V. Posea, G.S. Cojocar (eds.), RoCHI 2012

Sistem de Sinteză

Text – Vorbire pentru Limba Română

Teodor Paul Alin Becan
Universitatea Tehnică Cluj-Napoca
Str. Memorandumului nr. 28, 400114,
Cluj-Napoca
Becan.Paul@it-dynamics.ro

Dorian Gorgan
Universitatea Tehnică Cluj-Napoca
Str. Memorandumului nr. 28, 400114,
Cluj-Napoca
dorian.gorgan@cs.utcluj.ro

REZUMAT

Această lucrare îşi propune să prezinte realizarea şi implementarea unui sistem de sinteză a textului în limba română pentru producerea sunetelor corespondente printr-un analizator de text şi prelucrarea unor sunete pre-înregistrate.

Se prezintă abordarea utilizată pentru căutarea componentelor de sunet, care corespund pentru o intrare de tip text, dintr-o bază de date acustice predefinite/pre-înregistrate. Am integrat acest modul de căutare în componenta de Analizator Text-Sunet a unui sistem *Text-to-Speech(TTS)* pentru limba română. Normele de căutare sunt construite pe marginea contextului din textul original, din punct de vedere fonetic şi posibilitatea aplicării lui la un context similar al înregistrărilor din baza de date.

De la început am pornit cu premisa că rezultatele care ţin cont de context, vor avea o calitate superioară. Rezultate experimentale sunt prezentate de asemenea în conţinutul acestei lucrări pentru a contura cât mai bine finalitatea proiectului.

Cuvinte cheie

Sinteză text-vorbire, algoritmi căutare, context.

Clasificare ACM

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

INTRODUCERE

Sistemele cu interfeţe care dispun de limba vorbită joacă în prezent un rol important în interacţiunea Om-Calculator, devenind o necesitate pentru cele mai noi aplicaţii practice şi servicii cerute de modernitatea zilelor în care trăim. În mod special, caracteristica interfaţă-voce este deja în mare măsură necesară pentru a extinde potenţialul aplicaţiilor comerciale, adăugând flexibilitate, viteză şi naturaleţe la interfeţele existente.

Generarea de voce sintetică, care imită vorbirea umană, dintr-un text aleator, nu este o sarcină banală, deoarece acest lucru necesită cunoştinţe complexe cu privire la lumea reală, limba în care se face sinteza, contextul din care textul face parte, o înţelegere profundă a semanticii conţinutului de text şi a relaţiilor care stau la baza închegării acestor informaţii în vorbire fluentă.

Scopul nostru a fost acela de a dezvolta un sistem eficient, bazat pe algoritmi de căutare şi/sau tehnici care să fie capabile să selecteze unităţile cele mai potrivite care să fie mai apoi concatenate şi editate în sunetul final.

Analiza fonetică a textului iniţial ne generează unităţile de căutare ce urmează să fie utilizate. Aceasta componentă converteşte caracterele lexico ortografice într-o reprezentare fonetică, împreună cu unele informaţii suplimentare care pot oferi detalii semnificative precum accentul. Analizatorul fonetic este o componentă crucială dacă obiectivul este realizarea unor rezultate de foarte bună calitate, acoperind inflexiunile şi complexitatea limbii vorbite.

Spaţiul de căutare – Baza de date acustice – este compus din cuvinte simple şi fraze aleatorii. Pe aceasta s-a aplicat un analizator asemănător celui ortografic pentru a putea stabili unele taguri care vor putea mai apoi să identifice unitatea de sunet corespunzătoare.

O fonemă este unitatea de sunet fundamentală percepută. Este o unitate abstractă ce reprezintă fiecare sunet al unei limbi. Prin modificarea unui fonem al unui cuvânt, se generează fie un cuvânt inexistent dar perceput ca diferit de către vorbitorii limbii, fie un cuvânt cu alt sens. Fonemele nu sunt sunetele ca atare, ci perceperea lor la nivel mental. Unui fonem îi pot corespunde mai multe sunete fizice, pe care vorbitorii unei limbi date le percep ca fiind unul şi acelaşi sunet.

De exemplu, fonemul /h/ din cuvintele româneşti "har" şi "hidră" este perceput ca fiind identic, deşi în realitate în cele două cazuri el se articulează în locuri complet diferite (laringe, respectiv palatul gurii).

