DOI: 10.26565/2304-6201-2020-47-03

УДК 004.9

Алгоритм побудови моделі веб-сайту

Н.А. Гук, С.В. Диханов, О.Д. Матющенко

Гук Наталія Анатоліївна
д.ф.-м.н., професор; завідувач кафедрою комп'ютерних технологій Дніпровський національний університет імені Олеся Гончара, Дніпро, пр. Гагаріна, 72, Дніпро, Україна, 49010
e-mail: natalyguk29@gmail.com;
https://orcid.org/0000-0001-7937-1039

Диханов Станіслав Віталійович
аспірант Дніпровський національний університет імені Олеся Гончара, Дніпро, пр. Гагаріна, 72, Дніпро, Україна, 49010
e-mail: dykhanovstas@gmail.com;
https://orcid.org/0000-0001-9073-0784

Матющенко Олег Дмитрович
студент Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова 2, Харків, Україна, 61002
e-mail: matyushchenkk@gmail.com;
https://orcid.org/0000-0003-3596-5737

У роботі для аналізу структури веб-сайту запропоновано використовувати модель у вигляді графа. Для побудови моделі у вигляді веб-графу розроблено метод, алгоритм сканування сторінок веб-ресурсу. За допомогою фреймворку Scrapy та мови Python побудовано програмне забезпечення та виконано сканування веб-ресурсів з метою побудови їх веб-графів. Для візуалізації отриманих графів та обчислення деяких метричних характеристик застосовано Gephi. Виконано сканування веб-сайтів, побудовано веб-графи, з використанням метричних характеристик здійснено аналіз структурної зв'язності.

Ключові слова: *веб-сайт, веб-граф, краулинг, Scrapy, Gephi.*

Algorithm for building a website model

N. Huk, S. Dykhanov, O. Matiushchenko

Huk Natalia
Doctor Sciences in Physics and Mathematics, Professor; Head of the Department of Computer's Science
Oles Honchar Dnipro National University, 72, Gagarin Avenue, Dnipro, Ukraine, 49010

Dykhanov Stanislav
PhD student
Oles Honchar Dnipro National University, 72, Gagarin Avenue, Dnipro, Ukraine, 49010

Matiushchenko Oleh
Student
National Technical University «Kharkiv Polytechnic Institute» 2, Kyrpychova str., Kharkiv, Ukraine, 61002

The analysis of the structure of the website modeling has been carried out. The models of Internet space representation in the form of semantic networks, frame structures and ontology have been analyzed. The web graph model has been chosen to represent the web resource. The pages of a web resource are connected by hyperlinks, which form the internal structure of the resource. To build a model of a website in the form of a web graph, a method and algorithm for scanning the pages of a web resource have been developed. The web resource scanning is performed by in depth searching with the LIFO (Last In - First Out) method. Links are searched by sorting the lines of the page markup text and extracting links by using regular expressions. Only links to pages within the resource are taken into account in the search process, external links are ignored. The crawling procedure is implemented by using the Scrapy framework and the Python. To account for the presence of additional filters used to select pages with criteria, the rules for selecting URL in HTML code have been strengthened. Web resources are scanned to build their web graphs. Storing information by using a list of edges and an adjacency matrix is used in further work with the obtained web graphs. To visualize the obtained graphs and calculate some metric characteristics, the Gephi software

environment and the algorithm for stacking the vertices of the Yifan Hu graph has been used. The graph diameters, the average vertex degree, the average path length, the density factor of the graph are used for analysis of the structural connectivity of the graphs studied. The proposed approach can be applied during the site reengineering procedure.

Keywords: website, web graph, crawling, Scrapy, Gephi.

Алгоритм построения модели веб-сайта

Н.А. Гук, С.В. Дыханов, О.Д. Матющенко

Гук	*д.ф.-м.н., профессор; заведующая кафедрой компьютерных технологий*
Наталья	*Днипровский национальный университет имени Олеся Гончара, пр.*
Анатольевна	*Гагарина, 72, Днипро, Украина, 49010*
Дыханов	*аспирант*
Станислав Витальевич	*Днипровский национальный университет имени Олеся Гончара, пр. Гагарина, 72, Днипро, Украина, 49010*
Матющенко	*студент*
Олег Дмитриевич	*Национальный технический университет «Харьковский политехнический институт», ул. Кирпичева 2, Харьков, Украина 61002*

В работе для анализа структуры сайта предложено использовать модель в виде графа. Для построения модели в виде веб-графа разработан метод, алгоритм сканирования страниц веб-ресурса. С помощью фреймворка Scrapy и языка Python построено программное обеспечение и выполнено сканирование веб-ресурсов с целью построения их веб-графов. Для визуализации полученных графов и вычисления некоторых метрических характеристик используется Gephi. Выполнено сканирование веб-сайтов, построені веб-графы, с использованием метрических характеристик осуществлен анализ структурной связности.

Ключевые слова: *веб-сайт, веб-граф, краулинг, Scrapy, Gephi*

1 Вступ

Процес створення або реконструкції будь-якого веб-сайту починається з аналізу його структури. Грамотне проектування розділів, підрозділів і кінцевих сторінок робить сайт зручним та зрозумілим для користувача, а також покращує помітність ресурсу у мережі Інтернет для пошукових роботів, які формують видачу за запитами користувачів. Тому при розробці та реінжинірингу існуючих сайтів завжди намагаються приділити увагу логіці подачі матеріалу.

