НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ _____
(підпис) (ініціали, прізвище)
« _____» _____ 2015 р

Магістерська дисертація

зі спеціальності 8.05020102 «Комп'ютеризовані та робототехнічні системи»
(код і назва спеціальності)
на тему: Дослідження алгоритмів інтелектуального аналізу вмісту web-ресурсів для пошуку значимої інформації

Виконав: студент VI курсу, групи ІК-31м
(шифр групи)
Чернявський Дмитро Юрійович _____
(прізвище, ім'я, по батькові) (підпис)
Науковий керівник к.т.н., доц. Тимошин Ю.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)
Консультант охорона праці к.т.н., доц. Праховнік Н. А. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)
Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____
(підпис)

Київ – 2015 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

Факультет (інститут)_____Інформатики та обчислювальної техніки_____
(повна назва)

Освітньо-кваліфікаційний рівень _____«Магістр»_____
(назва ОКР)

Напрям підготовки _____6.050201 “Системна інженерія”_____
(код і назва)

Спеціальність _8.05020102 “Комп’ютеризовані та робототехнічні системи”_
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ __Ткач М.М.__
(підпис) (ініціали, прізвище)

«___»_____2015 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
_____Чернявському Дмитру Юрійовичу
_____
(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) _Дослідження алгоритмів інтелектуального аналізу вмісту web-ресурсів для пошуку значимої інформації__

Науковий керівник дисертації_____Тимошин Ю.А._____к.т.н. _доцент_____,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «_10_»_березня_ 2015 р. №_292/2С__

2. Строк подання студентом дисертації___27.05.2015_____

3. Об’єкт дослідження Алгоритми інтелектуального аналізу даних web-ресурсів

4. Предмет дослідження Ефективність алгоритмів інтелектуального аналізу вмісту web-ресурсів

5. Перелік завдань які необхідно розробити__аналіз роботи алгоритмів пошуку значимої інформації, визначення слабких місць алгоритмів та пошук їх вирішення, розробка алгоритму попередньої обробки документів, розробка програмного забезпечення для експериментальних досліджень ефективності алгоритму, аналіз приміщення, в якому була виконана дипломна робота.

6. Орієнтовний перелік ілюстративного матеріалу структурні схеми системи інтелектуального аналізу інформації, блок-схема запропонованого алгоритма, схема приміщення, в якому виконувалась дипломна робота.

7. Орієнтований перелік публікацій Підвищення ефективності роботи алгоритмів інформаційного пошуку/ Молода наука України. Перспективи та пріоритети розвитку.— Київ:ГО «ІОМП», 2015.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Розділ охорони праці	Праховнік Н.А., доцент		
2. Нормоконтроль	Пасько В.П., доцент		

9. Дата видачі завдання 26.09.2014

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Огляд найбільш поширених алгоритмів інтелектуального аналізу вмісту веб-ресурсів і принципів їх роботи	27 вересня – 10 листопада 2014 р.	
2	Аналіз існуючих проблеми відбору інформації та наявних методів їх вирішення	11 листопада – 20 грудня 2014 р.	
3	Розробка алгоритму обробки документів, що дозволить підвищити ефективність роботи пошукових алгоритмів.	21 грудня 2014 р. – 15 лютого 2015 р.	
4	Розробка програмного забезпечення.	16 лютого – 10 березня 2015 р.	
5	Виконання експериментальних досліджень.	11 березня – 20 квітня 2015 р.	
6	Оформлення розділу охорони праці.	21-30 квітня 2015 р.	
7	Оформлення роботи.	1-20 травня 2015 р.	

Студент _____ Д.Ю. Чернявський
(підпис) (ініціали, прізвище)

Керівник проекту (роботи) _____ Ю.А. Тимошин
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломної роботи складається з 4 розділів, 90 сторінок, містить 12 рисунків, 4 таблиці, 7 додатків, 23 джерел.

Дипломна робота присвячена дослідженню алгоритмів інтелектуального аналізу даних веб-сторінок та розробці алгоритму передньої обробки документів для видалення інформаційного шуму.

Розробка має забезпечити підвищення ефективності роботи алгоритмів інформаційного пошуку за рахунок виділення значимої частини веб-документів.

Основною цілю розробки алгоритму є знаходження ефективного методу аналізу HTML-сторінок, виділення та видалення блоків, що не містять інформації, цінної для виконання інформаційного пошуку.

У роботі наведено:

- аналіз методів вирішення задач відбору інформації;
- вирішення загальної задачі інформаційного пошуку;
- розробка алгоритму ефективної обробки веб-сторінок;
- програмне забезпечення для експериментальних досліджень;
- результати експериментальних досліджень;
- аналіз умов праці в приміщення, в якому буде використовуватись програмний продукт;
- висновки до роботи.

Результати експериментів показали, що застосування алгоритмів, подібних описаному, може бути ефективно для вузьких завдань інформаційного пошуку, коли апріорі відомо, що елементи оформлення групи індексованих ресурсів заважають інформаційного пошуку.

Ключові слова: інформація, документ, веб-сторінка, відбір інформації, інформаційний пошук, релевантність пошуку.

ABSTRACT

Explanatory note of master's thesis consists of 4 sections, 90 pages, includes 12 figures, 4 tables, 7 appendices, 23 sources.

Thesis is devoted to research of algorithms of web mining and developing of algorithm of document processing to remove noise information.

Development should provide the improving of the efficiency of information retrieval algorithms through significant allocation of web documents.

The main aims of the development of the algorithm are finding an effective method of analyzing HTML-pages allocation and removing blocks that do not contain information valuable to perform information search.

During the work was carried out:

• Analysis methods for solving problems of selection information;

• Solving the general problem of information retrieval;

• Development of efficient processing algorithm web pages;

• Software to perform experimental studies;

• The results of experimental studies;

• Analysis of working conditions in the room in which the software will be used;

• Conclusions to work.

The results of experimental research show that the use of algorithms similar to those described can be effectively narrow tasks of information retrieval when you know a priori that groups linked design elements interfere resources information search.

Keywords: information, document, webpage, information extraction, information retrieval, relevance of search results.

Пояснювальна записка
до магістерської дисертації

на тему: Дослідження алгоритмів інтелектуального аналізу вмісту web-ресурсів для пошуку значимої інформації

ЗМІСТ

ВСТУП

Разом із значним зростанням об’єму загальнодоступних джерел інформації і електронних документів в мережі Інтернет, зростає потреба в системах аналізу та відбору інформації. Останнім часом, все більше й більше дослідницьких груп концентрують свою увагу на розробці систем пошуку значимої інформації [1].

Відбір інформації застосовується для пошуку певної інформації з неструктурованого тексту природною мовою та представлення її у визначеному форматі, наприклад, у формі бази даних. Дослідження в цій області можуть бути умовно поділені на два напрямки: використання шаблонів для пошуку цільової інформації, і застосування алгоритмів машинного навчання для автоматичної побудови таких шаблонів [2].

На даний момент існує досить багато систем відбору інформації, частина з яких дають хороші результати, оброблюють дані набагато швидше за людину та за точністю можуть порівнюватись з обробкою вручну [1], [3].

В даній роботі увага зосереджується на основних компонентах шаблонів пошуку і алгоритмах машинного навчання, виконується їх дослідження та порівняння. Також пропонується алгоритм структурного аналізу web-сторінки, що призначений для збільшення ефективності застосування алгоритмів пошуку значимої інформації.

Метою даної роботи та дослідження алгоритмів інтелектуального аналізу вмісту web-ресурсів і розробка алгоритму для підвищення їх ефективності.

Об’єктом даного дослідження є алгоритми інтелектуального аналізу даних web-ресурсів.

Предметом дослідження є ефективність алгоритмів інтелектуального аналізу вмісту web-ресурсів.

Актуальність даного дослідження полягає в необхідності підвищення ефективності методів витягання знань на основі інформації, яка може вилучатись з розподілених різнорідних джерел, що дозволить інтелектуалізувати web-контент, доповнивши зміст інтернет-ресурсів такими формальними описами, які нададуть можливість інтелектуальним програмам або агентам автоматично переконуватися у достовірності в певному контексті наданої системою інформації.

Наукова новизна даної дисертації полягає у тому, що запропоновано новий алгоритм попередньої обробки web-сторінок, призначений для підвищення ефективності пошуку значимої інформації шляхом виокремлення змістовної частини вихідного документу та видалення інформаційного шуму, який негативно впливає на якість відбору інформації. Також проведено дослідження впливу застосування запропонованого алгоритму на ефективність роботи алгоритмів інформаційного пошуку.

РОЗДІЛ 1. ЗАДАЧІ ТА ЕТАПИ ПОШУКУ ЗНАЧИМОЇ ІНФОРМАЦІЇ

Алгоритми відбору інформації застосовуються для знаходження необхідної інформації в неструктурованих документах природною мовою і представлення результатів в структурованому форматі, такому як, наприклад, табличне представлення або база даних. З розвитком Інтернету за останні роки відбір інформації також широко застосовується для знаходження ключових фраз на web-сторінках, що відрізняються від тексту на природній мові своїм структурованим або слабо структурованим форматом [1].

Більшість сучасних систем відбору інформації з текстів побудовані для вирішення певних завдань, сформульованих заздалегідь. Немає системи, яка була б розрахована на максимальне рішення задачі вилучення – повний синтаксичний і семантичний аналіз довільного тексту і на отримання всієї інформації, яка міститься в цьому тексті. Будь-яка задача для системи отримання даних з тексту може бути сформульована в діапазоні від мінімальної до максимальної. Завдання-мінімум – витягти з тексту іменовані сутності (географічні назви, імена людей, посади, звання, назви організацій і т.д.). Завдання-мінімум успішно вирішується в багатьох сучасних системах добування інформації з тексту [4], [10]. Наступний крок – виділення з тексту складних структур, складніших, ніж просто об'єкти. Тобто необхідно пов'язати витягнуті вже на першому етапі аналізу об'єкти відносинами, інформація про які теж повинна бути отримана з тексту. Наприклад, збір інформації про організації та установи – хто очолює, де знаходяться філії, коли вони були відкриті, хто ними керує, інформація

про видані укази і розпорядження, результати виборів і референдумів тощо.

В даному розділі висвітлено поняття отримання та відбору інформації та різниця між ними, також розглядаються загальні принципи відбору інформації, що використовуються в природній мові. Далі описується застосування методики відбору інформації для web-сторінок.

1.1. Інформаційний пошук та відбір інформації

Інформаційний пошук є досить зрілою технологією, що використовується для добування набору релевантних матеріалів з великого об’єму вихідних даних. Типовим прикладом є пошук за ключовими словами в мережі Інтернет, і, як результат, отримання переліку web-сторінок, що їх містять. Число таких сторінок може бути дуже великим і не вся інформація, що міститься на сторінці є цінною. Відповідно, перегляд всіх сторінок для пошуку необхідної інформації може бути неможливим або потребувати значних затрат часу [2].

На відміну від інформаційного пошуку, відбір інформації вирішує описану проблему за допомогою знаходження в документі цільових фраз та перетворення їх в структурований формат. Рисунок 1.1 ілюструє приклад застосування відбору інформації. Дано документи з анонсом семінарів, які містять такі елементи як Дата, Час початку, Місце проведення, Доповідач і Тема, що можуть бути ідентифіковані і заповнені у відповідному шаблоні, визначеному наперед.

Для отримання найбільш ефективних результатів доцільно буде комбінувати обидва підходи, тобто спочатку зібрати набір релевантних документів серед загального об’єму за допомогою алгоритмів

інформаційного пошуку, а потім ідентифікувати тільки значиму інформацію за допомогою відбору [5].

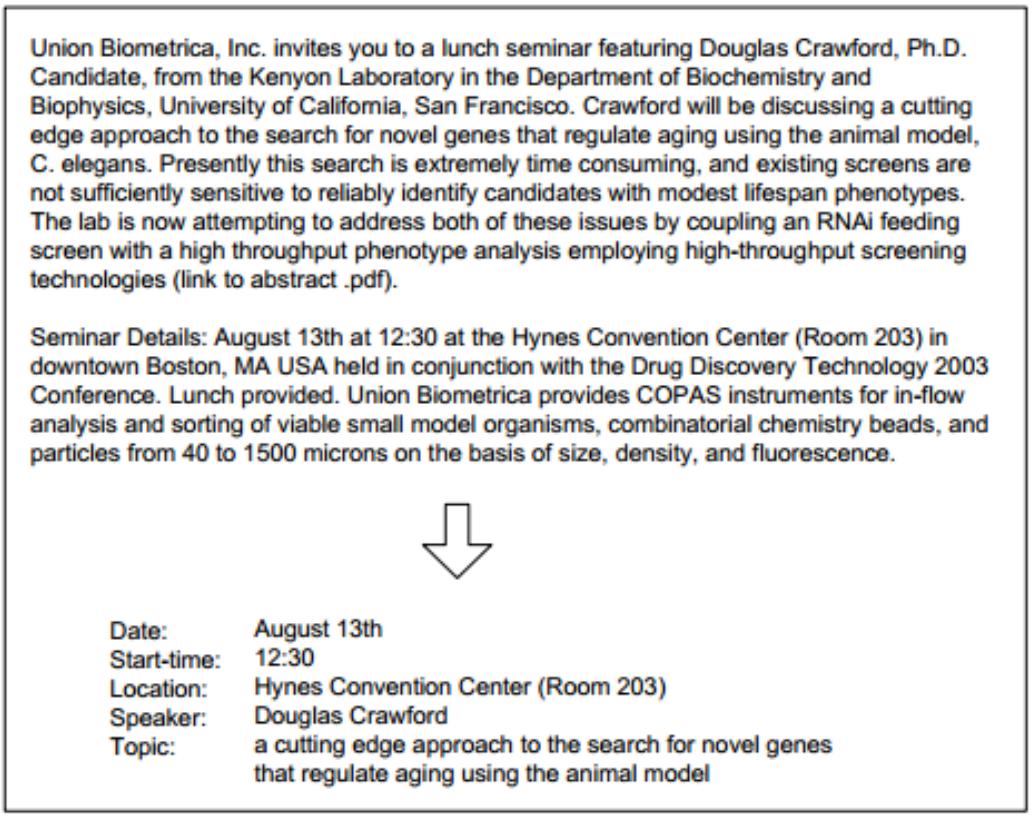


Рис. 1.1. Відбір інформації з оголошення про семінар

1.2. Відбір інформації та природна мова

Хоча інформація в мережі Інтернет представлена безліччю різних способів і в різних форматах, основним об'єктом уваги в даній статті буде текст на природній мові. Якщо наука поки тільки «підбирається» до побудови ефективних алгоритмів для розпізнавання візуальних образів, звуку і відео, то текст на природній мові, здавалося б, представлений в найбільш зручному для машинної обробки вигляді. Тим не менш, відбір інформації з текстів є досить складним завданням. Наведемо нижче загальний опис процесу аналізу тексту, а також список основних проблем. На вхід будь-якій системі для аналізу тексту, як правило, надходить текст у вихідному форматі, тобто в популярних

форматах документів DOC, OAT, RTF, PDF, HTML. Такі документи крім самого тексту містять також форматування, виноски, «зайві» ділянки тексту (реклама). Першою складністю при аналізі тексту є отримання тексту з документів, поданих на вхід. Далі, отриманий текст слід пропустити через графематичний аналізатор, завдання якого – визначити межі слів, речень, абзаців, а також відзначити числа, розділові знаки та інші символи і послідовності символів, які є словами, але які відіграють важливу роль у реченні. Складність даного етапу полягає в тому, що деякі послідовності символів часто буває неможливо ідентифікувати на цьому етапі аналізу. Якщо взяти до уваги необхідність обробки друкарських помилок, проблема ідентифікації слів стає непростим завданням.

Потім, коли пропозиції розмічені, і слова локалізовані, починає роботу морфологічний аналізатор. Застосовуючи наявний словник і набір словозмінних правил, цей аналізатор доповнює кожне слово набором граматичних характеристик. Серед них відомі всім відмінки, роду і числа, а також деякі менш відомі граматичні значення. На етапі морфологічного аналізу також виникає ряд складнощів з обробкою друкарських помилок, особливо якщо вони допущені в закінченні слова, що грає, як відомо, ключове значення у визначенні набору граматичних характеристик. Ряд складнощів виникає при аналізі невідомих слів, а також запозичених з інших мов.

Наступний етап аналізу – синтаксичний. Найбільш популярним підходом до цього аналізу є побудова дерева синтаксичного розбору пропозиції. У процесі побудови дерева вихідне пропозицію перебудовується в деревоподібну структуру, що представляє собою аналог дерева синтаксичного розбору деякої формальної граматики. На

жаль, часто доступна в мережі інформація представлена у вигляді розмовної мови і, як наслідок, слабо піддається повному синтаксичному розбору, так як навіть досвідчений лінгвіст не зможе побудувати дерево синтаксичного розбору для подібних пропозицій. Як вирішення цієї проблеми існує альтернативний підхід до синтаксичному аналізу, а саме – частковий синтаксичний аналіз. При частковому синтаксичному аналізі повне дерево розбору не будується, а замість цього аналізатор зосереджується на пошуку заздалегідь визначених синтаксичних конструкцій у тексті. Такий підхід цілком прийнятний, тому що природна мова не має граматики у звичному математикам сенсі.

