Revista Română de Interacţiune Om-Calculator 1 (2008), 57-72

Algoritmi de generare de paronime pentru corectarea malapropismelor

Costin-Gabriel Chiru, Ştefan Trăuşan-Matu, Traian Rebedea

Universitatea „Politehnica" Bucureşti
Splaiul Independenţei nr. 313
Institutul de Cercetări în Inteligența Artificială
Calea 13 Septembrie nr. 13 – Bucureşti, România
E-mail: chirucos@gmail.com, stefan.trausan@cs.pub.ro.com, trebedea@gmail.com

Rezumat. Paginile Web au fost folosite intensiv în ultimul timp pentru a extrage în mod automat sau semiautomat informaţii utile. Datorită naturii deschise a Web–ului, textele care să nu aibă greşeli reprezintă excepţii foarte rare. Cel mai răspândit tip de greşeala întâlnit în textele de pe Internet este malapropismul şi de aceea s–au căutat algoritmi pentru detectarea şi corectarea acestora. Algoritmii de detectare a malapropismelor se bazează pe coeziunea textelor în timp ce algoritmii de corectare a acestora folosesc dicţionare de paronime precompilate. De aceea, este foarte important să fie identificaţi algoritmi eficienţi de generare a paronimelor. În lucrarea de faţă se face o prezentare a paronimelor în general, precum şi a metodelor prin care se poate construi un dicţionar de paronime. De asemenea, se prezintă principalele greşeli care conduc la apariţia malapropismelor, precum şi modul în care aceste erori pot fi corectate cu ajutorul unui dicţionar de paronime.

Cuvinte cheie: Paronime, Malapropisme, Prelucrarea limbajului natural

1. Introducere

În ultimul timp se poate observa trecerea la o folosire intensivă a paginilor Web cu scopul de a extrage în mod automat sau semiautomat informaţii utile din aceste pagini. Datorită faptului că achiziţia de cunoştinţe reprezintă primul pas dintr-o serie de procesări oarecare, realizarea acesteia cu o acurateţe cât mai bună, capătă o importanţă deosebită. Cu cât prelucrările sunt mai complexe şi constau în mai mulţi paşi, cu atât creşte importanţa primului pas, deoarece erorile acestuia se propagă şi se amplifică în următoarele etape. Drept urmare, a crescut nevoia de a avea la dispoziţie texte scrise corect, pentru ca informaţiile obţinute să fie de cât mai bună calitate.

Cum web-ul are o natură deschisă, oricine având acces liber şi putând să publice orice, nu se poate pune bază pe corectitudinea textelor extrase din punctul de vedere al limbajului folosit şi de aceea este nevoie de instrumente automate de detectare a erorilor şi de corectare a acestora. Studiile lingvistice bazate pe rezultatele furnizate de motoarele de căutare au arătat în repetate rânduri că texte care să nu aibă nici o greşeala reprezintă excepţii foarte rare în multitudinea de texte scrise într-o anumită limbă şi, mai mult, că acest lucru este valabil pentru orice limbă (Gelbukh şi Bolshakov, 2004).

Cel mai răspândit tip de greşeala întâlnit în textele de pe Internet este malapropismul. Datorita acestui lucru, s-au căutat algoritmi pentru detectarea şi corectarea malapropismelor, care să propună alternative ori de cate ori se identifică o astfel de situaţie. Algoritmii de detectare a malapropismelor se bazează pe coeziunea textelor în timp ce algoritmii de corectare a acestora folosesc dicţionare de paronime precompilate din care să se poată extrage rapid variante de înlocuire a cuvintelor greşite. De aceea, este foarte important să fie identificaţi algoritmi eficienţi de generare a paronimelor.

Lucrarea continuă cu o introducere în problematica paronimelor. Apoi se introduc şi se compară mai mulţi algoritmi de generare de paronime.

2. Noţiuni generale despre paronime

2.1 Definiţia paronimelor

*Malapropisme*le apar datorită folosirii greşite a unui cuvânt în locul altuia care are în compoziţie litere/sunete asemănătoare dar care sunt incompatibile din punct de vedere semantic în contextul respectiv. Detectarea si corectarea lor este posibilă dacă se cunosc paronimele cuvintelor. Cuvântul *paronim* are origine greacă: *para* „lângă" şi *onoma* „nume". Paronimele pot fi definite în mai multe feluri, în funcţie de perspectiva considerată. Astfel, dacă se consideră că paronimele sunt generate din greşeală (de exemplu, sunt malapropisme), ele pot fi definite ca fiind cuvinte asemănătoare după formă, care pot fi uşor confundate în vorbire aşa cum menţionează Grădinaru (2007).

