Post Metadata from Web Forums. 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 438–444. doi:10.1109/WIIAT50758.2020.00065.

[4]. Choi, W., Yoon, H.-M., Hyun, M.-H., Lee, H.-J., Seol, J.-W., Lee, K. D., Yoon, Y. J., Kong, H. (2023). Building an annotated corpus for automatic metadata extraction from multilingual journal article references. PloS One, 18(1), e0280637.

[5]. Patnaik, S., Babu, C., & Bhave, M. (08 2021). Intelligent and Adaptive Web Data Extraction System Using Convolutional and Long Short-Term Memory Deep Learning Networks. Big Data Mining and Analytics, 4, 279–297. doi:10.26599/BDMA.2021.9020012.

[6]. Yu, L., Li, Y., Zeng, Q., Sun, Y., Bian, Y., & He, W. (2020). Summary of web crawler technology research. Journal of Physics: Conference Series, 1449(1), 012036. doi:10.1088/1742-6596/1449/1/012036.

[7]. Назаренко Г. И., Плотникова В. А., Смирнов И. В., Соченков И. В., Тихомиров И. А. (2010). Программные средства создания и наполнения полнотекстовых электронных библиотек. Электронные Библиотеки: Перспективные Методы и Технологии, Электронные Коллекции: XII Всероссийская Научная Конференция RCDL.

[8]. Najork, M. (2009). Web Crawler Architecture.

[9]. Kausar, M. A., Dhaka, V. S., & Singh, S. K. (2013). Web crawler: a review. International Journal of Computer Applications, 63(2), 31–36.

[10]. ElAraby, M. E., Moftah, H. M., Abuelenin, S. M., & Rashad, M. Z. (2018). Elastic web crawler service-oriented architecture over cloud computing. Arabian Journal for Science and Engineering, 43(12), 8111– 8126.

[11]. ElAraby, M. E., Sakre, M. M., Rashad, M. Z., & Nomir, O. (2012). Crawler architecture using grid computing. International Journal of Computer Science & Information Technology, 4(3), 113.

[12]. Якубчик В. С., Попов О. Р., Крамаров С. О. (2023). Специализированные web-краулеры: на пути к семантическим моделям организации информационного поиска. Universum: Технические Науки: Электрон. Научн. Журн., 4(109). Available at: https://7universum.com/ru/tech/archive/item/15315.

[13]. Печников А. А., Сотенко Е. М. (2017). Программы-краулеры для сбора данных о представительских сайтах заданной предметной области – аналитический обзор. Современные Наукоемкие Технологии, (2), 58–62. Available at: https://top-technologies.ru/ru/article/view?id=36585.

[14]. The most-comprehensive AI-powered DevSecOps platform. GitLab. Available at: https://about.gitlab.com/, accessed 31.03.2025.

[15]. Fast and reliable end-to-end testing for modern web apps. Playwright Python. Available at: https://playwright.dev/, accessed 31.03.2025.

[16]. PostgreSQL: The world's most advanced open source database. Available at: https://www.postgresql.org/, accessed 31.03.2025.

[17]. Digital Object Identifier. Available at: https://www.doi.org/, accessed 31.03.2025.

[18]. ArangoDB: Multi-Model Database for Your Modern Apps. Available at: https://arangodb.com/, accessed 31.03.2025.

[19]. MarkupLM. Available at: https://huggingface.co/docs/transformers/model_doc/markuplm, accessed 31.03.2025.

[20]. Li, J., Xu, Y., Cui, L., & Wei, F. (2022). MarkupLM: Pre-training of Text and Markup Language for Visually-rich Document Understanding. arXiv [Cs.CL]. Available at: http://arxiv.org/abs/2110.08518.

[21]. Неопоиск. Available at: https://promo.neopoisk.ru/about, accessed 31.03.2025.

Информация об авторах / Information about authors

Данил Сергеевич СЕРЕНКО является студентом кафедры математического моделирования и искусственного интеллекта РУДН имени Патриса Лумумбы, научным сотрудником Федерального исследовательского центра "Информатика и управление" Российской академии наук (ФИЦ ИУ РАН). Область научных интересов – искусственный интеллект, информационный поиск.

Danil Sergeevich SERENKO is a student at the Department of Mathematical Modeling and Artificial Intelligence of the Patrice Lumumba RUDN University, a researcher at Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences. His research interests include AI, information retrieval.

Егор Дмитриевич ТЕРЕНТЬЕВ является студентом кафедры математического моделирования и искусственного интеллекта РУДН имени Патриса Лумумбы, научным сотрудником Федерального исследовательского центра "Информатика и управление" Российской академии наук (ФИЦ ИУ РАН). Область научных интересов – искусственный интеллект, информационный поиск.

Egor Dmitrievich TERENTEV is a student at the Department of Mathematical Modeling and Artificial Intelligence of the Patrice Lumumba RUDN University, a researcher at Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences. His research interests include AI, information retrieval.

Денис Владимирович ЗУБАРЕВ является научным сотрудником Федерального исследовательского центра "Информатика и управление" Российской академии наук (ФИЦ ИУ РАН). Область научных интересов – искусственный интеллект, информационный поиск, поиск текстовых заимствований.

Denis Vladimirovich ZUBAREV is a researcher at Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences. His research interests include AI, information retrieval, text plagiarism detection.

Илья Владимирович СОЧЕНКОВ – кандидат физико-математически наук, ведущий научный сотрудник ФИЦ ИУ РАН, ведущий научный сотрудник ИСП РАН, ведущий научный сотрудник ИППИ РАН. Сфера научных интересов: обработка естественного языка, методы информационного поиска, обработка больших массивов текстовой информации.

Ilia Vladimirovich SOCHENKOV – Cand. Sci. (Phys.-Math.), lead researcher at FRC CSC RAS, lead researcher at ISP RAS, lead researcher at IITP RAS. Research interests: Natural Language Processing, Information Retrieval, Big Data & Text Mining.