

Componentele unui sistem de sinteză text-vorbire

Mihai Alexandru Ordean
iQuest Technologies
Cluj-Napoca, România
Mihai.Ordean@iquestint.com

Andrei Şaupe
iQuest Technologies
Cluj-Napoca, România
Andrei.Saupe@iquestint.com

Mihaela Ordean
iQuest Technologies
Cluj-Napoca, România
Mihaela.Ordean@iquestint.com

Dorian Gorgan
Universitatea Tehnica Cluj-
Napoca, România
Dorian.Gorgan@cs.utcluj.ro

REZUMAT

În acest articol vom descrie componentele unui sistem de sinteză text-vorbire (TTS) pentru limba română. Se prezintă arhitectura generală a sistemului TTS după care sunt detaliate componentele acestuia și despărțirea în silabe.

Cuvinte cheie

Sinteza text-vorbire, procesare de text, despărțirea în silabe.

Clasificare ACM

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

INTRODUCERE

Regulile de procesare a textului din cadrul sistemului TTS sunt conforme cu cele din DOOM II [1]. În prima etapă din sinteza text-vorbire se convertesc simbolurile ortografice lexicale într-o reprezentare fonemică, eventual utilizând informații diacritice, precum plasarea accentului [2]. Analiza fonetică este obligatorie într-un sistem de sinteză text-vorbire dacă obiectivele se axează pe obținerea unei calități ridicate pentru sunetul final, care să acopere toate inflexiunile și complexitatea unui limbaj.

Convertirea cuvintelor din forma scrisă în forma vorbită nu este un proces trivial și influențează puternic performanțele unui sistem de sinteză text-vorbire [2].

Componenta de analiză a textului într-un sistem TTS este responsabilă, în general, de determinarea structurii documentului de intrare, de conversia simbolurilor ne-ortografice și de înțelesul textului. Această componentă indică toate cunoștințele despre text, specifică și codează într-un format ușor de folosit în partea de sinteză a vorbirii.

Elementul de bază în componenta de analiză a textului este parserul de text care transcrie și dezambiguesază textul de intrare luând decizii referitoare la înțeles dintr-un areal nelimitat de posibilități.

Pentru limba română există puține abordări de realizare a unui sistem TTS. Astfel, Burileanu et al. [4] prezintă modalitatea de realizare a unui asemenea sistem, incluzând detalii pentru transcrierea fonetică. Pentru parsarea textului, în [4] se folosește o abordare clasică, Lex/Yacc. Trebuie totuși să menționăm că până în prezent, nu există nici un sistem de tip TTS comercial disponibil în România.

ARHITECTURA SI COMPONENTELE UNUI SISTEM DE SINTEZĂ TEXT-VORBIRE

Pentru încadrarea analizorului de text în cadrul unui sistem TTS este utilă amintirea componentelor acestuia. Astfel, un sistem de sinteză este alcătuit din (figura 1) [3]:

Procesorul de text – responsabil cu parsarea textului de intrare și generarea unei structuri de date pe care să o poată utiliza celelalte componente.

Componenta de selecție a unităților acustice – pe baza structurii text și a informațiilor adiționale, se realizează selecția unităților din cadrul bazei de date de unități necesare ulterior în procesul de procesare și concatenare.

Baza de date de unități acustice – conține toate unitățile acustice disponibile.

Procesorul de sunet – aplică procesările necesare (precum modificarea duratei, a pitch-ului, etc.) asupra listei de unități din baza de date de unități, care au fost procesate și concatenate.

În continuare vom face referire la componentele de procesare de text.

Textul de intrare

Textul de intrare este alcătuit dintr-o secvență de paragrafe care pot fi descompuse în fraze iar acestea, la rândul lor, vor fi descompuse în cuvinte.

Un cuvânt nu este doar o secvență de litere mici ci poate conține și alte caractere cu excepția spațiului, liniei noi și a caracterelor speciale ('@','#'). Un cuvânt poate începe și se poate termina doar cu caractere care nu sunt marcate. Aceste caractere valide pot fi separatori (spații), separatori de frază ('.' '!' '?'), separatori de paragraph (linie nouă), caractere de punctuație (';' ':' ',') sau caractere speciale ('@' '#').

Textul de intrare nu poate conține expresii matematice, simboluri intertextuale sau secvențe ambigue de caractere. În cazul în care textul de intrare conține caractere ambigue (altele decât cele menționate mai sus), acestea se înlocuiesc cu caractere speciale.

Arhitectura sistemului TTS și procesarea textului

Procesarea se referă la un set de operații care includ segmentarea textului, dezambiguarea textului și interpretarea semnelor de punctuație.

Segmentarea textului. Textul este parsat prima dată pentru delimitarea paragrafelor, frazelor și cuvintelor, rezultatul fiind o structură internă de tip arbore. În timpul parsării se realizează și dezambiguarea abreviațiilor prin consultarea unui dicționar de abreviații. Acest dicționar este activat la apariția simbolului '!'.