Abordarea concatenativă a acestei implementări, impune din start un fel de transcriere fonetică a bazelor de date şi căutare chiar şi în cazul în care se foloseşte o bază de date cu multe unităţi de sunet. Dacă un fragment de text nu este corelat cu nimic existent deja in baza de înregistrări, singura opţiune care rămâne este ca acel sunet să fie contruit din componentele care îl compun.

Astfel, în aplicaţie se determină trei tipuri de unităţi : foneme, cuvinte şi propoziţii. Acestea au fost folosite pentru alcătuirea sunetului final.

Pentru a putea crea unităţile în cauză, un algoritm special a fost necesar, astfel încât, acolo unde e cazul şi nu se

identifică unități de sunet corespondente să reușească compunerea chiar și a unor propuziții complete doar folosind fonema/sunetul de bază (litera – pentru o vizualizare mai clară) a fiecărui cuvânt constituent al bazei de sunete de care dispunem.

În ceea ce urmează voi încerca să prezint felul în care mediul a fost pregătit și algoritmii care s-au folosit.

Structura documentului de prezentare este după cum urmează. Secțiunea urmatoare prezintă o vedere de ansamblu a sistemului TTS şi motivează demersul abordat. Apoi în secțiunea următoare se intră în detalii ale algoritmilor de căutare, se descriu regulile şi modelele utilizate pentru excepţii. Voi continua cu secțiunea de prezentare a evaluărilor implementării, în timp ce ultima secţiune încheie lucrarea cu concluziile personale, precum și unele propuneri de îmbunătățiri ale aplicației.

ALTE REALIZĂRI

În domeniul sintetizării de voce pentru limba română nu există foarte multe implementări din păcate, fiind și una din limbile cu o gramatică complexă. Startul acestui proiect a fost pentru mine contactul cu o aplicație similară realizată de compania iQuest Technologies Cluj Napoca, unde un sistem similar a fost integrat cu o centrală telefonică și este folosit la Curtea de Apel pentru a reproduce sentințele telefonic. Aplicația se axează mai mult pe aspecte de integrare, astfel ca aș putea spune ca ceea ce am dezvoltat eu are rezultate de calitate comparabilă. Un alt aspect pe care îl consider util este dimensiunea aplicației. Aceasta este aplicabilă pe orice bază de înregistrări, iar atata timp cât acoperă cel puțin toate sunetele limbii române va produce rezultate. Se poate deci ușor integra și pe echipamente mobile. O versiune incipientă a prezentei aplicații am prezentat și în cadrul Sesiunii Științifice Studențești realizate de către UTCN și UBB.

GENERALITĂȚI ALE APLICAȚIEI

Sistemul implementat folosește o abordare concatenativă, metodă întâlnită și în cazul altor aplicații similare, de altfel este și metoda promovată de Huang [2].

În prima fază, textul brut va trece printr-un analizator de text (Procesare Text – Figura 1) care va determina unitățile din structura de intrare, va face normalizarea textului, analiza fonetică și împărțirea în silabe.

Pe baza informațiilor oferite de analiza anterioara, se trece la etapa următoare, Analizatorul Text-Sunet , va căuta să găsească corespondențe între unități cât mai mari între transcrierea fonetică a unităților de sunet și a textul inițial. În cazul în care intreaga unitate introdusă nu va avea un corespondent direct, pe baza analizelor efectuate și a informațiilor stocate despre text, se va incerca compunerea din unități subdivizare. Împreună cu informații de pronunțare, unitățile selectate sunt procesare și astfel în cadrul modulului de Procesare Sunet se crează vocea finală. Figura 1 prezintă arhitectura simplificată a sistemului. Aplicația se bazează pe selectarea de unități de sunet, asemeni altor implementări folosite si pentru alte sintetizatoare pentru alte limbi [7] unde principiul e acela de a avea calitatea maximă a rezultatului.

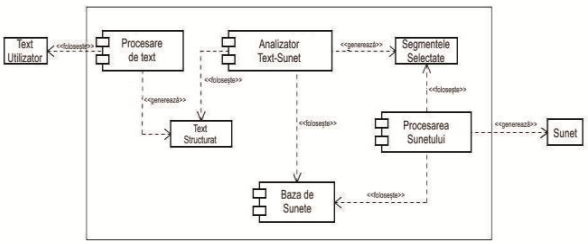

Figura 1. Arhitectura Sistemului

În această lucrare vom pune accentul pe partea prelucrării informațiilor obținute în urma analizelor fonetice cu scopul obținerii de corespondențe acustice. Se va încerca detalierea metodelor de căutare dar și exemplificarea datelor din bazele de date.

