УДК 004.912:004.032.26
DOI

ТЕХНОЛОГИЯ ПАРСИНГА ДАННЫХ С ПРИМЕНЕНИЕМ НЕЙРОСЕТИ И АЛГОРИТМА WEB-ДРАЙВЕРА

Егармин П.А., Панов Р.Е., Ахматшин Ф.Г., Егармина А.П., Золотухина И.Т.

Лесосибирский филиал Сибирского государственного университета науки и технологий имени академика М.Ф. Решетнева, Лесосибирск, e-mail: egarmi@yandex.ru, jimkrauz@gmail.com, farid.lfsibgtu.ru@mail.ru, alena.egarminamail.ru@gmail.com, irinalsib88@gmail.com

Технология парсинга используется в информационных сервисах или приложениях для автоматического сбора данных из различных источников в Интернете в целях последующей обработки и анализа. Однако в работе традиционных парсеров можно выявить ряд недостатков: сложность работы с сайтами, содержащими динамические элементы; ограничения в извлечении информации с последующим структурированием данных. Технология парсинга данных с использованием нейросети и веб-драйвера может решить указанные проблемы. Применение веб-драйвера позволит корректно обращаться с динамическими элементами веб-страниц и получать актуальные данные после их загрузки. Обращение к нейросети во время парсинга может улучшить точность извлечения данных из нерегулярных и сложных страниц, структурировать информацию, полученную от первичного парсинга. Разработанное на основе технологии приложение может быть адаптировано под решение задач в различных предметных областях: службы занятости населения (поиск актуальных вакансий с учетом имеющихся компетенций), приемные комиссии вузов (поиск потенциальных абитуриентов), риелторские компании (парсинг сообществ социальных сетей и сайтов по продаже недвижимости с целью изучения рынка недвижимости и последующего формирования объявлений о продаже недвижимости).

Ключевые слова: парсинг, нейросеть, web-драйвер, классификация данных

DATA PARSING TECHNOLOGY USING A NEURAL NETWORK AND A WEB DRIVER ALGORITHM

Egarmin P.A., Panov R.E., Akhmatshin F.G., Egarmina A.P., Zolotuhina I.T.

Lesosibirsk Branch of Reshetnev Siberian State University of Science and Technology, Lesosibirsk, e-mail: egarmi@yandex.ru, jimkrauz@gmail.com, farid.lfsibgtu.ru@mail.ru, alena.egarminamail.ru@gmail.com, irinalsib88@gmail.com

Parsing technology is used in information services or applications to automatically collect data from various sources on the Internet for subsequent processing and analysis. However, a number of disadvantages can be identified in the work of traditional parsers: the complexity of working with sites containing dynamic elements; limitations in extracting information with subsequent data structuring. Data parsing technology using a neural network and a web driver can solve these problems. Using a web driver will allow you to correctly handle dynamic elements of web pages and get up-to-date data after they are loaded. Accessing the neural network during parsing can improve the accuracy of data extraction from irregular and complex pages, and structure the information obtained from primary parsing. The application developed on the basis of technology can be adapted to solve problems in various subject areas: employment services (search for relevant vacancies, taking into account existing competencies), university admissions committees (search for potential applicants), real estate companies (parsing communities of social networks and real estate sites in order to study the real estate market and the subsequent formation of announcements about real estate sales).

Keywords: parsing, neural network, web driver, data classification

В настоящее время получение и анализ данных с веб-страниц являются неотъемлемой частью многих задач в области информационных технологий и бизнеса: финансовая аналитика (изучение котировок акций, новостей о компаниях, данных отчетности), академические исследования (получение данных опубликованных исследований, связанных с определенной тематикой), социальные медиа (изучение новостей, блогов, интернет-рекламы) [1–4].

Однако стандартные методы парсинга, например использование регулярных выражений или HTML-верстки кода страниц [5], могут быть недостаточно эффективными для решения следующих задач:

− получение данных с динамических веб-страниц. Большинство веб-страниц меняют свое содержимое с течением времени или в результате действий пользователя. При изменении содержимого страницы меняется и ее код. Это приводит к тому, что парсер «не видит» изменения и пропускает новые данные;

− классификация полученных данных. Классические парсеры работают на основе определенных форматов данных, что приводит к ограничениям в извлечении инфор-

мации и последующем структурировании данных.

