Aplicații Pentru Recunoaşterea Dinamică a Limbii

Paul FOGARASSY-NESZLY
BAUM Engineering
Arad 310175, str. T.Moşoiu nr. 8
pf@baum.ro

Vasile GHERHES
Universitatea ”Politehnica” Timişoara
Timişoara 300006, Piața Victoriei nr.2
vasile.gherhes@upt.ro

REZUMAT

In acest articol sunt prezentate două aplicații care implementează două metode diferite de recunoaştere automată a limbii, în scopul sintezei vocale. Sunt descrişi algoritmii implementați, condițiile specifice de funcționare a aplicațiilor de acest gen şi sunt prezentate interfețele aplicațiilor.

În timp ce identificarea limbii prin intervalul Unicode al textului consumă cel mai puțin resursele sistemului şi este foarte rapid, algoritmii de recunoaştere prin analiza statistică a textului consumă mai multe resurse şi sunt ceva mai lenți. Găsirea unui compromis optim cade atât în sarcina dezvoltatorului, cât şi in sarcina utilizatorului care trebuie să decidă în faza de configurare a aplicației care sunt condițiile concrete de lucru.

În ultima parte a lucrării sunt prezentate o serie de considerente privind inerția necesară la schimbarea limbii atunci când aplicația este utilizată în contextul sintezei vocale diferențiate lingvistic a textului.

Cuvinte cheie

Recunoaşterea limbii, sinteză vocală, accesibilitate, tehnologii asistive.

Clasificare ACM

H5.2. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCERE

Spre deosebire de recunoaşterea automată a limbii (RAL) în care este scris un text, recunoaşterea dinamică a limbii (RDL) presupune implicarea unor algoritmi diferiți, capabili să identifice limba pe baza unui flux continuu de date. Această restricție majoră de abordare a problemei este impusă de aplicațiile de tip sinteză vocală (Text To speech - TTS).

O altă aplicație a RDL are în vedere sinteza vocală diferențiată lingvistic la traducerea automată a unui text, bazată pe recunoaşterea vocală [1,2]

Sinteza vocală este folosită mai ales de către persoanele cu deficiente de vedere, sau dificultăți de citire (dislexici sau analfabeți) pentru accesibilizarea documentelor în format electronic. Atunci când există posibilitatea ca texte scrise în limbi diferite să alterneze, este necesar un algoritm capabil să discrimineze limba în care este scris fragmentul de text, deoarece sinteza vocală se bazează pe particularitățile fonetice ale limbii în care este scris textul, precum şi pe regulile de scriere specifice limbii respective. Aşadar, este obligatorie folosirea sintezei vocală corespunzătoare limbii, pentru ca rezultatul să fie comprehensibil.

Algoritmii utilizați la RDL se bazează pe identificarea intervalului de codificare a caracterelor (character encoding detection) [3] şi/sau pe modele statistice [4,5].

APLICAȚIE PENTRU RECUNOAŞTEREA AUTOMATĂ A LIMBII PRIN METODA INTERVALULUI DE CODIFICARE A CARACTERELOR

Una dintre metodele pentru identificarea limbii în care este scris un text (în mod dinamic sau nu) este prin identificarea codificării caracterelor (character encoding detection) pentru textul analizat [3]. Unicode este un format definit de către Unicode Consortium pentru codarea, stocarea şi interpretarea textelor în mediul informatic; acesta este standardul de codificare de facto utilizat la interpretarea datelor binare în format text [6].

Aplicația permite definirea limbii pentru intervalele Unicode stabilite, ceea ce permite schimbarea dinamică a limbii, în timpul citirii textului. Dezavantajul metodei constă in faptul că se comportă haotic în cazul în care texte scurte scrise cu caractere diferite sunt prezente frecvent în interiorul aceluiaşi text. Utilizatorul experimentează într-o asemenea situație modificări rapide a vocilor cu care este citit textul, ceea ce produce de obicei un rezultat greu de înțeles.

Figura 1 prezintă interfața de configurare a aplicației; după cum se poate vedea, se pot configura vocile cu care sunt citite textele scrise cu caractere latine obişnuite, cu caractere arabice, greceşti, ebraice şi chirilice. De asemenea, se pot configura parametrii de viteză şi volum a fiecărei voci în parte, cu scopul de a armoniza vocile cu care este citit un text.