Якщо ресурс має великий обсяг та складну ієрархію розділів, і грамотно не структурований, то перебування на такому сайті і пошук інформації стають проблематичними для більшості користувачів, також ускладнюється і індексація ресурсу пошуковими роботами.

Для усунення можливих помилок в організації і логіці структури ресурсу, при проведенні внутрішнього та технічного аудитів виконується аналіз структури сайту.

Для проведення подібного аналізу використовуються моделі, за допомогою яких зображення складної структури стає зрозумілим.

Моделі веб-сайту можуть слугувати аналітичним інструментом, пояснювальним інструментом, а також інструментом для прогнозування подальшого розвитку ресурсу [1], тому питання розробки моделей та алгоритмів їх побудові є актуальною задачею.

2 Огляд літератури

На сьогодні існує декілька моделей, за допомогою яких можна зображувати інтернет-ресурси.

Онтології [2] є ефективним засобом представлення та систематизації всіх видів інформації про модель предметної області. Вони використовуються для формалізації, систематизації та збору знань і даних, надають можливість проводити обробку та аналіз отриманих структур, а також забезпечують організацію зручного змістовного доступу до об'єктів. За допомогою онтологій існує можливість надати умовні характеристики об’єкту (атрибути) для понять предметної області, встановлювати семантичні взаємозв’язки (відношення) між елементами, заповнювати побудовану онтологію екземплярами. Редактори онтологій та даних здійснюють процес налаштування та управління системою знань і контентом.

У роботі [3] наведено застосування онтології для інтернет-технологій, побудовано моделі на мові OWL (Ontology Web Language) та показано, що подібні моделі можна застосовувати для представлення знань та управління ними. Також на основі онтологій можна створювати зручну навігацію по інтернет-ресурсам, адекватну інтерпретацію змісту текстових документів та пошукових запитів.

Однак побудова онтологій для веб-ресурсів є трудомістким процесом, який складно застосовувати для оперативного аналізу окремих сайтів.

За допомогою семантичної мережі існує можливість отримати модель веб-сайту в вигляді орієнтованого графу, вершинами якого є сторінки ресурсу, а дуги графа можуть відображати семантичні відношення між сторінками. Взагалі семантичні відношення розділяються на лінгвістичні, за допомогою яких встановлюється змістовний зв'язок між об'єктами, логічні відношення, за допомогою яких існує можливість обчислювати висловлювання, теоретико-множинні відношення, які встановлюють зв'язок типу «частина-ціле», та квантифіковані відношення, які використовують квантори узагальнення та існування.

Однак таку модель краще використовувати для інформаційного пошуку, побудованого з використанням семантичного значення змісту або інформації про сторінку.

Зображення веб-сайтів у вигляді фреймової моделі наведено у роботі [4], де модель сайту представлено у вигляді мережі фреймів, що містять інформацію про сторінки, набори їх властивостей та зв'язки між ними.

Використання подібної моделі є зручним для організації пошуку, порівняння сайтів між собою, виявлення однотипних сторінок та структур посилань.

Перелічені моделі найчастіше застосовуються, коли необхідно проаналізувати не лише структуру, а і змістовні зв'язки між сторінками, а також для набуття нових знань методами Web Data Mining [5].

Для оперативного аналізу структури сайтів та фрагментів веб-простору у якості моделей найчастіше використовуються веб-графи [6-10]. За допомогою такої моделі зручно відстежувати зв'язки між сторінками, аналізувати поведінку користувачів при переміщенні по сайту [11, 12], відокремлювати схожі за ознаками сегменти веб-простору.

Для побудови такої моделі веб-сайту шляхом отримання інформації про гіперпосилання сторінок у мережі Інтернет застосовується процедура веб-краулінгу. Основною задачею краулінгу є організований обхід веб-простору та збір інформації про гіперпосилання сторінок. Основні принципи роботи краулеров наведено у роботах [13-15].

Програми-краулери розділяють на універсальні та спеціалізовані. Універсальні здійснюють обхід веб-простору з використанням посилань, аналізують сторінки на наявність необхідної інформації, процес виконується до тих пір, поки не буде виконане обмеження на кількість проаналізованих сторінок або не досягнуто мети пошуку. Вони найчастіше є надлишковими та вимагають значної кількості обчислювальних ресурсів, програють у продуктивності спрямованого інформаційного пошуку за заданою метрикою значущості.

Спеціалізовані краулери розробляються для збору заздалегідь визначеного типу інформації, наприклад, для дослідження сайтів організацій, які працюють у одному сегменті ринку. В роботах [16-18] наведено огляд існуючих веб-краулерів з відкритим вихідним програмним кодом та безкоштовних розповсюджених сервісів для здійснення процедури краулінгу, інструментів для збору та аналізу даних з сайтів. Розглянуті інструменти класифіковано відповідно до необхідних для їх роботи ресурсів та обмежень щодо виконання спеціалізованих завдань.

В роботі [19] запропоновано модель веб-краулера, який складається з узагальненого ядра та спеціалізованих додатків, додатки налаштовуються для здійснення узагальненого пошуку по гіперпосиланням, тематичного пошуку з аналізом текстів сторінок, пошуку по популярним сторінкам та інші типами пошуку.

