DOI: 10.15514/ISPRAS-2024-36(5)-11

Автоматическое построение правил извлечения информации для новостных веб-сайтов

[1] *С.С. Дубовицкий*, ORCID: 0009-0001-1907-8030 <*s.dub@ispras.ru*>
[1, 2] *П.А. Бедрин,* ORCID: 0009-0008-7523-8298 <*pbedrin@ispras.ru*>
[1, 2] *А.К. Яцков,* ORCID: 0000-0002-1312-1675 <*yatskov@ispras.ru*>
[1] *М.И. Варламов,* ORCID: 0000-0002-1083-6210 <*varlamov@ispras.ru*>

[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
[2] *Московский государственный университет имени М.В. Ломоносова,*
Россия, 119991, Москва, Ленинские горы, д. 1.

Аннотация. В данной работе представлен метод автоматической генерации правил извлечения информации (карт сбора) для новостных веб-сайтов. Данный подход по набору новостных страниц одного сайта генерирует карту сбора, позволяющую извлекать атрибуты из произвольных новостных страниц этого сайта. В основе метода лежит применение дообученной нейросетевой модели MarkupLM для извлечения информации из веб-страниц. Предложенный метод обобщает предсказания модели на уровне сайта, создавая универсальные правила извлечения атрибутов. Проведённые эксперименты показали, что использование карт сбора, сформированных на основе дообученной модели, превосходит по качеству как существующие открытые инструменты, так и дообученный MarkupLM на уровне отдельных страниц. Разработанный метод может быть обобщён на другие предметные области при наличии релевантных данных для дообучения модели.

Ключевые слова: извлечение информации; сбор данных из глобальной сети; новостные веб-сайты; нейронные сети.

Для цитирования: Дубовицкий С.С., Бедрин П.А., Яцков А.К., Варламов М.И. Автоматическое построение правил извлечения информации для новостных веб-сайтов. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 153–162. DOI: 10.15514/ISPRAS–2024–36(5)–11.

Благодарности: Авторы статьи выражают благодарность Институту системного программирования им. В.П. Иванникова Российской академии наук за поддержку выполняемых исследований.

Automatic Construction of Information Extraction Rules for News Websites

[1] *S.S. Dubovitskii*, ORCID: 0009-0001-1907-8030 <*s.dub@ispras.ru*>
[1, 2] *P.A. Bedrin*, ORCID: 0009-0008-7523-8298 <*pbedrin@ispras.ru*>
[1, 2] *A.K. Yatskov*, ORCID: 0000-0002-1312-1675 <*yatskov@ispras.ru*>
[1] *M.I. Varlamov*, ORCID: 0000-0002-1083-6210 <*varlamov@ispras.ru*>

[1] *Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russia.

Abstract. This paper presents a method for the automatic generation of information extraction rules (sitemaps) for news websites. The proposed approach generates a sitemap based on a set of news pages from a single site, enabling attribute extraction from arbitrary news pages on that site. The method is based on applying a fine-tuned neural network model, MarkupLM, to extract information from web pages. This approach generalizes the model's predictions at the site level, creating universal rules for attribute extraction. Experimental results show that using sitemaps generated with the fine-tuned model surpasses both existing open-source tools and the fine-tuned MarkupLM applied at the individual page level. The developed method can be extended to other domains if relevant data for model fine-tuning is available.

Keywords: information extraction; web scraping; news websites; neural networks.

For citation: Dubovitskii S.S., Bedrin P.A., Yatskov A.K., Varlamov M.I. Automatic construction of information extraction rules for news websites. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 153-162 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-11.

Acknowledgements. The authors are grateful to the Ivannikov Institute for System Programming of the Russian Academy of Sciences.

1. Введение

Интернет – это обширное хранилище информации, объём которого непрерывно увеличивается. В связи с этим задача эффективного сбора данных из сети приобретает всё более важное значение [1]. Процесс сбора подразумевает использование специализированных инструментов и методов для извлечения релевантной информации из веб-сайтов различных предметных областей.

Так, маркетинговые и социальные исследования в значительной степени опираются на данные, собранные в Интернете. Анализ пользовательской активности в социальных сетях позволяет отслеживать настроения аудитории, её потребности и предпочтения.

