УДК 004.67

АВТОМАТИЗОВАНА ЕКСТРАКЦІЯ СТРУКТУРОВАНОЇ ІНФОРМАЦІЇ З МНОЖИНИ ВЕБ-СТОРІНОК

С.Д. Погорілий, А.А. Крамов

Обґрунтовано доцільність використання методів екстракції структурованих даних з множини HTML-сторінок для здійснення інформаційного пошуку в мережі Internet. Проаналізовано основні методи екстракції структурованих даних з множини веб-сторінок, які сформовані спільним сценарієм, але різними наборами даних. Розглянуто класифікацію методів за ступенем автоматизації (фактору впливу користувача) процесу формування шаблону. Детально описано принципи роботи основних неконтрольованих методів (Roadrunner, FiVaTech, Trinity), розглянуто їхні переваги та недоліки. Обґрунтовано доцільність використання методу Trinity для екстракції даних порівняно з іншими методами. Показано проблему вибору вхідних документів методу серед множини HTML-сторінок для формування узагальненого шаблону. Проведено експериментальну перевірку методу Trinity на множині HTML-сторінок англомовних статей українських наукових журналів. Для формування тестової множини HTML-сторінок виконано автоматизований обхід веб-сайтів журналів за допомогою пошукового роботу. Реалізацію пошукового роботу здійснено за рахунок обробки об'єктної моделі HTML-документів, отриманих з веб-сайтів. Шаблони (регулярні вирази), сформовані методом Trinity, застосовано до всього набору вхідних HTML-сторінок. Результати екстракції – структуровані дані про статті (назва, автори, анотація, ключові слова) – експортовано до бази даних з можливістю їх подальшого аналізу. Здійснено порівняння отриманих результатів з даними про статті, одержаними за допомогою аналізу об'єктної моделі веб-сторінок власноруч. Обраховано похибку використання методу Trinity на експериментальній множині HTML-сторінок.
Ключові слова: екстракція даних; методи екстракції; класифікація методів екстракції; метод Trinity; тернарне дерево документу; префіксний обхід дерева; об'єктна модель HTML-сторінки; автоматизоване збирання веб-сторінок; пошуковий робот; шаблон HTML-сторінки; формування регулярного виразу.

Обоснована целесообразность использования методов экстракции структурированных данных из множества HTML-страниц для осуществления информационного поиска в сети Internet. Проанализированы основные методы экстракции структурированных данных из множества веб-страниц, которые сформированы общим сценарием, но разными наборами данных. Рассмотрена классификация методов по степени автоматизации (фактора влияния пользователя) процесса формирования шаблона. Подробно описаны принципы работы основных неконтролируемых методов (Roadrunner, FiVaTech, Trinity), рассмотрены их преимущества и недостатки. Обоснована целесообразность использования метода Trinity для экстракции данных по сравнению с другими методами. Показана проблема выбора входных документов метода среди множества HTML-страниц для формирования обобщенного шаблона. Осуществлена экспериментальная проверка метода Trinity на множестве HTML-страниц англоязычных статей украинских научных журналов. Для формирования тестового множества HTML-страниц выполнено автоматизированный обход веб-сайтов журналов с помощью поискового робота. Реализацию поискового робота осуществлено за счет обработки объектной модели HTML-документов, полученных с веб-сайтов. Шаблоны (регулярные выражения), сформированные методом Trinity, применены ко всему набору входных HTML-страниц. Результаты экстракции – структурированные данные о статьях (название, авторы, аннотация, ключевые слова) – экспортировано в базу данных с возможностью их последующего анализа. Осуществлено сравнение результатов экстракции с данными о статьях, полученными с помощью самостоятельного анализа объектной модели веб-страниц. Рассчитана погрешность использования метода Trinity на экспериментальном множестве HTML-страниц.
Ключевые слова: экстракция данных; методы экстракции; классификация методов экстракции; метод Trinity; тернарное дерево документа; префиксный обход дерева; объектная модель HTML-страницы; автоматизированный сбор веб-страниц; поисковый робот; шаблон HTML-страницы; формирование регулярного выражения.

The expediency of using methods of structured data extraction from a set of HTML pages for the information search in the Internet is substantiated. The main methods of structured data extraction from the set of web pages, which are formed by a common scenario with different sets of data, are analyzed. The classification of methods according to the degree of automation (the factor of user influence) of the template formation process is considered. The principles of work of the main unsupervised methods (Roadrunner, FiVaTech, Trinity) are described in detail. Advantages and disadvantages of methods are shown. The expediency of using the Trinity method for data extraction in comparison with other methods is substantiated. The problem of choosing input documents for method among a set of HTML pages for generating a common template is considered. Experimental verification of Trinity method on the set of HTML pages, which represent articles of Ukrainian scientific journals, is made. To create a test set of HTML pages, an automated crawl of web site is performed. The realization of the search bot is done by processing the object model of HTML documents obtained from web sites. Templates (regular expressions) formed by the Trinity method are applied to the entire set of input HTML pages. Extraction results (structured data about articles) are exported to the database with the possibility of further analysis. The obtained results are compared with the data about the articles obtained by the manual analysis of the object model of web pages. The error in using the Trinity method on the experimental set of HTML pages is calculated.
Key words: data extraction; extraction methods; classification of extraction methods; Trinity method; ternary document tree; prefix tree traversal; object model of the HTML page; automated collection of web pages; search bot; HTML page template; formation of a regular expression.

