УДК 004.051

Н.С. Каргин, маг.; Н.И. Гурин, доц., канд. физ.-мат. наук
(БГТУ, г. Минск)

МЕТОД ИЗВЛЕЧЕНИЯ ОСНОВНОГО СОДЕРЖИМОГО ИЗ ВЕБ-СТРАНИЦЫ

Процесс извлечения основного содержимого затрудняется тем, что веб-документы содержат в себе множество неинформативной или нетекстовой информации: CSS стили, JavaScript код, навигационные и декоративные элементы, генерируемый пользователями контент (например, комментарии). Для решения данной проблемы в настоящее время применяются два основных подхода: на основе правил и на основе семантики веб-документов.

В работе предлагаются собственные алгоритмы извлечения основного содержимого из веб-страниц и методы решения проблем, затрудняющих извлечение основного содержимого, на основе визуальных характеристик и внутреннего содержимого элементов страницы. Для решения задачи извлечения основного содержимого разработан метод, который ориентируется на визуальные характеристики элементов, текстовые узлы документа и стремится определить только один корневой элемент с основным содержимым веб-страницы. Основные шаги метода:

1) сбор метаинформации веб-страницы;

2) формирование множества всех текстовых узлов веб-страницы;

3) определение корневого элемента с основным содержимым;

4) преобразование корневого элемента в JSON формат.

Все шаги выполняются в браузере с помощью подключения JS скриптов в тело страницы. Скрипт извлечения основного содержимого может быть выполнен как на сервере, например, в среде NodeJS с применением библиотеки Puppeteer, так и в браузере пользователя с помощью расширений.

Важно отметить, что этапы формирования множества текстовых узлов и поиска корневого элемента могут быть пропущены при условии, что в процессе преобразования других страниц с этого же веб-сайта происходит сохранение XPath или CSS селектора к корневому элементу с основным содержимым.

Большинство контент-ориентированных веб-сайтов используют шаблонизаторы для рендеринга HTML и у страниц меняется только часть с основным содержимым. Следовательно, при накоплении некоторого количества таких селекторов, для новых страниц искать этот

элемент уже нет необходимости: достаточно получить из базы данных наиболее часто встречающийся путь, что позволяет ускорить извлечение основного содержимого.

Разработанный метод был протестирован на 111 веб-страницах с 25 различных веб-сайтов. Тестирование включало в себя ручное сравнение извлеченного содержимого и оригинальных веб-страниц.

В результате тестирования было выявлено 6 веб-страниц, для которых не удалось корректно определить корневой элемент, что составляет 5.41% от всех страниц. Следует учитывать, что сканирование выполнялось для небольшого количества веб-страниц с одного веб-сайта и не применялась оптимизация с «запоминанием» корневого элемента для веб-сайта.

На восьми веб-страницах были выявлены недостатки, связанные с некорректным преобразованием рекламных блоков или «ленивых» изображений (изображения, которые начинаются загружаться только при приближении видимой области окна браузера к ним).

Для реализации ленивых изображений применяются различные техники с помощью JavaScript, а HTML в исходной разметке выглядит исключительно как заглушка и данную проблему нельзя решить только с помощью анализа HTML. Для решения данной проблемы была реализована функция, которая выполняет прокрутку страницы до конца корневого элемента с основным содержимым и ожидает загрузку изображений в нём.

Для оценки эффективности представления основного содержимого в JSON формате был произведен замер исходных HTML документов (без учета подключаемого JavaScript, CSS) и преобразованного основного содержимого:

– средний размер HTML документа составляет 291 килобайт;

– средний размер извлеченного содержимого в JSON формате 16 килобайт.

Таким образом, представление основного содержимого в JSON формате в 18 раз эффективнее. Минимальный размер JSON документа составил 3 килобайта, максимальный 63; минимальный размер HTML документа 16 килобайт, максимальный размер 1462 килобайта. При этом следует учитывать, что JSON документы уже содержат подробную информацию об изображениях и их уменьшенные копии.

Было проведено сравнительное тестирование разработанного метода и реализациях алгоритма Readability в браузерах Mozilla Firefox и Apple Safari. Для данного тестирования было разработано две визуально одинаковых страницы. Одна из страниц использует стандарт собственных элементов для представления содержимого, а вто-

рая использует семантические HTML элементы. Разработанные веб-страницы содержат небольшой текст, навигационные элементы и комментарии.

Браузер Firefox не смог определить элемент с основным содержимым на странице, где присутствуют собственные элементы. На странице с семантическими HTML элементами проблемы не возникло.

Браузер Safari смог корректно определить элемент с основным содержимым для двух страниц, но не смог корректно отобразить его на странице, где присутствуют собственные элементы: все подряд идущие абзацы текста сливаются в один.

Разработанный метод корректно определил и преобразовал основное содержимое на двух страницах, так как не ориентируется на семантику HTML элементов.

Технологии извлечения основного содержимого веб-страниц встречаются в самых разных приложениях и сервисах, которыми пользуются миллионы людей ежедневно: поисковые системы, новостные агрегаторы, приложения для чтения новостей, мессенджеры и многие другие.

Разработанный метод извлечения основного содержимого на основе визуальных характеристик и внутреннего содержимого элементов не ориентируется на семантику элементов HTML документа, что делает алгоритм устойчивым к разнообразию подходов к вёрстке и применению на странице собственных элементов.

Выделение корневого элемента с основным содержимым веб-страницы позволяет улучшить качество работы алгоритма (неправильное определение основного содержимого, скорость работы) при обработке большого количества страниц из одного источника.

Конвертирование основного содержимого в JSON формат, позволяет избавиться от сложности и неоднозначности HTML разметки, что значительно упрощает машинную обработку основного содержимого, а для отображения или озвучивания статей в таком формате не требуется браузер, что особенно актуально для мобильных устройств и встраиваемой техники с ограниченными ресурсами.