З аналізу літературних джерел стає зрозуміло, що для проведення аналітичних досліджень веб-простору та його сегментів можливе застосування результатів розширеного пошуку, здійсненого відомими пошуковими системами, використання відкритих баз даних, також можна здійснювати закупівлю вебометричних даних у аналітичних компаній, однак значну перевагу має розробка власних методів та алгоритмів збору даних про досліджувані об'єкти, оскільки це дозволяє налаштовувати інструменти збору даних під вимоги, які виникають під час проведення дослідження.

3 Постановка задачі

Для побудови веб-графу необхідно розробити метод сканування сторінок сайту (процедуру краулінгу), алгоритм та відповідне програмне забезпечення, за допомогою якого можна здійснювати обхід та сканування веб-сайту, будувати веб-граф у вигляді списку ребер та матриці суміжності. Результат сканування необхідно представити у зручному вигляді для подальшої обробки та візуалізації.

4 Математична модель web-сайту

Гіпертекстову модель веб-сайту H можна представити як набір, що складається з двох множин:

$$H = \{P, L\}, \tag{4.1}$$

де $P = \{p_1, ..., p_n\}$ – множина сторінок сайту; $L = \{l \mid \exists\, p_1, p_2 \in P : p_1, p_2 \in l(p_1, p_2)\}$ – множина гіперпосилань між сторінками.

Гіперпосилання є базовим елементом веб-простору, що зв'язує між собою веб-ресурси та веб-сторінки одного ресурсу. У найпростішому вигляді можна вважати, що гіперпосилання - це пара <URL-start, URL-end>, де URL-start є адресою веб-ресурсу (або веб-сторінки), з якого зроблено посилання. Це реалізує можливість переходу на веб-ресурс (веб-сторінку) з адресою URL-end. Зокрема, гіперпосилання, що з'єднують сторінки і документи одного веб-ресурсу, задають його внутрішню структуру та є внутрішніми посиланнями. Саме зв'язки між сторінками одного веб-ресурсу розглядаються у роботі.

Структурі гіпертекстової моделі веб-сайту відповідає математична модель у вигляді орієнтованого незваженого графа $G = (A, E)$, у якому $A = P$, $E = L$. У побудованому графі A – множина вершин, елементи якої описують сторінки сайту, E – множина ребер графу, елементи якої відповідають гіперпосиланням між сторінками.

5 Метод та алгоритм побудови веб-графа

Процес збору даних о гіпертекстової структурі ґрунтується на ідеях алгоритмів пошуку на графах з використанням локальної інформації. У структурі сторінки сайту необхідно знайти посилання на інші сторінки, зробити перехід по цім посиланням на інші сторінки, де знов необхідно здійснити пошук посилань.

Пошук посилань у веб-документі здійснюється шляхом перебору строк тексту розмітки сторінки та виділенні з них посилань за допомогою регулярних виразів, гіперпосилання вилучаються з html сторінки з тегів <a>. У процесі пошуку враховуються лише посилання на сторінки всередині ресурсу, зовнішні посилання ігноруються. Знайдені посилання зберігаються у в файл або базу даних в вигляді записів сторінка A_i – сторінка A_j (<URL-start, URL-end>).

У веб-графі це відповідає такій структурі: вершина A_i – ребро e_{ij} – вершина A_j. Процес виконується циклічно доки не буде досягнута одна з умов: виповниться обмеження на глибину пошуку або буде знайдено всі посилання для заданого веб-ресурсу. Додатково до процедури пошуку висуваються такі вимоги:

- ігноруються посилання з протоколом передачі даних, відмінним від заданого;
- ігноруються результати запитів по гіперпосиланнях в тих випадках, коли відповідь сервера в http-заголовки не вказує Content-type text / html або не вказує Content-type зовсім, тобто об'єктами сканування є тільки html-сторінки, гіперпосилання, що вказують на файли з розширеннями .rar, .docx, .js та інші, не розглядаються;
- виконується нормалізація посилань для усунення випадків завантаження одній і тій же сторінки за посиланнями з різним написанням;
- сторінки з однаковим вмістом вважаються однією і тією ж сторінкою.

Для організації алгоритму побудови веб-графа застосовується структура даних стек або черга, до якої будуть потрапляти пов'язані одна з одною сторінки, а потім витягуватися сторінки веб-сайту для здійснення обходу. Відповідна структура даних обирається в залежності від мети побудови веб-графу, якщо використовується черга, то буде здійснюватися обхід веб-ресурсу в глибину, при використанні стеку організується обхід в ширину.

Запропонований підхід до побудови процедури краулінгу було реалізовано наступним алгоритмом.

Вхідні данні: посилання на головну сторінку веб-сайту, з якої починається формування веб-графу.

Вихідні данні: Результуючий файл с записами вершина A_i, вершина A_j, де A_i - адреса сторінки, на якої знайдено посилання на сторінку A_j.

Ініціалізація: організувати структуру даних T для реалізації алгоритму; організувати динамічний масив міток сторінок $w[A_j]$, які вже розглянуто алгоритмом.

Початок.