Întrucât avem la dispoziție aplicații numeroase care ajută la etichetarea unităților acustice în mod automat, implementarea unui astfel de sistem de sinteză vocală e mult mai probabil. Metoda se bazează pe înregistrări efectuate folosind o vorbire naturală [6]. Astfel se poate acoperi o arie cât mai variată al domeniului de proveniență pentru textul de intrare. Multe alte implementări folosesc metode diferite de construcție a sunetului și astfel ajung să necesite ore întregi de sunet înregistrat. Metoda aleasă, bazată pe selecția unităților, folosindu-se de bucăți de dimensiune variabilă, începând de la foneme pana la cuvinte sau chiar propoziții întregi, se bazează pe o înregistrare de aproximativ 300 cuvinte, singulare și în cadrul unor construcții complexe. Cu toate astea, rezultatul este unul inteligibil și de o calitate semnificativă.

O altă caracteristică importantă a sistemului implementat este că pentru cazul în care se schimbă înregistrarea de bază, aplicația funcționează la fel, fără nici un fel de customizări. Astfel, se pot ușor implementa voci diferite, ton masculin/feminin, înregistrări de dimensiuni mici sau chiar dimensiuni semnificative. De asemenea, metoda aleasă oferă un număr mai mare de posibilități de a alege un corespondent pentru aceeași unitate, provenind din diferite instanțe (contexte/înregistrări).

ANALIZATORUL TEXT-SUNET

Sintetizatorul Text-Voce bazat pe abordarea concatenativă grupează segmente de sunet pre-înregistrat în scopul obținerii celor mai bune rezultate de vorbire sintetizată.

O parte însemnată a acestei aplicații, care face posibilă aplicarea de corespondențe ulterioare este transcrierea fonetică. Acestea sunt rezultatul unor procese de muncă asiduă și cercetare, care au avut loc anterior implementării efective.

Formatul de intrare al modulului de analiză este un fișier de tip XML, unde fiecare nod reprezintă o unitate componentă. Fișierul este rezultat în urma aplicării modulului de Procesare Text. Întrucât sunt module separare, în orice moment se pot verifica formele în care datele se află. De asemenea se pot salva/încărca datele din iterații anterioare.

S-au folosit delimitatoare speciale pentru a marca fiecare fonemă,silabă sau cuvânt. Acestea se pot vizualiza și în figurile 2 și 3.

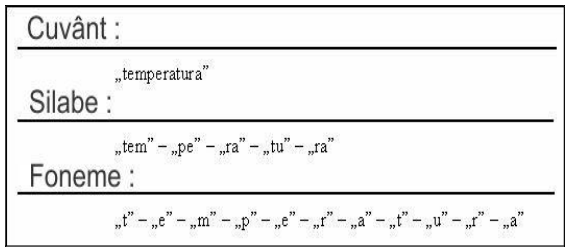

Figura 2. Compoziția sintactică a cuvântului ”temperatura”

De exemplu, fiecare cuvânt începe cu eticheta ”TwordBegin” aplicată fenemei de început și se încheie cu ”TwordEnd”. În același fel, în cazul silabelor, se vor folosi etichetele ”TSyllableBegin” și ”TSyllableEnd”.

```
<phoneme name="t">
        <syntax>
        TAffirmative,TWordBegin,TSyllableBegin
        </syntax>
</phoneme>

<phoneme name="e">
        <syntax>
        TAffirmative
        </syntax>
</phoneme>

<phoneme name="m">
        <syntax>
        TAffirmative,TSyllableEnd
        </syntax>
</phoneme>

<phoneme name="p">
                <syntax>
        TAffirmative,TSyllableBegin
        </syntax>
</phoneme>

<phoneme name="e">
        <syntax>
        TAffirmative,TSyllableEnd
        </syntax>
</phoneme>
```

Figura 3. Silabele ”tem” și ”pe” – intrare TTS

Folosind informațiile din fișierul rezultat, am considerat o structură de tip arborescent, ușurând astfel și aplicarea algoritmilor ce urmează a fi aplicați. Cunoscând etichetele de delimitare, fiecare cuvânt va fi nodul părinte, având asociate ca și noduri copil de grad I silabele componente și noduri copil de grad II fonemele din care cuvântul este compus (a se vedea Figura 4).

