ТЕХНИЧЕСКИЕ НАУКИ

УДК: 004.624

Сравнение и анализ методов и инструментов, применяемых для сбора данных

Н.С. Юлкина, М.С. Шмелев, И.К. Фомина
Государственный университет морского и речного флота имени адмирала Макарова, г. Санкт-Петербург, Российская Федерация

Аннотация. Вопрос классификации парсеров для выполнения конкретных задач по поиску необходимой информации в интернете еще мало изучен. Важность и актуальность его заключается в том, что сбор и анализ данных помогают упорядочить полученные сведения в соответствии с предъявляемыми к ним требованиями, оценить конкурентоспособность компании на рынке. Целью данной работы в связи с этим является определение преимуществ и недостатков различных методов и инструментов для поиска, сбора и анализа данных, а также проблем, которые они решают. Для достижения поставленной цели был проведен сравнительный анализ этих методов и инструментов, выявлены их особенности. В результате проделанной работы выявлены наиболее эффективные средства парсинга для конкретной области их применения. Данная статья поможет максимально точно и правильно выбрать инструментарий для решения задачи по сбору необходимой информации.

Ключевые слова: сбор данных, парсинг, облачные технологии, автоматизированный сбор информации, скрэппинг, анализ текста

Comparison and Analysis of Methods and Tools for Data Collection

Nadezhda S. Yulkina, Mikhail S. Shmelev, Inga K. Fomina
Admiral Makarov State University of Maritime and Inland Shipping, Saint Petersburg, Russian Federation

Abstract. The issue of classification of parsers to perform specific tasks of searching for necessary information on the Internet has not yet been studied enough. Its importance and relevance lies in the fact that data collection and analysis help to organize the information received in accordance with the requirements imposed on them, to assess the competitiveness of the company in the market. The aim of this work was to determine the advantages and disadvantages of each method, as well as the problems that it solved. To achieve this goal, various methods and tools for searching, collecting and analyzing data were considered, their comparative analysis was carried out, and their features were identified. As a result of the work carried out, the most effective parsing tools for each application were identified. This article will help you to choose the tools for the task being solved as accurately and correctly as possible.

Keywords: data collection, parsing, cloud technologies, automated information collection, scrapping, text analysis

Введение. Цифровизация современной жизни привела к тому, что в открытом доступе скопилось огромное количество самой разнообразной информации. Ее собирают различные компании и хранят в огромных базах данных, генерируют пользователи, создавая на ее основе статьи и ведя блоги, оставляя оценивающие комментарии. Если эти данные собрать и систематизировать, то можно использовать их для научных исследований, государственного управления и решения множества других задач. Например, проанализировав комментарии по поводу определённого товара на маркетплейсе, можно узнать, чем довольны и недовольны покупатели, какие они выделяют недостатки и положительные качества изделия. А исправив недочёты и улучшив сильные стороны товара, можно увеличить его продажи.

Сбор данных актуален, потому что они помогают:

– принимать обоснованные решения, проводить анализ и оценивать эффективность действий;

– выявить узкие места в бизнес-процессах и оптимизировать их для повышения эффективности работы и экономии ресурсов организации;

– анализировать тренды и формировать прогнозные модели, позволяющие адаптироваться к изменяющимся условиям рынка [1–3].

Существует множество областей применения такого сбора данных, поэтому возникает следующая проблема: какими методами и инструментами при сборе данных следует пользоваться в каждой конкретной области для получения от него наибольшего эффекта.

Несмотря на то, что тема парсинга достаточно хорошо освещена в научных работах, исследуемая в статье проблема фигурирует в них довольно редко, так как ее сложно описать теоретически. Целью статьи является проведение сравнительного анализа методов и инструментов, применяемых для сбора и анализа полученных данных. Для достижения этой цели необходимо решить следующие задачи:

1. Изучить понятие парсинга.
2. Обозначить средства и методы сбора и анализа данных.
3. Выявить негативные и положительные стороны этих средств и методов.
4. Определить области применения инструментов для парсинга.
5. Представить классификацию программных реализаций парсинга данных.
6. Написать собственный парсер и использовать его в конкретной предметной области.