1. Обрати головну сторінку сайту A_1 для початку обходу, розташувати A_1 у структуру даних T, позначити сторінку, як таку, що пройдено, $w[A_1]=1$.
2. Повторювати до тих пір, поки $T \neq \varnothing$:
   Витягти сторінку A_i зі структури даних T, повернути її, як таку що пройдено.
   Для всіх сторінок A_j, які пов'язані зі сторінкою A_i та відсутні у структурі даних T, виконати наступне:
   Отримати посилання з HTML коду сторінки A_i на сторінку A_j, перевірити коректність адреси та існування сторінки A_j. Якщо умови виконано, то зберегти в результуючий файл запис: вершина A_i, вершина A_j. Додати сторінку A_j у структуру даних T та позначити її, як таку, що пройдено $w[A_j]=1$.
   Кінець.

Під час роботи алгоритму з HTML коду сторінки додатково можна отримувати інформацію про вміст сторінки (заголовок, елементи керування, рисунки), яка може бути збережена в якості властивостей сторінки для подальшого аналізу.

Побудований алгоритм краулінгу веб-ресурсу було реалізовано у вигляді прикладного програмного забезпечення, архітектуру якого наведено на рис 5.1.

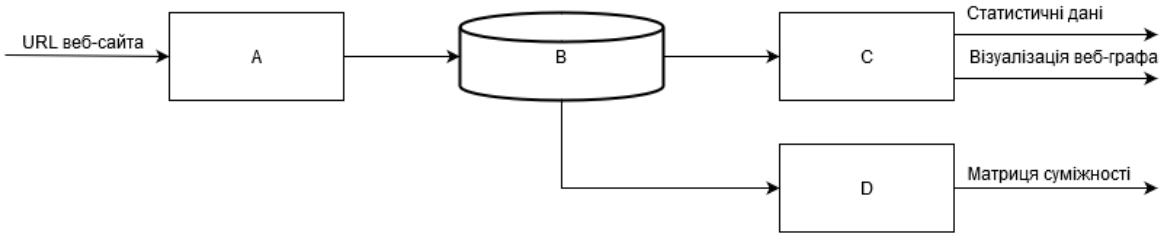


Рис. 5.1 Діаграма компонентів

Програмне забезпечення включає в себе чотири основних компоненти: програмний компонент A відповідає за побудову моделі веб-сайту та обробку даних. До його завдань входить сканування веб-сайту з метою відновлення гіпертекстової структури. Під час роботи відбувається завантаження веб-сторінок, витягування гіперпосилань з множини веб-сторінок, формування черги, нормалізація URL адрес, відсіювання веб-адрес з некоректними адресами, усунення дублювання адрес, формування списку суміжності у вигляді пар елементів <URL-start, URL-end> та для очищення цього списку суміжності від зайвих даних.

Компонент B забезпечує зберігання веб-графу сканованого веб-сайту у вигляді списку суміжності у форматі CSV.

Компонент C слугує для візуалізації графа, його укладання для поліпшення сприйняття та для обчислення деяких статистичних даних.

Компонент D здійснює перетворення списку суміжності в матрицю суміжності та пошук унікальних вершин.

6 Особливості програмної реалізації

Відповідно до розробленої архітектури програмна система побудована у вигляді консольних додатків з використанням мови програмування Python (компоненти A, D), а також із застосуванням існуючого у відкритому доступі програмного забезпечення Gephi для візуалізації побудованого графу (компонента C). Крім стандартних засобів платформи Java SE, використовувалися сторонні бібліотеки і компоненти.

Для здійснення сканування веб-сайту (компонента A) було використано вільно поширюваний Python-фреймворк Scrapy. Цей фреймворк забезпечує роботу по HTTP протоколу з будь-якою сторінкою, розташованої у мережі Інтернет. За його допомогою з HTML-коду сторінки зчитуються всі гіперпосилання на пов'язані сторінки, які потім зберігаються у відповідної

структурі даних. Дана операція повторюється для кожної сторінки, тим самим здійснюється обхід сайту. За отриманою гіпертекстової структурою будується список суміжності.

Завантаження Scrapy може проводитись через PIP – систему управління пакетами, яка використовується для установки і управління програмними пакетами, що побудовано із використанням мови Python.

За замовчуванням Scrapy відвідує сторінки, використовуючи чергу LIFO (Last In – First Out), тобто використовує обхід веб-ресурсу пошуком в глибину.

Для здійснення процедури зберігання даних (Компонента В) було використано бібліотеку CSV, за допомогою якої компоненти матриці зберігаються в зручному для сприйняття користувачем і комп'ютером форматі. Цей формат має статичну структуру, тому для відслідковування помилок можна застосовувати автоматичний режим.

Для візуалізації побудованого графа використано GUI (Graphical user interface) додаток Gephi, який містить в собі набір основних способів укладань графів, а також інші інструменти для аналізу характеристик графів. В роботі застосовується алгоритм укладання Yifan Hu [20], що робить сприйняття отриманих результатів зрозумілим та наочним.

