Information wants to be free

Pisanje pretraľivača zakona objavljenih u Narodnim novinama sa podrąkom za hrvatski jezik

Dobrica Pavlinuąić <dpavlin@rot13.org> 28.01.2002.

Informacije moraju biti dostupne

Narodne novine (u nastavku NN), izdavač sluľbenog glasila Republike Hrvatske, odlučile su od 15. rujna 2001. godine naplaćivati pristup web izdanjima Narodnih Novina.

Zbog toga su pokrenute mnoge inicijative (uključujući HrOpen-ovu i GONG-ovu) da bi se osigurao pristup i pretraľivanje elektroničkog izdanja NN (jer po nekim interpretacijama to ograničava prava građana Republike Hrvatske na pristup zakonima koji se objavljuju sluľbeno u NN). Nakon toga NN ponovo omogućuju pristup elektroničkom izdanju uz navođenje točne godine izdanja i broja NN.

Na taj način postoji mogućnost da pogledate zakon ako znate točan broj NN, ali ako trebate pronaći zakon bez poznavanja broja i godine izdanja (kao ąto većina povremenih posjetitelja web stranica NN i radi) tada vam je taj "klasični" (kao u knjiľnici) način pretraľivanja NN jednostavno nezadovoljavajući.

Sama prednost NN u digitalnom obliku je, naravno, pretraľivanje. Uklanjanjem te mogućnost (radi financijske dobiti od pretplatnika koji su takvu uslugu spremni platiti) NN su sebi stvorile mogućnost zarade (i pruľanja boljih usluga pretplatnicima), ali istovremeno onemogućile povremene korisnike NN da pronađu zakon koji ih zanima.

To je bila motivacija za implementaciju pretraľivača sadrľaja NN.

Open Source filozofija

Kako se ne bi ponovila priča sa originalnim pretraľivanjem NN, odlučio sam da sam pretraľivač bude objavljen pod Open Source licencom. To je od početka značilo da mogu koristiti gotove dijelove koda koje su razni autori (čiji je popis dan na kraju članka) napisali tokom godina, a mogu biti korisne pri implementaciji pretraľivača.

Osim jasnog sinergijskog efekta takve odluke, on omogućava da bilo tko moľe kasnije prepraviti pretraľivač tako da odgovara njegovim potrebama, ali da istovremeno ta poboljąanja vrati Open Source zajednici, tako da svi moľemo profitirati od toga. S druge strane, to također znači da originalni autor ne moľe ni u jednom trenutku proglasiti pretraľivač samo njegovim i početi naplaćivati njegovo koriątenje.

Kako se koriste gotove i javno raspoloľive komponente, mogu se koristiti znanja koja su drugi ljudi ugradili u njih. Iskreno, na početku ovog projekta, 24.01.2002. moje znanje o načinima pretraľivanja teksta bilo je prilično oskudno.

Implementacija pretraľivača

Prvi korak je bilo traľenje da li na mreľi već postoji neki primjer implementacije pretraľivača u praksi. Nakon duľeg traľenja, shvatio sam da članaka o samoj implementaciji zapravo nema, ali da ima pretraľivača koji rade, licencirani su pod jednom od Open Source licenci i rade dobro.

Međutim, lista zahtjeva za pretraľivač je bila donekle specifična, ne toliko za Narodne novine, koliko za Hrvatski jezik:

- ne inzistiranje na unoąenju naąih slova kod pretraľivanja (konverzija naąih slova prije indeksiranja u czs, tako da kuća i kuca upisani u pretraľivač vraćaju kuća)
- mogućnost upisivanja riječi u naslovu u bilo kojem obliku (dakle, "Zakon o gradnji kuća" će biti pronađen i za upit "zakon kuća" i "kuća zakon" -- to zapravo onemogućava koriątenje jednostavnog like upita u bazu na text polje koje ima naslov)
- koriątenje hrvatskog suffix file-a sa http://cvs.linux.hr/spell/
- jezik implementacije perl (neovisno o ciljnoj platformi -- idealno, cijeli kod pretraľivača bi trebao biti pisan u samo perl-u tako da se kao moľe preseliti na druge platforme bez potrebe da se neki dio kompilira)
- koriątenje relacijske baze podataka za spremanje podataka (PostgreSQL)
- mogućnost naprednog pretraľivanja koriątenjemm and, or, not, + operatora (koriątenjem `Text::Query` modula)
- pretraľivač treba vraćati URLove na sluľbeni site NN (www.nn.hr), ali kod indeksiranja sadrľaja treba ąto manje opterećivati server NN.

Međutim, kako je svaki početak teľak, sama implementacija odvijala se u fazama, jer je jednostavnije početi sa manjim problemom i onda proąirivati funkcionalsnost.

