МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження та проектування інформаційно-статистичної системи на прикладі автомобільного ринку України

Освітній ступінь "Магістр"
Спеціальність 123 "Комп'ютерна інженерія"

Науковий керівник роботи:	_____ (підпис)	С.О.Сафонова (ініціали, прізвище)
Консультант з охорони праці:	_____ (підпис)	Я.О.Критська (ініціали, прізвище)
Студент:	_____ (підпис)	Соловйов В.А. (ініціали, прізвище)
Група:		КІ-17зм

Сєвєродонецьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній ступінь магістр
Напрям підготовки _____
(шифр і назва)
Спеціальність 123 "Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
_____ І.С. Скарга-Бандурова
«_____» _____20____р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Соловйову Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та проектування інформаційно-статистичної системи на прикладі автомобільного ринку України

керівник проекту (роботи) Сафонова Світлана Олександрівна, к.т.н., доц.
(прізвище, м.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 221/48
2. Строк подання студентом роботи 19.06.2019
3. Вихідні дані до роботи Матеріали науково-дослідної практики, методи збору та та аналізу контенту інтернет-ресурсівс

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд технологій web content mining, дослідження методів аналізу веб контенту, розробка та реалізація системи пошуку оголошень автомобільного ринку, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2018

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	10.09.2018-15.09.2018	
2	Аналіз методів вилучення контентус	16.09.2018-22.09.2018	
3	Розробка модулю вилучення контент	26.09.2018-06.10.2018	
4	Розробка інтерфейсу користувача	07.10.2018-25.11.2018	
5	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2018-1.12.2018	
6	Оформлення пояснювальної записки, автореферату та презентації	2.01.2019-19.06.2019	
7			

Студент _____ Соловйов В.А.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Сафонова С.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Соловйов В.А. Розробка інформаційно-статистичної системи на прикладі автомобільного ринку України.

Проведено загальний аналіз технології Web Content Minig. Виявлено основні проблеми отримання даних з веб-сторінок. Виконано порівняльний аналіз існуючих технологій. Проведено дослідження основних методів вилучення інформації з веб документів, і проведено порівняльний аналіз за певними критеріями. Розроблено програму для вилучення інформації про нерухомість на мові php. Розроблено веб інтерфейс для пошуку і відображення витягнутої інформації про пропозиції автомобільного ринку.

Ключові слова: Web Content Minig, PHP, парсер, MySql. витяг контенту.

ABSTRACT

Solovyov V.A. Development of informational-statistical system on the example of the automobile market of Ukraine.

We carry out a general analysis of the Web Content Minig technology. The basic problem of extracting data from web pages. A comparative analysis of the existing technologies. A study of the main methods of extracting information from web documents, and a comparative analysis on specific criteria. A program extracts information about the realty in terms of using php had been developed. A web interface for searching and displaying the extracted information about the cars.

Key words: information model, database, information search, information analysis.

ЗМІСТ

ВСТУП

Актуальність теми. Необхідність інтенсифікації робочого часу та глобалізація світових процесів вимагає від фахівців на ринку нерухомості використання автоматизованого збору інформації.

Основною проблемою пошуку подібної інформації є її розосередження. Якщо всеукраїнські дошки оголошень добре структуровані, то значна частина місцевих оголошень зазвичай розміщується на неструктурованих під автомобільний ринок сайтах, тому підбір та аналіз таких оголошень значно ускладнюється.

Рішенням даної проблеми в значній мірі є створення ресурсу який би агрегованих інформацію про подібні проекти. Надавав би зручний інтерфейс для пошуку відповідних проектів. Процес додавання та оновлення інформації про проекти можна автоматизувати за допомогою одного з методу інтелектуального аналізу веб сторінок - Web Content Mining. Саме розробці подібного ресурсу, і дослідженню методу Web Content Mining буде присвячена дана дипломна робота.

У роботі представлені вирішення цієї проблеми у вигляді дослідження методів інтелектуального аналізу веб-контенту і розробки системи пошуку автомобілів.

При виконанні роботи автором були обрані і проаналізовані методи інтелектуального аналізу вмісту веб-сторінок, вивчені принципи роботи кожного з методів, розробленийа система пошуку автомобілів.

Мета і завдання дослідження. Метою атестаційної роботи є дослідження методів інтелектуального аналізу веб-контенту. Для досягнення цієї мети в роботі сформульовані і вирішені **наступні завдання**:

– проведено аналіз існуючих технологій пошуку, вилучення та об'єднання вмісту веб-сторінок;

– проведені дослідження методів інтелектуального аналізу веб-контенту;

– розроблено програмне забезпечення для вилучення вмісту з веб сторінок;

– розроблена інформаційна система яка дозволяє зберігати і надавати кінцевому користувачеві витягнуті дані.

Об'єктом дослідження є процеси пошуку та інтелектуального аналізу вмісту веб-сторінок.

Предметом дослідження - методи та інформаційні технології створення і функціонування автоматизованих систем вилучення інформації з веб-сторінок.

Методи вирішення поставлених завдань базуються на використанні технології Web Content Mining для вилучення інформації про автомобілі.

Наукова новизна отриманих результатів.

Отримали подальший розвиток методи Web Content Mining, які використані для вирішення завдання пошуку оголошень з нерухомості, що дозволило використовувати існуючі аналітичні методи для пошуку і виділення потрібної інформації зі сторінок з різними структурами і макетами.

Практичне значення отриманих результатів.

Дана розробка впроваджена і використовується на ТОВ «Твоє Авто» (м. Попасне).

Структура і обсяг роботи. Робота складається з вступу, чотирьох розділів, висновків, списку використаних джерел та додатки. Робота викладена на 157 сторінках машинописного тексту, містить 116 сторінок основного тексту, 19 рисунків, 8 таблиць, 5 додатків на 30 сторінках. Перелік джерел включає 98 посилання.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Перелік скорочень	Детальне розшифрування	Переклад
CT	Common format	Загальний формат
EC	Embedded catalog	Вбудований каталог
IE	Information Extraction	Видобуток інформації
SP	Sequential patterns	Послідовні шаблони
WI	Wrapper induction	Індукція обгорток
MUC	Message Understanding Conference	Конференція по Розумінню Повідомлень

1 ОГЛЯД ТЕХНОЛОГІЙ WEB CONTENT MINING

1.1 Аналіз актуальності досліджуваної проблеми

Об’єм накопичуваної людством інформації подвоюється кожні 2-3 роки. Цей безмежний потік приходить до нас з науки та бізнесу, глобальної мережі Інтернет та багатьох інших джерел. Якщо в 1989 р. один мегабайт вважався розміром великої бази даних, то вже в наші часи в наукових дослідженнях назріла необхідність мати справи з петабайтами.

Такий великий обсяг інформації призводить до того, що лише досить незначну її частку може побачити людське око. Ледь не єдиний спосіб зрозуміти та знайти щось корисне в цьому безмежному просторі інформації – це використання методів інтелектуального аналізу даних по виявленню шаблонів в Інтернеті (web data mining). Вони роблять можливим використання автоматизованих систем для виявлення і виводу даних з сторонніх серверів, і це дозволяє організаціям отримати як організовану так і неструктуровану інформацію від діяльності браузера, журналів сервера, веб-сайта, структури посилань, змісту сторінки і різних джерел.

Web (Data) Mining - це використання методів інтелектуального аналізу даних для автоматичного виявлення веб-документів і послуг, отримання інформації з веб-ресурсів і виявлення загальних закономірностей в Інтернеті [1].

Web Data Mining також включає в себе основні теми Data Mining і вилучення інформації (Text Mining), оскільки Web Mining використовує свої алгоритми і методи досить широко.

Термін Data Mining часто перекладається як видобуток даних, вилучення інформації, розкопка даних, інтелектуальний аналіз даних, засоби пошуку закономірностей, витяг знань, аналіз шаблонів, "витяг зерен знань з гір даних", розкопка знань в базах даних, інформаційна проходка даних, "промивання" даних. Поняття "виявлення знань в базах даних" (Knowledge Discovery in Databases, KDD) можна вважати синонімом Data Mining [2].

Поняття Data Mining, що з'явилося в 1978 році, набуло високу популярність в сучасному трактуванні приблизно з першої половини 1990-х років. До цього часу обробка і аналіз даних здійснювався в рамках прикладної статистики, при цьому в основному вирішувалися завдання обробки невеликих баз даних.

Область Data Mining почала активно розвиватися з семінару (англ. Workshop), проведеного один із засновників цього напрямку Григорієм Пятецким-Шапіро в 1989 році [3]. Він досить точно визначив технологію Data Mining як процес виявлення в сирих даних раніше невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності.

Раніше, працюючи в компанії GTE Labs, Григорій П’ятецький-Шапіро зацікавився питанням: чи можна автоматично знаходити певні правила, щоб прискорити деякі запити до великих баз даних. Тоді ж було запропоновано два терміни - Data Mining («видобуток даних» [4]) і Knowledge Discovery In Data (який слід перекладати як «відкриття знань в базах даних»).

Data Mining - це процес підтримки прийняття рішень, заснований на пошуку в даних схованих закономірностей (шаблонів інформації) [5].

Суть і мета технології Data Mining можна охарактеризувати так: це технологія, яка призначена для пошуку у великих обсягах даних неочевидних, об'єктивних і корисних на практиці закономірностей.

Про популярність Data Mining говорить і той факт, що результат пошуку терміна "Data Mining" в пошуковій системі Google (на квітень 2017 року) - понад 56 мільйонів сторінок.

Data Mining - мультидисциплінарна область, що виникла і розвивається на базі таких наук як прикладна статистика, розпізнавання образів, штучний інтелект, теорія баз даних та ін., див. рис. 1.1.

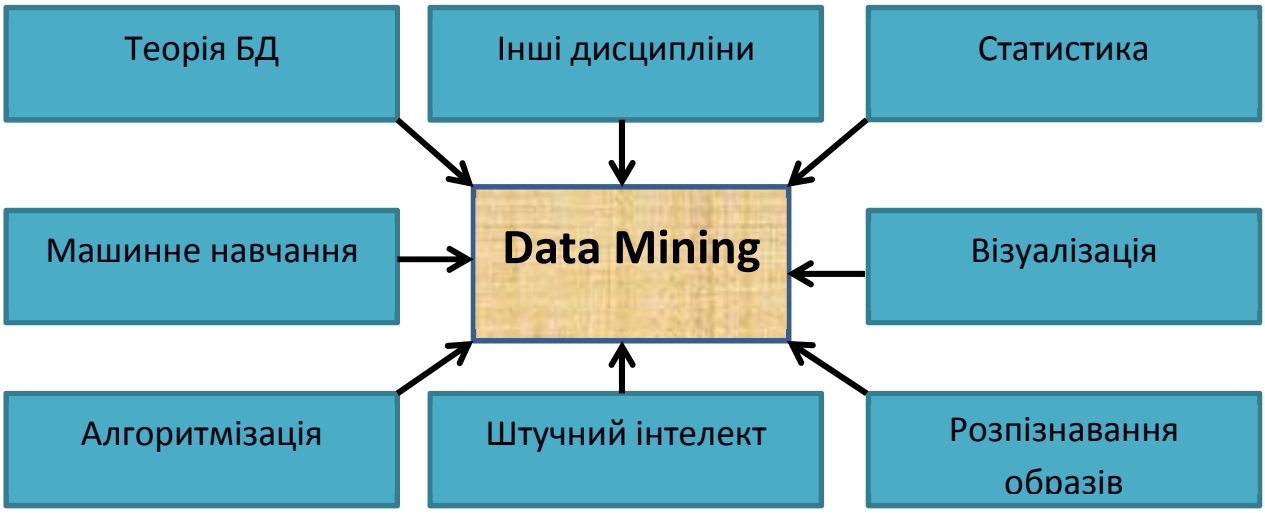


Рисунок 1.1 – Data Mining як мультидисциплинарна область

Метою інформаційного вилучення є перетворення колекції документів, зазвичай за допомогою інформаційно-пошукових систем, в легко засвоювану і проаналізовану інформацію. Процес вилучення інформації спрямований на виймання релевантних фактів з документів, в той час як процес інформаційного пошуку спрямований на селекцію релевантних документів. Перший зацікавлений в структурі або поданні документа, тобто працює на рівні тонкої деталізації, а другий розглядає текст документа як колекцію невпорядкованих слів. Проте, відмінності між двома процесами стають несуттєвими, якщо мета інформаційного пошуку - це вилучення інформації [6].

Завдяки динаміці і різноманітності веб-вмісту, створення ручного режиму систем інформаційного вилучення не представляється можливим. У зв'язку з цим, більшість систем по вилученню даних зосереджують увагу на конкретні веб-сайти. Інші використовують навчальні пристрої або методи інтелектуального аналізу даних і здатні витягувати веб-документи в автоматичному чи напівавтоматичному режимі. З цієї точки зору, Web Mining є частиною процесу вилучення інформації з Інтернету .

У Web Mining можна виділити наступні етапи:

– вхідний етап (англ. input stage) - отримання «сирих» даних з джерел (логи серверів, тексти електронних документів);

– етап попередньої обробки (англ. preprocessing stage) - дані подаються у формі, необхідної для успішної побудови тієї чи іншої моделі;

– етап моделювання (англ. pattern discovery stage);

– етап аналізу моделі (англ. pattern analysis stage) - інтерпретація отриманих результатів.

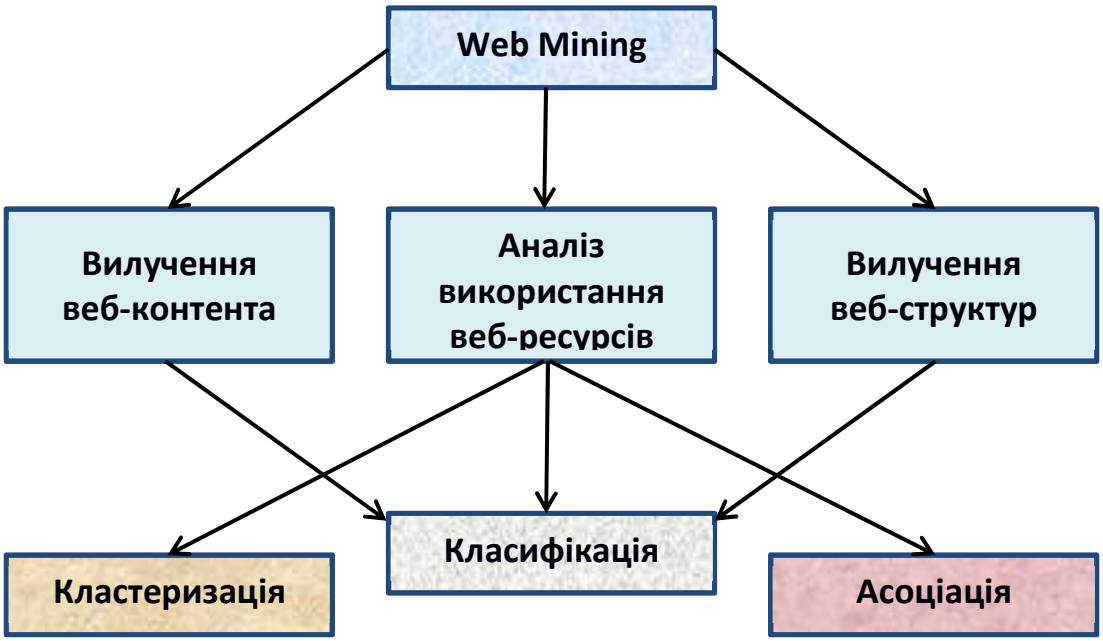


Рис. 1.2. Взаємозв'язок між категоріями Web Mining і завданнями інтелектуального аналізу даних

Це загальні кроки, які необхідно пройти для аналізу даних мережі Інтернет. Конкретні процедури кожного етапу залежать від поставленого завдання. У зв'язку з цим виділяють різні категорії Web Mining:

– Web Content Mining;

– Web Structure Mining;

– Web Usage Mining.

Web Content Mining (витяг веб-контенту) - процес вилучення знань з контенту документів або їх опису, доступних в Інтернеті [7]. Пошук знань в мережі Інтернет є непростим і трудомістким завданням. Саме цей напрям Web Mining вирішує її. Воно засноване на поєднанні можливостей інформаційного пошуку, машинного навчання та інтелектуального аналізу даних.

Web Structure Mining (витяг веб-структур) - процес виявлення структурної інформації в Інтернету [8]. Даний напрямок розглядає взаємозв'язок між веб-сторінками, ґрунтуючись на зв'язках між ними. Побудовані моделі можуть бути використані для категоризації і пошуку схожих веб-ресурсів, а також для розпізнавання авторських сайтів.

Web Usage Mining (аналіз використання веб-ресурсів) - це автоматичне виявлення шаблонів в маршруті пересування користувача і пов'язаних з ним даними, зібраними або набутими в результаті взаємодії з одним або декількома веб-сайтами [8]. Цей напрямок заснован на отриманні даних з логів веб-серверів. Метою аналізу є виявлення переваг відвідувачів при використанні тих чи інших ресурсів мережі Інтернет. Деякі користувачі можуть бути зацікавлені тільки в текстових даних, в той час як інші можуть більше приділяти уваги мультимедійним даним.

Таблиця 1.1 Класифікація пошукових та видобуткових технік та додатків

		Джерело інформації про дані		
		Будь-які дані	Текстові дані	Веб-пов'язані дані
Мета	Отримання відомих даних або документів ефективно і результативно	відновлення даних	інформаційний пошук	веб-індексування
	Пошук нових моделей або знань, раніше невідомих	Data Mining	Text Mining	Web Mining

1.2 Огляд сучасних патентів, проектів з тематики дослідження

Web Content Mining є досить поширеним напрямком наукових досліджень в різних галузях науки, бізнесу, освіти.

Метою даного огляду є порівняння існуючих реалізацій технології Web Content Mining сторонніх веб-ресурсів, здійснених дослідниками з України, країн - членів СНД, США в для пошуку, аналізу тематичних оголошень, текстів у різних сферах діяльності.

1.2.1 Огляд патентів на винаходи з тематики дослідження

В рамках роботи було проведено огляд патентів та заявок на винаходи та корисні моделі в сфері видобутку інформації (див. додаток А.) в Україні, СНД, США, а також міжнародні патенти та заявки на винаходи (WO, PCT, EA).

Внаслідок аналізу існуючих патенів що діють у сфері видобутку інформації встановлено, що в сфері видобутку інформації в тому числі з веб-сторінок відбуваються активні розробки та патентування. Найбільш поширені галузі, де ці технології практично використовується, це фінанси, наука, торгівля, навчання, соціальні дослідження, медицина, державне управління тощо.

1.2.2 Огляд сучасних бізнес-проектів

Під час дослідження перспектив практичного впровадження результатів роботи було зібрано та проаналізовано:

– інформацію про наявні релевантні переважно українські бізнес проекти як щодо збору / агрегації інформації в різних сферах так і безпосередньо стосовно автомобільного ринку України(див. Додаток Б.);

– інформацію про наявні Проекти, що надають послуги видобутку даних з веб-сторінок (див. Додаток В.);

– інформацію щодо програм та сервісів з веб-парсингу текстових даних даних з веб-сторінок (див. Додаток Г.).

Видобування тематичної інформації з веб-сторінок, як видно з наведених оглядів, досить динамічний сегмент IT-технологій, на якому зберігається великий попит в розробці систем агрегації даних.

1.3 Короткий критичний аналіз існуючих підходів до організації інтелектуального пошуку інформації

В главі розлянуті веб-технології, які можливо застосувати при професійному створенні синтаксичних аналізаторів.

1.3.1 Основні технології

Для складання первинного алгоритму роботи майбутнього парсеру необхідно проаналізувати вихідний код сторінок сайту-донора для чого необхідні знання HTML, CSS і Java Script.

Для більш глибокого занурення в тему бажано застосовувати технологію DOM, що дозволяє з максимальним ефектом працювати з ієрархічним деревом веб-документа.

На етапі написання аналізатора (найважливіший і складний) необхідно застосовувати інструменти текстової обробки. Для пошуку потрібних шматків тексту доцільно використовувати регулярні вирази, які є могутнім засобом для вирішення складних завдань.

Однак цей спосіб не є найкращим, а часто і небажаним. По-перше, все-таки, регулярні вирази не під силу навіть досить досвідченим фахівцям. По-друге, html-код на більшості сайтів невалідeн, а часто і некоректний. Навіть найзапекліші фахівці можуть заплутатися в метасимволах і квантіфікаторах, намагаючись передбачити всі випадки життя.

Тому, оптимальним виходом може бути використання готових бібліотек для парсинга, трохи нижче вказані найпопулярніші рішення для найпопулярніших мов веб-програмування - PHP, Ruby і Python.

Втім, це зовсім не означає що регулярні вирази можна і зовсім не застосовувати. Часто саме за допомогою них зручно вирішувати багато проблем, перед якими пасують всебічно розроблені бібліотеки для парсинга:

– Для ефективної роботи з ієрархічними структурами даних - необхідно володіти парадигмою **об'єктно-орієнтованого програмування**. Семантичне

дерево можна будувати і за допомогою багатовимірних асоціативних масивів, але цей спосіб занадто важкий. ООП відмінно підтримується всіма найпопулярнішими мовами веб-програмування.

– Фінальна обробка результатів передбачає збереження даних в структурованому вигляді. Зазвичай на виході потрібна база даних. Тому знадобиться використання SQL / MySql / PostgreSQL.

– Можуть знадобиться **функції для роботи з файлами.** Отримані дані можуть бути збережені в CSV-файли або сконвертовані в електронні таблиці.

– Відомості про **XML** і **XPath** можуть знадобитися як на етапі синтаксичного аналізу, так і на стадії кінцевого збереження результатів.

– Іноді отримані дані заливаються в нову базу даних за допомогою **JSON** .

1.3.2 Інструментарій для вирішення завдання на PHP

Для отримання вихідного коду html-сторінок знадобиться бібліотека cURL. Це компактний набір потужних функцій, призначених для роботи з серверами по різних протоколах. Найбільш корисною є стандартна php-функція file_get_contents, що дозволяє легко отримувати текстовий вміст віддалених файлів.

З спеціалізованих бібліотек на даний момент найбільшого поширення набули PHP Simple HTML DOM Parser, PHPQuery, Zend_DOM_Query, Nokogiri.

1.3.3 Інструментарій для вирішення завдання на Ruby

З ефективних ruby-бібліотек на окрему увагу заслуговує **Nokogiri**, з великим набором функцій, які можуть знадобитися при написання найскладнішого парсеру. Для вирішення специфічних завдань підійде Watir, що дозволяє отримувати дані, оновлювані через Ajax-запити.

1.3.4 Інструментарій для вирішення завдання на Python

Парсинг на Python переважно асоціюється з бібліотекою lxm. Втім вона не позбавлена недоліків. Поперед усім - проблеми з кодуванням кирилиці. Альтернативним рішенням є Grab, що вдає із себе навіть не бібліотеку, а зручний спеціалізований фреймворк. Також досить поширений фреймворк Scrapy – одна з найбільш популярних і продуктивних бібліотек Python для отримання даних з веб-сторінок, яка включає в себе більшість загальних функціональних можливостей.

1.3.5 Інструментарій для вирішення завдання на .NET

На даний момент найбільш популярні бібліотеки для роботи з HTML: AngleSharp, CsQuery, HtmlAgilityPack, Fizzler (надбудова до HtmlAgilityPack, що дозволяє використовувати селектори CSS.) і, звичайно ж, Regex (регулярні вирази).

Найактуальнішим парсером на даний момент є AngleSharp - зручний, швидкий парсер зі зручним API. API побудований на базі офіційної специфікації по JavaScript HTML DOM. Бібліотека розвивається дуже швидко. Кількість різних плагинів, що полегшують роботу, вражає, наприклад IHtmlTableElement , IHtmlProgressElement і тд. Для складних випадків є десятки спеціалізованих інтерфейсів, які допоможуть вирішити поставлену задачу.

CsQuery для вибору елементів використовує мову селекторів CSS. Назви методів скопійовані практично один-в-один, тобто для програмістів, знайомих з jQuery, вивчення буде простим.

Розробники .NET не рекомендують застосовувати Regex, але іноді виникає необхідність, так як парсери, що будують DOM, помітно вимогливіші до потужності комп’ютеру, ніж Regex: вони споживають більше і процесорного часу, і пам'яті.

Якщо дійшло до регулярних виразів, то потрібно розуміти, що не можливо побудувати на них універсальне і абсолютно надійне рішення. Однак якщо мета парсити конкретний сайт, то ця проблема не так критична.

1.3.6 Джерела інформації для вилучення даних

Серед джерел для отримання та аналізу даних можна виділити:

– структуровані дошки оголошень (https://www.olx.ua, https://auto.ria.com/ тощо);

– неструктурованої дошки оголошень (http://www.board24.lg.ua);

– соціальні мережи;

– пошук в Інтернет.

Цей підхід є відносно новим в напрямку видобутку інформації. Основна ідея підходу полягає у використанні Всесвітньої павутини як величезного сховища даних. Для доступу до інформації зазвичай застосовуються пошукові системи. У програмних комплексах PANKOW та C-PANKOW [10] Інтернет використовується в якості джерела даних для побудови семантичної анотації документів. В системі OntoSyphon [11] в Інтернет виконується пошук різних відомостей про поняття онтології. Необхідно відзначити, що багато систем використовують комбінації перерахованих методів, наприклад, алгоритми класифікації та пошуку в Інтернет.

Втім, отримання інформації про об’єкти автомобільного ринку України зазначним способом зазвичай малоефективна внаслідок малого ступеню релевантності інформації, значно ускладненого механізму видобутку та лагу в часі індексації веб-сторінок пошуковими сервісами.

1.4 Розгляд питань архітектури сервісу

Робота сервісу полягає в отриманні інформації про об'єкти автомобільного ринку України як з структурованих так і з неструктурованих

оголошень, одержуваних з різних джерел. Методи аналізу в неструктурованих текстах лежать на стику декількох областей: Data Mining, обробка природних мов, пошук інформації, вилучення інформації та управління знаннями.

Рішення базується на онтологічному підході. Побудова і заповнення онтологій тісно пов'язане з виділенням інформації з використанням онтологій (ontology-based information extraction, OBIE [12]). Загальна архітектура типового сервісу з використанням онтологій представлена на рис. 1.3.

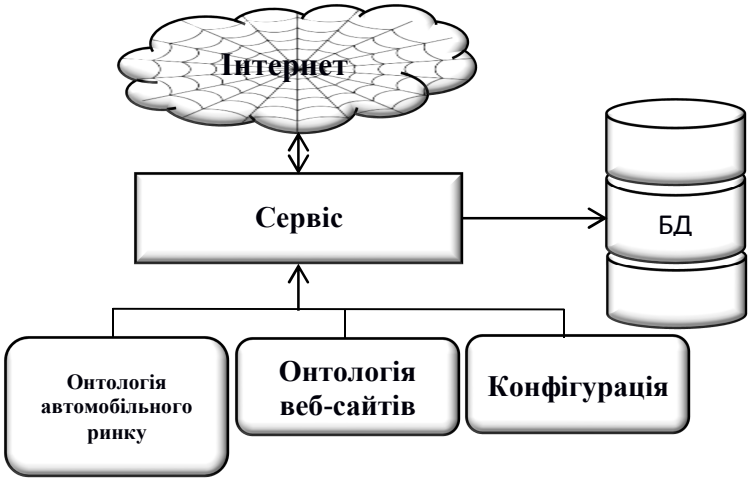


Рисунок 1.3 – Архітектура сервісу

Загальна схема роботи сервісу представлена на рис. 1.4.

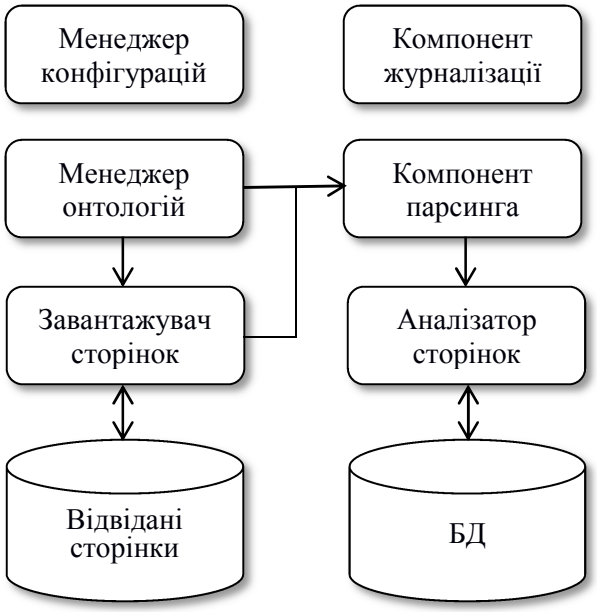


Рисунок 1.4 – Схема роботи сервісу

Компонент журналізації виконує запис інформації про роботу сервісу. Отримана за допомогою даного компонента інформація використовується для моніторингу роботи сервісу та його налагодження.

Менеджер конфігурації надає доступ до переданим сервісу налаштувань і виробляє динамічну настройку роботи сервісу при необхідності.

Менеджер онтологій реалізує операції з управління онтологічними ресурсами.

Онтологія — представлення деякою мовою знань про певну предметну область (середовище, світ). Онтологію неодмінно супроводжує деяка концепція цієї області інтересів. Найчастіше ця концепція виражається за допомогою визначення базових об’єктів (індивідуумів, атрибутів, процесів) і відношень між ними. Визначення цих об'єктів і відношень між ними зазвичай називають концептуалізацією.

Компонент завантаження сторінок створює локальну копію вихідної сторінки, а також виробляє її попередню обробку. Інформація про пройдені сторінках заноситься в спеціальну базу даних, що дозволяє оптимізувати роботу сервісу за рахунок виключення повторного проходу за однаковими посиланнями в поточну сесію пошуку. На базі онтології сайтів про автомобілі даними компонентом здійснюється витяг інформативною частині сторінки. Таким чином, на вхід компонента парсинга сторінок фактично передається передоброблений текст оголошення про продаж автомобілів, з якого за рахунок використання онтології об'єктів автомобільного ринку України витягуються знання, які приводяться до деякого стандартного вигляду (наприклад, відбувається приведення до однакових одиниць виміру пробігу автомобіля та ін.).

Компонент аналізу сторінок здійснює логічний висновок по онтології об'єктів автомобільного ринку ґрунтуючись на отриманих знаннях, а також перевіряє деякі додаткові евристики, після чого формує відомості про об'єкт автомобільного ринку, які заносяться до відповідної бази даних.

Слабка структурованість інформації та гетерогенний характер її джерел диктують застосування засобів і методів штучного інтелекту для вирішення даного завдання (наприклад, text mining, технологій Semantic Web та мультиагентні технології). Тому потребує вирішення завдання усунення дублювання інформації і пошуку протиріч.

Завдяки динаміці і різноманітності веб-вмісту, створення ручного режиму систем інформаційного вилучення не представляється можливим. У зв'язку з цим, більшість систем по вилученню даних зосереджують увагу на конкретні веб-сайти. Інші системи використовують навчальні пристрої або методи інтелектуального аналізу даних і здатні витягувати веб-документи в автоматичному чи напівавтоматичному режимі.

1.4.1 Питання онтології в сфері ринку автомобілів

Онтологія сайтів про автомобільний ринок зберігає специфічні для конкретних сайтів настройки.

Серед параметрів що нас цікавлять виділяють:

– опис позиції на сторінці, де найімовірніше знаходиться цікава інформація, а також опис, що дозволяє отримати заголовок цієї інформації.

– опис позиції на сторінці, де можуть перебувати корисні посилання.

– опис фільтрів, що дозволяють визначити «сміттєві» для сервісу посилання.

– налаштування механізму «перегортання» сторінок.

Онтологія об'єктів автомобільного ринку України містить деякі загальні поняття предметної області і зв'язки між ними.

В процесі парсинга сторінок виконується спроба «прив'язати» конкретні поняття, ґрунтуючись на знаннях, наявних в онтології. До кожного конкретного поняття в онтології приписані певні регулярні вирази. Виділяються регулярні вирази двох типів: загальні і налаштовані під конкретний сайт. Регулярні вирази другого типу можуть використовуватися для прив'язки тільки на

конкретних сайтах і в загальному випадку невірні (такі регулярні вирази дозволяють добре розбирати використовувані на сайті специфічні формулювання). Регулярні вирази загального типу побудовані таким чином, щоб спрацьовувати в загальних випадках. При прив'язці конкретних понять спочатку виконується спроба прив'язки по другому типу і, у разі невдачі, - на першу.

У структурі регулярних виразів можна виділити два типи елементів: ті, що говорять про знаходження збігів, і ті, що свідчать про помилкову прив'язку поняття. Наприклад, прив'язується поняття «зимова резина», до факту подачі об’яви у зимовій період. Елементи другого типу якраз і виявляють ознаки, що дозволяють визначити, що мова йде зовсім не про цікавить понятті.

Крім того, в процесі здобування знань про об'єкти встановлюються певні їх показники («Тип кузову», «Тип коробки перемикання швидкостей» і т.п.). Загальна структура регулярних виразів в цілому аналогічна описаній вище, проте додатково виділяються логічні частини, що дозволяють, наприклад, виконати переклад показників в деяку єдину систему (якщо була вказана ціна в тисячах гривень за автомобіль, то сервіс призведе цю характеристику до уніфікованої одиниці – усоінф одиниці і т.п.).

Під час ручної або автоматизованої побудови онтологій використовується граматика - це людиночитаємий текстовий документ, який легко розуміти, розширювати і виправляти при наявності помилок вилучення, на відміну від статистичних методів навчання, в яких єдине джерело зміни їхньої поведінки - зміна навчальної вибірки.

Найчастіше граматики складаються вручну, що ускладнює їх використання при необхідності витягувати безліч різнотипних об'єктів. Логічно, що в зв'язку з цим з'являється безліч методів генерації граматик. Зокрема, для регулярних виразів використовуються генетичні алгоритми [13, 14], методи, засновані на словниках шаблонів [15]. Однак аналіз даних робіт показує, що дані методи поки підходять лише для вузького кола завдань. Так, наприклад, в роботі [94] алгоритм використовується для збільшення точності вже існуючого шаблону, а в роботах [92, 93] використовується вузьке коло

прикладів (телефони і URL) з навчальними вибірками, зміщеними в бік простих однотипних об'єктів.

1.4.2 Налаштування сервісу

Налаштування сервісу зазвичай містить параметри, що відповідають за роботу сервісу. Серед параметрів можна виділити:

– шлях до списку завантажень, де вказані адреси, які буде сканувати сервіс.

– довжина сесії із сайтом - донором, затримки у видобутку інформації.

– шлях в файлової системі, куди будуть збережені вивантажені сторінки.

– період, через який сервіс відновить свою роботу (зупинка сервісу може бути пов'язана з тим, що він пройде за всіма зазначеними в списку закачування адресами).

– «глибина сканування сайтів» - довжина шляху, на яку переходить сервіс по посиланнях.

– за необхідності в базу даних заносяться лише ті об’єкти, що відповідають фільтру (адреса, орієнтир, фізичний стан тощо).

– За необхідності отримання копій зображень оголошень на веб-сайті по черзі. По завершенні черги зображення передаються до БД.

– використання у разі необхідності проксі-серверів, розпаралелювання операцій, розбиття сесії з базою даних на дрібні.

1.4.3 Проблеми та загальні рекомендації при парсингу HTML даних

Використання JavaScript / AJAX / асинхронних завантажень дуже ускладнюють написання парсеров; різні движки для рендеринга HTML можуть видавати різні DOM дерева (крім того, двигуни можуть мати помилки, які потім впливають на результати роботи парсерів); великі обсяги даних вимагають

писати розподілені парсери, що тягне за собою додаткові витрати на синхронізацію.

Не можна однозначно виділити підхід, який буде 100% може застосовуватись у всіх випадках, тому сучасні бібліотеки для парсинга HTML даних, як правило, комбінують, різні підходи. Наприклад, HtmlAgilityPack дозволяє аналізувати DOM дерево (використовувати XPath), а також з недавніх пір підтримується технологія Linq to XML. Data Extracting SDK використовує аналіз DOM дерева, містить набір додаткових методів для парсинга рядків, а аткже дозволяє використовувати технологію Linq для запитів в DOM моделі сторінки.