7 Аналіз результатів сканування веб-сайтів

Запропонована модель представлення веб-сайту, метод побудови краулеру та розроблене програмне забезпечення було застосовано для побудови та аналізу веб-графів існуючих у мережі Інтернет сайтів. Було розглянуто веб-сайти: офіційний сайт Дніпровського національного університету імені Олеся Гончара [Дніпровський національний університет імені Олеся Гончара [Електронний ресурс] – Режим доступу до ресурсу: http://dnu.dp.ua/], веб-сайт факультету прикладної математики Дніпровського національного університету імені Олеся Гончара [Сайт факультета прикладной математики ДНУ | [Електронний ресурс] – Режим доступу до ресурсу: http://fpm.dnu.dp.ua/.], веб-сайт інтернет-магазину "Насіння країни" [Насіння країни [Електронний ресурс] – Режим доступу до ресурсу: http://semena-dnepr.org.ua/].

Обчислення виконувались із застосуванням апаратного забезпечення на двоядерному процесорі Intel® Core™ i5-5200U [60] частотою до 2.7 ГГц та 8 Гб оперативної пам'яті DDR3, відеокарті Nvidia GeForce 940M частотою до 1176 МГц та відеопам'яті 2 Гб DDR3.

У табл. 1 для порівняння наведено характеристики веб-сайтів (кількість сторінок та посилань між ними) та час сканування.

Таблиця 1

Параметри сайтів	Веб-сайти		
	dnu.dp.ua	fpm.dnu.dp.ua	semena-dnepr.org.ua
Кількість унікальних сторінок	24460	472	468
Кількість посилань між сторінками	2045729	26504	12096
Час сканування веб-ресурсу для побудови веб-графу, хв.	183	47	39

За результатами роботи розробленого програмного забезпечення були побудовані список суміжності та матриця суміжності веб-графів, на рис 7.1 наведено фрагмент списку суміжності для веб-графа сайту dnu.dp.ua.

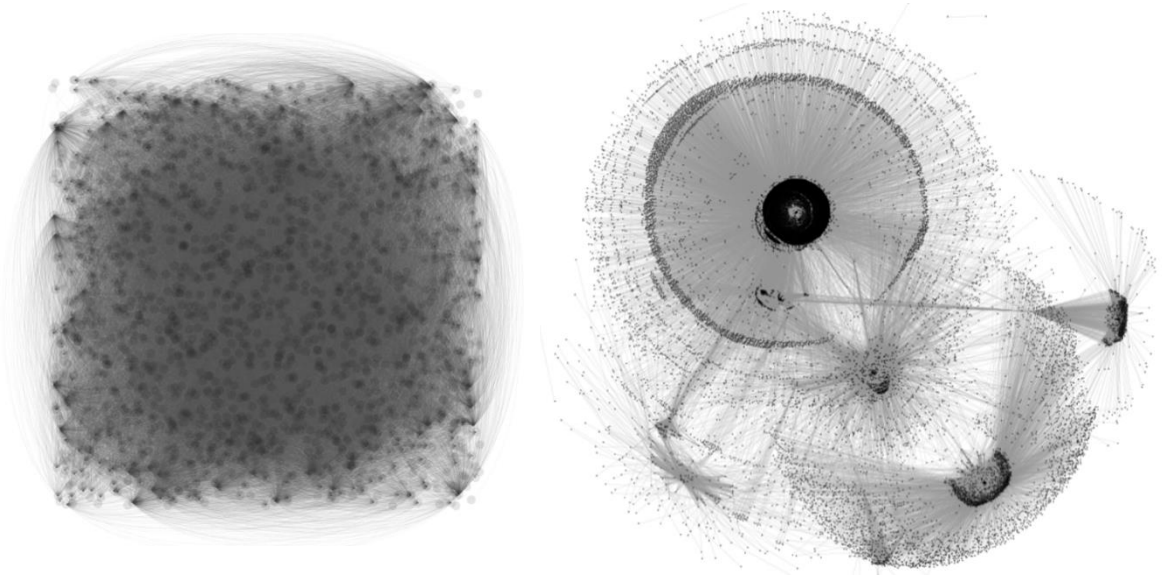
http://www.dnu.dp.ua/view/kolisinini	http://www.dnu.dp.ua/
http://www.dnu.dp.ua/news/3699	http://www.dnu.dp.ua/
http://www.dnu.dp.ua/news/3673	http://www.dnu.dp.ua/newsprint/3673
http://www.dnu.dp.ua/news/3673	http://www.dnu.dp.ua/view/ffilol
http://www.dnu.dp.ua/news/3673	http://www.dnu.dp.ua/view/fsocgum
http://dnu.dp.ua/view/our_partners	http://www.dnu.dp.ua/view/osvitni_programy
http://dnu.dp.ua/view/our_partners	http://www.dnu.dp.ua/vybir_desciplin
http://dnu.dp.ua/view/our_partners	http://www.dnu.dp.ua/view/normativna_baza_oisvitnyogo_processu
http://www.dnu.dp.ua/map	http://www.dnu.dp.ua/
http://www.dnu.dp.ua/view/legendu_dnu	http://www.dnu.dp.ua/

Рис. 7.1 Фрагмент списку суміжності для сайту dnu.dp.ua.

Під час побудови веб-графу сайту інтернет-магазину було виявлено, що для товарів встановлено додаткові фільтри для вибору продукції за деякими критеріями, які знаходяться у тегу <a href=””>. Якщо б краулер переходив по ним і записував усі знайдені ребра до списку суміжності, то зображення веб-графу було б некоректним, для кожної сторінки створювались би зайві посилання, а отриманий граф був би значно більшим. Для усунення зазначеної проблеми у правилах роботи Scrapy було посилено правила відбору URL в HTML-коді, здійснювалось видалення ребер, у яких хоча б одна з URL-адрес (вершин графу) мала змінні.