Природна мова не має термінальних і нетермінальних символів, правил виводу і не може бути віднесений ні до одного типу в ієрархії Хомського. Природні мови, як правило, мають досить вільні «правила» побудови граматичних конструкцій, слабо піддаються формалізації. В достатній мірі формалізуються лише деякі синтаксичні відносини:

1. Відносини узгодження і залежності при побудові словосполучень.

2. Валентності дієслів і віддієслівних дієприкметників.

3. Посилання на контекст (різні типи займенників).

4. Загальні правила побудови речень (просте речення, складносурядна, складнопідрядне речення та інші складні речення).

Заключним етапом аналізу тексту є семантичний аналіз результатів, отриманих на етапі синтаксичного аналізу. Для даного етапу не існує усталених моделей і підходів. У більшості систем роль семантичного аналізатора відіграє евристично реалізований модуль, що здійснює поставлене перед ним завдання. Ясно, що навіть за наявності

дерева повного синтаксичного розбору тексту, все одно складно реалізувати алгоритм, який видобуває корисну інформацію на основі отриманого дерева.

1.2.1. Оціночні показники

Розглянемо критерій оцінки «Повноти і точності», що еволюціонував із оцінок при отриманні інформації і наразі широко застосовується як оціночний показник.

Поняття «повноти» при відборі інформації показує відношення числа релевантних документів до числа всіх знайдених документів. В свою чергу «точність» визначається відношенням числа знайдених релевантних документів до загального числа релевантних документів. Формально «повнота» і «точність» визначаються так:

Крім того, для одночасного врахування повноти і точності існує їх комбінація (1.1), що має назву F-показник та має наступний вигляд:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{1.1}$$

де P – точність, R – повнота, а B – коефіцієнт балансу між оцінками.

Якщо $\beta = 1$, то обидва показники мають рівну вагу.

1.2.2. Загальна архітектура

Всі системи відбору інформації різняться між собою, але не дивлячись на це можна виділити узагальнену архітектуру, якій відповідає кажна система. Таким чином, покажемо п’ять головних компонентів типової системи відбору інформації [7].

Наведемо перелік компонентів системи відбору інформації.

- Токенізація та Тегування, що призначені для нормалізації вхідного тексту, розбиття його на лексеми і призначення семантичних тегів та за частинами мови.
- Аналіз речень, що використовується для визначення підмету, присудку, прийменників і т.д. за допомогою синтаксичного аналізу та розбору.
- Відбір, в якому визначаються цільові фрази. На відміну від перших двох компонентів, що є незалежними від тематики тексту, цільові фрази залежать від області знань, до якої належить текст.
- Злиття. Цей компонент застосовується для ідентифікації кореферентних відношень.
- Генерація шаблону, де створюються вихідні структури шляхом заповнення слотів знайденими фразами.

1.2.3. Підходи до побудови систем відбору інформації

Розглянемо методику проектування таких систем. Тут існує два основних підходи: Інженерія знань і Автоматичне навчання.

Підхід інженерії знань базується на залученні експертів, що є обізнаними не тільки в даній системі, а і у відповідній області знань. Людина повинна прочитати набір релевантних документів і вручну визначити правила поточної граматики. Продуктивність такої системи напряму залежить від людського фактору (обізнаності експертів, їх досвіду, та швидкості роботи). Даний підхід потребує залучення великої кількості людей та затрат часу [3].

На відміну від описаної методики, підхід автоматичного навчання не потребує залучення експертів для адаптації системи до нової області знань.

Роль людини зводиться до підготовки навчальної вибірки документів, до якої далі застосовується алгоритм навчання, що автоматично генерує правила відбору. Очевидно, що другий підхід працює значно швидше та є більш універсальним, однак потребує відносно великого об’єму навчальних даних [10].

1.3. Пошук значимої інформації і web-ресурси

В наш час, алгоритми відбору інформації не тільки використовуються для обробки текстів природної мови, а і широко застосовуються до web-сторінок. Ці сторінки містять багатоколонний текст, врізки, цитати, таблиці, інфографіку, коментарі читачів, різні типи рекламних блоків. Причому матеріал скомпонований характерним чином, що формує у читача необхідне думку і звертають його увагу на ключові моменти публікації. Саме ця компоновка елементів перетворює витяг тексту зі сторінки в проблему з нетривіальною рішенням [6].

Витяг даних (витяг інформації) – це процес отримання структурованих даних з слабоструктурованих або неструктурованих джерел. Класичний приклад – парсинг каталогу товарів з сайту інтернет-магазину, що найчастіше використовується для моніторингу цін, товарів, послуг (це не завжди абсолютно законно відповідно до угоди про використання магазина, але вельми затребуване), або для «брудних» і наближених до них SEO-технологій (створення сайтів-копій та ін.), соціологічні дослідження [2], створення агрегаторів, вхідний фільтр для пошукових систем. Витяг даних (data extraction) передує собою вилучення знань (data mining) [6] і є необхідним етапом попередньої обробки, від якого, в значній мірі залежить якість витягнутих знань. Навіть така проста задача, як витяг тексту з контенту

з web-сторінки пов'язана з певною складністю – потрібно відсікти «шапку» сайту, верхнє і вертикальне меню, блоки навігації, рекламні блоки, нижню частину сайту із зазначенням правовласника, студії-розробника, контактів, нижнього меню і інших елементів дизайну, що не містять, власне, корисної інформації. У даній статті під неструктурованими джерелами розуміються web-сторінки, що не мають значимої семантичної розмітки, а під слабоструктуровані – логічно взаємопов'язані фрагменти неструктурованого джерела (наприклад, таблиця), розміщені в межах однієї або кількох web-сторінок. Проблема вилучення тексту і слабоструктурованих даних з web-сторінок посилюється повсюдним порушенням і змішанням стандартів і рекомендацій по верстці web-сторінок. Викликано це може бути як цілями конкретного сайту (для газети – формування потрібного думки за допомогою спеціального оформлення і компонування матеріалу), так і простий недбалістю і помилками в коді сторінок сайту (може бути пов'язано з недостатньою кваліфікацією осіб, відповідальних за створення та / або функціонування сайту). Ситуація ще більше ускладнюється досконалістю web-браузерів, коректно відображають дані сторінки, однак спеціалізовані парсери зазвичай не здатні справляються з даною проблемою. Всі методи вилучення даних з сайтів можна розбити на три основні групи: Ручні методи – моніторинг цікавлять сторінок і переміщення необхідної інформації в базу здійснюється спеціальним оператором (людиною). Переваги підходу – висока якість витягання і низькі вимоги до кваліфікації оператора, недоліки – висока трудомісткість, загальна незначна швидкість роботи і людський фактор. На практиці кількість помилок може становити 2% і вище, що, часто буває неприйнятно високою величиною.

Напівавтоматичні методи – до цієї групи відносяться рішення, здатні витягувати інформацію після певної настройки. Сюди можна віднести як попередня розмітка оператором елементів сторінки в графічному інтерфейсі, так і більш складні випадки – наприклад, складання регулярних виразів або запитів на мові XQuery (один з предків даної мови поширений під ім'ям XPath). Проблема об'єднання даних полягає в наступному: витягувана конкретна одиниця інформації (запис) може бути представлена на сайті (або навіть сторінці) неодноразово, тому по закінченні виділення необхідно забезпечити видалення дублікатів. Це завдання легко можна вирішити по закінченні вилучення даних, однак парсинг в порівнянні з ухваленням рішення по закінченні більш ресурсномісткою операцією, тому оптимальніше дублікати виявляти до нього. У свою чергу, не завжди в принципі можливо визначити дублікати сторінок. Наприклад, в магазині радіодеталей існує безліч повністю однакових записів і тільки при відкритті конкретного товару можна визначити, що, наприклад, мова йде про різні партії від різних постачальників. Після здійснення даних етапів інформація з web-сторінки наводиться в систематизовану форму, придатну для імпорту в базу знань. Однак не завжди слід зосереджуватися на повністю автоматичному витяганням інформації з web-сторінок [10].

Розглянемо різні види тексту, що можуть бути оброблений, такі як неструктурований, структурований та частково структурований текст. Також опишемо посередники, що використовуються для обробки web-сторінок.

1.3.1. Різні ступені структурованості тексту

Системи відбору інформації для неструктурованого тексту застосовуються для обробки природної мови, тобто звичайних

інформаційних текстів, наприклад, новин. В цьому випадку необхідно попередньо виконати синтаксичну обробку і семантичне тегування [4].

Іншим можливим різновидом вхідних даних є так званий структурований текст. Структурований документ має визначений формат і зазвичай є результатом запиту до бази даних, тобто являє собою таблицю, що може мати різні описові елементи, зображення та ін.. Для пошуку конкретної інформації потрібно освоїти формат документу, в нашому випадку HTML-теги.

Частково структурований текст являє собою проміжний варіант між першими двома видами. Він не має чіткої структури а також визначеної граматики. Так як такий текст має неповні речення, від не може бути оброблений системами для природної мови, що базуються на синтаксичному аналізі. Аналогічно, системи, що розраховані на чітко визначений формат і його обмеження теж не можуть ефективно обробити вхідні дані такого роду [8].

1.3.2. Програми-посередники

Традиційний підхід полягає в отриманні даних з Web і відображенні отриманих даних в певний проміжний формат, а потім – у виконанні структурованих запитів вже до цих витягнутим даними. Підхід реалізується на основі механізму посередників (wrappers) – програм, які ідентифікують шукану інформацію у вихідному документі і відображають її в певний проміжний формат, наприклад, XML (eXtensible Markup Language, розширювана мова розмітки) [14].

Розробка та підтримка посередників вручну дуже трудомісткий процес (зокрема, через різнорідність і нерегулярності оброблюваної інформації). Тому різними колективами фахівців багатьох країн ведеться робота над засобами автоматизації цього завдання.

Для цієї мети запропонований ряд мов опису програм-посередників, які дозволяють значно знизити обсяг програмного коду, що розробляється вручну [13].

Подальша автоматизація ведеться в напрямку розробки алгоритмів, які автоматично генерують програми-посередники. Для цього проводиться аналіз набору документів, з якого належить витягувати інформацію, і, в ряді випадків, готуються навчальні приклади. Завдання вилучення є складною, оскільки потрібно витягти не тільки вид схеми даних, але також і пов'язану з ним семантичну інформацію (наприклад, тип конкретного елемента даних) [5]. Досягнення повної автоматизації в загальному випадку малоймовірно, і мова може йти про створення автоматизованих систем вилучення інформації з Інтернету.

Одна з основних труднощів, з якою стикаються розробники подібних систем, – це нерегулярність оброблюваної інформації. Зокрема, для автоматичної генерації витягає інформацію посередника потрібно підготувати набір документів, з яких посередник і буде витягувати інформацію. При цьому необхідно, щоб набір був відносно регулярним, тобто сторінки в цьому наборі повинні мати схожу структуру тих розділів (інформаційних блоків), яка містить видобуту інформацію. Дотепер існування цього набору передбачалося апріорі, питання його створення не розглядалися, що створювало певні труднощі при практичному використанні подібних систем, тому в даній роботі був проведений попередній аналіз деяких новинних сайтів, який дозволив уточнити алгоритми вилучення.

На відміну від систем обробки природної мови, посередники базуються на використанні розділових елементів, а не лінгвістичних правил. Синтаксична і семантична інформація використовується як

доповнення, особливо при роботі з частково структурованими джерелами.

Аналогічно до побудови систем відбору інформації, обгортки можуть створюватись вручну за допомогою експертів, або генеруватись автоматично в результаті роботи алгоритмів машинного навчання. Як вже було вказано в розділі 1.3.2 підхід ручного створення потребує залучення до роботи багатьох людей та багато часу. Також слід врахувати, що постійно створюються нові web-сторінки, а також може змінюватись структура вже відомих ресурсів. Отже, новий посередник повинен створюватись не тільки для нових сторінок, а і для тих, що змінились з часом. Очевидно, що необхідно використовувати автоматичну генерацію обгорток [13].

Важливим етапом відбору інформації є однозначна ідентифікація потрібної інформації із слабоструктурованих джерел і включення її в результуючий документ (наприклад, XML-документ, структура якого суворо визначена). Зазвичай для цього використовують інформацію про абсолютний шлях до потрібних елементів або прив'язку до їх змісту за допомогою правил на мові регулярних виразів, а також комбінацію цих способів. Обидва підходи можуть бути успішними для деяких видів джерел, але мало підходять для збору інформації із новин, тому неможливо прив'язатися до контексту новини, а структура новинних блоків на сайті також схильна до змін (змінюється кількість новинних статей, деякі блоки в якийсь момент можуть бути порожніми і т.д.). При цьому комбінація підходів, також буде неефективною. Тому в пропонованому методі застосовувалися складніші правила ідентифікації, що використовують, поряд з традиційними способами, прив'язку до типів елементів, що визначаються їх атрибутами і

враховують їх розмітку. На підставі цих правил будуються фільтри, які зв'язуються з конкретним джерелом [15].

Висновки до розділу

В даному розділі визначено поняття отримання та відбору інформації, інформаційного пошуку, описано задачі, що вони виконують та основні принципи їх роботи.

Розглянуто загальну архітектуру таких систем, структурні частини та основи їх побудови. Описані особливості застосування алгоритмів інформаційного пошуку та відбору інформації для обробки web-сторінок та природної мови.

Також виявлені проблеми, що є актуальними при виконанні даних задач та існуючи підходи до їх вирішення. Визначено, що слабким місцем є неоднорідність вхідних даних та велика кількість інформаційного шуму.

РОЗДІЛ 2. АНАЛІЗ ПРОЦЕСУ РОЗПІЗНАВАННЯ ЗНАЧИМОЇ ІНФОРМАЦІЇ

Обробка все більших і більших об'ємів інформаційних джерел і електронних документів, що знаходяться у вільному доступі в мережі Інтернет змушує дослідників сфокусувати свою увагу на проблемі відбору інформації. Запропоновано багато варіантів її вирішення, що використовують різні підходи для розпізнавання релевантної інформації і різні алгоритми машинного навчання для автоматичної побудови шаблонів відбору інформації для нових областей знань.

В даному розділі будуть описані алгоритми поперньої обробки даних в системах відбору інформації, що застосовуються для визначення ознак слів або токенів в заданих документах. Далі розглянуто різні обмеження, що можуть бути використані для знаходження цільових фраз. Також виконаємо огляд різних форматів представлення результатів роботи систем відбору інформації.

2.1. Попередня обробка

Перед використанням шаблонів або правил відбору інформації інколи потрібно застосувати засоби попередньої обробки для визначення в документах характиристик слів або токенів. Розглянемо синтаксичу і семантичну попередню обробку.

2.1.1. Синтаксична попередня обробка

Синтаксична попередня обробка даних зазвичай використовується для розбиття вхідних документів на маленькі частини та визначення синтаксичних характеристик окремих слів, фраз або токенів. Таким

чином, спочатку розглянемо деякі аналізатори речень для обробки природної мови.

На вхід системи надходить послідовність символів, і на першому етапі (лексичний аналіз) відбувається її розбиття на окремі слова і пропозиції. При цьому деякі послідовності символів (наприклад, тире і крапки) можуть трактуватися неоднозначно. Крім того, на етапі лексичного аналізу виникає завдання деобфускації – виявлення та виправлення навмисно спотворених слів. Типовим прикладом таких викривлень є заміна в спам-розсилках слова «drugs» (ліки) на «d.r.u.g.s» або «d-r-u-g-s».

На наступному етапі відбувається обробка окремих слів, яка часто зводиться до морфологічному аналізу – визначення характеристик слова і основний словоформи. Існує два підходи до проведення морфологічного аналізу. Перший (точна морфологія) передбачає побудову одного великого словника, що містить характеристики кожного слова. Цей підхід порівняно простий в реалізації, але має два важливих недоліки. По-перше, система буде коректно обробляти тільки слова, які є в словнику. По-друге, в багатьох мовах цей словник буде занадто великим.

Альтернативний підхід (неточна морфологія) до проведення аналізу слів полягає у використанні системи правил, згідно з якими по заданому слову визначаються його характеристики. Недоліком підходу є те, що він не завжди може гарантувати абсолютну точність результатів.

У завданнях повнотекстового пошуку і класифікації текстів не вимагається проведення повного морфологічного аналізу слів, а потрібна тільки перевірка того факту, що два зазначених слова

насправді є формами одного і того ж слова. Це може бути виконано шляхом їх лематизації (приведення до основної словоформи) або стеммінга, який полягає у виділенні певної незмінної частини слів. Однак морфологічний аналіз, лематизація і стеммінг не завжди здатні визначати споріднені слова, наприклад «безпека» і «захист». Завдання визначення споріднених слів вирішують шляхом використання спеціальних словників-тезаурусів, що представляють собою орієнтовані графи, у яких вершини відповідають словам, а дуги – семантично пофарбованим зв'язків між словами. Близькість двох слів визначається на основі найкоротшого шляху, що з'єднує дві відповідні вершини графа. Якщо необхідно враховувати контекст слів, то завдання значно ускладнюється, і її слід віднести до семантичної обробці тексту. Існують автоматизовані способи визначення пов'язаності слів на основі частоти їх спільної зустрічальності або ступеня збігу їх контекстів вживання.