На сьогодні одним з лідерів для парсинга HTML даних для .Net є саме бібліотека HtmlAgilityPack. Ця бібліотека буде особливо зручною, якщо доводиться стикатися з JavaScript. Також в ній доступні такі можливості як: Linq to Objects (via LINQ to Xml), XPATH, XSLT.

1.5 Аналіз предметної області

Число комп'ютерно доступних документів зростає так стрімко швидко, що користувачі мають масу проблем в процесі навігації в глобальному (World Wide Web) і локальному (корпоративні сховища даних) інформаційних просторах при пошуку і обробці тільки документів, релевантних їх поточним потребам.

Розробка сервісів з агрегації тематичних текстів не втрачає актуальності. В досліджуваній сфері сервіси здебільшого зорієнтовані на продаж подержаних автомобілів, здебільшого в якості джерел інформації використовуються сайти зі структурованими даними. Як правило, неструктуровані дані розташовані на неспеціалізованих регіональних сайтах та електронних версіях друкованих регіональних періодичних видань. Екстракція контенту таких даних неможлива без застосування технологій інтелектуального пошуку.

До того ж, на даний час в Україні відсутні сервіси, які б могли розрахувати ринкову вартість автомобіля на підставі автоматично зібраних ринкових даних про автомобіль.

В даний час у використанні знаходяться кілька відомих методик для допомоги користувачеві в орієнтації на багато документів, зокрема пошукові машини, карти сайтів і сторінки посилань, каталоги і індекси і т.п. На глибинному рівні всі вони базуються на «гіперпосиланнях», вбудованих в документи, і спеціальних засобах використання цих «гіперпосилань» для підтримки відповідної навігації.

Всі згадані вище методики базуються на використанні добре відомих Веб браузерів (наприклад, Microsoft Internet Explorer, Mozilla, Opera і ін.) для підтримки навігаційної діяльності користувача. І все таки браузери на глибинному рівні використовують, як правило, засновану на HTML або XML розмітку сторінок, вбудовану в ці сторінки їх авторами і / або розробниками Веб-сторінок. Таким чином, фактично, користувач може здійснювати навігацію по анотований гіперпосиланнями сторінці тільки заздалегідь визначеним чином, що може не відповідати його (її) поточним потребам. Така навігація трудомістка і вимагає багато часу і в багатьох випадках не забезпечує отримання корисних результатів.

З огляду на вищесказане, було б бажано мати інтелектуальні засоби навігації по колекціях документів, таких як Інтернет і / або корпоративні сховища знань (Knowledge Warehouses), що дозволяють користувачеві вільно фіксувати серед документів, що переглядаються тільки ті, які є семантично значущими з точки зору поточних потреб користувача, об'єктів і відносин між ними, присутніх в розглянутих документах без опори на зумовлені гіперпосилання. Такі засоби повинні також забезпечувати більш зручну методику для інтелектуальної навігації, ніж існуючі засоби навігації, забезпечені Веб-браузерами.

Специфіка робота фахівців по роботі з автомобілями полягає в постійному аналізі інформаційних потоків, тому для успішної діяльності їм необхідні засоби інтелектуального аналізу і моніторингу пропозицій на

автомобільному ринку. Велика частина такої інформації є слабкоструктурованих і при традиційному підході роботи з нею займає значну частину часу. Джерелами інформації для ділера є тематичні Інтернет-ресурси, паперові видання і спеціалізовані бази даних.

Інтернет-ресурси і сервіси, що акумулюють існуючі пропозиції на автомобільному ринку, прийнято називати агрегаторами. Основними характеристиками даних сервісів, що впливають на затребуваність користувачами, є повнота бази об'єктів, актуальність даних, достовірності інформації, можливості пошуку і фільтрації і ціна доступу.

Завдання агрегації інформації з різних джерел і її структуризація є надзвичайно актуальною.

Існуючі на даний момент ресурси можна класифікувати за двома ознаками: територіальному охопленням бази об'єктів і способу організації роботи з контентом. У першій класифікації виділяють два класи: глобальні, створені на платформі відомого порталу (olx.ua, trovit.com, lun.ua) і локальні, які стосуються регіональних проектів по автомобільному ринку. Друга класифікація передбачає поділ на описані нижче класи.

Дошка оголошень з'явилася однією з перших. Зазвичай це безкоштовна, тематично організована база даних. Професійною мовою це так звана «брудна» база, тобтонеорганізовані, практично не регламентовані системи.

Наступним важливим агрегатором інформації є електронні версії друкованих видань приватних оголошень. Одним з головних переваг, за оцінками експертів, дозволили цим ресурсам зайняти провідне місце в своїх ринках, є поєднання концепції газети безкоштовних оголошень з електронною версією.

Як правило, централізовані бази даних автомобілів доступні тільки для агентів з продажу автомобілів (наприклад, у США), які є членами асоціації автоділерів або груп асоціацій. У сукупності, ці бази даних називають Multiple Listing Service (MLS).

Мультилістингові системи є найбільш популярним і затребуваним видом ресурсів серед професійних автоділерів на Заході. Але в Україні на сьогодні

немає глобального порталу, який би об'єднував інформацію про всі пропозиції на автомобільному ринку.

Інформаційні портали по автомобілям або спеціалізовані сайти на сьогоднішній день - найбільш поширені агрегатори інформації про автомобільний ринок в мережі Інтернет.

Соціальні мережі також відносять до агрегаторами інформації. Проте, зараз спостерігається зближення сайтів-агрегаторів і соцмереж з точки зору спільних рис і застосовуваних сервісів.

Мета-агрегатори - клас систем, які об'єднують пропозиції з декількох ресурсів. Дані сервіси мають додаткові функції, наприклад, інтелектуальна фільтрація оголошень тільки від власників, а не від посередників.

Хоча MLS надає цінну інформацію, в MLS неефективне відображає реальні тенденції автомобільного ринку. Тенденції в сфері автомобільного ринку відбуваються з плином часу і викликають помітну зміну в загальному напрямку на ринку автомобілів. Коли якісь тенденції починаються, їх вплив не відображаються в інформаційних і податкових оцінках протягом декількох місяців (наприклад, до тих пір поки ціни продажу не оновляться) викликає період «лагу». У цей період «лагу», інформація MLS не надає продавцям і покупцям точну інформацію для визначення реальної вартості їх автомобіля.

Для полегшення операцій, учасники галузі повинні мати точну і актуальну інформацію. Щоб ефективно вести свою господарську діяльність учасники ринка комерційних автомобілів та пов'язаної з ними бізнес-спільноти вимагають щоденного доступу до поточних даних, таких як вартість податкових зборів, поточний попит на авто, рух орендарів авто, постачання, випуск нових авто, а також про інші важливі подіях на ринку. Такий збір даних забирає багато часу, як показано в 1996 році дослідження, в якому виявили, що комерційні фахівці з продажу подержаних автомобілів витратили 40% свого робочого часу [16] для збору і аналізу інформації про автомобільний ринок. Таким чином, існує необхідність в створенні єдиної моделі ефективність такого ринку а комерційних даних про нерухомості, де учасники автомобільного ринку і пов'язані з ними бізнес-спільноти можуть обмінюватися інформацією,

оцінити можливості з використанням національних стандартизованих даних, а також взаємодіяти один з одним на безперервній основі.

При використанні запропонованого способу споживач (оцінювач або інший фахівець у сфері продажу/купівлі автомобілів), який шукає інформацію в Інтернет, отримує точну інформацію (а не посилання, де можна знайти потрібну інформацію) в зручній для роботи і порівняння формі файлу. Користувачеві не доведеться ходити по сайтах в пошуках інформації, шукати на сайті місце, де лежить потрібна інформація, копіювати, приводити її в стандартизований вид для зручності порівняння, вибудовувати за географічними, якісними і кількісними ознаками. Споживач отримує файл (або базу даних), готовий до роботи. Таким чином, пошук і, найголовніше, збір інформації переходять на новий якісний рівень.

1.6 Формулювання завдань роботи

Метою даної дипломної роботи є дослідження методів інтелектуального аналізу веб-контенту, а також огляду технологій автоматичного збору інформації з відкритих Інтернет-джерел у сфері продажу автомобілів. Дані, які необхідно зібрати, витягнути і проаналізувати, уявляють собою оголошення щодо пропозицій до продажу автомобілів. Знайдена відповідна інформація буде розміщуватися в базі даних з можливістю подальшої роботи з інформацією про відібрані об’єкти автомобільного ринку.

Наповнювана база даних має містити відібрані оголошення з посиланнями на веб-сторінки та інформацію про адресу, ціну, характеристики, рік виробництва, стан, фотографії автомобілів, дату розміщення оголошення, тощо.

Для збору вихідних даних буде використовуватись такі веб-сайти, що місять оголошення щодо обьєктів автомобільного ринку:

Найменування веб-сайтів для видобутку даних	Примітки
https://www.olx.ua	Структурований текст
https://auto.ria.com/	Структурований текст
http://www.board24.lg.ua	Неструктурований текст

Пропонованим рішенням описаної задачі є розробка інтелектуального сервісу, що збирає інформацію про пропозиції на ринку продажу автомобілів з різних визначених джерел в єдину базу даних.

2 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ВЕБ КОНТЕНТУ

В даному розділі проводиться аналіз деяких методів автоматичного отримання інформації, і підходів до формування правил вилучення даних.

Для автоматичного отримання даних з Web-сайтів механізми виконання запитів взаємодіють з безліччю обгорток (wrapper). Обгортка - це специфічна для кожного Web-сайту програма, завданням якої є трансляція даних Web-сайту в форму, що дозволяє здійснювати подальшу їх обробку засобами системи інтеграції даних. Наприклад, обгортка може витягувати безліч кортежів з HTML-файлу.

Обгортка у видобутку даних являє собою програму, яка витягує вміст конкретного джерела інформації і перетворює його в реляційної формі. Багато веб-сторінок мають структуровані дані - телефонні довідники, каталоги продукції тобто відформатовані для перегляду людиною за допомогою мови HTML. Структуровані дані як правило це описи об'єктів, які отримані з основних баз даних і відображаються на веб-сторінках після певних фіксованих шаблонів. Програмні системи, що використовують такі ресурси повинні переводити вміст HTML в реляційну форму. Обгортки зазвичай використовуються в якості таких трансляторів. Формально обгортка є функцією від сторінки до набору кортежів, які вона містить.

Є два основні підходи до генерації обгортки (або екстрагування інформації): *індукційна обгортка* [17] (Wrapper induction, WI) та *автоматизоване вилучення даних*. Індукційна обгортка використовує контрольоване навчання, щоб дізнатися правила отримання даних з вручну мічених навчальних прикладів. Вона спрямована на структуровані та напівструктуровані документи типа веб-сторінок.

В останні кілька років були запропоновані багато підходів до побудови систем індукційної обгортки, виключаючи машинне навчання та техніки видобутку по шаблону з різним ступенем автоматизації. У цьому розділі ми розглядаємо раніше запропоновану таксономію для інструментів видобутку інформації (Information Extraction, IE), розроблених основними дослідниками.

Конференція по Розумінню Повідомлень [18] (Message Understanding Conference, MUC), яка була організована за підтримки DARPA (1987-1997) надихнула ранні роботи у видобутку інформації. Визначено п'ять основних завдань для текстового видобутку інформації, в тому числі розпізнавання іменованих об'єктів, роздільна здатність кореферентності, побудова елемента шаблону, побудова відносин шаблону і побудова шаблону сценарію. Значимість MUCs в області IE мотивує деяких дослідників класифікувати підходи IE в два різних класи: MUC підходи (наприклад, AutoSolg [19], LIEP [20], PALKA [21], HASTEN [22], та CRYSTAL [23]) та пост-MUC підходи (наприклад, WHISK [24], RAPIER [25], SRV [26], WIEN [27], SoftMealy [28] та STALKER [29]).

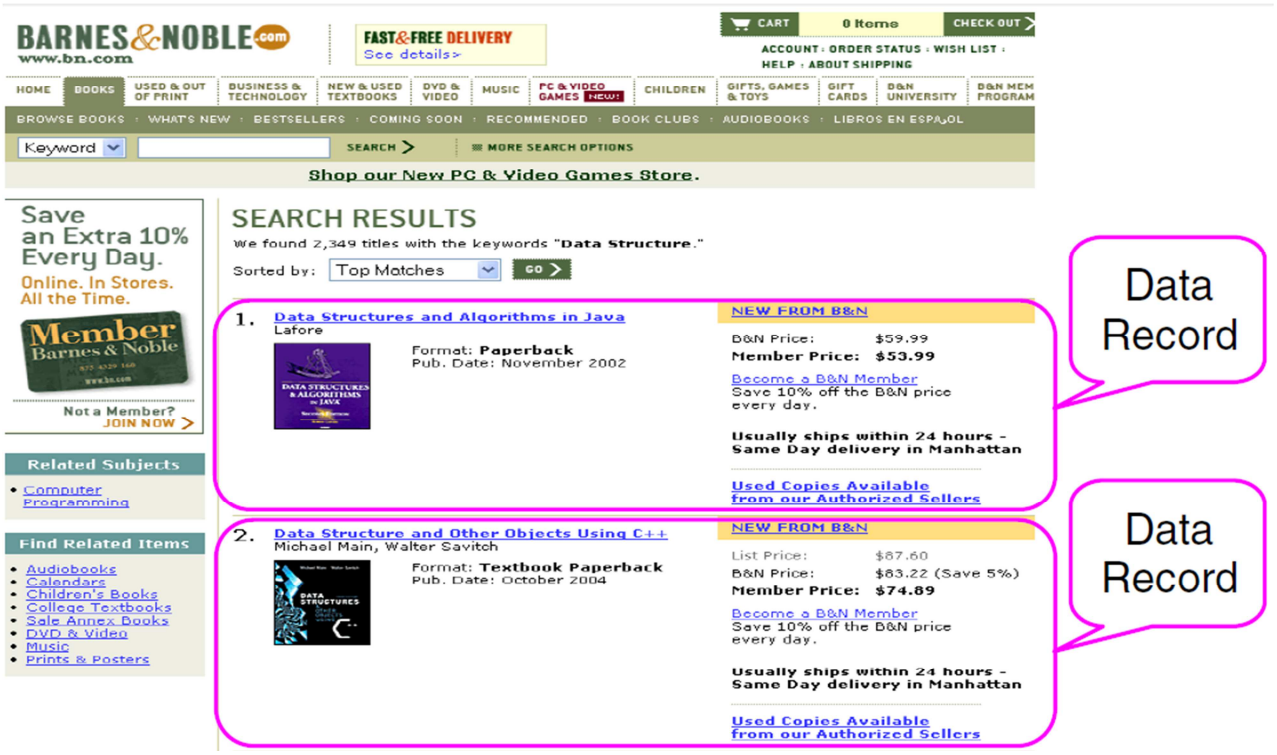


Рисунок 2.1. Напівструктурована сторінка, що містить записи даних (в прямокутних коробках) має бути вилучена.

Хсу і Зунг [30] класифікували обгортки на 4 різних категорії, в тому числі обгортки ручної роботи з використанням загальних мов програмування, спеціально розроблені мови програмування або інструменти, обгорток на евристичний основі та WI підходи. Чанг [31] продовжив цю таксономію

та порівняв системи WI з точки зору користувача і розподілив інструменти IE на основі ступеня автоматизації. Вони класифікували інструменти IE по чотирьох різним категоріям, включаючи системи, які потрібні програмістів, системи, які потребують зразків анотацій, систем вільних від анотацій та напівконтрольованих систем.

Айон Масли, який підтримує веб - сайт RISE [32] (Репозитарій мережевих джерел інформації, що використовуються в завданнях по видобутку даних), класицікував інструменти IE на 3 різних класи в залежності від типу вхідних документів і структури / обмежень шаблонів видобутку [33]. Перший клас включає в себе інструменти, що обробляють IE з довільного тексту з використанням шаблонів витягу, які в основному базуються на синтетичних/ семантичних обмеженнях. Другий клас називається системами індукції обгорток, які покладаються на використанні правил на основі роздільників, оскільки завдання IE оброблюють онлайн документи, такі як HTML-сторінки. І, нарешті, третій клас також обробляє IE з онлайн-документів; проте шаблони цих інструментів засновані як на роздільниках, так і на синтаксичних / семантичних обмеженнях.

Кашмерик класифікував багато з інструментів IE на дві окремих категорії інструменти кінцевого стану та реляційного навчання [34]. Правила видобутку в інструментах кінцевого стану формально еквівалентні регулярні граматикам і автоматам, наприклад WIEN, SoftMealy і STALKER, в той час як правила видобутку в інструментах реляційного навчання, по суті, в формі Пролого-подібні логічні програми, наприклад, SRV, Crystal, WebFoot [35], Rapier та Pinocchio [36].

Лаендер запропонував таксономию для інструментів вилучення даних на основі головної методики, яку використовує кожен інструмент для формування обгортки [37]. До них відносяться мови для розробки обгорток (наприклад, Minerva [38], TSIMMIS [39] та WebOQL [40]), HTML-залежні інструменти (наприклад, W4F [41], XWrap [42] та RoadRunner [43]), засновані на NLP інструменти (наприклад, WHISK, RAPIER та SRV), інструменти індукційних обгорток (наприклад, WIEN, SoftMealy і STALKER), моделювання на основі

інструмент (наприклад, NoDoSE [44] та DEByE [45], [46], та інструменти на основі онтологій (наприклад, BYU [47]). Лаендер порівняв інструменти за допомогою 7 наступних характеристик: ступінь автоматизації, підтримка комплексних об'єктів, зміст сторінки, наявність графічного інтерфейсу користувача, XML вихід, підтримка не HTML джерел, гнучкість та адаптивність.

Сараваджи класифікував HTML обгортки на 3 категорії відповідно до типів завдань екстракції [48]. Перша категорія - обгортки на рівні записів, використовують закономірності для встановлення меж записів, а потім витягують елементи єдиного списку однорідних записів зі сторінки. Друга категорія, обгортки на рівні сторінки, витягують елементи декількох видів записів. Нарешті, обгортки на рівні сайту наповнюють базу даних зі сторінок веб-сайту.

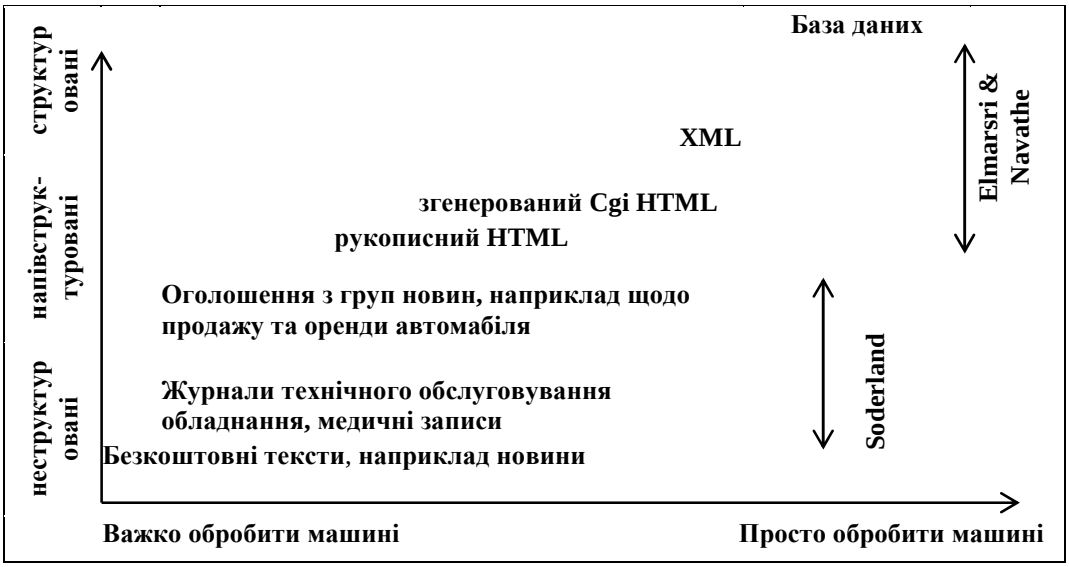


Рисунок 2.2. Структурування різних документів.

Кахлінс та Тредвел класифікували інструментарії для створення обгортки на дві основні категорії, на основі комерційної та некомерційної доступності [49]. Вони також протиставили інструментарії за допомогою деяких характеристик, таких як метод виведення, тип інтерфейсу, можливості веб-обходу і підтримка графічного інтерфейсу.

Для оцінки систем IE пропонується три основних виміри [50]. По-перше, відмінність довільного тексту IE і онлайн документів, зроблена Маслі, завдання

трьохрівневого вилучення запропонована Сараваджи, а також можливості обробки не HTML джерел, разом представляють перший вимір, що стосується труднощі доменного завдання до якого відноситься і завдання IE. По-друге, систематика правил регулярних виразів або Пролого-подібних логічних правил, і детермінованих кінцевих станів перетворювача або імовірнісно прихованих моделей Маркова, вказують на другий вимір, який стосується основних методів, що використовувані в системах IE. Нарешті, категоризації залучених програмістів, заснований на навчанні або вільний від анотацій підходи передбачають третій вимір, що стосується рівня автоматизації.

2.1 Три виміри для порівняння систем видобутку інформації

Продовжуючи аналіз різних таксономій, є три виміри, які будуть використовуватися при порівнянні. Перший вимір оцінює складність завдання з видобутку інформації (IE), який може бути використаний для відповіді на питання «чому системі IE не вдається впоратися з певними структурами деяких веб-сайтів?» Другий вимір порівнює методи, що використовуються в різних системах IE. Третій вимір оцінює як зусилля, зроблені користувачем для навчального процесу так і необхідності в поширенні систем IE на різні сфери діялності. З точки зору користувача, другий вимір є найменш важливим. Тим на менш, дослідники під час порівняння могли б отримати огляд який саме метод машинного навчання або інтелектуального аналізу даних був використаний для індукційної обгортки (WI). Кожен з цих вимірів наведений нижче, і для кожного з них обрано набір характеристик, які можуть бути критеріями для порівняння та оцінок систем IE з точки зору таких вимірів.

2.1.1 Складність завдання

Вхідний файл із завданням IE може бути структурованим, слабоструктурованим або довільним текстом. Як показано на рисунку 2.2., визначення цих термінів варіюється в залежності від областей досліджень.

Содерленд [51] звернув увагу що довільні тексти, наприклад статті новин, що написані на природних мовах, є неструктурованими, публікації в групах новин (наприклад, оренда автомобілів), медичні записи, журнали обслуговування обладнання є напівструктурированими, в той час як HTML сторінки структуровані. Однак, з точки зору дослідників баз даних [52], інформація, що зберігається в базах даних відома як структуровані дані; XML документи - це слабоструктуровані дані оскільки в інформаційну схему додається також значення даних, в той час як веб-сторінки на HTML є неструктурованими, тому що є дуже обмежена вказівка на тип даних. З нашої точки зору, XML - документи розглядаються як структуровані, оскільки є DTD або XML-схеми, які можна використовувати для опису даних. Довільні тексти неструктуровані, оскільки вони потребують суттєвої обробки природної мови. Для великого обсягу HTML-сторінок в Інтернеті, вони розглядаються як напівструктуровані [53], оскільки вбудовані дані часто безперебійно виявляються з використанням HTML - тегів.

Таким чином, напівструктуровані вхідні ресурси є документами досить регулярної структури, а дані в них можуть бути представлені в HTML або не HTML форматі. Одним з джерел цих великих напівструктурованих документів з глибинна мережа (deep web / hidden web) – частка всесвітньої мережі інтернет, вміст якої за жодних обставин не індексують стандартні пошукові онлайн-системи. У глибинній мережі знаходяться веб-сторінки, не пов'язані з іншими гіперпосиланнями (наприклад, тупикові веб-сторінки, динамічно створювані скриптами на самих сайтах за запитом на які не ведуть прямі посилання), а також сайти, доступ до яких відкритий тільки для зареєстрованих користувачів та веб-сторінки, доступні тільки по паролю. Глибинна мережа включає в себе динамічні веб-сторінки, які створюються зі структурованих баз даних з деякими шаблонами або макетами. Наприклад, набір сторінок книг з Amazon має один і той же макет для авторів, назв, ціни, коментарів і т.д. Веб - сторінки, які створюються з однієї й тієї ж бази даних одним й тим же шаблоном (програмою) утворюють клас сторінки. Є також слабоструктуровані HTML-сторінки, створені вручну. Наприклад, списки публікацій з домашніх сторінок

різних дослідників всі мають назву і джерело для кожного окремої публікації, хоча вони й виробляються різними людьми. Для багатьох задач IE, вхідними ресурсами є сторінки одного й того ж класу, однак деякі завдання IE зосереджуються на витяганні інформації зі сторінок різних веб-сайтів.

У доповнення до категоризації за вхідними документами, виконання завдання IE може бути класифіковано відповідно до мети екстракції. Наприклад, Сараваджи класифікував HTML обгортки щодо завдань IE на рівні записів, на рівні сторінок та на рівні сайту. Обгортки на рівні запису виявляють межи записів, а потім розділюють їх на окремі атрибути; обгортки на рівні сторінок витягують всі дані, що вбудовані в одну веб-сторінку, в той час як обгортки на рівні сайту заповнює базу даних зі сторінок веб-сайту, таким чином, атрибути об'єкта екстракції розкидані по сторінках веб-сайту. Академічні дослідники присвятили багато зусиль для розробки рівня записів та вилучення даних на рівні сторінок, в той час як практичні дослідники мають більший інтерес до повних наборів, які підтримують вилучення даних на рівні сайту.

Існують різні способи, щоб описати цілі для вилучення на сторінці. Найбільш загальна структура (як це запропоновано в NoDoSE, DEByE, та Stalker тощо) являє собою ієрархічне дерево, де листові вузли є основними типами в той час як внутрішні вузли є списком кортежів. Об'єкт даних може бути простою / вкладеною структурою. Простий текстовий об'єкт даних має тільки один внутрішній вузол (корінь), в той час як вкладені дані об'єкту містять більше двох рівнів та внутрішні вузли. Оскільки ці веб-сторінки призначені для читання людиною, кортежі одного й того ж списку, або елементи кортежу часто явно розділені або розмежовані для полегшення візуального сприйняття. Тим не менше, представлені формати або набори атрибутів, що утворюють об'єкт даних, є предметом для наступних варіантів:

– Атрибут може мати нуль або більше значень (список 1 кортежу) в об'єкті даних. Якщо атрибут має нульове значення, це називається *відсутній* атрибут; якщо він має більш ніж одне значення, він називається *багатозначним* атрибутом. Ім’я автора книги може бути прикладом

багатозначного атрибута, в той час як спеціальна пропозиція, яка існує тільки для певних книг, є прикладом відсутності атрибуту.

– Набір атрибутів ($A_1, A_2, ..., A_k$) може мати декілька упорядкувань, тобто атрибут A_i може мати варіантні позиції в різних екземплярах об'єкту даних. Назвемо цей атрибут *багато-впорядкованим*. Наприклад, сайт фільмів може перерахувати дати випуску перед назвами фільмів до 2017 року, але після назви для нових фільмів.

– Атрибут може мати формати варіантів поряд з різними екземплярами об'єктів даних. Якщо формат атрибуту не є фіксованим, то можливо знадобляться диз'юнктивні правила для узагальнення всіх випадків. Наприклад, сайт електронної комерції може містити ціни жирним шрифтом за винятком продажних цін червоним кольором. Таким чином, ціна може бути прикладом атрибута варіанту формату на цьому сайті. З іншого боку, різні атрибути об'єкту даних можуть мати той же формат, особливо в табличній формі, де одиночні теги <TD> використовуються для представлення різних атрибутів. У таких випадках порядок атрибутів є ключовою інформацією для розрізнення різних атрибутів. Однак, якщо трапляються відсутні ознаки або існує багато-впорядкованість, правила вилучення цих атрибутів повинні бути переглянуті.

– Більшість систем ІЕ обробляють вхідні документи як строки лексем оскільки їх легше обробляти, ніж строки символів. В залежності від використаних лексичних методів, іноді атрибут не може бути розкладений на окремі лексеми. Такий атрибут називається атрибутом *що не розбивається на лексеми (untokenized).* Наприклад, каталозі предметів училища код відділу не має роздільника, що відокремив би його від числа, в строках таких як «COMP4016» або «GEOL2001». Рівень деталізації цілей екстракції впливає на рішення / вибір схем токенізації для системи ІЕ.

Поєднання різних вхідних документів та зміна цілей екстракції спричиняє різні ступені складності завдання. Оскільки різні системи ІЕ призначені для вирішення різних завдань ІЕ, не доцільно порівнювати їх безпосередньо. Однак, аналізування яке завдання системи ІЕ має за ціль та як вона виконує завдання,

може бути використано для оцінки цієї системи та, ймовірно, може поширено на інші сфери завдань.

2.1.2 Використані методи

Обгортці для отримання даних з вхідних ресурсів необхідно розбити на лексеми (tokenize) вхідну строку, застосувати правила вилучення для кожного атрибута, зібрати витягнуті значення в записи, та повторити процес для всіх екземплярів об'єкта на вході. Існують різні рівні деталізації для токенізації вхідної строки, включаючи кодування на рівні тегів та на рівні слова. Перше кодування переводить кожен HTML - тег в якості маркеру (token) та переводить будь-яку текстову строку між двома тегами, як спеціальний маркер, в той час як пізніше, на рівні слова, розглядає кожне слово в документі в якості маркеру. Правила екстракції можуть бути викликані узагальненням зверху вниз або знизу вверх, видобутком за шаблоном або логічним програмуванням. Тип правил екстракції може бути виражений за допомогою регулярних граматик або логічних правил. Деякі з систем WI використовують шляхові вирази шляху дерева розбору HTML (наприклад, html.head.title та html->table[0]) в якості ознак в правилах видобутку; деякі використовують синтаксичні або семантичні обмеження, такі як POS-теги і семантичний клас WordNet; в той час як інші використовують обмеження на основі роздільників, таких як HTML - теги або літерні слова, в правилах видобутку. Архітектура екстрактору може вимагати один або кілька проходів по сторінках.

Таким чином, можливості для порівняння систем WI з точки зору методів, використовуваних включають: *схеми токенізації / кодування, кількість проходів сканування, тип правила екстракції, присутні особливості, та алгоритм навчання.*

2.1.3 Ступінь автоматизації

Як було описано вище, в програмі обгортки є багато фаз які мають бути виконані: збір навчальних сторінок, маркування навчальних сторінок,

узагальнюючі правила вилучення, витяг відповідних даних, та вивід результату у відповідному форматі. Більшість дослідників зосереджені на проміжних 3-х фазах, які включають основний процес видобутку, в той час як деякі забезпечують повне рішення, що включає краулер або пошуковий бот для збору навчальних сторінок (перша фаза) та підтримку виведення в форматі XML або серверну частину реляційної бази даних для подальшої інформаційної інтеграції (заключний етап). Взагалі кажучи, фаза маркування визначає / обирає вихід завдання вилучення та вимагає участі користувачів. Однак, деякі системи WI не вимагають зібраних прикладів навчання для маркувати до стадії навчання, натомість, маркування або анотування витягнутих даних може бути зроблено після генерації правил видобутку (за участі користувачів або без). Це викликає велику різницю в автоматизації: для деяких систем WI, користувачеві необхідно маркувати приклади навчання; для інших, користувач просто чекає коли система вичистить сторінки та вилучить дані. Однак автоматизація не приходить без обгрунтування. Її вартістю є можливість застосування цих підходів до іншої сфери завдань. Деякі навіть мають обмеження числа та типу вхідних сторінок.

Таким чином, характеристики систем вилучення веб-інформації з точки зору ступеня автоматизації можуть включати: досвід користувача, необхідний для маркування даних або для побудови правил вилучення, применимість цих підходів до іншої сфери завдань, обмеження на кількість/тип вхідних ресурсів, підтримка завантажень сторінки для збору навчальних сторінок, підтримка виходу та підтримки API для інтеграції додатків.

Розглядаються підходи які використовуються в таких розробках як TSIMMIS, XWRAP, WHISK, SRV, RoadRunner, DEPTA, IEPAD. Виділено критерії для порівняння різних підходів.

Ручні	Керовані	Напів-керовані	Не керовані
TSIMMIS **[Hammer1997]**	WIEN [Kushmerick1997]	**IEPAD** **[Chang2001]**	RoadRunner [Crescenzi2001]
Minerva [Crescenzi1998]	**SRV** **[Freitag1998]**	OLERA [Chang2004]	DeLa [Wang2002]
WebQOL [Arocena1998]	RAPIER [Califfl 998]	Thresher [Hogue2005]	EXALG [Arasu2003]
XWRAP **[Liu2000]**	NoDoSe [Adelberg1998]	IDE [Zhai2005]	**DEPTA** **[Znai2005]**
W4F [Saiiuguet2001]	SoftMealy [Hsu1998]		NET [Zhai2005]
	WHISK **[Soderland1999]**		IEKA [Wong2007]
	STALKER [Muslea1999]		ViDE [Liu2010]
	DEByE [Laender2002]		

Рисунок 2.3. Існуючі методи аналізу веб контенту

2.2 Аналіз методичного інструментарію

Програми які займаються витяганням даних з веб документів називаються парсером. Більшою мірою веб документи є текстовими файлами, тому завдання вилучення зводитися до аналізу текстових даних. Основною проблемою аналізу таких даних є те що дані є напівструктурованими, а іноді і неструктурованими. Незважаючи на те що веб документи представлені html файлами, що містять розмітку на стандартизованій мові, дана розмітка всього лише описує зовнішній вигляд документа, а опис конкретної суті залишається вільним.

Процес вилучення даних з веб документа можна розділити на два етапи. Перший етап - аналіз структури веб документа. На даному етапі проводитися виявлення закономірностей і правил для отримання даних. На другому етапі відбувається вилучення даних по сформульованим правилам.

Підхід до формування правил в першу чергу залежить від методу яким будуть вилучатись дані. Розглянемо основні методи отримання інформації:

1) Аналіз DOM дерева, використання XPath.

2) Посимвольний аналіз.

3) Парсинг рядків.

4) Використання регулярних виразів.

5) XML парсинг.

6) Візуальний підхід.

2.2.1 Видобуток даних за допомогою аналізу DOM структури та з використанням XPath

При отриманні даних з HTML або XML документів вхідні дані на відміну від попередніх методів розглядаються як структура даних. На першому етапі з вхідний текстової інформації будується об'єктна модель документа - DOM дерево. DOM - це не залежить від платформи і мови програмного інтерфейсу, що дозволяє програмам і скриптам отримати доступ до вмісту HTML-, XHTML- і XML-документів [54]. На другому етапі відбувається вилучення вмісту з вузлів дерева, за певними правилами.

Наступним етапом аналізу DOM дерева є використання стандартизованої мови запитів XPath [55] - тобто засобу, що широко використовується при парсингу XML даних. Суть даного підходу в тому, щоб за допомогою простого синтаксису описувати шлях до елементу без необхідності поступового руху вниз по DOM дереву. Простий запит на мові xpath є шляхом до вузла дерева. Приклад такого запиту наведено нижче

```
1.  .//div/ul/li/div/li/p/a
```

Поглянемо на цей запит, ".//" - рекурсивний спуск на нуль або більше рівнів ієрархії від поточного контексту, в даному випадку запит буде витягувати щодо кореня документа. Далі слідує шлях до конкретного елемента, кроки розділені роздільником - "/". В даному випадку у всіх вузлах ul будуть знайдені всі вузли li, потім у знайдених вузлах li будуть знайдені всі вузли div і.т.д Таким чином шлях описує рекурсивний спуск по DOM дереву. Також мова xpath дозволяє фільтрувати вузли по їх атрибутам, визначати перший або останній дочірній елемент, сестринські елементи і.т.д. Таким чином можна

виділити вузли з певними атрибутами. Наприклад, певні елементи на веб сторінці можуть мати певний клас (атрибут class), який використовується для каскадних таблиць стилів.