O altă definiţie accentuează dimensiunea asemănării semantice, paronimele fiind considerate „cuvinte care au aceeaşi rădăcină" Grădinaru (2007). Oricum, toate definiţiile paronimelor se caracterizează în general

prin fonetism apropiat dar sensuri diferite Grădinaru (2007), conform şi definiției dată de către situl DEX Online (varianta on-line a DEX – Dicționar Explicativ al Limbii Române): „**PARONÍM,** *paronime*, s.n. Cuvânt asemănător cu altul din punctul de vedere al formei, dar deosebit de acesta ca sens (şi ca origine)".

O perspectivă foarte interesantă asupra fenomenului paronimiei este cea pragmatică, care se ocupă de *atracția paronimică*, legată de imaginarul lingvistic şi cu o importanță foarte mare asupra fenomenului limbii şi vorbirii: "Atracția paronimică creează un teren propice pentru jocurile de cuvinte, pentru discursul paremiologic, repetat şi cel aforistic. În acelaşi context, au fost reliefate trăsăturile paronomazei (caracterul lapidar, elipsa, structura binară, funcția mnemotehnică ş.a.) care au menirea de a captiva publicul cititor/ascultător şi a spori calitatea, eficacitatea şi expresivitatea limbajului." Grădinaru (2007).

2.2 Clasificarea paronimelor

Conform (Bolshakov şi Gelbukh, 2003), paronimele pot fi clasificate în funcție de natura lor în trei mari categorii:

1. ***Paronime literale*** – cuvinte care diferă de cuvintele de bază prin câteva litere. Aceste cuvinte sunt apropiate în spațiul şirurilor de litere (ex: barcă, marcă). Ele pot fi , la rândul lor clasificate după cum urmează Grădinaru (2007):

    - **Paronime alcătuite din acelaşi număr de foneme, dar cu distribuție diferită**.

      aerometrie – areometrie, aerometru – areometru, antinomie – antonimie, cardan – cadran, manej – menaj, monogramă – nomogramă, revela – releva.

    - **Paronime cu foneme perechi** (alcătuite din acelaşi număr de foneme, dar unui fonem dintr-un cuvânt îi corespunde un alt fonem în celălalt cuvânt):

        - cu alternanță vocalică

          accident – incident, afectiv – efectiv, aferent – eferent, alimentar – elementar, aluzie – iluzie, alocație – alocuție, pronume – prenume, proscris – prescris, sabot – sabat,

modela – modula, iminent – imanent, imposibil – impasibil.

- cu alternanță consonantică

  abces – acces, abil – agil, acid – avid, bison – vison, constanță – constantă, infecta – infesta, escadrilă – espadrilă, lacună – lagună, dezinfecție – desinsecție, fluvial – pluvial, antigel – antigen, mangustă – langustă. adjuvant – adjutant, agape – agave – agate, şifon – sifon.

- **Paronime cu un fonem în plus:**
  - cu vocală inițială: ***credita – acredita, locație – alocație, lustru – ilustru.***
  - cu consoană inițială: ***estival – festival, latitudine - platitudine.***
  - cu vocală intercalată: ***artrită – arterită, predicție – predicație, etologie – etiologie.***
  - cu consoană intercalată: ***cauză – clauză, facțiune – fracțiune.***

2. ***Paronime fonetice*** – cuvinte care se citesc la fel (spre deosebire de cele anterioare) dar diferă de cuvintele de bază prin câteva litere. În limba română, deoarece cuvintele se citesc aşa cum se scriu, acest gen de paronime nu apare, dar în alte limbi frecvența acestora poate fi destul de mare (de exemplu, în engleza: *sun - son* sau *right - write – Wright,* vezi Gelbukh şi Bolshakov, 2004).
3. ***Paronime morfemice*** – cuvinte care au aceeaşi rădăcina şi parte de vorbire, dar diferă prin sufixele şi/sau prefixele care se adăuga rădăcinii. Aceste cuvinte sunt apropiate în spațiul şirurilor de simboluri morfemice.
   - **Paronime cu acelaşi radical dar cu afixe diferite**
     - Paronime sufixale, care diferă prin sufixele anexate la radical

       ceremonial – ceremonios, flotă – flotație, formal – formalism – formație – formă – format, literă – literal – literar - literat, numeral – numerar - popular – populist, citit – citire – cititor.