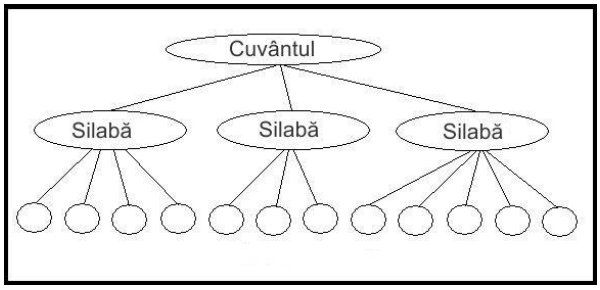

Figura 4. Forma structurilor de date

Pentru a ușura procesul de căutare, o formă echivalentă s-a construit și pentru baza acustică. Problema care apare e că baza audio e construită din fișiere de tip sunet, iar pentru adnotare este nevoie de etichete text. Structura echivalentă de tip arbore trebuie să conțină tot trimiteri text.

Soluția este la îndemână, deoarece există programe soft care permit creerea unei structuri similare pentru baza re date audio. S-a folosit aplicația Praat, care permite prin folosirea unor utilitare de adnotarea pentru etichetarea înregistrărilor conform cerințelor aplicației noastre.

Tabelul 1. Transcripție corespondente Praat

Grafema	Fonema 1	Fonema 2	Fonema 3
a	a		
˘a	˘a		
ˆa	ˆı		
b	b		
c	k		
d	d		
f	f		
g	g		
h	h		
ˆı	ˆı		
j	j		
l	l		
m	m		
n	n		
p	p		
q	k		
r	r		
s	s		
s¸	s¸		
t	t		
t¸	t¸		
v	v		
w	v		
x	ks		
z	z		
e	e	˘e	
i	i	˘i	[*i*]
o	o	˘o	
u	u	˘u	
y	i	˘i	

Aspectul contextual al căutărilor a fost ilustrat prin implementarea unor tipuri variate de algoritmi. Căutarea care nu ține cont de context are avantajul de a fi foarte rapidă, întrucât nu implică etape suplimentare de condiționare a rezultatului.

Aplicația poate folosi la un moment dat un număr nedeterminat de înregistrări, astfel că s-a ivit necesitatea adăugarii unor noduri care să conțină informații despre calea fișierului sursă iar pentru că în cazul fonemelor pot fi sute sau chiar mii de duplicate, un nod trebuie să conțină un identificator unic. Alte informații conținute indică date precum frecvența sau amplitudinea ne oferă date de prozodie. Bineînțeles că vom avea nevoie de anumite etichete temporale care să delimiteze pozițiile de start și sfârșit ale respectivei unități de sunet în cadrul înregistrării originale. Fișierele în forma lor brută conțin doar noduri de tip fonemă. Structura arborescentă, silabele și cuvintele sunt contruite doar în cadrul aplicației.

```
<phoneme name="r">
    <uid value="612"/>
    <syntax>
        DBSyllableBegin
    </syntax>
    <filename>Resources\\Sound\\file1.wav</filename>
    <position start="108.759" end="108.793"/>
    <F0values>
        <F0Window start="0.00430853" end="0.0168705"
frequency="79"/>
        <F0Window start="0.0168705" end="0.0283867"
frequency="86"/>
    </F0values>
    <pitchvalues>
        <pitch_window start="0" end="0.0105895"
mark="0.00430853"
         amplitude="-2147483648"/>
        <pitch_window start="0.0105895" end="0.0226286"
mark="0.0168705"
        amplitude="-2147483648"/>
        <pitch_window start="0.0226286" end="0.0343057"
mark="0.0283867"
        amplitude="-2147483648"/>
    </pitchvalues>
</phoneme>
```

Figura 5. Adnotarea bazei de date acustice

Cunoscând toate aceste informații despre fiecare fonemă, putem obține informații despre fiecare unitate compusă, cum ar fi silabele și cuvintele.

ALGORITMI

Structurile de date fiind definite, următorul pas e detalierea implementarii metodelor de căutare.

Algoritmi fără context

Structura gândită ne permite sa implementăm diferite feluri de algoritmi pentru a obține, în funcție de nevoi, rezultate cu timpi foarte scurți sau rezultate cu un grad de inteligibilitate mai ridicat.

