<html>
<p align=justify>
{Вставити текст лекції}
</P>
Розпочати тестування
</html>

2. Кожне тестове питання потрібно сформувати в окремий html-файл за таким шаблоном: створити текстовий файл, зберегти його як файл-html, вставити у файл код програми:

<html>
<body>
{Вставити текст тестового питання}
</html>
</body>

Цей файл повинен міститись на диску в папці програми Teacher/Client/Courses/{назва папки створеного курсу}/Tests.

Висновок

Розроблена система є універсальною, оскільки дозволяє створювати навчальні курси з будь-якої предметної сфери. Для кожного навчального курсу передбачено створення контрольних робіт, які містять запитання різної складності. Для кожного питання є можливість встановити потрібну кількість балів, залежно від складності питання. Відповідно до набраної кількості балів система дає/не дає можливість користувачеві перейти на іншу лекцію для вивчення.

УДК 004.415

П. Жежнич, А. Пелещишин, Ю. Сєров, Д. Тарасов
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

АРХІТЕКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТА РАНГУВАННЯ НОВИННИХ ІНТЕРНЕТ-РЕСУРСІВ

Розглянуто архітектуру та принципи побудови автоматизованої системи класифікації та рангування новинних Інтернет-ресурсів. Призначення системи — автоматизована агрегація новин, поданих у електронній формі, їх класифікація, фільтрація, рангування, впорядкування, зручне подання.

Використання системи значно спрощує роботу користувача новинних каналів у форматі RSS, позбавляє користувача необхідності опрацьовувати нетематичні та малоінформаційні новини.

The paper considers actual problem of intelligent system of information gathering and analysis from news Internet-resources development. The system work is based on news gathering from RSS-channels of news Web-sites. System is intended for new messages fast download, these messages classifying, ranking, ordering and presentation of retrieved messages in proper view.

System consists of 4 components: Crawler, Grabber Analyzer, Representer. The proposed system allows simpler and faster news processing received from Internet-resources.

Постановка проблеми у загальному вигляді

Для забезпечення користувачів Інтернет актуальною інформацією про новини, події, нові технології існує величезна кількість Інтернет-сайтів, котрі надають інформацію про новини в

різноманітних сферах життя. Окремі сайти мають суто новинний характер як за змістом, так і за технологіями подання інформації (блоги, blogs).

Кількість та різноманітність цих ресурсів настільки значна, що переважно людина не може повністю сприйняти та всебічно опрацювати таку кількість нової інформації. Наприклад, кількість новинних ресурсів навіть за вузькоспеціалізованими темами вимірюють сотнями або тисячами. Опрацювати таку кількість швидко старіючої інформації, відфільтрувати такі повідомлення людині складно. Тому виникає потреба у створенні автоматизованого засобу збирання, фільтрації, класифікації, аналізу та відображення у зручній формі новин ресурсів з Інтернет.

Автоматизовану систему системи класифікації та рангування новинних Інтернет-ресурсів призначено для розв'язання описаних проблем. Ця система повинна забезпечити користувачу засоби швидкого стягування нових повідомлень, їх класифікацію, рангування, впорядкування, а також зручну форму подання. Реалізація комплексу таких вимог значно спростить, прискорить та зробить ефективнішою роботу користувача, позбавить його необхідності опрацьовувати зайву інформацію.

Аналіз останніх досліджень

Результати аналізу Інтернет-ресурсів новин свідчать, що більшість з них подає новини у форматах, основаних на стандарті XML. Переважно для цього використовують мови RSS та RDF або Atom. Відповідно до цих мов новини подають як потік (канал) структурованої інформації у вигляді повідомлень, де виділено такі елементи:

• Для каналу новин – {title, link, description, language, copyright, managingEditor, webMaster, lastBuildDate, rating, docs, cloud, ttl, image, rating, textInput, skipHours, skipDays}.

• Для повідомлення – {title, link, description, author, category, comments, enclosure, guid, pubDate, source}.

Таке подання інформації є дуже зручним, оскільки позбавляє користувача необхідності відвідувати усі необхідні йому сайти. Користувачу достатньо встановити на своєму комп'ютері програму зчитування RSS-каналів (RSS-reader або RSS-агрегатор) та налаштувати цю програму так, щоб вона з певною періодичністю зчитувала новини з заданих каналів. Хоча персональні RSS-агрегатори виконують таке завдання, як групування отриманих повідомлень, але не виконують таких важливих завдань, як узагальнення, оброблення та аналіз даних.