Технология парсинга с использованием нейросети и веб-драйвера позволит:

− повысить точность сбора данных за счет анализа динамически загружаемых элементов. Веб-драйвер способен эмулировать действия пользователя на веб-страницах (заполнение форм, щелчки по элементам, прокрутка страницы), тем самым предоставляя доступ к данным, не доступным через обычный HTTP-запрос;

− выполнить обработку данных сложной структуры. Парсинг, сочетаемый с возможностями нейросетей [6], может быть использован для поиска изображений, глубокого анализа текстов, записанных на естественном языке. Нейросеть способна помочь в извлечении значимых признаков собранных данных.

Цель исследования – создание десктопного приложения, осуществляющего парсинг данных с применением нейросети и алгоритма web-драйвера.

Задачи работы:

1) разработка алгоритма работы парсера;

2) проектирование архитектуры приложения;

3) реализация парсера на платформе .NET;

4) интеграция веб-драйвера в программный алгоритм парсера;

5) интеграция нейросети в программный алгоритм парсера;

6) разработка модуля фильтрации данных;

7) проектирование пользовательского интерфейса;

8) тестирование и внедрение приложения в риелторскую организацию.

Материал и методы исследования

Для реализации технологии были выбраны язык программирования C#, набор библиотек Selenium WebDriver и модель нейросети OpenAI ChatGPT 3.5 Turbo (табл. 1).

Selenium WebDriver – набор программных библиотек для автоматизации действий браузера, включающий:

− функции для работы с динамическими сайтами с технологиями AJAX, JavaScript;

− поддержку браузеров: Google Chrome, Firefox, Safari, Microsoft Edge, Opera и др.;

− простой и понятный API для автоматизации действий браузера;

− поддержку современных языков программирования, таких как Java, C#, Python, JavaScript.

Выбор ChatGPT 3.5 Turbo обусловлен следующими причинами:

− модель является одной из последних версий GPT от OpenAI, способной генерировать качественные и связные тексты. Все это позволяет использовать ее для обработки и генерации текстовых данных, полученных парсером [7];

− модель основана на непрерывном обучении от OpenAI, то есть продолжает улучшаться и получать обновления [8].

В качестве языка программирования выбран объектно-ориентированный язык C# платформы .NET, включающий:

− интеграцию с .NET Framework и .NET Core;

− набор библиотек, которые обеспечивают работу с веб-сервисами, обработку JSON, работу с базами данных.

Результаты исследования и их обсуждении

Технология парсинга данных с использованием нейросети и веб-драйвера была успешно использована при создании парсера для агентства недвижимости. Основные возможности парсера:

− поиск информации в объявлениях, размещенных в группах социальных сетей, по ключевым словам («продам», «сдам», «аренда», «квартира», «комната», «дом», «участок», «дача», «гараж»), по теме объявления (недвижимость);

− сохранение найденной информации в виде текстовых файлов следующей структуры: описание объявления, прикрепленные к объявлению изображения;

− определение ключевых характеристик объявления: тип жилья, количество комнат, наличие изображений;

− группировка файлов по ключевым характеристикам объявления.

Таблица 1

Инструменты, используемые для реализации технологии

Наименование инструмента	Тип программного обеспечения	Возможности
Selenium WebDriver	Бесплатный	Широкая поддержка браузеров и языков программирования
OpenAI ChatGPT 3.5 Turbo	Условно бесплатный	Генерация и классификация текста
C#	Бесплатный	Широкий набор библиотек по работе с веб-сервисами и обработке данных