- Paronime prefixale, care diferă prin prefixe:

  defula – refula, revoluție – involuție, conflagrație – deflagrație, explozie - implozie.

Paronimele mai pot fi clasificate şi în funcție de distanța dintre cele două cuvinte, aşa cum se arată în (Gelbukh şi Bolshakov, 2004) (ex: paronime de distanța 1: latitudine-platitudine, armă-artă, care-acre, parc-arc; paronime de distanța 2: arc-act-artă-acră, faceți-asceți, etologie-etimologie şamd.).

O altă clasificare poate fi făcută din perspectivă pragmatică, a contextului folosirii lor:

- Malapropisme, greşeli.
- Cuvinte apărute prin atracție paronimică, fenomen considerat de mulți o ”etimologie populară” Grădinaru (2007).
- Paronime introduse conştient, în scop artistic sau stilistic.

Studii referitoare la numărul paronimelor au fost efectuate pentru limbile spaniolă şi rusă, limbi puternic flexionate (în engleză, ”inflexional”). Astfel, limba spaniolă are în jur de 800.000 de forme de cuvinte (în engleză, ”word forms”) în timp ce limba rusă are aproximativ 1.200.000 (față de limba engleză care are în jur de 300.000). Conform sitului DEX Online, limba română are în jur de 365.000 de forme. Pe situl Academiei Române se specifică existența unui dicționar pentru limba romana având aproximativ 150.000 de cuvinte. Conform (Gelbukh şi Bolshakov, 2004), pentru limba spaniolă s-a construit un dicționar de paronime, în care s-au introdus în total 114.393 de forme morfologice gramaticale (grameme, în engleză, ”grammemes”) rezultând 38,379 paronime (ceea ce reprezintă aproximativ 33.6%). În (Bolshakov şi Gelbukh, 2004), se prezintă un dicționar pentru limba rusă ce conține aproximativ 120.000 de grammeme şi în care s-au identificat 86.600 de conexiuni de tip paronime morfemice şi 24.200 de conexiuni de tip paronime literale.

Studiile au arătat că în orice limbă, numai un număr limitat de cuvinte au paronime, precum şi faptul că grupurile de paronime sunt în medie destul de mici. De aceea, s-a ajuns la concluzia că cel mai bine este să se întocmească un dicționar de paronime înainte de utilizarea acestora (Bolshakova et al, 2005).

S-a demonstrat că utilizarea paronimelor literale poate duce la o scădere drastică a căutării de alternative pentru cuvintele întâlnite în malapropisme (căutarea scade de pâna la 360 ori), în timp ce paronimele morfemice permit

corectarea erorilor care nu au fost încă studiate/întâlnite şi care sunt specifice vorbitorilor ce nu au limba respectiva ca limba materna (Bolshakov şi Gelbukh, 2003).

Tot în (Bolshakov şi Gelbukh, 2003) este specificat faptul că, pentru orice limbă, deşi în medie cuvintele se află la distanţe mari unele de altele, totuşi acestea tind să se organizeze în grupuri de paronime.

2.3 Modul de creare al paronimelor

Paronimele sunt cuvinte obţinute prin aplicarea unor operaţii asupra literelor/sunetelor unor alte cuvinte sau ca urmare a aplicării de sufixe/prefixe. În cazul în care se doreşte obţinerea unor paronime literale/fonetice de distanţa 1, acest lucru poate fi realizat plecând de la cuvântul de bază şi realizând una din operaţiile descrise în cele ce urmează:

- înlocuirea unei litere cu o alta în cazul paronimelor literale (ex: armă-artă), respectiv a unui sunet cu altul în cazul paronimelor fonetice;
- permutarea a 2 litere/sunete adiacente între ele (ex: care-acre);
- adăugarea în interiorul cuvântului de bază a unei noi litere (ex: latitudine-platitudine) sau a unui nou sunet;
- ştergerea unei litere sau a unui sunet din cuvântul de bază (ex: clauză-cauză).