Цікаві важливі повідомлення неодноразово дублюються на різних сайтах, кількість яких зростає експоненційно, тоді як кількість джерел, вартих уваги, зростає лінійно. Виявити очевидні дублі не так вже й складно, але задача виявлення повідомлень, які однакові за змістом, є значно складнішою. Такі задачі вирішують за допомогою алгоритмів порівняння контенту та ймовірнісних оцінок. Спрощення цієї задачі можливе завдяки застосуванню методів "глибинного аналізу текстів" (Text mining).

Крім того, програми зчитування RSS-каналів не вирішують задач фільтрації повідомлень за деякими ознаками (ключові слова, рейтинг ресурсу тощо), рангування повідомлень, фільтрації спаму та однакових повідомлень з різних джерел.

Формування цілей

Існуючі програми зчитування RSS-каналів не вирішують багато важливих задач з оброблення отриманих повідомлень, тому створення системи, яка б допомагала користувачу легко, швидко й ефективно опрацьовувати інформацію, є нагальною проблемою.

Призначення інтелектуальної системи збирання та аналізу інформації з Інтернет-ресурсів новин полягає у стягуванні, фільтруванні, рангуванні, класифікації та зручному поданні інформації, отриманої з RSS-каналів новин.

Можна виділити такі етапи функціонування системи:

1. Стягування RSS-файлів (feeds, фідів) за вказаними адресами, оброблення отриманих файлів, виокремлення повідомлень (парсання) та занесення отриманих повідомлень до бази даних системи, а також оновлення інформації про канал;

2. Оброблення нових повідомлень: відкидання порожніх, некоректних, а також уже наявних в базі даних повідомлень, рангування та класифікація повідомлень за певними класифікаційними ознаками (наявність в тексті заданих ключових слів та словосполучень, частота появи цих слів та ін.);

3. Аналіз повідомлень і підготовка для зручного подання користувачеві (аналіз актуальності, тематики і ін.);

4. Зручне подання нових повідомлень з різноманітними можливостями впорядкування та класифікації.

Відповідно до описаних етапів інтелектуальна система збирання та аналізу інформації з Інтернет-ресурсів складатиметься з таких компонент:

1. Компонента збирання даних з каналів новин (Crawler);
2. Компонента класифікації (Grabber);
3. Компонента аналізу (Analyzer);
4. Компонента відображення (Representer).

Кожна з компонент вирішуватиме такі задачі:

• Crawler – сканування каналів новин і стягування нових повідомлень.

• Grabber – фільтрування некоректних повідомлень та повторів, рангування і класифікації нових повідомлень за вказаними ознаками.

• Analyzer – аналізування повідомлень на предмет їх важливості (визначення відповідності повідомлень до заданої тематики) та актуальності в часі.

• Representer – відображення опрацьованих актуальних повідомлень у зручному Web-інтерфейсі з можливостями їх впорядкування за різними ознаками (за часом надходження, за розділами тематики, за важливістю тощо).

Основний матеріал

Автоматизована система збирання та аналізу інформації з Інтернет-ресурсів новин реалізована за допомогою стандартних технологій, які традиційно використовують у Web-середовищі (MySQL, Apache, Perl, PHP).

Програмна частина реалізована за допомогою мов Web-програмування Perl та PHP. База даних реалізована за допомогою СУБД MySQL.

Розглянемо схему бази даних та особливості функціонування кожної з підсистем.

Схема бази даних системи

Зобразимо схему бази даних за допомогою діаграми "сутність–зв'язок" (ERD).

Розглянемо таблиці бази даних. Таблиці ob_channel (об'єкт) та oh_channel (історія) описують RSS-канал. Вони містять поля, котрі відповідають основним елементам структури RSS-каналу (рис. 1).

Таблиці dc_message, tdc_message та dt_message призначені для зберігання в них повідомлень, видобутих з RSS-каналів. У таблиці dc_message зберігатимуться повідомлення. Структура таблиці dc_message містить основні структурні елементи повідомлення в RSS-каналі. У таблиці dt_message зберігається додаткова інформація про повідомлення (рейтинг, категорія повідомлення). Таблиця tdc_message є тимчасовою і має структуру, ідентичну до dc_message. Вона призначена для тимчасового зберігання повідомлень після скачування, перед тим як вони будуть класифіковані і переміщені в основну таблицю dc_message.

Таблиці rf_theme та ob_category описують теми та категорії, а таблиця od_category призначена для зберігання правил, котрі визначатимуть належність/неналежність повідомлення до тієї чи іншої категорії. Правила належності/неналежності повідомлення до категорії задають за допомогою шаблонів ключових слів або фраз.