При синтаксичному аналізі послідовність слів вихідного тексту перетвориться в деревоподібну ієрархію, у якій листя відповідають окремим словами, вузли – групам слів, а дуги – взаємозв'язкам між словами і групами слів. Це перетворення здійснюється на основі заданої граматики мови, яка по суті є фіксованим набором правил. Використання граматик пов'язано з очевидними труднощами – для природної мови складно розробити систему правил, що її описує, причому це особливо важко для мов зі складною морфологічної моделлю і довільним порядком слів (таких як російська). Крім того, переважна більшість написаних людиною текстів містять помилки або друкарські помилки. З цієї причини будь-яка граматика може виявитися

непридатною, а спроби врахувати всі можливі варіанти помилок результату не дадуть.

В основі більшості систем синтаксичного аналізу тексту лежать підходи, що припускають використання різних варіантів граматик. Спроби порівняння синтаксичних аналізаторів проводилися неодноразово, і виявилося, що існуючі системи занадто різнорідні, а результати їх роботи важко привести до спільного знаменника. Мало того, за кадром залишається вельми серйозна проблема – синтаксичний аналіз сам по собі не має практичної цінності, а є лише проміжним етапом вирішення більш загальної задачі. При розробці та оцінці модулів синтаксичного аналізу тестові дані повинні бути прикладами вхідних даних конкретної системи. Припустимо, для кадрового агентства створюється система обробки резюме претендентів – тоді модулі синтаксичного аналізу повинні тестуватися саме на таких текстах.

Положення синтаксичного аналізу серед інших методів попередньої обробки текстів двояко. З одного боку, синтаксична структура пропозиції досить точно визначає зв'язки між словами, що необхідно в ряді практичних додатків, таких як машинний переклад або вилучення інформації. З іншого, деякі завдання (наприклад, повнотекстовий пошук або класифікація текстів) вирішуються і без синтаксичного аналізу, без слідування традиціям і глибокого аналізу тексту.

У завданнях, що вимагають розуміння сенсу, подальша обробка тексту полягає у виявленні та вирішенні слів-посилань. Найпростіший спосіб виконання даної операції полягає у використанні системи правил. Наприклад, слово «який» зазвичай пов'язане з останнім

використаним іменником чоловічого роду. У більш складному випадку для вирішення посилань виду «цей факт» необхідно враховувати контекст. У міру ускладнення оброблюваних посилань система поступово рухається до побудови «локального» тезауруса документа. Далі будь-яке слово потрібно оцінювати як з точки зору локального, так і з точки зору глобального тезауруса.

У підсумку виходить, що дозвіл посилань є окремим випадком розуміння сенсу, який залежить від локального тезауруса. Почасти вірно і зворотне – локальний тезаурус залежить від змісту тексту, при цьому для коректного розуміння окремих фраз іноді доводиться задіяти цілі понятійні шари мови. Наприклад, згадка теорії відносності автоматично «підключає» відповідну термінологію і понятійний апарат.

Далі надамо опис систем розбору web-сторінок (парсерів), що використовуються для сегментації текстового вмісту web-сторінок. В наступній частині розділу розглянемо кілька простих алгоритмів просвоєння синтаксичних тегів.

Розбір речень

Попередня синтаксична обробка зазвичай необхідна для інформаційних систем, що працюють к природною мовою, наприклад, новостні статті, з метою виокремлення відношень між словами і фразами вихідного речення.

Концептулаьний аналізатор речень CIRCUS використовується в системі AutoSlog. Він приймає на вхід речення і розбиває його на частини, далі розбирає ці частини за синтаксичними складовими,такими як підмет, присудок, додаток і т.д.

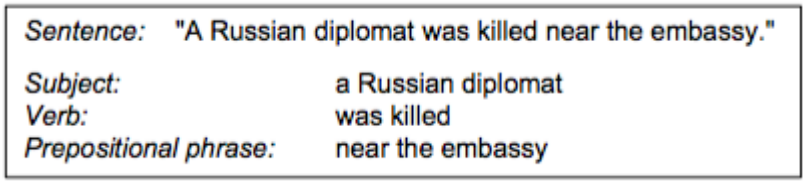


Рис. 2.1. Приклад синтаксичного аналізу з використанням CIRCUS

На рис. 2.1 зображено приклад роботи синтаксичного аналізатора CIRCUS. Як бачимо, система розбила речення на підмет, присудок і прийменникову групу.

Система CRYSTAL працює схожим чином до AutoSlog та використовує аналізатор речень BADGER для відбору інформації з тексту природною мовою.

Базова версія BADGER працює аналогічним чином до CIRCUS та розбиває речення на синтаксичні складові. Також існує модифікація даного аналізатора, яка додаткого визначає частини, що відносяться до підмета, присудка, прийменникової групи тощо.

Необхідно відзначити, що обидві версії BADGER і CIRCUS не генеруються дерево розбору, а тільки текстовий результат переліку складових наданого тексту, тобто всі складові знаходяться на одному рівні.

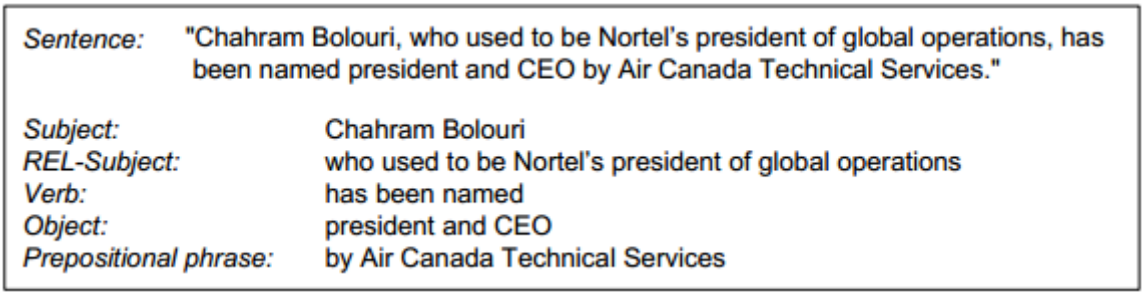


Рис. 2.2. Синтаксичний розбір за допомогою аналізатора BADGER

Приклад розбору речення на частини зображено на рис.2.2. Як бачимо, речення розбито на такі частини: підмет, присудок, додаток, прийменникова група і частина, що відноситься до підмета.

Система WHISK також використовує BADGER, та є першою системою, що працює з усіма видами текстів: неструктурованим, частково структурованим та строго структурованим. Для обробки останніх двох видів необхідна тільки мінімальна попередня обробка, а BADGER використовується для аналізу неструктурованого тексту.

Синтаксичний аналізатор на основі граматики зв'язків також може використовуватись за розбору речення. На відміну від CRYSTAL і BADGER, даний аналізатор відображає відношення між словами і фразами, на просто позначає окремі частини заданого речення. Синтаксичний аналізатор на основі граматики зв'язків розрізняє багато різних видів зв'язків між частинами речення. На рис.2.3 зображено частину розбору тестового речення даним аналізатором. Слово "Corp" за допомогою зв'язку S з'єднується зі словом "said", тобто можна зробити висновок, що «Corp» є підметом відносно присудку «said».

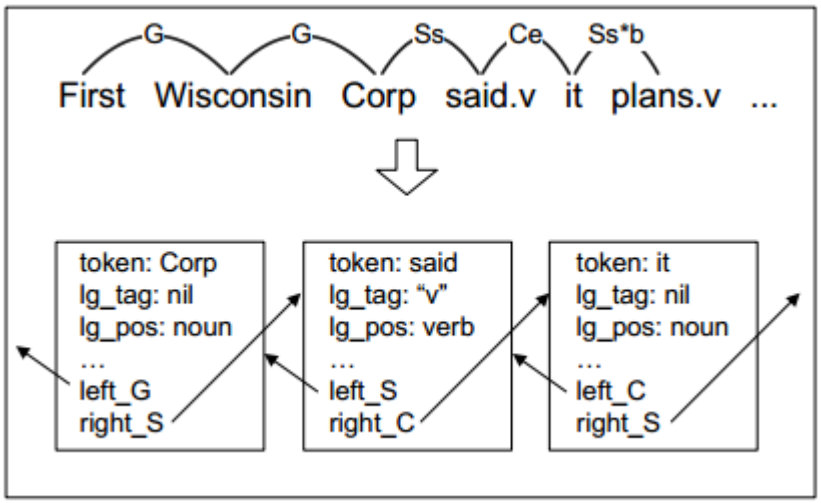


Рис. 2.3. Результат роботи синтаксичного аналізатора на основі граматичних зв'язків

Відповідно до знайдених зв'язків, ми можемо зробити висновки щодо ознак кожного члена розбору, та використати їх для відбору та навчання.

Аналізатори web-сторінок

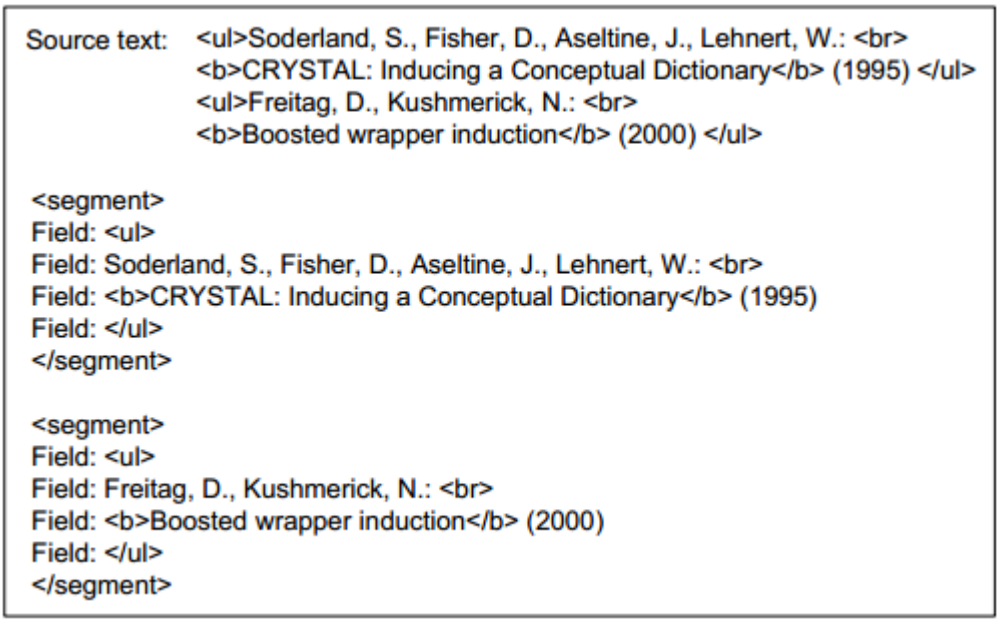
Для знаходження релевантної інформації з слабко структурованого та структурованого тексту система обробки приймає на вхід HTML-документ. В деяких системах, такий документ спочатку повинен бути розділений на маленькі частини. Так як HTML-документ не є простим текстом природною мовою, він не може бути оброблений за допомогою звичайного аналізатора речень. Отже, необхідно здійснювати розбір HTML-структури документу.

Для розбору слабко структурованих документів за допомогою системи CRYSTAL додатково застосовується парсер web-ресурсів Webfoot.

Webfoot починає роботу з розбиття HTML-документу, яке відбувається на основі особливостей структури макету. Одна частина має являти собою групу логічно пов’язаної інформації, що відокремлюється від інших за домопогою розділювачів першого рівня. Далі виконується подальше розбиття на менші частини застосовуючи розділювачі вищих рівнів. Процес триває до тих пір, поки розмір частин не досягне розміру в 20 слів без врахування HTML-тегів. Далі відбувається розбиття на поля.

На рис. 2.4 зображено приклад розбиття вихідного тексту на частини і поля за допомогою парсера Webfoot. Кожна частина, що містить перелік авторів і заголовок статті поділяється на окремі поля.

Подібно до CIRCUS і BADGER, парсер Webfoot генерує результат не деревовидної структури, тобто всі елементи знаходяться на одному рівні.


Source text: Soderland, S., Fisher, D., Aseltine, J., Lehnert, W.:
CRYSTAL: Inducing a Conceptual Dictionary (1995)
Freitag, D., Kushmerick, N.:
Boosted wrapper induction (2000)

Field:
Field: Soderland, S., Fisher, D., Aseltine, J., Lehnert, W.:
Field: CRYSTAL: Inducing a Conceptual Dictionary (1995)
Field:

Field:
Field: Freitag, D., Kushmerick, N.:
Field: Boosted wrapper induction (2000)
Field:

Рис. 2.4. Розбиття за допомогою парсера web-сторінок Webfoot

Методика ієрархічних каталогів

Система STALKER представляє HTML-документи за допомогою формалізму ієрархії каталогів. На відміну від Webfoot, такий опис має деревовидну структуру, коренем якої є весь документ, а кожний листок – елемент з набору, що містить частину значимої інформації, яка може бути відібрана. Внутрішня вершина визначає набір, в якому елемент може бути листком або іншим вкладеним набором.

На рис. 2.5 зображено ієрархічний опис результатів пошуку з web-ресурсу. Сторінка містить перелік готелів, кожен з яких представляється як набір з п’яти елементів: назва, рейтинг, ціна, адреса і відстань, яка є кладеним списком, що складається с місця розташування і кілометражу.

Вміст кожної вершини являє собою повну послідовність токенів. Наприклад, вміст кореневого елементу – це загальна послідовність токенів, і кожна дочірня вершина містить в собі деяку її підмножину батьківського набору. Для того щоб отримати дані, що містяться в деякій вершині дерева для заповнення певного слоту, система визначає

шлях від кореневого елементу до цільового листка і ітеративно знаходить кожну вершину шляху за допомогою застосування правил відбору то батьківсього елементу. Таким чином, система STALKER знаходить релевантну інформацію за допомогою кількох простих правил, замість того, щоб застосовувати одне складне.

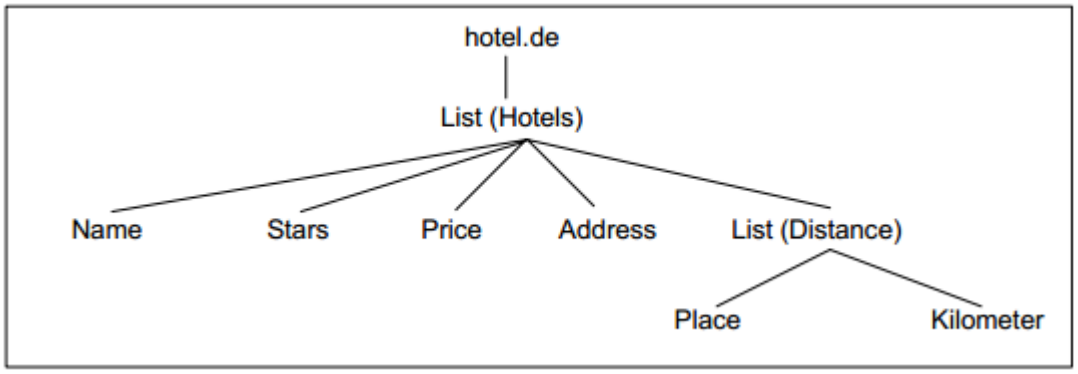


Рис. 2.5. Ієрархічний опис сторінки за результатом пошукового запиту

Система Lixto використовує Java Swing Parser для побудови дерева об'єктної моделі документа за вхідною HTML-сторінкою. На відміну від попедньо розглянутої системи, яку використовує STALKER, чиї листки описують елементи набору даних, парсер Swing є незалежним від конкретної області знань і базується на використанні тільки HTML-ієрархії. З іншого боку, дана система відрізняється в попередньої також вмістом вершин. Замість простої послідовності токенів вершини в даній системі є набором пар атрибут-значення, які містять:

- атрибут, що відображає ім'я вершини;
- атрибути тегів, що належать до даної вершини. Наприклад, *bgcolor* для вершини тегу <body>, а також *border* і *width* для вершини <table>;
- спеціальний атрибут «elementtext», який описує вміст вершини.

Для вершин, що не є листками дерева останній атрибут буде містити набір всіх листкових дочірніх елементів.

Приклад результуючого дерева розбору з кількома описами вершин зображено у Додатку Б. Пари чисел в дужках вказують на початок і кінець позиції конкретного елемента. Деякі листки позначені елементом <content>, для якого такі теги як <b>, <href> і т.д. розглядаються як атрибути, а значення атрибуту містить текст, що знаходиться всередині тегів <td> і <p>.

Синтаксична розмітка

Розглянемо алгоритми синтаксичної розмітки, що визначають прості синтаксичні ознаки для окремих частин незалежно одна від одної, однак такий підхід не відображає відношення між ними. Дані алгоритмі працюють більш стабільно та швидко і знаходять синтаксичні ознаки, які є корисними при відборі інформації при обробці тексту з граматикою або без.

Система RAPIER використовує розмітку за частинами мови, яка робить класифікацію кожного слова вхідного тексту окремо, тобто чи є це слово іменником, займенником, дієсловом чи прийменником.