În cazul în care se doreşte obţinerea unor paronime de distanţe mai mari, se pot aplica în mod repetat mai multe operaţii din cele descrise mai sus.

2.4 Corectarea paronimelor

În cazul în care se doreşte corectarea paronimelor literale de distanţă 1, trebuiesc făcute NR = A * (2 * L + 1) + L – 1 comparaţii, unde A = dimensiunea alfabetului iar L = lungimea cuvântului. Defalcat NR este obţinut în felul următor:

- înlocuirea unei litere cu o alta din alfabet (pe fiecare poziţie din cele L ale cuvântului, poate să apară una din celelalte litere din alfabet, rezultând L * (A – 1) posibilităţi);
- permutarea a 2 litere adiacente între ele (având L litere în cuvânt, se pot crea L – 1 grupuri de câte două litere adiacente, rezultând L – 1

posibilități);

- adăugarea în interiorul cuvântului a unei noi litere (noua literă poate lua orice valoare din alfabet, iar ea poate fi inserată oriunde în noul cuvânt ce va avea L + 1 litere, rezultând (L + 1) * A posibilități);
- omiterea unei litere din cuvânt (oricare literă din cuvânt poate să lipsească, rezultând L posibilități).

NR poate ajunge la valori de ordinul sutelor, mergând până la 500-600 în cazul cuvintelor lungi (ex: pentru un cuvânt de 9 litere este nevoie de 31 * (2 * 9 + 1) + 9 – 1 = 597 operatii). Cu cât cuvântul este mai lung cu atât este mai mare valoarea lui NR. În cazul în care se doreşte identificarea şi corectarea paronimelor de distanță 2, acest număr poate ajunge la 360.000 de comparații (Gelbukh şi Bolshakov, 2004). Acest lucru poate fi realizat prin aplicarea în mod repetat a algoritmilor necesari detectării paronimelor de distanță 1, cu observația că aceşti algoritmi trebuiesc modificați astfel încât să rețină şi formele care nu se regăsesc în dicționar, dar care printr-o modificare ulterioară pot duce la identificarea unui cuvânt care este în dicționar.

În general nu se încearcă identificarea paronimelor de distanță mai mare ca 2 deoarece numărul de teste necesare pentru a obține cuvântul inițial devine mult prea mare (creşte exponențial) făcând aplicația impracticabilă. În plus, este posibil ca pornind de la un anumit cuvânt, să se obțină două cuvinte diferite prin acelaşi număr de paşi dar realizați în altă ordine, făcând astfel imposibila corectarea.

Corectarea paronimelor fonetice este similară cu cea a paronimelor literale.

3. Algoritmi posibili pentru generarea paronimelor

3.1 Algoritm generic pentru identificarea paronimelor literale/fonetice

Alg_generic

- se pleacă de la un dicționar (o listă indexată) ce conține toate cuvintele din limbă reprezentate fonetic sau literar;
- se identifică două cuvinte candidate la formarea unei perechi de paronime;

- se testează dacă cele două cuvinte formează intr-adevăr o pereche de paronime (cu alte cuvinte, se verifică dacă cele doua cuvinte sunt înrudite ca sens).

În cazul în care nu se doreşte folosirea unui dicţionar care să conţină toate formele unui cuvânt (din motive de viteză a prelucrărilor) se poate folosi un dicţionar de leme, dar trebuie ţinut cont de faptul că este posibil ca unele paronime să se piardă (datorită faptului că nu toate lemele sunt regulate, iar doar prin reţinerea lemei se pierd formele neregulate ale cuvântului respectiv). Un exemplu în acest sens îl constituie paronimul de distanţă 2 „faceţi – asceţi", care, în momentul în care se folosesc leme în loc de cuvinte, devine „fac – ascet" care este un paronim de distanţă 4. Cum în general se reţin numai paronimele de distanţă maxim 2, acest paronim se pierde în cazul în care se folosesc leme.