Основная часть. Парсинг — автоматизированный сбор и систематизация данных. Преимуществом его являются скорость сбора данных и, как следствие, возможность собрать большой объём информации за короткое время. Недостаток — в необходимости большого объема знаний в области программирования.

Процесс парсинга можно разделить на два этапа:

1) получение информации с веб-ресурса, то есть получение текста;

2) разбор полученной информации, то есть получение данных из текста.

Процессы получения данных для статических и динамических сайтов будут отличаться. Динамические сайты не содержат информации прямо в коде веб-страницы. Для того чтобы получить ее, пользователь должен выполнить определенные действия, например, нажать на кнопку, только после этого содержимое документа изменится, то есть данные будут получены не сразу, а поэтапно и требуют определенных действий со стороны пользователя.

Статические сайты позволяют получить информацию посредством одного HTTP GET-запроса, потому что данные в них не изменяются. После того как текст с информацией получен, нужно провести анализ документа и извлечь необходимую информацию.

В данном исследовании рассматривается динамический сайт. Для парсинга данных используются два следующих метода:

1-й метод — сбор HTML-кода с нужной информацией с сайта с использованием HTTP-запроса при помощи библиотеки Requests. Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту.

Библиотека Requests для Python позволяет работать с HTTP-запросами любого уровня сложности, используя простой синтаксис. Это помогает не тратить время на написание кода, а быстро взаимодействовать с серверами.

2-й метод — разбор собранных данных на составляющие (в нашем случае это заголовок, описание, ссылка URL, ссылка на обсуждение, количество комментариев) с помощью библиотеки BS4.

Beautiful Soup — это библиотека Python для извлечения данных из файлов HTML и XML. Она работает с парсером, чтобы предложить естественные способы навигации, поиска и изменения дерева разбора. Она обычно экономит программистам часы и дни работы.

Основными областями применения парсинга являются:

1. Веб-скрапинг — извлечение данных с веб-страниц, таких как цены товаров, информация о компаниях или новости.
2. Анализ текста — извлечение и анализ информации из текстовых документов, таких как отчеты, статьи или социальные медиа.
3. Обработка данных — извлечение и преобразование данных из различных форматов, таких как XML.
4. Машинное обучение — предварительная обработка данных перед их использованием в моделях машинного обучения.
5. Автоматизация задач — автоматизирование процессов, таких как сбор информации, обновление баз данных или генерация отчетов.
6. Научные исследования — извлечение данных из научных статей, баз данных или других источников для анализа и исследования.

В данной работе используется Python, потому что этот язык отличается гибкостью и простотой изучения, это делает его идеальным для начинающих разработчиков. Кроме того, он может похвастаться обширной поддержкой сообщества и многочисленными библиотеками для парсинга.

При парсинге данных возникают следующие трудности:

– шаблонный поиск информации на сайте. Многие парсеры придерживаются определённых шаблонов в работе, например, при поиске и копировании информации. Веб-сайты, анализируя данные действия, выявляют определённые шаблоны работы, что позволяет установить, что веб-сайт посещал не человек, а робот, и принять определённые меры. Чтобы предотвратить санкции со стороны сайта из-за использования шаблонной работы парсера, нужно прописывать разные пути поиска информации;

– большая нагрузка на сайт. При парсинге данных нужно не забывать, что возможности сайта небезграничны. При большом количестве запросов сайт может не выдержать (упасть), чтобы не возникало таких проблем, нужно собирать информацию в щадящем для сайта режиме, прописывать случайные паузы в работе парсера, чтобы его действия были приближены к действиям человека;

– Игнорирование Cookie. При сборе данных не следует забывать о файлах Cookie, ведь они нередко облегчают работу парсера. Cookie хранят в себе информацию о взаимодействии пользователя с сайтом, следовательно, cookie могут хранить в себе такие данные, как логин, пароль, иные настройки пользователя. Чтобы не вводить эти данные при каждом новом сборе информации и тем самым нагружать веб-сайт, следует хранить в коде файлы cookie.