Вступ

Обсяг інформації в мережі Internet постійно збільшується. Зважаючи на високу динаміку приросту обсягу інформації, розміщеної на веб-ресурсах, та відсутність її стандартної структуризації, інформаційний пошук в мережі Internet зіштовхується з проблемою обробки Big Data. Робота з Big Data розглядається як швидкісний пошук необхідної інформації серед великого обсягу *неструктурованих даних* (множина HTML-сторінок, які відповідають запиту користувача). Постійне зростання обсягу інформації спонукає компанії створювати сервіси і застосування для автоматизованого пошуку та обробки даних у мережі Internet. Обробка отриманих даних здійснюється з метою формування нових бізнес-стратегій, передбачення напрямків розвитку конкурентів чи аналізу діяльності компанії. Для ефективної обробки обсягу інформації, відповідного Big Data, виконується

паралелізація алгоритмів обробки даних водночас з використанням потужних графічних процесорів [1–2]. Згадані вище застосування зіштовхуються ще з однією проблемою обробки Big Data: наявністю неструктурованих даних. Відсутність стандартного представлення набору даних на веб-ресурсі призводить до необхідності аналізу кожної веб-сторінки окремо, що значно ускладнює процес пошуку інформації. Нова концепція розвитку мережі Internet – семантичний веб – впроваджується організацією W3C для стандартизації формату представлення даних на веб-ресурсі [3]. Стандартизований формат представлення даних дозволить здійснювати машинну обробку даних, розміщених на веб-сайтах, незалежно від структури веб-сайту та розмітки веб-сторінок. Однак процес впровадження метаданих на веб-ресурсах ще знаходиться на проміжній стадії: лише 40 % веб-сайтів використовують мікророзмітку, RFDa, JSON-LD тощо [4]. Тому для створення нових застосувань і сервісів, призначених здійснювати автоматизоване збирання та аналіз даних, актуальною залишається проблема пошуку необхідної інформації серед сукупності неструктурованих даних. Зрозуміло, що потужність отриманої множини може виявитися занадто великою для того, щоб користувач знайшов необхідну для нього інформацію серед веб-документів самотужки. Потрібно виконати впорядкування елементів множини у такий спосіб, щоб отримати тільки *корисну інформацію* (дані, які цікавлять користувача, що здійснює пошук). Наприклад, якщо користувач шукає товари, то його цікавлять назва продукту, ціна, габарити та інше. Потрібно зазначити, що корисна інформація є унікальною для кожної HTML-сторінки, сформованої спільним сценарієм, але різними наборами даних. Тобто користувача цікавить саме набір даних, за допомогою якого генерується веб-документ: йому непотрібна спільна для всіх сторінок інформація про меню сайту чи його розробників. Задачу пошуку корисної інформації серед HTML-сторінок можна трактувати як *задачу знаходження (екстракції) наборів змінних даних*, за допомогою яких серверним сценарієм генерується веб-документ.

Далі в роботі аналізуються основні методи екстракції змінних значень з множини HTML-сторінок, сформованих спільним сценарієм, але різними наборами даних; здійснюється перевірка ефективності роботи певного методу на множині HTML-документів різних веб-сайтів.

Аналіз методів екстракції змінних значень з веб-сторінок

На цей день існують різні методи отримання даних з веб-сторінки. Вихідним результатом роботи кожного методу є шаблон, відповідний вхідним документам. Залежно від ступеню автоматизації (фактору впливу користувача) процесу формування шаблону, методи екстракції даних з веб-сторінок поділяють на 4 категорії [5]:

1) методи екстракції даних власноруч;
2) контрольовані методи;
3) напівконтрольовані методи;
4) неконтрольовані методи.

У випадку *екстракції даних власноруч* користувач формує шаблон з урахуванням об’єктної моделі HTML-сторінки самостійно. *Контрольовані* методи здійснюють генерацію шаблону за допомогою аналізу набору маркованих веб-сторінок з передбачуваним вихідним результатом (навчання з вчителем). *Напівконтрольовані* методи використовують водночас набори маркованих і немаркованих прикладів (різновид навчання з вчителем). *Неконтрольовані* методи аналізують виключно немарковані веб-сторінки, які класифікуються автоматично (навчання без вчителя). На рис. 1 [5] показано методи екстракції даних з веб-сторінки згідно з вище розглянутою класифікацією.