Сбор данных из интернет-магазинов является важным инструментом для исследования рынка и принятия бизнес-решений. Он помогает решать такие задачи, как мониторинг цен и ассортимента, анализ пользовательских отзывов и оценка популярности товаров. Эти данные используются для определения конкурентной среды, прогнозирования спроса и разработки персонализированных рекомендаций для пользователей. Информация, полученная в результате сбора, помогает бизнесу оперативно реагировать на изменения на рынке, оптимизировать ценообразование и разрабатывать более точные маркетинговые стратегии.

Другой важной сферой применения сбора данных является мониторинг новостей. Он необходим для анализа новостной повестки и выявления текущих тенденций в информационном поле. Новостные страницы представлены на сайтах различных категорий, включая информационные агентства, телеканалы, газеты, корпоративные ресурсы и порталы государственных органов.

Таким образом, сбор данных из Интернета предоставляет возможность получить большие объёмы информации из Интернета для дальнейшего анализа. В данной работе мы сосредоточимся на сборе данных из новостных сайтов.

Для сбора данных используются сборщики. Сборщик данных из новостных сайтов – это программное обеспечение или скрипт, предназначенный для автоматического обхода сайта и извлечения структурированных данных из страниц с новостными публикациями: заголовков, текстов, авторов, дат публикации и обновления, тегов и других атрибутов. Сборщик принимает на вход URL-адрес новостного сайта, обходит сайт и извлекает атрибуты из новостных страниц, которые затем могут быть сохранены в базу данных, файл или другую систему хранения для последующего использования.

Среди существующих подходов к созданию сборщиков можно выделить:

- Программирование сборщика для каждого ресурса. Этот подход требует наличия специалиста, который сможет исследовать HTML-код страниц ресурса или запросы, которые отправляются со страниц к серверу, и составить правила извлечения нужной информации. Этот процесс может занимать существенное время. Тем не менее, такие сборщики обычно позволяют обеспечить высокое качество данных, поскольку они тщательно адаптированы к определенному ресурсу.
- Инструменты визуальной разметки (Octoparse [2], Web Scraper [3] и другие). Такие инструменты предоставляют графический интерфейс, с помощью которого пользователь может в интерактивном режиме указать элементы страницы, содержащие интересующие его данные. Система сама составит правила для их извлечения. Этот подход снижает требования к специалисту и позволяет быстрее создавать и редактировать сборщики для сайтов в больших объёмах. Тем не менее, для каждого сайта всё равно вручную создаётся сборщик.
- Использование автоматических сборщиков. Заранее запрограммированные сборщики применимы в большом числе ситуаций, но имеют проблемы с производительностью и качеством сбора данных. Они также могут не поддерживать определённые ресурсы из-за их уникальных особенностей, и в них отсутствует возможность редактирования алгоритма работы.

Частью работы сборщика является обход сайта по какому-либо алгоритму: рекурсивно по всем ссылкам, с использованием RSS-ленты или карты сайта (sitemap.xml). Далее выполняется классификация страниц: на основе HTML-кода сборщик определяет, являются ли они новостными. Для новостных страниц затем применяется один из инструментов извлечения.

Для автоматического извлечения атрибутов новостных страниц используются различные методы. Это могут быть как готовые библиотеки, например, trafilatura [4, 5] и newsplease [6, 7], так и инструменты, основанные на нейронных сетях (MarkupLM [8, 9], WIERT [10], SimpDOM [11] и другие).

Недостатком существующих автоматических сборщиков является то, что при извлечении новостные страницы обрабатываются независимо. Это может приводить к нестабильной работе сборщика внутри одного сайта. Если бы мы могли автоматически генерировать общие правила извлечения на уровне сайта, сборщики было бы проще проверять и редактировать, они бы показывали более высокое и стабильное качество извлечения.

Таким образом, несмотря на разнообразие имеющихся подходов и инструментов, проблема гибкости и универсальности сборщиков остаётся актуальной. Существующие решения либо слишком общие и не обеспечивают необходимого качества данных в конкретных случаях, либо требуют значительных ресурсов или времени для разработки. Хотя ручное создание сборщиков для отдельных сайтов с использованием инструментов, таких как Beautiful Soup

[12], может показаться быстрым и эффективным решением, при масштабировании на сотни или тысячи сайтов оно становится значительно более трудоёмким и ресурсоёмким.