Помимо собственных написанных парсеров, есть и готовые программы, нацеленные на решение определенных задач. Программы и инструменты для парсинга соответствуют следующей классификации:

1. Облачные парсеры. Главное достоинство облачных парсеров — независимость от платформы и устройства. Все хранится и производится в облаке, нужно только скачать результаты работы алгоритмов. У таких парсеров может быть веб-интерфейс и/или API. Это полезно, когда нужно автоматизировать парсинг данных и делать его регулярно. Из русскоязычных облачных парсеров можно, к примеру, назвать Xmldatafeed, Диггернаут, Catalogloader [4]. Веб-интерфейс — это посредник в обмене данными между пользователем и приложением. Примером может служить электронная почта. API (Application Programming Interface, или программный интерфейс приложения) — это совокупность инструментов и функций в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой. Любой из сервисов, приведенных выше, можно опробовать бесплатно. Однако этого достаточно только для ознакомления с функционалом, так как в бесплатной версии есть ограничения по объему данных, которые можно парсить, или по времени использования.

2. Десктопные парсеры. Это программы для компьютера, в основном они разработаны для Windows, поэтому на macOS их нужно запускать при помощи виртуальных машин. Также некоторые парсеры имеют портативные версии — их можно запускать с флешки или внешнего накопителя. Примерами десктопных парсеров являются ParserOK, Datacol, Screaming Frog, ComparseR, Netpeak Spider [5].

Виды парсеров по технологии:

1. Браузерные расширения. Собирают нужные данные из исходного кода страниц и позволяют сохранять в удобном формате. Такие парсеры полезны, когда нужно собрать данные с одной или двух страниц. Вот популярные парсеры для Google Chrome: Parsers, Scraper, Data Scraper, Kimono.

2. Google Таблицы. С помощью двух формул (IMPORTXML, IMPORTHTML) и Google Таблицы можно собирать любые данные с сайтов бесплатно.

Функция IMPORTXML использует язык запросов XPath и позволяет парсить данные с XML-фидов, HTML-страниц и других источников. XPath — язык запросов к элементам XML-документа. XML-фид — это структурированный файл, который служит для хранения всевозможной информации об объектах, услугах или товарах, открывая возможности для удобного импорта данных на другие платформы. Функция может принять два значения:

- ссылка на страницу либо фид, с которых необходимо получить информацию;
- XPath-запрос (специальный формат запроса, указывающий на определенный элемент с необходимыми данными для извлечения).

Главная особенность применения данной функции — не нужно изучать синтаксис XPath-запросов. Чтобы получить XPath-запрос для элемента с данными, нужно открыть код страницы в браузере, кликнуть правой кнопкой мыши по нужному элементу и выбрать: копировать → копировать XPath.

У функции IMPORTHTML меньше возможностей, с ее помощью можно собрать данные из таблиц или списков на странице.

Она способна принять три значения:

– ссылка на страницу, с которой необходимо собрать данные;

– характеристика элемента, которая содержит нужные данные. Для сбора информации из таблицы укажите table. Для парсинга списков — параметр list.

– число — порядковый номер элемента в коде страницы.

Виды парсеров по сферам применения:

1. Для организаторов совместных покупок. Существуют ресурсы, которые разрабатываются для организаторов совместных покупок. Производители товаров, к примеру, одежды, устанавливают эти программы на свои сайты. Любой посетитель сайта может использовать эту программу и получить полный список товаров, доступных для покупки.

Чем удобны эти парсеры:

- интуитивно понятный интерфейс;
- возможность выгружать отдельные товары, разделы или весь каталог;
- можно выгружать данные в удобном формате, например, в формате CSV или JSON.

Популярные парсеры для организации совместных покупок: SPparser.ru, Облачный парсер, Турбо.Парсер, PARSER.PLUS, Q-Parser.

