УДК 622

ПРИМЕНЕНИЕ ПАРСИНГА САЙТОВ В SEO С ИСПОЛЬЗОВАНИЕМ PYTHON

К.В. Иушина, студент гр. ПИм-231, II курс
Научный руководитель – А.А. Тайлакова, к.т.н., доцент
Кузбасский государственный технический университет имени Т. Ф. Горбачева, г. Кемерово

В современном мире оптимизация веб-ресурсов играет ключевую роль в обеспечении их высокой видимости в поисковых системах. Для эффективного продвижения сайтов SEO-специалистам необходимо анализировать их структуру, мета-данные, заголовки, канонические ссылки и другие параметры, влияющие на ранжирование. Однако ручной сбор такой информации может быть трудоемким и неэффективным, особенно при работе с большим объемом информации.

Одним из способов такого процесса является парсинг (parsing) — метод автоматического анализа веб-страниц для извлечения структурированной информации. Этот процесс, также известный как веб-скрапинг, позволяет значительно упростить сбор показателей. Например, вместо того чтобы вручную искать и копировать заголовки страниц и ссылки, можно использовать программный скрипт, который выполнит эту задачу автоматически и сохранит полученные данные в удобном формате.

На данный момент существует множество программ для парсинга, но они ограничены лимитами. Например, Screaming Frog SEO Spider - это программное обеспечение, которое сканирует сайты, анализирует мета-данные, статус индексации, ошибки 404, редиректы и многое другое. Но в бесплатной версии, она анализирует только 500 URL, а платный тариф стоит от 259 долларов в год.

Классические методы требуют значительных временных затрат и усилий, что вызывает необходимость в автоматизации процесса извлечения информации, сохранении ее в структурированном формате и проведении анализа на основе этих данных.

Предложенная методика основывается на использовании Python с библиотеками requests, BeautifulSoup и csv.

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin
import csv
```

1. Функция get_page_info получает необходимые данные с заданного URL-адреса:

- Заголовок страницы извлекается из тега <title>.
- Мета-описание извлекается из тегов <meta> с атрибутом name="description".
- Заголовки от H1 до H6 извлекаются с помощью цикла и soup.find_all().
- Канонический URL проверяется через <link rel="canonical">.
- Установка флага noindex осуществляется на основе наличия соответствующего текста на странице.

```
try:
   response = requests.get(url, timeout=5)
   soup = BeautifulSoup(response.text, 'lxml')
   title = soup.title.string if soup.title else ''
   description = soup.find("meta", attrs={"name":
"description"})
   description = description["content"] if description
else ''
   h_tags = {f"H{level}": [h.get_text(strip=True) for
h in soup.find_all(f"h{level}")] for level in range(1,
7)}
   canonical = soup.find("link", rel="canonical")
   canonical = canonical["href"] if canonical else ''
   noindex = "noindex" in response.text.lower()
   page_format = response.headers.get("Content-Type",
"").split(";")[0]
```

2. Функция get_all_links находит все внутренние ссылки на заданном сайте и извлекает их. Она использует обход в глубину (DFS) для посещения каждого URL. Каждая страница анализируется на наличие ссылок. Непосещенные ссылки добавляются в очередь на дальнейшее изучение.

```
    visited = set()
    to_visit = {site_url}
    all_links = set()
    while to_visit:
       url = to_visit.pop()
       if url in visited:
           continue
       visited.add(url)
       try:
           response = requests.get(url, timeout=5)
           soup = BeautifulSoup(response.text, 'lxml')
```

```
            for link in soup.find_all("a", href=True):
                full_url = urljoin(url, link["href"])
                if site_url in full_url and full_url not
in visited:
                    to_visit.add(full_url)
                    all_links.add(full_url)
```

3. Основная функция main собирает ссылки, получает информацию о страницах и сохраняет результаты в CSV-файл.

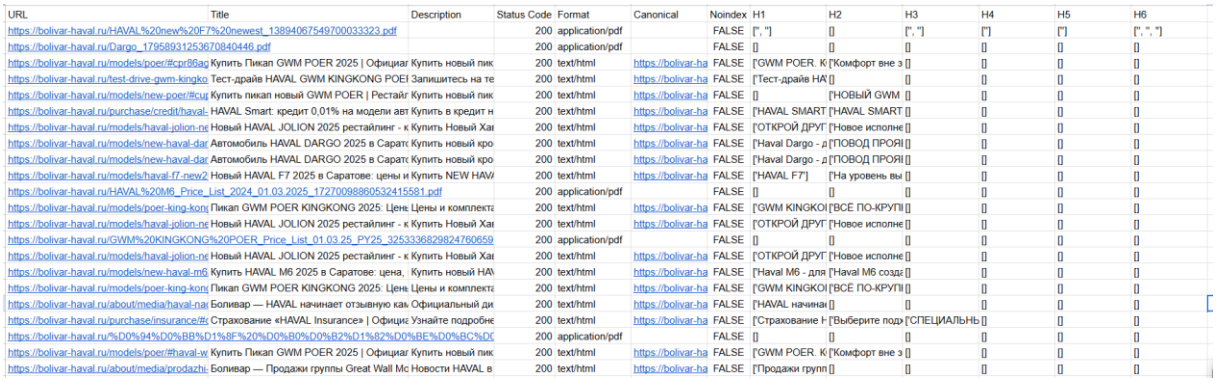
```
    all_urls = get_all_links(site_url)
    data = []
    for url in all_urls:
        info = get_page_info(url)
        if info:
            data.append(info)