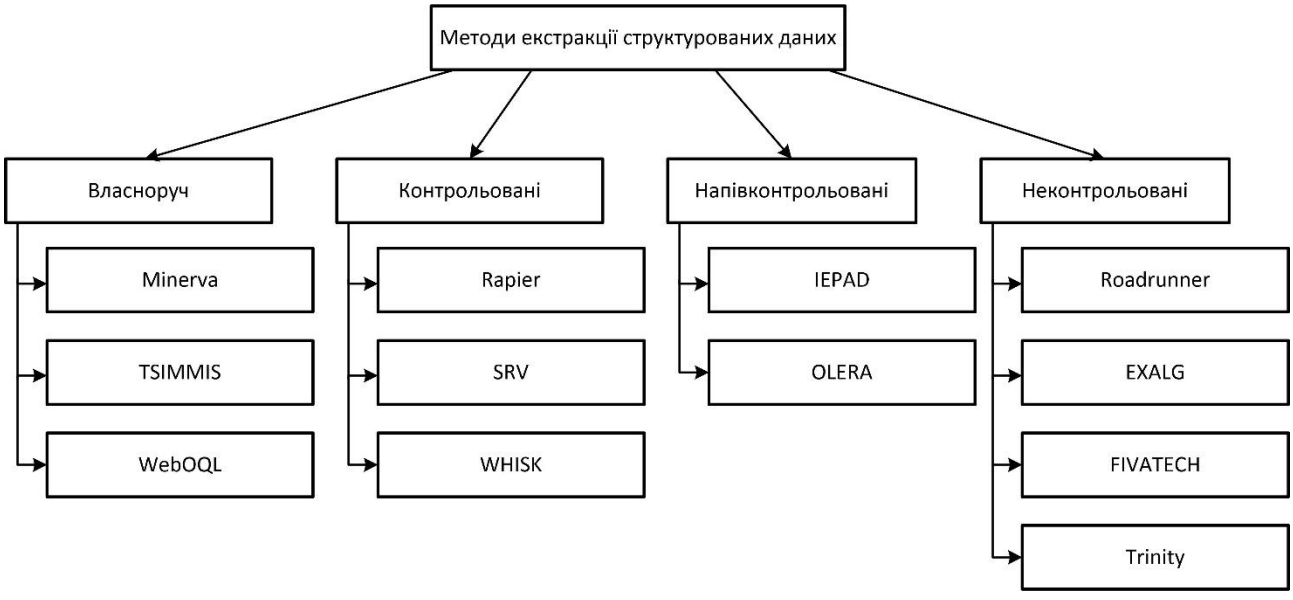

Рис. 1. Класифікація методів екстракції інформації з веб-сторінок

Варто звернути увагу на неконтрольовані методи отримання даних, для яких не потрібно аналізувати об'єктну модель HTML-сторінки чи формувати маркований і немаркований набори даних. Розглянемо детальніше принципи роботи певних неконтрольованих методів.

Roadrunner. Метод Roadrunner був запропонований у 2001 році [6]. Виявлення шаблону сторінки здійснюється за допомогою аналізу відповідності різних веб-сторінок одна одній. В аналізі використовується підхід, який автори Roadrunner називають ACME (Align, Collapse under Mismatch, and Extract). Застосування ACME можливе одночасно тільки для двох документів: зразку та обгортки. ACME намагається сформувати регулярний вираз, відповідний обгортці та зразку, аналізуючи невідповідності між ними. Невідповідність з'являється тоді, коли певний символ зразка не відповідає правилам розмітки обгортки. У такому разі ACME намагається вирішити проблему шляхом узагальнення розмітки обгортки за допомогою регулярних виразів. Результат роботи алгоритму вважається успішним, якщо отримана обгортка (регулярний вираз) відповідає зразку.

Основним недоліком методу Roadrunner є можливість одночасного порівняння тільки двох документів. Таким чином, сформований шаблон з великою ймовірністю може виявитися невідповідним іншим HTML-сторінкам, тобто шаблон не буде універсальним. Також варто зазначити, що розмітка вхідних документів методу Roadrunner повинна бути сформована згідно зі специфікацією XHTML, що вимагає додаткової попередньої обробки HTML-сторінок.

FiVaTech. Принцип роботи методу FiVaTech [7] полягає в аналізі об'єктної моделі HTML-сторінки для виявлення постійних та змінних елементів розмітки. FiVaTech використовує різні моделі представлення структурованих даних – базовий тип, кортежі, множини. Вхідними даними методу є об'єктні моделі веб-сторінок, представлені у вигляді дерев. Метод FiVaTech здійснює злиття всіх вхідних дерев в одне дерево, яке називається шаблонним. Процес злиття здійснюється за допомогою 4 послідовних операцій:

1) розпізнавання однорангових вузлів;
2) вирівнювання матриці однорангових вузлів;
3) знаходження повторюваних елементів;
4) злиття додаткових вузлів.

У шаблонному дереві розпізнаються змінні листкові вузли для базового типу даних та вузли, які мають властивість повторюваності, для множинного типу. Результуюче шаблонне дерево використовується для формування регулярного виразу, який відповідає всім вхідним HTML-сторінкам.

Алгоритм FiVaTech працює тільки з тими вхідними документами, які сформовані згідно зі специфікацією XHTML, адже документ розглядається як структурована DOM-модель. Тому необхідна попередня обробка веб-сторінок для корегування неправильно сформованої розмітки, що дещо сповільнює роботу алгоритму та потребує використання сторонніх інструментів вирівнювання розмітки.

Trinity. Метод Trinity [8], на відміну від алгоритму FiVaTech, розглядає вхідні документи як рядки, а не DOM-дерева. Trinity базується на гіпотезі, що спільні частини вхідних рядків не містять змінні значення, тому належать до загального шаблону. Знаходження загального шаблону відбувається за допомогою побудови тернарного дерева з вхідних рядків. Вхідні рядки формують кореневий вузол дерева. Спочатку здійснюється пошук максимально довгих спільних частин рядків. Коли така частина рядків знайдена, відбувається розбиття вхідних документів на префікс, розділювач і суфікс згідно з виявленим елементом шаблону. Спільна частина рядків записується як значення вузла; префікс, розділювач та суфікс стають його дочірніми вузлами. Створені нові вузли дерева аналізуються у рекурсивний спосіб до тих пір, поки не залишиться жодного спільного шаблону серед рядків вузла.