Побудовані графи було зображено у графічному вигляді. За замовчуванням граф укладається випадковим чином – тобто розташування вузлів графу відносно друг друга відбувається довільно. Для спрощення сприйняття отриманих графів застосовуються різні укладки вершин. В роботі із застосуванням алгоритму Yyifan Hu вузли укладаються по колу у такий спосіб: вузли, які належать до однієї підмножини, притягуються один до одного.

На рис. 7.2 для порівняння впливу укладання на результуюче зображення графу досліджуваний граф сайту dnu.dp.ua зображено довільним чином та із застосуванням укладання. Слід зазначити, що застосування алгоритмів укладання значно спрощує сприйняття структури графу, можна бачити гетерогенність структури сайту.



а) *б)*

Рис. 7.2 а) граф dnu.dp.ua без укладання; б) граф dnu.dp.ua з укладанням Yyifan Hu

Для аналізу отриманих графів було застосовано вбудовані у Gephi алгоритми для розрахунку статистичних показників графа. Серед метричних характеристик використовується діаметр графу, середня степінь вершини, середня довжина шляху, коефіцієнт щільності графа. Діаметр графа визначає максимальну відстань між двома вузлами через інші вузли. Середній ступінь вузла дорівнює середньої кількості вершин, з якими пов’язана вершина. Середня довжина шляху між двома вузлами дорівнює середньої кількості ребер, що зв’язують вершини одну з одною.

Для аналізу зв’язності отриманої структури веб-сайту використовувалось таки показники, як коефіцієнт модулярності та коефіцієнт кластеризації графу. Коефіцієнт модулярності - це скалярна величина, що належить відрізку [-1, 1] та зображує, наскільки при заданому розбитті графа на підмножини щільність зв'язків всередині підмножини більше, ніж щільність зв'язків між підмножинами, тобто визначає якість розбиття графу на підмножини. Коефіцієнт кластеризації зображує, наскільки сильно вершини схильні утворювати кластери (групи пов’язаних між собою вершин), які характеризуються тим, що вершини з однієї групи з'єднані між собою набагато щільніше, ніж з усім іншим графом.

Результати розрахунків зазначених величин наведено у таблиці 2:

Таблиця 2

Метричні характеристики	Веб-сайти		
	dnu.dp.ua	fpm.dnu.dp.ua	semena-dnepr.org.ua
Діаметр графа	207	5	13
Середня степінь вершин	60.561	18.845	18.2
Середня довжина шляху	21.039	3.042	2.306
Щільність графа	0.002	0.012	0.039
Коефіцієнт модулярності	0.206	0.189	0.11
Коефіцієнт кластеризації	0.33	0.24	0.176

Аналіз значень коефіцієнтів модулярності показав, що для графу сайту ДНУ цей показних значно вищий, ніж для інших досліджуваних сайтів. Високі показники коефіцієнту модулярності говорять про високу гетерогенність структури сайту, оскільки зростає кількість зв'язків всередині підграфів відносно до кількості зв'язків з іншими підграфами. Підтвердження цього висновку можна також отримати, якщо подивіться на рис. 7.2 з зображенням веб-графу сайту ДНУ з укладанням. Спостерігається наявність чіткої структури з можливістю бачити окремі зв'язні підграфи.

Наявність чіткої структури та існування зв'язних підграфів для сайту підтверджує й значення коефіцієнту кластеризації. Для сайту ДНУ цей показник дорівнює 0,33.

Сайти fpm.dnu.dp.ua та semena-dnepr.org.ua мають менший коефіцієнт модулярності та візуально не мають чітко виділених зв'язних підграфів, це підтверджує й більш низький коефіцієнт кластеризації, значення якого дорівнює 0,189 та 0,1 відповідно.

За результатами метричного аналізу можна робити висновки щодо зв'язності веб-графів та їх підграфів, відокремлювати тематичні кластери, перевіряти окремі веб-сторінки (наприклад, сторінки товарів інтернет-магазину) на належність до певних категорій, будувати рекомендаційні системи, виконувати процедури реінжинірингу сайтів.

8 Висновки

У роботі зроблено аналіз моделей представлення інтернет простору у вигляді семантичних мереж, фреймових структур, онтологій та обрано модель представлення веб-ресурсу у вигляді веб-графу. Для побудови моделі веб-сайту у вигляді веб-графу розроблено метод, алгоритм сканування сторінок веб-ресурсу та відповідне програмне забезпечення, що реалізує процедуру краулінгу. Програмне забезпечення побудовано із використанням фреймворку Scrapy та мови програмування Python, за допомогою розробленого програмного забезпечення можливе сканування веб-ресурсів з метою побудови їх веб-графів. Для подальшої роботи з отриманими веб-графами передбачено зберігання інформації у зручному вигляді за допомогою списку ребер та матриці суміжності. Для візуалізації отриманих графів та обчислення деяких метричних характеристик застосовано програмне середовище Gephi. Виконано сканування існуючих у мережі Інтернет веб-сайтів, побудовано їх веб-графи, із застосуванням метричних характеристик здійснено аналіз структурної зв'язності досліджуваних графів.

