УДК 004.9

К.т.н., доцент Петрашенко А.В., студент Нечипоренко О.О.

**Національний технічний університет України
«Київський політехнічний інститут»**

ПРОГРАМНІ ЗАСОБИ СТРУКТУРНОГО АНАЛІЗУ ВЕБ-ДОКУМЕНТІВ

Abstract

Andriy V. Petrashenko, assoc. prof., PhD; Oleg Nechyporenko, student
Software to structural analysis of Web documents

The paper describes a developed method of analysis Web documents to select the main content. This method is based on the number of generated rules about the page elements and their relationship. The main objective of this work is to improve the quality analysis of Web documents by removing several items from them. The paper shows that the proposed algorithm can effectively allocate main part web pages.

Вступ

Веб – це велике сховище даних, яке дуже активно використовується. За останні 10 років кількість веб-сайтів зросла в 10 разів [1, 3], а кількість даних в них зростає в 2 рази кожні 1,5 року [2]. В той же час чималу частину вмісту веб-сторінок складають елементи, котрі до основного контенту відношення не мають.

Існує багато алгоритмів виділення контенту веб-сторінок [1-4]. Одні базуються на виділенні спільних частин веб-сторінок, другі враховують розташування елементів на сторінках та ін.

У даній статті пропонується метод структурного аналізу веб-документів на основі DOM-моделі (DOM – Document Object Model), який базується на гіпотезі, що можна вивести ряд правил аналізу DOM, керуючись якими, можна отримати основний вміст веб-сторінки.

Постановка задачі

Задача полягає в розробці ряду правил для аналізу DOM-моделі та програмної реалізації цих правил.

Термінологія

Веб-сторінка (англ. Web-page) — інформаційний ресурс, доступний в мережі World Wide Web (Всесвітня павутина), який можна переглянути у

веб-браузері. Зазвичай, ця інформація записана в форматі HTML або XHTML, і може містити гіпертекст з навігаційними гіперпосиланнями на інші веб-сторінки.

Вміст (інформаційне наповнення, контент) — будь-яке інформаційно значуще наповнення інформаційної системи (зокрема веб-вузла) – тексти, графіка, мультимедіа.

Контент-аналіз — стандартна методика досліджень в області природничих наук, предметом аналізу якої є вміст текстових масивів, веб-сторінок та продуктів комунікативної кореспонденції.

DOM (англ. Document Object Model - «об'єктна модель документа») — це незалежний від платформи і мови програмний інтерфейс, що дозволяє програмам та скриптам отримати доступ до вмісту HTML, XHTML і XML-документів, а також змінювати вміст, структуру та оформлення таких документів.

Тег table — контейнер для елементів, що визначають вміст таблиці.

Тег div — блочний елемент, призначенням якого є виділення фрагменту документа з метою зміни вигляду його вмісту.

Опис алгоритму аналізу структурного аналізу веб-документів

Частину правил розбору, за якими працює алгоритм, можна вивести навіть не вдаючись у вміст сторінок:

1. В секції head основного контенту бути не може. Виняток становить лише тег title, однак його вміст майже завжди повторюється на сторінці.

2. Ряд тегів взагалі не може мати текстового вмісту (елементи форм, горизонтальні та вертикальні лінії, тексти скриптів, таблиць стилів та ін.). Здається, що в цю категорію потрапляють і зображення, однак частина з них може бути ілюстраціями до основного вмісту сторінки.

3. Коментарі в HTML-коді також є другорядними елементами.

Однак інша частина правил потребує глибокого аналізу структури типової веб-сторінки.

4. Очевидно, що 99.9% веб-сторінок мають навігаційну частину. Основною відмінністю її від інших блоків на сторінці є велика концентрація посилань на інші сторінки. Це і буде основних критерієм її виділення в DOM-моделі (блок, де кількість посилань по відношенню до іншого тексту, більша за вибрану величину). Експериментально було виявлено, що в середньому ця величина становить 25%.

5. Також на сторінках популярних веб-сайтів є блоки з рекламою. Їх пошук може бути таким, як і пошук навігаційних елементів (рекламні блоки – це також блоки з великою кількістю посилань на інші веб-сайти). Виходячи з цього, можна зробити правило, що рекламний блок – це блок, де багато зовнішніх посилань. Іншим шляхом виділення реклами є пошук

«стоп-посилань» - тобто посилань, що завідомо ведуть на рекламні сайти (банерні мережі та ін.). Знайшовши одне або декілька таких посилань в блоці, видаляємо його як рекламний. Цей спосіб має низьку швидкодію, тому що рекламних та банерних мереж зараз дуже багато і перевірка кожного посилання на належність до однієї з них буде займати багато часу. Тому цей спосіб залишається теоретично можливим, але не практичним.