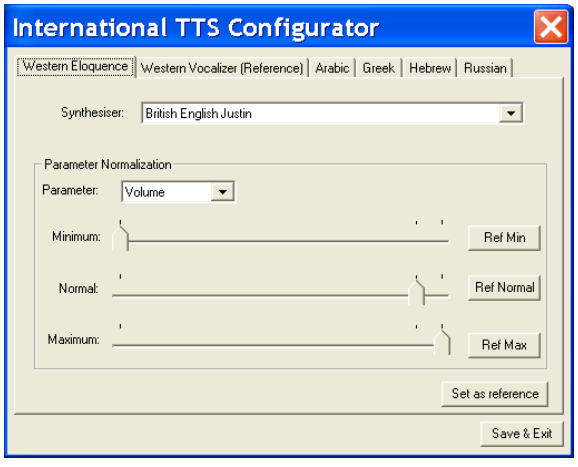

Figura 1 Interfața de configurare a aplicației

Avantajul metodei utilizate este că încarcă foarte puţin procesorul şi nu necesită memorie; dezavantajul constă că trebuie prevăzut în mod explicit o oarecare inerţie la schimbarea vocii, pentru a evita schimbarea prea frecventă a vocii în cazul textelor insuficient prelucrate, aşa cum rezultă acestea de obicei din procesul de recunoaştere optică a caracterelor (OCR - Optical character Recognition).

APLICAŢIE PENTRU RECUNOAŞTEREA LIMBII PRIN METODE STATISTICE

Cea mai avansată metodă de identificare a limbii constă în analiza n-gramelor. Aceasta este cea mai utilizată metodă statistică şi are mai multe variante, fiecare cu avantajele şi dezavantajele sale, aplicabilă pentru RAL şi RDL

O n-gramă este o sub-secvenţă de n elemente dintr-o secvenţă dată; în general secvenţa de elemente poate fi orice, de la caractere şi până la cuvinte. În analiza lingvistică n-gramele sunt utilizate mai mult pentru cuvinte sau pentru caractere. În această lucrare, prin n-gramă se înţelege o secvenţă de n-caractere succesive dintr-un text. Atunci când este vorba de două caractere (n = 2) se mai foloseşte termenul de bigramă (sau digramă), iar când este vorba de succesiuni de trei caractere (n = 3) termenul consacrat este trigramă. În Tabelul 1 este prezentat un exemplu de descompunere în n-grame a cuvântului „analiză”.

Tabelul 1 Bigramele, trigramele şi 4-gramele cuvântului „analiză”

	analiză
2-grame	_a , an , na , al , li , iz , ză, ă_
3-grame	_an , ana , nal , ali , liz , iză , ză_
4-grame	_ana , anal , nali , aliz , liză iză_

În Tabelul 3.1, începutul de cuvânt şi sfârşitul de cuvânt a fost marcat printr-un caracter special, ne-literal (_). Este relevantă din punct de vedere lingvistic această marcare, deoarece localizarea unui grup de caractere la începutul sau la sfârşitul cuvântului este semnificativă statistic pentru caracterizarea unei limbi.

Discriminarea limbilor pe baza n-gramelor (în general) pleacă de la observaţia că pentru fiecare limbă anumite n-grame apar mai frecvent decât altele. Studii experimentale realizate de [7] au arătat că utilizarea trigramelor conduce la cele mai bune rezultate. Identificarea limbii se face prin compararea frecvenţei de apariţie a trigramelor în textul analizat cu frecvenţa acestora în corpusurile limbilor care sunt avute în vedere.

În faza de „antrenare” a aplicaţiei se construieşte spectrul de frecvenţe al n-gramelor pentru fiecare limbă în parte. Acesta se bazează pe un corpus relevant pentru limba avută în vedere şi domeniul de aplicare (dacă este cazul). Din studii realizate de [8, 9] rezultă că un corpus de circa 50.000 de cuvinte oferă o precizie foarte bună care nu mai creşte semnificativ prin mărirea volumului. Frecvenţele relative ale n-gramelor reprezintă caracteristica fiecărei limbi, iar frecvenţa n-gramelor din textul analizat se realizează în timpul rulării, în acelaşi mod. În funcţie de modul de construire a acestui spectru, metoda poate fi mai rapidă sau mai lentă, mai precisă sau mai puţin precisă.