Identificarea cuvintelor candidate la formarea unei perechi de paronime literale se poate face prin doua metode clasice: una pasivă şi una generativă. De asemenea, dacă procentajul de paronime din limba română respectă proporţiile identificate în limbile spaniolă şi rusă, atunci s-ar putea aplica şi un algoritm aleator pentru identificarea paronimelor. Un astfel de algoritm este condiţionat de un procentaj ridicat al elementelor căutate. Dacă acest procentaj ajunge la 20% - 30%, atunci astfel de algoritmi pot produce rezultate foarte bune într-un interval de timp scăzut. Trebuie însă specificat că rezultatele sunt cu atât mai bune cu cât algoritmul este lăsat să ruleze mai mult timp. De asemenea, trebuie precizat că două rulări consecutive ale aceluiaşi algoritm pot duce la soluţii diferite. În continuare vom prezenta câţiva algoritmi care pot fi folosiţi pentru realizarea unui dicţionar de paronime literale.

3.1.1 Algoritm orientat pe dicţionar (algoritm pasiv)

Un astfel de algoritm pleacă de la perechi de cuvinte şi testează dacă ele pot fi candidate pentru a alcătui o pereche de paronime sau nu.

Alg_1

pentru fiecare cuvânt c din dicţionar

pentru fiecare cuvânt w din dicţionar (w != c)

verifică_paronime(c, w) //verifică dacă cuvintele sunt sau nu

paronime

dacă sunt paronime

atunci se reţin perechile corespunzătoare

altfel continuă cu următorul cuvânt din dicţionar

3.1.2 Algoritm orientat pe cuvânt (algoritm generativ)

Un astfel de algoritm pleacă de la un cuvânt şi încearcă să genereze toate cuvintele care împreună cu cuvântul de la care s-a plecat pot alcătui o pereche de paronime.

Alg_2

pentru fiecare cuvânt c din dicţionar

pentru fiecare literă din cuvânt (c[i])

dacă ştergând litera c[i] se obţine un cuvânt care să existe în dicţionar

atunci se reţin perechile corespunzătoare //*pentru cazul în care se şterge o literă*

pentru fiecare literă j din alfabet

dacă j!=c[i] şi înlocuind litera c[i] cu litera j se obţine un cuvânt care să existe în dicţionar

atunci se reţin perechile corespunzătoare //*pentru înlocuirea unei litere cu o alta*

dacă adăugând litera j înaintea literei c[i] se obţine un cuvânt care să existe în dicţionar

atunci se reţin perechile corespunzătoare //*pentru omiterea unei litere înainte de litera curentă*

dacă c[i] este ultima literă din cuvânt

pentru fiecare literă j din alfabet

dacă adăugând litera j înaintea literei c[i] se obţine un cuvânt care să existe în dicţionar

atunci se reţin perechile corespunzătoare //*pentru*

omiterea unei litere după ultima litera din cuvânt

pentru fiecare literă c[j] din cuvânt cu j > i

dacă interschimbând literele între ele se obține un cuvânt care să existe în dicționar

atunci se reține perechile corespunzătoare //*pentru interschimbarea a două litere între ele, chiar dacă acestea nu sunt adiacente → dacă se doreşte adiacența, se impune condiția j = i + 1*

3.1.3 Algoritm aleator

Alg_3

până la oprire repetă

generează aleator două numere (n1 şi n2) cuprinse între 1 şi numărul maxim de cuvinte

dacă n1 = n2

atunci mai generează un număr

altfel dacă verifică_paronime(n1, n2)

atunci se reține perechile corespunzătoare

3.1.4 Funcția pentru a verifica dacă două cuvinte pot fi paronime

Verifică dacă două cuvinte au structură potrivită pentru a putea forma o pereche de paronime.

verifică_paronime(c, w)

dacă diferența între numărul de litere al celor 2 cuvinte este >= 2

atunci întoarce fals //*nu sunt paronime de distanță 1*

dacă diferența între numărul de litere al celor 2 cuvinte este 1

atunci

pentru fiecare literă din cuvântul mai lung

dacă prin ştergerea acesteia se obține cuvântul mai scurt

atunci **întoarce** adevărat

altfel **întoarce** fals

altfel //*diferența e 0*

dacă literele celor două cuvinte sunt aceleaşi **ŞI** în acelaşi număr

atunci **întoarce** adevărat //*în acest caz detectez şi paronime de distante mai mari*

altfel

dacă cuvintele variază prin mai mult de 1 literă (eventual prin XOR între literele celor două cuvinte)