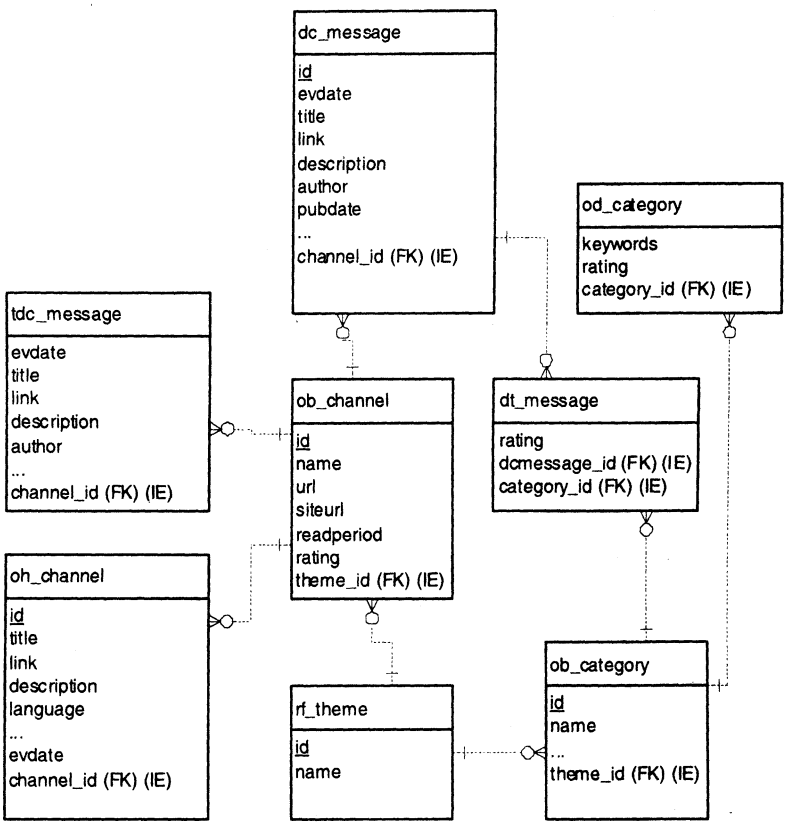

Рис. 1. Схема бази даних, зображена за допомогою діаграми "сутність–зв'язок"

Розглянемо приклад. У межах теми "Інформаційні технології" виберемо категорію "Програмне забезпечення". Ключовими словами для цієї категорії можуть бути:

Ключові слова, які визначають рейтинг повідомлення в темі

Слово (keyword)	Рейтинг
%Software%	5
%software%	4
%SOFTWARE%	10
%warez%	-100
%Warez%	-100
%WAREZ%	-100

Серед ключових слів перші три слова відповідають за приналежність до категорії "Програмне забезпечення", а останні три – за неприналежність. Це пов'язано з багатозначністю слів, яка притаманна природним мовам. У цьому прикладі передбачено, що наявність у повідомленні слова "warez" означає приналежність повідомлення до спаму та неприналежність до теми "Інформаційні технології".

Компонента збирання даних з каналів новин

Завданням компоненти збирання даних з RSS каналів (crawler) є сканування каналів новин та стягування нових повідомлень. Перед початком роботи цієї компоненти в таблицю ob_channel повинні бути занесені адреси ресурсів, які потрібно сканувати.

Crawler функціонує так:

• З таблиці ob_channel вибираємо адресу каналу (поле url) і перевіряємо відповідно до заданого періоду читання (readperiod), чи потрібно сканувати цей канал;

• Стягуємо RSS-файл (фід) за цією адресою, виділяємо з нього повідомлення та заносимо їх в таблицю tdc_message.

У результаті виконання цих дій в тимчасову таблицю tdc_message будуть занесені нові повідомлення. На цьому робота компоненти Crawler завершена, а отримані нові повідомлення слугують вхідними даними для компоненти класифікації.

Компонента класифікації

Компонента класифікації (grabber) призначена для фільтрування, рангування і класифікації нових повідомлень.

Вхідними даними для цієї компоненти є нові повідомлення, котрі містяться в таблиці tdc_message.

Grabber функціонує так. Послідовно, в циклі, виконує такі дії:

• З таблиці tdc_message вибираємо заголовок (title) і текст повідомлення (description);

• Відкидають некоректні повідомлення та повідомлення, які було стягнуто раніше і вміщено в таблиці dc_message.