Corpusul trebuie să fie omogen din punct de vedere al limbii caracterizate de acesta şi trebuie să fie corect gramatical şi sintactic; calitatea corpusului are o influenţă hotărâtoare asupra preciziei de identificare a limbii.

Compararea spectrului analizat cu cele de referinţă se poate face în diverse feluri. Cel mai simplu criteriu este suma abaterilor absolute (ecuaţia 1) sau suma abaterilor pătratice (ecuaţia 2).

$$A_L = \sum_{i=1}^{m} \left| f_{ai} - f_{Li} \right| \qquad (1)$$

unde A_L este abaterea frecvenţelor pentru limba L, m este numărul de n-grame din textul analizat, f_{ai} este frecvenţa n-gramei i din textul analizat, iar f_{Li} este frecvenţa n-gramei i din spectrul de frecvenţe al limbii L.

$$A_L = \sum_{i=1}^{m} \left(f_{ai} - f_{Li} \right)^2 \qquad (2)$$

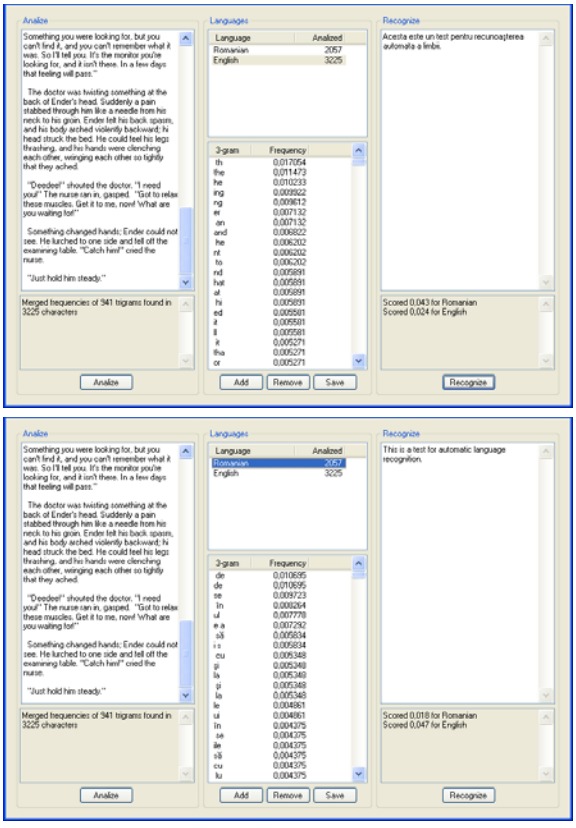

Figura 2. Interfaţa aplicaţiei de testare a algoritmilor de analiză statistică

Limba identificată cea mai probabil este limba L pentru care A_L este minim.

Criterii mai sofisticate de calcul al abaterii A_L ţin cont printr-un coeficient de pondere de probabilitatea mult mai mare sau mult mai mică (chiar zero) a unor n-grame pentru o anumită limbă. În acest caz, abaterea A_L se calculează conform relaţiei (3).

$$A_L = \sum_{i=1}^{m} k_{Li} \left(f_{ai} - f_{Li} \right)^2 \qquad (3)$$

unde k_{Li} este coeficientul de pondere pentru n-grama i din limba L; spre deosebire de ecuațiile (1) şi (2), deoarece k_{Li} poate fi pozitiv sau negativ, şi A_L poate avea valori pozitive sau negative.

Figurile 3.1 prezintă interfața aplicației de testare a algoritmilor de analiză statistică. În exemplul prezentat este evaluat cel mai simplu algoritm; sunt introduse două mini-corpusuri de numai câteva mii de caractere pentru limba română (2057 caractere) şi limba engleză (3225 caractere). În urma analizei acestora, sunt identificate toate trigramele şi se calculează frecvența acestora. Textul analizat este la rândul său descompus în trigrame, iar pentru fiecare dintre acestea se construieşte o sumă a frecvenței în cele două corpusuri. Limba în care este scris textul este probabil cea care are cea mai mare sumă caracteristică. După cum se poate vedea, diferența dintre cele două sume este semnificativă pentru ambele texte analizate.