2. Парсеры цен конкурентов. Существуют специализированные парсеры для интернет-магазинов, которые позволяют отслеживать цены конкурентов на товары. При помощи этих инструментов вы можете указать ссылки на сайты конкурентов, сравнивать их цены с вашими и вносить корректировки при необходимости. К ним относятся Marketparser, Xmldatafeed, ALL RIVAL.

3. Парсеры для быстрого наполнения сайтов. Такие сервисы собирают информацию с сайтов-доноров, включая названия товаров, описания, цены, изображения. Затем эти данные могут быть выгружены в файл или загружены непосредственно на сайт, что значительно ускоряет процесс наполнения сайта и экономит время.

В таких парсерах также есть возможность автоматического добавления наценки (например, при получении данных с оптовыми ценами от поставщика). Кроме того, можно настроить автоматический сбор или обновление данных по расписанию. Примеры таких парсеров: Catalogloader, Xmldatafeed, Диггернаут.

Ниже представлены наиболее популярные парсеры, их основные возможности и функции:

ComparseR. Стоимость: 2000 рублей за лицензию. Доступна ограниченная демоверсия. Этот десктопный парсер позволяет:

1) анализировать технические ошибки на сайте (ошибки 404, дубликаты заголовков, страницы, закрытые от индексации, и т. д.),

2) узнать, на какие страницы обращают внимание поисковые роботы при сканировании сайта.

Анализ сайта от SE Ranking. Это платный облачный сервис с двумя моделями оплаты: ежемесячная подписка или оплата за проверку. Стоимость минимального тарифа — $ 7 в месяц (при оплате годовой подписки). Парсер имеет следующие возможности:

– сканирование всех страниц сайта;

– анализ технических ошибок;

– поиск страниц без мета-тегов title и description, определение страниц со слишком длинными тегами;

– проверка скорости загрузки страниц;

– анализ изображений (поиск неработающих картинок, поиск тяжелых изображений, которые замедляют загрузку страниц);

– анализ внутренних ссылок.

Scraper API. Это сервис, который работает через API, с подробной документацией. Особенности сервиса — автоматическое подставление прокси-адресов и повторение неудачных запросов. Прокси-сервер действует как посредник между клиентом и сервером, перенаправляя запросы и отвечая на них от имени клиента, то есть прокси-адрес — это IP, который скрывает ваш настоящий IP.

Ввод капчи. Капча — это средство защиты сайта от разных автоматических систем, представляет собой простой ребус, который предлагает вам пройти сайт при регистрации или в подтверждение каких-либо действий. Боты не умеют решать такие ребусы. Таким образом, капча нужна, чтобы определить, кто заходит на сайт, человек или робот. Работает через API и требует знания кода. Язык сервиса — английский [4]. В данном исследовании разрабатывается программная реализация парсера для маркетплейса (рис. 1).