• Залежно від тематики та слів, які зустрічаються, проводять класифікацію та рангування повідомлення. Це відбувається за допомогою такого SQL-запиту:

```
SELECT ob.id, ob.minrating,
        SUM(IF(tdc.title LIKE od.keywords,$tit_rating,$desc_rating)*od.rating*tdc.rating)
        AS rating
FROM (ob_category ob INNER JOIN od_category od ON ob.id=od.category_id)
        INNER JOIN tdc_message tdc ON ob.theme_id=tdc.theme_id
WHERE tdc.msgid='$tmsg_row->{'msgid'}'
        AND (tdc.title LIKE od.keywords OR tdc.description LIKE od.keywords)
GROUP BY ob.id, ob.minrating
HAVING SUM(IF(tdc.title LIKE
        od.keywords,$tit_rating,$desc_rating)*od.rating*tdc.rating)>=ob.minrating;
```

У цьому запиті $tit_rating та $desc_rating — це коефіцієнти рейтингу повідомлення, які відповідно характеризують приналежність ключового слова до заголовка та тексту повідомлення. Змінна $tmsg_row->{'msgid'} характеризує поточне повідомлення.

Загальний рейтинг повідомлення за кожною категорією обчислюють як суму рейтингу відповідності повідомлення до кожного ключового слова цієї категорії, помноженого на рейтинг RSS-каналу.

Рейтинг відповідності повідомлення до ключового слова категорії обчислюється як добуток рейтингу ключового слова і однієї з констант $tit_rating та $desc_rating залежно від того, чи зустрічається ключове слово в заголовку (title) чи в тексті повідомлення (decription). Повідомлення вважають прийнятним для заданої категорії, якщо його загальний рейтинг не менший ніж мінімальний рейтинг категорії.

• Якщо повідомлення є прийнятним хоча б для однієї з категорій, то воно заноситься в таблицю dc_message.

• Якщо повідомлення не належить до жодної категорії, то зменшується рейтинг RSS-каналу.

Компонента аналізу

Компонента аналізу (analyzer) призначена для аналізу повідомлень на предмет їх важливості та актуальності в часі. Ця компонента аналізує повідомлення на відповідність певним критеріям.

Критеріями можуть бути:

• Актуальність в часі (вибір повідомлення за заданий проміжок часу);

• Рейтинг (вибір повідомлення, які відповідають категоріям у межах заданого проміжку рейтингу);

• Категорія (вибір повідомлень заданих категорій).

Одне повідомлення може відповідати багатьом темам та категоріям.

Компонента аналізу дозволяє формувати список повідомлень, які цікавлять користувача. Analyzer готує список повідомлень для безпосереднього відображення, які виконує Representer.

Компонента відображення

Компонента (representer) призначена для відображення відібраних аналізатором актуальних повідомлень. Форма подання повідомлень – HTML або XML документ (Web-сторінка).

Representer подає повідомлення користувачу у такому вигляді (рис. 2)

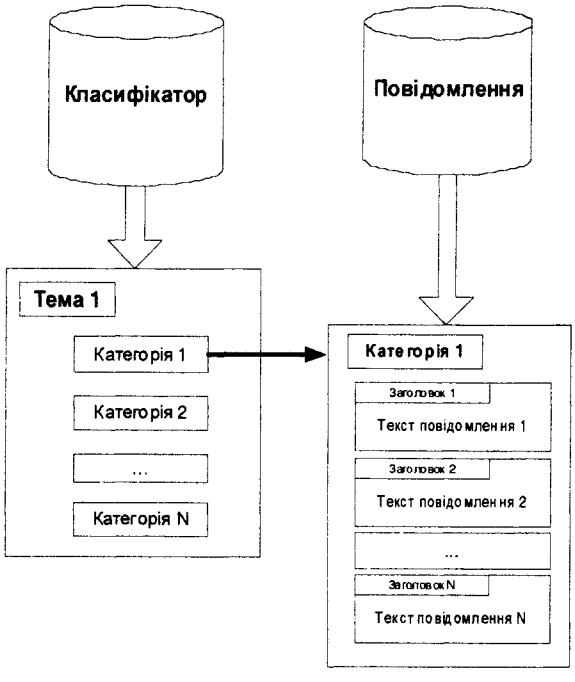

Рис. 2. Схема подання повідомлень користувачу

На рис.2 зображено два типи сторінок:

1. Список категорій за заданою темою;
2. Список повідомлень за заданими критеріями вибору.