ЛІТЕРАТУРА

1. Kumar R., Raghavan P., Rajagopalan S., Sivakumar D., Tomkins A., Upfal E. The Web as a graph. *In Proceedings of the 19th Symposium on Principles of Database Systems (PODS). ACM Press*. 2000. P. 1-10. https://www.researchgate.net/publication/221559387_The_Web_as_a_Graph
2. Ling Liu and Özsu Tamer M. Ontology to appear in the Encyclopedia of Database Systems [Электронный ресурс]. *Springer-Verlag*. 2008. URL: http://tomgruber.org/writing/ontology-in-encyclopedia-of-dbs.pdf (дата звернення: 06.10.2020)
3. Каунг Мьят Хту Анализ языка Веб онтологии (owl) и семантическая веб-технология. *Auditorium*. 2017. №4 (16). URL: https://cyberleninka.ru/article/n/analiz-yazyka-veb-ontologii-owl-i-semanticheskaya-veb-tehnologiya (дата звернення: 06.10.2020)

4. Ольшевский А.И., Кондратьева А.А. Описание способов представления web-сайтов в виде фреймовой модели для реализации функциональных операций в Интернет-клиентских системах. *Искусственный интеллект*. 2008. №1. С. 110–116. http://dspace.nbuv.gov.ua/handle/123456789/6551
5. Bing L. Web Data Mining. Springer. 2011. 624 p. https://www.springer.com/gp/book/9783642194597
6. Sheu P., Yu H., Ramamoorthy C., Joshi A., Zadeh L. Link Analysis in Web Mining: Techniques and Applications. *Semantic Computing Digital Object Identifier*. 2010. P. 69–86. https://www.researchgate.net/publication/229459726_Link_Analysis_in_Web_Mining_Techniques_and_Applications
7. Lai Wei, Huang Xiaodi From graph data extraction to graph layout: Web information visualization. *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on Digital Object Identifier*. 2010. P. 224–229. https://bit.ly/3omK2Qa
8. Hall W., Tiropanis T. Web evolution and Web Science. *Computer Networks*. 2012. Vol. 56. № 18. P. 3859–3865. https://bit.ly/3qB6NBG
9. Шокин Ю.И., Веснин А.Ю., Добрынин А.А., Клименко О.А., Рычкова Е.В. Анализ веб-пространства академических сообществ методами вебометрики и теории графов. *Информационные технологии*. 2014. № 12. С. 31–40. http://www.ict.nsc.ru/jspui/handle/ICT/252
10. Ермолин Н.А., Мазалов В.В., Печников А.А. Теоретико-игровые методы нахождения сообществ в академическом Вебе. Труды СПИИРАН. 2017. 6(55), 237–254. URL: https://doi.org/10.15622/sp.55.10 (дата звернення: 06.10.2020)
11. Горбунов А.Л. Марковские модели посещаемости веб-сайтов. *Интернет-математика 2007: сб. работ участников конкурса научных проектов по информационному поиску*. 2007. С. 65–73. https://elar.urfu.ru/handle/10995/1334
12. Liu Y., Ma Z. M., Zhou C. Web Markov Skeleton Processes and Their Applications // *Tohoku Mathematical Journal*. 2011. No. 63. P. 665–695. https://bit.ly/3gg5vra
13. Yadav M., Goyal N. Comparison of Open Source Crawlers- A Review. *International Journal of Scientific & Engineering Research*. 2015. V. 6. P. 1544-1551. https://www.ijser.org/researchpaper/Comparison-of-Open-Source-Crawlers--A-Review.pdf
14. Udapure T.V., Kale R.D., Dha R.C. Study of Web Crawler and its Different Types. *Journal of Computer Engineering*. 2014. Vols. 16(1). P. 3–4. https://bit.ly/3qtBXep
15. Ahuja M.S., Bal J.S., Varnica Web Crawler: Extracting the Web Data. *International Journal of Computer Trends and Technology*. 2014. №13(3). С. 132–137. https://bit.ly/2VLlrrV
16. Печников А.А., Сотенко Е.М. Программы-краулеры для сбора данных о представительских сайтах заданной предметной области – аналитический обзор. *Современные наукоемкие технологии*. 2017. № 2. С. 58-62. https://top-technologies.ru/ru/article/view?id=36585
17. Пудикова Е.М. Обзор веб-краулеров для решения задачи сбора данных о представительских сайтах заданной предметной области. *Системный анализ*. 2016. С. 1–16. https://bit.ly/36PsTbL
18. Гудков К.В., Тонкушин М.В. Анализ автоматизированных систем сбора информации в сети интернет. *Современные информационные технологии*.2015. № 28. С. 27–31. http://www.penzgtu.ru/fileadmin/filemounts/confcit/articles/autumn_2018/04.pdf
19. Блеканов И.С., Сергеев С.Л., Мартыненко И.А. Построение тематико-ориентированных веб-краулеров с использованием обобщенного ядра. *Научно-технические ведомости санкт-петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление*. 2012. № 5(157). С. 9–15.
20. Hu Y. Efficient and high quality force-directed graph drawing. *Mathematica Journal*. 2006. №10. P. 37–71. https://bit.ly/37W0bph