Prvi pokuąaj

Prvi i logičan način pretraľivanja je bio po naslovima zakona. Ako sve naslove pospremimo u RDBMS, logičan način pretraľivanja je SQL upit oblika

```
SELECT * FROM NN WHERE NASLOV LIKE '%riječ%'
```

gdje je "riječ" ono ąto je korisnik pretraľivača upisao. Međutim, to nije i najbolji način pretraľivanja jer ako imamo zakon koji se zove

```
"Zakon o posebnom porezu na kavu"
```

korisnik će imati problema jer će morati upisati točan dio naslova (npr. "porezu na kavu" a ne samo ključne riječi (npr. "porez kava").

Treba naglasiti da se ne pretraľuje zapravo originalni naslov, već njegova kopija koja je pretvorena u samo mala slova i kojoj su hrvatski palatali (ćčľąđ) zamjenjeni s ekvivalentima bez kvačica. Zbog toga će u daljnjem tekstu pod naslov biti podrazumjevan ovaj oblik prilagođen za pretraľivanje. Originalni naslov koristi se samo za prikazivanje rezultata (da bi ispisani naslov imao velika i mala slova kao i naąe znakove).

Drugi pokuąaj

Slijedeći logičan korak bio je pokuąaj da se riječi upisuju u posebnu tablicu, a da se korisnikov upit pretvori u viąestruke upite u bazu. To je nuľno kako bi se mogle upisati riječi za pretraľivanje u bilo kojem redoslijedu.

```
NN                      NASLOVI                 RIJECI
----------------        ----------------        ----------------
ID  NASLOV              ID      RIJEC_ID        ID      RIJEC
----------------        ----------------        ----------------
10  Zakon o posebnom... 10      101             101     zakon
                        10      102             102     posebnom
                        10      103             103     porezu
                        10      104             104     kavu
```

Sve riječi upisuju se u posebnu tablicu, dodaje se vezna tablica koja povezuje tablicu s riječima i tablicu s brojevima NN, a upiti izgledaju kao:

```
SELECT * FROM NN WHERE ...
        ID IN ( SELECT NASLOVI.ID FROM NASLOVI WHERE RIJEC_ID IN (
                SELECT ID FROM RIJECI WHERE RIJEC IN ('porezu','kavu')
                ) GROUP BY NASLOVI.ID )
```

Osim ąto takav pristup rjeąava naą problem i traľene riječi se mogu upisivati bilo kojim redoslijedom, on također ima i nekoliko problema. SQL upiti su prilično sloľeni i moraju se dinamički generirari. Pravilo rule-of-the-thumb za SQL upite je da sub-selecti i operator IN obično nisu dobra ideja ąto se tiče performansi. Drugi problem je u tome ąto sub-selecti nisu podrľani u svim slobodno dostupnim bazama podataka. Iako to nije bio problem za PostgreSQL, bolje je implementirati rjeąenje koje ne zahtjeva toliku funkcionalnost jer bi kasnije prenoąenje na neku drugu bazu moglo predstavljati ozbiljan problem.

Međutim, rjeąnje koje omogućuje takvo pretraľivanje (bez obzira na redoslijed riječi u originalnom naslovu) je jednostavno nuľno te sam već bio spreman implementirati takav pristup kada sam (pretraľujući Internet u potrazi za dosadaąnjim iskustvima u implementaciji pretraľivača) na CPAN-u (Comprehensive Perl Archive Network, www.cpan.org, cpan.linux.hr, arhiva modula za PERL) pronaąao modul `Text::Query` koji tom problemu pristupa na drugi način.

Napredno pretraľivanje

Modul `Text::Query` pristupa problemu pretraľivanja na drugačiji način: definira gramatiku pretraľivanja (slično AltaVista-i) gdje je podrazumijevani operator ili (or), a moľe se eksplicitno zahtjevati prisutnost neke riječi (koriątenjem znaka + ispred riječi) ili njezina odsutnost (koriątenjem znaka - isprijed riječi). Moguće je koristiti i zagrade za grupiranje dijelova upita. Nakon specificiranja upita, on se prevodi u regular expression kojim se onda pretraľuje string. To zapravo znači da se pretraľuje jedno polje u bazi koriątenjem regex-pa. Rezultat je puno jednostavniji upit koji izvrąava sam RDBMS. Takvi upiti su donekle specifični za pojedine RDBMS-ove, ali `Text::Query::BuildSQL` podrľava PostgreSQL, MySQL i Fulcrum. Koriątenjem tog modula mogli smo se "vratiti" na jednostavan model s samo jednom tablicom koja se onda pretraľuje jednostavnim SQL upitom koji ima regex.