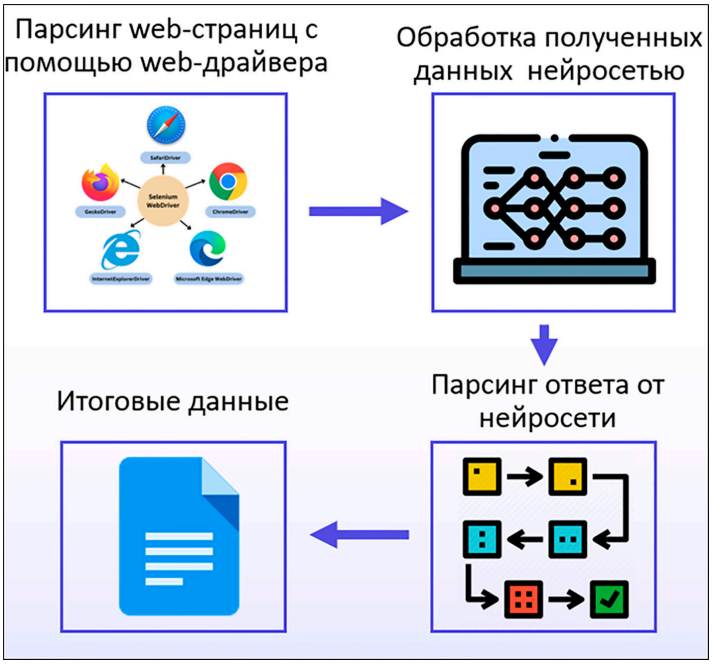

Рис. 1. Алгоритм работы парсера с веб-драйвером и нейросетью

На рисунке 1 представлен алгоритм работы парсера.

На первом этапе веб-драйвер открывает целевую веб-страницу, эмулирует действия пользователя, загружает все данные с веб-страницы, включая изображения и текстовое описание. Кроме того, веб-драйвер позволяет получать данные, обращаясь к HTML-элементам с помощью CSS-селекторов, XPath-запросов или названий классов HTML-элементов (рис. 2). За счет этого происходит возникновение событий в коде страницы (рис. 3), предоставляющих доступ к данным, недоступным при классическом парсинге.

Далее, на основе полученных данных, формируются запросы к нейросети, которая анализирует и классифицирует их по ключевым характеристикам – типу жилья, количеству комнат. Запросы представляют собой наборы данных, состоящих из следующих пар: «входные данные HTML-кода» и «ожидаемый результат» или «полученные данные из первичного HTML-парсинга» и «ожидаемый результат». Модель анализирует загруженный HTML-код или результат первичного парсинга и находит соответствующие элементы, которые нужно извлечь (рис. 4).

Заключительным этапом является процесс вторичного парсинга. На данном этапе происходят обращение к нейросети и извлечение данных, которые она смогла сконфигурировать. Извлечение происходит в более удобном формате, так как в данном случае парсер обращается к специальным тегам (классификаторам), заданным нейросетью, и получает структурированные данные (рис. 5). Результаты вторичного парсинга позволяют организовать хранение объявлений с группировкой по ключевым признакам.

```
// initializing list of components with text using web river variable
var topicElements = _driver.FindElements(By.ClassName("media-
text_cnt_tx"));
foreach (var topicElement in topicElements)
{
   // creating a list of links to ads on social networks
   IWebElement linkElement =
topicElement.FindElement(By.ClassName("media-text_a"));
   topicUrls.Add(linkElement.GetAttribute("href"));
}
```

Рис. 2. Фрагмент кода с извлечением текстового содержания объявления

```
private void ScrollToTheBottom()
{
// Creating an object to execute JavaScript using a web driver
    IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)_driver;
// Saving the page scroll height value
    long scrollHeight = (long)jsExecutor.ExecuteScript("return
Math.max(document.documentElement.scrollHeight,
document.body.scrollHeight);");
// Saving the page scroll height value
    while (true)
    {
        jsExecutor.ExecuteScript("window.scrollTo(0,
document.documentElement.scrollHeight);");
// Updating the scroll height
        Thread.Sleep(2000);
        long newScrollHeight = (long)jsExecutor.ExecuteScript("return
Math.max(document.documentElement.scrollHeight,
document.body.scrollHeight);");
// Checking to reach the end of the page
        if (newScrollHeight == scrollHeight) break;
        scrollHeight = newScrollHeight;
    }
    Thread.Sleep(2000);
}
```

Рис. 3. Метод, содержащий логику работы веб-драйвера при листинге страницы

```
// Creating a list of identifiers based on regular expressions
private List<Tuple<string, string>> typePatterns = new List<Tuple<string,
string>>()
{
    Tuple.Create("_Cottage",@"[Cc]ottage"),
private string ExtractType(string text)
{
    for (int i = 0; i < typePatterns.Count; i++)
    {
// Checking whether a line of text matches a regular expression
        Match match = Regex.Match(text, typePatterns[i].Item2,
RegexOptions.IgnoreCase);
        if (match.Success)
        {
            string result = typePatterns[i].Item1;
            if (result.Contains('_'))
            {
                result = result.Replace("_", "");
                return result;
            }
            else return "Detected";
        }
    }
    return "Not detected";
}
```