    with    open("site_data.csv",    "w",    newline="",
encoding="utf-8") as csvfile:
        fieldnames  =  ["URL",  "Title",  "Description",
"Status Code", "Format", "Canonical", "Noindex", "H1",
"H2", "H3",
                       "H4", "H5", "H6"]
        writer          =          csv.DictWriter(csvfile,
fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(data)
```

Для запуска программы, необходимо ввести URL сайта, который нужно проанализировать (рис. 1).

```
Введите URL сайта:
```

Рисунок 1 - Запуск программы.

После выполнения кода, результаты сохраняются в файл site_data.csv, который содержит информацию о сайте (рис. 2).



URL	Title	Description	Status Code	Format	Canonical	Noindex	H1	H2	H3	H4	H5	H6
https://bolivar-haval.ru/HAVAL%20new%20F7%20newest_13894067549700033323.pdf			200	application/pdf		FALSE	['', '']	[]	['', '']	['']	['']	['', '', '']
https://bolivar-haval.ru/Dargo_17958931253670840446.pdf			200	application/pdf		FALSE	[]	[]	[]	[]	[]	[]
https://bolivar-haval.ru/models/poer/#cpr86ag	Купить Пикап GWM POER 2025 | Официаг	Купить новый пик	200	text/html	https://bolivar-ha	FALSE	['GWM POER. К	['Комфорт вне з	[]	[]	[]	[]
https://bolivar-haval.ru/test-drive-gwm-kingko	Тест-драйв HAVAL GWM KINGKONG POEI	Запишитесь на те	200	text/html	https://bolivar-ha	FALSE	['Тест-драйв HA	[]	[]	[]	[]	[]
https://bolivar-haval.ru/models/new-poer/#cuj	Купить пикап новый GWM POER | Рестайл	Купить новый пик	200	text/html	https://bolivar-ha	FALSE	[]	['НОВЫЙ GWM	[]	[]	[]	[]
https://bolivar-haval.ru/purchase/credit/haval-	HAVAL Smart: кредит 0,01% на модели авт	Купить в кредит н	200	text/html	https://bolivar-ha	FALSE	['HAVAL SMART	['HAVAL SMART	[]	[]	[]	[]
https://bolivar-haval.ru/models/haval-jolion-ne	Новый HAVAL JOLION 2025 рестайлинг - к	Купить Новый Хав	200	text/html	https://bolivar-ha	FALSE	['ОТКРОЙ ДРУГ	['Новое исполне	[]	[]	[]	[]
https://bolivar-haval.ru/models/new-haval-dar	Автомобиль HAVAL DARGO 2025 в Сарат	Купить новый кро	200	text/html	https://bolivar-ha	FALSE	['Haval Dargo - д	['ПОВОД ПРОЯВ	[]	[]	[]	[]
https://bolivar-haval.ru/models/new-haval-dar	Автомобиль HAVAL DARGO 2025 в Сарат	Купить новый кро	200	text/html	https://bolivar-ha	FALSE	['Haval Dargo - д	['ПОВОД ПРОЯВ	[]	[]	[]	[]
https://bolivar-haval.ru/models/haval-f7-new2	Новый HAVAL F7 2025 в Саратове: цены и	Купить NEW HAV	200	text/html	https://bolivar-ha	FALSE	['HAVAL F7']	['На уровень вы	[]	[]	[]	[]
https://bolivar-haval.ru/HAVAL%20M6_Price_List_2024_01.03.2025_1727009886053241558 1.pdf			200	application/pdf		FALSE	[]	[]	[]	[]	[]	[]
https://bolivar-haval.ru/models/poer-king-kon	Пикап GWM POER KINGKONG 2025: Цен	Цены и комплекта	200	text/html	https://bolivar-ha	FALSE	['GWM KINGKO	['ВСЁ ПО-КРУП	[]	[]	[]	[]
https://bolivar-haval.ru/models/haval-jolion-ne	Новый HAVAL JOLION 2025 рестайлинг - к	Купить Новый Хав	200	text/html	https://bolivar-ha	FALSE	['ОТКРОЙ ДРУГ	['Новое исполне	[]	[]	[]	[]
https://bolivar-haval.ru/GWM%20KINGKONG%20POER_Price_List_01.03.25_PY25_3253336829824760659			200	application/pdf		FALSE	[]	[]	[]	[]	[]	[]