Даний метод є більш ефективним при розборі великих веб сторінок ніж розбір попередніми методами. Додатковий рівень абстракції у вигляді DOM дерева дозволяє спростити правила отримання інформації. Однак варто враховувати що корисна інформація може міститися всередині вузла дерева у вигляді простого тексту, для вилучення такої інформації необхідно використовувати методи описані раніше.

Переваги підходу:

– можна отримати дані будь-якого типу та будь-якого рівня складності;

– знаючи розташування елемента, можна отримати його значення, прописавши шлях до нього.

Недоліки підходу:

– різні HTML / JavaScript движки по-різному генерують DOM дерево, тому потрібно прив'язуватися до конкретного движку;

– шлях елемента може змінитися, тому, як правило, такі парсери розраховані на короткочасний період збору даних;

– DOM-шлях може бути складний і не завжди однозначний.

2.2.2 Видобуток даних шляхом посимвольного аналізу

Метод посимвольного аналізу - метод в якому вхідні дані розглядаються як масив символів. Процес формування правил зводитися до виявлення закономірностей в послідовності символів. Даний підхід є самим низькорівневим методом вилучення даних.

2.2.3 Видобуток даних шляхом аналізу (парсингу) рядків

Іноді дані відображаються за допомогою деякого шаблону (наприклад, таблиця характеристик мобільного телефону), коли значення параметрів стандартні, а змінюються лише їх значення. У такому випадку дані можуть бути отримані без аналізу DOM дерева, а шляхом парсинга рядків.

Хоча цей підхід і не можна застосовувати для написання серйозних парсерів, використання набору методів для аналізу рядків іноді (частіше - простих шаблонних випадках) більш ефективний ніж аналіз DOM дерева або XPath.

2.2.4 Видобуток даних з використанням регулярних виразів

Наступний метод це вилучення даних за допомогою регулярних виразів. Регулярні вирази - формальна мова пошуку і здійснення маніпуляцій з підрядками в тексті, заснований на використанні метасимволів. Даний метод дозволяє описати закономірності символів на формальній спеціалізованій мові, що спрощує вилучення даних щодо попереднього методу. Регулярні вирази добре підходять для вилучення даних які мають строгий формат, наприклад, номери телефонів, банківських рахунків, календарних дат, часу, поштових скриньок і.т.д.

З точки зору теорії формальних мов, регулярні вирази складаються з констант і операндів. Константами є: порожня множина, символьний літерал, порожній символ, множина. Можливі операції над константами: конкатенація, диз'юнкція, чергування, замикання Кліні.

Нижче наведено приклад регулярного виразу який витягує тип палива (газ, евро4, евро5 та інші), які записані після фрази "газ". Тип палива закінчується крапкою:

```
1.  /.*газ:\s*([а-я\s]+)+\./
```

2.2.5 Видобуток даних з використанням XML парсингу

XML парсинг є неефективним підходом оскільки розглядає HTML як XML дані. Тому HTML рідко буває дієвим, тобто таким, що його можна розглядати як XML дані. Бібліотеки, що реалізують такий підхід, більше часу приділяють перетворенню HTML в XML і вже потім безпосередньо парсингу даних. Тому цього підходу краще уникати.

2.2.8 Видобуток даних з використанням візуального підходу

В даний момент візуальний підхід знаходиться на початковій стадії розвитку. Суть підходу в тому, щоб користувач міг без використання програмного мови або API «налаштувати» систему для отримання потрібних даних будь-якої складності і вкладеності. Досить ймовірно, що парсери майбутнього будуть саме візуальними.

2.3 Аналіз процесу формування правил

Метою WI є автоматична генерація обгортки, яка використовується для вилучення цілей з інформаційних ресурсів. Розглянемо як користувач взаємодіє з системою WI. Раніше системи призначались для допомоги програмістам у написанні правил вилучення, в той час як більш пізні системи представляють машинне навчання для автоматичного узагальнення правил. Таким чином, взаємодія з користувачем еволюціонувала від написання правил вилучення даних до маркування цілі вилучення. В останні роки все більше зусиль спрямовувалось на скорочення маркування та створення систем WI з немаркованими навчальними прикладами. Дотримуючись цієї тенденції, системи WI подалі частіше поділяють на чотири класи [56]: *побудовані вручну* IE системи, *контрольовані* IE системи, *напівконтрольовані* IE системи та *неконтрольовані* IE системи.

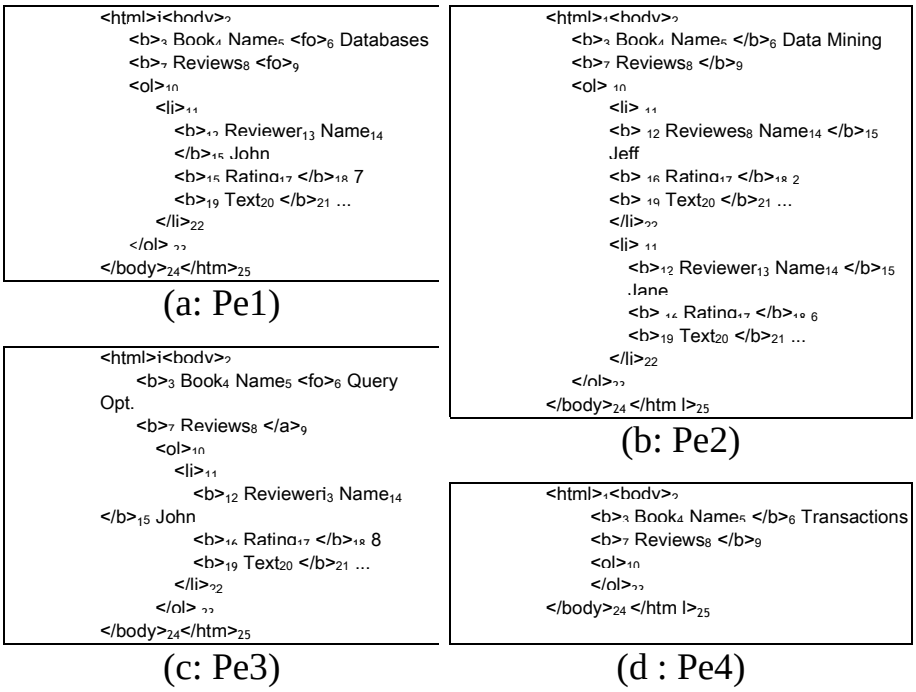


Рисунок 2.4 – Зазначений приклад чотирьох веб - сторінок (pe1-pe4).

При ручному підході правила описуються або прикладною мовою програмістом, або у спеціальному файлі конфігурації користувачем для кожного сайту. Даний підхід при своїй трудомісткості дозволяє максимально точно визначити правила для конкретного джерела даних. Контрольовані методи як правило надають користувачеві графічний інтерфейс та алгоритми засновані на методах машинного навчання. Такі системи на вході приймають документи для обробки, та приклади даних для навчання, на виході система формує ряд правил для отримання даних. Неконтрольовані системи не вимагають навчальних прикладів та іншої взаємодії з користувачем при формуванні правил вилучення контенту.

На рис. 2.5 показано застосування різних методів в залежності від типу аналізованих даних.

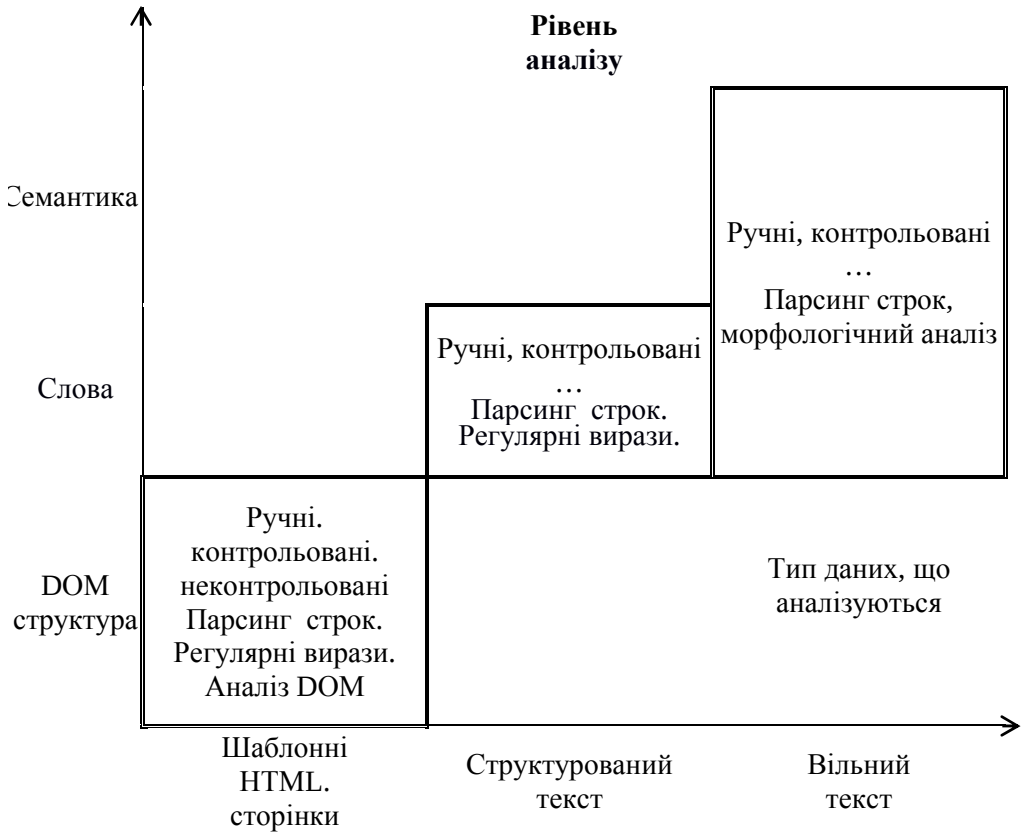


Рисунок 2.5 – Застосування методів аналізу даних в залежності від типу даних

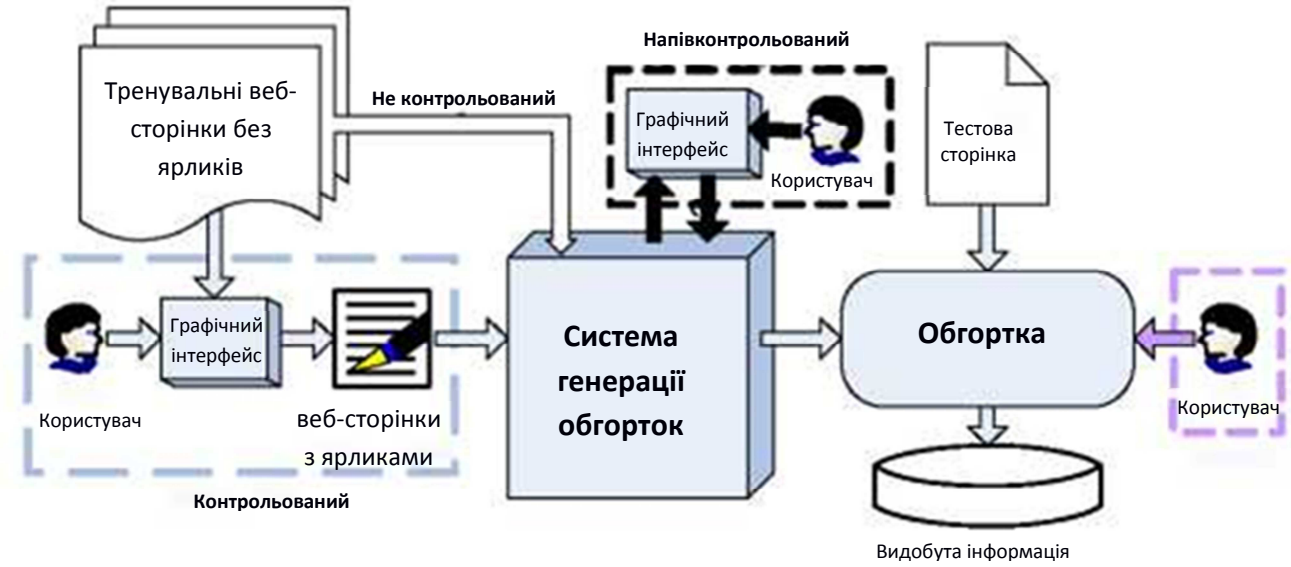


Рисунок 2.6 – Загальний вигляд систем генерації обгорток

Нижче надамо огляд найбільш відомих і сучасних підходів IE. Для того, щоб зробити такі підходи більш зрозумілими, розглядається загальне завдання IE і описується згенерована обгортка, яка може бути використана для отримання інформації з інших аналогічних документів для кожного підходу. На

рисунку 2.4 показані чотири веб-сторінки в якості вхідних даних задачі IE. Бажаним виходом є назва книги і відповідні огляди, в тому числі ім’я оглядача, рейтинг та коментарі.

2.4 Огляд підходів до формування правил видобутку інформації

2.4.1 Ручне формування правил видобутку інформації

Як показано на правому боці рисунку 2.4, в побудованих вручну системах IE, користувачі програмують обгортку для кожного веб – сайту вручну з використанням загальних мов програмування, таких як Perl або за допомогою спеціальних розроблених мов. Ці інструменти вимагають, щоб користувач мав суттєві комп'ютерне та програмне оточення, тому стає дорогим. Такі системи включають TSIMMIS, Minerva, Web-OQL, W4F і XWRAP.

Розробка TSIMMIS [57] ставитися до першої категорії ручного формування правил [58]. Основним компонентом цього проекту є оболонка, яка приймає в якості вхідних даних в файл специфікації, який декларативно затверджує (послідовність команд даної програмістами), де на сторінках знаходяться корисні дані і як дані повинні бути «упаковані» в об'єкти. Наприклад, на рисунку 6(а) показаний файл специфікації для задачі IE з рисунку 2.4. Кожна команда має вигляд: [змінні, джерело, шаблон], де джерело визначає вхідний текст, який буде розглянутий, шаблон безпосередньо описує правило добування інформації та визначає, як знайти інтересний текст в джерелі, і змінні це список змінних, в які записується витягнуті результати.

```
1  [ "root", "get('pe1 .html')",
2  [ 'Book", "root", "*<body>#</body>"],
3  [ 'BookName", "Book", "*</b>#<b>"],
4  [ 'Reviews", "Book", "*<ol >#</ol >"],
5  [“Author”, “book”, “<div class=’author’>#</div>”],
6  [ "_Reviewer", "split(Reviews, '<li>')", "#"],
7  [ 'Reviewer", "_Reviewer[0:0]", "#"],
8  [ 'ReviewerName, Rating, Text", "Reviewer",
   "*</b>#< b>*</b>#< b>*</b>#*"] ]
```

(а)

```
root complex {
    book_name string "Databases"
    reviews complex{
        ReviewerName string John

        Rating int 7
        Text string ...
    }
}
```

(б)

Рисунок 2.7 – Файл специфікації TSIMMIS (а) та OEM вихід (б).

Приклад шаблону:

1. <h1>#</h1>

Даний шаблон вибере вміст тега h1. Спеціальний символ # вказує, що саме буде збережено в змінну. Також в шаблонах може використовуватися спеціальний символ "*" який позначає відкидання. Потім TSIMMIS виводить дані в моделі обміну об'єктів (наприклад, малюнок 2.7(б)), який містить потрібні дані разом з інформацією про структуру та зміст результату. TSIMMIS забезпечує два важливих оператори: *split* та *case*. Оператор *split* використовується для поділу списку вхідного елемента на окремі елементи (масив підрядків, наприклад, рядок 5) за вказаним роздільником. Оператор *case* дозволяє користувачеві обробляти нерегулярні зв'язки в структурі вхідних сторінок.

Minerva намагається поєднувати переваги підходу декларативної граматики на основі з гнучкістю процедурального програмування при роботі з неоднорідностями та виключеннями [59]. Це робиться шляхом механізму явного включення процедур обробки вилучень всередині регулярної граматики. Процедури обробки вилучень записується в Мінерві, використовуючи спеціальну мову, що називається Editor. Граматика що використовується Мінервою визначаються в стилі EBNF, де визначається набір правил; кожний набір правил визначає структуру не-термінальних символів (передує ‘$’) граматики. Наприклад, на рисунку 7 показаний набір правил, які можуть бути використані для вилучення (також включення в базу даних), відповідні атрибути для певної задачі IE. Як завжди в EBNF позначеннях вираження [*p*] позначає додатковий шаблон *p*; Вираз *(p)** означає, що *p* може повторюватися нуль або більше разів. Нетермінальні продукції $bName, $rName, $rate, і $text безпосередньо випливають з їх використання у визначенні $Book. Таким чином, назві книги передує "Book Name" і потім "", як зазначає шаблон "*(?)", який перевіряє відповідність усього перед тегом . Остання продукція на рисунку 7 визначає спеціальний нетермінальний $TP (Tuple Production - кортеж продукцій), який використовується для вставки кортежу в базу даних

після парсингу кожної книги. Для кожного набору правил, можна додати обробник виключень, який містить частини редактора коду (Editor), який може обробляти нерегулярності знайдені в інтернет - даних. Всякий раз, коли зазнає невдачі парсинг цього набору правил, збуджується виключення і виконуються відповідний обробник виключень.

```
Page Book_Reviews
$Book_Reviews: <html><body> $Book </body></html>;
$Book: <b>Book Name </b> $bName <b> Reviews </b>
[<ol> ( <li><b> Reviewer Name </b> $rName <b>
    Rating </b>$rate <b> Text </b> $text $TP )* </ol>];
$bName: *(?<b>);
$rName: *(?<b>);
$rate: *(?<b>);
$text: *(?</li>);
$TP: {
       $bName, $rName
       $rate
       $text
END    }
```

Рисунок 2.8 – Граматика Minerva в стилі ENBF.

WebOQL це функціональна мова, який може бути використаний в якості мови запитів для Web, для слабоструктурованих даних та для реструктуризації веб-сайту [60]. Основна структура даних, що підтримується WebOQL, є *гіпердерево*. Гіпердерева є деревами упорядковані за поміченими дугами, які можуть бути використані для моделювання реляційної таблиці, файлу BibTeX, ієрархії каталогів і т.д. Рівень абстракції моделі даних підходить для підтримки колекцій, вкладеності і впорядкованості. На рисунку 8 показане гіпердерево для стор. p_{e1} наведеного прикладу. Як показано на рисунку, деревоподібна структура схожа на структуру DOM дерева, де дуги позначені записами з трьома атрибутами Tag, Source, Text, що відповідають імені тегу, частині HTML коду і тексту за винятком розмітки, відповідно. Основною конструкцією що забезпечується WebOQL є загальновідома обрати-з-куди (select-from-where). Мова має можливість моделювати всі операції в вкладеній реляційній алгебрі і обчислювати транзитивне замикання на довільному бінарному відношенні. В якості прикладу, наступний запит витягує імена рецензентів

«Jeff» та «Jane» зі сторінки p_{e2}, де лапки і знак оклику визначають перше піддерево і замикаюче дерево, відповідно. Змінні, в залежності від випадку, перебирають на простих деревах або на замикаючих деревах гіпердерева зазначеного після оператора «in».

```
Select [ Z!'.Text]
From x in browse ("pe2.html")', y in x', Z in y'
Where x.Tag = "ol" and Z.Text="Reviewer Name"
```

Крім управління даними з використанням гіпердерев, мова також може бути використана для Web реструктуризації, що робить результат запиту читаємим для інших додатків.

W4F (WYSIWYG Web обгорточний завод) є Java інструментаріэм для створення веб - обгорток [61]. Процес розробки обгортки складається з трьох незалежних шарів: *пошуковий*, *витяговий* і *відображувальний* шари. У пошуковому шарі, опрацьований документ що витягується (з Інтернету по протоколу HTTP), очищають, а потім подають в HTML - аналізатор, який будує дерево згідно об'єктної моделі документа (DOM). В шарі екстракції, правила вилучення застосовуються на дереві синтаксичного аналізу для видобування інформації, для подальшого їх зберігання у внутрішньому форматі W4F називається список вкладених строк (Nested String List, NSL). У *відображувальному* шарі, NSL структури експортуються в додатки верхнього рівня відповідно до правил відображення. Правила екстракції виражаються з допомогою HEL (HTML Extraction Language), який використовує дерево HTML розбору (тобто DOM дерево) шлях для доступу до даних, які потрібно встановити.

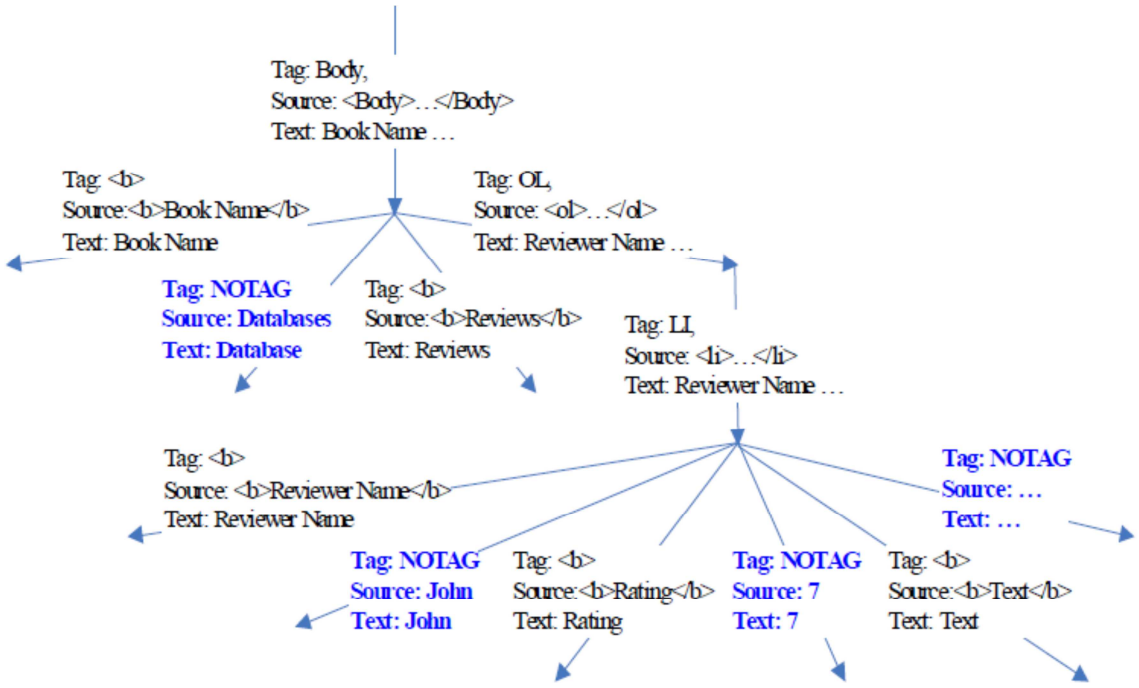


Рисунок 2.8 – WebOQL гіпердерево для сторінки pe1 на рисунку 2.4.

Наприклад, для адресації до імен рецензентів «Jeff» та «Jane» з $p_{e2,}$ ми можемо використовувати вираз <<html.body.ol[0].li[*].pcdata[0].txt>> де символ [*] може відповідати будь-якому числу (в даному випадку, 0 та 1). Мова також пропонує регулярні вирази і обмеження для адресації до більш дрібних часток даних. Наприклад, користувачі можуть використовувати регулярні вирази, щоб *співставити (match)* або *розділити (split)* (відповідно до синтаксису Perl) рядки, отриманого шляхом DOM дерева. Нарешті, оператор *вилка (fork)* дозволяє будувати вкладений список строк, слідуючи декількома множинними підшляхами одночасно. Для того, щоб допомогти користувачеві з адресацію DOM дерева шляху, інструментарій розроблений з WYSIWYG (що ви бачите, що ви отримуєте) підтримкою з допомогою смарт-майстрів.

XWrap являє собою систему, яка використовує інформацію про форматуванні веб-сторінок, щоб припустити, основну семантичну структуру сторінки [62], [63]. Він кодує гіпотетичну структуру і вилучені знання з веб-сторінок декларативною мовою ґрунтованою на правилах, розробленою спеціально для XWrap. Процес генерації обгортки включає в себе два етапи: *аналіз структури* та *генерація XML для конкретного джерела*. На першому етапі, XWrap витягує, очищає і створює

деревоподібну структуру сторінки. Потім система ідентифікує області, семантичні маркери, які представляють інтерес і корисні ієрархічні структури розділів сторінки, шляхом взаємодії з користувачами через об'єкт (запис) та стадії екстракції елементів. На другому етапі, система генерує XML - файл шаблону на основі маркерів контенту і специфікації вкладеності ієрархії, а потім створює генератор XML для конкретного джерела. У певному сенсі, XWRap може бути класифікований як контрольована система WI для якої не потрібно написання правил. Однак, це вимагає розуміння від користувачів дерева розбору HTML, визначення поділяючих тегів для рядків і стовпців в таблиці, і т.д. Таким чином, класифікуються як системи, що вимагають спеціальних знань користувачів. З іншого боку, тут не використовується ні один конкретний алгоритм навчання; правила видобутку в основному базуються на адресації шляху DOM-дерева.

Система XWrap на відміну від TSIMMIS яка використовувала для вилучення метод простого парсинга строк, дана система використовує метод аналізу DOM дерева. На першому кроці система отримує html документ, і розбирає його структуру в DOM дерево, далі дані витягуються шляхом вибору вузлів дерева.

Дана система має графічний інтерфейс, а створена конфігурація записується в файл конфігурації.

2.4.2. Контрольовані системи (WHISK, SRV)

Як показано в лівій нижній частині рисунку 2.4, контрольовані системи WI приймають безліч веб-сторінок, мічених з прикладами даних, які повинні бути вилучені і вихідних обгорток. Користувач надає початковий набір мічених прикладів і систему (з GUI) може запропонувати додаткові сторінки для користувача етикетки. Для таких систем, звичайні користувачі, взамін програмістів можуть бути навчені використовувати графічний інтерфейс маркування, таким чином зменшуючи вартість виробництва обгортки. Такі

системи включають SRV, RAPIER, WHISK, WIEN, STALKER, SoftMealy, NoDoSE and DEByE.

SRV є реляційним алгоритмом зверху вниз, що генерує однослотові правила вилучення [64], [65]. Це стосується IE як типу завдань класифікації. Вхідні документи токенізуються (лексемізуються) і всі підстроки безперервних токенів (лексем або маркерів, тобто фрагментів тексту) позначаються або як мета для екстракції (позитивні приклади) або ні (негативні приклади).

Правила, які генеруються SRV є логічними правилами, які покладаються на наборі лексем-орієнтованих ознак (або предикатів). Ці ознаки мають два основних різновиди: прості і реляційні. Проста ознака є функцією, яка відображає маркер в деяку дискретну величину, таку як довжина, тип символів (наприклад, числовий), орфографія (наприклад, заголовна літера) і частина мови (наприклад, дієслово). Також враховується ставлення однієї лексеми до іншої. Реляційна функція відображає маркер в інший маркер, наприклад, в контекстний (попередній або наступний) маркер вхідних маркерів. Алгоритм навчання протікає як FOIL (First Outer Inner Last), починає з усім набором прикладів і стрімко додає предикати, охоплюючи якомога більше позитивних прикладів та якомога менше негативних прикладів. Наприклад, щоб витягти рейтингову оцінку для наведеного прикладу, SRV повертатиме правило, як на рисунку 2.9 (a), в якому йдеться про те, рейтинг це одиничне числове слово і зустрічається в межах тегів HTML переліку.

RAPIER також фокусується на видобутку на місцевому рівні, але використовує алгоритм реляційного навчання знизу-вгору (на основі стиснення) [66], тобто він починається з самих конкретних правила, а потім замінює їх на більш загальні правила. RAPIER вивчає поодинокі шаблони вилучення слота, що роблять використання синтаксичної та семантичної інформації, в тому числі маркував по частині мови або лексикону (WordNet). Правила видобутку складаються з трьох різних моделей. Перша це дофільтраційний шаблон, який співставляє текст безпосередньо перед наповнювачем, друга це шаблон, який співставляє фактичний наповнювач

слоту, нарешті остання модель це пост-наповнювальний шаблон, який співставляє текст одразу ж після наповнювача. Як приклад, на рисунку 9 (б) показано правило вилучення назви книги, яке безпосередньо передує словам «Book», «Name» та «», за якими одразу ж слідує слово «». «Наповнювач шаблону» вказує, що заголовок складається з не більше як двох слів, які були позначені як "nn" або "nns" за допомогою маркувальника POS (тобто один або два одиничних чи множинні загальні іменники).

Rating extraction rule-	**BookTitle extraction rule-**
length (=1), every (numeric true), every (in_list true).	Pre-filler pattern Filler pattern Post-filler pattern (1) word: Book list: len: 2 word: (2) word: Name Tag: [nn, nns] (3) word:
(a)	(b)

Рисунок 2.9 – Правила видобутку SRV (а) та RAPIER (б).

WIEN: Кашмерік ідентифікував сімейство з шести класів-обгорток: LR, HLRT, OCLR, HOCLRT, N-LR і N-HLRT для напівструктурованого вилучення веб-даних [67]. WIEN фокусується на витяжних архітектурах. Перші чотири обгортки використовуються для напівструктурованих документів, а решта дві обгортки використовуються для ієрархічно вкладених документів. Обгортка LR - вектор $2K$ роздільників для сайту, що містить K атрибутів. Наприклад, вектор ('Reviewer name ', '', 'Rating ', '', 'Text ', '
') може бути використаний для витягнення 3-слотів оглядів книг для нашого прикладу. Клас HLRT використовує два додаткових роздільники, щоб пропустити потенційно заплутаний текст у голові або хвості сторінки. Клас OCLR використовує два додаткових роздільники, щоб ідентифікувати весь кортеж в документі, а потім використовує стратегію LR для вилучення кожного атрибута по черзі. HOCLRT обгортка поєднує в собі два класи OCLR і HLRT. Дві обгортки N-LR і N-HLRT є продовженням LR і HLRT і призначені спеціально для вкладеного вилучення даних. Слід зазначити, що оскільки WIEN

передбачає упорядкованість атрибутів у запису даних, відсутність атрибутів та їх перестановка не можуть бути оброблені.

WHISK [68] використовує супроводжувальний алгоритм навчання для генерації правил вилучення множинних слотів для широкого спектра документів, починаючи від структурованого закінчуючи довільним текстом [69]. При використанні на довільному тексті, WHISK найкраще працює з вхідними даними, що анотовані за допомогою синтаксичного аналізатора і семантичного маркувальника. Правила WHISK засновані на формі шаблонів регулярних виразів, які визначають контекст відповідних фраз і точних роздільників цих фраз. Він приймає набір помічених вручну навчальних примірників, щоб направляти створення правил і протестувати продуктивність запропонованих правил.

WHISK визиває правила зверху-вниз, починаючи з самого загального правила, яке охоплює всі екземпляри, а потім розширює правило додаванням компонентів по одному за раз.

Для неструктурованих документів які є вільним текстом, вхідні дані обробляються синтаксичним аналізатором який розмічає вхідний текст тегами.

Для створення 3-слотних оглядів книги, він починає з порожнього правила «* (*) * (*) * (*) *», де кожна дужка вказує на фразу, що має бути вилучена. Фраза в межах першого набору дужок пов'язана з першою змінною $1, а другого з $2, і надалі. Таким чином, правило на рисунку 2.10 може бути використано для витягу прикладу 3-слотних оглядів книги. Якщо частина вхідних даних залишається після того, як правило вдалося, правило повторно застосовуватися до решти вхідних даних. Таким чином, логіка екстракції схожа на логіку LR обгортки WIEN.

Pattern:: * ‘Reviewer Name ’ (*Person*) ‘’ * (*Digit*) ‘Text’(*) ‘’
Output:: BookReview {Name $1} {Rating $2} {Comment $3}

Рисунок 2.10 – Правило екстракції WHISK.

Нижче наведено приклад тексту який пройшов процедуру синтаксичного аналізу [70].

Input text:
C. Vincent Protho, chairman and chief executive officer of this maker of semiconductors, was named to the additional post of president, succeeding John W. Smith, who resigned to pursue other interests.

Succession event
PersonIn: C. Vincent Protho
PersonOut: John W. Smith
Post: president

Область послідовного управління: вхідне речення та кадр вихідного випадку.

```
@S[
  {SUBJ   @PN[ C. Vincent Protho ]PN , @PS[chairman and chief
          executive officer ]PS of this maker of semiconductors, }
  {VB     @Passive was named @nam }
  {PP     to the additional post of @PS[ president ]PS , }
  {REL_V  succeeding @succeed @PN[ John W. Smith ]PN ,
          who resigned @resign to pursue @pursu other interests. }
]@S 8910130051-1
```

Синтаксично проаналізоване вхідне речення.

Рисунок 2.11 Результат роботи синтаксичного аналізатора

Далі аналізуючи контекст фраз та їх точні роздільники формуються регулярні вирази. Для створення правил використовується алгоритм навчання зверху-вниз, спочатку генерується загальне правило, яке охоплює всі випадки, а потім правила поширюються шляхом одночасного додавання слів. Наприклад, для отримання інформації про книгу з такою вихідної рядки:

```
1.  <h1>Перетворення</h1><b>F.Kafka</b>
```

Буде сформовано наступне загальне регулярний вираз:

```
1.  *(*)*(*)*
```

Дужки в регулярних виразах позначають запис обраної інформації в змінну. Далі в результат роботи алгоритму дане правило набуде вигляду:

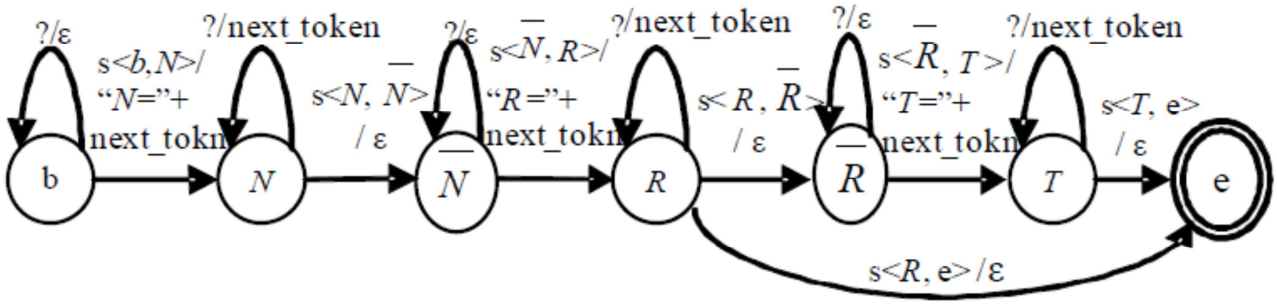
```
1.  \s*<h1>(.*)</h1>\s*<b>(.*)</b>\s*
```

NoDoSE: В протилежність WIEN, де приклади навчання отримуються з декількох оракулів, що можуть ідентифікувати цікаві типи полів в документі, NoDoSE забезпечує інтерактивний інструмент для користувачів ієрархічної декомпозиції слабоструктурованих документів (включаючи звичайний текст або HTML-сторінки). Таким чином, NoDoSE здатний обробляти вкладені об'єкти. Система намагається визначити формат / граматику вхідних документів двома евристичними компонентами видобутку: один видобуває текстові файли,

а інший розбирає HTML-код (компоненти майнінгу та парсингу). Подібно WIEN, алгоритми видобутку намагаються знайти спільні префікси та суфікси як роздільники для різних атрибутів. Незважаючи на те, що він не бере на себе порядок атрибутів в запису, що має бути фіксованим, він прагне знайти абсолютно послідовний порядок для різних атрибутів в запису. Результатом цього завдання є дерево, яке описує структуру документа. Наприклад, для створення обгортки для наведеного прикладу користувач може взаємодіяти з графічним інтерфейсом користувача NoDoSE для декомпозиції документа у вигляді запису з двома полями: заголовок книги (атрибут типу строка) та список рецензента, який в свою чергу є записом трьох полів RName (строка), Rate (ціле число) та Text (строка). Далі, NoDoSE потім автоматично обробляє їх та формує правила видобутку.