В этой связи возникает потребность в разработке нового алгоритма, который бы объединил преимущества существующих подходов, минимизировав их недостатки. Полностью автоматические инструменты, хотя и обладают способностью обрабатывать произвольные сайты, часто не могут достичь желаемого уровня качества из-за обобщённости алгоритмов. Существует явная потребность в разработке решения, которое бы сочетало масштабируемость автоматических методов с качеством и адаптивностью ручного подхода.

В этой работе мы будем рассматривать сборщики новостных сайтов в виде карт сбора. Карта сбора состоит из правил извлечения атрибутов новостной страницы. Правило извлечения состоит из CSS-селектора [13] и флага множественности (нужно извлечь одно значение или список). Не ограничивая общности, рассматриваются только правила, извлекающие значения из текста страницы.

Гипотеза исследования заключается в том, что можно разработать алгоритм, который, используя инструмент для извлечения атрибутов новостей на основе нейросетевых моделей, позволит автоматически создавать сборщики данных для новых новостных сайтов, которые были бы сравнимы по качеству извлечения информации с существующими подходами. При этом такие сборщики обеспечат более быстрое извлечение информации, и их можно будет изменять после создания.

2. Постановка задачи

Цель исследования — разработка метода, способного по набору HTML страниц с новостными статьями с одного сайта сгенерировать карту сбора, которая позволит извлекать данные из любой новостной страницы на этом сайте. Карта сбора сайта должна содержать правила для извлечения следующих атрибутов: заголовок, подзаголовок, дата публикации, текст, авторы, категории и теги. Для создания карты могут использоваться нейросетевые модели извлечения информации. Полученный метод должен быть применим для новых сайтов, не использовавшихся при обучении моделей, а также показывать качество не хуже существующих автоматических методов.

Для оценки эффективности разработанного решения необходимо провести тестирование, сравнив качество сбора данных сгенерированными картами с качеством, достигаемым полностью автоматическим методом на уровне отдельных страниц.

3. Обзор существующих решений

В данной главе рассматриваются существующие решения для автоматического извлечения атрибутов новостных страниц. Эти методы позволяют извлекать атрибуты без необходимости ручного создания карт сбора для конкретного сайта.

3.1 Проприетарные сервисы для извлечения атрибутов новостей

Существует ряд закрытых платных сервисов, таких как Zyte Automatic Extraction [14] и Diffbot [15], которые предоставляют готовое решение для автоматического извлечения структурированных данных из новостных страниц.

Эти инструменты принимают на вход URL новостной страницы, отображают её в headless-браузере (браузер без графического интерфейса, который используется для загрузки и обработки содержимого веб-страниц) и извлекают атрибуты с помощью методов машинного обучения.

3.2 Открытые библиотеки для извлечения атрибутов новостей

Инструменты с открытым исходным кодом, такие как newspaper3k [16], newsplease, trafilatura – python-библиотеки, предоставляющие простой в использовании интерфейс для извлечения данных из новостных статей, таких как заголовок, авторы, дата публикации, текст и изображения. Они принимают на вход URL и HTML-код новостной страницы и возвращают извлечённые данные в структурированном виде. Принцип работы открытых инструментов основан как на эвристиках, таких как плотность ссылок и количество слов, так и на машинном обучении.

Инструменты также включают в себя дополнительные функции для обработки текста на основе NLP, например, извлечение ключевых слов и создание резюме статей.

3.3 Нейросетевые подходы для извлечения атрибутов новостей

Исследования последних лет посвящены использованию нейронных сетей для решения задачи автоматического извлечения структурированных данных из веб-страниц.

Модель SimpDOM, основанная на LSTM-сети, решает задачу классификации узлов DOM-дерева. BiLSTM-сеть принимает векторное представление узлов на основе текстовых признаков, а далее классификатор анализирует её выход вместе с набором эвристических структурных признаков.