https://bolivar-haval.ru/models/haval-jolion-ne	Новый HAVAL JOLION 2025 рестайлинг - к	Купить Новый Хав	200	text/html	https://bolivar-ha	FALSE	['ОТКРОЙ ДРУГ	['Новое исполне	[]	[]	[]	[]
https://bolivar-haval.ru/models/new-haval-m6	Купить HAVAL M6 2025 в Саратове: цена,	Купить новый HAV	200	text/html	https://bolivar-ha	FALSE	['Haval M6 - для	['Haval M6 созда	[]	[]	[]	[]
https://bolivar-haval.ru/models/poer-king-kon	Пикап GWM POER KINGKONG 2025: Цен	Цены и комплекта	200	text/html	https://bolivar-ha	FALSE	['GWM KINGKO	['ВСЁ ПО-КРУП	[]	[]	[]	[]
https://bolivar-haval.ru/about/media/haval-nar	Боливар — HAVAL начинает отзывную кам	Официальный ди	200	text/html	https://bolivar-ha	FALSE	['HAVAL начина	[]	[]	[]	[]	[]
https://bolivar-haval.ru/purchase/insurance/#(	Страхование «HAVAL Insurance» | Официа	Узнайте подробне	200	text/html	https://bolivar-ha	FALSE	['Страхование H	['Выберите под	['СПЕЦИАЛЬНЫ	[]	[]	[]
https://bolivar-haval.ru/%D0%94%D0%BB%D1%8F%20%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0			200	application/pdf		FALSE	[]	[]	[]	[]	[]	[]
https://bolivar-haval.ru/models/poer/#haval-w	Купить Пикап GWM POER 2025 | Официаг	Купить новый пик	200	text/html	https://bolivar-ha	FALSE	['GWM POER. К	['Комфорт вне з	[]	[]	[]	[]
https://bolivar-haval.ru/about/media/prodazhi-	Боливар — Продажи группы Great Wall Mc	Новости HAVAL в	200	text/html	https://bolivar-ha	FALSE	['Продажи групп	[]	[]	[]	[]	[]

Рисунок 2 - Данные файла site_data.csv.

В заключении хочется отметить, что автоматический парсинг сайта позволяет существенно сократить время на анализ структуры страниц, выявление ошибок в мета-данных и оптимизацию контента. Такой подход облегчает работу, позволяя оперативно получать актуальную информацию о сайте. Это мощный инструмент для анализа и оптимизации сайтов, который позволяет собирать и анализировать информацию в удобном формате.

Таким образом, автоматизация процесса сбора SEO-данных с помощью Python и его библиотек значительно упрощает анализ веб-ресурсов, сокращая время на ручной обход. Использование парсинга позволяет быстро извлекать ключевую информацию о страницах, такую как заголовки, мета-описания, канонические ссылки и другие важные параметры, влияющие на ранжирование в поисковых системах. Это особенно актуально для крупных сайтов, где ручной анализ становится неэффективным. Разработанный скрипт предоставляет удобный и доступный инструмент для SEO-специалистов, помогая им в оптимизации веб-ресурсов и повышении их конкурентоспособности в цифровой среде.

Список литературы

1. Парсинг сайтов на Python: для чего нужен и как написать скрипт [Электронный ресурс]. – Режим доступа: https://blog.skillfactory.ru/parsing-saytov-na-python/
2. Парсинг [Электронный ресурс]. – Режим доступа: https://www.unisender.com/ru/glossary/chto-takoe-parsing/
3. The Python Standard Library [Электронный ресурс]. – Режим доступа: https://docs.python.org/3/library/index.html
4. Парсинг сайтов. «SITEANALYZER» как инструмент для проведения технического SEO- аудита сайтов [Электронный ресурс]. – Режим доступа: https://libeldoc.bsuir.by/bitstream/123456789/40206/1/Sitnik_Parsing.pdf