Такий же алгоритм використовується в системі KnowItAll для знаходження як окремих словосполучень, так і їх наборів.

Система SoftMealy потребує попередньої обробки вхідних даних, тобто розбиття HTML-документа на частини. Кожна частина це рядок v, що віднесено до класу t і позначається як t(v). Наведемо приклади можливих класів в даній системі:

- рядок містить тількі великі літери – CAlph(KPI);
- рядок містить тільки маленькі літери OAlph(diploma);
- рядок починається з великої літери C1Alph(Kyiv);
- рядок складається тільки з цифр Num(23);
- пунктуація Punc(!);

- спеціальні символи NL(1), Tab(3);
- HTML-теги – HTML(<b>).

Також класи, що зазначені вище, можуть групуватися в загальні класи, такі як «Слово» а бо «Не слово», що дає можливість суттєво спростити правила відбору. Відповідно, перші 4 класи будуть належати до класу «Слово», а наступні три – до «Не слово».

2.1.2. Семантична попередня обробка

Додатково до синтаксичної обробки використовується семантична попередня обробка для визначення семантичного класу кожного окремого слова чи фрази. Цей процес застосовується в більшості систем відбору інформації, що займаються обробкою вільного тексту і частково структурованого тексту.

Слова-тригери

Словом-тригером називається деяка особлива точка, що використовується для активації конкретного правила відбору. Вона задається за допомогою визначення поняття вузла в системі AutoSlog. Додатково задається набір допустимих станів для визначення конкретних форм слова, тобто, наприклад, пасивна чи активна форма дієслова. Щоб правило спрацювало, слово-тригер повинно бути в одному із зазначених допустимих станів.

Як зображено на рис.2.7 тут тригером виступає слово “killed” та зазначається, що воно має бути в пасивній формі, отже фрази “was killed”, “were killed”, “has been killed” також задовольняють правило.

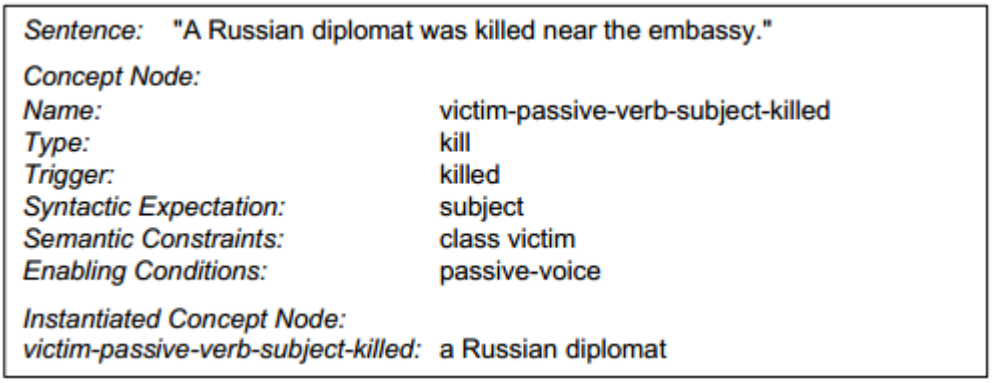


Рис. 2.7. Застосування слова-тригера у системі AutoSlog

У системі KnowItAll кожне правило являє собою не одне, а набір слів-тригерів, які застосовуються як пошукові запити для знаходження web-сторінок, що містять ці слова.

Слова-тригери геруються відповідно до універсального шаблону даної області знань, що використовується у правилі, а також мітки цільового класу.

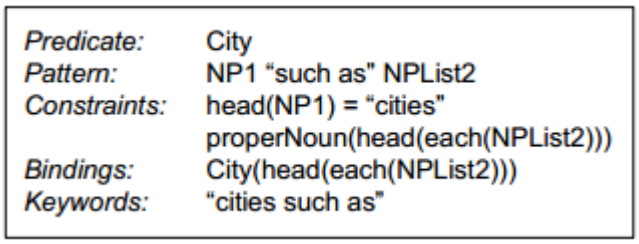


Рис. 2.8. Правило відбору в системі KnowItAll

Наприклад, клас City може позначатися, наприклад, як «city» і «town». На рис. 2.8 зображено одне з можливих правил відбору міст, на основі загального шалону NP1 “such as” NPList2. Тригер-слово "такі міста, як" складається з позначення «місто» у формі множини і контекстного рядку "такі як" в шаблоні. Аналогічно, «towns such as» може бути іншим пошуковим запитом для знаходження сторінок, но містять перелік міст.

Отже, слова-тригери системи KnowItAll використовуються не тільки для знаходження інформативних сегментів, що містять цільові

дані, в документах, як це відбувається в системі AutoSlog, але також і для автоматичного пошуку самих документів.

Семантична розмітка

Семантичні класи є специфічними для кожної області знань і можуть задаватися як семантичним словником, так і визначатися користувачем. Наприклад, в тематиці «новини про тероризм», можуть бути такі семантичні класи: «Ціль», «Жертва», «Злочинець» тощо, а корпоративна тематика матиме класи: «Ім'я людини», «Корпоративне повідомлення», «Організація» тощо.

За допомогою семантичних класів правило відбору або шаблон можна адаптувати для знаходження групи слів, що матиме такі ж семантичні ознаки, як і окреме слово. Використання таких класів дозволяє здійснити узагальнення правила чи шаблону, тобто зробити його більш універсальним.

Попередня семантична розмітка є одним з етапів розробки AutoSlog, так як і CRYSTAL. Різниця між процесами класифікації цих систем в тому, что що всі семантичні класи AutoSlog визначені на одному рівні, в той час як CRYSTAL може працювати з ієрархічною структурою класів, тобто класи можуть містити набір підкласів і в той же час бути підкласом іншого класу. Ієрархічна система дозволяє здійснювати індукцію правил за рахунок розвантаження рівнів семантичних класів.

Системи REPIER і SRV можуть використовувати тематико-незалежний словник, що має назву WordNet та застосовується для здійснення семантичної розмітки. Даний словник має велику базу даних слів, а також забезпечує семантичну ієрархію.

Семантична розмітка не є обов’язковим етапом, але вона допомагає при попередній обробці у системі WHISK при обробці природної мови. Система WHISK визначає такі специальні семантичні класи як «Цифра» і «Число», а також у користувача додатково є можливість визначати семантичні класи у вигляді списку слів, що мають однакове семантичне значення для заданої області знань. Семантичний клас <Bdrm> в оголошеннях про орендну плату може бути визначається як диз'юнкція різних скорочень або вказівок слова "bedroom":

<Bdrm> = (brs | br | bds | bdrm | bd | bedrooms | bedroom | bed)

На відміну від CRYSTAL і RAPIER, система WHISK не має семантичної ієрархії.

2.2. Обмеження

Розглянемо різні обмеження, що можуть бути застосовані для знаходження релевантної інформації. Для цього використовуються синтаксичні, семантичні обмеження, обмеження за певним словом, групою символів, і негативні обмеження. Далі будуть описані видимий і невидимий опис границь цілей та їх контекстів.

2.2.1. Синтаксичні ознаки

Синтаксичні мітки, що забезпечуються синтаксичним аналізатором, відіграють важливу роль в пошукових шаблонах, що використовуються для відбору інформації з граматичного тексту. В даному розділі ми спочатку розглянемо системи що використовують синтаксичні складові, такі як суб’єкт, об’єкт і т.д. для знаходження цільового тексту. Далі системи, що використовують лише обмежену синтаксичну інформацію, таку як капіталізація та частина мови.

Синтаксичні складові

Системи AutoSlog і CRYSTAL працюють на основі використання синтаксичних складових при обробці природної мови для пошуку значимої інформації. Заповнювач слоту, що вони визначили має бути синтаксичним полем, в якому визначені конкретні обмеження. На відміну від згаданих систем, для WHISK, що також застосовується для обробки природної мови, використання синтаксичних складових є опціональним. Допускається використання синтаксичних міток, які є результатом роботи BADGER, або ж вони можуть ігноруватися у WHISK.

На рис. 2.9 зображено приклад правила, що використовувався у WHISK для області знань «управління персоналом». Синтаксичний тег «{PP» потребує, щоб заповнювач слоту Organization містився в прийменниковій групі, а синтаксичні теги для заповнювачів Person_In і Position ігнорувалися. Додатково, синтаксичний тег «@passive» відповідає дієслову в пасивній формі у вхідному реченні.

В системах AutoSlog і CRYSTAL застосовуються різні пошукові шаблони для ідентифікації певної фрази, синтаксичні компоненти якої відрізняються в різних заданих текстах. Однак, у системі WHISK є можливість частково вирішити це питання за допомогою ігнорування деяких синтаксичних тегів.

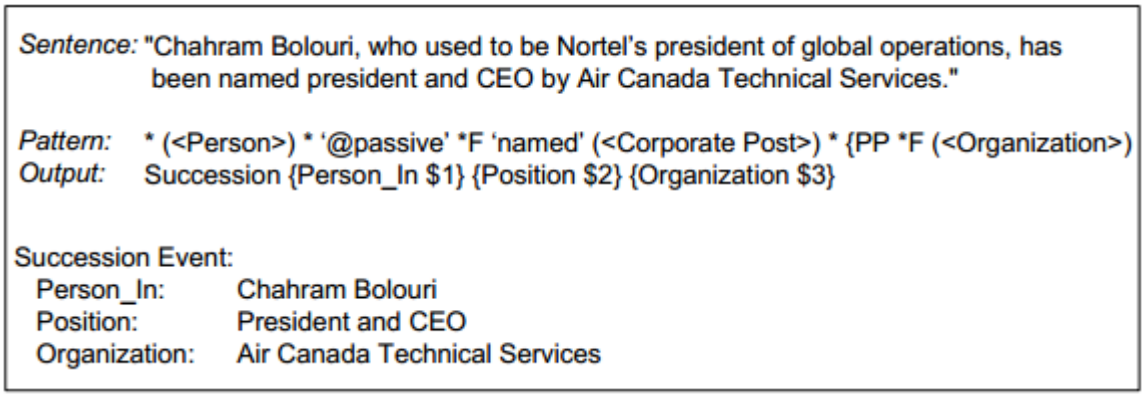


Рис 2.9. Правило відбору, використане для обробки природної мови у системі WHISK

Прості синтаксичні ознаки

Системи SoftMealy, STALKER і BWI застосовують класи токенів для опису простих синтаксичних особливостей. Токеном може бути число, HTML-тег, пунктуація, спеціальний символ тощо. Синтаксичні обмеження для всіх перелічених систем використовуються тільки для представлення роздільників, що визначають границі заповнювача слоту, а не його вміст.

Сепаратори, що використовуються у системі SoftMealy представляються у формі послідовності помічених токенів і класів токенів, в той час як система STALKER може зберігати класи токенів у «пам'ятках», які використовуються як аргументи фукцій, наприклад:

- SkipTo(landmark) зазначає, що всі токени, що містяться в пам'ятці повинні бути пропущені;
- SkipUntil(landmark) подібним чином до попередньої функції показує токени, що повинні бути пройдені. Однак, зазначена пам'ятка не пропускається;
- NextLandmark(landmark) використовується для підтвердження відповідності наступної пам'ятку. Подібно до попередньої функції, сама пам'ятка не повинна розглядатись.

Синтаксичні обмеження, що використовуються у системі RAPIER є мітками частин мови, які повинні бути дотримані для успішного відбору. На відміну від SoftMealy і STALKER, мітки за частинами мови можуть застосовуватися в обох шаблонах початку та кінця заповнювача слоту, а також і в шаблоні самого заповнювача. Крім того, шаблон може містити елементи, що описуються для визначення конкретного

слова або символу, або переліку шаблонів, для якого може зазначатися максимальна довжина N, щоб обмежити кількість слів чи символів, що можуть міститись в шаблоні. Зазначимо, що всі описані елементи можуть застосовуватися як для заповнювачів слоту, так і для контексту.

Система SRV визначає предикати для знаходження цільового тексту, що описують особливості, які повинні мати сам заповнювач слоту, а також його контекст, як і в RAPIER. SRV працює з токен-орієнтованими особливостями у двох категоріях, описаних далі.

Першою категорією є прості ознаки, що визначають конкретне значення токену, та зазвичай приймають значення true або false. Така ознака може бути певною частиною мови, наприклад, іменник або дієслово, або довжина слів, ща також використовуються системою RAPIER. Потрібно зазначити, то в SRV подібно до SoftMealy і STALKER, також можуть розрізнятися такі типи токенів, як числа, пунктуація, орфографічні ознаки тощо. Наприклад, capitalized(token1) = false.

Відповідно до пошуку інформації в HTML-документі, його особливості розширюються, тобто враховуються такі ознаки, як IN, і його похідні IN_TITLE, IN_B, що вказують на те, що шукане значення повинно бути всередині зазначених тегів, тобто в даному випадку <title> і <b> відповідно.

Другою категорією є знаки відношень, що вказують на зв’язок одного токена з іншим. Наприклад, якщо token2 з’являється у тексті одразу ж за token1, відношення між ними буде мати вигляд:

next token(token1) = token2

pre token(token2) = token1

Аналогічно, такі ознаки відношень як table_next_col, table_pre_col і т.д. можуть бути додані для ідентифікації попередньої та наступної колонки деякої таблиці, що є частиною HTML-документа.

На відміну від RAPIER і SRV, алгоритм KnowItAll допускає використання обмежень за частинами мови тільки для заповнювачів слотів. Такі обмеження можуть знайти мітки за частиною мови, що зазначаються для усієї фрази, головного її члена у слоті, або головного члена кожної фрази, якщо слот містить кулька заповнювачів. Наприклад, обмеження properNoun(head(each(NPList2))), що зображено на рис. 2.8 означає, що головний член кожної фрази всередині NPList має бути власним ім'ям.

Умови за атрибутами HTML-тегів

На відміну від синтаксичних обмежень, розглянутих вище, що застосовуються до слів та словосполучень у вхідному документі, деревовидний алгоритм відбору Lixto визначає умови для опису шуканих пар атрибут-значення для конкретних вершин у дереві розбору HTML-документу.

Умова за атрибутом є триплетом, що складається з наступних елементів:

- ім'я атрибуту, що має міститься у вершині;
- шукане значення у вигляді рядку тексту або регулярного виразу;
- параметр *exact*, *substr* або *regvar*, які визначають, що атрибут повинен повністю співпадати з рядком, або містити його, або відповідати зазначеному регулярному виразу відповідно.

Набір таких умов може включатися в опис шляху до елементу (селектор) разом із шляхом дерева.

Наприклад, селектор вигляду (*.td22, [(bgcolor,"yellow", *exact*), (width, "%", *substr*)]) відповідає всім коміркам таблиці, що мають жовтий фон і ширину, що визначена у відсотках.

Описані селектори позначаються як «epd» і використовуються у предикатах Elog, таких як:

- Предикати визначення ознак відбору, для пошуку значень, що будуть відібрані. Наприклад, subelem(s, epd, x) приймає значення true, якщо s є дочірнім елементом частини дерева, коренем якої є x, з додатково визначеним селектором.
- Предикати умови за контекстом, для пошуку визначених елементів, що повинні або не повинні знаходитись до чи після шуканого об’єкта. Наприклад, notbefore(s, x, epd, d) приймає значення true, якщо немає такого елемента, який відповідає селектору і передує області дерева x не більше ніж на відстані d.
- Предикати внутрішніх умов, для визначення елементів, що повінні або не повинні міститись в шуканому об’єкті. Наприклад, contains(x, epd, y) приймає значення true, якщо піддерево *y*, корінь якого відповідає селектору epd, міститься в області дерева *x*.

Отже, як і CRYSTAL, RAPIER, SRV і WHISK, так і обмеження можуть бути використані для знаходження цільового об’єкту, а також навколишнього контексту.

2.2.2. Семантичні обмеження

У даному підході аналізовані документи є інформаційними об'єктами і описуються в онтології деяким поняттям, наприклад, поняттям Документ. Текст, який представляє утримання об'єктів класу

Документ (або будь-якого іншого класу, що описує текстовий ресурс), аналізується з метою отримання значимої інформації або контенту.

Контент документа являє собою набір інформаційних об'єктів і їх зв'язків, опис яких зустрілося в тексті документа. Для того, щоб зв'язати контент з документом вводиться спеціальне ставлення, що дозволяє вказувати для кожного екземпляра відносини (в тому числі і атрибутного) індекс документа, в тексті якого він знайдений. (При цьому знайдені ІО прив'язувати не потрібно, тому ідентифікатором ІО в тексті завжди виступає наявність хоча б одного атрибутного відносини, що зв'язує сам об'єкт з його найменуванням.)

При аналізі документа використовується формальне подання структури його тексту, яка залежить від типу або жанру документа.

Відповідно до [2] текст в електронній формі має принаймні три рівні формальної структури – фізичний, логічний і жанровий. Перший являє презентацію тексту на сторінці, наприклад, за допомогою тегів таблиці стилів. До другого рівня відносяться такі елементи як текст, абзац, рядок, пропозицію і т.п. Третій рівень представлений розбиттям тексту на жанрові частини, наприклад, текст ділового листа [3] має наступні жанрові розділи: заголовок (відправник, адресат, резюме та звернення), основний розділ (текст листа, примітки і додатки) і підпис.