Tabelul 2 prezintă rezultatul exemplului prezentat în condițiile din Figura 2.

Tabelul 2. Rezultatele testului preliminar

Text analizat	Limba	Scor
Acesta este un test pentru recunoaşterea automată a limbii.	**Română**	**0,043**
	Engleză	0,024
This is a test for automatic language recognition.	Română	0,018
	Engleză	**0,047**

Algoritmii care se bazează pe analiza statistică a textului în vederea recunoaşterii limbii sunt mai complecşi şi pentru a nu încărca procesorul în timpul rulării este necesară optimizarea bazelor de date obținute în faza de antrenare. De asemenea, algoritmii trebuie optimizați pentru un consum de memorie cât mai redus, în special în cazul rulării aplicației pe dispozitive mobile.

Optimizările sunt necesare deoarece în mod normal în paralel cu aplicația de recunoaştere funcționează o aplicație de sinteză vocală, aplicația client care foloseşte sinteza vocală, foarte probabil mai funcționează un cititor de ecran (cu sau fără funcție de magnificare a ecranului) şi în sfârşit, aplicația utilizatorului.

CONSIDERENTE PRIVIND COMPORTAMENTUL APLICAȚIILOR DE RECUNOAŞTERE A LIMBII

În mod intuitiv, prin recunoaşterea limbilor în care este scris un text se doreşte etichetarea tuturor fragmentelor acestuia în mod corespunzător. Acest lucru poate fi util pentru aplicații de clasificare şi analiză bibliografică.

Scopul aplicațiilor prezentate în această lucrare este de a permite sinteza vocală diferențiată a unui text scris în două sau mai multe limbi diferite, aşa cum se întâlneşte relativ frecvent în zonele multiculturale.

Teste preliminare au arătat că o etichetare excesiv de riguroasă a unui text poate duce la modificarea limbii în care este citit textul mult prea frecvent. Fiecare limbă conține împrumuturi lingvistice sau substantive proprii din alte limbi. La realizarea unei cărți audio sau la lectura prin voce sintetică, comună nevăzătorilor, dar şi dislexicilor, a citi cu o voce diferită un nume de origine străină poate fi foarte deranjant. La fel de deranjant poate fi şi modificarea vocii pentru redarea unor fragmente scurte, sau chiar a unor cuvinte disparate.

De exemplu, în limba română au pătruns în ultimele decenii (justificat sau nu) multe cuvinte şi expresii din limba engleză. Dacă azi a devenit comună prezența în textele tehnice a unor cuvinte cum sunt „mouse” sau „webpage”, în texte comune găsim frecvent cuvinte sau expresii cum sunt „weekend”, „city break”, „team building”, „fitness” etc. De asemenea, în limba română se păstrează (aproape întotdeauna) ortografia originală a toponimelor şi a numelor proprii; în română este firesc să scriem „New-York” şi nu „Nowy Jork”, cum se scrie în poloneză.

Algoritmii de analiză statistică vor identifica repede faptul că fragmentele care conțin literele „k”, „y” şi „w” sunt mult mai probabil în limba engleză, iar literele duble „ee” sau „ss” apar foarte rar în limba română, în timp ce în engleză sunt relativ frecvente. Citirea în engleză a acestor cuvinte presărate într-un text scris aproape integral în română poate fi însă foarte supărătoare.

Pentru a evita asemenea situații există mai multe posibilități: fie este introdusă o anumită inerție la schimbarea limbii, prin care se evită citirea cu voce diferită a unor cuvinte sau fragmente relativ scurte, fie se defineşte o listă de cuvinte străine frecvente în limba de bază, care sunt ignorate de aplicație şi deci vor fi citite cu vocea considerată implicită.

Pe de altă parte, aplicațiile de sinteză vocală lucrează sincron, pe baza unui flux de date (text în acest caz) şi generează cu o mică întârziere tot un flux de date (audio în cazul sintezei vocale). Datorită acestui lucru şi în funcție de decalajul temporar admis între cele două fluxuri, este aproape imposibil ca aplicația de recunoaştere să reacționeze imediat şi să schimbe corespunzător vocea. Inevitabil, la schimbarea limbii va exista cel puțin un fragment care va fi citit cu vocea anterioară. În cazul unor cuvinte izolate sau a unor fragmente scurte, acest lucru poate duce la rezultate greu de înțeles pentru utilizator.

