RSS-Reader - gruparea şi verificarea la "unicitate" a noutăţilor

Gheorghe Mânzat
Technical University of Moldova
manzat.gheorghe@gmail.com

***Abstract* — Article present a consept of information system «RSS – Reader». Presented software concept belongs to RSS-agregator software class but in comparation to other representatives it has some special features of RSS news grouping and uniqueness verification. The work contains: general objectives, conceptual scheme of information system and its basic components, description of informational space of system, realization of news uniqueness verification algorithm on basis of Shingles algorithm.**

***Index Terms* — RSS News reader and analzyer, RSS agregator, Semantic Web, Shingles algorythm imlementation, development and innovation sphere.**

INTRODUCERE

RSS (Rich Site Summary) este conţinutul unui blog, site etc cu o prezentare într-un format special. Foarte multe bloguri, site-uri de ştiri, ziare online îşi fac public conţinutul într-un feed RSS care este accesibil oricărui vizitator.

RSS rezolvă o problema majoră a celor care folosesc Internetul cu regularitate. Permite să fii la curent cu tot ce este nou prin simpla accesare a feed-urilor respectivelor bloguri sau site-uri. În acest mod nu mai este nevoie navigharea fiecărui site în parte. Numărul blogurilor şi site-urilor care oferă opţiunea de RSS feed este în continua crestere [1].

SCOPUL PROIECTULUI

Proiectul are drept obiectiv principal colectarea noutăţilor, precum şi gruparea acestora şi verificarea lor la "unicitate".

Unicitatea conţinutului pentru noutăţi va fi realizată prin intermediul Algoritmului Shingles, astfel ca utilizatorului să nu fie afişate două sau mai multe noutăţi cu acelaşi conţinut.

Procesul de colectare a noutăţilor va fi automatizat, astfel înregistrarea lor se va efectua după un anumit timp specificat de utilizator în preferinţe. Noutăţile pe parcursul colectării conform conţinutului său vor fi grupate după tematică, precum şi după data la care aceasta a fost emisă, fiecare noutate va fi verificată la unicitate cu alte noutăţi, astfel excluzînd faptul de dublare a aceluiaşi conţinut si aceleeaşi noutăţi, ceea ce va crea uşurinţă în utilizarea sistemului precum şi la mărirea randamentului de utilizare a acestuia.

Sistemul va avea un modul de configurare a proxy server-ului, a modului de vizualizare a noutăţilor precum si procentul cu care se va efectua unicitatea noutatilor. Schema conceptuală a sistemului este prezentată în figura 1.

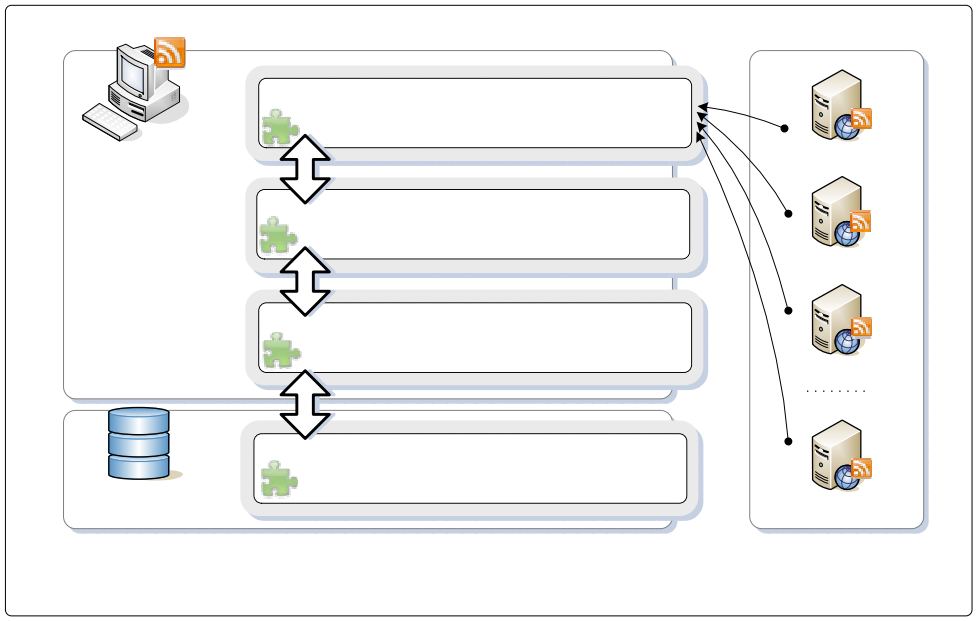

Figura 1.

Sistemul va fi compus din mai multe module. Colectarea noutăţilor se va efectua prin intermediul Modulului RSS Manager. Colectarea acestora se va face în mod automat şi dinamic, în dependenţă de sursele şi setările efectuate de utilizator. După va fi necesar de a verifica fiecare noutate la unicitate, procedura dată va fi efectuată cu ajutorul Algoritmului Shingles. După analiza acestui algoritm a fost necesar de a efectua careva schimbari, din motivul necesităţii preciziei precum şi a timpului de determinare a coincidenţei conţinutului unei noutăţi faţă de alta. Acest algoritm permite de a verifica dacă două obiecte sunt parţial egale după sens sau nu. Prin obiecte se subînţeleg textele sau alte tipuri de date ce sunt destinate pentru transmiterea informaţiei. Presupunem că avem două texte şi e necesar de a stabili dacă conţinutul acestora este practic acelaşi sau nu.