Будь-яку формальну структуру тексту будемо називати сегментом і описувати за допомогою маркерів. Маркер задається списком альтернативних елементів m, де елементом mi може бути:

1) будь-який символ або рядок;

2) лексичний об'єкт, отриманий після лексичного аналізу, що задається класом (семантичний клас, граматичний клас, службовий клас), або конкретним ідентифікатором, або назвою (для слова – це

нормальна форма; для лексичної конструкції, описуваної шаблоном – це назва шаблону);

3) сегмент іншого типу.

Побудова сегмента здійснюється на підставі наступних обмежень:

- Single – сегмент не перетинається з сегмента ми того ж типу, окремий випадок цього обмеження – відсутність вкладеності;
- Min – вибирається мінімальний з можливих сегментів на даній ділянці;
- Max – вибирається максимальний з можливих сегментів на даній ділянці.

Ієрархії класів понять і задані на них семантичні відносини дозволяють уявити структуру висловлювання з предметної області у вигляді факту, безліч яких становить пропозіціональное зміст документа.

Інформаційний зміст системи представлено об'єктами або екземплярами понять онтології, тому в даному підході завданням аналізу є отримання тільки тих фактів, які дозволяють виявити такі об'єкти (і їх властивості і відносини) з тексту. Тобто можна розглядати факт, як засіб подання контенту документа в ІС. Декларативне опис структури факту і умов його виявлення будемо називати схемою факту.

Схема факту задає безліч аргументів факту (для нашої системи будемо поки розглядати тільки унарні і бінарні факти), де аргументом може бути:

- поняття онтології;
- об'єкт або клас Тезауруса;
- факт (тип факту);
- інформаційний об’єкт документа, текст якого аналізується.

Для того, щоб застосувати схему факту її аргументи повинні задовольняти заданим обмеженням на сумісність аргументів. Виділяються семантичні та структурні обмеження.

З точки зору результату виділяються динамічні і статичні схеми. В результаті застосування динамічної схеми для кожного факту створюється новий об'єкт, поява якого може послужити підставою для застосування іншої схеми. Результатом застосування статичної схеми є зміна вже існуючого об'єкта, що виступає в якості одного з аргументів, або інформаційний об’єкт документа. У загальному випадку результатом є безліч об'єктів або відносин знайдених на заданій ділянці тексту.

Семантичне обмеження – це обмеження на семантичний клас аргументів факту. Обмеження явно задає пару поєднуваних елементів, де елемент або клас (успадковані від класу аргументу), або словниковий термін, або факт (якщо аргумент є фактом).

Для кожної схеми Факту, описаної в онтології, може бути згенерована таблиця семантичних поєднань, яку повинен заповнювати експерт. Призначення цієї таблиці полягає в наступному:

- звуження безлічі варіантів поєднань текстових одиниць;
- облік впливу аргументів один на одного (наприклад, уточнення семантичного класу);
- уточнення атрибутів результуючого об'єкта.

Структурні обмеження

Крім семантичних обмежень, необхідно враховувати обмеження інших мовних рівнів, такі як синтаксичні та жанрові обмеження.

Для кожної схеми факту ми будемо задавати додаткові умови на її аргументи:

- умова на сегмент, тобто в рамках сегмента якого типу повинні розташовуватися аргументи;
- взаємне розташування аргументів в тексті (контактність, пре- і постпозиції, пріоритетність позиції при багатоваріантності вибору);
- наявність синтаксичних умов (валентності термінів, прийменниково-відмінкові сполучення тощо);
- правила освіти поєднання (однорідність, проективної, максимальна зв'язність).

Перевірка синтаксичної сумісності може здійснюватися або зіставленням граматичних ознак термінів, або побудовою локального синтаксичного дерева залежностей [4].

2.3. Отримання web-структур

Якщо в задачі витягу Web-контенту з метою його збереження в базі даних інтерес викликає внутрішня структура Web-документа, то в задачі витягу Web-структур, насамперед, інтерес викликає структура гіперпосилань в межах Web-мережі (міждокументна структура). Для вирішення цієї завдання вимагається подання документів і відношень між ними, що враховує гіперпосилання. У зв'язку з цим такі моделі мають відмінності від описаних раніше підходів у поданні окремих документів. Гіперпосилання моделюються з різним рівнем деталізації залежно від застосування моделі. У найпростіших моделях гіперпосилання можуть бути представлені як спрямований граф:

$$G = (D, L), \tag{2.1}$$

де D – це набір вузлів, документів або сторінок; L – набір посилань.

Залежно від інформації, яка враховується при побудові моделі, вони можуть бути більш-менш точними. Грубі моделі можуть не враховувати текст, супроводжуючий гіперпосилання. Найбільш точні моделі враховують не тільки текст документа (точніше модель документа), а й розподіл термів в документі залежно від його належності певної предметної області. У деяких моделях вихідні документи представляються як послідовність термів з вихідними від них гіперпосиланнями. Таке подання може бути корисно, щоб встановити специфічні відносини між певними термами і гіперпосиланнями (а не між документами цілком).

У деяких випадках доцільно розглядати розподіл термів документів в залежності від їх тематики. Наприклад, розподіл термів документа з області "велосипедного спорту" повністю відрізняється від такого ж документа, що відноситься до теми "археологія". На відміну від журналів з археології або велосипедного спорту, Web-сторінки не ізольовані і можуть посилатися на документи з інших тем. Якщо необхідно враховувати зв'язки не тільки між окремими документами, а й між темами, то необхідно включати в модель також зв'язки між темами.

Можна виділити три основні завдання, які можуть бути вирішені на підставі аналізу Web-структури:

- оцінка важливості структури Web (документа або вузла), і вплив їх один на одного;
- пошук Web-документів з урахуванням гіперпосилань, що містяться в них;
- кластеризація структур для їх можливого явного об'єднання.

2.4. Оцінка важливості web-структур

Вирішення першого завдання в основному базується на методах побудови соціальних мереж. Соціальна мережа будується для вимірювання значущості того чи іншого індивідуума на підставі його зв'язків з іншими індивідуумами. Така соціальна мережа представляється у вигляді графа, в якому вузли відповідають індивідуумам, а дуги – їх відношенню між ними. При цьому граф є зваженим, тобто кожній дузі призначається вага, що відповідає силі впливу одного індивідуума на іншого. На основі аналізу побудованого таким чином графа обчислюється значимість кожного вузла (індивідуума) на підставі інформації про дугах і їх вагах, вхідних і вихідних від вузла. Як нескладно побачити, модель соціальної мережі дуже близька моделі, відповідній структурі Інтернет-простору. У зв'язку з цим і методи, що застосовуються до соціальних мереж, можуть використовуватися для вирішення тих же завдань у мережі Інтернет.

Починаючи з 1996 р, методи аналізу соціальних мереж були застосовані до графа Web з метою ідентифікувати найбільш значущі сторінки, відповідні запиту користувача.

Л. Катц (L. Katz.) запропонував для обчислення значущості Web-структур використовувати шляхи, засновані на вхідних посиланнях. Відповідно з цією ідеєю кількість шляхів довжиною r від вузла i до j позначається P_{ij}^{r} [11]. Загальна кількість шляхів різної довжини обчислюється за формулою:

$$Q_{ij} = \sum_{r=1}^{\infty} b^r P_{ij}^r , \qquad (2.2)$$

величина br <0 повинна вибиратися таким чином, щоб забезпечити збіжність формули для кожної пари. Значимість вузла j обчислюється як сума кількостей шляхів від всіх вузлів:

$$s_j = \sum_i Q_{ij}. \quad (2.3)$$

Для визначення значущості впливу та впливу Web-структур також широко використовуються метрики, що застосовуються для ранжирування знайдених документів в пошукових системах. Так, широке застосування в даній категорії завдання Web Mining знайшла метрика, використовувана пошуковою системою Google – PageRank [10]. PageRank – статична величина, призначена для оцінки якості сторінок на підставі інформації про кількість посилань на неї. Дана метрика не залежить від запитів користувачів і обчислюється заздалегідь для кожної сторінки, що забезпечує швидке її визначення при обробці запиту користувача. За основу PageRank був обраний підхід, який використовується в бібліометріке для оцінки важливості публікації автора по числу її згадувань у бібліографічних посиланнях інших авторів. Для застосування методу до Web в нього були внесені зміни: вага кожного посилання враховується індивідуально і нормується за кількістю посилань на сторінці, що посилається. Ідея методу полягає в наступному: якщо переміщатися по Web-мережі випадковим чином, то при знаходженні на сторінці p користувач може перескочити на іншу сторінку в мережі, обрану псевдовипадковим чином, або він може перейти за посиланням на поточній сторінці, які не повертаючись при цьому і не відвідуючи одну і ту ж сторінку двічі.

Іншою популярною метрикою визначення важливості Web-сторінки є HITS (Hyperlink-Induced Topic Search) [10]. Якщо PageRank обчислюється один раз глобально для всіх сторінок в індексі, то в рамках моделі HITS передбачається, що важливість сторінки залежить від запиту, т. К. В різних тематичних спільнотах різні авторитети. Тому HITS обчислюється локально для кожного запиту. Цей алгоритм

визначає важливість Web-сторінок не тільки по вхідних посиланнях ("авторитетність сторінки"), а й за вихідними посиланнями ("хабам"). Сторінки з даного алгоритму розбиваються як би на спільноти, які ймовірно є тематичними спільнотами. Усередині даних спільнот для кожної сторінки p обчислюється значення "авторитетності" ap, т. Е. "Ваги" кожної сторінки як сума "хаб" -вага посилань, і "хаб" -вага hp кожної сторінки як сума ваг "авторитетності" цитованих сторінок :

$$a_p = \sum_{u \to p} h_u \text{ и } h_p = \sum_{p \to v} a_v . \quad (2.4)$$

Ці значення застосовуються при визначенні релевантності сторінки запиту. Через завісімості побудови графа від запиту користувача метод HITS повільніше, ніж PageRank. Варіант цього методу був використаний Дж. Діном (J. Dean) і М. Хенцігером (M. Henzinger), щоб знайти подібні Webстраниц, грунтуючись тільки на аналізі посилань.

Вони підвищили швидкість методу за рахунок вибірки Web-графа з пов'язаності серверів, який був побудований на підставі попереднього дослідження істотних частин Web. Розширення графа HITS іноді призводить до зашумлення теми. Наприклад, Web-сторінки по темі нагород у кінематографі тісно пов'язані посиланнями (і до деякої міри відноситься до справи) сторінками кінокомпаній. Хоча кінонагороди вужча тема, ніж кінофільми, топові кінокомпанії з'являються як переможці після застосування методу HITS. Це частково пов'язано з тим, що в HITS (так само як і в методі PageRank) у всіх ребер графа однакові значення.

Зашумлення може бути зменшено за рахунок врахування контенту гіперпосилання, який розміщується біля вказівника (анкер-елемента HTML, який зв'язує Web-документи). Тоді можна зробити висновок, що посилання, що містять слова запиту користувача (у нашому прикладі

слово "нагорода"), більш важливі для цього запиту, ніж інші ребра. Системи автоматичної компіляція ресурсів (Automatic Resource Compilation – ARC) і Clever включають подібну запросозавісімую модифікацію ваг ребер. При цьому результати запиту значно поліпшуються [14]. К. Барат (K. Bharat) і М. Хенцігер (M. Henzinger) запропонували інший спосіб об'єднання посилань і текстового змісту, щоб уникнути зашумлення графа [105]. Вони для кожної сторінки будують модель "векторного простору".

На кожному кроці розширення графа, на відміну від методу HITS, вони не включають всі вузли на відстані від попереднього результату запиту. Замість цього вони скорочують розширення графа у вузлах, чиї вектори термів є викидами щодо набору векторів, відповідних документів, безпосередньо відновленим з пошукової машини [15]. У наведеному раніше прикладі можна було б сподіватися, що відповідь на запит "кінонагорода" від початкового пошуку за ключовим словом буде містити більше сторінок, пов'язаних з нагородженнями, а не з компаніями; таким чином, розподіл ключових слів на цих сторінках дозволить даному алгоритму ефективно скоротити вузли, які є викидами і близькі до терму "кінокомпанія".

Висновки до розділу

В результаті аналізу процесу відбору інформації, що реалізується різними алгоритмами та системами, виявлено, що найбільш проблемним є аналіз неструктурованого тексту природною мовою. Такі вхідні дані є надто неоднорідними, тому створити універсальний підхід неможливо.

Визначено, що спільним фактором для всіх алгоритмів, що знижує якість їх роботи є зашумленість даних. Самі алгоритму інформаційного

пошуку та відбору інформації не виконують фільтрацію складових блоків документу, в результаті чого сторінка може бути визнана релевантною помилково.

Для вирішення описаної проблеми можна створити алгоритм попередньої обробки даних, що здійснюватиме фільтрацію вмісту документу, та підготовку його до виконання інформаційного пошуку.

РОЗДІЛ 3. ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ПОШУКУ ЗНАЧИМОЇ ІНФОРМАЦІЇ

З метою підвищення ефективності роботи алгоритмів пошуку розробляються методи автоматичного розділення HTML-сторінок на навігаційну і змістовну частини. Нашою гіпотезою є припущення про те, що виключення (зниження ваги) навігаційної частини сторінок з індексу пошукової машини може бути корисно для вирішення завдань інформаційного пошуку.

3.1. Аналіз структури документа

Завдання аналізу структури HTML-сторінок, що автоматично генеруються є дуже важливим етапом здійснення інформаційного пошуку у web-просторі. Виділення структурної інформації використовується для різних завдань обробки документів, інформаційного пошуку і для відстеження змін і кешуванні web-сторінок. В роботі [7] показано, що існуюча практика оформлення сайтів з використанням засобів динамічної генерації web-сторінок порушує базові припущення, на яких будуються сучасні пошукові системи в Інтернеті. А саме, такими припущеннями є:

1) припущення про релевантні посилання (Relevant Linkage Principle) – гіпертекстові посилання, встановлені авторами сторінки, вказують на авторитетні ресурси по темі документа;

2) ресурси, що посилаються один на одного, тематично близькі (Topical Unity Principle);

3) текст, що знаходиться поряд з гіперпосиланням, релевантний темі ресурсу, на який йде посилання (Lexical Affinity Principle).

На відміну від [7], ми розглядаємо, в першу чергу, тематичний зміст web-сторінок і класичний пошук документів без урахування структури посилань між web-сторінками. У цьому випадку стає можливим проведення оцінки якості інформаційного пошуку з використанням вільно доступних даних (РОМІП). Застосовувані методи аналізу структури web-сторінок можна розділити на:

1) методи, засновані на аналізі dom-дерева окремої HTML-сторінки;

2) методи, засновані на виділенні повторюваних фрагментів сторінок з одного сайту [6, 7, 9, 10, 11]. Стандартним способом формування HTML-сторінок є додавання навігаційної частини по краях сторінки (змістовна частина знаходиться посередині). Сторінка оформляється у вигляді таблиці або вкладених таблиць HTML, при цьому окремі елементи сторінки винесені в різні осередки.

Методи, засновані на аналізі dom-дерева, працюють на основі аналізу структури HTML-сторінки і виділяють змістовну частину сторінки на основі евристичних припущень про "стандартні" способи оформлення документів. Такі методи зазвичай поєднуються з методами, заснованими на виділенні повторюваних фрагментів сторінок одного сайту [9]. Методи, засновані на виділенні повторюваних фрагментів сторінок з одного сайту працюють на основі припущення про те, що у сторінок одного сайту елементи оформлення повторюються, а змістовна частина – різна.

Відзначимо також, що в роботі [7] повторювані фрагменти аналізуються для різних сайтів з метою визначення спільних рис пов'язаних між собою ресурсів.

3.2. Запропонований алгоритм

У даному дослідженні використовується підхід, заснований на виділенні повторюваних фрагментів сторінок одного сайту. Нами запропоновано алгоритм для вирішення даного завдання.

Опис алгоритму

На вхід алгоритму подається директорія з файлами, відповідними сторінкам одного сайту. Алгоритм аналізує дані файли і виділяє в них фрагменти, які вважаються навігаційної частиною. В залежності від налаштувань, алгоритм або видаляє навігаційну частина з файлів, або виділяє навігаційну частина спеціальними тегами.

Алгоритм складається з наступних основних етапів.

1) Розбиття файлів на неподільні при порівнянні послідовності символів – токени.

2) Пошук підмножини файлів (кластер), з однаковим набором ланцюжків токенів (кластери відповідають підмножині сторінок з однаковою навігаційної частиною). Набір ланцюжків токенів складається з одніого або декількох ланцюжків (послідовностей поспіль) токенів. Бажано, щоб набір ланцюжків був якомога довше і зустрічався у великій кількості файлів.

3) Видалення знайденої послідовності ланцюжків токенів з усіх файлів кластеру.

4) Якщо кластер не покриває всю множину файлів, то повторити кроки 2-4 для решти документів.

5) Якщо залишилися необроблені файли (з яких нічого не було видалено), то в цих файлах проводиться пошук і видалення навігаційних частин – ланцюжків токенів – з інших кластерів.