SoftMealy: Для того, щоб обробляти відсутні атрибути та їх перестановки на вході, Хсу та Данг представли ідею перетворювача кінцевих станів (finite-state transducer - FST) для розширення варіацій структур вилучення [71]. FST складається з двох різних частин: перетворювача тіла, який витягує частину сторінки, що містить кортежі (по аналогії з HLRT в Wien), і кортеж перетворювача, який ітеративно витягує кортежі з тіла сторінки. Кортеж перетворювача приймає кортеж і повертає його атрибути. Кожен окремий атрибут перестановки на сторінці може бути закодований в якості успішного шляху від початкового стану до кінцевого стану кортежу перетворювача; переходи між станами визначаються шляхом зіставлення контекстних правил, що описують контекстне розмежування двох суміжних атрибутів. Контекстні правила складаються з окремих роздільників, які представляють собою невидимі кордони між суміжними маркерами, а індуктивний алгоритм узагальнення використовується щоб отримати ці правила з прикладів навчання. На рисунку 2.12 показаний приклад FST, який може бути використаний для отримання атрибутів оглядів книги: ім'я рецензента (N), рейтинг (R), а також коментар (T). На додаток до початкового та кінцевого станів, за кожним атрибутом A, слідує фіктивної стан A. Кожна дуга позначена контекстуальним правилом, що дозволяє перехід та маркерами на

вихід. Наприклад, коли стан переходу доходить до R стану, перетворювач буде видобувати (extract_the) атрибут R, поки він відповідатиме контекстним правилам (_the contextual _rules) S<R, R > (який складається з S<R, R > L та s<R, R > R). Стан R та кінцевий стан пов'язані, якщо ми припустимо що коментар може і не існувати.



s<N, R > :: = HTML() ClAlph(Rating) HTML()
s<N, R>^{R} ::= Spc(-) Num(-)
s<R R >^{L} :: = Num(-)
s<R R >^{R} :: = NL(-)HTML()

Рисунок 2.12 – FST для веб-сторінок у поточному прикладі.

STALKER являє собою систему, яка виконує WI вилучення ієрархічних даних [72]. Він вводить поняття формалізму вбудованого каталогу (embedded catalog, EC) для опису структури широкого спектра напівструктурованих документів. EC опис сторінки є деревоподібною структурою, в якій листя є атрибутами, що будуть витягнуті, та внутрішні вузли є списками кортежів. Для кожного вузла в дереві, обгортка вимагає правило витягу цього вузла з його батьківського вузла. Крім того, для кожного вузла списку, обгортка вимагає правила ітерації списку, яке розкладає список в окремі кортежі. Таким чином, STALKER перетворює складну проблему вилучення даних з довільного комплексного документу в ряд більш простих завдань вилучення з більш високого рівня на більш низький рівень. Крім того, екстрактор використовує багатоходове сканування для обробки відсутніх атрибутів та множинних перестановок. Правила екстракції генеруються використанням алгоритму послідовного покриття, який починається від лінійно орієнтованого автомату, для охоплення якомога більшої кількості позитивних прикладів, а потім намагається генерувати нові автомати для інших прикладів. Дерево STALKER

ЕС, що описує структуру даних під керуванням, показано на рисунку 2.13(а). Деякі з правил вилучення показані на рисунку 2.13(б). Наприклад, рейтинги рецензента може бути вилучені за допомогою першого застосування правила екстракції List(Reviewer) (що починається з "
") та закінчується "
")) до всього документа, а потім правило Рейтингу видобутку до кожного окремого оглядача (Reviewer), яке формується шляхом застосування правила ітерацій для List(Reviewer). У певному сенсі, STALKER еквівалентний багатопрохідному Softmealy [73]. Однак шаблони для вилучення кожного атрибута можуть бути послідовними на відміну від безперервних шаблонів, які використовуються Softmealy.

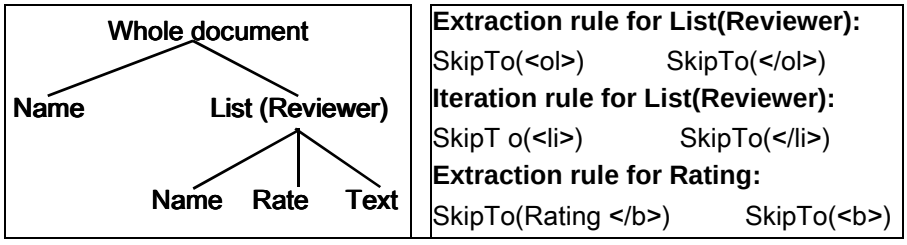


Рисунок 2.13 – Дерево ЕС (а), та правила витягу STALKER (б).

DEByE (вилучення даних за прикладом - Data Extraction By Example): як і NoDoSE, DEByE забезпечує інтерактивний графічний інтерфейс користувача для генерації обгортки [74], [75]. Різниця полягає в тому, що в DEByE користувач зазначає тільки атомарні значення (атрибут), щоб зібрати вкладені таблиці, в той час як вузловий користувач розкладає весь документ зверху вниз. Крім того, DEByE приймає стратегію вилучення знизу-вгору, яка відрізняється від інших підходів. Головна особливість цієї стратегії є те, що він витягує атомарні компоненти, а потім збирає їх в (вкладені) об'єкти. Правила екстракції, звані шаблони пари атрибут-значення (AVPS), для атомних компонентів ідентифікуються контекстним аналізом: починаючи з довжини контексту 1, якщо число збігів перевищує розрахункову кількість входжень, наданих користувачем, воно додає зразку додаткові умови до числа збігів менше розрахункової одиниці. Наприклад, DEByE генерує AVP шаблони,

“Ім'я</ b>* Відгуки”, “Ім'я*Рейтинг”, “Оцінка*Текст” і “ * ” для назви книги, рецензент назва, рейтинг і коментар відповідно (* позначає дані, які повинні бути вилучені). Отримані AVPs потім використовуються для складання малюнка вилучення об'єкта (OEPs). OEPs дерева, що містять інформацію про структуру документа. Субдерева в OEP самі по собі є OEPs, моделювання структури складових об'єктів. У нижній частині ієрархії лежить AVP, які використовуються, щоб ідентифікувати атомні компоненти. Зібрати атомарних значень в списки або кортежі заснований на припущенні, що різні входження об'єктів не перекривають один одного. Для неоднорідних об'єктів, користувач може вказати більше одного прикладу об'єкта, таким чином, створюючи різні OEP для кожного прикладу.

2.4.3 Напівкеровані системи

Напівкеровані системи базуються на ідеї виділення повторюваних конструкцій в тілі документа. Наприклад сторінка на якій міститься список книг, кожен елемент списку має однакову структуру html тегів.

Такі системи включають IEPAD, OLERA і Thresher. На відміну від контрольованого підходу, в OLERA і Thresher прийнятий грубий (замість повного і точного) приклад від користувачів для вироблення правил вилучення, тому вони й називаються напівкерованими. IEPAD, хоча й вимагає навчальних сторінок без маркування, з боку користувача вимагаються післязусилля, для обрання шаблону мети, а також зазначення даних, які будуть вилучені. Всі ці системи призначені для вирішення завдань вилучення на рівні записів. Оскільки ніяких цілей видобутку не визначені для таких систем, для користувачів необхідний графічний інтерфейс, для зазначення мети отримання після фази навчання. Таким чином, вимагається контроль користувачів.

IEPAD є однією з перших систем IE, що узагальнює шаблони вилучення з немічених веб-сторінок [76]. Система IEPAD [77] використовує структуру даних - суфіксне дерево (PAT), для виявлення повторюваних шаблонів на сторінці. Суфіксне дерево (бор, англ. trie) – структура даних, що містить всі

суфікси наданого тексту в якості ключів, та їх позиції у тексті як їх значення [78]. Суфіксні дерева дозволяють особливо швидкі реалізації багатьох важливих строкових операцій.

Цей метод використовує той факт, що якщо веб-сторінка містить численні (однорідні) записи даних для витягнення, вони часто регулярно обробляються використовуючи той же шаблон для гарної візуалізації. Таким чином, повторювані шаблони можуть бути виявлені, якщо сторінки добре закодовані. Таким чином, навчальні обгортки можуть бути вирішені шляхом виявлення повторюваних шаблонів. IEPAD використовує структуру даних, що зветься PAT дерева, яка є бінарним суфіксним деревом, щоб виявити повторювані шаблони на веб-сторінці. Оскільки така структура даних записує тільки точну відповідність для суфіксів, в IEPAD додатково застосовується алгоритм центральної зірки для вирівнювання декількох строк, які починаються з кожного входження повторення та кінця до початку наступної появи. Нарешті, щоб охопити всі записи даних використовується представлення підпису для позначення шаблону.

При побудові PAT дерева вміст тега як звичайний текст ігнорується та замінюється спеціальним токеном. Розглянемо наступний HTML код:

```
1.    <b>Congo</b><i>242</i><br>
2.    <b>Egypt</b><i>20</i><br>
```

Розібравши ці строки на токени отримаємо:

```
1.html(<b>)text(_)html(</b>)html(<i>)text(_)html(</i>)html(<br>)
html(<b>)text(_)html(</b>)html(<i>)text(_)html(</i>)html(<br>)
```

Далі кожен токен кодується числом і виконується побудова суфіксного дерева. На основі суфіксного дерева обираються повторювані шаблони. Однак у реальних сторінках може бути безліч помилкових повторень які не несуть у собі корисну інформацію, тому кожне повторення проходить процедуру валідації. У IEPAD використовується фільтрація по частоті повторень. Далі відфільтровані повторення проходять процедуру узагальнення і апроксимації і виконується формування правил витягів. Правила вилучення це набір регулярних виразів.

Для нашого загального прикладу тільки сторінка pe2 можна бути використана в якості вхідних даних для IEPAD. Кодуванням кожного тега в якості окремих маркерів та будь-якого тексту між двома сусідніми тегами спеціальним токеном "T", IEPAD винаходить шаблон “
TTTT TT” з двома входженнями. Потім користувач повинен вказати, наприклад, другий, четвертий та шостий токен «T», як відповідні дані (позначаючи ім'я рецензента, рейтинг та коментар відповідно).

Thresher [79] також напівкерований підхід, який схожий на OLERA. Графічний інтерфейс для Thresher вбудований в браузер Haystack, який дозволяє користувачам вказати приклади змісту, виділивши їх і описати їх значення (помаркувати їх). Тим не менш, він використовує відстань редагування дерева (замість строки редагування відстані, як в OLERA) між піддеревами DOM цих прикладів, щоб створити обгортку. Потім вона дозволяє користувачеві зв'язати класи семантичної веб-мови RDF (Resource Description Framework) та предикати до вузлів цих обгорток.

2.4.4. Не керовані системи видобутку інформації

Як показано на лівій верхній частині Рисунок2.6., неконтрольовані системи IE не використовують ніяких мічених навчальних прикладів і не мають ніяких користувальницьких взаємодій, щоб згенерувати обгортку. Неконтрольовані системи IE RoadRunner і EXALG, призначені для вирішення завдання вилучення на рівні сторінок, в той час як Dela і DEPTA призначені для завдання вилучення на рівні записів. На відміну від контрольованих систем IE, де мети одержання вказані користувачами, цільова екстракції визначаються як дані, який використовується для генерації сторінки або без тегів текстів в областях даних багатих на сторінці введення. У деяких випадках кілька схем можуть відповідати навчальним сторінок завдяки наявності обнуляє атрибутів даних, що призводить до неоднозначності [80]. Вибір визначення правильної схеми залишається для користувачів. Точно

так же, якщо потрібно не всі дані, після обробки може знадобитися для користувача, щоб вибрати відповідні дані і дати кожній частині даних власна назва.

Dela: В якості розширення IEPAD, Dela [81], [82] видаляє взаємодію користувачів в екстракційних правилах узагальнення і угод з витяганням вкладеного об'єкта. Процес генерації обгортки в Dela працює в два послідовних кроки.

По-перше, алгоритм *витягу розділів насичених даними* (Data-rich Section Extraction, DSE) призначений для вилучення розділів насичених даними з веб-сторінок шляхом порівняння DOM дерев для двох веб-сторінок (з того ж веб-сайту), і відкидання вузлів з однаковими підрозділами дерев.

По-друге, використовується *шаблон екстрактору*, щоб виявити безперервно повторювані шаблони з використанням суфіксних дерев. Збереженням останнього входження для кожного виявленого шаблону, він ітеративно виявляє нові повторювані шаблони з нової послідовності, утворюючи вкладену структуру.

Наприклад, якщо послідовність рядків "<P><A>T</A><A>T</A>T</P> <P><A>T</A>T</P>", Dela виявить "<P><A>T</A>T<P>" від безпосередньої послідовності "<P><A>T</A>T</P><P><A>T</A>T</P>" та поверне шаблон в дужках "(<P>(<A>T</A>)*T<P>)*", щоб позначити вкладену структуру. Оскільки виявлений шаблон може перетнути межи об'єкта даних, Dela опитує K сторінок та обирає одну з найбільшою підтримкою сторінок. Знову ж, кожне входження регулярного вираження являє собою один об'єкт даних. Об'єкти даних потім перетворюються в реляційну таблицю, де багато значень одного атрибута розподіляються на кілька рядків таблиці. Нарешті, мітки присвоюються стовпцям таблиці даних за чотирма евристиками, в тому числі ярлики елементів в формі пошуку або таблиці сторінки та максимальний префікс і максимальний суфікс розподіляється між всіма клітинами колонки.

RoadRunner розглядає процес створення сайту, як кодування вихідного вмісту бази даних в строки HTML коду [83]. Як наслідок, вилучення даних розглядається як процес декодування. Тому, створюючи обгортку для набору

HTML-сторінок відповідає виведенню граматики для HTML коду. Система RoadRunner [84] заснована на технології зіставлення ACME (Align, Collapse under Mismatch, and Extract). Алгоритм зіставлення працює на двох об'єктах (приклад і аналізований текст). Алгоритм аналізує 2 html сторінки одного й того ж класу та створює обгортку на основі їх подібності та розбіжності. Система починає з порівняння двох сторінок, використовуючи техніку ACME для вирівнювання маркерів що збіглися і згортання для маркерів що не збіглися. Розбіжності можуть бути двох видів: *розбіжності строк*, що використовуються для виявлення ознак (#PCDATA) та *розбіжності тегів*, що використовуються для виявлення ітераторів (+) та додаткових матеріалів (?). Розбіжність строк безпосередньо пов'язана з даними що вилучаються. Розбіжності тегів використовується для пошуку ітераторів регулярних виразів.

На Рисунок 2.14 показаний приклад зіставлення для перших двох сторінок прикладу та генерування його обгортки. Оскільки може бути декілька вирівнювань, RoadRunner приймає UFRE (union-free regular expression – непов’язаних регулярних виразів), щоб зменшити складність. Результат вирівнювання перших двох сторінок, надалі порівнюється з третьою сторінкою в класі сторінок. У доповнення до модуля для дедукції шаблону, RoadRunner також включає в себе два модулі, класифікатор і *роздавач етикеток* для полегшення створення обгортки. Перший модуль, *Класифікатор*, аналізує сторінки та збирає їх в кластери з однорідною структурою, тобто сторінки з однаковим шаблоном згруповуються разом. Другий модуль, *роздавач етикеток*, встановлює імена атрибутів для кожного класу сторінки.

Наприклад, на сторінці книги може міститися фотографія обкладинки книги, а для іншої книги фотографії немає, отже тег img може бути відсутнім, таким чином генерує правило:

```
1.  (<img src=.../>)?
```

Символ ? у регулярних виразах означає нуль або одне входження. Для пошуку повторюваних шаблонів використовується ознака першої невідповідності термінального тегу.

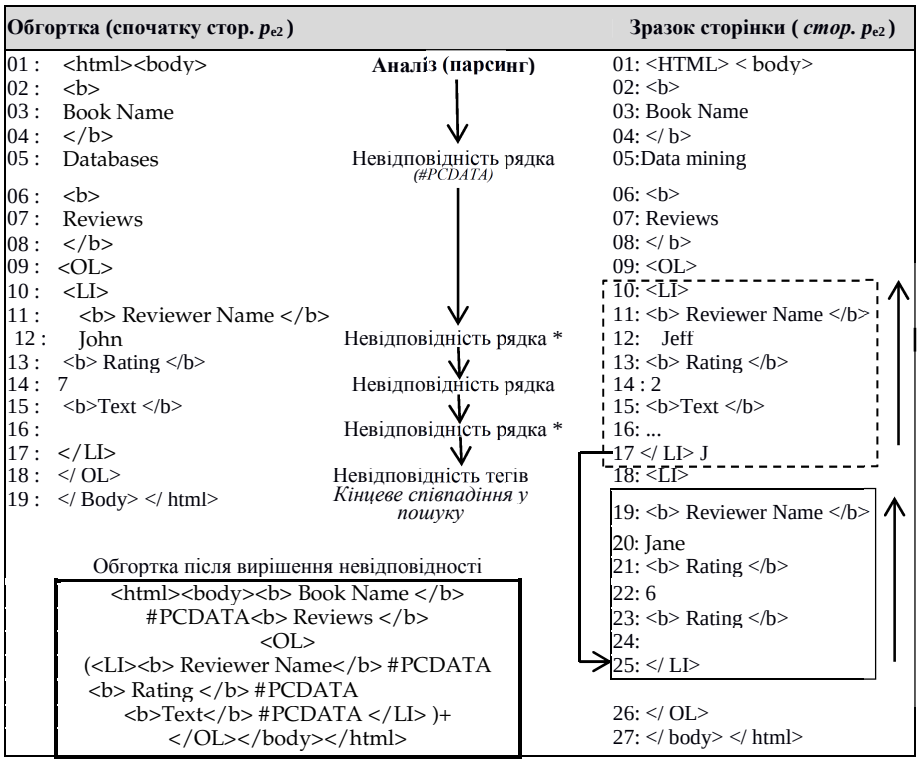


Рисунок 2.14 – Зіставлення перших двох сторінок наведеного прикладу

EXALG: Арасу та Моліна представили ефективне формулювання завдання отримання даних з веб-сторінок [85]. Вхід EXALG являє собою набір сторінок, створених з невідомого шаблону *T* та значення що мають бути вилучені. EXALG виводить шаблон *T* та використовує його для витягу набору значень з закодованих сторінок на виході.

EXALG встановлює невідомий шаблон за допомогою двох методів *диференційних ролей* та *класів еквівалентності* (EC). У першому методі входження з двома різними шляхами конкретних маркерів мають різні ролі. Наприклад, в поточному прикладі, роль «Name», коли зустрічається в «Book Name» (тобто, $Name_5$) відрізняється від її ролі, коли зустрічається в «Reviewer Name» (тобто, $Name_{14}$). В останньому методі клас еквівалентності є максимальним набором лексем, що мають однакові частоти входження на навчальних сторінках (*вектор поширеності*). Так, наприклад, на Рисунок 2.4, два маркери <html>$_1$ і <body>$_2$ мають той же вектор поширеності (<1, 1, 1, 1>), тому вони належать до одного класу еквівалентності. Розуміння в тому, що

маркери шаблону, що охоплюють кортеж даних, мають один і той же вектор виникнення і утворюють клас еквівалентності. Однак, щоб уникнути випадкового утворення класів еквівалентності з маркерів даних, вбудовані каталоги (EC) з недоліками підтримки (кількість сторінок, що містять маркер) та розміром (кількість маркерів в EC) відфільтровують. Крім того, для відповідності ієрархічній структурі схеми даних, класи еквівалентності мають бути взаємно вкладеними, а маркери в EC повинні бути впорядковані. Ці придатні вбудовані каталоги потім використовуються для побудови оригінального шаблону.

Система **DEPTA** отримує дані на основі часткового дерева вирівнювання. Як IEPAD та Dela, DEPTA може бути застосовна тільки до веб-сторінок, що містять два або більше записів даних в області даних. Тим не менше, замість виявлення повторення підстроки на основі дерев суфіксів, який порівнює все суфікси строк HTML-тегів (як кодований маркер строки, описаний в IEPAD), він порівнює тільки сусідні підстроки з початкової мітки, які мають один і той же батьківський вузол в дереві HTML-тегів (схожий на дерево HTML DOM, але враховуються тільки теги). Розуміння в тому, що записи даних тієї ж області даних відображаються в дереві тегів веб-сторінки при тому ж батьківському вузлі. Таким чином, підстроки що не відносяться до справи не потрібно порівнювати одну з однією, як в підходах на основі суфіксів. Крім того, порівняння підстроки може бути обчислено з допомогою відстані редагування строки замість точного збігу строки при використанні суфіксних дерев, де виявляються тільки повністю аналогічні підстроки.

Описаний алгоритм, званий MDR [86], працює в три етапи. По-перше, він будує дерево HTML тегів для веб-сторінки, як показано на рисунку 2.15, де текстові строки ігноруються. Однаковими областями вважаються ті області для яких:

- всі вузли області мають загальний батьківський вузол;
- всі вузли мають однакову довжину;
- всі вузли суміжні;

– при порівняння строк відстань Левенштейна нижче порогової.

Алгоритм проходить всі вузли дерева, для пар дочірніх елементів виконується розрахунок відстані Левенштейна, якщо схожість більше граничної то вважається що вузли схожі. Потім за допомогою часткового дерева вирівнювання [87], виконується вилучення даних. Варто зауважити той факт що обидві розробки приймають за дані лексеми без тегів, тобто вони не аналізують довільний текст всередині тега.

По-друге, він порівнює підстроки для всіх дочірніх вузлів з одного батьківського вузла. Наприклад, нам потрібно зробити два строкових порівняння, (b_1, b_2) та (b_2, ol), під батьківським вузлом <body>, де вузол строки тегу <ol> представлений як "
<b><b><b>
<b><b><b>". Якщо схожість більша, ніж попередньо визначене порогове значення (як показано на затінених вузлах на рисунку 2.15), вузли реєструються як область даних.

Третій крок призначений для обробки ситуацій, коли запис даних не відображається безперервно, як передбачається, в попередніх роботах. Нарешті розпізнавання елементів даних або атрибутів в записі виконується шляхом часткового дерева вирівнювання [88]. Дерево вирівнювання краще, ніж вирівнювання строк оскільки структура дерева, таким чином, зменшує число можливих вирівнювань. Алгоритм першим обирає запис дерева з найбільшим числом елементів даних, як центр, а потім зіставляє інші записи дерев з центром дерева. Однак DEPTA тільки додає вузли тегів до центрального дерева, коли позиція вузлів тегів може бути однозначно визначена в центральному дереві. Для решти вузлів, вони обробляються в наступній ітерації після того, як будуть оброблені всі теги дерева.

Зверніть увагу, що DEPTA передбачає, що лексема без тегів є елементами даних, призначених для вилучення, таким чином, він витягує не тільки ім'я рецензента, рейтинг і коментарі, а також ярлики "Reviewer Name", "Rating", та "Text" на стор p_{e1} в прикладі. Крім того, DEPTA має обмеження з обробки вкладених записів даних. Тому новий алгоритм NET, розроблений для обробки таких записів даних за допомогою виконання післявпорядкувального обходу

візуалізованого дерева тегів веб-сторінки та зіставлення піддерев в процесі що використовує метод відстані редагування дерева та візуальні підказки [89].

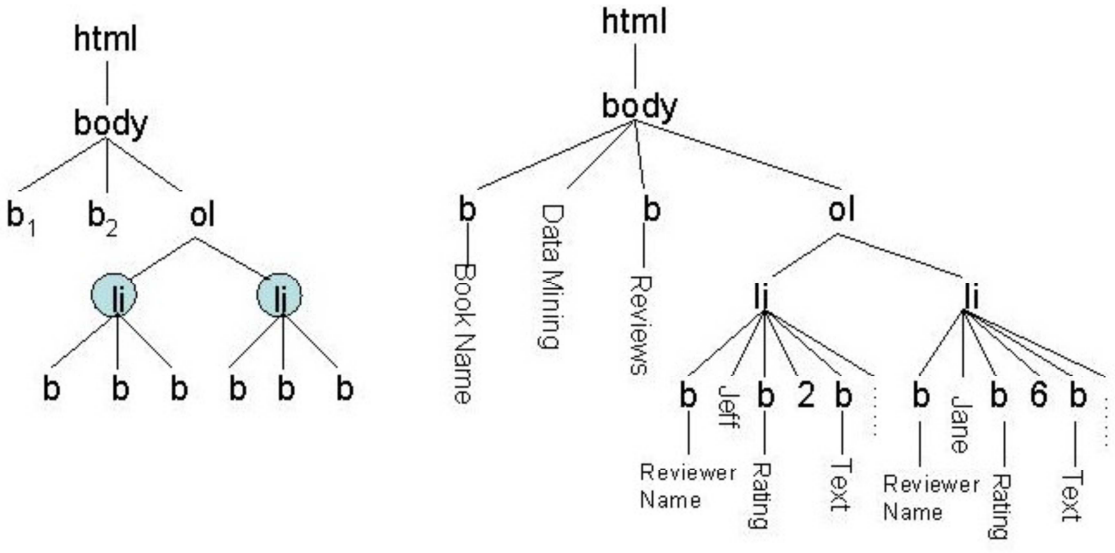


Рисунок 2.15 – Дерево тегів (ліворуч) і дерево DOM (як порівняння) для стор. p_{e2} з рисунку 2.4

В неконтрольованих підходах індукції обгорток (WI), важливим питанням є диференціювання ролей кожного маркера: або маркер даних або маркера шаблону. Для спрощення питання дехто припускає, що кожен HTML тег генеруються шаблоном, а інші маркери є елементами даних (як в Dela і DEPTA). Однак припущення не виконується для багатьох колекцій сторінок (тому, IEPAD і OLERA просто залишають це питання розрізнення між даними і маркерами шаблонів на розсуд користувачів). RoadRunner також передбачає, що кожен HTML тег генерується шаблоном, але інші зіставні строкові лексеми також розглядаються як частина шаблону. Для порівняння, EXALG має найбільш деталізований метод лексичного аналізу тексту, при цьому й більш гнучке припущення що кожен маркер може бути маркером шаблону, якщо є достатня кількість маркерів, щоб сформувати клас еквівалентності що часто зустрічається.

З іншого боку, DEPTA проводить процес видобутку з окремих веб-сторінок, в той час як RoadRunner і EXALG роблять аналіз з численних веб-сторінок (у той час як DeLa має переваги у численності вхідних сторінок для витягу секцій насичених даними, а також в узагальненій побудові шаблону, він

виявляє постійно повторювані шаблони з окремих веб-сторінок). Наступнім з нашої точки зору, є ключовий момент, що використовується для диференціації ролі кожного маркеру. Таким чином, кілька сторінок одного й того ж класу також використовуються для виявлення секцій насичених даними (як в Dela) або для усунення гучної інформації (як і в [90]). Тим часом, адаптація відповідності дерева в DEPTA (а також в Thresher) також забезпечує кращий результат, ніж методи узгодження строк, які використовуються в IEPAD та Roadrunner. Так само EXALG не здійснює повного використання деревоподібної структури, хоча інформація про шляхи дерева DOM використовується для диференціації маркерів ролі. Нарешті, оскільки вилучення інформації є лише частиною програми-обгортки або системи інформаційної інтеграції, залишається обробити додаткові завдання такі як скачування сторінок, призначення міток та співставлення з іншими джерелами веб-даних.

ViNTs [91] являє собою систему генерації обгорток на рівні записів, який використовує візуальну інформацію, щоб знайти роздільники між областями даних з результатів пошуку сторінок. Проте, алгоритм може бути застосований лише до сторінок, які містять, принаймні, чотири записи даних. Існує інший пов'язаний підхід, який був застосований на веб-сайтах для видобування інформації з таблиць [92]. Ця технологія заснована на використанні додаткових посилань на сторінки з інформацією, що містять додаткову інформацію про таблицю. Паралельно із зусиллями щодо виявлення Web-таблиць, інші дослідники працювали у виявленні таблиць серед текстових документів (наприклад, державні статистичні звіти) і сегментуванні їх в записи [93]. Оскільки ці підходи не вирішують проблему відрізнення маркера даних від маркера шаблону, методи розглядаються в якості напівкерованих підходів.

2.5 Порівняльний аналіз підходів видобутку інформації

Розглянуті вище системи реалізують різні підходи до вилучення інформації з веб сторінок. Кожна з розглянутих систем має як свої переваги так

і недоліки. Вибір методу в першу чергу залежить від цілей конкретного завдання. Для узагальненого порівняльного аналізу необхідно виділити критерії в рамках яких буде проведено порівняння. Першим критерієм можна виділити **рівень аналізу даних**, найвищий рівень - аналіз DOM дерева або іншої структури документа, найнижчий рівень аналізу - вільний текст. Розглянуті системи такі як TSIMMIS, WHISK, SRV можуть аналізувати вільний текст в тій чи іншій формі, а системи DEPTA, IEPAD вважають вільний текст за кінцеві дані. У цілому можна виділити те що системи з ручним та контрольованим способом формування правил більшою мірою дозволяють створювати правила для отримання даних з тексту. У неконтрольованої системі RoadRunner виконується порівняння рядків що дозволяє вибирати дані з тексту з однотипною структурою.

Наступний критерій який можна виділити це **рівень автоматизації**. Під рівнем автоматизації будемо вважати обсяг роботи який необхідно виконати користувачу системи. Розглянуті системи TSIMMIS і XWRAP мають низький рівень автоматизації, тому що користувач повинен вручну сформувати правила вилучення даних для кожної сторінки. Для формування таких правил користувач повинен володіти базовими знаннями про структуру веб документів, частково дана проблема вирішується впровадженням редактора з графічним інтерфейсом. Системи WHISK, SRV володіють середнім рівнем автоматизації тому використовую методи машинного навчання. В даному випадку від користувача як правило потрібно надати системі набір навчальних матеріалів. Системи IEPAD, RoadRunner, DEPTA маю досить високий рівень автоматизації, вони не вимагають від користувача навчальних матеріалів або формування правил, однак можуть зажадати пост обробки витягнутих даних. Оскільки алгоритми даних систем працюють на ідеї виявлення повторюваних структур, то необхідна достатня кількість записів або сторінок з вмістом.

Ще одним критерієм можна виділити використовувані **методи аналізу та вилучення даних**. Система TSIMMIS використовує власні шаблони для опису правил витягів, тому даний метод можна віднести до символьного аналізу. У таких системах як XWrap, RoadRunner, IEPAD, DEPTA використовують методи

аналізу DOM дерева, але кінцеві правила для отримання даних в системах RoadRunner, IEPAD, DEPTA представлені регулярними виразами. У системі SRV використовується набір правил заснованих на властивостях токенів, довжина, тип символу, частина мови і.т.д. Система WHISK використовує синтаксичний аналіз для побудови правил на основі регулярних виразів. Використання регулярних виразів і методів синтаксичного аналізу дозволяють краще вибирати інформацію з вільного тексту. Аналіз DOM дерева спрощує аналіз великих структурованих веб документів.

2.5.1 Порівняння систем IE на основі сфери завдань

Можливості обстежених систем IE для підтримки різних завдань IE, показані у таблиці 2.1. Особливості в цьому вимірі, включають варіацію вхідних ресурсів, такі як тип сторінок, підтримка не HTML, та варіацію вихідних даних, такі як рівень витягу, варіація атрибутів та варіація шаблону.

Тип сторінки: веб - сторінки відповідно до рівня структуризації можуть бути *структуровані*, *напів-структуровані* або *з вільним текстом*. Наприклад, ручні або контрольовані системи IE призначені для вилучення інформації зі сторінок крос-сайту (наприклад, дані професорів з різних університетів), а частково під наглядом і контролем системи IE призначені в основному для отримання даних з глибокої Web (сторінки шаблону). Таким чином, останні системи в значній мірі залежать від загального шаблону, який використовується для створення веб - сторінок, в той час як колишній включили більше можливостей маркерів (наприклад, кількість символів, частка великих літер і т.д.) для стимулювання видобутку правила. Об'єднавши більш характеристики шаблонів сторінок, неконтрольовані системи IE представляють автоматизації високого ступеня для вилучення правила узагальнення; на відміну від цього, розширення сторінок без шаблонів є досить обмеженим.

Підтримка не-HTML (NHS): підтримка не-HTML входів залежить від знання характеристик або серверної частини, які використовуються в системах IE. Більшість контрольованих системи можуть підтримувати не HTML

документи шляхом зміни ієрархії узагальнення (наприклад, Softmealy) або додавання нових функцій лексем (наприклад, SRV). Ручні системи, такі як Мінерва і TSIMMIS, де правили видобутку написані вручну, можуть бути адаптовані розробником обгортки для обробки не HTML документів. Деякі обгортки, наприклад WebOQL, W4F, XWrap і DEPTA, в значній мірі залежать від використання DOM дерев інформації в своїх системах, тому вони не можуть підтримувати не HTML документи, в той час як підходи на основі послідовності, такі як IEPAD, OLERA, RoadRunner і Dela можуть бути пристосовані для обробки не HTML документів, додаванням власних схем кодування. Технологія класів еквівалентності в EXALG також підтримує не HTML документи, але успіх залежить від диференціації ролей маркерів.

Рівень екстракції: завдання IE можна розділити на чотири категорії: на місцевому рівні, на рівні записів, на рівні сторінок і на рівні сайту. Rapier та SRV призначені для вилучення однослотових записів, що еквівалентно екстракції на рівні полів. Обгортки в EXALG і RoadRunner витягують вбудовані об'єкти даних зі всіх сторінок, які можуть містити записи кількох типів, тому обгортки в цих системах на рівні сторінки. Інші системи що залишилися в таблиці 2.1 є прикладами завдань IE на рівні записів, хоча деякі з них можуть бути розширені для вилучення на рівні сторінок, наприклад, NoDoSE, STALKER і т.д. Більшість IE систем на рівні записів виявляє межі записів, а потім розділяє їх на окремі об'єкти, в той час як стратегія вилучення з низу до верху в DEByE витягує набір атрибутів, а потім збирає їх, щоб сформувати звіт. Таким чином немає систем IE на рівні сайту.

Варіація цілі видобутку: Багато веб-сторінок організовані ієрархічно з декількома рівнями укладення. Як правило, ця незв’язана комплексна структура, представляючи варіації зі слабоструктурованих даних. Комплексний рівень мети екстракції (об'єкта даних) залежить від появи відсутніх атрибутів (MA), багатозначних атрибутів (MVA), багатовпорядковані атрибути (MOA), і вкладені об'єкти даних. Для обробки цих варіацій, процедура екстракції потребує особливого догляду, на додаток до своєї звичайної логіки, де з'являються ознаки рівно один раз без замовлення і вкладених питань.

Розуміння того, як різні системи IE підтримують ці зміни можуть допомогти нам вирішити, як адаптувати систему IE до нових завдань. Слід зазначити, що для систем видобутку на рівні поля (SRV та Rapier), обробка цих змін не представляє особливих труднощів, так як вони не мають справу з відносинами атрибутів об'єктів даних.

Більшість системи IE окрім WIEN та WHISK підтримують відсутність атрибутів та витяг багатозначних атрибутів. Спеціального догляду в системах IE, заснованих на програмуванні, наприклад Мінерва, W4F і WebOQL як правило вимагає обробник виключень. У TSIMMIS, два оператора "case" та "split" призначені для обробки відсутніх атрибутів та багатозначних атрибутів. Багато системи IE не підтримують множинного упорядкування атрибутів, оскільки їх правила видобутку залежать від розташування полів у записі. Хсу був піонером, який намагався подолати проблему множинного упорядкування атрибутів. Однак ситуації, що він оброблював, були випадки відсутніх атрибутів. Таким чином SoftMealy має обмеження для обробки MOA з використанням однопрохідного кінцевого перетворювача (finite state transducer, FST). Використання FST в SoftMealy, також дозволяє обробляти MA і MVA. В цілому, SoftMealy може обробляти об'єкти вкладених структур через багатоходової FST. Stalker може обробляти MOA та витяг вкладеного об'єкта за допомогою багатоходових сканувань вхідних даних. Інші системи IE (IEPAD, OLERA і Dela), використовують техніку вирівнювання, щоб сформувати диз’юнктивні правила для обробки MA, MVA, MOA. Крім того, використання декількох схем кодування в IEPAD і OLERA дає їм можливість обробляти більш складні вкладені об'єкти даних. Два евристичні компоненти видобутку в NoDoSE і стратегія знизу доверху (де набір атрибутів розпізнається, витягується і зберігається в наборі змінних перед самим об'єктом) в DEByE дає цим системам можливість обробляти MOA і вкладені дані об'єктів в цілому. RoadRunner та EXALG не підтримують MOA, тому що їх правила видобутку залежать від розташування атрибутів в запису, хоча в цілому, вони можуть обробляти вкладені об'єкти даних. DEPTA, теоретично може підтримувати вкладені об'єкти даних за рахунок використання структури дерева тегів. MOA

не можлива в DEPTA, оскільки дерево часткових збігів засноване на унікальному порядку дочірніх тегів з єдиним батьківським вузлом.