Inerția la modificarea vocii reprezintă cea mai bună metodă de evitare a unor asemenea situații şi este perfect admisibilă în cazul sistemelor care funcționează sincron.

Desigur, în cazul aplicațiilor de conversie text-audio, utilizate la realizarea cărților audio sau a cărților Daisy, textul poate fi parcurs în întregime şi etichetat corespunzător înainte de conversia audio propriu-zisă. În acest gen de aplicații întârzierile mari între fluxul text de la intrare şi fluxul audio rezultat sunt acceptabile.

Soluția cea mai bună pentru managementul erorilor de conversie admisibile datorită inerției se rezolvă în mai multe feluri. Pe lângă inerția, menționată anterior, s-au elaborat algoritmi care păstrează un fragment de text sintetizat anterior ca şi referință. În cazul în care limba unui fragment de text este discriminată cu incertitudine (definită statistic în faza de configurare), atunci limba de referință este cea din fragmentul anterior. O altă opțiune de configurare permite definirea unei limbi implicite, atunci când fragmentele scrise într-o limbă diferită sunt puțin frecvente.

Din testele preliminare, a rezultat că algoritmii de recunoaştere a limbii sincroni se comportă mult mai bine în cazul în care sunt definite doar două limbi posibile. În cazul în care un text poate fi scris în trei limbi, acest gen de algoritmi necesită configurări specifice suplimentare pentru ca rezultatul să rămână foarte bun.

CONCLUZII ŞI DIRECŢII DE CERCETARE

Această lucrare prezintă două aplicații generice de RAL aplicabile şi pentru RDL. Sunt prezentate particularitățile algoritmilor în timpul rulării împreună cu cititoarele de ecran, pe dispozitive mobile, relativ sărace în resurse hardware (memorie şi procesor), precum şi faptul că textul care trebuie analizat este scurt.

Analiza performanţelor algoritmilor implementaţi, precum şi propunerile de îmbunătăţire urmează să fie identificate în timpul activităţii de testare în diverse condiţii, cum sunt de exemplu texte scurte dar care conţin totuşi cuvinte din limbi diferite. Un amplu program experimental semi-automatizat urmează să valideze algoritmii care vor fi propuşi de analişti. De asemenea, tot experimental, se vor identifica pragurile optime de incertitudine.

CONFIRMARE

Această lucrare a fost elaborată în cadrul contractului 29DPST/13.09.2013, „*Aplicaţie pentru Conversia din Text în Voce Sintetică cu Recunoaşterea Automată a Limbii*”, în Programului Inovare, Dezvoltare Sisteme-Produse-Tehnologii a UEFISCDI.

REFERINŢE

1. Mathias, L. (2007). Statistical Machine Translation and Automatic Speech Recognition under Uncertainty. PhD thesis, Johns Hopkins University
2. Moore, R. C. and Quirk, C. (2008). Random restarts in minimum error rate training for statistical machine translation. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 585–592, Manchester, UK. Coling 2008 Organizing Committee
3. Timothy Baldwin and Marco Lui, Language Identification: The Long and the Short of the Matter, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 229–237, Los Angeles, California, June 2010
4. Gavin Churcher, „Distinctive character sequences”, 1994, personal communication
5. Dunning, T. "Statistical Identification of Language". Technical Report MCCS 94-273, New Mexico State University, 1994
6. The Unicode Standard, Version 5.0, Fifth Edition, The Unicode Consortium, Addison-Wesley Professional, 27 October 2006
7. Cavnar, W., and Trenkle, J. (1994). N-gram-based text categorization. Proc. 3rd Symp. on Document Analysis and Information Retrieval (SDAIR-94)
8. Dunning, T. (1994) Statistical Identification of Language. Technical Report MCCS 94-273, New Mexico State University
9. Ljubeši´c, N., Mikeli´c, N., and Boras, D. (2007) Language identification: How to distinguish similar languages. In Lužar-Stifter, V. and Hljuz Dobri´c, V., editors, Proceedings of the 29th International Conference on Information Technology Interfaces, pages 541–546, Zagreb SRCE University Computing Centre