atunci **întoarce** fals

altfel **întoarce** adevărat

Pentru a determina paronimele literale de ordinul 2, trebuie să se folosească în prima fază o variantă modificată a algoritmilor de mai sus care să nu mai verifice dacă respectivele cuvinte sunt sau nu în dicționar, ci pur şi simplu să rețină toate şirurile de caractere la care s-a ajuns printr-o operație oarecare. În continuare, în a doua etapă se poate folosi unul din algoritmii propuşi mai sus pornind însă de la întreaga listă produsă în primul pas de algoritmul modificat. De asemenea, trebuie avut în vedere că în această etapă, şirurile de litere care se dovedesc a fi cuvinte din dicționar, trebuiesc comparate cu cuvântul de bază, de la care s-a plecat în etapa întâi. De aceea trebuie modificată şi metoda în care sunt comparate aceste cuvinte sau să se încerce aceeaşi abordare ca şi în cazul algoritmilor (aplicarea de două ori a metodei verificare, dar reținând şirurile de caractere intermediare provenite din prima aplicare).

Pentru determinarea paronimelor fonetice se poate adopta aceeaşi abordare ca în cazul paronimelor literale, cu observația că în timp ce în reprezentarea literară distanța dintre două cuvinte poate să fie 1 sau 2 sau chiar mai mare, în reprezentarea fonetică acestea pot avea aceeaşi formă. Un exemplu este limba engleză unde avem cuvinte de genul right - write (Gelbukh şi Bolshakov, 2004) unde distanța literară este mult mai mare (4) decât cea fonetică (0). Din această cauză, este posibil ca numărul de paronime fonetice să fie mult mai mare decât cel de paronime literale. În acelaşi timp, sunt limbi în care nu există sau există pe o scară foarte

restrânsă un astfel tip de paronimie (ex. limba română), şi folosirea acestui algoritm poate încetini obținerea rezultatelor, în loc să o ajute.

După determinarea perechilor care sunt candidate pentru a deveni paronime, trebuie verificat dacă nu cumva cele două cuvinte sunt înrudite ca sens, făcând parte din aceeaşi familie lexicală, caz în care nu mai sunt paronime. Acest lucru se poate face în mai multe feluri în funcție de resursele de care se dispune, dintre care menționăm două variante care ni se par mai eficiente:

- dacă cuvintele sunt grupate pe familii lexicale, se verifică dacă cele două cuvinte fac sau nu parte din aceeaşi familie lexicală. Dacă nu fac parte, atunci am obținut o pereche de paronime;
- dacă este posibil să se botina lemele cuvintelor respective, atunci fiecare cuvânt dintr-o astfel de pereche este adus la lema din care a fost obținut, iar lemele sunt comparate. Dacă acestea diferă, înseamnă că am obținut o pereche de paronime;
- dacă se dispune de un translator pentru limba engleză, se pot traduce cuvintele în engleză, pentru ca după aceea să se verifice în WordNet dacă cele doua cuvinte fac parte din acelaşi synset. În caz contrar, înseamnă că cele două cuvinte formează o pereche de paronime.

3.2 Algoritm generic pentru identificarea paronimelor morfemice

Determinarea paronimelor morfemice este un pic diferită de metodele de determinare a celorlalte tipuri de paronimie deoarece în acest caz se pleacă de la aceeaşi rădăcină, şi prin intermediul adăugării de prefixe şi/sau sufixe, se încearcă obținerea unor noi cuvinte care să fie diferite ca sens de primele cuvinte. Pentru rezolvarea acestei probleme propunem următorul algoritm care pleacă de la lemele cuvintelor:

- se pleacă de la
  - o listă ce conține toate lemele din limbă
  - un dicționar (o listă indexată) ce conține toate cuvintele din limbă
  - o listă ce conține toate prefixele din limbă
  - o listă ce conține toate sufixele din limbă
- pentru fiecare lemă din listă

  - se încearcă completarea lemei cu sufixe/prefixe din listă până se obțin două cuvinte care să fie în dicționar
- se testează dacă cele două cuvinte formează intr-adevăr o pereche de paronime (se verifica dacă cele doua cuvinte sunt înrudite ca sens sau nu)

3.4 Discuție pe baza complexității diferiților algoritmi propuşi

Complexitatea algoritmului Alg_1 este:

O*(dimensiune_dicționar^2 * dimensiune_medie_cuvinte).*

Complexitatea algoritmului Alg_2 este:

O*(dim_dicț* dim_medie_cuvinte * nr_cuvinte_alfabet * tp_nec_căutării_cuvânt_în_dicț).*

Pentru algoritmul aleator Alg_3 nu este relevantă complexitatea, ci numărul de paşi după care probabilitatea de a fi găsit toate paronimele ajunge la 99.9%, respectiv timpul de execuție necesar atingerii acestui punct.