Варіація шаблонів: Труднощі в індукції правил витягів приходять з варіативності форматів екземплярів даних. Атрибут може мати варіантні формати (VF), які як правило вимагають підтримки диз'юнктивних правил або підтримки правила послідовності. Деякі системи IE підтримують обидва диз'юнктивних правила та послідовні шаблони (SP) для узагальнення правил. WIEN, W4F, XWrap, NoDoSE та RoadRunner не підтримують диз'юнктивні правила. Проте, W4F і XWrap підтримують послідовну модель для узагальнення правил. А регулярні вирази, що містять довільні символи, є прикладом послідовних шаблонів. Послідовні шаблони можуть бути узагальнені методом вирівнювання або шляхом послідовного шаблону видобутку (наприклад, Stalker). У той же час, різні атрибути можуть мати один і той же формат відображення – так званий загальний формат (CT). Більшість систем IE використовують переваги порядку атрибутів для їх вилучення. Інші, наприклад, DeBYE і Stalker, додають більше обмежень для формування більш тривалого правила вилучення. Далі слід, що точність вилучення може значно зменшитися у разі відсутності атрибутів або множинного порядку атрибутів.

Атрибути не розбиті на лексеми (UTA): До сих пір ми бачили три підходи обробки не розбитих на лексеми атрибутів. Перший з них через пост-обробку. Наприклад, оператор split в W4F пропонує регулярні вирази і обмеження для доступу до більш дрібних шматків даних. Другий - контекстними правилами замість правил на основі роздільників. Запропонована Softmealy ідея роздільників, а також контекстні правила допомагають користувачу мати доступ до даних будь-якого масштабу. Нарешті, багаторівневе кодування також дозволяє системам IE для адресувати дані різного масштабу, не жертвуючи перевагами абстракцій для узагальнення правил, як в IEPAD і OLERA.

Таблиця 2.1 – Аналіз систем IE на основі сфери завдань

Інструменти		Тип сторінок	Підтримка не HTML	Рівень екстракції	Варіація цілі видобутку			Варіація шаблонів		UTA
					MA / MVA	MOA	Вкладення	Формати варіантів	Загальний формат	
Керовані	Minerva	Напів структ.	Так	рівень запису	Так	Так	Так	обидві	За порядком	Так
	TSIMMIS	Напів структ.	Так	рівень запису	Так	Ні	Так	диз'юнк.	За порядком	Ні
	WebOQL	Напів структ.	Ні	рівень запису	Так	Так	Так	диз'юнк.	За порядком	Ні
	W4F	Частк.кер.	Ні	рівень запису	Так	Так	Так	послід.шабл	За порядком	Так
	XWRAP	Частк.кер.	Ні	рівень запису	Так	Ні	Так	послід.шабл	За порядком	Так
Контрольовані	RAPIER	довільний	Так	рівень поля	Так	-	-	диз'юнк.	інші обмеження	Так
	SRV	довільний	Так	рівень поля	Так	-	-	диз'юнк.	більше обмежень	Так
	WHISK	довільний	Так	рівень запису	Так	Так	Ні	диз'юнк.	За порядком	Так
	NoDoSe	Напів структ.	Так	сторінка/ запис	Так	обмежена	Так	Ні	За порядком	Ні
	DEByE	Напів структ.	Так	рівень запису	Так	Так	Так	диз'юнк.	більше обмежень	Ні
	WIEN	Напів структ.	Так	рівень запису	Ні	Ні	обмеженою	Ні	За порядком	Ні
	STALKER	Напів структ.	Так	рівень запису	Так	Так	Так	обидві	більше обмежень	Ні
	SoftMealy	Напів структ.	Так	рівень запису	Так	обмежена	мульти Pass	диз'юнк.	За порядком / один прохід	Так
Напів-контрольовані	IEPAD	Частк.структ.	обмеженою	рівень запису	Так	обмежена	обмеженою	обидві	За порядком	Так
	OLERA	Частк.структ.	обмеженою	рівень запису	Так	обмежена	обмеженою	обидві	За порядком	Так
Не контрольован	Dela	Частк.структ.	обмеженою	рівень запису	Так	обмежена	Так	обидві	За порядком	Ні
	RoadRunner	Частк.структ.	обмеженою	рівень сторінок	Так	Ні	Так	Ні	За порядком	Ні
	EXALG	Частк.структ.	обмеженою	рівень сторінок	Так	Ні	Так	обидві	За порядком	Ні
	DEPTA	Частк.структ.	Ні	рівень запису	Так	Ні	обмеженою	диз'юнк.	За порядком	Ні

2.5.2 Порівняння систем IE на основі методів

Для цього порівняння враховані основні використані методи. Отримані результати наведені в таблиці 2.2.

Сканування частот: це порівняння свідчить про кількість проходів сканування, необхідних для вилучення інформації з вхідного документа. Більшість систем WI побудовані так що екстрактор сканує вхідний документ разове (однопрохідний екстрактор), в той час як інші (наприклад, STALKER і багатоходова SoftMealy) для завершення екстракції сканують вхідний документ кілька разів. Екстрактор DEByE також потребує кілька проходів, щоб витягти кожен з атомарних атрибутів. Взагалі кажучи,

однопрохідні пакувальники є більш ефективними, ніж багато прохідні обгортки. Проте, багатопрохідні обгортки є більш ефективними при обробці об'єктів даних з необмеженими перестановками атрибутів або комплексним витяганням об'єкта. SRV і Rapier можуть генерувати тільки єдині правила слота, тому екстрактору необхідно зробити кілька проходів над сторінкою введення для вилучення відповідних даних.

Тип правил екстракції: більшість систем WI використовують правила видобутку, які представлені у вигляді регулярних граматик, для визначення початку та кінця відповідних даних, в той час як Rapier та SRV використовують правила видобутку виражені за допомогою логіки першого порядку. Правила регулярних виразів є потужними для напівструктурованих вхідних даних, особливо для сторінок на основі шаблонів, оскільки ми зазвичай знаходимо загальні маркери, що оточують дані для вилучення. Навіть якщо загальні маркери не існують, можна індукувати правила шляхом включення генералізації ієрархії маркерів як фонових знань (наприклад, Softmealy). Проте, для довільно-текстових матеріалів, в яких можна знайти дуже мало загальних маркерів, ми повинні включати додаткові функції, наприклад, щільність цифр, довжина, POS теги і т.д., щоб узагальнити загальні характеристики серед різних маркерів. Саме тому правила логіки першого порядку використовується для завдань IE довільного тексту (наприклад, SRV та Rapier).

Використані особливості: Більш ранні системи IE були призначені для обробки веб-сторінок без шаблонів, скажімо, веб-сторінки відділів комп’ютерних наук з різних університетів. Таким чином, вони використовували як HTML-тег так і літеральні слова, в якості обмежень на основі роздільників. Для шаблону на основі веб-сторінок, можна використовувати шляхи DOM дерева, щоб позначити певну частину інформації на веб-сторінці. Наприклад, W4F, XWrap і інші комерційні продукти використовують шляхи DOM дерева для адресування веб-сторінки. Оскільки дані для вилучення, часто розміщені в тому ж шляху DOM дерева, це робить процес навчання правила набагато простішим. Для вільного вилучення текстової інформації в якості додаткових функцій використовуються методи обробки

природної мови, такі як маркувальник частей мови та семантичні класи WordNet. SRV також використовує ортогональні функції, довжину маркерів, і посилання граматик. Нарешті, EXALG використовує статистичну інформацію маркерів на веб-сторінках, щоб зробити їх обгортки.

Таблиця 2.2 – Аналіз на основі використовуваних методів

Інструменти	Кількість сканувань	Тип правил екстракції	Використані особливості	Алгоритм навчання	Лексемізація шаблонів
Minerva	одне	Регулярні вирази.	HTML теги / Літеральні слова	ні	вручну
TSIMMIS	одне	Регулярні вирази.	HTML теги / Літеральні слова	ні	вручну
WebOQL	одне	Регулярні вирази.	гіпердерево	ні	вручну
W4F	одне	Регулярні вирази.	DOM дерево шлях адресації	ні	рівень тегу
XWRAP	одне	Контекстно-незалежний	DOM дерево	ні	рівень тегу
RAPIER	множинне	правила логіки	Синтаксичний / семантичний	ILP (знизу вверх)	рівень слова
SRV	множинне	правила логіки	Синтаксичний / семантичний	ILP (зверху вниз)	рівень слова
WHISK	одне	Регулярні вирази.	Синтаксичний / семантичний	Покриваючий набір (зверху вниз)	рівень слова
NoDoSE	одне	Регулярні вирази.	HTML теги / Літеральні слова	моделювання даних	рівень слова
DEByE	множинне	Регулярні вирази.	HTML теги / Літеральні слова	моделювання даних	рівень слова
WIEN	одне	Регулярні вирази.	HTML теги / Літеральні слова	Ad-Hoc (від низу до верху)	рівень слова
STALKER	множинне	Регулярні вирази.	HTML теги / Літеральні слова	Ad-Hoc (від низу до верху)	рівень слова
SoftMealy	обидва	Регулярні вирази.	HTML теги / Літеральні слова	Ad-Hoc (від низу до верху)	рівень слова
IEPAD	одне	Регулярні вирази.	HTML-теги	Pattern Mining, Вирівнювання строк	багаторівнева
OLERA	одне	Регулярні вирази.	HTML-теги	рядок Вирівнювання	багаторівнева
DeLa	одне	Регулярні вирази.	HTML-теги	Pattern Mining	рівень тегу
RoadRunner	одне	Регулярні вирази.	HTML-теги	рядок Вирівнювання	рівень тегу
EXALG	одне	Регулярні вирази.	HTML теги / Літеральні слова	Еквівалентних класів, роль диференціювання по DOM шляху дерева	рівень слова
DEPTA	одне	Дерево тегів	HTML теги tree HTML теги	Pattern Mining, порівняння рядків, часткове вирівнювання дерев	рівень тегу

Алгоритм навчання: Обгортки в системах WI на основі програмування пишуться від руки і приймають в якості вхідних даних специфікації, який декларативно встановлює, де дані інтересу знаходяться в HTML-сторінці і як дані упаковуються в об'єкти. Таким чином, в цих системах не використовуються ніякі алгоритми навчання. Rapier це реляційна система навчання знизу вверх натхненна методами ILP, в той час як SRV є реляційним алгоритмом зверху вниз. WHISK це система навчання з покриттям зверху вниз. Її модель має два компоненти, що визначають *контекст* та *точні роздільники* фрази що повинна бути вилучена. DEByE та NoDoSE вимагають велику кількість підтримки користувачів для моделювання даних в

документах. Вони зосереджені на розробці інтерфейсу і застосовують дуже прості методи навчання шаблонів вилучення, тобто загальний префікс і суфікс значень даних, які будуть вилучені. З іншого боку, Stalker та SoftMealy використовують методи узагальнення Ad-Hoc для вивчення правил вилучення. Вони зосереджені на методах навчання та архітектури екстрактора і використовують ієрархію класів лексем для їх узагальнення, що досить сильно відрізняється від NoDoSE і DEByE, де правила видобутку просто ґрунтуються на зовнішні або літеральні слова.

Напівкеровані або неконтрольовані системи IE в основному застосовують методи інтелектуального аналізу даних для відкриття різних шаблонів. IEPAD виявляє регулярні і суміжні максимальні моделі з використанням PAT дерев і методів вирівнювання рядка, в той час як Dela додатково виявляє вкладені структури з безперервно повторних шаблонів. OLERA застосовує наближене співпадіння строк та методи вирівнювання строк у встановлених користувачем межах, операції буріння вниз / розгортання вгору. RoadRunner аналізує вхідні сторінки шляхом порівняння рядків з використанням методу ACME. EXALG використовує статистичну інформацію для створення шаблону і схеми веб-сторінок за допомогою *класів еквівалентності* та *диференціації ролей* методів. DEPTA для видобутку записів даних з веб-сторінки застосовує техніку видобутку та часткове вирівнювання дерева. Для порівняння, IEPAD і DEPTA DISCOVER повторюють шаблони з однієї HTML сторінки, в той час як Roadrunner і EXALG виявляють повторювані шаблони з кількох HTML-сторінок.

Схеми лексемізації: Обгортувальники в Minerva та TSIMMIS написані вручну, тому не вимагається розмічування вхідних сторінок. Більшість систем WI для веб-сторінок підтримують на рівні тегів. Деякі системи підтримують навіть лексемізацію на рівні слова, наприклад, системи контрольованих WI та EXALG. WebOQL, W4F, XWrap, RoadRunner та Dela використовують схему кодування тегів рівня для транслювання вхідних сторінок в навчальні лексеми. Крім того, вхідна HTML сторінка в W4F та XWrap розбирається для побудови дерева розбору, яке відображає його ієрархію HTML тегів відповідно

до об'єктної моделі документа (DOM). Нарешті, IEPAD і OLERA дозволяють багаторівневе кодування для вхідних навчальних сторінок.

Порівняння систем IE на основі ступеня автоматизації

Для порівняння і оцінки систем IE використовувані характеристики ступеня їх автоматизації. Результати наведені в таблиці 2.3.

Досвід користувача: Керування системами IE вимагає від користувачів навиків програмування, для написання коректних правил видобутку. Контрольована та напівкерована системи WI вимагають від користувачів помаркувати цілком або частину даних, що будуть витягнуті, таким чином, спеціальних знань не потрібно. Для неконтрольованих систем, вони не вимагають допомоги користувача (окрім вибору шаблону). Для IEPAD і OLERA, хоча вони не вимагають маркувань перед відкриттям шаблону, для просівання необхідних даних вимагається післямаркування, в той час як робота з відокремлення маркерів шаблона від маркерів даних здійснюється шляхом неконтрольованих систем IE. Строго кажучи, мітка даних, що витягнених неконтрольованими системи IE залишається бути призначеною, і тільки Dela владнав цю проблему.

Підтримка отримання навчальних сторінок: Більшість систем IE зосереджуються на узагальненні правил вилучення та використанні набору сторінок, які вручну скачуються в якості навчальних прикладів. Деякі системи спеціально підтримують скачування сторінок при будівництві обгортки. Наприклад, W4F має компонент під назвою *RetrieveAgent*, який використовується для отримання веб-джерел шляхом введення його URL. Крім того, компонент XWrap *синтаксичний нормалізатор* приймає URL, який Ви самі ввели, видає HTTP запит на віддалений сервер, ідентифікуємий за допомогою URL-адресу і отримує відповідну веб-сторінку. Інші системи також пропонують нові інструменти для підтримки отримання сторінки. Наприклад, WNDL є мовою, запропонованою Хсу та ін. для опису веб-навігації для отримання сторінки в Softmealy та IEPAD []. ASByE, подальший розвиток DEByE, є інструментом для збору статичних і динамічних веб- сторінок. Dela використовує існуючий краулер глибинної мережи *HiWe*,

що автоматично збирає ярлики елементів з веб-сайтів та відправляє запити до конкретного веб-сайту.

Підтримка виводу інформації / API: Виведення відповідних витягнутих даних є порівняно простим, тому більшість систем IE його підтримують. Системи Minerva, W4F, XWrap, NoDoSE, DEByE, SoftMealy, OLERA та RoadRunner виводять отримані дані в форматі XML. Крім того, NoDoSE підтримує інші формати, такі як OEM, а DEByE підтримує вихідний формат бази даних SQL. З іншого боку, підтримка API є важливою, оскільки це зв'язок між згенерованими обгортками та системами інтеграції інформації. Системи IE на основі програмування мають підтримку API, в той час як інші не згадують про це в своїх роботах.

Застосування: застосовність підходів стосується того, як легко вони можуть бути поширені й на інші домени завдань. Ключовим фактором високої застосовності є те, що специфічна для конкретного домену інформація відокремлюється від базисного механізму навчання. Для різних завдань IE, ручні та контрольовані системи мають гарну модульність, а напівкеровані або некеровані системи мають меншу придатність, оскільки вони підштовхнули домен конкретної інформації до межі високого ступеня автоматизації.

Обмеження: Нарешті, ми розглянемо вимоги до множинного запису даних або введення декількох навчальних сторінок. Хоча ми можемо розглядати такі вимоги, як різні вхідні завдання IE, розглянемо їх як обмеження цих підходів для різних систем WI що необхідно порівняти в одній і тій же області завдань. Візьміть шаблон IE сторінок, наприклад, системи IE, що потребує багато-записних навчальних веб-сторінок. Він не може бути застосований на сайт, що включає в себе веб-сторінки одиничного запису. Як показано в таблиці 2.3, для ручних та керованих систем IE не існує ніяких обмежень у відношенні змісту і кількості навчальних сторінок. IEPAD, Dela та DEPTA для створення обгортки вимагають введення сторінок з множинними записами. Dela, RoadRunner, EXALG для власних методів роботи вимагають більше однієї навчальної сторінки в якості вхідних даних.

Таблиця 2.3 – Аналіз на основі ступеня автоматизації

Інструменти	Досвід користувача	Підтримка скачування прикладів	Вихід / API підтримка	Застосовність	Обмеження
Minerva	програмування	Немає	XML	висока	Не обмежене
TSIMMIS	програмування	Немає	текст	висока	Не обмежене
WebOQL	програмування	Немає	текст	висока	Не обмежене
W4F	програмування	Так	XML	середня	Не обмежене
XWRAP	програмування	Так	XML	середня	Не обмежене
RAPIER	маркування	Немає	текст	середня	Не обмежене
SRV	маркування	Немає	текст	середня	Не обмежене
WHISK	маркування	Немає	текст	середня	Не обмежене
NoDoSE	маркування	Немає	XML, OEM	середня	Не обмежене
DEByE	маркування	Так	XML, SQL DB	середня	Не обмежене
WIEN	маркування	Немає	текст	середня	Не обмежене
STALKER	маркування	Немає	текст	середня	Не обмежене
SoftMealy	маркування	Так	XML, SQL DB	середня	Не обмежене
IEPAD	Вибір шаблону, післямаркування	Немає	текст	низька	Множинні записи сторінок
OLERA	часткове етикетування	Немає	XML	низька	Не обмежене
DeLa	вибір шаблону	Так	текст	низька	Множинні записи сторінок, більше однієї сторінки
RoadRunner	вибір шаблону	Так	XML	низька	Більше однієї сторінки
EXALG	вибір шаблону	Немає	текст	низька	Більше однієї сторінки
DEPTA	вибір шаблону	Немає	SQL DB	низька	Множинні записи сторінок

2.5.3 Порівняння систем IE в цілому

Вище було порівняно різні системи IE з трьох критеріїв, між ними існує кореляція. Наприклад, сторінки на основі шаблонів мають більш високу ступінь автоматизації, ніж сторінки без шаблонів і довільні текстові документи оскільки вхідні документи представляють структуроване середовище, що може бути досягнене за допомогою неконтрольованих підходів. Однак, це не означає, що вилучення даних зі сторінок на основі шаблонів простіше, ніж з інших сторінок. Замість цього, виникають нові проблеми, наприклад, відмінність між шаблоном та лексемами даних, призначення мітки лексемам даних.

Як показано на рисунку 2.16, ручні системи IE можуть бути застосовані до всіх видів вхідних документів, до тих пір поки відповідні характеристики забезпечуються системами, хоча складання правил видобутку залежить від методів програмістів. Напівкеровані та неконтрольовані системи IE можуть бути застосовані тільки до сторінок на основі шаблонів, оскільки їх успіх залежить від існування шаблонів. Крім того, ми також бачимо, що неконтрольовані системи зазвичай застосовують зовнішні характеристики, такі

як HTML - теги для правил регулярних виразів, оскільки вони призначені для сторінок на основі шаблонів. Для IE зі міжсайтових сторінок і довільних текстів, вимагаються семантичні ознаки (наприклад, орфографічні особливості, довжина лексеми і т.д.), оскільки існують менш поширені теги та слова серед вхідних документів.

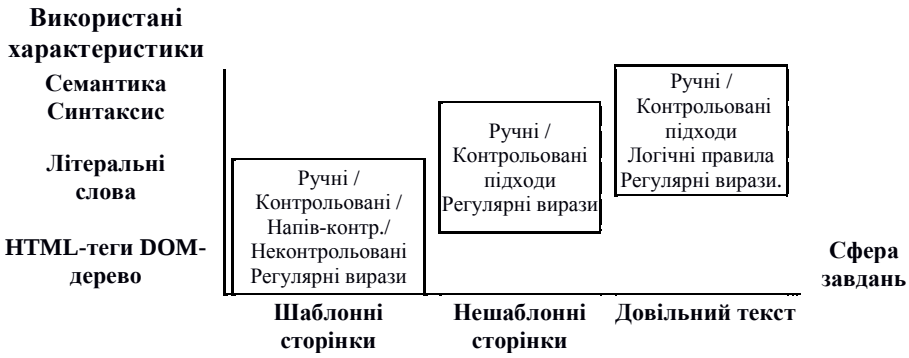


Рисунок 2.16 – Порівняння у цілому.

Для практикуючого, який хоче знати, які методи є ефективними, добре буде згадати і про точність. Оскільки ці системи працюють з різними даними і мають різні функції, не представляється можливим оцінити їх на несуперечливій основі. Таким чином, ми можемо тільки порівняти їх за їх застосовністю. Напівконтрольовані і неконтрольовані системи IE мають вбудовані в їх системи евристику що спостерігається з шаблонів сторінок, наприклад, з суміжної області даних (IEPAD), з несуміжних записів даних (DEPTA), з вкладених об'єктів даних (Dela). Оскільки існує багато їх варіантів в Інтернеті, немає ніякої гарантії, що такі методи працюють для всіх веб-сторінок, хоча недавно запропоновані підходи можуть вирішити більше сторінок, ніж минулі підходи. Що стосується контрольованих підходів, оскільки дані для вилучення помічаються користувачами, їх придатність порівняно краще, ніж неконтрольованих систем. Тим не менше, немає ніякої гарантії успіху у індукції правил.

Для дослідника, який хоче знати, який метод застосовувати при прискореній адаптації наявних систем до нового домену завдань IE. Як вже говорилося вище, методи, використовувані в неконтрольованих системах

IE важко поширити на довільні тексти і навіть на сторінки без шаблону, оскільки багато евристики може застосовуватись тільки до сторінок на основі шаблону. Для контрольованих підходів, ми бачили добре відомі методи навчання (наприклад, ILP і встановити покриття в SRV, WHISK і т.д.), а також Ad-Hoc навчання (узагальнення знизу доверху в Stalker, Softmealy і т.д.). Методи навчання Ad-Hoc більш швидкі в процесі навчання шляхом включення маркера ієрархії для узагальнення. Контрольовані підходи високо ціняться, оскільки можливо додати нові функції в існуючі системи без зміни алгоритмів навчання. Хоча тільки ILP і покриває безліч алгоритмів що використовуються в даний час, було б цікаво подивитися й на застосовування інших алгоритмів навчання (наприклад, підтримка векторної машини і т.д.).

Висновки до розділу 2

Виходячи з вищесказаного можна зробити наступні висновки. Добре структуровані документи, в яких корисна інформація міститься всередині тегів в кінцевому вигляді дозволяють застосовувати підходи з високим ступенем автоматизації. Крім гарної структури на сторінці повинно бути досить однотипних записів, щоб алгоритми могли виявити структури даних. У тих випадках коли всередині тегу може міститися деяка зайва інформація, то необхідно аналізувати вільний текст. З подібним завданням краще справляються розглянуті ручні і контрольовані системи. У ручних системах користувач може точно описати правило для добування інформації, а до контрольованих системах уявити великий навчальний набір який враховує різні варіанти.

Якщо кількість сайтів з яких необхідно отримувати інформацію невелике і всі вони мають різну структуру, то ефективніше в даному випадку застосувати ручні системи тому підготовка навчальних матеріалів може виявитися більш трудомістким завданням. Таким чином більш автоматизовані системи будуть більш ефективні на великій кількості сайтів зі схожою структурою видобутих даних.

Використання напівкерованих та керованих методів найбільш доцільне при необхідності подальшого розповсюдження дії програмного комплексу на інші предметні онтології та розширенні онтології сайтів.

Завдяки використанню регулярних виразів швидкість семантичної обробки на порядок вищій ніж для інших методів.

Тому під час розробки використані загальні алгоритми методів видобутку даних Minerva та TSIMMIS, що використовують ручну лексемізацію шаблонів та використовують регулярні вирази для формулювання правил екстракції.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ ПОШУКУ ОГОЛОШЕНЬ АВТОМОБІЛЬНОГО РИНКУ

Даний розділ присвячений розробці програми для видобування інформації про автомобілі з структурованих та неструктурованих веб-сайтів та розробці веб-інтерфейсу для відображення знайдених оголошень. У розділі розглянуто загальну архітектура додатків, описані алгоритми які застосовуються для вилучення інформації. Також розглянута процедура збереження отриманих результатів в БД. Розглянуто загальну структуру веб-інтерфейсу і робота системи фільтрів для фільтрації об’єктів автомобільного ринку в базі даних за такими критеріями як: дата публікації, тип кузову, тип коробки перемикання швидкостей, ціна.

Сервіс був розроблений на мові програмування Php 5.5. Веб-сервіс буде розташований на локальному сервері Apache.

Для первинного отримання веб сторінок використовувалися також стандартні бібліотеки php – cURL, Xpath.

Побудова онтологій автомобільного ринку для неструктурованих сайтів розроблялася методом регулярних виразів [95], за допомогою яких можна отримувати досить складні ланцюжки символів, що представляють собою, наприклад, дату і час, телефон, URL і т.п.

Отримані під час аналізу дані передаються в базу даних СУБД MySQL.

3.1 Аналіз вихідних сторінок

Дошка оголошень http://board24.lg.ua/ насичена досить великою кількістю оголошень, що не зустрічаються у структурованих https://olx.ua/ або https://auto.ria.com/.

В ході вивчення структури сайтів, що містять оголошення про об'єкти автомобільного ринку, було з'ясовано, що здебільшого вони містять динамічні списки, де представлені тільки посилання на самі оголошення. Оскільки

оголошень на сайті досить багато, вони можуть бути розташовані на декількох сторінках, перехід за якими зазвичай здійснюється кнопками навігації.

Розроблений механізм «перегортання» сторінок здійснює послідовний перехід по сторінках зі списками. Необхідні для цього налаштування індивідуальні для кожного сайту і зберігаються в онтології сайтів про автомобільний ринок.

Загальний алгоритм отримання даних з окремого сайту з неструктурованими даними в розробеному сервісі виглядає таким чином:

– скрапінг / краулінг початкової сторінки шляхом застосування cURL, побудови DOM об’єкту XPath;

– отримання посилань на проміжні сторінки (зазвичай отриманий перелік посилань багатосторінковий) з початкової сторінки шляхом XPath - запитів;

– отримання кінцевих сторінок з проміжних сторінок;

– нормалізація вихідного тексту;

– видобуток значущої інформації шляхом застосування регулярнх виразів (PCRE);

– видалення дублікатів, не релевантних або недостатньо насичених значимою інформацією;

– передання отриманої структурованої інформації до бази даних Estate утвореної в phpMyAdmin (СУБД MySQL).

Парсер необхідно налагодити як під вимоги певного сайту так і під вимоги певної предметної області. На вхід програми надходить веб документ і правила обробки, далі за певними правилами виконується перетворення даних у набір об'єктів предметної області.

Загальна схема приведена на рис. 3.1

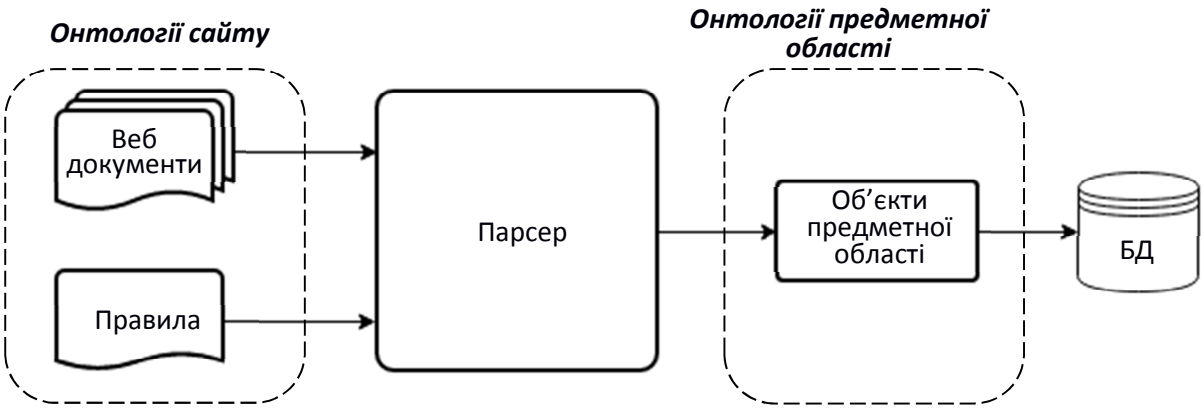


Рисунок 3.1 - Загальна структура парсеру

Правила описують якими методами буде витягнута інформація і сформований об'єкт. Метод - це алгоритм витягнення, фільтрації, перетворення вхідних даних, наприклад, витяг з DOM дерева по CSS селектору, фільтрація по регулярному виразу і.т.д. Результат роботи методу може бути записаний у поле результуючого об'єкта, або переданий іншим методом.

Розглянемо набір правил на прикладі сайту http://board24.lg.ua/, рисунок 3.2. Об'єктом предметної області є автомобіль. Для прикладу виділимо 3 властивості: текст оголошення, дата публікації, телефон. На даній сторінці представлена інформація про легкові автомобілі класу С у вигляді списку на декількох сторінках. Необхідно визначити правило, що відокремлює конкретний об'єкт. Проаналізувавши HTML код вручну або за допомогою модулю FireBug FirePath можна помітити що елементи списку оголошень знаходяться всередині елементів вузлу div скорочений путь до якого на мові XPath може виглядати таким чином: .//*[@id='content']/div/div[2]/div[2] . Таким чином можна сформулювати перше правило витягу об'єктів (сторінки з кінцевими посиланнями) зі заздалегідь сформованого XPath-примірнику документу:

$resultsPageUrls = $resultsPageXPath->query('//*/div/div[2]/div[2]/a/@href');

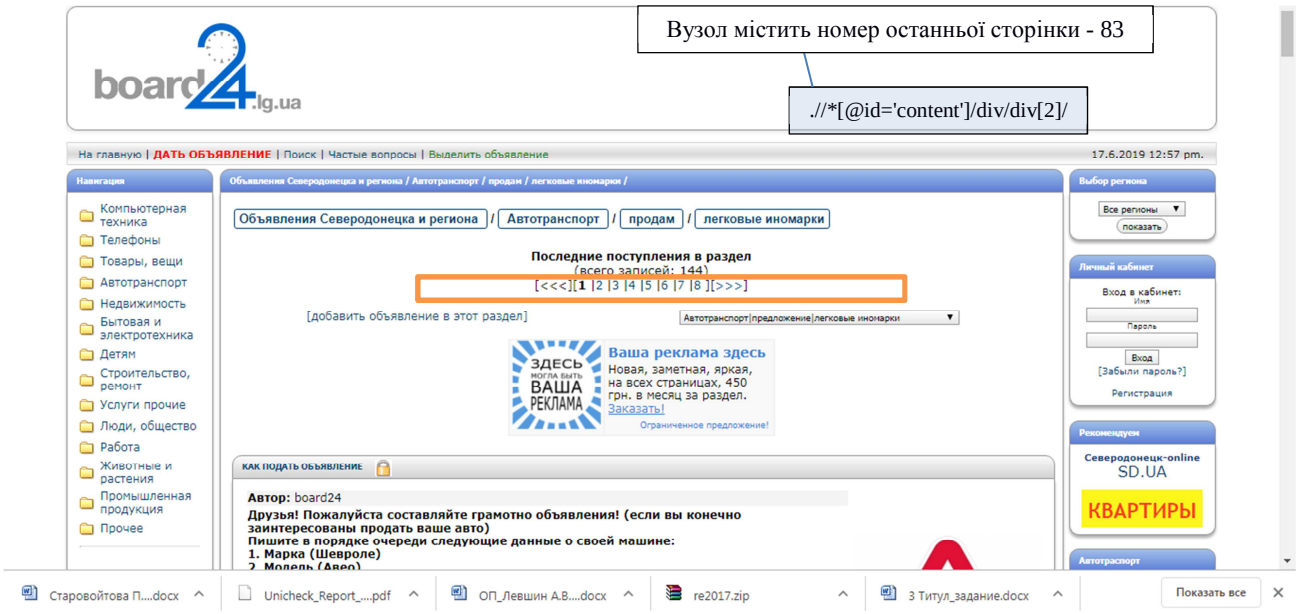


Рисунок 3.2 – Аналіз структури початкової сторінки

Вузол //*/div/div[2]/div[2] також містить номер кінцевої сторінки – списку, за допомогою якого встановлюється граничний номер сторінки з посиланнями. Після циклічного проходу по всіх сторінках (в прикладі на рис. 3.2. їх 83) формуємо кінцеві посилання. Перелік сторінок з посиланнями генерується в циклі з додаванням шаблону адресації таких сторінок:

```
for ($i=1; $i<$maxpage; $i++)
        $resultsPages[] = 'http://www.board24.lg.ua/real/sell/room1/page'.$i.'.html';
```

Кожна сторінка містить 20 кінцевих посилань, з якими й відбуватиметься основна робота.

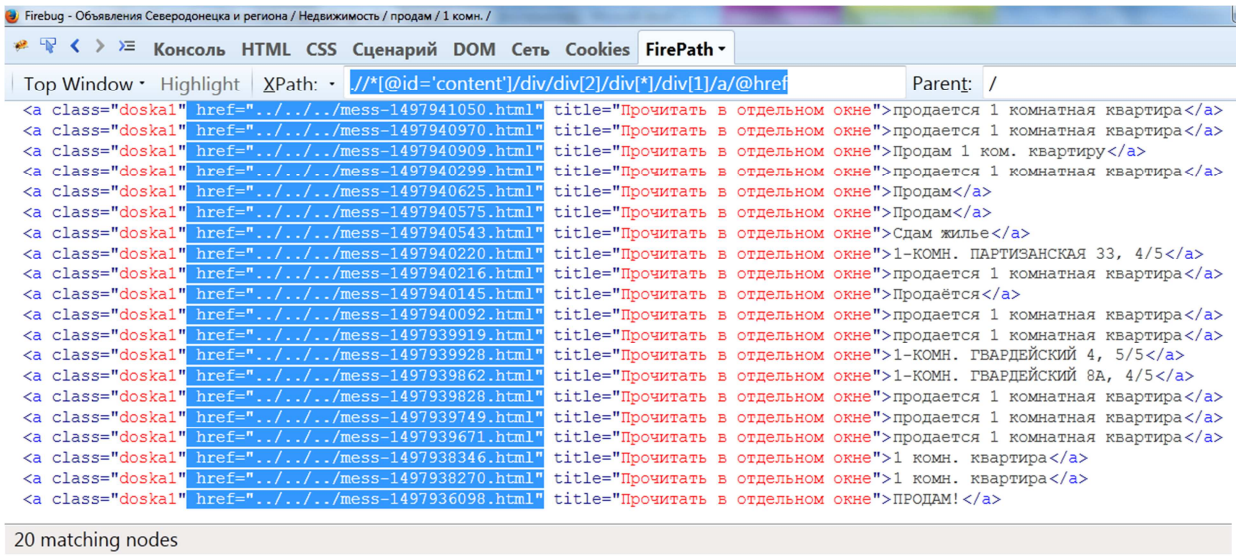


Рисунок 3.3 – Аналіз в FirePath структури вузлу з кінцевими посиланнями

Аналізуємо кінцеві сторінки з оголошеннями.