Етап 1. Токенізація HTML-файла

Файл розбивається на неподільні при порівнянні та виділення оформлення послідовності символів – токени.

Визначимо такі токени:

1) окремий тег HTML;

2) текст між тегами.

Пропуски між тегом і текстом ігноруються.

Етап 2. Пошук кластера з однаковою навігаційної частиною

Кожен файл представляється безліччю ланцюжків токенів. Ланцюжок токенів – це послідовність поспіль йдучих токенів довжини min_tokens_chain, де min_tokens_chain – параметр, що задає мінімальну довжину шматка, що вирізається з файлу в токенах (в експериментах використовувалось значення min_tokens_chain = 6). Для кожного ланцюжка токенів довжини min_tokens_chain обчислюється хеш-функція за алгоритмом CRC32. Таким чином, кожен файл відображається як безліч чисел – хеш-значень ланцюжків токенів довжини min_tokens_chain. Для кожного ланцюжка токенів довжини min_tokens_chain запам'ятовується довжина вихідного тексту в байтах.

Побудова кластеру проводиться в кілька кроків:

1) Пошук першої пари документів в кластері.

Перебираються всі пари документів сайту, для кожної пари обчислюється довжина співпадаючих ланцюжків токенов в байтах.

Якщо збіг між файлами занадто великий (більше 70% від загальної довжини файлу), то ці файли вважаються дублями і не підходять для утворення першої пари в кластері.

Пара файлів, для яких збіг є максимальним, додаються в кластер. Перетин цих файлів (набір хеш-кодів ланцюжків токенів) – будемо

називати його *навігаційною частиною кластера* – запам'ятовуємо для подальших обчислень.

Визначимо поріг min_nav_length рівним 80% від довжини навігаційної частини кластера в байтах. Надалі при побудові даного кластеру навігаційна частина буде зменшуватися, але не далі порога min_nav_length.

2) Пошук документів для додавання в кластер.

Проводиться пошук документа, який максимально перетинається з навігаційною частиною кластеру. Файл додається в кластер, якщо довжина збігу з навігаційною частиною кластера не менш min_nav_length. Навігаційна частина кластера стає рівною перетину з доданим кластером. Цей крок повторюється до тих пір, поки є підходящі файли для додавання в кластер.

Етап 3. Видалення навігаційної частини з усіх документів кластеру. Якщо в побудованому кластері виявилося не менше 4 документів, то кластер вважається успішно побудованим, зі всіх документів кластера видаляється "навігаційна частина кластера" (загальна частина всіх файлів кластера).

Етап 4. Пошук інших кластерів

Якщо після побудови кластера залишилося не менше 4 документів, то виконуються кроки 2-4 для решти документів.

Якщо кластер побудувати не вдається, то значення порога min_nav_length знижується до 40% з кроком 20% і виконуються кроки 2-4 з новим значенням порога min_nav_length.

3.3. Оцінка алгоритму

Для оцінки якості роботи алгоритму використовувалися наступні методи. По-перше, проводився експертний аналіз результатів роботи

алгоритму. Експерт аналізував документи, оброблені алгоритмом, з метою визначення, чи дійсно відалений фрагмент можна віднести до оформлення сторінки. По-друге, оцінювався вплив застосування алгоритму на якість результатів пошуку. Для цього використовувалася колекція і методика оцінки Російського семінару з Оцінці Методів Інформаційного Пошуку.

3.3.1. Експертна оцінка результатів роботи алгоритму

Експертна оцінка алгоритму проводилася на двох колекціях документів:

1) колекції сайтів українських університетів;

2) колекції РОМІП, що використовувалась в рамках семінару РОМІП.

Колекція сайтів українських університетів складається з web-сторінок, закачаних з сайтів кількох українських університетів (КПІ, НАУ, КНУ й ін.) Ця колекція включає в себе більше 35000 web-сторінок, закачаних з сайтів різних підрозділів (факультетів, кафедр, лабораторій тощо) даних університетів. Колекція включає велику різноманітність варіантів оформлення сайтів, тому що, незважаючи на спільність організацій, сайти підрозділів створюються різними колективами з великою різноманітністю використовуваних технологій web-програмування.

Колекція РОМІП надана компанією Яндекс для цілей тестування методів інформаційного пошуку в рамках семінару РОМІП. Колекція складається з більше 700000 сторінок з 22000 сайтів домена narod.ru. Експерт візуально аналізував вирізані частини сторінок в стандартному web-браузері, використовуючи підсвічування вирізаної частини страниць. На рис. 4.1 показаний приклад сторінки з виділеним

оформленням. Суцільним сірим кольором показана частина сторінки, віднесена програмою до оформлення.

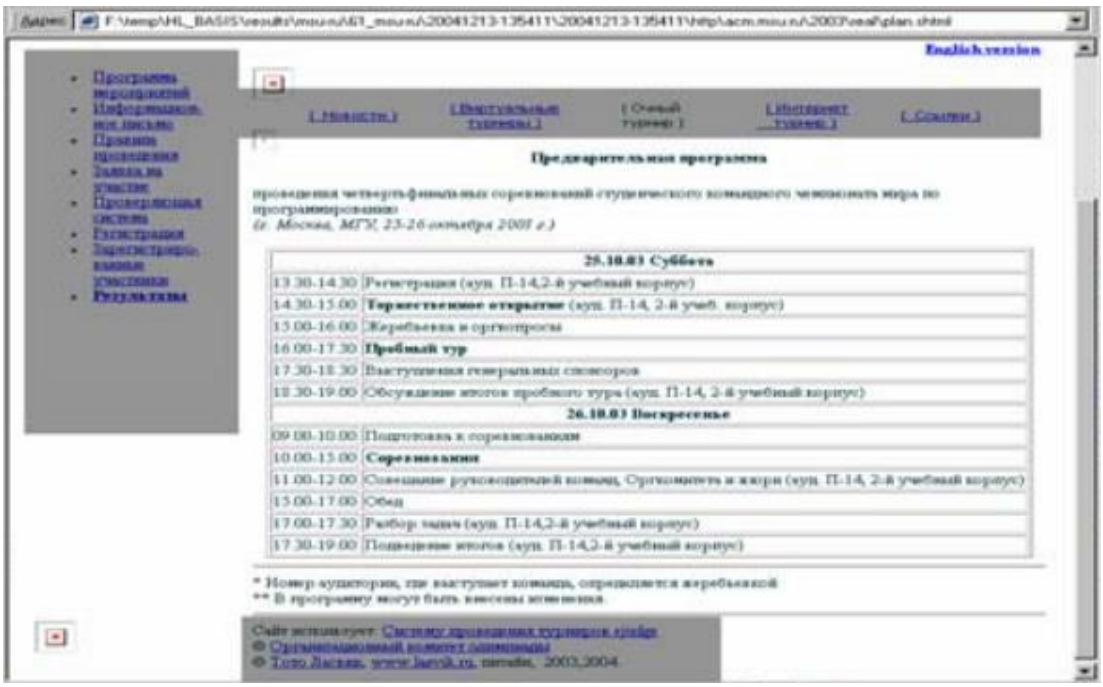


Рис. 3.1. Інтернет-сторінка з виділеним оформленням

Для колекції сайтів українських університетів експертом було проаналізовано 57 випадково вибраних сайтів і результатів обробки програмою виділення навігаційної частини. Для колекції РОМІП – 30 сайтів. Для кожного проаналізованого сайту експерт виставляв оцінку роботи алгоритму від «2» (навігаційна частина не виділена або алгоритм «захопив» змістовну частину) до «5» (навігаційна частина виділена правильно). Оцінки «3» і «4» характеризували проміжні варіанти, коли експерт виявляв незначні помилки.

Таблиця 3.1 Результат експертної оцінки роботи програми

Оцінка	Колекція сайтів українських університетів		Колекція РОМІП	
	Кількість сайтів	Кількість сайтів у %	Кількість сайтів	Кількість сайтів у %
2 3 4 5	4 3 10 40	7% 5% 18% 70%	3 3 9 15	10% 10% 30% 50%
Кількість оцінених сайтів	57		30	
Середній бал	4.5		4	

Як бачимо, робота алгоритму оцінена в середньому на 4 і 4.5, що можна вважати хорошими показними. Отже, запропонований алгоритм успішно виконує завдання виділення значимої частини web-сторінки, та відфільтровування інформаційного шуму.

3.3.2. Оцінка впливу розробленого алгоритму на якість інформаційного пошуку: постановка експерименту

Проведено експеримент з оцінки впливу виділення оформлення сайту на якість інформаційного пошуку. Для даного експерименту використано колекцію документів РОМІП. В рамках семінарів РОМІП учасниками були створені матриці оцінки відповідності знайдених документів запитам (для розглянутої колекції оцінено 66 запитів). Для кожного документа, який був знайдений хоча б однією системою, що

брала участь в РОМІП, асесори РОМІП проставили експертну оцінку відповідності документів запитам.

Дані матриці релевантності використані для оцінки якості роботи пошуку для розробленого методу. Ми використовували матрицю релевантності «or_relevant-minus_table.xml» (тобто документ вважається релевантним, якщо хоча б один з асессорів поставив оцінку «скоріше релевантний» для даного документа). В якості базового методу пошуку документів («відправної точки») використовувався алгоритм пошуку інформаційно-пошукової системи.

Даний алгоритм здійснює пошук документів за словами, з урахуванням морфології російської мови, з ранжуванням документів за класичною формулою TF * IDF та врахуванням взаємної відстані між словами. Детальний опис алгоритму пошуку викладений в [1].

Порівнювалися два методи пошуку документів:

1) пошук документів без усічення навігаційної частини (будемо називати його алгоритм «BasicLine»);

2) пошук документів, оброблених програмою відсікання навігаційної частини (будемо називати його алгоритм «CutNav»).

Програма відсікання обрамлення сайту застосовувалася окремо для кожного сайту з колекції РОМІП – домену 3-го рівня в домені narod.ru. Застосовувалися ті ж налаштування алгоритму, що і при застосуванні до колекції сайтів університетів, описаної в попередньому розділі.

3.3.3. Результати експерименту

В якості основної метрики оцінки якості пошуку використана метрика «середня точність» (AvgPrecision). Дана метрика є комбінованою метрикою оцінки повноти і точності запиту. Наочно

метрику AvgPrecision можна інтерпретувати як площа під графіком повноти у відношенні до точності для даного запиту.

Графік середньої точності для різних запитів показаний на рис.3.2. Запити впорядковані в порядку убування AvgPrecision для алгоритму «BasicLine».

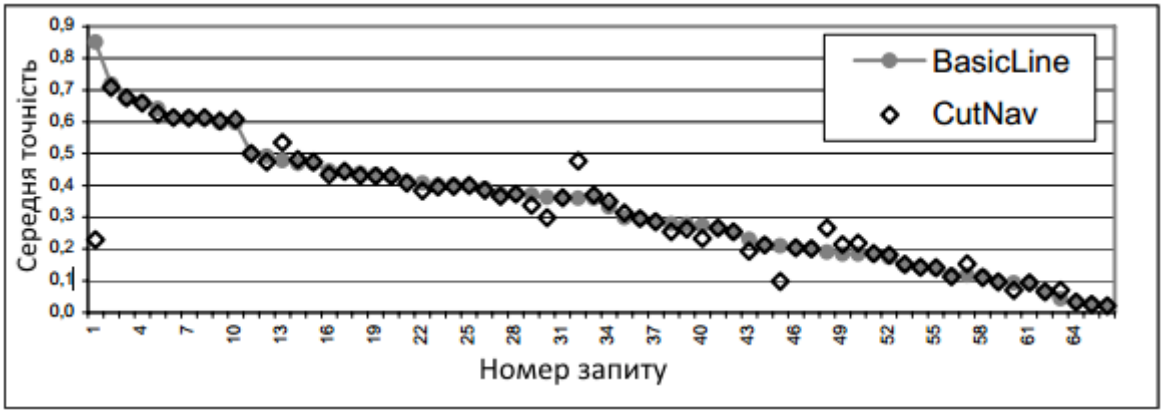


Рис. 3.2. Середня точність для різних запитів для алгоритмів «BasicLine» і «CutNav»

Середнє значення точності по всім запитам становить:

- для алгоритму «BasicLine» – 33,22%;
- для алгоритму «CutNav» – 32,26%.

Тобто після застосування алгоритму видалення обрамлення в середньому якість пошуку знизилася. Подальший аналіз результатів обробки виявив деякі причини погіршення якості пошуку.

3.3.4. Аналіз «провалів» алгоритму

Графік на рис 3.2 показує, що для деяких запитів можна спостерігати підвищення якості пошуку, а для деяких – навпаки, зниження якості пошуку. Проведено аналіз роботи алгоритму для запитів, на яких застосування алгоритму видалення обрамлення призвело до різкого погіршення якості пошуку. Для цього обрано 5 запитів, для яких різниця метрик AvgPrecision для алгоритмів

«BasicLine» і «CutNav» найбільш висока. Вихідними передумовами для такого дослідження були наступні гіпотези про можливі причини погіршення якості пошуку:

1) можливо, поточна реалізація алгоритму не враховує деякі варіанти оформлення web-сторінок, що призводить до вирізання змістовної частини сторінок;

2) можливо, принцип роботи алгоритму – виділення однакових частин сторінок – наводить до вирізання змістовної частини сайту;

3) можливо, обрамлення сайту містить корисну інформацію, що відноситься до запиту користувача;

4) можливо, винесене асессором РОМІП рішення про релевантності документа є спірним.

Проаналізуємо запити в порядку спадання різниці між результатом (середньою точністю) алгоритму «BasicLine» і алгоритмом «CutNav».

Запит №1 «телевізори, домашній кінотеатр» (85,16% BasicLine / CutNav 22,82%)

За даним запитом середня точність алгоритму «BasicLine» становить 85,16%, алгоритму «CutNav» – 22,82%, 74 документа визнано релевантними.

Основна розбіжність результатів пошуку сталося на документах сайту http://tvpanasonic.narod.ru. Алгоритм «BasicLine» знайшов 51 релевантний (з точки зору асессорів РОМІП) документ з цього сайту, а алгоритм «CutNav» ці документи не знайшов.

Аналіз показав, що документи мають такий вигляд, що верхня (основна частина) сторінки містить опис деякої моделі телевізора, а в нижній частині знаходиться довгий (3 екрани) список ключових слів, що «оптимізують сайт», однаковий для всіх сторінок сайту. Цей список

ключових слів містить, зокрема, слова «домашній кінотеатр», «мелодії для телефонів Panasonic», «автомагнітоли Panasonic». Більшість пунктів цього списку не мають ніякого відношення до змістовної частини сторінки.

Алгоритм «CutNav» вирізав «список ключових слів», так як він однаковий для всіх сторінок сайту. Через це документи не були знайдені за словами «домашній кінотеатр».

Перевірка показала, що лише 3 з 51 документа відповідають запитом «телевізори, домашній кінотеатр». Три релевантних документа містять опис широкоформатних плазмових телевізорів і знаходяться за запитом завдяки списку «оптимізаційних слів». Ще 25 документів містять описи великих телевізорів, тому їх можна віднести до домашніх кінотеатрів з великою натяжкою (швидше нерелевантні). Решта 23 документа містять опис звичайних телевізорів невеликого розміру – однозначно нерелевантні.

Однак питання про те, наскільки правильно вирізати такі списки «оптимізаційних слів», є спірним.

Запит №2 «кінотеатр мрія» (20,91% / 9,74%).

За даним запитом середня точність алгоритму «BasicLine» становить 20,91%, алгоритму «CutNav» – 9,74%, 32 документа визнані релевантними. Розбіжність результатів пошуку сталося на документах сайту http://gocinema.narod.ru/. Алгоритм «BasicLine» знайшов 5 релевантних (з точки зору асессоров РОМІП) документів з цього сайту, а алгоритм «CutNav» ці документи не знайшов. Дані документи містять навігаційну частина – лівий тулбар – в якому міститься список кінотеатрів, у тому числі «Мрія» (без посилання на сторінку з описом кінотеатру «Мрія»). В основній частині документа міститься опис

іншого кінотеатру (не "Мрія"). Отже, ці п'ять сторінок не є релевантними до запиту «кінотеатр мрія» (на сторінках немає ніякої інформації про кінотеатр «Мрія» або інші об'єкти, які називалися б «Мрія»). На наш погляд, алгоритм «CutNav» покращив результат, але оцінка асесора РОМІП неправильна.

Запит №3 «що взяти в пологовий будинок» (36,23% / 29,91%)

За даним запитом середня точність алгоритму «BasicLine» становить 36,23%, алгоритму «CutNav» – 29,91%, 17 документів визнані релевантними.

Розбіжність результатів (1 документ) відбувся через те, що алгоритм «BasicLine» некоректно обробив HTML-файл, закодований у вигляді символів юнікоду виду &#1023.

Запит №4 «з чого почати бізнес» (27,28% / 23,27%)

За даним запитом середня точність алгоритму «BasicLine» становить 27,28%, алгоритму «CutNav» – 23,27%, 196 документів визнані релевантними.