Рис. 4. Фрагмент кода для определения типа жилья в объявлении

Созданный парсер способен эффективно работать с фильтрами, извлекать и классифицировать данные из веб-страниц, в которых не предусмотрена строгая организация информации: блоги, социальные сети, чаты, сайты с объявлениями. Помимо агентств недвижимости, парсер может быть адаптирован к решению задач и в других областях, например в службах занятости населения, приемных комиссиях вузов.

```
// Connecting a neural network
var RequestData = new Request()
{
    ModelId = "gpt-3.5-turbo",
    Messages = messages
};
// Executing an HTTP POST request
using var response = await httpClient.PostAsJsonAsync(endpoint,
requestData);
// Reading the response from the neural network in JSON format
ResponseData? responseData = await
response.Content.ReadFromJsonAsync<ResponseData>();
// Determining property values Housing type and Number of rooms
// Return of the Ad object
public Advertisement CreateAdvertisement(string neuroformattedData)
{
    ad.HousingType = ExtractType(neuroformattedData);
    ad.RoomCount = ExtractType(neuroformattedData); return ad;
}
```

Рис. 5. Вторичный парсинг данных

Заключение

Описанный метод парсинга имеет ряд преимуществ:

– улучшенная обработка и анализ данных: обращение парсера к нейросети позволит распознавать и классифицировать текстовую и визуальную информацию, более точно анализировать и обрабатывать данные;

– расширенная функциональность: использование нейросети даст возможность расширить функциональность парсера, например выполнять семантический анализ текстов;

– повышение точности и надежности парсера: использование нейросети позволит улучшить точность извлечения данных из нерегулярных и сложных страниц, таких как веб-форумы или блоги, где структура данных менее предсказуема;

– работа с динамическими сайтами: если страница содержит динамические элементы, то использование веб-драйвера позволит более корректно обращаться с такими элементами и получать актуальные данные после их загрузки. Это особенно важно при парсинге страниц, при работе с которыми нужно ожидать загрузки элементов или выполнения определенных действий.

Работа выполнена при поддержке Краевого государственного автономного учреждения «Красноярский краевой фонд поддержки научной и научно-технической деятельности».

Список литературы

1. Ермоленко А.В., Котелина Н.О., Старцева Е.Н., Юркина М.Н. О востребованности подготовки в области парсинга данных для web-разработчиков // Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика. 2021. № 1 (38). С. 56-69.

2. Меньшиков Я.С. Преимущества автоматического сбора данных в сети интернет над ручным сбором данных // Universum: технические науки. 2022. № 10 (103). URL: https://7universum.com/ru/tech/archive/item/14383 (дата обращения: 05.03.2024).

3. Суханов А.А., Маратканов А.С. Анализ способов сбора социальных данных из сети Интернет // International scientific review. 2017. № 1 (32). С. 25-28.

4. Крамаров С.О., Овсянников В.А., Сахарова Л.В., Усатый Р.С., Лукьянова Г.В. Автоматизированный сбор данных ключевых финансовых показателей предприятий IT-отрасли региона // Вестник кибернетики. 2022. № 3 (47). С. 39-45. DOI: 10.34822/1999-7604-2022-3-39-45.

5. Angelo Borsotti, Luca Breveglieri, Stefano Crespi Reghizzi, Angelo C. Morzenti General parsing with regular expression matching // Journal of Computer Languages. 2022. Vol. 2.2. DOI: 10.1016/j.cola.2022.101176.

6. Mengtian Yin, Llewellyn Tang, Chris Webster, Jinyang Li, Haotian Li, Zhuoquan Wu, Reynold C.K. Cheng Two-stage Text-to-BIMQL semantic parsing for building information model extraction using graph neural networks // Automation in Construction. 2023. Vol. 152. DOI: 10.1016/j.autcon.2023.104902.

7. Ruopeng An, Yuyi Yang, Fan Yang, Shanshan Wang Mlwa Use prompt to differentiate text generated by ChatGPT and humans // Machine Learning with Applications. 2023. Vol. 14. DOI: 10.1016/j.mlwa.2023.100497.

8. Biao Zhao, Weigiang Jin, Javier Del Ser, Guang Yang Neucom Exploring potentials of ChatGPT on cross-linguistic agricultural text classification // Neurocomputing. 2023. Vol. 557. DOI: 10.1016/j.neucom.2023.126708.