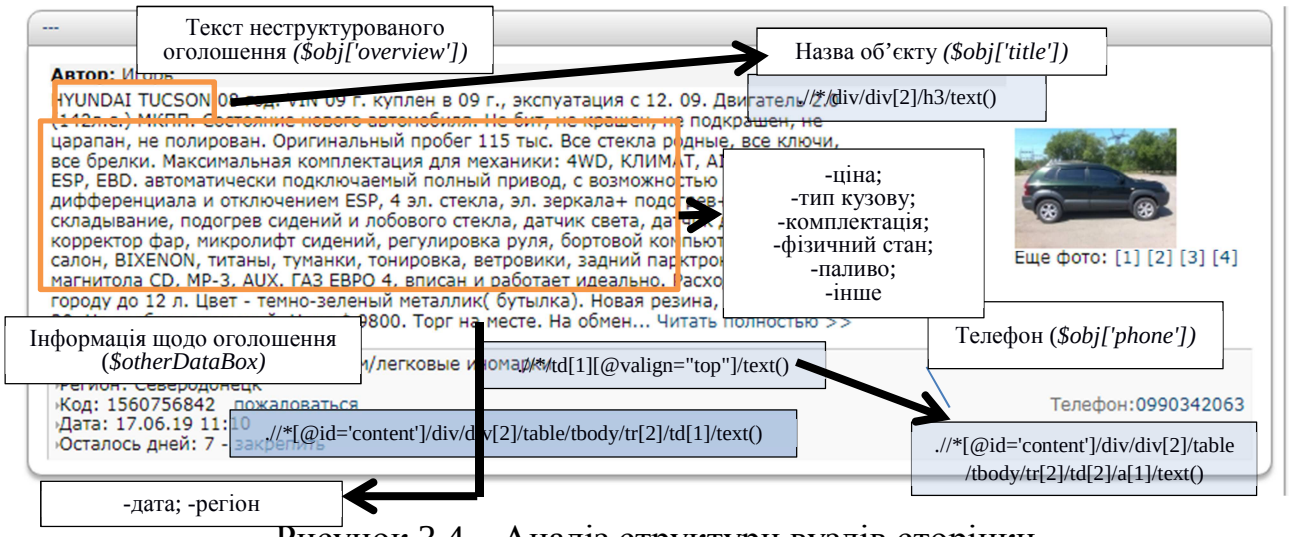


Рисунок 3.4 – Аналіз структури вузлів сторінки

Далі можна помітити те що назви об'єктів знаходяться всередині вузлів .//*/div/div[2]/h3/text(). Контактний телефон перебувати всередині тега <a> батьківського тегу з класом id="content".

У блоку тексту з інформацією щодо оголошення у вузлі .//*[@id='content']/div/div[2]/table/tbody/tr[2]/td[1]/text() міститься частково структурована інформація про дату публікації та населений пункт. Однак дати та населеного пункту в даному тезі міститься зайва інформація.

З отриманого результату необхідно вибрати дату та населений пункт використовуючи метод отримання даних по регулярним виразам '/(?<=Дата:\s)(\d\d\.\d\d\.)(\d\d)/iu' та '/(?<=Регион:\s)\w+/iu', а після перетворити текстову дату в unix time для зручної роботи з датою.

Основна неструктурована частина оголошення міститься у вузлі .//*/td[1][@valign="top"]/text()'. Складність є у тому, що сайт оголошень не містить спеціальних тегів для найбільш важливої для аналізу інформації щодо автомобілів. До того ж користувачі, які публікують оголошення, припускають багато синтаксичних та орфографічних помилок, користуються досить різною стилістикою та скороченнями. Суттєва інформація може бути видобута шляхом складання правил регулярних виразів для всіх ще нерозглянутих елементів онтології автомобільного ринку.

Надалі основна частиина прогами – семантичний аналізатор реалізує базову логіку методу обробки інформації.

Таблиця 3.1. Використані для аналізу шаблони регулярних виразів

Елементи онтології автомобілей	Основний шаблон
Вартість авто	'/(\d{0,3}(?:[|\.|,]{0,1})\d{3}).{0,2}[\$|y\.\s?e\.|USD|доллар]/ui' , '/\d{0,2}[\s|\.|,{0,1}\d{3,}\s[(?=гривен)|(?=грн)]/ui'
Загальний пробіг	'/\d+,?\d?\s?(?:км)|(?:тис.км)|(?:тыс.км)/ui' , '/(?:общ\..?пробег|общий пробег|общим пробегом)\D{0,3}\d{2,3}[,\.]?\d?/ui'
Пасажирських місць	'/\d{2,4,7}\s?\/\s?\d{2,4,7}/'
Фізичний стан	'/(?:[,\.])(\s?[а-яА-ЯЁё]+\s?)*(?:норм.?|авар.?|ремонт.? |окрас.?)\s?[а-я]*/ui'
Об'єм двигуна	'/\d{0,9}\s?[.]?/ui',
Терміновість продажу	'/срочн/ui'

3.2 Компоненти програми

Можна виділити наступні основні компоненти розроблюваної програми для пошуку та аналізу даних: завантажувач веб документів, обробник інформації, експортер даних. Загальна схема приведена на рис. 3.5.

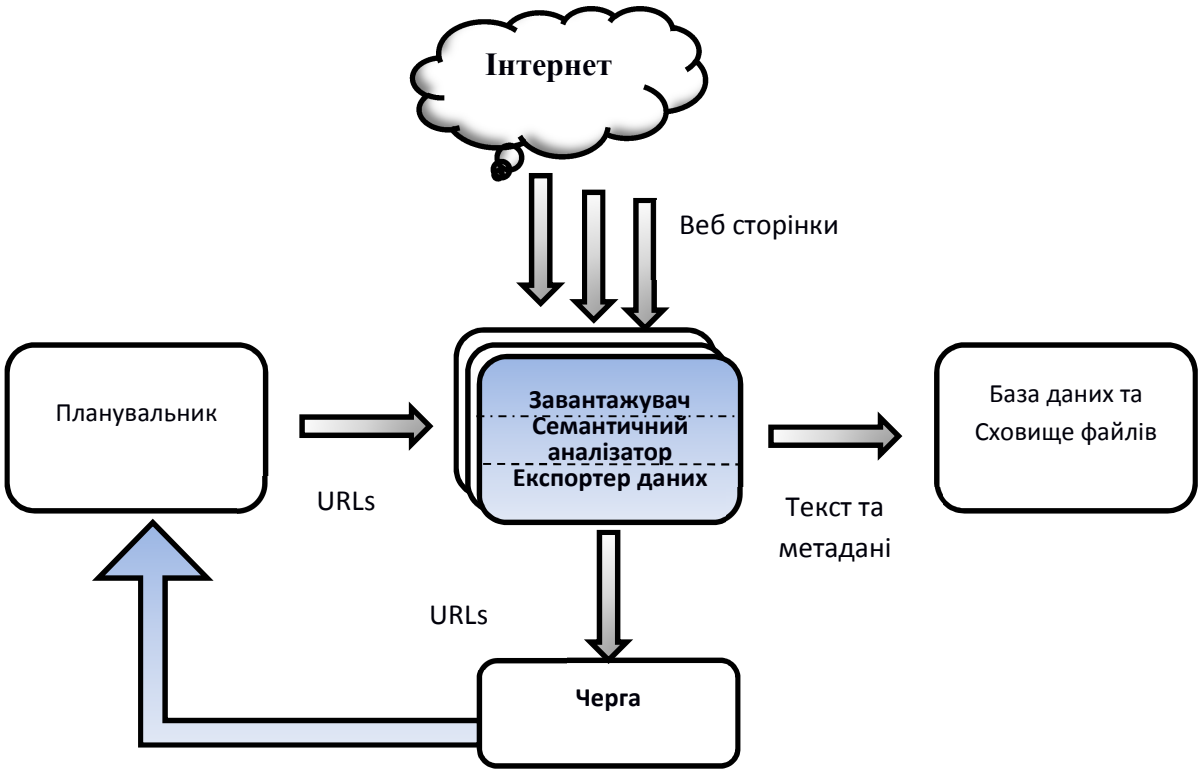


Рисунок 3.5 – Структура програми

3.3 Завантажувач веб документів

Для завантаження веб документів використана службова бібліотека командного рядка cURL. Дана утиліта підтримує протоколи: FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, LDAP, POP3, IMAP і SMTP [96]. Також cURL підтримує роботу з HTTP-cookie і туннелювання через проксі сервер, що дозволить в подальшому вирішити дві проблеми видобутку контенту з веб ресурсів - авторизація на сайті і обмеження кількості запитів з однієї адреси. PHP має вбудовану підтримку cURL (є відповідна бібліотека).

Модуль завантажувача містить метод curlGet для завантаження контенту та returnXPathObject для отримання структури DOM сторінки.

```
function curlGet($url) {
	$ch = curl_init();	//Initialising cURL session
	//Setting cURL options
	//curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
	curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);	//Returning transfer as a string
	curl_setopt($ch, CURLOPT_FOLLOWLOCATION, TRUE);	//Follow location
	curl_setopt($ch, CURLOPT_URL, $url);	//Setting URL
	$results = curl_exec($ch);	//Executing cURL session in the results
	return $results;	//Return the results
}

function returnXPathObject($item) {
	$xmlPageDom = new DomDocument();	//Instanting a new DomDocument object
	@$xmlPageDom->loadHTML($item);	//Loading the HTML from downloaded page
	$xmlPageXPath = new DOMXPath($xmlPageDom);	//Instanting new XPath DOM object
	return $xmlPageXPath;	//Return XPath object
}
```

Рисунок 3.6 – Лістінг модуля завантаження

3.3.1 Методи вилучення даних

Вибрати дані з html файлу можна аналізуючи його синтаксис, теги, класи, ідентифікатори і.т.д. Для розбору html файлу можна використовувати мову XPath. XPath - мова запитів до елементів DOM дерева документа. Дана мова

має широкі можливості для вибору елементів такі як строкові, числові, логічні функції які дозволяють складати умова перевірки.

Інший спосіб вибрати елемент з DOM дерева - використання CSS селекторів. Даний метод має менше можливостей, але має більш простим і лаконічний синтаксис.

Для роботи з DOM деревом був обраний компонент FireBug FirePath. Даний компонент дозволяє виконувати XPath запити.

Фрагмент коду реалізує вилучення даних по XPath запити наведено нижче:

```
1. $overview = $objPageXpath->query('.//*/td[1][@valign="top"]/text()');
   // Запрос тела неструктурированного объявления
2. if ($overview->length > 0) {
3. for ($i = 0; $i < $overview->length; $i++)
4. $obj['overview'][$n][] = $overview->item($i)->nodeValue.' '; } //
   Добавить текст объявления в массив
```

Для отримання даних з простого тесту використовуються регулярні вирази.

Модуль регулярних виразів отримує дані за регулярними виразами та зберігає в архиві до передачі у базу даних.

3.3.2 Метод зіставлення

Одна з проблем при отриманні даних є зіставлення даних з конкретного сайту з даними всередині бази даних. Наприклад, в програмі є словники кузовів та видів палива, а на сайті по автомобілям ці ж об’єкти можуть міститися у довільному тексті оголошення, але з представленням різними наборами символів. Так само одне і теж слово може бути написано в різних формах (множина, відмінки і т.д) для вирішення цієї проблеми можна скористатися методами морфологічного аналізу тексту. Використовуючи дані методи можна привести слова до загальної форми і зробити порівняння. Однак подібне

рішення не враховує слова синоніми. Рішенням цієї проблеми може послужити база синонімів, або технології машинного навчання.

Блок перевірки співпадінь реалізує алгоритмом пошуку ключового слова в тексті на основі поповнюванго переліку районів та переліку вулиць. Однак в алгоритмі простого пошуку існує ряд недоліків. Основним недоліком є те що слово може мати різні морфологічні форми, що робить просте посимвольного порівняння неможливим. Для вирішення даної проблеми можна скористатися відстанню Левенштейна або методами морфологічного аналізу.

Відстань Левенштейна - це мінімальна кількість операцій вставки одного символу, видалення одного символу і заміни одного символу на інший, необхідних для перетворення одного рядка в іншу. При використанні даного алгоритму проводитися розрахунок відстані для двох пар слів, вихідного слова з тексту і шуканого слова, якщо відстань менше максимально допустимого відстані, то вважається що це одне і теж слово. Даний алгоритм реалізований в стандартній бібліотеці php і має складність O (m * n), де n і m - довжини рядків [97]. До основних недоліків даного методу можна віднести відносно високий рівень помилкових спрацьовувань.

Використання методів морфологічного аналізу в значній мірі допоможе скоротити кількість помилкових спрацьовувань. Ідея даного методу полягає в приведенні всіх слів до початкової формі. Коли все слова приведені до початкової формі, і до єдиного регістру тоді можна виконувати просте порівняння рядків. Для приведення слів до початкової формі існує php бібліотека phpMorphy. Дана бібліотека підтримує безліч мов таких як: Російська, англійська, німецька (AOT). Український, Естонський (на основі ispell). Є можливість додати підтримку інших мов за допомогою myspell словника [98].

3.3.3 Метод виділення валюти

Під час аналізу автмобіля, важливо визначити цінову пропозицію продавця. У видобуванні вартості можна виділити дві проблеми. Користувачі використовують різні формати запису та визначення типу валюти, яку необхідно надалі сконвертувати.

Для вирішення цієї проблеми можна використовувати регулярні вирази або використовувати методи машинного навчання.

Блок парсингу ціни продажу реалізує виділення грошових величин з тексту на основі регулярних виразів. Першим кроком алгоритму є видалення незначущих символів. До таких символів можна віднести прогалини і коми розділяють сотні в числі. За допомогою функції preg_replace відбувається видалення незначущих символів, код виклику даної функції наведено нижче:

```
1.  preg_replace(“/(\d)(, \.)(\d)/”,'$1$3', $clearedData);
```

Також на першому кроці виконується приведення всіх символів до нижнього регістра, це необхідно для уніфікації записи грошових одиниць, наприклад: "грн.", "ГРН.", "Грн."

В результаті видалення незначущих символів тепер можна вибрати всі числа регулярним виразом:

```
1.  /(\d+)/
```

Однак у вхідному тексті можуть міститися і інші числа не є грошовою сумою. Ознакою грошової суми буде позначення валюти до або після числа. Таким чином можна сформувати наступні регулярні вирази:

```
1.  /(грн.|$)\s*(\d+)/
2.  /(\d+)\s*(грн.|$)/
```

У випадку зазначення ціни у гривні буде виконана конвертація у валюту, в якій переважно публікуються оголошення з продажи авто – долар США.

Нижче наведено фрагмент коду для вилученя валюти.

```
3. if (preg_match('#(\d{0,3}(?:[ |\.|,]{0,1})\d{3}).{0,2}[\$|y\.\s?e\.|USD|доллар]#ui', $obj['overview'][$n], $obj['price'][$n])) $obj['cur'][$n]='USD';//
4. if (!$obj['cur'][$n]='USD')
```

```
5.  {if (preg_match_all('/(\d)+ *(?=грн|гривен)/ui', $obj['overview'][$n], $obj['price'][$n])) $obj['cur'][$n]='UAH';
6.  if (preg_match('#\d{0,2}[\s|\.|,{0,1}\d{3,}\s[(?=гривен)|(?=грн)]#ui', $obj['overview'][$n], $obj['price'][$n])) $obj['cur'][$n]='UAH';
7.  }
8.  if (is_null($obj['price'][$n])) preg_match('#(цена.?\s{0,2})\d{0,3}[\.]{0,1}\d{3}#ui', $obj['overview'][$n], $obj['price'][$n]);
9.  $obj['price'][$n] = preg_replace('/[^\d]/', '', $obj['price'][$n]);
```

Далі наведено фрагмент коду для конвертації валюти.

```
1. function getCourse($date){
2. $date=preg_replace('/(\d\d)(\d\d)(\d\d)/iu', '20${3}${2}${1}', '140617');  //YYYYMMDD
3. $link = file_get_contents('https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?valcode=USD&date='.$date.'&json');
4. $postValueFromLink = json_decode($link, true);
5. implode($postValueFromLink);
6. return $postValueFromLink[0]['rate'];
7. }
```

3.2.4 Перетворення даних

Іноді отримані дані вимагають перетворення. Наприклад для перетворення строкової дати в UnixTime під час передачі до БД програма конвертує вхідну інформацію в UnixTime.

3.4 Збереження отриманих результатів в БД

Результатом роботи парсера є масив об'єктів. Для збереження результатів в базу даних використовується технологія об'єктно реляційного відображення (ORM). Під час роботи використана СУБД MySQL. Модуль зберігання реалізує збереження отриманих об'єктів в базу даних. Для збереження об'єктів, у них повинен бути визначений унікальний ключ, він необхідний для того щоб визначити чи міститься даний об'єкт в базі даних і вимагає оновлення, або ж

потрібно додати новий об'єкт. Унікальний ключ формується під час завантаження інформації в БД.

Для подальшого використання отриманої інформації достатньо простої структури бази даних з однією таблицею.

Таблиця 3.2. Загальний вигляд структури БД

№ з/п	Ім’я	Тип даних	Комментар	Додатково
1	**id**	int(9)	Ідентифікатор запису	AUTO_INCREMENT
2	**name**	varchar(100)	Назва оголошення	
3	**region**	varchar(50)	Регіон	
4	**city**	varchar(20)	Город	
5	**marka**	varchar(60)	Марка	
6	**model**	varchar(20)	Модель	
7	**year**	int(4)	Рік	
8	**dist**	int(10)	Пробіг	
9	**kuzov**	int(10)	Тип кузову	
10	**mesta**	int(10)	Кількість місць	
11	**val**	float(10,1)	Обсяг двигуна	
12	**patrol**	varchar(20)	Паливо	
13	**price_total_usd**	int(10)	Цена общая $	
14	**price_total_uah**	int(10)	Цена общая грн	
15	**color**	varchar(20)	Колір	
16	**price1_uah**	int(10)		
17	**date_publish**	Date	Дата публікування оголошення	
18	**date_load**	Date	Дата завантаження оголошення	
19	**url**	varchar(100)	URL оголошення	
20	**photo**	Blob	Файли з фотографіями	
21	**add_info**	varchar(5000)	Додаткова інформація	
22	**contacts**	varchar(1000)	Контакти	
23	**urgence**	tinyint(1)	Терміновість	
24	**screenshot**	blob	Копія екрану з оголошенням	

3.5 Результати роботи сервісус

Розроблений онлайн сервіс розміщений на локальному сервері Apache. Запуск програми здійснюється вручну після обрання параметрів пошуку. Під час запиту можливо обрати населений пункт, в якому здійснюється збір даних, тип кузову, а також внести додаткові орієнтири для коригування типа автомобіля, що продається.

URL	http://board24.lg.ua/motor/sell/car/page2.html
otherData	Продам Хонда CR-V , чистый 2008 г.в., двиг. 2.4, АКПП, максимальная комплектация кроме кожи, установлена дорогая газовая установка, технически безупречное состояние. Мотор, коробка в идеале, после большого ТО, замена всех жидкостей, расходников. Без ДТП,Кузов целый, не перекрашеный. Кто ищет целый, обслуженный автомобиль вам сюда.Шикарный семейный кроссовер, с динамичным, надежным и экономичным мотором, расход газа по городу до 12л/100 км, трасса 8/100. Цена 11500 у.е. Торг у капота, варианты адекватного обмена с вашей доплатой.
overview	
title	ПРОДАМ
pubDate	16.06.19
city	Северодонецк
price	11 500
marka	Хонда
model	CR-V
year	2008
dist	12л/100 км
kuzov	кроссовер
mesta	семейный
val	2.4
patrol	газовая установка

Рисунок 3.7 –Зовнішній вигляд бази даних після зберігання отриманих даних

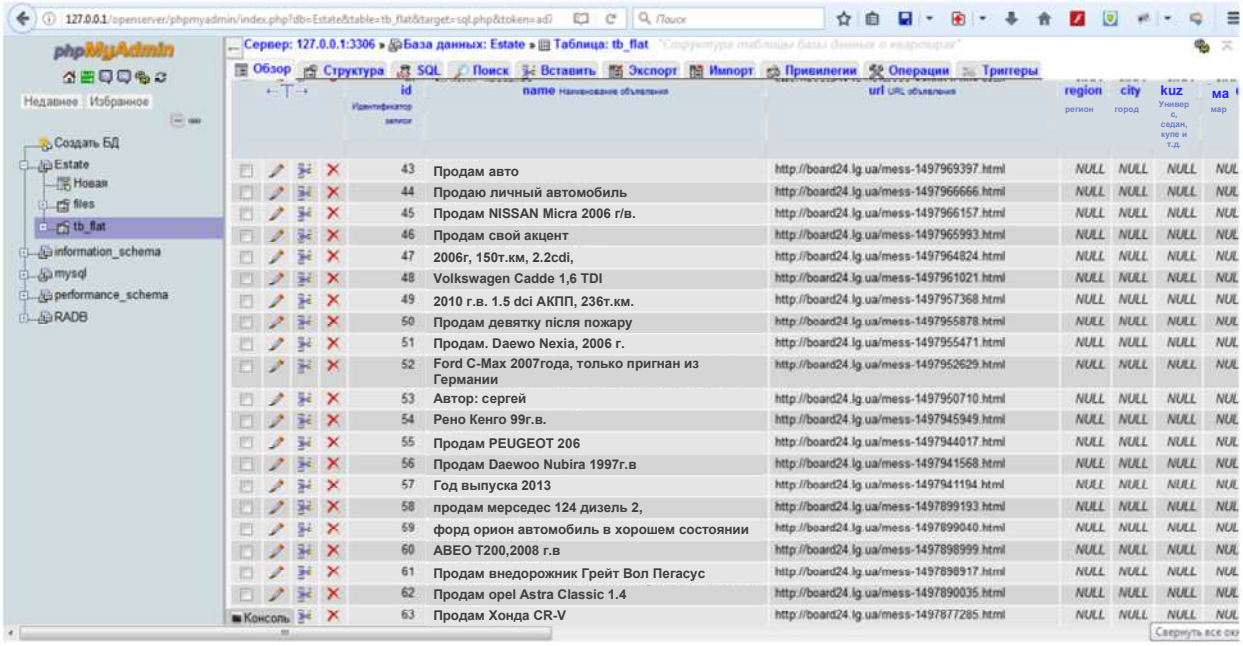


Рисунок 3.8 –Зовнішній вигляд бази даних після зберігання отриманих даних

Під час випробування було отримано інформацію щодо 2110 об’єктів які зайняли в базі даних 416 кб.

Висновки до розділу 3

Розроблено додаток для вилучення інформації щодо продажу автомобілів з різних веб сайтів. Розроблений додаток має гнучку систему конфігурації що дозволяє налаштовувати додаток на будь-який сайт. Також виділення різних алгоритмів вибірки даних в окремі класи дозволило забезпечити легку розширюваність системи. Був розроблений веб інтерфейс, який відображає знайдені об’єкти та надає інструменти пошуку.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 НПАОП 0.00-4.12-05 [98].

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі,

спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 НАПБ Б.02.005-2003[101].

Обов’язковими вимогами враховане наступне:

– не слід допускати до роботи осіб, що в установленому порядку не пройшли навчання, інструктаж та перевірку знань з охорони праці, пожежної безпеки та цих Правил.

– на підприємстві/організації, де експлуатуються ЕОМ з відео дисплейними терміналами (ВДТ) і периферійними пристроями (ПП), розробляється інструкція з охорони праці відповідно до Положення про розробку інструкцій з охорони праці [99].

– ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

– перед допуском до самостійної роботи кожен працівник має право на навчання з питань охорони праці і роботодавець зобов’язаний, і проводить таке навчання у вигляді двох інструктажів з питань охорони праці: вступного та первинного.

4.2 Аналіз стану умов праці

Робота над проектом проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп’ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	2,3
Висота, м	3,2
Площа, $м^2$	11,5
Об’єм, $м^3$	36,8

Згідно з [97] розмір площі для одного робочого місця оператора персонального комп’ютера має бути не менше 6 кв. м, а об’єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [95] (табл. 5.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷800

Приміщення кабінету знаходиться на другому поверсі п’яти поверхової будівлі і має об’єм 36,8 м3, площу – 11,5 м2. У цьому кабінеті обладнано одне місця праце та два укомплектовані ПК.

Температура в приміщенні протягом року коливається у межах 14–28°С, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система вентилювання приміщення — природна неорганізована, а опалення — автономне.

4.2.3 Навантаження та напруженість процесу праці

Під час виконання випускної роботи:

за фізичним навантаженням робота відноситься до категорії легкі роботи (Іа), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав’язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи на ведені в [95].

Роботу за дипломним проектом визнано, такою, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви - для розробників програм тривалістю 10-15 хв. через кожну годину роботи;

4.2.4 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціювання.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому

проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;

2) застосування в конструкції швидкодійних засобів захисного відключення можливих джерел запалення;

3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Згідно [9] таке приміщення, площею 36,8 м^2, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до [101] відносяться до пожежонебезпечної зони класу П-IIа. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [93].

4.2.5 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму

короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.3 Гігієнічні вимоги до параметрів виробничого середовища

4.3.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [94] і наведені в табл. 5.4:

Таблиця 4.3 – Норми мікроклімату робочої зони об’єкту

Період року	**Категорія робіт**	**Температура C^0**	**Відносна вологість %**	**Швидкість руху повітря, м/с**
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [96]. Рівні позитивних і негативних іонів у повітрі мають відповідати [96].

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

4.3.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає [94]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових – 1/10:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10}\right) \cdot S_n, \qquad (4.1)$$

де Sb – площа віконних прорізів, м2;

Sn – площа підлоги, м2.

Sn = $a \cdot b = 5 \cdot 2{,}3 = 11{,}5$ м2,

S = $1/8 \cdot 11{,}5 = 2{,}3$ м2.

Приймаємо 1 вікно площею S=3,08 м2.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \qquad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м^2; S = 11,5 м^2;

Z – поправочний коефіцієнт світильника (Z = 1,15 для ламп розжарювання та ДРЛ; Z = 1,1 для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300*11.5*1.1*1.5}{5400*0.575*2} \approx 0.96$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.3.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, що супроводжує роботу користувачів персональних комп’ютерів (зумовлений як роботою системних блоків, клавіатури, так і друкуванням на принтерах, а також зовнішніми чинниками), коливається у межах 50–65 дБА [96]. У залах опрацювання інформації та комп’ютерного набору рівні шуму не повинні перевищувати 65 дБА.

Віброізоляція можливо здійснювати за допомогою спеціальної прокладки під системний блок, який послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам [96].

4.3.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення > 40 м3 на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.3.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) Заходи безпеки під час експлуатації персонального комп’ютера та периферійних пристроїв передбачають:

– правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;

– експлуатацію сертифікованого обладнання;

– дотримання заходів електробезпеки;

– забезпечення оптимальних параметрів мікроклімату;

– забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);

– облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

– постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під’єднують до електромережі;

– постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

– не тягнути за мережевий кабель, щоб витягти вилку з розетки;

– не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

– не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

– не залишати включені електроприлади без нагляду;

– не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

– не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Вимоги безпеки при надзвичайних ситуаціях:

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток. При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво.

2) При замиканні, перевантаженні електричного струму на електричному обладнанні, внаслідок ураження грозової блискавки та ймовірної небезпеки ураженням електричним струмом, приймають наступне:

– попередження замикання здійснюється правильним вибором, монтажем експлуатації мереж;

– застосування захисту схем у вигляді швидкодіючих реле, а також вимикачів, плавких запобіжників.

Також застосовують різні електричні захисні засоби від ураження струмом:

1) Ізолюючі - ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.

2) Основні - володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою.

3) Запобіжні - володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою.

4.4 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [102], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт

використання вертикальних заземлювачів $\eta_в$ в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4…0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_д \cdot R_{пр.з.}}{R_{пр.з.} - R_д}, \qquad (4.4)$$

де $R_{пр.з.}$ – опір природних заземлювачів; $R_д$ – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.}=R_д$.

Підставивши числові значення у формулу (4.4), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ, Ом·м. Приблизне значення питомого опору глини приймаємо ρ=40 Ом•м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в}$, і горизонтальних $\rho_{розр.г}$, Ом·м за формулою:

$$\rho_{розр..} = \psi \cdot \rho, \qquad (4.5)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в}$=1,7 і горизонтальних $\rho_{розр.г}$=5,5 Ом·м.

$$\rho_{розр.в} = 1,7 \cdot 40 = 68 \text{ Ом•м}$$

$$\rho_{розр.г} = 5,5 \cdot 40 = 220 \text{ Ом•м}$$

4) Розраховується опір розтікання струму вертикального заземлювача $R_в$, Ом, за (5.5).

$$R_в = \frac{\rho_{розр.в}}{2 \cdot \pi \cdot l_в} \cdot \left(\ln \frac{2 \cdot l_в}{d_{ст}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_в}{4 \cdot t - l_в} \right), \quad (4.6)$$

де $l_в$ – довжина вертикального заземлювача (для труб - 2−3 м; $l_в$=3 м);

$d_{ст}$ – діаметр стержня (для труб - 0,03−0,05 м; $d_{ст}$=0,05 м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.7):

$$t = h_в + \frac{l_в}{2}, \quad (4.7)$$

де $h_в$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

Підставивши числові значення у формулу (4,6) отримуємо:

$$R_в = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання $\eta_в$:

$$n = \frac{2 \cdot R_в}{R_д} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.8)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_в$ =0,57 (табличне значення).

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання $n_в$, шт:

$$n_в = \frac{2 \cdot R_в}{R_д \cdot \eta_в} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \qquad (4.9)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача $l_с$, м:

$$l_с = 1,05 \cdot L_в \cdot (n_в - 1), \qquad (4.10)$$

де $L_в$ – відстань між вертикальними заземлювачами, (прийняти за $L_в$ = 3м);

$n_в$ – необхідна кількість вертикальних заземлювачів.

$$l_с = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) $R_г$, Ом:

$$R_г = \frac{\rho_{розр.г}}{2 \cdot \pi \cdot l_с} \cdot \ln \frac{2 \cdot l_с^2}{d_{см} \cdot h_г}, \qquad (4.11)$$

де $d_{см}$ – еквівалентний діаметр смуги шириною b, $d_{см}$ =0,95b, b = 0,15 м;

$h_г$ – глибина закладання горизантальних заземлювачів (0,5 м);

$l_с$ - довжина з'єднувальної стрічки горизонтального заземлювача $l_с$, м

$$R_г = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача $\eta_{с.}$ відповідно до необхідної кількості вертикальних заземлювачів $n_в$.

Коефіцієнт використання з'єднувальної смуги η_c=0,3 (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = \frac{R_в \cdot R_г}{R_в \cdot \eta_c + R_г \cdot n_в \cdot \eta_в} \le R_д. \qquad (4.12)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: Rзаг <4 Ом, а саме:

$$R_{заг} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \le R_д$$

При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

– іскри і дуги коротких замикань;

– перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;

– іскри при розмиканні і розмиканні ланцюгів;

– розряди статичної електрики;

– необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

4.5 Охорона навколишнього природного середовища

Виконання дипломної роботи у купі з іншими факторами впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення

і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на знешкодження, утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря робочої зони.

В процесі діяльності аналізу математичних методів оцінки надійності БСМ і програмних продуктів для імітаційного моделювання БСМ, вибіру найбільш підходящої системи для оцінки працездатності БСМ та оцінки впливу перешкод і потужності передачі радіосигналу на працездатність БСМ.виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки
- батарейки та акумулятори (малі) -III клас небезпеки
- акумулятор для джерел безперебійного живлення -III клас небезпеки
- змінні носії інформації - IV клас небезпеки
- відходи друкуючих пристроїв - IV клас небезпеки
- макулатура - IV клас небезпеки
- матеріали пакувальні пластмасові забруднені (ємності з-під тонеру, фарби, інш.) - IV клас небезпеки
- побутові відходи - IV клас небезпеки

Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над

запропонованим проектом, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп’ютері.

ВИСНОВКИ

Метою даної роботи було дослідження методів вилучення вмісту веб-сторінок, а також розробка системи пошуку об’єктів автомобільного ринку.

Для досягнення цієї мети було проведено дослідження основних методів інтелектуального аналізу веб-контенту, описаних в літературі. Також розглянуті підходи до вилучення даних в існуючих розробках і виконано порівняння цих підходів за такими критеріями як: рівень автоматизації, рівень аналізу даних, використані методи. Представлена класифікація існуючих систем за рівнем взаємодії з користувачем. В ході дослідження були зроблені висновки про ефективність використання певних методів в залежності від типу аналізованих даних. Наприклад, для аналізу великої кількості однотипних сторінок, без аналізу вільного тексту ефективно застосовувати неконтрольовані системи володіють високим ступенем автоматизації, а для аналізу вільного тексту краще підходять контрольовані і ручні системи.

На основі проведеного аналізу було зпроектовано і реалізовано систему вилучення даних про продаж автомобілів з веб сторінок. Система була розроблена на мові php, для зберігання даних використовується реляційна БД MySQL.

Розроблена система використовується в практичній діяльності суб’єкта оціночної діяльності – ТОВ «Твоє авто» (м.Попасне).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) Web Mining: Machine learning for Web Applications, p. 291 // Cronin B. Annual Review of Information Science and Technology. — ARIST, 2004. — 674 c. — ISBN 1573872091.

2) Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth and Ramasamy Uthurusamy. Advances in Knowledge Discovery and Data Mining,, (Chapter 1) AAAI/MIT Press 1996

3) http://www.kdnuggets.com/gpspubs/piatetsky-interview-computerra.pdf

4) http://www.lingvoda.ru/forum/actualthread.aspx?bid=10&tid=3001&hl=data+mining

5) Parsaye K. A Characterization of Data Mining Technologies and Processes. The Journal of Data Warehousing. 1998.№ 1

6) Kosala R., Blockeel H. Web Mining Research: A Survey. — ACM SIGKDD, 2000. — P. 2-3.

7) Sivaramakrishnan J., Balakrishnan V. Web Mining Functions in an Academic Search Application. — Dubai: BITS – PILANI, 2009. — P. 132.

8) Srivastava J., Desikan P., Kumar V. Web Mining — Concepts, Applications, and Research Directions. — 2004., p. 3.

9) Bing L. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. — Springer, 2011. — 642 c. — ISBN 978-3642194597, p. 527

10) Buitelaar, P. Ontology learning and population: bridging the gap between text and knowledge / P. Buitelaar, P. Cimiano. — 167 edition. — Ios Pr Inc, 2008. — P. 273.

11) http://www.google.com/patents/US6871140

12) Cimiano, Philipp. Towards the self-annotating web / Philipp Cimiano, Siegfried Handschuh, Steffen Staab // Proceedings of the 13th conference on World Wide Web WWW 04. — 2004. — Vol. 462-471. — P. 462.

13) Cimiano, Philipp. Gimme’the context: context-driven automatic semantic annotation with C- PANKOW / Philipp Cimiano, G Ladwig, Steffen Staab // Assessment. — 2005. — Pp. 332-341.

14) Freitag D. Boosted wrapper induction // D. Freitag, N. Kushmerick // Proc. of 17-th National Conf. on Artificial Intelligence. — 2000. — P. 577–583.

15) Grishman R., Sundheim B. Message Understanding Conference — 6: A Brief History // Proceedings of the 16th International Conference on Computational Linguistics (COLING). I. Kopenhagen, 1996. P.466–471.

16) Riloff, E., Automatically constructing a dictionary for information extraction tasks. Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), pp. 811-816, AAAI Press/The MIT Press, 1993.

17) Huffman, S., Learning information extraction patterns from ex¬amples. Connectionist, statistical, and symbolic Approaches to Learning for Natural Language Processing, Springer-Verlag, 1996.

18) Kim, J. and Moldovan, D., Acquisition of linguistic patterns for knowledge-based information extraction. IEEE Transactions on Knowledge and Data Engineering 7(5): 713-724, 1995.