Astfel, folosindu-ne de structura arborescenta, prima unitate care este căutată va fi cuvântul întreg. Fără nici o alta constrângere, orice cuvânt care din punct de vedere fonetic, corespunde criteriilor impuse de transcrierea fonetică a textului va fi luat in considerare. Dacă nu se găsește cuvântul întreg, se va căuta următoarea componenta, deci căutăm silabele cuvântului. Dacă una sau mai multe din silabele componente nu se găsesc, se va trece la căutarea pe foneme.

Căutarea se face în cel mai simplu mod, astfel că orice potrivire este luată în considerare. Dacă un rezultat pozitiv apare, se ia în considerare și căutarea pe acea ramură se întrerupe. Din moment ce nu se consideră sub nici o formă contextul din care un cuvânt, o silabă sau o fonemă provine, timpul de obținere al unui rezultat este foarte scurt. Implementarea metodei de căutare în afara contextului a fost aplicată pe două direcții, două implementări diferite.

În primul dintre cazuri, cel mai simplu si rapid posibil, se ia unitatea de text și se face o căutare în baza acustică, urmărind prima potrivire. Dacă un cuvânt nu este găsit, ne oprim și atunci începem să căutam prima silabă componentă și/sau fonemă. Singura condiție fiind o corespondență minimă și potrivire fonetică între cele două unități, căutarea se finalizează foarte repede și rezultatul este produs imediat prin concatenarea unităților găsite.

O aplicabilitate foarte potrivită pentru acest tip de căutare este în cazul sintetizării de numere și un exemplu ar putea fi generatoarele de numere aleatoare care pot fi comunicate verbal unui utilizator în vederea confirmării unei înregistrări pe un anume site de specialitate pentru a se preveni potențiali viruși sau agenți virtuali rău intenționați.

O altă abordare pentru același tip de căutare este concentrat pe căutarea și găsirea tuturor rezultatelor posibile din înregistrarile utilizate și alegerea la întâmplare a uneia dintre rezultate. Întrucât este vorba de un algoritm fără context, alegerea este facută folosind un generator de număr aleator din rezultatele posibile. Deși nu are un nivel de performanță masurabil la nivel concret, în mod paradoxal, nivelul de inteligibilitate și naturalețe în cazul din urmă a crescut în medie cu pâna la 70%.

Algoritmi contextuali

O abordare diferită este necesară dacă rezultatul se vrea a fi unul care oferă o claritate a vorbirii mult mai apropiată de cea uzuală, umană, mult mai inteligibilă și naturală. Asta pentru că folosind metodele enumerate anterior, aspectul de artificialitate al vocii este greu de trecut cu vederea. Vocea are un sunet distinct, robotic.

Ideea de la care se pornește este aceea de a încerca să plasăm textul pentru care se face căutarea în același context din care a fost luat. Cantitatea detaliilor care se încorporează când se face transcrierea este suficientă pentru a ne putea oferi posibilitatea de a spune ceva din punct de vedere al contextului. În acest sens am ales să abordăm o serie algoritmi contextuali, relevanți pentru aplicația dezvoltată.

În vorbirea de zi cu zi, comunicarea este deosebit de complexă și se bazează pe mult mai multe aspecte. Nu

putem spune ca vorbirea este formată doar din unități independente și stătătoare. La cel mai de bază nivel, fonemele unui cuvânt se contopesc unele cu altele. Oricât am încerca să delimităm unitățile fundamentale, fonemele, datorită fluctuațiilor vocii din timpul vorbirii, acestea vor împrumuta din sunetul fonemei vecine. Este usor de dedus faptul că aceste sunete care provin mai mult sau mai puțin dintr-o unitate în care sunetele alăturate sunt aceleași, rezultatul se va apropia în mod cert mai aproape de calitatea urmărită.

O soluție oarecum evidentă este că trebuie să facem în așa fel încât informații despre fonemele alăturate să fie stocate sub o formă în aplicația noastră și să folosim acest lucru în căutările ulterioare. Exact acest lucru l-am făcut în prima implementare contextuală.