Для формування шаблону здійснюється прямий (префіксний) обхід створеного дерева. Шаблон формується зі значень вузлів та елементів регулярних виразів. Всі змінні значення позначаються мітками. Автори методу пропонують подальше спрощення отриманого регулярного виразу за допомогою трансформації виразу в детермінований скінченний автомат, його мінімізації та оберненого перетворення автомату в регулярний вираз.

На рис. 2 [8] показано тернарне дерево, сформоване для трьох вхідних документів. Вхідний набір HTML-сторінок зберігається у вершині *N1*. Спочатку здійснюється пошук найдовшого спільного шаблону, який на рис. 2 підкреслено лінією, серед рядків вершини *N1*. У випадку знаходження спільного шаблону створюються три вершини з префіксами (*N2*), розділювачами (*N3*) і суфіксами (*N4*). Варто зазначити, що спільні шаблони знайдено на початку вхідних документів вершини *N1*, тому префікси є пустими рядками і відповідно позначаються символом *e*. Всі спільні шаблони зустрічаються тільки одного разу, тому розділювачі відсутні і позначаються як *nill*. Обробка даних відбувається далі на створених вершинах у рекурсивний спосіб. Ознакою закінчення побудови дерева є відсутність вузлів, які містять рядки зі спільними фрагментами тексту.

Побудова спільного шаблону здійснюється шляхом прямого проходу дерева. При розгляді листкового елемента дерева генерується мітка (capturing group), яка вказує на наявність змінної інформації

(наприклад, мітка створюється для вершин *N11* чи *N17,* показаних на рис. 2). В іншому випадку повертається спільний шаблон вузла та інколи оператор, який описує тип шаблону: додатковий чи такий, що має властивість повторюваності. Далі наведений шаблон, сформований внаслідок обходу вище розглянутого дерева.

```
<html><head><title>Results</title></head><body><h1>Results:</h1>
{A}<br/><b>(({B}</b><br/>{C}<br/><br/>{D})?<br/><b>)*
{E}</b><br/>{F}<br/></body></html>
```

Літерами у фігурних дужках позначені мітки; символ ? означає, що вказаний фрагмент не є обов'язковим, тобто може бути відсутнім у вхідних HTML-сторінках; символ * вказує на можливість повторного послідовного використання цієї частини шаблону.

В таблиці 1 наведена порівняльна характеристика вище розглянутих методів екстракції інформації з веб-сторінок.

Таблиця 1. Порівняння методів екстракції інформації з веб-сторінок

Метод	Кількість документів для формування шаблону	Тип вхідних даних	Часова складність
Roadrunner	2	DOM-дерево	Поліноміальна
FiVaTech	2 і більше	DOM-дерево	–
Trinity	2 і більше	Рядок	Поліноміальна

Для програмної реалізації екстракції даних з веб-сторінок доцільно обрати метод Trinity. Головною перевагою методу Trinity є можливість роботи з HTML-документами, які мають некоректно сформовану розмітку, що дозволяє уникнути попередньої обробки вхідних документів. Крім того, метод Trinity має графічне представлення інформації – тернарне дерево, тому дозволяє відслідкувати процес знаходження спільних елементів шаблону та проаналізувати хід процесу формування загального шаблону.

Проблемою застосування будь-якого з наведених вище методів екстракції даних з веб-сторінок є процес формування колекції HTML-документів з множини вхідних документів для побудови шаблону. Шаблон повинен бути універсальним, тобто відповідати всім варіантам веб-сторінок, сформованих спільним сценарієм, але різними наборами даних. Тому до колекції документів мають входити веб-сторінки, які містять елементи розмітки чи навіть фрагменти тексту, унікальні відносно інших веб-сторінок. Наприклад, розглянемо наступні фрагменти вхідних HTML-документів, які описують інформацію про випуск наукового журналу:

```
<b>Data Rec., Storage & Processing. — 2017. — Vol. 19, N 1.</b>
<b>Data Rec., Storage & Processing. — 2016. — Vol. 1, N 2.</b>
<b>Data Rec., Storage & Processing. — 2016. — Vol. 2, N 3.</b>
```

Фрагмент сформованого шаблону для наведених вище рядків виглядатиме так:

```
<b>Data Rec., Storage & Processing. — 201{A}. — Vol. {B}, N {C}.</b>,
```

де A, B, C – мітки, які описують змінне значення. Але такий шаблон не відповідатиме рядку, який міститиме інформацію про випуск журналу 2005 року:

```
<b>Data Rec., Storage & Processing. — 2005. — Vol. 1, N 1.</b>
```

Тобто, вибір вхідних документів для формування шаблону є важливим кроком перед початком роботи будь-якого з наведених вище методів, адже саме різноманітність розмітки та даних веб-сторінок дозволить сформувати узагальнений регулярний вираз, відповідний всім можливим варіантам досліджуваної веб-сторінки.