Розбіжність результатів сталася на семи документах з чотирьох різних сайтів. Три релевантних документа з сайту http://business.narod.ru були знайдені алгоритмом «BasicLine». Ці документи є дублями, і тому алгоритм «CutNav» видалив весь вміст документів. Відповідно, документи не були знайдені алгоритмом «CutNav». Сторінка містить безліч блоків з посиланнями на інші сторінки. Зокрема в лівому тулбарі міститься посилання «Зароби гроші -> З чого почати». Даний документ знайдений алгоритмом «BasicLine». Алгоритм «CutNav» вирізав лівий тулбар, і даний документ не був знайдений. Ще три документа були знайдені алгоритмом «BasicLine», але не були знайдены алгоритмом «CutNav» через те, що алгоритм

«BasicLine» некоректно обробив HTML-файли, закодовані у вигляді символів юнікоду виду &#1071.

Запит №5 «старовинні свята» (23,04% / 19,31%)

За даним запитом середня точність алгоритму «BasicLine» становить 23,04%, алгоритму «CutNav» – 19,31%, 45 документів визнані релевантними. Розбіжність результатів пошуку сталося на двох документах сайту http://rusvarga.narod.ru. Документи http://rusvarga.narod.ru/ts.htm і http://rusvarga.narod.ru/modernlit.htm знайдені алгоритмом «BasicLine», але не знайдені алгоритмом «CutNav». Алгоритм «CutNav» правильно вирізав обрамлення. Документи знайдені за словами, що містяться в обрамленні. Визнаємо ці документи нерелевантні (вміст документа нічого не говорить про свята). Робимо висновок, що алгоритм «CutNav» покращив результат, а оцінка асесора РОМІП неправильна, отже документи нерелевантні.

Отже, проаналізовано причини зниження оцінок якості роботи алгоритму на п'яти запитах, для яких погіршення найбільш сильно виражено. Для інших сайтів різниця результатів несуттєва (менше 3%), або алгоритм «CutNav» покращив результати пошуку.

Всього проаналізовано 66 сторінок, які були знайдені алгоритмом «BasicLine» і визнані релевантними асессорами РОМІП, але не були знайдені алгоритмом «CutNav». У табл. 2 зведена інформація про причини «провалів» алгоритму для проаналізованих сторінок. Причини провалів алгоритму «CutNav» класифіковані відповідно до гіпотез, які ми сформулювали спочатку експерименту. Деякі сторінки віднесені відразу до декількох категорій, тому сума чисел у другому стовпці більше 100%.

Таблиця 2 – Класифікація причин «провалів» алгоритму

Кількість сторінок	Кількість сторінок у %	Причина
7	11%	Поточна реалізація алгоритму не враховує деякі варіанти оформлення web-сторінок, що призводить до вирізання змістовної частини сторінок
5	8%	Обрамлення не можна виділити на на основі аналізу спільних частин сторінок сайту
4	6%	Обрамлення сайту містить корисну інформацію
55	83%	Винесене РОМІП рішення про релевантність документа є спірним

Варто відзначити, що частка спірних оцінок релевантності така висока (83%) тому, що обрані для аналізу документи є «найбільш спірною частиною» з усіх знайдених документів – ці документи були знайдені за словами, які знаходяться в обрамленні документа.

Проведений аналіз дозволяє зробити наступні висновки:

1) зниження оцінки якості пошуку для алгоритму «CutNav» на один відсоток пов'язано, в основному, з проблемами оцінки релевантності документів;

2) навіть після здійснення ретельної перевірки оцінок релевантності не варто очікувати істотного (більше 5%) підвищення оцінок якості пошуку на даній колекції документів.

Дійсно, оформлення документів найчастіше містить безліч посилань, нерелевантних до теми документа (найчастіше велика частина посилань документа знаходиться в обрамленні).

Висновки до розділу

Представлено алгоритм поділу web-сторінки на навігаційну і змістовну частини. Експертна оцінка результатів роботи алгоритму показала, що алгоритм в цілому виконує поставлене завдання.

Проведено експеримент з оцінки впливу відсікання обрамлення сторінок на якість інформаційного пошуку в мережі Інтернет на колекції РОМІП-WEB-narod.ru. У зв'язку з виявленими проблемами оцінки релевантності сторінок, точно оцінити результати експерименту виявилося неможливо. Можна припустити, що навіть після рішення даних проблем не варто очікувати істотного (більше 5%) підвищення оцінок якості пошуку в умовах відсутності обмежень на тематику запитів.

Проведений експеримент виявив, що є випадки, коли оформлення сайту містить інформацію, важливу для пошуку. Крім того, є випадки, коли оформлення неможливо виявити тільки на основі аналізу співпадаючих частин сторінок.

Застосування алгоритмів, подібних описаному, може бути ефективно для вузьких завдань інформаційного пошуку, коли апріорі відомо, що елементи оформлення групи індексованих ресурсів заважають інформаційного пошуку. Зокрема, такими завданнями можуть бути:

1) періодичний моніторинг фіксованого списку сайтів;

2) пошук інформації на сайтах форумів, блогів, web-конференцій, які мають стандартну структуру.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Загальні положення

В даному розділі описано приміщення, в якому буде використовуватись розроблена обчислювальна система. Приміщення являє собою офіс, що знаходиться в будівлі, несуча конструкція якої виконана з цегли. Параметри приміщення вказано у табл. 4.1:

Таблиця 4.1 – Параметри приміщення

Параметр	Значення	Од. виміру
Довжина (L)	4,6	м.
Ширина (D)	6,8	м.
Висота (H)	3,2	м.
Площа (S)	31,28	кв. м.
Об’єм (V)	100,096	куб. м.
Кількість працівників	3	чол.

Приміщення відповідає вимогам [16], згідно з якими, площа на одне робоче місце повинна бути не менша 6 кв. м., а об’єм не менше ніж 20 куб. м. План приміщення зображено на рис. 4.1.

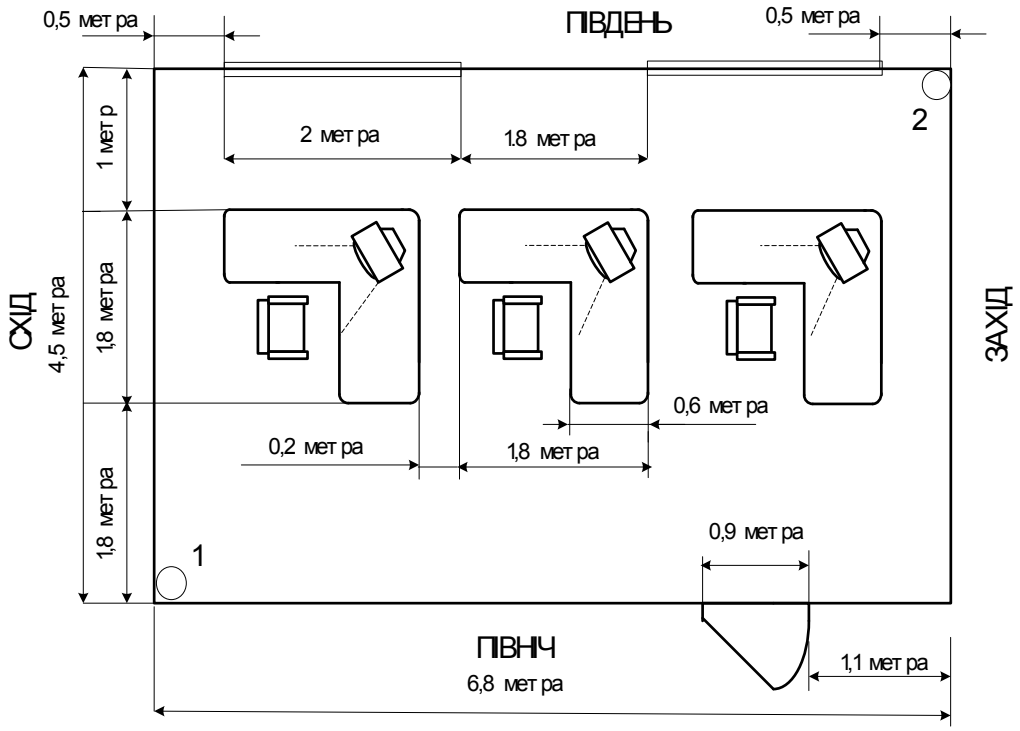


Рис. 4.1. План приміщення

Приміщення обладнано двома вікнами, що спрямовані на південь та мають розміри: 2.7 метра у висоту та 2 метра у ширину.

Столи "Г" – образні, і їх довжина дорівнює 1,8 метра. Вони розташовані так, щоб вікна знаходилися по ліву сторону від адміністратора.

Виходячи з того, що приміщення використовується для роботи адміністратора, і враховуючи можливість працювати віддалено, то характер роботи – непостійний, легкий (не використовуються ніякі фізичні навантаження). Отже, категорія робіт – легка, Іб.

4.2. Аналіз шкідливих та небезпечних факторів

Для аналізу шкідливих та небезпечних факторів виділимо п’ять основних факторів, які можуть впливати на працівників: виробничий шум, мікроклімат приміщення, випромінювання ВДТ, природне та штучне освітлення та електробезпека.

4.2.1 Аналіз виробничого шуму

Джерелами шуму в приміщенні є вентилятори охолодження ЕОМ (максимальний рівень шуму – 35 дБА) та «спліт»-кондиціонер (максимальний рівень шуму – 45 дБА). Звук можна вважати постійним так, як його рівень протягом робочого дня змінюється не більше ніж на 5 дБА.

Рівень шуму від декількох джерел розраховується за формулою:

$$L_{\text{заг}} = 10 \cdot \lg\left(\sum_{i=1}^{n}\left(t_{\text{р}} \cdot N \cdot 10^{0.1 \cdot L_i}\right)/T\right), \quad (4.1)$$

Де n – загальна кількість типів приладів, $t_р$ – час роботі приладу, N – кількість приладів, L_i – середній рівень шуму приладу, T – робочій день.

Фактичний рівень шуму у приміщенні протягом робочого дня становить:

$$L = 10 \cdot lg((8 \cdot 3 \cdot 10^{0,1\ 35} + 8 \cdot 1 \cdot 10^{0,1\ 45}) / 8) = 46,1 \text{ (дБа)} \quad (4.2)$$

Допустимий еквівалентний рівень звуку згідно [17] наступний: для програміста ЕОМ нормований звуковий тиск повинен бути не вище 50дБА. У даному випадку сумарний рівень звукового тиску не перевищує нормованого значення.

4.2.2. Аналіз мікроклімату приміщення

Допустимі значення мікроклімату приміщення наведені в табл. 4.2, потрібно перевірити відповідність необхідних значень параметрів з їх фактичними значеннями.

Таблиця 4.2 – Норми мікроклімату для приміщень ЕОМ.

Пора року	Температура повітря (градуси С)	Відносна вологість	Швидкість руху повітря
Холодна	21-23	40-60	0,1
Тепла	22-24	40-60	0,2

Для створення й автоматичної підтримки в приміщенні незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, у холодну пору року використовується водяне опалення, у теплу пору року застосовується кондиціонування повітря.

Відносна вологість повітря у приміщенні складає 45%, швидкість руху повітря менша 0,1 м/с. Отже, дане приміщення відповідає нормам.

4.2.3. Аналіз випромінювання

ПК є значним джерелом електромагнітного випромінювання. Згідно з [16], припустима інтенсивність потоку енергії 10 Вт/м^2, а напруженість електричного поля в електричній складовій на відстані 0,5м від екрану – 10 В/м.

Сучасні дисплеї мають інтенсивність потоку не більше 1 Вт/м^2, що відповідає встановленим нормам.

4.2.4. Аналіз природнього та штучного освітлення

Для запобігання прямого відблиску світла дисплей розміщується під кутом 75° вікна.

У приміщенні використовується бокова система природного освітлення та загальна система штучного освітлення. Тому нормоване значення освітлення повинно бути 300 лк, а КПО 1.0.

Фактична площа вікон 10,8 м^2, в приміщенні застосовується штучне освітлення.

У приміщенні встановлені світильники типу Л201-03 на дві лампи ЛБ-40 кожен із загальним світловим потоком 2200 лм (2х1100). Світильники встановлені в два ряди. Формула для обчислення кількості світильників:

$$N = \frac{\text{Е}\cdot Кз\cdot S}{Fл\cdot n\cdot \eta\cdot z}, \quad (4.3)$$

де $E=E_н$=300 лк;

Кз – коефіцієнт запасу = 1,5;

S – площа, що освітлюється = 31,28;

Z – коефіцієнт нерівномірності освітлення = 1,1;

$F_л$ – світловий потік, створюваний однією лампою, лм. Fл = 1100 лм;

n – число ламп в світильнику = *2;*

η – коефіцієнт використання світлового потоку(для даного приміщення він становить 0,6);

Отже,

N = (300·1,5·31,28) / (1100·2·0,6·1.1) = 9,69.

$E_н$ = 300 лк

$E_ф$ = (N·$F_л$·n· η ·z)/(Кз·S)=(10·1100·2·0,6·1.1)/(1.5·31.28)=309,46

$E_ф$ > $Е_н$, тобто необхідна кількість світильників 10.

Розміщення 10 світильників показано на рис. 4.2

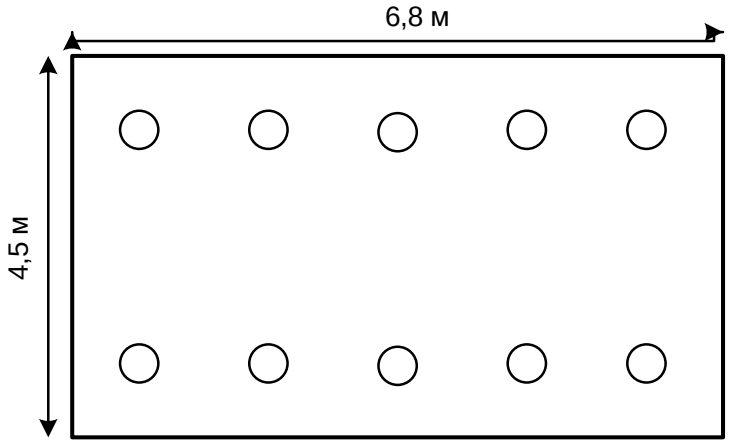


Рис. 4.2. Розташування освітлювальних установок

4.2.5. Аналіз електробезпеки

Згідно з [19] за ступенем небезпеки враження електрострумом приміщення відноситься до приміщення без підвищеної небезпеки. У приміщення проведено 3-х фазне електроживлення напругою 220 В, частотою 50 Гц та з максимальним струмом у 32 А.

Данне приміщення має відносну вологість повітря 40-60%, температура повітря не перевищує 24°, регулярно проводиться вологе прибирання, покриття підлоги – ламінат. Тому дане приміщення відноситься до групи без підвищеної електричної небезпеки.

Електробезпека у лабораторії забезпечується технічними і організаційними заходами.

Організаційними заходами є інструктаж і перевірка знань правил безпеки і інструкцій.

Технічними заходами є те, що всі елементи електроприладів виконані згідно умов техніки безпеки і мають ізоляційне покриття.

4.3 Пожежна безпека

4.3.1. Категорія приміщення за вибухопожежною безпекою

Оскільки в даному приміщенні знаходяться легкозаймисті матеріали, що містяться в апаратурі ПК, то воно відноситься до категорії В, відповідно до [21].

4.3.2. Причини та наслідки виникнення пожежі

Пожежа в приміщенні може призвести до жахливих наслідків (травмування або загибель людей, втрата цінної інформації, псування майна, тощо), тому необхідно:

- виявити та усунути всі можливі причини виникнення пожежі;
- розробити план заходів для ліквідації пожежі в приміщенні;
- розробити план евакуації людей з приміщення;
- розробити інструкцію про заходи пожежної безпеки у приміщенні;
- забезпечити приміщення необхідними засобами пожежогасіння і пожежною сигналізацією та зв’язком;

Причинами виникнення пожежі можуть бути:

- використання пошкоджених електроприладів;
- коротке замикання у електропроводці або розетці;
- виникнення пожежі внаслідок влучення блискавки в будинок;
- загоряння будинку внаслідок зовнішніх впливів;
- необережне поводження з вогнем і недотримання заходів пожежної безпеки.
- використання електронагрівальних приладів з відкритими нагрівальними елементами.

4.3.3. Засоби пожежогасіння

Для гасіння пожежі в пергу чергу використовуються вогнегасники. Для приміщення категорії В використовуються вуглекислотно-брометилові вогнегасники (ОУ-3, ОУ-5, ОУ-7 та ін.), що призначені для гасіння невеликих вогнищ горіння волокнистих і інших твердих матеріалів, а також електроустановок. За вимогами норм належності вогнегасників приміщення категорії В площею від 25 до 50 м^2 включно необхідно обладнати вогнегасниками з масою вогнегасної речовини як мінімум 8 кг. Приміщення достатньо обладнати двома вогнегасниками: ОУ-5 (містить 3,5 кг вогнегасної речовини) та ОУ-7 (5 кг вогнегасної речовини), що зображені на рис 4.1 з індексами 1 і 2 відповідно.