REFERENCES

1. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal The Web as a graph. *In Proceedings of the 19th Symposium on Principles of Database Systems (PODS). ACM Press*. 2000, P. 1–10. https://www.researchgate.net/publication/221559387_The_Web_as_a_Graph
2. Ling Liu and Özsu Tamer M. Ontology to appear in the Encyclopedia of Database Systems [Электронный ресурс]. Springer-Verlag, 2008. URL: http://tomgruber.org/writing/ontology-in-encyclopedia-of-dbs.pdf (Last accessed: 06.10.2020)

3. Kaung Myat Htu “Web Ontology Language Analysis (owl) and Semantic Web Technology”, *Auditorium*, №4 (16), 2017. URL: https://cyberleninka.ru/article/n/analiz-yazyka-veb-ontologii-owl-i-semanticheskaya-veb-tehnologiya (date of access: 06.10.2020) [in Russian]
4. A.I. Olshevsky, A.A. Kondratyeva “Description of ways to represent web sites in the form of a frame model for the implementation of functional operations in Internet client systems”, *Artificial Intelligence*, №1, C. 110–116, 2008. http://dspace.nbuv.gov.ua/handle/123456789/6551 [in Russian]
5. L. Bing *Web Data Mining*. Springer. 2011, 624 p. https://www.springer.com/gp/book/9783642194597
6. P. Sheu, H. Yu, C. Ramamoorthy, A. Joshi, L. Zadeh “Link Analysis in Web Mining: Techniques and Applications”, *Semantic Computing Digital Object Identifier*, P. 69–86, 2010. https://www.researchgate.net/publication/229459726_Link_Analysis_in_Web_Mining_Techniques_and_Applications
7. Lai Wei, Huang Xiaodi “From graph data extraction to graph layout: Web information visualization”, *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on Digital Object Identifier*, P. 224–229, 2010. https://bit.ly/3omK2Qa
8. W. Hall, T. Tiropanis “Web evolution and Web Science”, *Computer Networks*, Vol. 56, № 18, P. 3859–3865, 2012. https://bit.ly/3qB6NBG
9. Yu.I. Shokin, A.Yu. Vesnin, A.A. Dobrynin, O.A. Klimenko, E.V. Rychkova “Analysis of the web space of academic communities using webometrics and graph theory”, *Information technologists*, № 12, C. 31–40, 2014. http://www.ict.nsc.ru/jspui/handle/ICT/252 [in Russian]
10. N.A. Ermolin, V.V. Mazalov, A.A. Pechnikov “Game-theoretic methods for finding communities in the academic Web”, *SPIIRAS Proceedings*, 6(55), P. 237–254, 2017. URL: https://doi.org/10.15622/sp.55.10 (date of access: 06.10.2020) [in Russian]
11. A.L. Gorbunov “Markov Models of Website Traffic”, *Internet Mathematics 2007: collection of articles. works of participants in the competition of scientific projects on information retrieval*. C. 65–73. 2007 https://elar.urfu.ru/handle/10995/1334
12. Y. Liu, Z.M. Ma, C. Zhou “Web Markov Skeleton Processes and Their Applications”, *Tohoku Mathematical Journal*. № 63. P. 665–695. 2011 https://bit.ly/3gg5vra
13. M. Yadav, N. Goyal “Comparison of Open Source Crawlers- A Review”, *International Journal of Scientific & Engineering Research*, V. 6, P. 1544-1551, 2015. https://www.ijser.org/researchpaper/Comparison-of-Open-Source-Crawlers--A-Review.pdf
14. T.V. Udapure, R.D. Kale, R.C. Dha “Study of Web Crawler and its Different Types”, *Journal of Computer Engineering*, Vol. 16(1), P. 3–4, 2014. https://bit.ly/3qtBXep
15. M.S. Ahuja, J.S. Bal, Varnica “Web Crawler: Extracting the Web Data”, *International Journal of Computer Trends and Technology*, №13(3), P. 132–137, 2014. https://bit.ly/2VLlrrV
16. A.A. Pechnikov, E.M. Sotenko “Crawler programs for collecting data on representative sites of a given subject area - an analytical review”, *Modern high technology technologists*, № 2, C. 58-62, 2017. https://top-technologies.ru/ru/article/view?id=36585 [in Russian]
17. E.M. Pudikova “Review of web crawlers for solving the problem of collecting data on representative sites of a given subject area”, *System analysis*, P. 1–16, 2016. https://bit.ly/36PsTbL [in Russian]
18. K.V. Gudkov, M.V. Tonkushin “Analysis of automated systems for collecting information on the Internet”, *Modern information technologies*, № 28, C. 27–31, 2015. http://www.penzgtu.ru/fileadmin/filemounts/confcit/articles/autumn_2018/04.pdf [in Russian]
19. I.S. Blekanov, S.L. Sergeev, I.A. Martynenko “Build subject-oriented web crawlers using a generic kernel”, *Scientific and technical statements of the St. Petersburg State Polytechnic University. Informatics. Telecommunications. Control*. № 5(157), C. 9–15, 2012. [in Russian]
20. Y. Hu “Efficient and high quality force-directed graph drawing”, *Mathematica Journal*, №10, P. 37–71, 2006. https://bit.ly/37W0bph

Надійшла у першій редакції 01.09.2020, в останній -11.09.2020.