Complexitatea amortizată a metodei/funcției de determinare a situației în care două cuvinte pot deveni candidate la formarea unei perechi de paronime este **O*(dimensiune_medie_cuvinte)***.

Această complexitate, precum şi cea a algoritmilor Alg_1 şi Alg_2 pot fi îmbunătățite în cazul în care cuvintele din dicționar sunt ordonate după lungimea cuvintelor, astfel încât să avem cuvintele de aceeaşi lungime grupate cât mai aproape. În acest fel, se poate construi un vector care să specifice pentru fiecare număr de litere din cuvânt, care sunt indecşii între care se pot căuta paronime, astfel încât să se reducă mult căutările. Astfel, ştiind care este lungimea unui cuvânt (fie L această lungime) şi faptul că paronime de distanță 1 nu se pot găsi decât fie la cuvinte care au aceeaşi lungime, fie la cuvinte de lungime egală cu lungimea cuvântului inițial +/- 1, atunci se restrânge căutarea în intervalul [indexul primului cuvânt de lungime L-1; indexul ultimului cuvânt de lungime L+1]. De asemenea, se poate elimina verificarea faptului că diferența dintre cele două cuvinte trebuie să fie < 2 în cadrul metodei/funcției verifică_paronime(c, w). Această îmbunătățire este cu atât mai vizibilă cu cât numărul de litere dintr-un cuvânt creşte, deoarece în astfel de situații numărul de cuvinte din dicționar care se încadrează în intervalul respectiv scade foarte mult, astfel scăzând şi cantitatea de informație ce trebuie verificată în cadrul căutărilor

făcute.

O altă metodă de îmbunătăţire a performanţelor acestor algoritmi pleacă de la observaţia că într-o limbă cuvintele tind să fie organizate în grupuri de paronime, menţionată în (Bolshakov şi Gelbukh, 2003). Astfel, prima dată când se obţine o pereche de cuvinte care sunt candidate la formarea unei perechi de paronime, se formează un grup care să conţină cele două cuvinte. Oricând în viitor unul din aceste cuvinte devine candidat alături de un al treilea cuvânt pentru a crea o pereche de paronime, se verifică dacă noul cuvânt poate fi introdus în grupul determinat de primele cuvinte. Acest lucru este posibil numai în situaţia în care acest cuvânt poate forma o pereche de paronime cu oricare cuvânt existent în grupul respectiv. Dacă acest lucru nu este posibil, atunci se salvează perechile candidate la formarea unei perechi de paronime obţinute prin combinarea acestui al treilea cuvânt cu cele deja existente în grup dacă există astfel de perechi, după care se trece mai departe. Dacă în schimb cuvântul poate fi adăugat în grupul respectiv, atunci se salvează şi noile perechi candidate şi după aceea se verifică dacă cuvântul respectiv nu face la rândul lui parte dintr-un alt grup. În cazul în care nu face parte, se adăuga pur şi simplu noul cuvânt în vechiul grup. Dacă cuvântul făcea parte dintr-un grup, atunci se încearcă unirea celor două grupuri prin verificarea tuturor combinaţiilor posibile între cuvintele acestora. În cazul în care unirea este posibilă, se salvează toate combinaţiile posibile, iar după aceea cele două grupuri de cuvinte sunt unite. În caz contrar, se reţin numai combinaţiile posibile, eventual adăugând în cel de-al doilea grup cuvintele primului grup care sunt paronime cu toate cuvintele din acel grup. O astfel de abordare poate conduce la identificarea mult mai rapidă a paronimelor, astfel reducând mult spaţiul căutărilor precum şi eliminând o serie de operaţii care astfel nu mai sunt necesare.