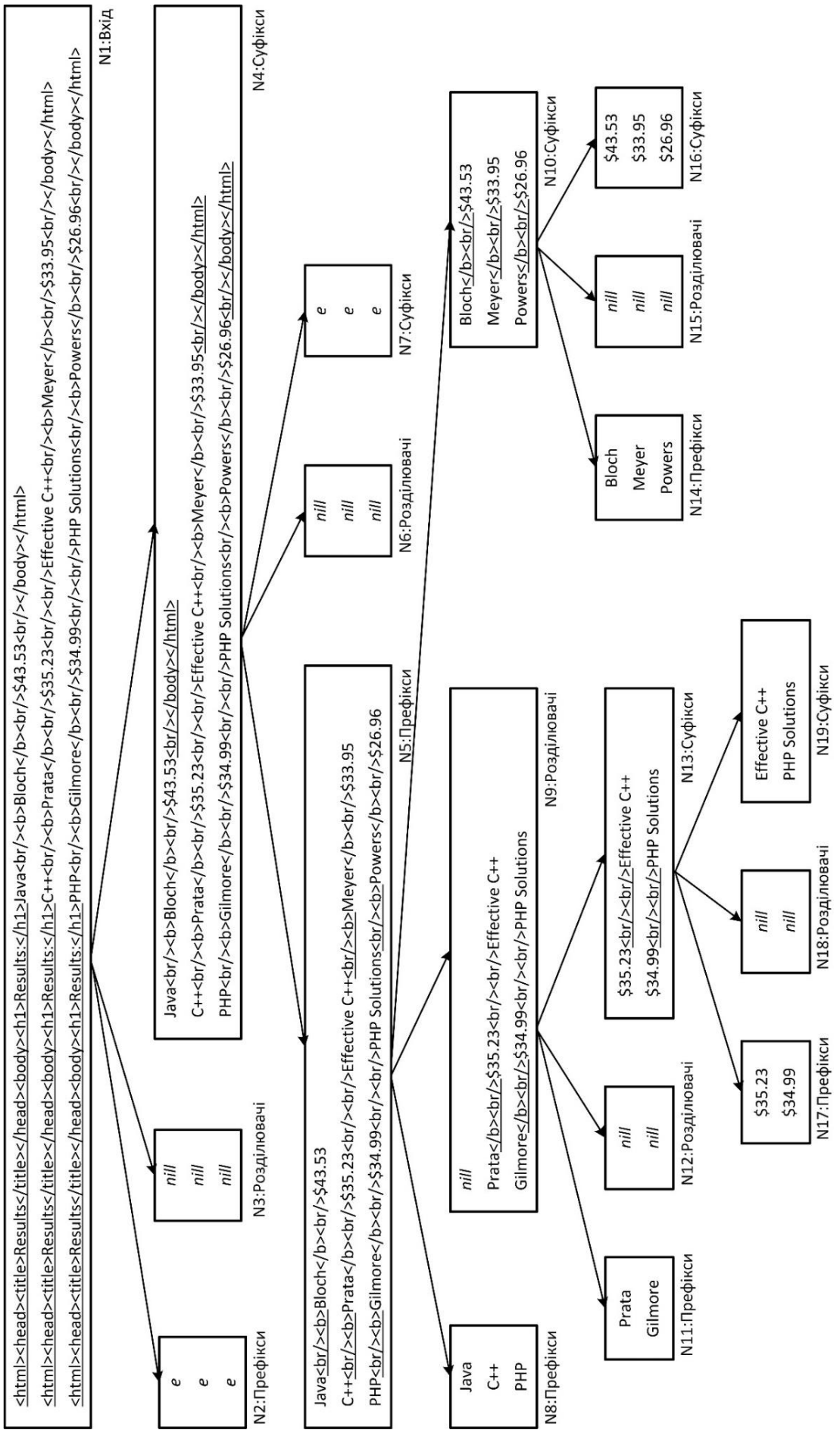

Рис. 2. Приклад побудови тернарного дерева методом Trinity

Експериментальна перевірка функціонування методу Trinity

Для програмної реалізації методу Trinity та перевірки його функціонування було створено застосування. Предметом аналізу обрано веб-сторінки статей різних українських наукових журналів з метою отримання корисної інформації про статтю: назва, автори, анотація тощо. Завданням застосування є автоматизоване збирання інформації про статті з веб-сайтів журналів, отримання набору даних з веб-сторінок за допомогою методу Trinity та перевірка коректності отриманих результатів.

Приклад роботи застосування. Розглянемо на прикладі обробки журналу «Реєстрація, зберігання і обробка даних» [9] основні етапи роботи застосування.