Ряд моделей ориентируются на визуальное представление узлов. Например, модель CoVA [17] вместо текстовой информации анализирует только визуальные и структурные характеристики HTML-элементов.

В современных исследованиях рассматриваются BERT-подобные модели, которые сначала предобучаются на задачах понимания HTML-документов, а затем могут быть дообучены на прикладные задачи. Описываемые модели используют механизм внутреннего внимания (self-attention) и называются трансформерами.

Так, модель MarkupLM использует текстовые признаки и XPath в качестве информации разметки. Авторы опубликовали предобученную модель и интерфейс дообучения для задачи извлечения информации, формулируемую как задача классификации токенов. При этом в процессе обучения сохраняется информация о принадлежности токенов DOM-узлам.

Мультимодальные модели, такие как WebLM [18], совмещают использованием текстовых и структурных признаков узлов с анализом визуального представления соответствующего им элемента на скриншоте веб-страницы.

3.4 Вывод к обзору

В данной главе были рассмотрены различные подходы к извлечению атрибутов новостных страниц. Для использования в генерации карт сбора была выбрана актуальная BERT-подобная модель MarkupLM. Она позволяет извлекать как текст атрибутов, так и их DOM-узлы, а также имеет в открытом доступе предобученную модель для применения на прикладных задачах.

4. Метод построения карт сбора для новостных сайтов

В данной главе описывается разработка метода автоматической генерации карт сбора на основе предсказаний дообученной модели.

4.1 Алгоритм построения карты сбора

Алгоритм построения карты сбора включает следующие шаги:

- Идентификация атрибутов на веб-страницах. На начальном этапе загружаются несколько HTML страниц по их URL, и алгоритм автоматического извлечения

анализирует их для идентификации XPath [19] атрибутов новостей.

- Преобразование XPath в CSS-селекторы. Далее происходит преобразование позиционного XPath, который является полным путем до элемента на странице, в уникальный CSS-селектор с помощью плагина css-selector-generator [20], который принимает на вход от одного и более XPath и возвращает CSS-селектор. Если атрибут затрагивает несколько DOM-узлов или является многозначным, для узлов находится общий CSS-селектор.
- Выбор подходящего селектора. Если для атрибута существует несколько селекторов на разных страницах, то выбирается наиболее часто встречающийся. Если есть несколько наиболее часто встречающихся селекторов, то выбирается любой из этих селекторов.

На выходе формируется карта сбора данных, содержащая спецификации для каждого из атрибутов. Для построения карты сбора необходимо выбрать несколько новостных страниц.

4.2 Дообучение модели MarkupLM

Для решения задачи извлечения DOM-узлов с атрибутами новостных статей была дообучена модель MarkupLM. Дообучение MarkupLM проводилось на размеченном наборе данных новостных страниц из исследования [21], включающем 724 страницы со 114 сайтов русскоязычных СМИ. В новостях размечались заголовок, дата публикации, основной текст, авторы, теги и некоторые другие атрибуты.

По HTML-коду страниц строилось DOM-дерево, из которого извлекались текстовые узлы. Из текста удалялись лишние пробелы и управляющие символы. Для каждого узла сохранялся его текст, XPath и метка атрибута. При отсутствии атрибута ставилась метка «None». Так как модель MarkupLM является англоязычной, все тексты дополнительно переводились на английский — как на этапе обучения, так и при использовании модели далее. Для перевода использовалась библиотека argos-translate [22].

Далее тексты токенизировались, каждому токену сопоставлялась метка и токенизированный XPath. Так как модель ограничена максимальной длиной входа в 512 токенов, последовательность токенов страниц была разбита на части этой предельной длины с перекрытием в 170 токенов (около трети длины).

4.3 Генерация селекторов атрибутов

Интерфейс для использования дообученной модели MarkupLM предназначен для языка Python. Однако для преобразования XPath в CSS-селекторы используется плагин на JavaScript – css-selector-generator, поскольку он позволяет создавать селектор, ориентированный на несколько элементов одновременно. Для того чтобы встроить JavaScript код в Python использовался Selenium [23].