Se caută unitatea de sunet corespondentă iar când se găsește, se aplică un filtru care verifică dacă acel context al textului inițial se găsește și în cazul unității de sunet corespondente. Dacă o potrivire nu este găsită, atunci unitatea de căutare este divizată și se începe căutarea pe segmente în aceleași condiții contextuale. Rezultatele obținute astfel sunt surprinzător de asemănătoare vorbirii uname naturale. Desigur, dacă se lucrează pe înregistrări limitate ca diversitate, putem întâlni cazuri în care o potrivire nu este întotdeauna găsită și atunci unele reguli pentru aceste excepții trebuiesc implementate. Algoritmul e axat pe găsirea unei potriviri exacte de context. Dacă un rezultat respectând aceste condiții nu este găsit, atunci se aplică un filtru care incearcă în primul rând să găsească o potrivire de stânga sau dreapta. Dacă însă se întâmplă ca nici în aceste cazuri să nu se găsească o soluție, deși probabilitatea este foarte mică, se va folosi ramura de căutare fără context.

O altă implementare urmărește îmbunătățirea suplimentară a rezultatelor și folosește un număr suplimentar de filtre care verifică atribute de prozodie precum durata, frecvența sau accentul (vezi Figura 6). În cazul în care o potrivire este găsită atunci rezultatul este semnificativ îmbunătățit.

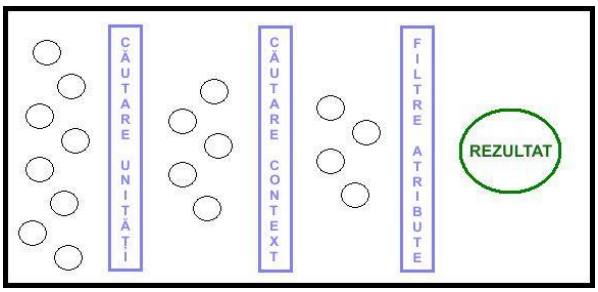

Figura 2. Căutare contextuală, cu filtru de atribute

REZULTATE EXPERIMENTALE

În timpul dezvoltării și după finalizare mai multe teste au fost necesare pentru a putea stabili calitatea rezultatelor obținute folosind fiecare din metodele implementate. Ca și etalon pentru stabilirea calității rezultatelor am folosit nivelul de inteligibilitate și cel de naturalețe, întelegerea textului și fluiditatea vorbirii.

Întrucât dezvoltarea a fost pe 4 direcții diferite, pentru fiecare s-a considerat un test de caz separat, iar ca și date de intrare, secvențe de text care au însumat peste 1000 cuvinte pentru a putea acoperi cât mai multe demenii.

Folosind datele obținute am reușit să îmbunătățim suplimentar metodele dezolvate și de asemenea, s-au aplicat îmbunătățiri interfetei aplicației pentru a ușura interacțiunea, conform *feedback-urilor* primite.

A fost evident încă de la început că algoritmii contextuali consumă semnificativ mai multe resurse și au un timp de computare mai mare. Rezultatul este deci obținut cu un cost mai mare, dar dacă ne raportăm la etalonul ales ca reper, costul este justificat. Calitatea sunetului este la un nivel superior, atât din punct de vedere al inteligibilității cât și al naturaleții.

Folosirea algoritmilor fără context este potrivită ăentru situațiile în care un rezultat imediat este necesar, putând face abstracție de calitatea acestuia. Rezultatele testelor au arătat faptul că pentru acest tip de algoritmi computarea este aproape imediată, bineînțeles depinzând de lungimea textelor. Sunetul rezultat are un aspect robotic în majoritatea cazurilor dar este totuși inteligibil.

O medie a rezultatelor obținute poate fi vizualizată în Tabelul 2.

Tabelul 2. Media performanțelor rezultatelor experimentale (cuvinte generale)

Algorim	Medie inteligibilitate vorbire	Medie fluență vorbire	Medie Timpi computare
1.Primul Găsit	60 %	55 %	0.1-5 sec
2.Alegere Aleatoare	65 %	70 %	1-7 sec
3.Căutare Contextuală	95 %	89 %	5-15+ sec
4.Aplicare Filtre Proz.	98 %	95 %	9+ sec

E de notat faptul că se pot obține rezultate foarte rapide în detrimentul calității acestora și vice-versa. Acestea sunt cele 2 valori care în contextul aplicației sunt învers proporționale. Calitatea vorbirii provine semnificativ și din numărul unităților de dimensiuni mari găsite. Acest lucru se poate obține folosind înregistrări mai multe și bogate în sunete dintre cele mai diferite.