19) Krupka, G., Description of the SRA system as used for MUC-6. Proceedings of the sixth Message Understanding Conference (MUC-6), pp. 221-235, 1995.

20) Soderland, S., Fisher, D., Aseltine, J., and Lehnert, W., CRYSTAL: Inducing a conceptual dictionary. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI), 1995.

21) Soderland, S., Learning information extraction rules for semi-structured and free text. Journal of Machine Learning, 34(1- 3): 233-272, 1999.

22) Califf, M. and Mooney, R., Relational learning of pattern-match rules for information extraction. Proceedings of AAAI Spring Symposium on Applying Machine Learning to Discourse Process¬ing Stanford, California, March, 1998.

23) Freitag, D., Information extraction from HTML: Application of a general learning approach. Proceedings of the Fifteenth Confer¬ence on Artificial Intelligence (AAAI-98).

24) Kushmerick, N., Weld, D., and Doorenbos, R., Wrapper induction for information extraction. Proceedings of the Fifteenth Interna¬tional Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.

25) Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. Journal of Informa¬tion Systems 23(8): 521-538, 1998.

26) Muslea, I., Minton, S., and Knoblock, C., A hierarchical approach to wrapper induction. Proceedings of the Third International Conference on Autonomous Agents (AA-99), 1999.

27) Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. Journal of Informa¬tion Systems 23(8): 521-538, 1998.

28) Chang, C.-H., Hsu, C.-N., and Lui, S.-C. Automatic information extraction from semi-Structured Web Pages by pattern discovery. Decision Support Systems Journal, 35(1): 129-147, 2003.

29) http://www.isi.edu:80/info-agents/RISE/index.html

30) Muslea, I., Minton, S., and Knoblock, C., A hierarchical approach to wrapper induction. Proceedings of the Third International Conference on Autonomous Agents (AA-99), 1999.

31) Kushmerick. N., Adaptive Information Extraction: Core technolo¬gies for Information agents. In Intelligent Information Agents R&D in Europe: An AgentLink perspective (Klusch, Bergamaschi, Edwards & Petta, eds.). Lecture Notes in Computer Science 2586, Springer, 2003.

32) Soderland, S., Learning to extract text-based information from the world wide web. Proceedings of the third International Confer¬ence on Knowledge Discovery and Data Mining (KDD), pp. 251¬254, 1997.

33) Ciravegna, F., Learning to tag for information extraction from text. Proceedings of the ECAI-2000 Workshop on Machine Learning for Information Extraction, Berlin, August 2000.

34) Laender, A. H. F., Ribeiro-Neto, B., DA Silva and Teixeira, A brief survey of Web data extraction tools. SIGMOD Record 31(2): 84-93, 2002.

35) Crescenzi, V., and Mecca, G., Grammars have exceptions. Infor¬mation Systems, 23(8): 539-565, 1998.

36) Hammer, J., McHugh, J. and Garcia-Molina, Semistructured data: the TSIMMIS experience. In Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems (ADBIS), St. Petersburg, Rusia, pp. 1-8, 1997.

37) Arocena, G. O. and Mendelzon, A. O., WebOQL: Restructuring documents, databases, and Webs. Proceedings of the 14th IEEE International Conference on Data Engineering (ICDE), Orlando, Florida, pp. 24-33, 1998.

38) Saiiuguet, A. and Azavant, F., Building intelligent Web applica¬tions using lightweight wrappers. Data and Knowledge Engineer¬ing 36(3): 283-316, 2001.

39) Liu, L., Pu, C., and Han, W. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources, Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE), San Diego, California, pp. 611-621, 2000.

40) Crescenzi, V., Mecca, G. and Merialdo, P., RoadRunner: towards-automatic data extraction from large Web sites. Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109-118, 2001.

41) Adelberg, B., NoDoSE: A tool for semi-automatically extracting structured and semi-structured data from text documents. SIG- MOD Record 27(2): 283-294, 1998.

42) Laender, A. H. F., Ribeiro-Neto, B. and DA Silva, A., S., DEByE - Data Extraction by Example. Data and Knowledge Engineering, 40(2): 121-154, 2002.

43) Ribeiro-Neto, B., A., Laender, A., H., F. and DA Silva, A., S., Ex¬tracting semi-structured data through examples. Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM), Kansas City, Missouri, pp. 94¬101, 1999.

44) Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Kai Ng, Y., Quass, D. and Smith, R. D., Conceptual-model-based data ex¬traction from multiple-record Web pages. Data and Knowledge Engineering, 31(3): 227-251, 1999.

45) Sarawagi, S., Automation in information extraction and integra¬tion, Tutorial of The 28th International Conference on Very Large Data Bases (VLDB), 2002.

46) Kuhlins, S and Tredwell, R. Toolkits for generating wrappers, Net.ObjectDays 2002: Objects, Components, Architectures, Ser¬vices and Applications for a Networked World, http://www.netobjectdays.org/, LNCS 2591, 2002.

47) Chang C.-H., Kayed M., Girgis M.R., Shaalan K.. A Survey of Web Information Extraction Systems. IEEE Transactions on Knowledge and Data Engineering archive. Volume 18 Issue 10, October 2006, p. 1411-1428

48) Soderland, S., Learning to extract text-based information from the world wide web. Proceedings of the third International Confer¬ence on Knowledge Discovery and Data Mining (KDD), pp. 251¬254, 1997.

49) Elmasri, R. and Navathe, S. B. Fundamentals of Database Sys¬tems, 4th Ed. Addison Wesley, 2003.

50) Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. Journal of Informa¬tion Systems 23(8): 521-538, 1998.

51) Регулярные выражения. - [Електроний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/Регулярные_выражения

52) Document Object Model. - [Електроний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/Document_Object_Model

53) XPath. - [Електронний ресурс] - Режим доступу: https://ru.wikipedia.org/wiki/XPath

54) Chang C.-H., Kayed M., Girgis M.R., Shaalan K.. A Survey of Web Information Extraction Systems. IEEE Transactions on Knowledge and Data Engineering archive. Volume 18 Issue 10, October 2006, p. 1411-1428

55) Hammer, J., McHugh, J. and Garcia-Molina, Semistructured data: the TSIMMIS experience. In Proceedings of the 1st East-European Symposium on Advances in Databases and Information Systems (ADBIS), St. Petersburg, Rusia, pp. 1-8, 1997.

56) Laender, A. H. F., Ribeiro-Neto, B., DA Silva and Teixeira, A brief survey of Web data extraction tools. SIGMOD Record 31(2): 84-93, 2002.

57) Crescenzi, V., and Mecca, G., Grammars have exceptions. Infor¬mation Systems, 23(8): 539-565, 1998.

58) Arocena, G. O. and Mendelzon, A. O., WebOQL: Restructuring documents, databases, and Webs. Proceedings of the 14th IEEE International Conference on Data Engineering (ICDE), Orlando, Florida, pp. 24-33, 1998.

59) Saiiuguet, A. and Azavant, F., Building intelligent Web applica¬tions using lightweight wrappers. Data and Knowledge Engineer¬ing 36(3): 283-316, 2001.

60) Liu, L., Pu, C., and Han, W. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources, Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE), San Diego, California, pp. 611-621, 2000.

61) Crescenzi, V., and Mecca, G., Grammars have exceptions. Infor¬mation Systems, 23(8): 539-565, 1998.

62) Freitag, D., Information extraction from HTML: Application of a general learning approach. Proceedings of the Fifteenth Confer¬ence on Artificial Intelligence (AAAI-98).

63) Arocena, G. O. and Mendelzon, A. O., WebOQL: Restructuring documents, databases, and Webs. Proceedings of the 14th IEEE International Conference on Data Engineering (ICDE), Orlando, Florida, pp. 24-33, 1998.

64) Califf, M. and Mooney, R., Relational learning of pattern-match rules for information extraction. Proceedings of AAAI Spring Symposium on Applying Machine Learning to Discourse Process¬ing Stanford, California, March, 1998.

65) Kushmerick, N., Weld, D., and Doorenbos, R., Wrapper induction for information extraction. Proceedings of the Fifteenth Interna¬tional Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.

66) Hammer, J., McHugh, J. and Garcia-Molina, Semistructured data: the TSIMMIS experience. In Proceedings of the 1st East-European Symposium on

Advances in Databases and Information Systems (ADBIS), St. Petersburg, Rusia, pp. 1-8, 1997.

67) Soderland, S., Learning information extraction rules for semi-structured and free text. Journal of Machine Learning, 34(1- 3): 233-272, 1999.

68) Soderland, S., Learning Information Extraction Rules for Semi-Structured and Free Text, Machine Learning (1999), Volume 34, Issue 1, pp 233–272.

69) Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. Journal of Informa¬tion Systems 23(8): 521-538, 1998.

70) Muslea, I., Minton, S., and Knoblock, C., A hierarchical approach to wrapper induction. Proceedings of the Third International Conference on Autonomous Agents (AA-99), 1999.

71) Hsu, C.-N. and Chang, C.-C. Finite-State Transducers for Semi¬Structured Text Mining. In Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, Stock¬holm, Sweden, 1999. Page 38-49.

72) Adelberg, B., NoDoSE: A tool for semi-automatically extracting structured and semi-structured data from text documents. SIG- MOD Record 27(2): 283-294, 1998.

73) Ribeiro-Neto, B., A., Laender, A., H., F. and DA Silva, A., S., Ex¬tracting semi-structured data through examples. Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM), Kansas City, Missouri, pp. 94¬101, 1999.

74) Chang, C.-H. and Lui, S.-C., IEPAD: Information extraction based on pattern discovery. Proceedings of the Tenth International Con¬ference on World Wide Web (WWW), Hong-Kong, pp. 223-231, 2001.

75) Saiiuguet, A. and Azavant, F., Building intelligent Web applica¬tions using lightweight wrappers. Data and Knowledge Engineer¬ing 36(3): 283-316, 2001.

76) Liu, L., Pu, C., and Han, W. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources, Proceedings of the 16th IEEE

International Conference on Data Engineering (ICDE), San Diego, California, pp. 611-621, 2000.

77) Hogue, A. and Karger, D. Thresher: Automating the Unwrapping of Semantic Content from the World Wide. Proceedings of the 14th International Conference on World Wide Web (WWW), Ja¬pan, pp. 86-95, 2005.

78) Yang, G., Ramakrishnan, I. V. and Kifer, M. On the complexity of schema inference from Web pages in the presence of nullable data attributes, Proceedings of the 12th ACM International Conference on Information and Knowledge Management (CIKM), pp. 224¬231, 2003.

79) Wang, J. and Lochovsky, F. H., Wrapper induction based on nested pattern discovery. Technical Report HKUST-CS-27-02, Dept. of Computer Science, Hong Kong, U. of Science & Technol¬ogy, 2002.

80) Wang, J. and Lochovsky, F. H., Data extraction and label assign¬ment for Web databases, Proceedings of the Twelfth International Conference on World Wide Web (WWW), Budapest, Hungary, pp. 187-196, 2003.

81) Crescenzi, V., Mecca, G. and Merialdo, P., RoadRunner: towards-automatic data extraction from large Web sites. Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109-118, 2001.

82) Adelberg, B., NoDoSE: A tool for semi-automatically extracting structured and semi-structured data from text documents. SIG- MOD Record 27(2): 283-294, 1998.

83) Arasu, A. and Garcia-Molina, H., Extracting structured data from Web pages. Proceedings of the ACM SIGMOD International Con¬ference on Management of Data, San Diego, California, pp. 337¬348, 2003.

84) Liu, B., Grossman, R. and Zhai, Y., Mining data records in Web pages. KDD, 601-606, 2003.

85) Crescenzi, V., Mecca, G. and Merialdo, P., RoadRunner: towards-automatic data extraction from large Web sites. Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109-118, 2001.

86) Zhai, Y. and Liu, B. Web Data Extraction Based on Partial Tree Alignment. Proceedings of the 14th International Conference on World Wide Web (WWW), Japan, pp. 76-85, 2005.

87) Liu, B. and Zhai, Y., NET - A System for Extracting Web Data from Flat and Nested Data Records. WISE 2005, 487-495, 2005.

88) Lan Yi, Bing Liu, and Xiaoli Li. "Eliminating Noisy Information in Web Pages for Data Mining." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Min¬ing (KDD-2003), Washington, DC, USA, August 24 - 27, 2003.

89) Zhao, H., Meng, W., Wu, Z., Raghavan, V., and Yu, C. Fully Automatic Wrapper Generation For Search Engines. Proceedings of the 14th International Conference on World Wide Web (WWW), Japan, pp. 66-75, 2005.

90) Lerman, K., Getoor, L., Minton, S. and Knoblock, C. A., Using the structure of Web sites for automatic segmentation of tables. SIG- MOD Conference, 119-130, 2004.

91) Pinto, D., McCallum, A., Wei, X. and Croft, B. C., Table extraction using conditional random fields. SIGIR, 235-242, 2003.

92) Hsu, C.-N., Chang, C.-H., Hsieh, C.-H., Lu, J.-J. and Chang, C.-C. Reconfigurable Web Wrapper Agents for Biological Information Integration, JASIST (SCI expanded), Special Issue on Bioinformat¬ics, Vol. 56, No. 5, pp. 505--517, 2005.

93) ГОСТ 12.1.044-89 ССБТ. Пожежовибухонебезпека речовин і матеріалів. Номенклатура показників і методи їх визначенняя. Постанова від 12.12.1989 № 3683 МВС СРСР. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=51048

94) ДБН В.2.5-28:2018 Природнє і штучне освітлення. Режим доступу: www. URL: https://dbn.co.ua/load/normativy/dbn/dbn_v_2_5_28/1-1-0-1188

95) ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Постанова №14 від 14.07.1999. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/v0014410-96

96) ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. Постанова N 37 від 01.12.99. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va037282-99

97) ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. Постанова N 42 від 01.12.99. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99

98) НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці. Наказ №15 від 26.01.05. Режим доступу: www. URL:https://dnaop.com/html/33829/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_0.00-4.12-05

99) НПАОП 0.00-4.15-98 Про розробку інструкцій з охорони праці. Наказ №9 від 29.01.98 року. Режим доступу: www. URL: https://dnaop.com/html/64/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_0.00-4.15-98

100) НПАОП 0.00-6.03-93 Порядок опрацювання та затвердження власником нормативних актів про охорону праці. Наказ №132 від 21.12.93. Режим доступу: www. URL: https://dnaop.com/html/32357/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_0.00-6.03-93

101) ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. Наказ від 15.06.2016 № 158. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419

102) НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок. Наказ №257 від 06.10.97. Режим доступу: www. URL: https://dnaop.com/html/1691/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_40.1-1.01-97

103) НПАОП 40.1-1.32-01 Правила устройства электроустановок. Электрооборудование специальных установок. Наказ №272 від 21.06.2001. Режим доступу: www. URL: https://dnaop.com/html/1692/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_40.1-1.32-01

104) ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної вібрації. Постанова №39 від 01.01.99 МОЗ України. Режим доступу: www. URL: https://zakon.rada.gov.ua/rada/show/va039282-99

105) ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування. Режим доступу: www. URL: https://dbn.co.ua/load/normativy/dbn/1-1-0-1018

106) ГОСТ 12.1.018-93 ССБТ.Пожаровзрывобезопасность статического электричества. Общие требования. Дата прийняття 21.10 93. Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48681

107) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров`я працівників під час роботи з екранними пристроями. Наказ від 14.02.2018 № 207 Режим доступу: www. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=77160

ДОДАТОК А.
Огляд патенів

UA № 90764, дата Подання 13.05.2008, Опубліковано 25.05.2010 бюл. № 10 2010 р. «Спосіб пошуку інформаційних об'єктів та система для його здійснення», згідно з яким приймають запит, здійснюють збір інформації, пошук по інформаційному сховищу на предмет об'єктів, що відповідають запиту користувача та, згідно з винаходом, включає визначення елементів уточнення на основі щонайменш одного з рейтингом елементів уточнення та інформації про елементи уточнення, після чого обирають елементи уточнення, включаються вибрані елементи в запит, остаточно редагують та підтверджують запит, визначаються еквіваленті дескрипторів та елементів уточнення, отримуються від користувача команду на підключення інтеграції запиту, обробляють пошуковою системою запит шляхом вибору документів або текстів із сховище даних та здійснюють виведення результатів.

Недоліками запропонованого способу є недостатня точність проведення пошукових робіт та відносно довгий час обробки інформації.

UA 60952 U. Опубліковано: 25.06.2011. «Спосіб пошуку інформації в масиві текстів»

Задачею корисної моделі є створення способу пошуку інформації в масивах текстів в базі даних сховища даних, який би розширював функціональні можливості та підвищував техніко-експлуатаційні характеристики пошукових робіт, підвищував швидкість, актуальність та чистоту пошуку. Також важливим питанням, що вирішує запропонований спосіб є визначення релевантності для результатів пошуку.

UA 80549 C2 «Спосіб пошуку та ідентифікації інформації з електронних баз даних»

Спосіб пошуку інформації в електронних систематизованих базах даних, які розміщені в доступних каталогах та файлах та містить в банку даних

ототожнюючу та інформаційну частину даних, які відтворюють, зчитують і зіставляють з ототожнюючими частинами даних та наносять на носії інформації, які містять інформаційні та додаткові частини даних, який відрізняється тим, що інформаційну частину даних розміщують на екрані дисплея шляхом створення таблиці, яка містить стовпчики та рядки, а потім точки їх перетину переміщують в будь-який сегмент таблиці та проводять запит відносно додаткової частини інформації на інформаційному підрівні, причому всі запити виносять в верхню частину таблиці, після чого проводять ідентифікацію інформаційних документів та зчитують інформаційну частину даних згідно з точкою перетину координат стовпчиків та рядків.

Переваги способу.

Існуючі пошукові системи і роботи надають тільки посилання на сайти, де можна пошукати потрібну інформацію, а не саму інформацію. Інформацію доводиться людині збирати самостійно, гуляючи по всіх запропонованих сайтів та Інтернет сторінок. Причому, коли він заходить на запропоновані сторінки, то не знаходить там стандартизованої інформації, що знаходиться в певному місці сайту. Це уповільнює роботу з пошуку, копіювання інформації з сайту до себе в комп'ютер і аналізу зібраної інформації.

RU 2 442 214 C2 «Семантична навігація по веб-контенту і колекціям документів»

Винахід відноситься до семантичної навігації по безлічі документів.

Технічним результатом є розширення функціональних можливостей семантичної навігації по Веб-контенту і колекціям документів за рахунок семантичної розмітки документів.

Даний винахід забезпечує спосіб і пристрій, включаючи комп'ютерні програмні продукти. Спосіб розмітки безлічі електронних документів містить семантичну розмітку електронних документів відповідно до визначеної моделлю предметної області, представленої у формі предметної онтології, таким чином створюючи результати розмітки, представлені в форматі Мови Онтологій Веб; і збереження результатів розмітки і посилань на розмічені

електронні документи, представлених Універсальними Ресурсними Покажчиками, в сховище Середовища

RU 2473967 "Система оцінки благонадійності індивіда-споживача фінансових послуг"

Винахід відноситься до засобів автоматизації фінансової та банківської діяльності. Технічним результатом є підвищення надійності оцінки благонадійності індивіда-споживача фінансових послуг за рахунок побудови психологічного і соціального портрета індивіда. Система оцінки благонадійності індивіда-споживача фінансових послуг містить сукупність взаємодіючих між собою засобів введення даних, аналізу введених даних, збору інформації в соціальних мережах, збору даних про мережеві взаємодії, опитування контрагентів індивіда по соціальним мережам, обчислення чисельно виражається і нормованої оцінки благонадійності споживача фінансових послуг .

Система оцінки благонадійності індивіда на основі аналізу даних соціальних мереж, описана в патентній заявці US 2009327054, опублікованій 31.12.2009, і придатна для використання в оцінці благонадійності споживача фінансових послуг. Відома з US 2009327054 система оцінки благонадійності індивіда включає сукупність об'єднаних в інформаційну мережу і взаємодіючих між собою засобів збору і обробки даних, що представляють собою в граничних випадках або ділянки пам'яті робочої станції (сервера), або окремі робочі станції (сервера). Засоби збору і обробки даних є компонент введення даних індивідом - передбачуваним споживачем фінансових послуг - компонент опитування індивіда, в тому числі отримання інформації про участь в соціальних мережах; компонент аналізу введених даних; компонент збору даних про особистість - збору інформації в соціальних мережах, учасником яких є індивід, що включає засоби взаємодії з, щонайменше, однієї соціальною мережею; компонент збору даних про мережеві взаємодії - взаємодії індивіда в соціальних мережах, що включає засоби взаємодії з, щонайменше, однієї соціальною мережею; компонент аналізу мережевих взаємодій між індивідами-

учасниками соціальних мереж і оцінки цих взаємодій; компонент опитування учасників соціальних мереж - контрагентів індивіда по соціальним мережам, що включає засоби взаємодії з, щонайменше, однієї соціальною мережею; компонент обчислення чисельно виражається і нормованої оцінки благонадійності індивіда - передбачуваного споживача фінансових послуг - соціального рейтингу індивіда. Також кошти збору і обробки даних включають базу даних індивідів, що взаємодіють з системою, інформація про яких була оброблена системою, і результатів, сформованих системою щодо цих індивідів, з інтерфейсом оператора бази даних. Відома система не володіє достатньою надійністю (відсутній об'єктивний і незалежний від «голосування» в соціальній мережі аналіз особистості споживача і його соціального оточення, ненадійність результатів такого «голосування»), крім того, при роботі відомої системи не повністю використовуються для користувача засоби, призначені для роботи з соціальними мережами.

RU0002565473 «Метод побудови корпусу текстів на основі інтернет-форумів»

Система містить блок збору інформації в мережі Інтернет, що забезпечує збір інформації в Інтернет, в окремому випадку, за допомогою Пошукових систем загального призначення, а також, при необхідності, у внутрішній базі даних клієнта системи (наприклад, агентства з підбору персоналу), управляє зберіганням завантажених текстових полів бази даних і документів та формує чергу ідентифікаторів (універсальних покажчиків ресурсів або URL), необхідних для отримання профілів кандидатів з мережевих ресурсів. Блок також може опитувати Внутрішню Базу Даних користувача з метою отримання інформації, що відноситься до вже внесених до Внутрішньої Базу Даних кандидатах, для визначення корисності зібраної на мережевих ресурсах інформації. При цьому вже наявна у Внутрішній Базі Даних інформація може оновлюватися (або відсіюватися) за допомогою наявної на мережевих ресурсах інформації, якщо вона відповідає кандидату, вказаною на мережевому ресурсі, або може бути використана для подальшої перевірки достовірності даних, якщо

унікальна інформація відноситься до різних профілів кандидата. Також зібрана інформація може використовуватися для уточнення зберігаються в Базі Даних відомостей. Отриманий з мережі документ (мережевий профіль користувача, текст наукової статті, опублікований кандидатом, повідомлення на форумі професійного спілкування та ін.) Може бути збережений в Базі Даних Документів або в подальшому переданий разом з вмістом відповіді на запит Блоку аналізу інформації - "парсером", здійснює витяг корисної інформації.

EA 200501304 A1 PCT / US2004 / 004674 WO 2004/075466 2004.09.02 "Система для пошуку, управління, збору, виявлення, доставки та подання семантичного знань"

Даний винахід направлено на інтегровану концептуальну структуру реалізації та результуючий носій для пошуку, управління, збору, виявлення, доставки та подання знань. Система відповідальна за підтримку в робочому стані семантичної інформації.

EA201300375 «Спосіб організації пошукової бази даних з використанням нечітких критеріїв»

Винахід відноситься до систем і методів створення корпусів текстів для різних дослідницьких і інших цілей. Технічним результатом є підвищення точності відділення текстів користувачів від решти контенту веб-сторінок при автоматичному побудові корпусу текстів. У способі побудови корпусу текстів на основі інтернет-форумів для комп'ютерної системи будують об'єктну модель документа у вигляді дерева DOM-структури даних. Виділяють групу однотипних вершин в дереві об'єктної моделі документа. Видаляють необов'язкові елементи оформлення зі сторінок. Здійснюють злиття нелістових вершин з однаковими іменами в дереві об'єктної моделі та об'єднання листових вершин з однаковими властивостями. Виконують оцінку вершин і фільтрації груп. Будують вираження XPATH і застосовують отримані вирази XPATH до набору файлів, що містять всі документи з обраного форуму. 3 н. і 7 з.п. ф-ли, 3 мул.

ЕА 200501304 А1, заявка PCT / US2004 / 004674, заявка WO 2004/075466 2004.09.02 "Система для пошуку, управління, збору, виявлення, доставки та подання семантичного знань "

Даний винахід направлено на інтегровану концептуальну структуру реалізації та результуючий носій для пошуку, управління, збору, виявлення, доставки та подання знань. Система відповідальна за підтримку в робочому стані семантичної інформації.

ЕА 003114, дата публікації патенту 2003.02.27 "Спосіб публікації, пошуку, збору та обміну інформацією в глобальних і локальних мережах"

Спосіб обміну інформацією, представленою в глобальних і / або локальних мережах, що включає пошук і / або збір інформації пошуковим агентом в глобальних і / або локальних мережах відповідно до запиту споживача інформації;

Відомий також близький спосіб обміну інформацією (патент США 6 038 668), що включає пошук інформації пошуковим агентом або роботом, систематизацію інформації, зберігання посилань і додаткової інформації, що включає індекси класифікації та інші атрибути в каталогах, а також подальший відбір за запитами відповідних посилань.

Проте, цей спосіб теж не дає змоги вчасно відслідковувати всі зміни інформації, а також вимагає звернення до повного сайту для отримання детальної інформації та відповідно має перераховані вище недоліки

ЕА 008675В1 20070629 200400068 А1 «Система і спосіб пошуку, управління, доставки та подання знань»

Даний винахід направлено на реалізовану інтегрованим чином базову структуру і одержувану в результаті середу для вилучення, управління, доставки та подання знань. Система містить перший серверний компонент, призначений для додавання і підтримки специфічної для предметної області семантичної інформації, і другий серверний компонент, який містить семантику

та інші знання для використання першим серверним компонентом, які працюють спільно для забезпечення послуг добування інформації, яка залежить від контексту і від часу, для клієнтів, які керують платформою уявлення через комунікаційне середовище. В системі всі об'єкти і події в заданій ієрархії є активними агентами, семантично пов'язаними один з одним і представляють запити (утворені з лежачого в їх основі коду дій), які повертають об'єкти даних для подання клієнту відповідно до попередньо визначеної та замовною темою або «Поверхнею ». Ця система забезпечує різні засоби для клієнта, щоб налаштовувати і сполучати «Агентів» і відповідні пов'язані з ним запити для оптимізації уявлення одержуваної в результаті інформації.

Сучасна мережа Web не має підтримки для орієнтованого на користувача агрегування інформації. Користувач може тільки здійснювати доступ до одного сайту або до одного пошукового сервера одночасно, в контексті одного сеансу перегляду ресурсів мережі. Навіть якщо є контекстна або з тимчасовою залежністю інформація в інших джерелах інформації, яка відноситься до переглядається користувачем інформації, такі джерела не можуть бути представлені цілісним чином в поточному контексті завдання користувача.

Семантична мережа Web також не забезпечує орієнтованого на користувача агрегування інформації. Саме середовище є розширенням сучаснішою мережі Web. Як такі, користувачі будуть як і раніше здійснювати доступ до одного сайту або одному пошукового сервера в кожен даний момент часу і не будуть мати можливість агрегувати інформацію з інформаційних сховищ контекстним методом або з використанням тимчасової залежності.

З огляду на зростаючу потребу в отриманні "знань на кінчиках пальців", а також недоліки сучасної мережі Web і концептуальної семантичної мережі Web, багато з яких відзначені вище, існує необхідність в новій і всеосяжної системи і способі пошуку, управління і доставки знань.

У кращому варіанті здійснення користувач клієнта може створювати і редагувати інтелектуальних агентів за допомогою майстра «створити інтелектуального клієнта», що дозволяє йому переглядати ресурси семантичної середовища через діалог «відкрити агента» і додавати зв'язку з певних агентств.

Заявка PCT: EP 2007/054900 (21.05.2007) «Семантична навігація по веб-контенту і колекціям документів»

Винахід відноситься до семантичної навігації по безлічі документів. Технічним результатом є розширення функціональних можливостей семантичної навігації по Веб- контенту і колекціям документів за рахунок семантичної розмітки документів. Даний винахід забезпечує спосіб і пристрій, включаючи комп'ютерні програмні продукти. Спосіб розмітки безлічі електронних документів містить семантичну розмітку електронних документів відповідно до визначеної моделлю предметної області, представленої у формі предметної онтології, таким чином створюючи результати розмітки, представлені в форматі мови онтологій веб; і збереження результатів розмітки і посилань на розмічені електронні документи, представлених універсальними ресурсними покажчиками, в сховище середовища описи ресурсів.

Винахід стосується семантичної навігації по безлічі документів, а саме до способу і пристрою, включаючи комп'ютерні програмні продукти, для семантичної навігації по Веб- контенту і колекціям документів на основі результатів, отриманих за допомогою блоку добування інформації з (Веб) документів з області інтересів користувача.

Для забезпечення навігації по Веб-сторінок в патентній заявці США № US 6,862,710 B1, 1.02.2005, «Використання гнучких гіперпосилань для Інтернет -навігація» представлено спосіб для Інтернет-навігації, заснований на використанні «гнучких» гіперпосилань, які створюються «на льоту» за допомогою представленої системи використання «в тіні» пошукової машини для пошуку списку релевантних термінів і представлення перших чотирьох термінів, які були повернуті пошуковою машиною, в спеціальній компас-подібної навігаційній карті, дозволяючи користувачеві рух від поточного документа до одного документу зі списків документів, релеван них потреб користувача.

Крім того, для підтримки процесу навігації в патентній заявці США No. US 2006/0282409 A1, Листопад 14, 2006, «Автоматизована World Wide Web

Навігація і Витяг Контенту » представлено спосіб для вилучення контенту і перетворення результатів в спеціальну логічну модель, яка може бути використана для автоматичної генерації гіперпосилань між певними фрагментами однієї сторінки і / або між різними сторінками без втручання користувача.

З іншого боку, для забезпечення тематично орієнтованої Інтернет - навігація в патентній заявці США No. US 2006/0129549 A1, 15.06.2006, «Тематично-сфокусована Веб Навігація» представлені спосіб і пристрій, включаючи комп'ютерні програмні продукти, для тематично-сфокусовану Веб-навігацію, яка підтримується плагіном браузера, який має можливості пошуку списку відповідностей, отриманих, наприклад, від пошукової машини, і запам'ятовування їх у локальній базі даних з ранжируванням їх для гарантії, що посилання, які посилаються на відповідності в списках, на візуалізованою сторінці виділені кольором відповідно до теми і їх рангом для фокусування я навігації тільки на тих сторінках, які релевантні для користувача і т.п.

Ці заявки, що мають відношення до Веб-навігації, хоча вони і можуть бути корисні для вдосконалення навігації, виконують свою роботу без розуміння сенсу придбаних термінів, тим і т.п. Таким чином, існує потреба в нових способах і пристроях для семантичної навігації по колекціях документів, таких як Інтернет і / або корпоративні сховища знань (Knowledge Warehouses), засновані на отриманні інформації, керованому знаннями, і використанні результатів для підтримки інтелектуальної навігації.

US6871140 B1, заявка 09/693988, 22.03.2005 “Система і спосіб для збору, поширення і використання інформації в зв'язку з комерційною нерухомістю”

Даний винахід забезпечує єдиний, надійний і об'єктивний Інтернет ринок для комерційної нерухомості. Система має доступ до баз даних, що містить незалежні і порівняльні дані про комерційну нерухомість, які постійно оновлюються і зібрані професійними дослідниками. Система включає в себе базу даних з докладним описом офісних і промислових приміщень в темі ринку, що є всеосяжною, точної і актуальної. Система також включає в себе

орендар інформаційної бази даних інформації про орендарів, що дозволяє користувачам ідентифікувати і призначатися найбільш ймовірними орендарями в оренду місця і визначити основний попит на комерційну нерухомість в своєму ринку.

US2014236919 (A1) “Executing a fast crawl over a computer-executable application”

Описано технології, пов'язані з обходом виконуваного комп'ютером додатки. Повний обхід виконується над додатком, де виконують повний обхід включає в себе викликаючи додаток для виведення безлічі сторінок. Додаток витягує вміст з World Wide Web при створенні сторінок для виведення. Після цього швидко повзати виконуються над додатком, де виконання швидких повзати займає менше часу в порівнянні з часом, необхідним для виконання повного обходу.

US19980086379 “Collaborative team crawling: Large scale information gathering over the internet”

Розподілена колекція веб-сканерів збирає інформацію з великої частини кіберпростору. Ці роботи здійснюють загальне сканування через схему кіберпростір розділу. Крім того, вони співпрацюють один з одним за допомогою балансування навантаження, щоб максимально використовувати обчислювальні ресурси кожного з негідників. Винахід використовує перевагу ієрархічної природи простору імен кіберпростору і використовує синтаксичні компоненти структури URL в якості основного транспортного засобу для поділу і присвоєння краулер навантаження на індивідуальний краулер. Схема секціонування повністю розподіляється, в якому кожен краулер приймає рішення секціонування на основі свого власного статусу сканувати і глобально реплицировать структуру розділів даних дерева.

US7676553 (B1) “Incremental web crawler using chunks”

Передбачені система і спосіб, що забезпечують інкрементне сканування веб-сторінок з використанням фрагмента (ів). Система може бути використана, наприклад, для полегшення системи веб-сканування, яка сканує (наприклад, безперервно) Інтернет для отримання інформації (наприклад, дані) і індексує інформацію, щоб вона могла використовуватися як частина механізму веб-пошуку. Система полегшує поетапний повторний обхід і / або вибіркове оновлення інформації (наприклад, документів) з використанням структури, званої шматком, для спрощення процесу покрокового обходу контенту.

US7236985, US7249146 “Комп'ютеризована система і спосіб для збору та аналізу даних, що відносяться до нерухомості”

Значна кількість загальнодоступною і приватної інформації, що відноситься до будівель в області вводиться в комп'ютерну базу даних, запрограмованої для отримання статистичних стандартів (норм), які потім можуть бути основою для подальшого порівняння з конкретними, обраними, структурами. Інформація організована і відформатована для полегшення аналізу різних доходів і статей витрат (включаючи податки), що відноситься до обраних структурам. Різні спеціальні екрани (форма) використовуються для полегшення введення даних, що відносяться до окремих будівель і портфелям будівель в комп'ютерну базу даних і в інтерактивному режимі допомогти в аналізі даних, введеному в зв'язку з інформацією, що міститься в комп'ютерній базі даних, в тому числі і формування та відображення суттєвої статистичної інформації про обрані властивості і генерації різних звітів. Система включає в себе програму, які дозволяють ефективно порівнянні різних структур щодо їх доходів і витрат, а також для генерації звітів для локального властивості і / або з метою оподаткування доходів, оцінки детермінації, цілі управління і / або інвестиційні рішення.

US 8468088 B2 US12 / 876581 18.06.2013 “Автоматизований видобуток і обробка даних, пов'язаних з нерухомістю”

Комп'ютеризовані процеси використовуються для ефективної видобутку і аналізу інформації, пов'язаної з конкретними іпотечними кредитами, позичальниками і власністю. Описані процеси використовують дані власності, пов'язані з агрегованими з декількох джерел і юрисдикцій, серед інших завдань, визначити власність, що належить фізичним особам.

US 7072890 B2 Method and apparatus for improved web scraping

Спосіб і пристрій для компоненту синтаксичного аналізатору веб-пошукової системи для адаптації у відповідь на часті веб-зміни формату сторінки на веб - сайтах. Parser «дізнається» з набору певних посилань HTTP, як знайти і аналізувати веб - сторінки, що повертаються із запиту пошукової системи. Винахід інтелектуально знаходить різні символьні рядки, які будуть правильно витягувати атрибути, пов'язані з повернутим пунктом. Цей винахід може працювати автоматично або в режимі за участю користувача.