- Обработка страниц с помощью инструмента на основе MarkupLM. Инструмент на основе MarkupLM анализирует представленные веб-страницы и выделяет XPath для DOM-узлов, соответствующих различным атрибутам новости.
- Преобразование XPath в CSS селекторы. Для каждой страницы и каждого атрибута, используя плагин css-selector-generator, преобразуем XPath в CSS селекторы. Если для один атрибут затрагивает несколько DOM-узлов, то для них находится общий CSS селектор.
- Выбор популярных селекторов. Определяем самые часто встречающиеся на страницах CSS селекторы для каждого атрибута.
- Создание карты сбора данных: на основе полученных данных формируется карта сбора.

В ходе разработки возникала проблема с плагином css-selector-generator, который иногда находит селекторы, основываясь на уникальных атрибутах. Это не оптимально, так как подобные селекторы могут быть нестабильными при изменениях на веб-страницах. Чтобы улучшить результаты, было решено запретить плагину использовать для генерации селекторов псевдокласс nth-of-child и атрибуты в формате [title=*], [href=*], [src=*]. Эти изменения позволили сделать генерируемые селекторы более универсальными и стабильными, что улучшило качество построения карты сбора данных.

5. Тестирование

В данной главе представлена экспериментальная оценка дообученной модели MarkupLM и разработанного метода генерации карт сбора. Целью тестирования было оценить качество дообученной модели в сравнении с существующими инструментами с открытым исходным кодом, а также оценить качество извлечения по автоматически сгенерированным картам сбора данных в сравнении с использованием дообученной модели на каждой странице.

5.1 Дообученная модель MarkupLM

Для MarkupLM использовались гиперпараметры, предлагаемые авторами модели. Наилучшее качество показало дообучение в течение одной эпохи. Оценка качества проводилась на описанном ранее русскоязычном новостном наборе данных с использованием 5-Fold кросс-валидации и метрики F1. Для атрибутов применялись три стратегии сопоставления, предложенные авторами статьи [21]:

- Заголовки, подзаголовки и тексты сравнивались как мешки 4-грамм (последовательности из 4 слов), потому что отдельные слова можно увидеть как в правильно выделенной части текста, так и в нежелательной его части, в то время как для n-грамм это значительно менее вероятно. Процесс выглядит следующим образом:
    - Токенизация текста: текст преобразуется в список слов;
    - Генерация 4-грамм: из списка слов формируются все возможные последовательные комбинации из четырех слов;
    - Сопоставление 4-грамм: определяет количество совпадений 4-грамм между эталонным и извлечённым текстом.
- Подсчитывается TP — количество 4-грамм, которые присутствуют в обоих текстах, FP – количество 4-грамм, которые присутствуют в извлеченном тексте, но отсутствуют в эталоне, FN – количество 4-грамм, которые присутствуют в эталонном тексте, но отсутствуют в извлеченном;
- Остальные атрибуты, исключая дату, сопоставлялись как множества значений. TP – истинноположительное предсказание атрибута, FP – ложноположительное предсказание атрибута, FN – ложноотрицательное предсказание атрибута;

Даты приводились к единому формату с помощью библиотеки dateparser [24] и затем сопоставлялись как множества значений, TP, FP и FP аналогичны прошлой стратегии.

Результаты в сравнении с открытыми эвристическими инструментами представлены в табл. 1.

Результаты экспериментальной оценки показывают, что дообученный MarkupLM превосходит по качеству извлечения открытые инструменты по всем атрибутам.

5.2 Карты сбора

Было проведено сравнение качества извлечения атрибутов из новостных веб-страниц с помощью автоматически сгенерированных карт сбора и дообученной модели MarkupLM, применяемой на каждой странице.

Тестирование охватывало 16 случайных сайтов, собранных с агрегатора Mediametrics [25], которых не было в обучающих данных для MarkupLM. С каждого ресурса было выбрано по 20 случайных новостных страниц.

Табл. 1. Оценка качества MarkupLM и эвристических инструментов (F1-мера).
Table 1. Performance evaluation of MarkupLM and heuristic tools (F1 score).