*Отримання множини веб-сторінок статей***.** На першому кроці здійснюється автоматизоване збирання колекції HTML-документів, сформованих спільним сценарієм, але різними наборами даних. Пошуковий робот здійснює обхід сайту та зберігає вихідний код веб-сторінок статей як локальні текстові документи. Нижче наведений приклад обходу сайту для пошуку даних про статтю.

```
Архів журналу – Список робіт за 2017 рік – Defining the relative expert
competence during aggregation of pair-wise comparisons

Архів журналу – Список робіт за 2017 рік – Features for providing of cyber
security
…
Архів журналу – Список робіт за 2005 рік – The Analysis of Potentialities for
Creating High-Resolution Recording Systems for Master Disks
```

*Генерація вхідного набору веб-сторінок***.** Серед елементів отриманої множини веб-сторінок відбувається відбір документів, які будуть формувати вхідний набір даних методу Trinity. Вибір документів здійснюється з урахуванням статей журналу різних років та випусків для кращого узагальнення шаблону.

```
Defining the relative expert competence during aggregation of pair-wise
comparisons (2017 — Т. 19, № 2)

A mathematical apparatus for data analysis for forecasting nonlinear non-
stationary processes (2017 — Т. 19, № 1)
…
A method for the optimisation of algorithms for deciding of linear equation
systems with a corrupted right part over the residue ring modulo 2N
(2005 — Т. 7, № 1)
```

*Формування шаблону***.** Отриманий вхідний набір веб-сторінок аналізується методом Trinity. Після побудови тернарного дерева здійснюється його обхід для формування шаблону. Шаблон представлений у вигляді регулярного виразу, де мітки (літери, які позначають змінні значення) описані відповідними операторами. Нижче наведений фрагмент сформованого регулярного виразу (мова програмування Java).

```
<tr><td width="100">Annotation</td><td>\E(.+)\Q</td></tr>

<tr><td width="100">Key words</td><td>\E(.+)\Q</td></tr>
```

Екстракція змінних значень. Сформований регулярний вираз застосовується до всіх елементів множини веб-сторінок. Результатом застосування регулярного виразу до HTML-документу є колекція рядків. Розглянемо приклад результату екстракції даних з веб-сторінки.

```
1) Matov;
2) 06. — Vol. 8, N 4;
3) 64-74;
4) width="25px" alt="">PDF,DOC;
5) Generalized Noise Combating Codes in the Tasks of Providing Integrity of
   Information Hold-ing Objects in the Conditions of Natural Factors
   Influence;
```

6) Matov O.Ya., Vasylenko V.S.;

7) Generalized noise combating codes for the use in the tasks of providing integrity of information holding objects in the conditions of natural factors influence are proposed. Fig.: 3. Refs.: 4 titles.;

8) distortion discovery, distortion correction, integrity control, noise combating correcting codes.;

9) References</td><td>1. Матов О.А., Василенко В.С. Узагальнені завадостійкі коди в задачах забезпечення цілісно-сті інформаційних об’єктів. Код умовних лишків. // Реєстрація, зберігання і оброб. даних. — 2006. — Т. 8, № 3. — С. 48–66.
2. Дубровский В.В. CDMA — взгляд глазами профессионала // mailto:v_dubrovskii@mail.ru.
3. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. — М.: Сов. радио, 1966. — 421 с.
4. Василенко В.С., Будько М.М., Короленко М.П. Контроль и відновлення цілісності інформації в автоматизованих системах // Правове, нормативне та метрологічне забезпечення Системи захисту інформації в Україні. — К.: НТУУ «КПІ», 2002. — Вип. 4. — С. 119–128.;

10) File;

11) stattja.doc;

12) null.

Серед отриманих даних варто зазначити назву статті, авторів, анотацію і ключові слова (рядки № 5–8). Інші рядки можуть не мати змістовного навантаження для користувача без контексту. Наприклад, рядок № 2 описує рік та номер журналу, а рядок № 3 вказує на сторінки розміщення статті у журналі.

Отримання інформації про статті власноруч. Для перевірки коректності результатів, отриманих методом Trinity, здійснюється екстракція даних з веб-сторінок шляхом обробки об’єктної моделі HTML-документу. Внаслідок аналізу розмітки веб-сторінки створюється алгоритм пошуку атрибутів статті у DOM-дереві документа для поточного веб-сайту.

Оцінка похибки функціонування методу Trinity. Атрибути статей, отримані шляхом застосування методу Trinity та аналізу об’єктної моделі HTML-сторінки, порівнюються між собою. У випадку невідповідності одного з атрибутів статті фіксується випадок помилкової екстракції даних. Похибка розраховується як відношення кількості помилково розпізнаних статей до загальної кількості вхідних документів.

Створене застосування реалізовано у вигляді двох сценаріїв, які здійснюють наступні функції:

1) автоматизоване збирання веб-сторінок статей з веб-сайтів наукових журналів;

2) екстракція змінних значень з множини HTML-документів за допомогою методу Trinity та оцінка отриманих результатів.

Консольні сценарії реалізовано мовою програмування Java (версія SE 8). Розглянемо більш детально роботу кожного зі сценаріїв.

Сценарій 1. Автоматизоване збирання веб-сторінок. Автоматизоване збирання веб-сторінок здійснено за допомогою реалізації пошукового роботу для сайтів наукових журналів. Алгоритм обходу для кожного сайту є унікальним. Робот починає пошук з початкової HTML-сторінки, заданої користувачем. На сторінці, яка розглядається роботом як DOM-дерево, здійснюється пошук необхідних посилань на сайті. Знайшовши посилання, робот переходить до наступних сторінок; якщо робот потрапив на сторінку статті, він здійснює мінімізацію отриманого HTML-документу та зберігає вміст документу до текстового файлу.