4.3.4. Пожежна сигналізація

В даному приміщенні встановлена система пожежної сигналізації System Sensor. Обладнання сигналізації складається з трьох основних частин наведених нижче:

- прилад приймально-контрольний пожежний АРТОН-04П. (призначений для організації централізованої і автономної охорони різних об'єктів від пожеж, шляхом цілодобового контролю стану до 4-х шлейфів пожежної сигналізації);
- телефонний комунікатор БСКТ-1 (є блоком розширення приладів приймально-контрольних пожежних «Артон-04П», «Артон-08П» і призначений для автоматичного дозвону по 8 запрограмованим номерами і передачі кодованих повідомлень про зміну стану ППКП на ПЦН або у вигляді голосових повідомлень, а також для віддаленого моніторингу та управління ППКП);

- двох сповіщувачів пожежний димових оптичних точкових СПД-3 (призначені для виявлення спалахів в закритих приміщеннях різних будівель і споруд, що супроводжуються появою диму і передачі сигналу «ПОЖЕЖА» на ППК).

Вибір пожежних сповіщувачів здійснюється в залежності від характерних приміщень, виробництв, технологічних процесів відповідно Додатка К до [22] "Пожежна автоматика будинків і споруд". Згідно з цим, приміщення обладнане адресованими оповіщувачами.

4.4. План ліквідації надзвичайної ситуації

У ході роботи потрібно створити план локалізації та ліквідації аварійних ситуацій, які можуть виникнути під час роботи на підприємстві.

4.4.1. Можливі аварійні ситуації

В нашому випадку існує можливість радіаційного забруднення під час аварійної ситуації комунального рівня на атомній електростанції.

В Україні діють 5 атомних електростанцій з 16 енергетичними ядерними реакторами, 2 дослідних ядерних реактори та більше 8 тис. підприємств і організацій, які використовують у виробництві, науково-дослідній роботі та медичній практиці різноманітні радіоактивні речовини, а також зберігають і переробляють радіоактивні відходи. Найближчою АЕС до міста Київ є Чорнобильська АЕС з реакторами типу РБМК. На даний момент вона повністю виведена з експлуатації, проте на території Росії до цього часу використовується 11 реакторів цього типу.

Приміщення, в якому проводилась розробка, знаходиться в зоні можливої дії радіоактивного забруднення в разі аварії на АЕС, що

знаходиться поряд. Необхідно оцінити радіаційну обстановку, що може скластися в районі об’єкту та вибрати найбільш ефективний спосіб захисту робітників і службовців об’єкту.

4.4.2. Оцінка радіаційної обстановки в зоні радіаційного забруднення

Потрібно розрахувати радіаційну обстановку в зоні радіаційного забруднення в результаті аварії на реакторі типу РБМК. Оцінку будемо проводити згідно з інструкціями в [23].

Аварія сталась о 3:00. Виміряний рівень радіації на 3-30 становить 50 радіан/годин. Астрономічний час початку робіт щодо ліквідації наслідків аварії – 5:30. Тривалість робіт оцінюється 4. години. Допустима доза радіації становить 20 радіан. Коефіцієнт ослаблення радіації – 3. Всі розрахунки для оцінки радіаційного стану прийнято проводити у відносному часі. Точкою відліку вважається момент аварії. Тобто, $t_{\text{а}_{\text{астрон}}} = 3{:}00$, $t_{\text{а}_{\text{відн}}} = 0.3$ огляду на це, відносний час початку роботи є наступним: $t_{\text{п}_{\text{астрон}}} = 5{:}30$, $t_{\text{п}_{\text{відн}}} = 2{:}30$. Слід також розрахувати час кінця роботи. $t_{\text{к}_{\text{відн}}} = t_{\text{п}_{\text{відн}}} + T = 6{:}30.$

Для наступних розрахунків слід вирахувати рівень радіації на першу годину після аварії. Для цього маючи коефіцієнт для перерахунку рівнів радіації на рівний час після аварії на АЕС $K_{0.5_{\text{п}}} = 0{,}81$ та виміряний на цей час рівень радіації – 50 Р/год по формулі знайдемо шукане значення.

$$\text{P}_1 = \text{P}_{\text{вим}} \cdot K_{t_{\text{вим}}} \qquad (4.4)$$

За формулою 4.4 маємо, що рівень радіації на першу годину після аварії – 40,5 Р/год.

Далі розрахуємо дозу отриманої радіації під час роботи. Розрахунки будемо проводити за формулою 4.5

$$Д = \frac{Р_{ср} \cdot t_р}{K_{осл}} \quad (4.5)$$

Для використання формули 4.4 слід розрахувати значення середнього випромінювання, яке вимірюється як середнє арифметичне рівня радіації на початок робіт та на кінець робіт. Рівень радіації на певний момент розраховується за формулою 4.6.

$$P_t = \frac{Р_1}{K_{t_н}} \quad (4.6)$$

На початок ліквідаційних робіт рівень радіації – 31,15, на кінець ліквідаційних робіт – 23,14. Середній рівень радіації – 27,145. Провівши необхідні обчислення за формулою 4.5 отримаємо, що доза отриманого при роботі випромінювання рівна 36,193.

Далі слід розрахувати допустимий час роботи на зараженому об'єкті. Для цього слід розрахувати коефіцієнт α. Для цього скористаємось формулою 4.7.

$$\alpha = \frac{Р_1}{K_{осл} \cdot Д_{доп}} \quad (4.7)$$

Вирахувана величина α рівна 0,675. Маючи її значення та час початку роботи з ліквідації $t_{п_{відн}} = 2{:}30$ скориставшись графіком на рисунку 4.5 виміряємо допустимий час роботи на реакторі.

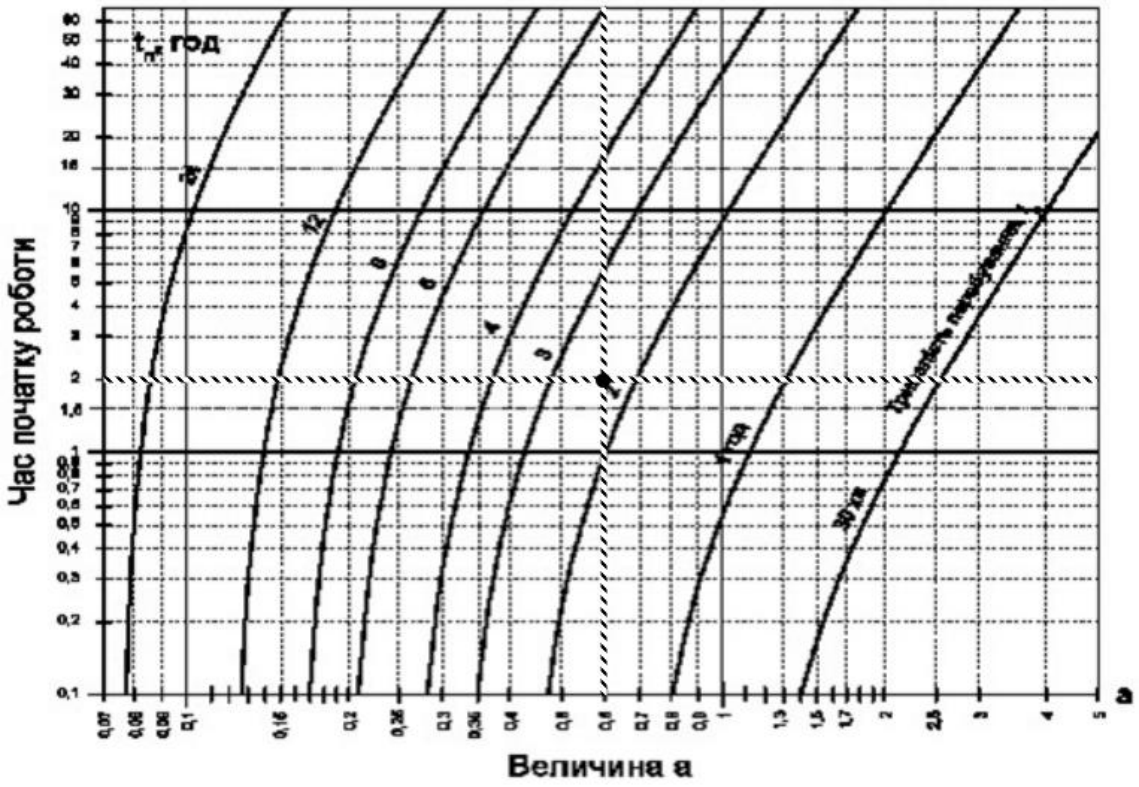


Рис. 4.5. Графік визначення тривалості перебування в зоні радіоактивного зараження при аварії на АЕС з реактором РБМК-1000

За графіком бачимо, що в даних умовах допустимий час роботи – 3 год.

Отже, слід обов’язково проводити інструктаж з техніки безпеки та завжди мати напоготові службу екстреного реагування на випадок аварійної ситуації. Також забезпечити весь особовий склад необхідною захисною екіпіровкою.

4.5. Техніка безпеки

Напруга живлення офісної техніки (220 В) є небезпечною для життя людини. Тому, незважаючи на те що в конструкції комп`ютера передбачена достатня ізоляція від струмопровідних ділянок, необхідно знати та чітко виконувати ряд правил техніки безпеки.

А) Перед початком роботи слід переконатися у справності:

- електропроводки,
- вимикачів,
- штепсельних розеток, за допомогою яких обладнання включається в мережу,
- наявності заземлення комп'ютера, його працездатності.

Б) Щоб уникнути пошкодження ізоляції проводів і виникнення коротких замикань забороняється:

- вішати що-небудь на дроти,
- зафарбовувати і білити шнури і дроти,
- закладати дроти і шнури за газові і водопровідні труби, за батареї опалювальної системи,
- висмикувати штепсельну вилку з розетки за шнур, зусилля повинен бути додані до корпусу вилки.

Для виключення ураження електричним струмом забороняється:

- часто вмикати і вимикати комп'ютер без необхідності,
- торкатися до екрану і до тильної сторони блоків комп'ютера,
- працювати на засобах обчислювальної техніки і периферійному устаткуванні мокрими руками,
- працювати на засобах обчислювальної техніки і периферійному устаткуванні, що мають порушення цілісності корпусу, порушення ізоляції проводів, несправну індикацію включення живлення, з ознаками електричної напруги на корпусі,
- класти на засоби обчислювальної техніки і периферійне устаткування сторонні предмети.

Забороняється:

- під напругою очищати від пилу і забруднення електрообладнання;
- перевіряти працездатність електрообладнання в непристосованих для експлуатації приміщеннях із струмопровідними підлогами, сирих, що не дозволяють заземлити доступні металеві частини;
- проводити ремонт засобів обчислювальної техніки і периферійного обладнання під напругою. Ремонт електроапаратури проводиться тільки фахівцями-техніками з дотриманням необхідних технічних вимог.

Щоб уникнути ураження електричним струмом, при користуванні електроприладами не можна торкатися одночасно яких-небудь трубопроводів, батарей опалення, металевих конструкцій, з'єднаних із землею.

При користуванні електроенергії в сирих приміщеннях дотримуватися особливої обережності.

В) Після закінчення роботи необхідно знеструмити всі засоби обчислювальної техніки і периферійне устаткування. У разі безперервного виробничого процесу необхідно залишити включеними тільки необхідне обладнання.

Висновки до розділу

Проведено аналіз основних аспектів охорони праці. Детальний аналіз показав, що всі показники відповідають встановленим нормам згідно з чинним законодавством.

Параметри приміщення відповідають нормам. Рівень шуму не перевищує нормативних обмежень. Встановлено які пожежні сигналізації і які системи пожежогасіння встановлені у робочій кімнаті.

Проведено оцінку радіаційну обстановку в разі аварійної ситуації на атомній електростанції. В цілому місце розробки проекту відповідає нормам.

Проаналізовані фактори відповідають чинним нормативним вимогам.

ВИСНОВКИ

1. Розглянуто та проаналізовано принципи роботи основних алгоритмів інтелектуального аналізу вмісту web-ресурсів для пошуку значимої інформації.

2. На основі проведеного аналізу об’єкту дослідження запропоновано алгоритм попередньої обробки web-документів для поділу web-сторінки на навігаційну і змістовну частини. Експертна оцінка результатів роботи алгоритму показала, що алгоритм в цілому виконує поставлене завдання.

3. Реалізовано програмний продукт на основі запропонованого алгоритму, що оброблює набори сайтів, збережених з мережі Інтернет, та використовувався для проведення експериментальних досліджень.

4. Проведено експеримент з оцінки впливу видалення обрамлення сторінок на якість інформаційного пошуку. У зв'язку з виявленими проблемами оцінки релевантності сторінок у контрольному наборі, точно оцінити результати експерименту виявилося неможливо.

5. Аналіз результатів експерименту виявив, що є випадки, коли оформлення сайту містить інформацію, важливу для пошуку. Крім того, інколи оформлення неможливо виявити тільки на основі аналізу співпадаючих частин сторінок.

6. Застосування алгоритмів, подібних запропонованому, може бути ефективним для вузьких завдань інформаційного пошуку, коли апріорі відомо, що елементи оформлення групи індексованих ресурсів заважають виконанню інформаційного пошуку.

7. Запропонований алгоритм має практичну цінність для окремих задач пошуку значимої інформації та може застосовуватись для підвищення ефективності роботи пошукових алгоритмів.

8. Проведено аналіз офісного приміщення в контексті основних аспектів охорони праці. Детальний аналіз показав, що всі показники відповідають встановленим нормам згідно з чинним законодавством.

ПЕРЕЛІК ПОСИЛАНЬ

1. Niki R. Kapadia Web content mining techniques – a comprehensive survey, 2012 – №2 [Електронний ресурс]. – Режим доступу: http://www.ijert.org/download/1950/web-content-mining-techniques-a-comprehensive-survey

2. Алексеєв С.С., Морозов В.В.. Методы машинного обучения в задачах извлечения информации из текстов по эталону, 2013, – Режим доступу: http://rcdl.ru/doc/2009/237_246_Section07-1.pdf

3. Кутукова Е.С. Технология Text mining, 2011, [Електронний ресурс]. – Режим доступу: http://www.sworld.com.ua/simpoz3/3.pdf

4. Ланде Д.В. Поиск знаний в Internet. Профессиональная работа. – СПб.: Діалектика, 2005. – 272 с.

5. Рассел Кей. Технология RSS – инструменты и методы // Computerworld. – 2004. – № 32. – С. 22-25.

6. Семененко А.В. Варианты корпоративных технологий добычи текстовых данных – 2002. – № 5. – С. 43-47.

7. C-News – видання про високі технології // http://www.cnews.ru.

8. Манченко Є.В., Кишинський И.Ю. GALAKTIKA-ZOOM. Тезисы доклада // Науково-практична конференція «Проблеми обробки значних масивів неструктурованих текстових документів». – Москва, Фонд эфективной політики, 21-22 травня 2001 г.

9. Колесов А.Н. Извлекая знания из хаоса информации // PC Week. – 2003. – № 43. – С. 18-20.

10. Вороной С.М., Москальов В.Э. Интегрированные инструментальные средства приобретения, конструирования и обновления знаний интеллектуальных обучающих систем // Проблеми програмування. – 2000. – № 1-2. – С. 484-487.

11. Агеєв М.С. Вершинников І. В. Извлечение значимой информации из web-страниц для задач информационного поиска – 2005, [Електронний ресурс]. – Режим доступу: http://elar.urfu.ru/bitstream/10995/1414/1/IMAT 2005 15.pdf

12. Єршов А. «Мы фанаты машинного обучения» – [Електронний ресурс]. – Режим доступу: http://lenta.ru/articles/2013/06/17/.

13. Chow T., Lin Y., Chan W. The Development of a Web-based Demographic Data Extraction Tool for Population Monitoring // Transactions in GIS, 2011. Vol. 15. P. 479—494.

14. IBM Watson: a sophisticated data analytics & insight engine [Электронный ресурс]. – Режим доступу: http://www-01.ibm.com/software/ebusiness/jstart/watson/.

15. Li Z., Ng W., Sun A. Web data extraction based on structural similarity// Knowledge and Information Systems, 2005. с. 438—461.

16. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин : затверджені наказом Держкомітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 № 65 [Електронний ресурс]. – Режим доступу: http://dnop.com.ua.

17. ДНС 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку МОЗ України [Електронний ресурс]. – Режим доступу : http://dnop.com.ua/dnaop/act4878.htm.

18. ДБН.В2.5.-28-2006. Штучне та природне освітлення [Електронний ресурс]. – Режим доступу: http://dnop.com.ua/dnaop/5.htm.

19. Правила улаштування електроустановок вид. №3, перероб. і доп. – 736 с.

20. ДНАОП 0.00-1 31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин – Київ, 1999 р. – 30 с.

21. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою : Затвердж. наказом МНС від 03.12.2007 № 833.

22. ДБН В.2.5-56-2010 «Інженерне обладнання будинків і споруд. Системи протипожежного захисту» [Електронний ресурс]. – Режим доступу : www.mns.gov.ua/files/2012/2/9/87__56_.doc.

23. Стеблюк М.І. Цивільна оборона та цивільний захист: Підручник. – 2-ге вид., переробл. – К.: Знання, 2009 р. – 487 с.