Атрибут	Инструмент			
	newspaper3k	newsplease	trafilatura	MarkupLM
Заголовок	0.94	**0.98**	0.96	**0.98**
Подзаголовок	0.08	0.40	0.41	**0.63**
Дата публикации	0.50	0.57	0.67	**0.96**
Текст	0.78	0.68	0.78	**0.93**
Авторы	0.23	0.23	0.43	**0.62**
Категории	0	0	0.20	**0.52**
Теги	0.62	0	0.29	**0.87**

Для сравнения извлечённых и эталонных значений атрибутов применялись те же стратегии сопоставления, что и при оценке качества дообученной модели MarkupLM ранее. Метрики со всех сайтов и страниц были усреднены и представлены в "табл. 2".

Качество извлечения атрибутов с помощью автоматически сгенерированных карт сбора по полноте, точности и F1-мере превосходит в большинстве атрибутов вариант с применением дообученной нейросетевой модели MarkupLM к каждой странице сайта.

Автоматический генератор карт сбора уступает MarkupLM по атрибуту "text" по полноте и F1-мере, поскольку на некоторых сайтах для каждой новостной страницы MarkupLM выбирал разные наборы DOM-узлов. Это привело к созданию нескольких CSS-селекторов, из которых был выбран один, не охватывающий весь текст всех новостных страниц сайта.

Табл. 2. Результаты измерений метрик точности, полноты и F1-меры.
Table 2. Measurement results of accuracy, completeness and F1-measure metrics.

Поле	F1		Точность		Полнота	
	Карта сбора	MarkupLM	Карта сбора	MarkupLM	Карта сбора	MarkupLM
Заголовок	**1.0**	0.939	**1.0**	0.939	**1.0**	0.939
Подзаголовок	**0.786**	0.710	**0.787**	0.739	**0.787**	0.702
Дата публикации	**0.938**	0.805	**0.938**	0.805	**0.938**	0.805
Текст	0.805	**0.823**	**0.841**	0.840	0.882	**0.893**
Автор	**0.952**	0.813	**0.947**	0.816	**0.969**	0.815
Категория	**0.938**	0.642	**0.938**	0.641	**0.938**	0.643
Теги	**0.922**	0.877	**0.916**	0.891	**0.933**	0.874

6. Заключение

В результате исследования был разработан метод, способный по ограниченному набору новостных страниц сайта сгенерировать его карту сбора, которая позволит извлекать атрибуты любой новостной страницы этого ресурса. Новизна метода заключается в том, что он обобщает полученные моделью предсказания на уровне всего сайта.

Для генерации карт сбора была дообучена нейросетевая модель MarkupLM, используемая для извлечения атрибутов новостных страниц. Дообученный MarkupLM превзошёл по F1-мере открытые эвристические инструменты.

Был предложен метод обобщения предсказаний MarkupLM на уровне сайта и вывода по ним правил извлечения атрибутов в виде карт сбора. Эксперименты показали, что извлечение атрибутов по таким картам превосходит по качеству использование дообученного MarkupLM на уровне отдельных страниц.

Разработанный метод может быть обобщён на другие предметные области при условии дообучения модели на соответствующих данных.

Список литературы / References

[1]. Ferrara E., De Meo P., Fiumara G., Baumgartner R. Web data extraction, applications and techniques: A survey. Knowledge-Based Systems, 2014, vol. 70, pp. 301-323. DOI: 10.1016/j.knosys.2014.07.007

[2]. Octoparse. Available at: https://www.octoparse.com/, accessed 25.09.2024

[3]. Web Scraper. Available at: https://webscraper.io/, accessed 25.09.2024

[4]. Barbaresi A. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. Association for Computational Linguistics, Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, 2021, pp. 122-131. DOI: 10.5281/zenodo.3460969

[5]. Barbaresi A. Trafilatura: Discover and Extract Text Data on the Web. Available at: https://github.com/adbar/trafilatura/, accessed 25.09.2024

[6]. Hamborg F., Meuschke N., Breitinger C., Gipp B. news-please: A Generic News Crawler and Extractor. Proceedings of the 15th International Symposium of Information Science, 2017, pp. 218-223. DOI: 10.5281/zenodo.4120316

[7]. Hamborg F., Meuschke N., Breitinger C., Gipp B. news-please. Available at: https://github.com/fhamborg/news-please/, accessed 25.09.2024