Algoritmul Alg_2 poate fi şi el îmbunătăţit dacă luăm în considerare faptul că, în general dicţionarele de paronime sunt folosite pentru a corecta malapropismele făcute din neatenţie. Acest lucru poate însemna că în momentul în care se înlocuieşte o literă cu alta, această nouă literă se află pe tastatură în vecinătatea literei iniţiale. De aceea, în momentul în care se înlocuieşte o literă cu o alta din alfabet, se pot considera numai caracterele din vecinătatea literei respective. Astfel, se verifică cel mult 9 caractere în loc să se verifice fiecare literă din alfabet. În schimb, pentru a face o asemenea optimizare este nevoie de o mapare a modului de organizare a tastelor într-o structură de date precum şi o metodă de extragere rapidă a

vecinilor unei anumite litere (taste). Mai mult, trebuie ținut seama că există mai multe tipuri de tastaturi, fiecare cu o anumită dispunere a literelor pe tastatură. Mai mult, în funcție de limba folosită, mai pot să apară şi alte modificări dictate de specificitățile şi diacriticele specifice limbii (de exemplu y-z pentru folosirea limbii române).

3.5 Observații referitoare la algoritmii propuşi

- De fiecare dată, la reținerea perechilor se verifică în prealabil dacă respectiva pereche nu era deja salvată;
- De câte ori se salvează o pereche de paronime, se salvează şi perechea inversă pentru a nu fi nevoie să se repete algoritmul de căutare în sens invers;
- Pentru uşurința şi viteza operațiilor de extragere, inserare şi verificare a existenței paronimelor, se recomandă utilizarea unei baze de date;
- Pentru obținerea unor complexități cât mai mici, se recomandă organizarea dicționarului sub forma unor hash-map-uri sau sub forma unor Arbori B, astfel încât complexitatea la căutarea diferitelor şiruri de caractere în dicționar să fie cât mai mică;
- La algoritmul aleator trebuie să se poată salva starea în care se află programul în momentul întreruperii, pentru a se putea relua activitatea din acel punct în cazul în care se reporneşte programul.

4. Concluzii

S-a observat că paronimele literale sunt utile pentru corectarea erorilor caracteristice persoanelor neatente şi/sau cu nivel educațional scăzut; paronimele fonetice sunt indispensabile persoanelor slab pregătite; în timp ce paronimele morfemice sunt foarte importante atât pentru persoanele slab pregătite cât mai ales pentru străini (Bolshakov şi Gelbukh, 2003).

Referințe

Bolshakov, I.A., Gelbukh, A. *Paronyms for Accelerated Correction of Semantic Errors.* International Journal on Information Theories & Applications. V. 10, N 2, p. 198-204, 2003. (www.foibg.com/ijita/vol10/ijita10-2-p13.pdf)

Bolshakov, I.A., Gelbukh, A. *Very Large Dictionary with Paradigmatic, Syntagmatic, and Paronymic Links between Entries*. International Workshop on Enhancing and Using Electronic Dictionaries at International Conference on Computational Linguistics COLING 2004, Geneva, Switzerland, pp. 54–57, August 2004. (www.aclweb.org/anthology-new/W/W04/W04-2110.pdf)

Bolshakova, E., Bolshakov, I.A., Kotlyarov, A. *Experiments in Detection and Correction of Russian Malapropisms by Means of the Web*. International Journal on Information Theories & Applications. V.12, N 2, p 141-149, 2005. (www.foibg.com/ijita/vol12/ijita12-2-p06.pdf)

Gelbukh, A., Bolshakov, I.A. *On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms*. Jesus Favela, Ernestina Menasalvas, Edgar Chávez (Eds.) Advances in Web Intelligence (AWIC-2004, 2nd International Atlantic Web Intelligence Conference, May 16–19, 2004, Cancun, Mexico). Lecture Notes in Artificial Intelligence (indexed by SCIE), N 3034, Springer-Verlag, ISSN 0302-9743, ISBN 3-540-22009-7, pp. 105–114, 2004. (www.gelbukh.com/CV/Publications/2004/AWIC-2004-**Paronyms**.pdf)

Grădinaru A., *Atracţia paronimică în limba franceză*, Teză de doctor în filologie, Universitatea De Stat Din Moldova, Facultatea De Limbi Şi Literaturi Străine, Chişinău, 2007. (http://www.cnaa.acad.md/thesis/6130/)

Situl DEX Online (http://dexonline.ro/search.php?cuv=paronim)

Situl Academiei Române (http://www.academiaromana.ro/academia2002/acadeng/pag_cont03_1.htm)