Însă dacă situația o cere și este necesară limitarea, în special datorită dimensiunilor, algoritmi care să obțină rezultate de o calitate foarte bună chiar formând fiecare unitate de sunet din segmente componente care sunt supuse unor filtre succesive pentru sporirea exactității se pot implementa și folosi cu succes.

Un set de teste similar s-a efectuat folosind cuvinte/unități de sunet identificate ca făcând parte din înregistrarea inițială. Întrucât a fost sigură folosirea unor unități de sunet mai mari, rezultatele au fost bineînțeles mult mai bune după cum se poate vedea și în Tabelul 3.

Tabelul 2. Media performanțelor rezultatelor experimentale (cuvinte alese)

Algorim	Medie inteligibilitate vorbire	Medie fluență vorbire	Medie Timpi computare
1.Primul Găsit	80 %	75 %	0.1-5 sec
2.Alegere Aleatoare	90 %	80 %	1-7 sec
3.Căutare Contextuală	100 %	89 %	5-15+ sec
4.Aplicare Filtre Proz.	100 %	100 %	9+ sec

CONCLUZII

Lucrarea curentă prezintă o implementare a unui sistem de sinteză text-vorbire dezvoltat pentru limba română, folosind o abordare concatenativă, cu selecția de unități și abordând două implementări diferite, cu și fără context.

Avântul tehnologic ar fi putut ușor crea falsa impresie că sinteza vorbirii este o problemă rezolvată, mai exact, că o voce sintetizată ar putea înlocui vorbirea umană. În nici un caz nu s-a ajuns la un asemenea nivel, toate aspectele vorbirii neputând fi încă cuprinse într-o aplicație de vorbire artificială.

Ceea ce se reușește însă, folosind abordări de tip selecție a unității, este obținerea unor rezultate surprinzător de bune atât ca și inteligibilitate cât și la nivel de fluiditate a vorbirii [10].

Orice sintetizator de vorbire bazat pe acest tip de sintetizare, are la bază o bază acustică de calitate superioară, de obicei într-un mediu controlat, iar acest lucru aduce un plus în materie de calitate.

Folosind metode precum cele prezentate în această lucrare e posibil să se obțină vorbire sintetizată de o calitate surprinzător de bună, foarte inteligibilă și o naturalețe rezonabilă.

REFERINŢE

[1] Romanian Academy, Iorgu Iordan -Al. Rosetti Lingvistic Insti-tute, Romanian Language Orthographic, Orthoepic and Morpho-logical Dictionary, 2nd Edition. Univers Enciclopedic Bucharest, 2005.

[2] X. Huang, A. Acero, and H. Hsiao-Wuen, Spoken Language Pr-cessing: A Guide to Theory, Algorithms, and System Develop-ment. Prentice Hall, 2001.

[3] T. Dutoit, An Introduction to Text-to-Speech Synthesis. Kluwer Academic Publishers, 1997.

[4] M. Ostendorf, I. Bulyko, "The Impact of Speech Recognition on Speech Synthesis," Keynote Paper in: Proceedings IEEE 2002 Workshop on Speech Synthesis, Santa Monica, Sept. 11-13, 2002.

[5] A. Hunt, A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," Proc. ICASSP, vol. 1, pp. 373-376, 1996.

[6] A. D. Conkie, "Robust Unit Selection System for Speech Synthesis," in: Joint Meeting of ASA, EAA, and DAGA, paper 1PSCB_10, Berlin, Germany, 15-19 Mar., 1999

[7] A. J. Hunt and A. W.Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in ICASSP ’96: IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing. IEEE Computer Society, 1996, pp. 373–376.

[8] J.de Guznam, H. Kaiser, and D. Nuffer, Spirit. http://spirit.sourceforge.net, consulted on 18 March 2009.

[9] M. A. Ordean, A. Saupe, L. Teodorescu, M. Boldizsar, M. Ordean, and G. C. Silaghi, “Efficient parsing of romanian language for text-to-speech purposes,” in submitted to 12th Intl. Conf. on Text, Speech and Dialogue (TSD2009), Pilsen, Czech Republic, 2009.

[10] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Möbius, B. Säuberlich, "Restricted Unlimited Domain Synthesis," in: Proc. Eurospeech 2003, Geneva, 1321-1324, Sept. 1-4, 2003