Realizarea algoritmului se efectuează prin intermediul a următoarelor etape [2],[3]:

1. canonizarea textului
2. divizarea textului pe shingle
3. determinarea codurilor hash
4. determinarea subsecvenţelor identice

Mai concret în algoritmul Shingles se realzează compararea codurilor hash al textelor. Pentru a obţine codul hash se poate utiliza CRC32, SHA1, MD5 sau altele [4]. După câte se cunoaşte codurile hash (sau sumele de control) sunt funcţii statice ce sunt foarte sensibile la schimbări, de exemplu codurile hash pentru textele de mai jos se vor diferenţia complet:

- „My war is over.” - codul hash al caruia este 1759088479
- „My war is over!” - codul hash al caruia este -127495474

La determinarea aproximativă a dublicatelor nu ne interesează semnele de punctuaţie (. , ! „ : ? ...), ci ne interesează doar cuvintele (dar nu perechi sau prepoziţii *(şi, deci, sau, nu, până la ...)),* deci este nevoie de adus textele la o forma canonică. Pentru asta se determina un set de semne de punctuaţie şi cuvinte (prepoziţii, cuvinte ajutătoare) de care nu e nevoie şi care trebuiesc eliminate din conţinutul ambelor texte, astfel obţinând forma canonică a acestora.

După obţinerea formelor canonice ale textelor e necesar de a diviza textul pe subsecvente – shingles, fiecare shingle va avea o anumita lungime. În dependenţă de exactitatea rezultatului se alege lungimea unui shingle. Pasul de regulă se alege de lungimea unui cuvînt sau a mai multor, de regulă pentru texte medii lungimea shingle se alege 10 [5].

Divizarea pe shingle are loc prin suprapunerea peste fiecare cuvânt, dar nu la încheietură. Numărul total de shingles va fi egal cu numărul cuvintelor din text minus lungimea shingle plus 1 (`len(source)-(shingleLen-1)`).

Spre exemplu avem următorul text:

„Raţiunea pentru om e dată pentru aceea, ca el sa traiasca raţional, dar nu numai pentru ca el sa înţeleagă că el trăieşte neraţional.” - V. G. Belinschii.

Forma canonică a textului va arăta în următorul mod:

ratiunea om e dată el traiasca rational el înţeleagă el trăieste nerational

Lungimea unui shingle o luăm egală cu 10. După aplicarea celor spuse mai sus vom obţine următoarea listă de shingles:

- sh1 = *ratiunea om e dată el traiasca rational el înţeleagă el*
- sh2 = *om e dată el traiasca rational el înţeleagă el trăieste*
- sh3 = *e dată el traiasca rational el înţeleagă el trăieste nerationa*

După ce este determinată lista cu toate shingles se determină codurile hash pentru fiecare shingle printr-o oarecare funcţie – fie MD5. Deci din exemplul de mai sus obţinem 3 coduri hash:

[1313803605, -1077944445, -2009290115]

Având codurile hash pentru ambele texte, se compară fiecare cod hash din primul tex cu fiecare cod din al doielea text, după ce se memorează numărul de apariţii a acestora. Procentul de similaritate a textelor se determinta din relaţia de mai jos:

p = nrAparitii*2/(len(source1) + len(source2))*100 [6]

După câte se observă, pentru implementarea acestui algoritm este necesar de multe resurse, din acest motiv e nevoie de ales functia pentru determinarea codului hash cât mai optimala.

După analiza şi verificarea noutăţilor la unicitate rezultatele vor fi estimate (prin intermediului Modulului de estimare a rezultatelor) în depednenţă de setările efectuate de utilizator, în caz dacă noutatea va corespunde cerinţelor utilizatorului, aceasta va fi înscrisă în baza de date a sistemului prin intermediul Modulului de gestionare a BD.

REFERINŢE

1. A basic tutorial introduction to RSS feeds and aggregators for non-technical people» Software Garden. http://rss.softwaregarden.com/aboutrss.html
2. A. Broder. On the resemblance and containment of documents. Compression and Complexity of Sequences (SEQUENCES'97), pages 21-29. IEEE Computer Society, 1998. http://ftp.digital.com/pub/Digital/SRC/publications/broder/positano-final-wpnums.pdf
3. Алгоритм Шинглов — поиск нечетких дубликатов текста, http://www.codeisart.ru/python-shingles-algorithm/
4. S. Ilyinsky, M. Kuzmin, A. Melkov, I. Segalovich. An efficient method to detect duplicates of Web documents with the use of inverted index. WWW Conference 2002. http://www2002.org/CDROM/poster/187/
5. A. Chowdhury. Duplicate Data Detection. http://ir.iit.edu/~abdur/Research/Duplicate.html
6. A. Chowdhury, O. Frieder, D. Grossman, M. McCabe. Collection statistics for fast duplicate document detection. ACM Transactions on Information Systems (TOIS), Vol. 20, Issue 2 (April 2002). http://ir.iit.edu/~dagr/2002collectionstatisticsfor.pdf
7. S.-T. Park, D. Pennock, C. Lee Giles, R. Krovetz, Analysis of Lexical Signatures for Finding Lost or Related Documents, SIGIR'02, August 11-15, 2002, Tampere, Finland