6. Після обробки документа часто залишаються "шматки" тексту від різних другорядних блоків (наприклад, заголовки на кшталт "Друзі сайту", "Реклама" і т.д.). Часто такі тексти вставлені на сторінку таким чином, що не можуть бути видалені разом із блоком, до якого відносяться. І в результаті виникає ситуація, що на обробленій сторінці залишаються зайві фрагменти тексту. Щоб уникнути цього було запропоноване таке правило - "текст, що знаходиться на відстані більше ніж N тегів від найбільшого текстового блоку, не відноситься до основного вмісту сторінки і може бути відкинутий". Значення N для різних веб-сторінок коливається, але не виходить за рамки діапазону 3-8 тегів. До вводу цього правила для алгоритму було проблемою видалення блоків коментарів зі сторінки (на багатьох сайтах є можливість коментування для користувачів). Оскільки такі блоки є текстовими, вони не потрапляли під інші правила і залишались на сторінці. Під час експериментів було встановлено, що це правило відфільтровує до 70% другорядного тексту.

7. Також слід зауважити стосовно вибору найбільшого текстового блоку. Очевидно, що, якщо не вводити ніяких обмежень, то найбільшим блоком з текстом буде сама веб-сторінка. В запропонованому алгоритмі такий блок шукається лише серед тегів table та div, що не мають вкладених у себе table та div відповідно. Завдяки цьому, алгоритм генерує практично "чистий" контент веб-сторінок.

8. Після виконання всіх правил виконується "збирання сміття", що могло залишитися на сторінці. Найчастіше це порожні (без тексту) теги, які вже не мають бути на сторінці. Виняток тут становлять теги, що впливають на розмітку документа (елементи таблиць, наприклад). Тому у "збирача сміття" є "білий список" тегів, які він не вилучає.

По завершенню всіх етапів користувачу надається відфільтрована сторінка з мінімумом зайвих елементів, яку йому зручно переглядати.

Результати

При тестуванні використовувались матеріали зі 100 сайтів різних тематик, структур та мов. Більша частина їх не перевищувала 100Кб (сам HTML-код, без скриптів, таблиць стилів та картинок). З кожного сайту брали по одній сторінці. Сайти являли собою типові інформаційні портали та корпоративні сайти/блоги.

За критерії якості роботи алгоритму було обрано відсоток "сміттєвих" елементів на сторінці після її обробки та відсоток помилкових результатів роботи алгоритму (тобто ситуації, коли алгоритм видаляв зі сторінки частину потрібного контенту).

Аналіз результатів проводився «вручну» – експерт порівнював вихідну сторінку та сторінку, отриману після роботи алгоритму. Слід зауважити, що експерт міг корегувати налаштування алгоритму згідно правил (п. 4 – 6), якщо, на його думку, алгоритм міг покращити свій результат.

В табл. 1 та 2 наведені оцінки експерта.

Таблиця 1.

Результати роботи алгоритму без корегувань експерта

Оцінка	**Кількість сторінок**	**Наявність «сміттєвих» елементів (*)**	**Помилкові спрацювання (**)**
2	4	3	3
3	17	13	11
4	40	25	19
5	39	0	0
Середній бал	4.1		

Таблиця 2.

Результати роботи алгоритму з корегуваннями експерта

Оцінка	**Кількість сторінок**	**Кількість корегувань експерта**	**Наявність «сміттєвих» елементів (*)**	**Помилкові спрацювання (**)**
2	4	0	3	3
3	12	0	8	9
4	23	5	12	14
5	61	22	0	0
Середній бал	4.41			

* означає, що в даній групі на заданій кількості сторінок були «сміттєві» елементи.

** означає, що в даній групі на заданій кількості сторінок були помилкові спрацювання алгоритму.

Висновки

При тестуванні роботи алгоритму було виявлено ряд сильних та слабких його сторін.

Алгоритм безпомилково обробляє сторінки, основним вмістом яких є текст та зображення (незалежно від кількості другорядних елементів). Однак алгоритм помилково спрацьовує на блоки посилань, що є всередині основного вмісту (це викликано тим, що такі блоки відфільтровуються раніше, ніж відбувається пошук текстових блоків).

Алгоритм знаходить основний вміст сторінки, навіть якщо він дуже малий (до 5% від загального об'єму сторінки). Виняток становили тестові

сторінки, в яких блок неосновного тексту був більший, ніж основного. Для таких випадків в алгоритм доцільно додати урахування позиції блоку в DOM-дереві.

Література

1. *Журавлев С.В., Юдина Т.Н.*, Информационная система РОССИЯ // НТИ. Сер.2. — 1995. — № 3. — С.18-20.

2. *Кураленок И.Е., Некрестьянов И.С.*, Павлова Е.Ю. РОМИП 2003: Опыт организации. // Труды РОМИП'2003, октябрь 2003, — СПб: НИИ Химии СПбГУ — С.9-30.

3. *Кураленок И.Е., Некрестьянов И.С.*, отчет организаторов Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004 — С.27-29.

4. *Ziv Bar-Yossef, Sridhar Rajagopalan,* Template Detection via Data Mining and its Applications // In Proceedings of WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA – P. 72.