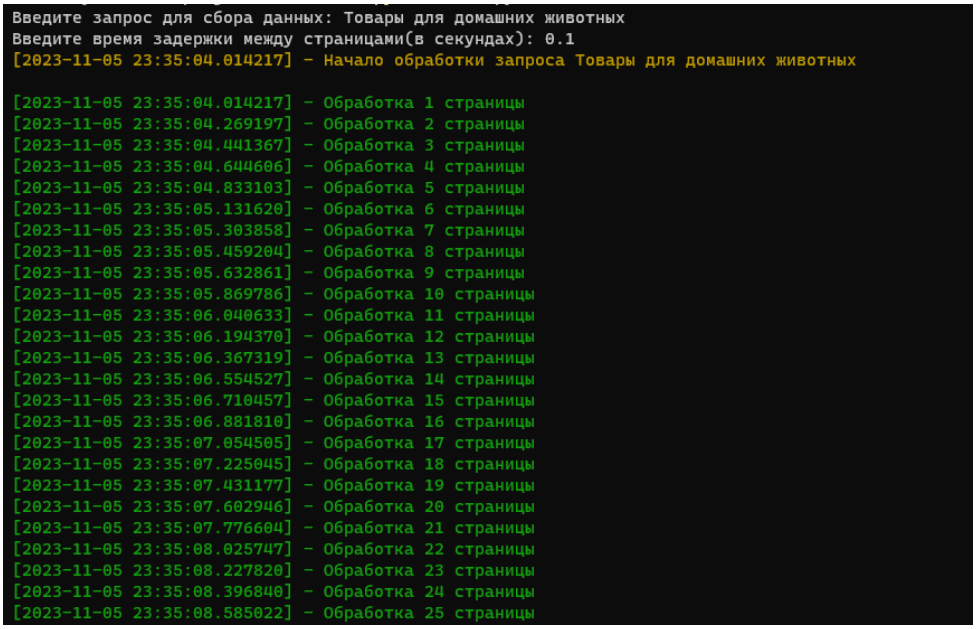

Рис. 1. Ввод данных

Разработка собственного парсера. Во время разработки плана работы парсера было выявлено, что большинство программ сложны в использовании или не обладают должным функционалам. С учетом этих недостатков авторами было создано собственное программное решение. Для его реализации был выбран язык программирования Python и сопутствующие ему библиотеки для сбора и обработки данных.

Техническое задание: программа на вход получает поисковый запрос и время задержки, а на выходе создаёт csv файл в той же директории с результатом поиска (рис. 2).

	A	B	C	D
1	Артикул	Название	Стоимость	Ссылка
2	95224586	Отпугиватель мышей ультразвуковой, тараканов, грызунов, кры	595.0	https://www.wildberries.ru/catalog/95224586/detail.aspx
3	158351869	пуходерка чесалка для кошек и собак дешеддер	244.0	https://www.wildberries.ru/catalog/158351869/detail.aspx
4	91945204	Поглотитель ликвидатор от запаха мочи. Антизапах для лотка	287.0	https://www.wildberries.ru/catalog/91945204/detail.aspx
5	32457988	Средство от запаха кошачьей мочи	327.0	https://www.wildberries.ru/catalog/32457988/detail.aspx
6	70506619	Наполнитель древесный для кошачьего туалета впитывающий	400.0	https://www.wildberries.ru/catalog/70506619/detail.aspx
7	64759981	Витамины для кошек	158.0	https://www.wildberries.ru/catalog/64759981/detail.aspx
8	109583425	Толстовка для собак THE DOG FACE	464.0	https://www.wildberries.ru/catalog/109583425/detail.aspx
9	108689869	Отпугиватель мышей и насекомых ультразвуковой	354.0	https://www.wildberries.ru/catalog/108689869/detail.aspx
10	58472660	Пеленки одноразовые впитывающие 40х60 60х40 60 40 см 30 ш	280.0	https://www.wildberries.ru/catalog/58472660/detail.aspx
11	180601686	Защита от царапания для кошек 200мл	369.0	https://www.wildberries.ru/catalog/180601686/detail.aspx
12	115947803	Шампунь для собак от запаха животных мытья щенков лап эко 1	372.0	https://www.wildberries.ru/catalog/115947803/detail.aspx
13	115706094	Нейтрализатор - ликвидатор запаха меток кошек	425.0	https://www.wildberries.ru/catalog/115706094/detail.aspx
14	146476243	Нейтрализатор запаха животных меток пота мочи Trash Buster	244.0	https://www.wildberries.ru/catalog/146476243/detail.aspx
15	77349906	Игрушка для кошек дразнилка интерактивная	283.0	https://www.wildberries.ru/catalog/77349906/detail.aspx
16	67250089	Гриндер для когтей собак и кошек с подсветкой электрический	1219.0	https://www.wildberries.ru/catalog/67250089/detail.aspx
17	174907035	Игрушки для собак мелких и щенков средних пород	327.0	https://www.wildberries.ru/catalog/174907035/detail.aspx
18	138806778	Прогулочный шар для грызунов хомяков	150.0	https://www.wildberries.ru/catalog/138806778/detail.aspx
19	75989879	Пакеты для уборки за животными 240 шт	266.0	https://www.wildberries.ru/catalog/75989879/detail.aspx
20	160928259	Когтерезка для кошек и собак мелких пород с ограничителем	179.0	https://www.wildberries.ru/catalog/160928259/detail.aspx
21	16443021	Миски для кошек и собак на подставке, 2х240мл	297.0	https://www.wildberries.ru/catalog/16443021/detail.aspx
22	106979816	Лежанка для животных собак и кошек	919.0	https://www.wildberries.ru/catalog/106979816/detail.aspx
23	118177627	Средство для уборки за животными нейтрализатор запаха коше	265.0	https://www.wildberries.ru/catalog/118177627/detail.aspx
24	164912483	Сумка переноска для кошек и собак	1098.0	https://www.wildberries.ru/catalog/164912483/detail.aspx
25	133770789	Средство для уборки за животными нейтрализатор запахов моч	304.0	https://www.wildberries.ru/catalog/133770789/detail.aspx
26	169882140	Клетка для животных большая в квартиру	2397.0	https://www.wildberries.ru/catalog/169882140/detail.aspx