Razni oblici riječi

Jedna od specifičnosti hrvatskog jezika su mnogobrojni oblici iste riječi. Zbog toga je naą primjer od prije bio donekle proizvoljan. Naime, upit "porez kava" zapravo ne bi naąao zakon "Zakon o posebnom porezu na kavu", dok bi upit "porezu kavu" imao kao rezultat traľeni zakon. Zbog toga je zanimljivo naći u kojim se sve oblicima neka riječ moľe zapisati i u bazu zapisati sve moguće oblike te riječi. To zapravo znači da će zakon "Zakon o posebnom porezu na kavu" zapravo biti zapisan kao "zakon o posebnom posebno posebna posebne posebnu posebni posebnoh posebn posebnim posebnih posebnog posebnoj porezu poreza poreze porezi porezom porez porezo

porezim porezih porezog porezoj porezem na kavu kava kave kavi kavom kav kavo kavim kavih kavog kavoj kavem" da bi se mogao pronaći bilo koji oblik riječi u naslovu. Neke od tih kombinacija nisu ispravne, ali kako su nastale statističkom analizom, a ne gramatičkom, za početak ćemo se morati zadovoljiti time.

Sam postupak pronalaženja svih mogućih oblika riječi je nešto u što se nisam želio upuštati sam: ipak se ja ne bavim lingvistikom. Međutim, imao sam sreće: upravo prije nekog vremena pojavio se hrvatski riječnik za ispell. Taj riječnik dolazi i s tzv. affix datotekom koja opisuje prefixe i sufixe koji se pojavljuju u nekom riječniku.

Implementacija čitanja affix file u perlu je bila relativno jednostavna. Nakon toga dobili smo mogućnost generiranja svih mogućih oblika riječi koji se nalaze u affix datoteci. Naľalost, osnovna funkcija affix datoteke je da se smanji veličina dictionary datoteke, a ne da dobijemo sve moguće gramatičke oblike neke riječi. Drugi problem je da je affix datoteka ograničena na samo 26 različitih kombinacija jer se svaki nastavak definira svojim slovom.

Kreiranje savršenog affix-a

U ovom trenutku je veoma pomogao Denis Lacković, originalni autor affix datoteke za hrvatski jezik koji je poslao raw file koji je rezultat programa findaffix i koji sadrľi mnogo više od 26 različitih nastavaka. Pomoću takvih podataka mogu se generirati sve moguće varijacije za neku riječ. Na taj način naš pretraživač je postao još primjenjiviji i jednostavniji za korištenje jer korisnici ne moraju razmišljati o obliku riječi koju su upravo upisali.

Koraci za budućnost

Ono što ostaje za napraviti u vezi pretraživača je:

- kontaktirati GONG i ponuditi im zamjenu postojećeg pretraživača ovime (ili barem stavljanje linka na stranice ovog pretraživača, `nn.rot13.org`)
- objaviti kompletni source kod na `www.rot13.org/~dpavlin/nn.html`
- poboljšati "gramatički" affix file za hrvatski jezik

Praktična usporedba s GONG-ovim pretraživačem

Kao test koliko je ovaj pretraživač bolji, i da li je ova implemetacija uopće imala smisla, otišao sam na GONG-ov pretraživač NN i tamo upisao riječi "kava zakon". Rezultat je bio: "Traženi podaci nisu pronađeni !". Sa druge strane, moj pretraživač je ispisao veliki broj zakona u kojima se pojavljuju riječi "zakon" ili "kava" u bilo kojem obliku. Da bi dobili samo zakone u kojima se pojavljuju obje riječi moramo upisati to kao: "+zakon +kava". Tako ćemo dobiti točno tri zakona koji se time bave.

Vraćanje Open Source zajednici

Napokon, 12. veljače 2002. modul `Lingua::Spelling::Alternative` je poslan na CPAN. Na taj način će svi korisnici perl-a imati koristi od dijela koda koji je napisan za potrebe ovog pretraživača.

Naravno, i sam pretraživač je prepravljen da koristi sam modul, a kako modul omogućava i učitavanje datoteka koje su rezultat findaffix programa (dakle bez limita na broj zapisa kao affix datoteke) sama mogućnost pretraživanja je time povećana.

Modul moľete dohvatiti iz CVS-a, sa CPAN-a ili direktno sa mojih stranica.

Zahvale ljudima iz Open Source zajednice

- Hrvatski spell checker
  Denis Lacković (denis.lackovic@fer.hr)
- perl, programski jezik
  Larry Wall (larry@wall.org) i mnogi drugi
- `Text::Query` CPAN modul
  Eric Bohlman (ebohlman@netcom.com)
  Loic Dachary (loic@senga.org)
- `Text::Query::BuildSQL` CPAN modul
  Loic Dachary (loic@senga.org)
- PostgreSQL relacijska baza podataka
  Lista ljudi koji su radili na Postgres-u (iz kojeg je nastao kasnije PostgreSQL) je toliko duga, da vas jednostavno samo mogu usmjeriti na web stranice.

Povijest ovog dokumenta

- 2002-01-28 napisana prva verzija ovog dokumenta
- 2002-02-12 modul `Lingua::Spelling::Alternative` poslan na CPAN
- 2002-02-13 Pretraľivač izaąao u Hrvatskom web-u, dijelu Internet monitora