[8]. Junlong L., Yiheng X., Lei C., Furu W. MarkupLM: Pre-training of Text and Markup Language for Visually-rich Document Understanding. Association for Computational Linguistics, Dublin, Ireland, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6078-6087. DOI: 10.18653/v1/2022.acl-long.420

[9]. Junlong L., Yiheng X., Lei C., Furu W. MarkupLM. Available at: https://huggingface.co/docs/transformers/model_doc/markuplm/, accessed 25.09.2024

[10]. Zimeng L., Bo S., Linjun S., Ming G., Gen L., Daxin J. WIERT: Web Information Extraction via Render Tree. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, vol. 37, num. 11, pp. 13166-13173. DOI: 10.1609/aaai.v37i11.26546

[11]. Yichao Z., Ying S., Nguyen H. V., Nick E., Sandeep T. Simplified DOM Trees for Transferable Attribute Extraction from the Web. arXiv, 2021. DOI: 10.48550/arXiv.2101.02415

[12]. Richardson L. Beautiful Soup. Available at: https://www.crummy.com/software/BeautifulSoup/, accessed 25.09.2024

[13]. Selectors Level 3. Available at: https://www.w3.org/TR/selectors-3/, accessed 25.09.2024

[14]. Zyte Automatic Extraction. Available at: https://docs.zyte.com/zyte-api/usage/extract.html, accessed 25.09.2024

[15]. Diffbot. Available at: https://www.diffbot.com/products/extract/, accessed 25.09.2024

[16]. Ou-Yang L. Newspaper3k: Article scraping & curation. Available at: https://github.com/codelucas/newspaper?tab=readme-ov-file/, accessed 25.09.2024

[17]. Kumar A., Morabia K., Wang J., Chang K. C. C., Schwing A. CoVA: context-aware visual attention for webpage information extraction. Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5), 2022, pp.80-90. DOI: 10.18653/v1/2022.ecnlp-1.11.

[18]. Xu H., Chen L., Zhao, Z., Ma D., Cao R., Zhu Z., & Yu K. Hierarchical Multimodal Pre-training for Visually Rich Webpage Understanding. Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 864–872. DOI: 10.1145/3616855.3635753

[19]. XML Path Language (XPath) 3.1. Available at: https://www.w3.org/TR/xpath-31/, accessed 25.09.2024

[20]. Fridrich R. CSS Selector Generator. Available at: https://github.com/fczbkk/css-selector-generator/, accessed 25.09.2024

[21]. Varlamov M., Galanin D., Bedrin P., Duda S., Lazarev V., Yatskov A. A Dataset for Information Extraction from News Web Pages. 2022 Ivannikov Ispras Open Conference
[22]. Finlay P. J. Argos Translate. Available at: https://github.com/argosopentech/argos-translate/, accessed 25.09.2024
[23]. Selenium. Available at: https://www.selenium.dev/, accessed 25.09.2024
[24]. Dateparser. Available at: https://github.com/scrapinghub/dateparser/, accessed 25.09.2024
[25]. Mediametrics. Available at: https://mediametrics.ru/, accessed 25.09.2024

Информация об авторах / Information about authors

Сергей Сергеевич ДУБОВИЦКИЙ – программист Института системного программирования. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации.

Sergei Sergeevich DUBOVITSKII – programmer at the Institute of System Programming of the RAS. Research interests: data collection from web resources, automation of the data collection process, information extraction.

Павел Александрович БЕДРИН – старший лаборант Института системного программирования, магистрант ВМК МГУ. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Pavel Alexandrovich BEDRIN – senior laboratory assistant at the Institute of System Programming of the RAS, master student of the CMC faculty of Lomonosov Moscow State University. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.

Александр Константинович ЯЦКОВ – младший научный сотрудник Института системного программирования, ведущий программист ВМК МГУ. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Alexander Konstantinovich YATSKOV – junior researcher at the Institute of System Programming of the RAS, leading programmer at the CMC faculty of Lomonosov Moscow State University. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.

Максим Игоревич ВАРЛАМОВ – научный сотрудник Института системного программирования. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Maxim Igorevich VARLAMOV – researcher at the Institute of System Programming of the RAS. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.