Рис. 2. Вывод данных в формате csv

Преимущества данного парсера:

– простота в использовании;

– возможность задать задержку времени;

– нет ограничений на количество собранных данных;

– удобный формат вывода данных.

Заключение. В результате проведенного сравнительного анализа и классификации инструментов для сбора и анализа данных были выявлены эффективные методы парсинга для решения конкретных задач, что соответствует достижению обозначенной в статье цели.

По итогам проделанной работы можно сделать вывод, что для большинства проектов целесообразнее использовать готовые парсеры. Если проект небольшой, то будет достаточно использовать пробный период у рассмотренных в этой статье сервисов. Однако для крупных проектов, где требуется парсить большие объемы данных и производить сложную обработку, более выгодным будет создание собственного парсера для заданной предметной области.

В результате сравнительного анализа были получены результаты, на основе которых можно сделать следующий прогноз: цифровизация общества ведет к увеличению количества информации и используемых данных, это делает неэффективным создание инструментов для решения только узких задач. Все указывает на то, что парсеры будут изменяться в сторону универсальности и многозадачности.

Список литературы

1. Крамаров С.О., Овсянников В.А., Сахарова Л.В. и др. Автоматизированный сбор данных ключевых финансовых показателей предприятий IT-отрасли региона. *Вестник кибернетики.* 2022;3(47):39–45.

2. Подчернина А.М., Коновалова А.И., Коновалов Е.Ю. и др. Автоматизированный сбор и анализ медицинских статистических данных. *Московская медицина*. 2019;5(33):86–88.

3. Noah Kalson. Сбор данных в 2022 году: все, что вам нужно знать. URL: https://ru-brightdata.com/blog/why-brightdata-ru/web-data-collection-2022 (дата обращения: 20.10.2023)

4. 30+ парсеров для сбора данных с любого сайта. Хабр. URL: https://habr.com/ru/companies/click/articles/494020/ (дата обращения: 25.10.2023).

5. Степная Елена. Как выбрать решение для парсинга сайтов: классификация и большой обзор программ, сервисов и фреймворков. Хабр. URL: https://habr.com/ru/articles/521646/ (дата обращения: 03.11.2023).

Об авторах:

Юлкина Надежда Сергеевна, студентка кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5–7), sunday623zer@gmail.com

Шмелев Михаил Сергеевич, студент кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5–7), mihashmelev2004@gmail.com

Фомина Инга Константиновна, профессор кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5–7), fominga@list.ru

About the Authors:

Nadezhda S. Yulkina, Student of the Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5–7, Dvinskaya Str., St. Petersburg, 198035, RF), sunday623zer@gmail.com

Mikhail S. Shmelev, Student of the Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5–7, Dvinskaya Str., St. Petersburg, 198035, RF), mihashmelev2004@gmail.com

Inga K. Fomina, Professor of the Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5–7, Dvinskaya Str., St. Petersburg, 198035, RF), fominga@list.ru