Пошуковим роботом було здійснено пошук англомовних веб-сторінок статей журналів. В таблиці 2 наведений результат роботи пошукового роботу для сайтів наукових журналів. Для обробки HTML-документів використано бібліотеку Jsoup [11], яка містить набір інструментів для пошуку елементів об’єктної моделі веб-сторінки. Пошук здійснюється за допомогою використання CSS-селекторів та інструментів обходу DOM-дерева. Також за допомогою бібліотеки Jsoup здійснено мінімізацію документів для пришвидшення роботи наступного сценарію.

Таблиця 2. Результат роботи пошукового роботу

Назва журналу	Наявність англомовної версії	Початковий рік статей на сайті	Кінцевий рік статей на сайті	Кількість знайдених статей
Реєстрація, зберігання і обробка даних	Так	2005	2017	517
Системні дослідження та інформаційні технології [10]	Так	2008	2017	478

Сценарій 2. Екстракція змінних значень з множини HTML-документів. Для формування шаблону обрано 20 % вхідних документів. Попередньо множину вхідних документів було впорядковано за датою та номером випуску журналів. Вибір вхідних документів методу Trinity здійснено рівномірно по впорядкованій множині веб-сторінок. Такий підхід обумовлений необхідністю врахування особливостей розмітки та даних веб-сторінок статей всіх років для кращого узагальнення шаблону.

Результатом застосування сценарієм методу Trinity до набору вхідних документів є шаблон (регулярний вираз). Отриманий шаблон сформований згідно з синтаксисом регулярних виразів у мові програмування Java. Для роботи з регулярними виразами використано пакет java.util.regex.

Далі застосовано шаблон до множини всіх статей, отриманих з веб-версії журналу. Результатом застосування шаблону до веб-сторінки є набір змінних значень. У випадку невідповідності шаблону веб-сторінці набір таких значень відсутній. Результати екстракції даних з веб-документів були експортовані до бази даних MySQL.

Наступним кроком є здійснення перевірки коректності рядків, які відповідають інформації про статтю:

1) назва;
2) автори;
3) анотація;
4) ключові слова.

Для перевірки коректності наведених вище рядків було створено окремий програмний компонент сценарію. Функцією програмного компонента є отримання атрибутів статті за допомогою аналізу DOM-дерева різних веб-сторінок. Для роботи з DOM-деревом використано бібліотеку Jsoup. Пошук атрибутів статті здійснено за допомогою CSS-запитів та обходу об'єктної моделі веб-сторінки. Отримані структуровані дані про статті було експортовано до бази даних MySQL. Результат роботи програмного компоненту було порівняно з вихідними даними методу Trinity. Стаття вважалася нерозпізнаною у випадку невідповідності щонайменше одного з атрибутів. Порівняння результатів було здійснено за допомогою виконання відповідних SQL-запитів. Результати аналізу отриманих даних наведено в таблиці 3.

Таблиця 3. Результати аналізу екстрагованої інформації

Назва журналу	Кількість змінних значень на сторінці	Загальна кількість статей	Відсоток статей, які не відповідають шаблону	Відсоток помилково розпізнаних статей
Реєстрація, зберігання і обробка даних	12	517	2.5 %	1.6 %
Системні дослідження та інформаційні технології	11	478	0 %	3.7 %

Відсоток статей, які не відповідають шаблону, дозволяє зробити висновок про успішність вибору вхідного набору веб-сторінок для формування регулярного виразу. Ненульові показники цього значення (2.5 %) означають, що серед вхідних веб-сторінок методу Trinity були відсутні деякі документи з унікальними елементами розмітки чи тексту. Врахування цих документів для створення регулярного виразу дозволить сформувати більш універсальний шаблон та зменшити наведений вище показник невідповідності.

Показники помилково розпізнаних статей вказують на похибку роботи методу Trinity та якість вибору документів для формування шаблону. Невідповідність результатів методу Trinity фактичній інформації про статтю обумовлена наявністю в отриманих атрибутах статті певних додаткових елементів розмітки, що може вказувати на некоректність розмітки HTML-документа чи особливості генерації веб-сторінок на сервері.

Висновки

Проаналізовано методи екстракції змінних значень з веб-сторінок і обґрунтовано ефективність (з точки зору узагальнення сформованого шаблону) застосування методу Trinity. Обмеженням методу Roadrunner є можливість формування шаблону тільки з двох документів, а для коректної роботи методу FiVaTech необхідне попереднє вирівнювання вхідних документів згідно зі специфікацією XHTML. На відміну від двох останніх метод Trinity може працювати з довільною кількістю HTML-документів, в тому числі з тими, які мають некоректно сформовану розмітку. Крім цього, шаблон, сформований методом Trinity, має коректну розмітку незалежно від структури вхідних документів, що дозволяє здійснювати подальшу обробку шаблона як об'єктної моделі HTML-сторінки.

Отримано експериментальні результати застосування методу Trinity до веб-сторінок статей українських наукових журналів і показано можливість використання цього методу для екстракції структурованих даних з HTML-документів та їх подальшої обробки. Зменшення похибки роботи методу можливе за рахунок покращення алгоритму вибору вхідного набору HTML-документів для генерації шаблону.