Засоби реалізації системи

Система реалізована за допомогою стандартних технологій, які традиційно використовують у Web-середовищі. Зокрема компоненти Crawler; Grabber; Analyzer реалізовані за допомогою мови програмування Perl з використанням стандартних бібліотек, призначених для роботи з RSS-документами.

Компоненту Representer розроблено за допомогою мови Web-програмування PHP та мови створення Web-сторінок HTML. Компонента відображення використовує Web-сервер Apache.

Як СУБД вибрано СУБД, яка найчастіше використовується для створення систем, орієнтованих на WWWW – MySQL.

Використані засоби дають змогу зробити кросплатформну, Інтернет-орієнтовану систему яка здатна агрегувати, класифікувати та рангувати декілька тисяч RSS-каналів. Класифікована інформація надається користувачам та містить посилання на оригінальні документи сайтів з RSS-каналами.

Висновки

У статті розглянуто актуальну задачу побудови системи збирання та аналізу інформації з Інтернет-ресурсів новин. Робота системи ґрунтується на стягуванні новин з RSS-каналів сайтів

новин. Формати поширення новин (RSS, RDF), базовані на XML, стають поширенішими і популярнішими, оскільки вони забезпечують швидкий та зручний обмін інформацією.

Запропонована система збирання та аналізу інформації з Інтернет-ресурсів новин значно спрощує та прискорює процес опрацювання новин. Система володіє можливостями класифікації, рангування, аналізу актуальності повідомлень, тому можна працювати з найсвіжішими та найактуальнішими даними, впорядковувати повідомлення за тематикою та категоріями.

Додатковою функцією системи є кешування інформації RSS-ресурсів та надання кешованої інформації користувачам автоматизованої системи без повторних звернень до серверів RSS ресурсів. У цьому випадку економиться трафік RSS-ресурсів.

Ефективність роботи автоматизованої системи значною мірою залежить від якісного налаштування функцій стягування, фільтрування, рангування і класифікації, що входять до компонент Crawler і Grabber.

1. Жежнич П.І., Кравець Р.Б., Пасічник В.В., Пелещишин А.М. Основні правила побудови семантично відкритих інформаційних систем // Вісник Національного університету "Львівська політехніка" "Інформаційні системи та мережі". – 1999. – №383. – С. 84–95. 2. Жежнич П.І., Кравець Р.Б., Пасічник В.В., Пелещишин А.М. Семантично відкриті інформаційні системи // Вісник Національного університету "Львівська політехніка" "Інформаційні системи та мережі". – 1999. – №383. – С. 73–84. 3. Сєров Ю.О. Технології пошуку та видобування даних у WWW (аналіз проблеми) // Вісник Національного університету "Львівська політехніка" "Інформаційні системи та мережі". – 2003. – №489. – С. 276–286. 4. У чому таємниця популярності блогів? Andriy Peleschyshyn, http://it.ridne.net/blogpopulars. 5. Эффективный сбор новостей. Дмитрий Ландэ, http://infostream.com.ua/publ/iua. 6. Редкостный Синтез Сайтов. Дмитрий Ландэ, http://infostream.com.ua/publ/rss. 7. RSS 2.0 Specification http://blogs.law.harvard.edu/tech/rss.

УДК 681.3.06

М. Назаркевич, Т. Марусенкова
Національний університет "Львівська політехніка",
кафедра автоматизованих систем управління

РОЗРОБКА ГАРНІТУРИ ШРИФТУ ПІВУСТАВ ДЛЯ КОМП'ЮТЕРНОГО СКЛАДАННЯ ТЕКСТУ

Розробку можна використовувати для створення електронних варіантів книг, обкладинок чи рекламних видань із елементами старовини, а також створення електронних видань.

This development can be used for creation of electronic variants of books, covers or publicity editions with the elements of antiquity, and creation of electronic editions.

Вступ

Під шрифтом розуміють сукупність символів відтворення мови, графічно виконаних в одному стилі. Шрифт є інструментом дизайну, за його допомогою привертають увагу до окремих фрагментів тексту, полегшують запам'ятовування, формують сприйняття читача, поліпшують передавання змісту тексту.

Тому важливо правильно підбирати існуючі шрифти для відтворення тієї чи іншої інформації, а в деяких випадках – застосовувати авторські шрифти, які можуть стати "візитною карткою" видавництва.

Отже, задача створення шрифтів є актуальною. Зокрема, важливою задачею є створення комп'ютерних аналогів стародавніх шрифтів для комп'ютерного складання історичних документів.