US 20060287989 A1 “Extracting structured data from weblogs”

Спосіб вилучення окремих повідомлень з блогу включає в себе етапи, на яких: (а) отримання посилання, пов'язаного з веб-журналами; і (б) екран вискоблювання веб-журналу в поданні для веб-журналу повідомлень з використанням даних вихідних даних, що містить частковий зміст веб-журналу.

US 20030036963 A1 09/766760 20.02.2003

“Спосіб і система для агрегації контенту інформації про нерухомість онлайн в обчислювальному середовищі”

Он-лайн система агрегування інформації про нерухомість для супроводу угод з нерухомістю шляхом агрегування реальної інформації нерухомості, пов'язані в центральному науково-дослідному центрі. Науково-дослідний центр інструмент дозволяє користувачам проводити дослідження конкретних властивостях, ринків, субринкам або типами нерухомості на ринку, щоб досліджувати і перевіряти кредитоспроможність потенційних орендарів, а також шукати комерційні новини нерухомості статті з агрегированной бази

даних. Он-лайн нерухомість система агрегації інформації містить розподілену комп'ютерну мережу, безліч клієнтських комп'ютерів, з'єднаних з розподіленою комп'ютерною мережею, і платформу сервера послуг власності, підключеної до джерела даних. Хостинг на серверній платформі послуги власності є модулем дослідження ринку для доступу до інформації досліджень на доступних властивості і ринках, кредитний звітів модуль для доступу до кредитної звіт інформації про приватних осіб і компаніях, новини і аналізі модуля для доступу до інформації, новин нерухомості, а замовлення доступних звіти модуль, розміщений на зазначену платформу сервера служби власності для надання можливості покупки ринкових звітів.

US 7945583 B2 17.05.2011 «Методика інтелектуального аналізу даних (data mining) з використанням веб-служби»

Розкривається методика розгортання алгоритмів інтелектуального аналізу даних на веб-службі, наприклад IBM WebSphere Application Server. Замість того, щоб розгортати моделі інтелектуального аналізу даних з даними, дані можуть переноситися на веб-сервер як частина повідомлення. Моделі можна кешувати на веб-сервері і легко змінювати за допомогою операцій, що виконуються клієнтом. Це дозволяє ефективно адмініструвати операційну середу. Оскільки середовище веб-сервісів за своєю природою масштабируема, сервери можуть бути прозоро включені на вимогу. Крім того, зв'язок з веб-сервісами здійснюється через об'єкти даних в пам'яті, що забезпечує простоту їх реалізації та операційну ефективність.

20170046787 A1 «Web Data Scraping, Tokenization, and Classification System and Method. Система і спосіб вискаблювання веб-даних, токенізація і класифікація»

Веб-сервер отримує дані URL для електронного ресурсу про об'єкт і стирає дані контенту про об'єкт з ресурсу. Процесор вмісту виконує токенізацію даних контенту, генерує дані підрахунку маркерів і зберігає дані підрахунку маркерів в одному або декількох пристроях зберігання даних.

Прогностичний процесор моделі застосовує дані підрахунку маркерів до навченої моделі прогнозування, підготовленої для генерування перших даних, що вказують, щонайменше, одну промислову класифікацію, яка застосовується до об'єкта, і другі дані, що вказують ймовірність того, що перші дані застосовні до об'єкту. Веб-сервер налаштований так, щоб надавати за допомогою пристрою зв'язку призначеному для користувача пристрою і реагувати на застосування навченої комп'ютеризованої моделі прогнозування на дані підрахунку маркерів, відображення, яке включає в себе перші дані і другі дані.

US 8655831 B1 13/562783 02/18/2014 «Smart parsing of data Розумний синтаксичний аналіз даних»

Методи, системи і пристрої, включаючи комп'ютерні програми, закодовані на комп'ютерному носії, для автоматичного аналізу даних з розрізнених джерел даних. У деяких реалізаціях дії включають в себе прийом перших даних від першого джерела даних, ідентифікацію першого регулярного виразу, яке відповідає формату даних перших даних, вибір першого набору правил синтаксичного аналізу з безлічі правил синтаксичного аналізу на основі першого регулярного виразу, Аналіз перших даних на основі першого набору правил синтаксичного аналізу для забезпечення першого набору допоміжних даних, заповнення полів даних першого об'єкта даних відповідними суб-да тах з першого набору допоміжних даних і передачу перших даних Об'єкт до обчислювальному пристрою.

US 7523125 «Система парсинга» Надається система розбору неструктурованих або частково структурованих даних, зокрема, даних імені та адреси на будь-якій мові або сценарії. Система обробляє, щонайменше, частину даних інкрементним чином, переважно використовуючи кілька етапів аналізу, причому кожен етап аналізу виконується за допомогою консультації з механізмом виведення, який використовує стратегію виведення. База знань використовується з механізмом логічного висновку для аналізу даних на одному або більше рівнях аналізу. База знань може за бажанням включати

рівень лексико-граматичного аналізу та орфографічний рівень, семантичний рівень, контекстний рівень.

ДОДАТОК Б.
Інформація про сучасні бізнес-проекти в галузі

Найменування та стисла інформація про сучасні бізнес-проекти	Ціна продажу / обсяг необхідних вкладень	Посилання
Агрегатор українського сегмента ринку легкових автомобілів. Пошуковий сервіс надає користувачам доступ до бази даних оголошень про продаж автомобілів, розміщених на більшості зі спеціалізованих сайтів. Актуалізація бази проводиться в реальному часі	$ 120.000	https://startup.ua/investor/investitsionnye-proekty/autostar.html
Джобагрегатор - всі вакансії країни в одному місці.	$ 300.000	https://startup.ua/investor/investitsionnye-proekty/hotwork-14365.html
Парсер "Кадровий скоринг" - Індексує резюме з безлічі відкритих джерел в інтернеті, включаючи сайти служб занятности і дошки оголошень, індексує власну базу резюме роботодавця. Виробляє аналіз текстів резюме і вакансій, застосовуючи передові алгоритми native language processing і витягує факти використовуючи внутрішню базу знань, яка містить описи галузей, посад, навичок і зв'язків між ними. Під час вилучення фактів використовуються алгоритми machine learning, навчені на десятках тисяч вакансій і резюме, розмічених вручну фахівцями. Заявлена 89% точність аналізу резюме. Розроблено процедуру скорингу резюме під параметри вакансії. Потрібно поставити вимоги вакансії і впорядкувати їх за пріоритетами. Система оцінить резюме на відповідність вимогам і видасть список найбільш підходящих. Конкуренти: - агрегатори вакансій (indeed.com); - Amazing Hiring (є істотні відмінності)	$ 500.000	https://startup.ua/investor/investitsionnye-proekty/kadrovyy-skoring-emply-15218.html
Новинний агрегатор "UA News" - серія програм для Android і iOS.Також працюють версії для інших країн (Росія, Білорусь та ін.). Додатки працюють більше 5 років, опубліковані в Google Play і App Store. Сумарний трафік складає більше 3 млн. Переглядів на місяць.Дохід від реклами на сьогодні становить від 400 до 600 дол. в місяць (встановлений тільки 1 невеликий банер внизу екрану, є можливість збільшити дохід за рахунок додаткової реклами). Конкуренти: "All News", "Новини України" та ін.	$ 49.000 Розмір частки до продажу 80%	https://startup.ua/investor/biznes-startap-na-prodazhu/novostnoy-agregator-ua-news-82664.html
Прайс агрегатор. Створення нового прайс агрегатора для роздрібної торгівлі через інтернет магазини використовуючи гібридну бізнес модель. Конкуренти: hotline.ua, price.ua, nadavi.com.ua	$ 10.000	https://startup.ua/investor/investitsionnye-proekty/new-price-agreator-21351.html
Прайс агрегатор.	$ 50.000	https://startup.ua/investor/investitsionnye-proekty/cmyklr.html
Агрегатор розпродажів Конкуренти: (Росія) Яндекс.Маркет, WikiMart, товары@mail.ru, groupon, biglion, LeroyMerlin, OBI, Castorama.	$ 100.000	https://startup.ua/investor/investitsionnye-proekty/postro-im-postro-im.html
Агрегатор служб доставки Vsedostavki	$ 100.000	https://startup.ua/investor/inve

Програмне рішення по моделі SaaS, яке дозволяє: порівнювати, вибирати, оформляти, відстежувати, економити за рахунок об'єднання всіх служб доставки в одному місці і інтеграції з інтернет магазинами. У проекті вже розроблені завдання і бізнес процеси; функціональна специфікація; технічне завдання; архітектура бази даних; серверна частина; калькулятор розрахунку вартості; підключені 3 служби доставки по API; прототипирование WEB інтерфейсу. Конкуренти: Teleport (delivery service) – хмарне рішення, може бути сильним конкурентом, реалізований базовий функціонал; myCargo, eDostavka, cargoUA - пропонують перелік служб доставок, мінімальний функціонал.		stitsionnye-proekty/vsedostavki-13526.html
Сайт-агрегатор знижок. Конкуренти: http://kuponi.com.ua Всього 5-6 реальних конкурентів по Україні.		https://startup.ua/investor/investitsionnye-proekty/pokupator-14081.html
Портал нерухомості - онлайн-послуги всім учасникам ринку нерухомості: власники, ріелтори, агентства, забудовники, юр. послуги, банки. Конкуренти: lun.ua, address.ua, domik.net, novostroykin.ru	$ 300.000	https://startup.ua/investor/investitsionnye-proekty/dom2000.html
Мобільний агрегатор новин Summly. Збирає найбільш релевантну інформацію зі статей різних ЗМІ і показує її користувачам у вигляді коротких абзаців з посиланнями на вихідні статті. Швидко знаходить новини по темам і використовуючи штучний інтелект стискання довільну статтю в резюме до 400 знаків, підбирає відповідні і привабливі картинки для оформлення на екрані мобільного пристрою. Розробник отримав $ 300,000 в першому раунді інвестицій, через рік ще $ 1,230,000.	Проданий Yahoo в 2013р. за $ 30000000.	https://habrahabr.ru/post/174237/

ДОДАТОК В.
Інформація щодо проектів, що надають послуги парсингу даних

Проекти, що надають послуги парсингу даних	Посилання
Парсинг сайтів оголошень по нерухомості. Парсинг і аналіз в автоматичному режимі безлічі оголошень з купівлі / оренді нерухомості / гаражів і т.д. Послуга парсинга сайтів нерухомості має на увазі створення спеціальної програми (або скрипта) для збору, перетворення і аналізу інформації з профільних (тематичних) майданчиків. Результатом роботи такої програми (парсеру) є готова база даних або окремий файл. Ви регулярно отримуєте з найбільших сайтів нерухомості (наприклад, MetrPrice, ціан ГРУП, РБК-Нерухомість, hata.by або ts.by) дані, які на них публікуються. Парсер розміщується онлайн і працює 24/7/365.	http://parsing.by/
basilisklab Пошук, збір і обробка інформації, обробку даних. Веб-аналітику і дослідження. Ви можете отримати будь-які дані, які можна зібрати в інтернеті. Парс все. Багато відомих інтернет-маркетологи і інтернет-компанії, що займаються розвитком сайтів з великим об'ємом контенту.	https://basilisklab.com/
Parser Acula.Pro У вартість входить збір даних з сайту донора. Коректіровкі зібраних даних, наприклад часто потрібно очистити отриманий матеріал від HTML тегів. Підготовка отриманих даних під імпорт в CMS або базу даних. Оголошення по нерухомості упорядковано відповідно до районам, вулицями, номерами будинків, типам оголошень і іншим параметрам.	http://parsing.website/
Parsing.center - парсинг даних з різних джерел, наприклад: парсинг товарів з інтернет магазину, парсинг статей з певного сайту, парсинг резюме, парсинг телефонів, парсинг соц мереж, збір інформації з сайтів оголошень, номери телефонів, e-mail адреси, назви організацій і тд. Обробка отриманих даних: -конвертірованіе в потрібний для Вас формат виводу, наприклад .xml, .xls, .csv і тд; -машинний переклад контенту через yandex translate; -роспізнавання зображень номерів телефонів. Автоматичний онлайн парсинг різноманітних даних з інтернет ресурсів.	https://parsing.center
Парсинг Контент Парсинг оголошень з різних сайтів (наприклад оголошення про нерухомість, про автомобілі)	парсинг-контент.рф
Каталог і магазин баз даних «База БД» Замовлення парсинга сайтів різної спрямованості, в т.ч. дощок оголошень	http://bazabd.com/parser
Парсинг сайтів, збір даних. Послуги вилучення даних з різних веб-сайтів або пошукових систем і їх подальшу обробку.	http://grablab.org/ru/

ДОДАТОК Г.
Інформація щодо програм та сервісів з веб-парсингу текстових даних

Найменування та характеристика програм та сервісів з веб-парсингу текстових даних	Коментар до назви (до читання)	Розробник	Посилання
Простий онлайн сервіс копіювання статей / новин, практично з будь-якого сайту в інтернеті, на Ваш сайт в автоматичному режимі за допомогою нашого робота. Налаштування сервісу не вимагає знання верстки та програмування, розуміння, що таке теги html і т.д. і розрахована на простого користувача. Вся робота виконується в інтуїтивно зрозумілій візуальному редакторі.	Парсер контенту.Онлайн-сервіс копіювання інформації з сайтів	студія Qteam	https://marketplace.1c-bitrix.ru/solutions/qteam.parser/ http://www.qteam.ru/parser/
Безкоштовний граббер Proxy Gear Pro v2.1 (Proxy Combine: checker, parser, grabber). Трохи про можливості та особливості: Зручний інтерфейс Граббер Google (з підтримкою капчи) URL граббер Висока швидкість Адаптивні фільтри Автопарсер проксі з тексту / файлів / URL Відкриття відразу декількох файлів загальним обсягом> 2ГБ Визначення країни проксі Експорт списків в будь-якій зручній масці Визначення таймаута прокс.і Установка системного проксі сервера прямо з програми (до 1000 потоків і управлінням пріоритетом). Можливість встановити таймаут з'єднання Визначення типу проксі (Transparent, Anonymous або Elite) Робота через власний чек-сервер	програма	Vahtang Gegechkori	http://proxy-base.com/ http://gegechkori.ru/
AftParser - безкоштовний парсер для Вордпресс. Дозволяє збирати контент з одного або декількох джерел і обробляти його, підганяючи під потрібний формат використовуючи всі можливості мови PHP. Присутня можливість відкладеного парсинга. Складається з чотирьох сторінок: головна, парсер посилань, парсер RSS, сторінка налаштувань. Плагін складається з 24 файлів (не рахуючи картинки), включає в себе 4 дуже потужних класу, 12 оброблювачів ajax і купу коду. Весь код супроводжується розгорнутими коментарями і добре оформлений.	Плагін для Вордпресс		aftamat4ik.ru/portfolio/aftparser
Парсинг сайтів. Парсинг онлайн. Парсинг даних з різних джерел, наприклад збір інформації з сайтів оголошень, номери телефонів, e-mail адреси, назви організацій і тд; Обробка отриманих даних: конвертація в потрібний формат виведення, наприклад .xml, .xls, .csv і тд; машинний переклад контенту через yandex translate; розпізнавання зображень номерів телефонів; На сайті можна скористатися online парсинга, тобто за допомогою сайту можна отримати дані (Спарс) з інших сайтів абсолютно безкоштовно.	онлайн сервіс	info@parsing.center	https://parsing.center/ru/

ДОДАТОК Д.
Електронні плакати

Мета і завдання дослідження

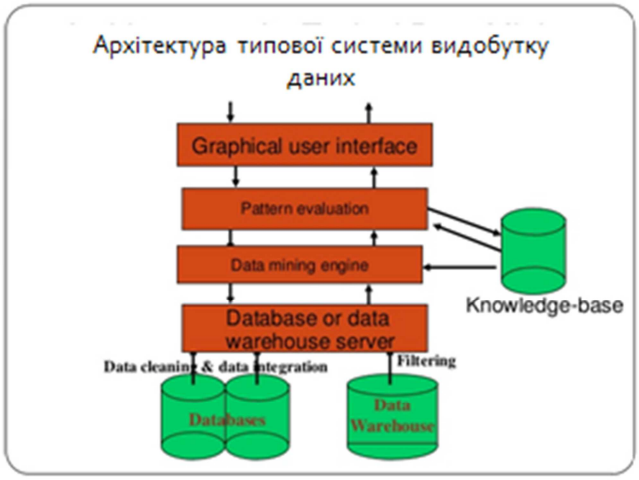
1. Проаналізувати існуючі технології пошуку, вилучення та об'єднання вмісту веб-сторінок.
2. Дослідити методи інтелектуального аналізу веб-контенту.
3. Розробити програмне забезпечення для вилучення вмісту з веб сторінок щодо продажу на автомобільному ринку.
4. Розробити інформаційну систему яка дозволяє зберігати і надавати кінцевому користувачеві витягнуті дані.

Аналіз теорії і досліджень по темі диплома

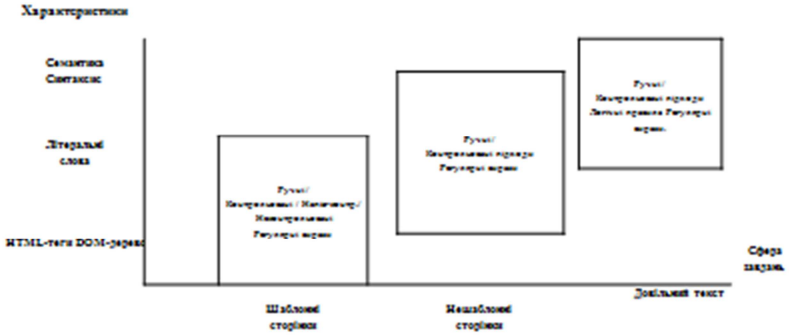
Етапи Web Mining :

- **вхідний етап (англ. input stage)** - отримання «сирих» даних з джерел (логи серверів, тексти електронних документів);
- **етап попередньої обробки (англ. preprocessing stage)** - дані подаються у формі, необхідної для успішної побудови тієї чи іншої моделі;
- **етап моделювання (англ. pattern discovery stage);**
- **етап аналізу моделі (англ. pattern analysis stage)** - інтерпретація отриманих результатів.

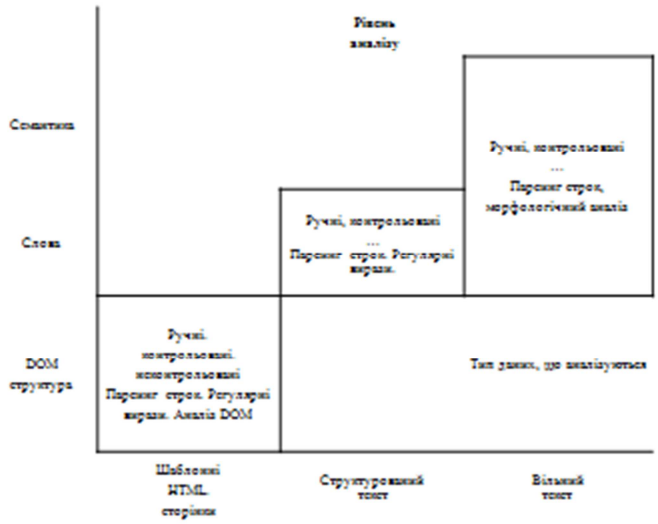
Питання архітектури сервісу



Зв'язок сфери завдань та методів обробки



Методи аналізу даних застосовуються в залежності від типу даних



Основні методи аналізу веб контенту

Ручні	Керовані	Напів-керовані	Не керовані
TSIMMIS [Hammer1997]	WIEN [Kushmerick1997]	IEPAD [Chang2001]	RoadRunner [Crescenzi2001]
Minerva [Crescenzi1998]	SRV [Freitag1998]	OLERA [Chang2006]	DeLa [Wang2002]
WebOQL [Arocena1998]	RAPIER [Califf 998]	Thresher [Hogue2005]	EXALG [Arasu2003]
XWRAP [Liu2000]	NoDoSe [Adelberg1998]	IDE [Zhai2005]	DEPTA [Zhai2005]
W4F [Sahuguet2001]	SoftMealy [Hsu1998]		NET [Zhai2005]
	WHISK [Soderland1999]		IEKA [Wong2007]
	STALKER [Muslea1999]		ViDE [Liu2010]
	DEByE [Laender2002]		

Основний компонент сервісу – обгортка (wrapper).

Для автоматичного отримання даних з Web-сайтів механізми виконання запитів взаємодіють з безліччю обгорток (wrapper). Обгортка - це специфічна для кожного Web-сайту програма, завданням якої є трансляція даних Web-сайту в форму, що дозволяє здійснювати подальшу їх обробку засобами системи інтеграції даних. Наприклад, обгортка може витягувати безліч кортежів з HTML-файлу.

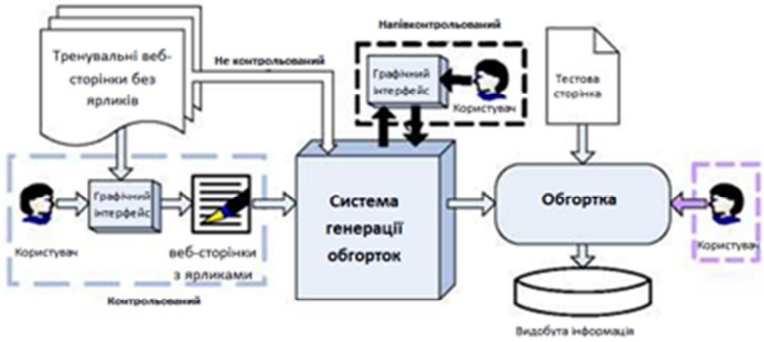
Обгортка у видобутку даних являє собою програму, яка витягує вміст конкретного джерела інформації і перетворює його в реляційної форми. Багато веб-сторінок мають **структуровані дані** - телефонні довідники, каталоги продукції тобто відформатовані для перегляду людиною за допомогою мови HTML. Структуровані дані як правило це описи об'єктів, які отримані з основних баз даних і відображаються на веб-сторінках після певних фіксованих шаблонів. Програмні системи, що використовують такі ресурси повинні переводити вміст HTML в реляційну форму. Обгортки зазвичай використовуються в якості таких трансляторів. Формально обгортка є функцією від сторінки до набору кортежів, які вона містить.

Є два основні **підходи до генерації обгортки** (або екстрагування інформації): індуктійна обгортка (Wrapper induction, WI) та автоматизоване вилучення даних. Індуктійна обгортка використовує контрольоване навчання, щоб дізнатися правила отримання даних з вручну мічених навчальних прикладів. Вона спрямована на структуровані та напівструктуровані документи типа веб-сторінок.

HTML обгортки на можна розділити на 3 категорії відповідно до типів завдань екстракції.

Обгортки на рівні записів	Обгортки на рівні сторінки	Обгортки на рівні сайту
Використовують закономірності для встановлення меж записів, а потім витягують елементи єдиного списку однорідних записів зі сторінки	Витягують елементи декількох видів записів	Наповнюють базу даних зі сторінок веб-сайту

Загальний вигляд систем генерації обгорток



Системи геперації обгорток за рівнем участі корстувача

1. Контрольовані
2. Напівконтрольовані
3. Не контрольовані

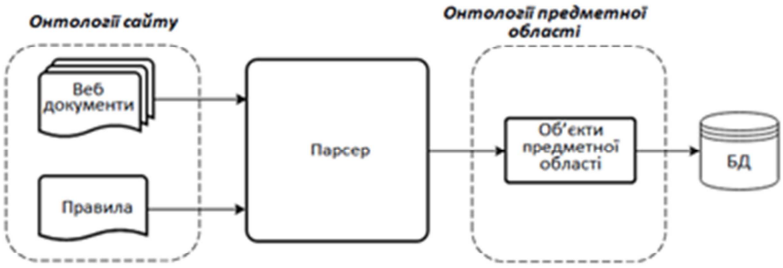
Аналіз інструментів Web Mining на основі використаних методів

Інструменти	Кількість скануємих	Тип правил екстракції	Використані особливості	Алгоритм навчання	Лексемізація шаблонів
Minerva	один	Регулярні вирази	HTML теги / літеральні слова	ні	вручну
TSIMMIS	один	Регулярні вирази	HTML теги / літеральні слова	ні	вручну
WebOQL	один	Регулярні вирази	гіпердерево	ні	вручну
WIF	один	Регулярні вирази	DOM дерево шлях адреації	ні	рівень тегу
XWRAP	один	Контекстно-незалежний	DOM дерево	ні	рівень тегу
RAPIER	множина	правила логіки	Синтаксичний / семантичний	ILP (вчиу вчерх)	рівень слова
SRV	множина	правила логіки	Синтаксичний / семантичний	ILP (вчерху вчиз)	рівень слова
WHISK	один	Регулярні вирази	Синтаксичний / семантичний	Покривальний набір (вчерху вчиз)	рівень слова
NoDoSE	один	Регулярні вирази	HTML теги / літеральні слова	мультивказні деяк	рівень слова
DEByE	множина	Регулярні вирази	HTML теги / літеральні слова	мультивказні деяк	рівень слова
WIEN	один	Регулярні вирази	HTML теги / літеральні слова	Ad-Hoc (від шиу до верху)	рівень слова
STALKER	множина	Регулярні вирази	HTML теги / літеральні слова	Ad-Hoc (від шиу до верху)	рівень слова
SoftMealy	обидва	Регулярні вирази	HTML теги / літеральні слова	Ad-Hoc (від шиу до верху)	рівень слова
IEPAD	один	Регулярні вирази	HTML-теги	Pattern Mining, вирівнювання строк	багаторівнева
OLERA	один	Регулярні вирази	HTML-теги	рядок вирівнювання	багаторівнева
DeLa	один	Регулярні вирази	HTML-теги	Pattern Mining	рівень тегу
RoadRunner	один	Регулярні вирази	HTML-теги	рядок вирівнювання	рівень тегу
EXALG	один	Регулярні вирази	HTML теги / літеральні слова	Диференціювання класів, реінформування по DOM шляху дерева	рівень слова
DEPTA	один	Дерево тегів	HTML теги поо HTML теги	Pattern Mining, порівняння рядків, часткове вирівнювання дерев	рівень тегу

Загальний алгоритм отримання даних з окремого сайту в розробленому сервісі

- скрапінг / краулінг початкової сторінки шляхом застосування cURL, побудови DOM об'єкту XPath;
- отримання посилань на проміжні сторінки (зазвичай отриманий перелік посилань багатосторінковий) з початкової сторінки шляхом XPath - запитів;
- отримання кінцевих сторінок з проміжних сторінок;
- нормалізація вихідного тексту;
- видобуток значущої інформації шляхом застосування регулярнх виразів (PCRE);
- видалення дублікатів, не релевантних або недостатньо насичених значимою інформацією;
- передання отриманої структурованої інформації до бази даних Estate утвореної в phpMyAdmin (СКБД MySQL).

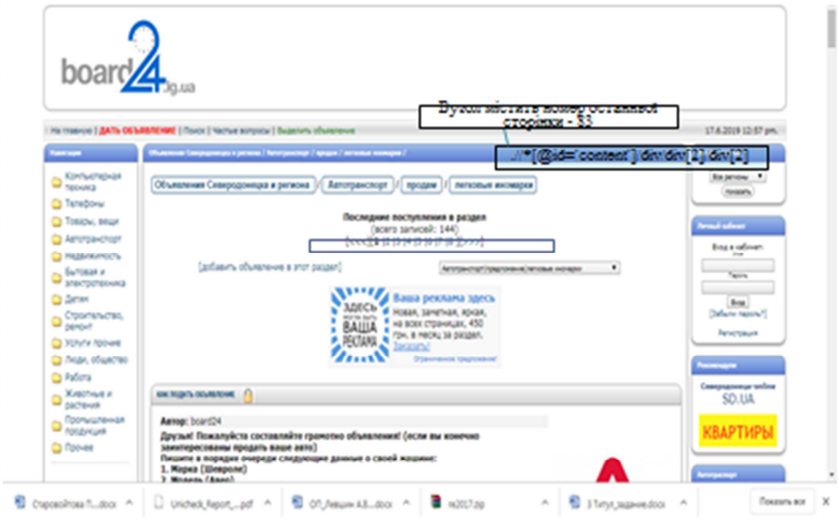
Загальна структура парсеру на основі онтологій



Парсер необхідно налагодити як під вимоги певного сайту так і під вимоги певної предметної області.

На вхід програми надходить **веб документ** та **правила обробки**, далі за певними правилами **виконується перетворення** даних у набір об'єктів предметної області.

Аналіз структури вузлів сторінки



Онтологія автомобілей

На попередньому малюнку відзначені інформаційні області на сайті http://board24.lg.ua/, дані з яких заносяться в базу даних автоматизованої інформаційної системою, такі як:

- марка
- модель
- рік виробництва
- пробіг
- тип кузову
- кількість місць
- обсяг двигуна
- тип палива
- колір
- ціна
- інформація про дату розміщення оголошення;
- контактний телефон продавця;
- фотографії об'єкта.

Використані для аналізу шаблони регулярних виразів (PERL-синтаксис)

Елементи онтології автомобілей	Основний шаблон
Вартість авто	'/(\d{o,3}(?:[|\,][o,1})\d{3}).{o,2}[\$|y\.\s?e\.|USD|доллар]/ui' , '/\d{o,2}[\s|\.],{o,1}\d{3,}\s[(?=гривен)|(?=грн)]/ui'
Загальний пробіг	'/\d+,?\d?\s?(?:км)|(?:тис.км)|(?:тыс.км)/ui' , '/(?:общ\..?пробег|общий пробег|общим пробегом)\D{o,3}\d{2,3}[,\.]?\d?/ui'
Пасажирських місць	'/\d{2,4,7}\s?\/\s?\d{2,4,7}/'
Фізичний стан	'/(?:[,\.])(\s?[а-яА-ЯЁё]+\s?)*(?:норм.?|авар.?|ремонт.? |окрас.?)\s?[а-я]*/ui'
Об'єм двигуна	'/\d{o,9}\s?[.]?/ui',
Терміновість продажу	'/срочн/ui'

Структура таблиці бази даних пропозицій автомобілей Estate

№ п/п	Ім'я	Тип даних	Коментар	Додатково
1	id	int(11)	ідентифікатор запису	AUTO_INCREMENT
2	name	varchar(100)	Назва оголошення	
3	region	varchar(50)	Регіон	
4	city	varchar(100)	Город	
5	marka	varchar(50)	Марка	
6	model	varchar(100)	Модел	
7	year	int(4)	Рік	
8	run	int(10)	Пробіг	
9	kuzov	int(10)	Тип кузову	
10	mass	int(10)	Кількість місць	
11	val	float(10,2)	Об'єм двигуна	
12	petrol	varchar(20)	паливо	
13	price_beta_usd	int(10)	Ціна оголош $	
14	price_beta_uah	int(10)	Ціна оголош грн	
15	seller	varchar(20)	продав	
16	price_uah	int(10)		
17	data_publish	date	Дата публікації оголошення	
18	data_load	date	Дата завантаження оголошення	
19	url	varchar(100)	URL оголошення	
20	photo	blob	Файл з фотографією	
21	add_info	varchar(2000)	Додаткова інформація	
22	contacts	varchar(1000)	Контакти	
23	urgency	tinyint(1)	Терміновість	
24	screenshot	blob	Копія екрану оголошення	

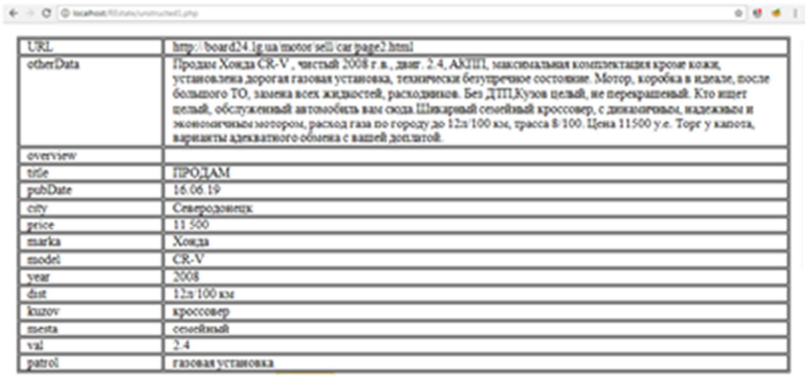
Зовнішній вигляд онлайн-сервісу для автоматизованого збору контенту з веб ресурсів

Розроблений онлайн сервіс розміщений на локальному сервері.

Запуск програми здійснюється вручну після обрання параметрів пошуку.

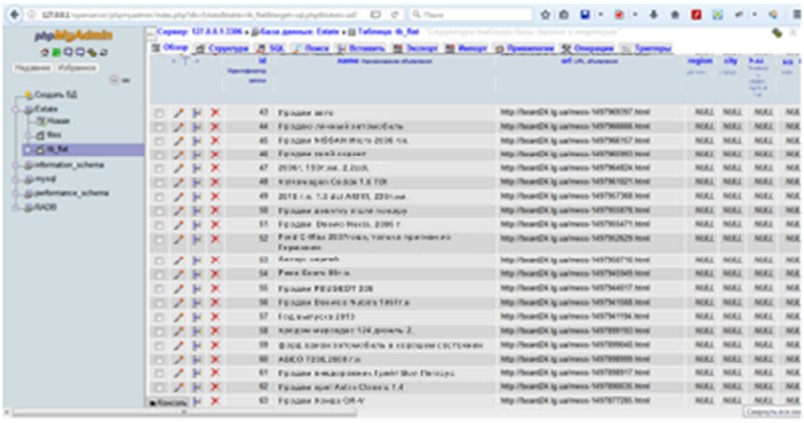
Під час запиту можливо обрати населений пункт, в якому здійснюється збір даних, тип кузову, а також внести додаткові орієртири для коригування бази.

Результати роботи сервісу (пооб'єктний вивід)



URL	http://board24.lg.ua/motor/sell/car/page2.html
otherData	Продам Хонда CR-V , чистый 2008 г. в., двиг. 2.4, АКПП, максимальная комплектация кроме кожи, установлена дорогая газовая установка, технически безупречное состояние. Мотор, коробка в идеале, после большого ТО, замена всех жидкостей, расходников. Без ДТП,Кузов целый, не перекрашеный. Кто ищет целый, обслуженный автомобиль вам сюда.Шикарный семейный кроссовер, с динамичным, надежным и экономичным мотором, расход газа по городу до 12л/100 км, трасса 8/100. Цена 11500 у.е. Торг у капота, варианты адекватного обмена с вашей доплатой.
overview	
title	ПРОДАМ
pubDate	16.06.19
city	Северодонецк
price	11 500
marka	Хонда
model	CR-V
year	2008
dist	12л/100 км
kuzov	кроссовер
mesta	семейный
val	2.4
patrol	газовая установка

Зовнішній вигляд бази даних після зберігання отриманих дани



Висновки по роботі відповідно поставленої мети

- Розроблено додаток для вилучення інформації щодо продажу автомобілей з різних веб сайтів.
- Додаток має систему конфігурації що дозволяє налаштовувати його на будь-який сайт. Також виділення різних алгоритмів вибірки даних в окремі класи дозволило забезпечити легку розширюваність системи.
- Розроблений веб інтерфейс, який відображає знайдені об'єкти та надає базові інструменти пошуку.
- Використання сервісу підвищує ефективність роботи фахівців з оцінки автомобільного ринку.

Напрямки подальшого вдосконалення інтелектуального сервісу

- використання методів машинного навчання для побудови онтологій;
- підвищення зручності та функціональності графічного інтерфейсу в тому числі відображення знайдених об'єктів на карті;
- підвищення адаптивності пошукових шаблонів під невизначене коло дошок оголошень;
- підвищення гнучкості використання онтологій для можливостей поширення дії сервісу на суміжні напрямки оцінки;
- впровадження можливостей суцільного пошуку в інтернет;
- впровадження аналітичних можливостей;
- розширення регіону та напрямків пошуку (за видами комерційних послуг автомобільного ринку);
- автоматичний збір скріншотів по відібраних об'єктах;
- оптимізація черг завдань на краулінг, парсинг та передачу результатів до бази даних;
- використання декількох проксі-серверів під час краулінгу;
- доповнення сервісу модулем автоматизації незалежної оцінки автомобілей.