Література

1. Potebnia A., Pogorilyy S. Innovative GPU accelerated algorithm for fast minimum convex hulls computation. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 2015. Vol. 5. P. 555–561.
2. Pogorilyy S., Shkulipa I. A Conception for Creating a System of Parametric Design of Parallel Algorithms and their Software Implementations. *Cybernetics and System Analysis.* 2009. Vol. 54, No. 6. P. 952–958.
3. Usage of structured data formats for websites [Електронний ресурс]. Дата оновлення: 05.07.2017. Режим доступу: https://w3techs.com/technologies/overview/structured_data/all (дата звернення: 01.02.2018).
4. Semantic Web [Електронний ресурс]. Режим доступу: https://www.w3.org/standards/semanticweb (дата звернення: 12.02.2018).
5. Patel D., Thakkar A. A Survey of Unsupervised Techniques for Web Data Extraction. *International Journal Of Computer Science*. 2015. Vol. 6, No. 2. P. 1–3.
6. Crescenzi V., Mecca G., Merialdo P. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. *VLDB 2001*: Proceedings of the 27th International Conference on Very Large Data Bases, Rome, Italy, September 11–14, 2001. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2001. P. 109–118.
7. Kayed M., Chang C.-H. FiVaTech: Page-level web data extraction from template pages. *IEEE Transactions on Knowledge and Data Engineering*. 2010. Vol. 22, No. 2. P. 249–263.
8. Sleiman H.A, Corchuelo, R. Trinity: On Using Trinary Trees for Unsupervised Web Data Extraction. *IEEE Transactions on Knowledge and Data Engineering*. 2014. Vol. 26, No. 6. P. 1544–1556.
9. Реєстрація, зберігання і обробка даних [Електронний ресурс]. Режим доступу: http://www.ipri.kiev.ua/index.php?id=52 (дата звернення: 03.01.2018).
10. Системні дослідження та інформаційні технології [Електронний ресурс]. Режим доступу: http://journal.iasa.kpi.ua (дата звернення: 10.01.2018).
11. Jsoup: Java HTML Parser [Електронний ресурс]. Режим доступу: https://jsoup.org/apidocs/overview-summary.html (дата звернення: 11.01.2018).

References

1. POTEBNIA, A. AND POGORILYY, S. (2015) Innovative GPU accelerated algorithm for fast minimum convex hulls computation. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 5. p. 555–561.
2. POGORILYY, S. AND SHKULIPA, I. (2009) A Conception for Creating a System of Parametric Design of Parallel Algorithms and their Software Implementations. *Cybernetics and System Analysis.* 54 (6). p. 952–958.
3. WORLD WIDE WEB CONSORTIUM (2018) *Semantic Web*. [Online] Available from: https://www.w3.org/standards/semanticweb [Accessed: 12 February 2018].
4. W3TECHS – WEB TECHNOLOGY SURVEYS (2017) *Usage of structured data formats for websites.* [Online] Available from: https://w3techs.com/technologies/overview/structured_data/all [Accessed: 1 February 2018].
5. PATEL, D. AND THAKKAR, A. (2015) A Survey of Unsupervised Techniques for Web Data Extraction. *International Journal Of Computer Science*. 6 (2). p. 1–3.

6. CRESCENZI, V., MECCA, G., MERIALDO, P. (2001) *RoadRunner: Towards Automatic Data Extraction from Large Web Sites.* Proceedings of the 27th International Conference on Very Large Data Bases. Rome, Italy, 11–14 September 2001. San Francisco, CA: Morgan Kaufmann Publishers Inc.
7. KAYED, M. AND CHANG, C.-H. (2010) FiVaTech: Page-level web data extraction from template pages. *IEEE Transactions on Knowledge and Data Engineering*. 22 (2). p. 249–263.
8. SLEIMAN, H.A AND CORCHUELO, R. (2014) Trinity: On Using Trinary Trees for Unsupervised Web Data Extraction. *IEEE Transactions on Knowledge and Data Engineering*. 26 (6). p. 1544–1556.
9. INSTITUTE FOR INFORMATION RECORDING (2017) *Data Rec., Storage & Processing.* [Online] Available from: http://www.ipri.kiev.ua/index.php?id=52 [Accessed: 3 January 2018].
10. SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES (2017) *Archives.* [Online] Available from: http://journal.iasa.kpi.ua [Accessed: 10 January 2018].
11. JSOUP: JAVA HTML PARSER (2017) *jsoup Java HTML Parser 1.11.2 API*. [Online] Available from: https://jsoup.org/apidocs/overview-summary.html [Accessed: 11 January 2018].

Про авторів:

Погорілий Сергій Дем'янович,
доктор технічних наук, професор,
завідувач кафедри комп'ютерної інженерії
факультету радіофізики, електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 200.
Кількість наукових публікацій в зарубіжних виданнях – 25.
Індекс Хірша: (Google Scholar) – 6,
Індекс Хірша: (Scopus) – 2.
https://orcid.org/0000-0002-6497-5056.

Крамов Артем Андрійович,
аспірант факультету радіофізики, електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка.
https://orcid.org/0000-0003-3631-1268.

Місце роботи авторів:

Київський національний університет імені Тараса Шевченка.
03022, Київ, проспект Академіка Глушкова, 4Г.
Тел.: (044) 521 3559.
E-mail: sdp77@i.ua,
artemkramov@gmail.com