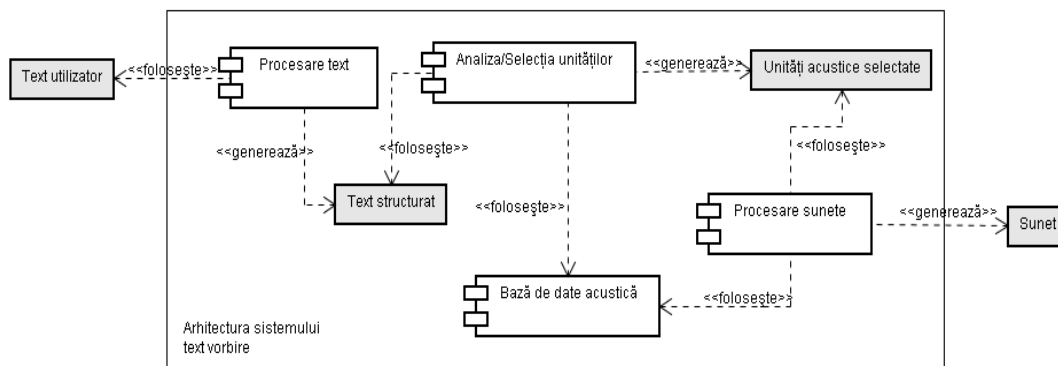


Figura 1. Arhitectura sistemului TTS

Dezambiguarea textului. Textul este parsat a doua oară pentru cuvintele de tip dată, timp și număr. Tot în cadrul acestui pas se realizează transcrierea fonetică și construcția silabelor. Prima verificare în acest pas este pentru cuvintele de tip dată, timp și număr. Dacă se întâlnește un astfel de cuvânt acesta este expandat după care, pentru textul expandat, se realizează transcrierea fonetică și construirea silabelor.

Interpretarea semnelor de punctuație. La nivelul frazelor este aplicat un algoritm de prozodie pentru interpretarea semnelor de punctuație și adăugarea pauzelor la sfârșitul propozițiilor.

Pentru procesarea textului de intrare se utilizează trei gramatici. Prima gramatică interpretează diferite predicat (cuvinte, fraze, paragrafe) activând module specifice pentru salvarea indecșilor și construirea structurii interne de tip arbore. Pentru expandarea cuvintelor (abreviații, dată, timp, numere) se utilizează cea de-a doua gramatică. A treia gramatică se utilizează pentru transcrierea fonetică și construirea silabelor.

Majoritatea modulelor componente ale gramaticilor utilizate sunt pați statice care sunt încărcate în memorie o singură dată. O mică parte a acestor gramatici este dinamică și se încarcă în memorie în momentul execuției. Gramaticile folosite la parsarea textului sunt prezentate în detaliu în [5].

Procesarea textului de intrare ar putea fi realizată într-o singură traversare prin combinarea celor trei gramatici într-una singură. Acest lucru ar determina scăderea timpului de procesare având însă drept dezavantaj creșterea costurilor de mentenanță și scăderea posibilităților de extensie. Considerând timpul câștigat prin cuplarea gramaticilor drept o îmbunătățire minoră, s-a ales parsarea în doi pași care permite o arhitectură modulară și structurată.

Toate cuvintele obținute sunt căutate în dicționar iar, dacă nu sunt găsite, se realizează transcrierea fonetică și despărțirea în silabe.

Gramaticile utilizate sunt construite cu toleranță la erori astfel încât permit ca o propoziție să înceapă cu literă mică sau să existe un text vid la intrare. Cu toate acestea, gramaticile nu rezolvă toate ambiguitățile. Similar cu [4], cu cât textul de intrare este mai clar (mai puține ambiguități și erori) cu atât se obțin rezultate mai bune la ieșire în fișierul de sunet generat.

Componentele modului de procesare de text

Modulul de procesare a textului conține următoarele componente (figura 2):

Dicționarul – conține cuvinte (atât cuvintele comune cât și excepțiile).

Logică de Inteligență artificială – este componenta de inteligență artificială care conține regulile de transcriere fonetică și cele de despărțire în silabe. Se utilizează utilitarul Weka pentru determinarea regulilor pe baza unui set de antrenare. Pentru transcrierea fonetică s-au implementat regulile prezentate în DOOM II. Însă, DOOM II nu acoperă totalitatea cazurilor care pot apărea. Pentru excepții (de exemplu “cea” sau „cia”), se generează arbori de decizie folosind Weka. Acești arbori de decizie sunt învățați pe baza unor exemple bine selectate și realizează cu o precizie foarte bună identificarea fonemelor în cazurile critice. Prezentarea în detaliu a modului de realizare a acestor arbori de decizie iese de sub incidența acestei lucrări.

Prozodia – această componentă analizează structura de date și adaugă informație prozodică precum intonația, accentul și pauzele de vorbire.

SELECȚIA UNITĂȚILOR DIN BAZA DE DATE

Componenta de Selectare Unități pe baza datelor primite de la componenta de Procesare Text va face o selectare a celor mai potrivite unități din Baza de Date. Acestea vor fi apoi concatenate și modelate de compoanta de sinteză a sunetului [4]. Definim ca unitate o reprezentare abstractă a unui segment de vorbire.

Unele sisteme nu folosesc modelul concatenativ, ci generează sunetul. Dezavantajul la un astfel de sistem este faptul că se pierde naturalitatea semnalului sonor deoarece această naturalitate este greu de reprezentat printr-un set de reguli de generare. În sistemele concatenative sunetul final este obținut prin selectarea și concatenarea unor unități din Baza de Date care se potrivesc textului de intrare. Avantajul acestei abordări este păstrarea naturalității vorbitorului. Dezavantajul este că se produc concatenări de unități care nu se potrivesc. Segmentele din vorbire sunt foarte afectate de coarticulare astfel, dacă încercăm să concatenăm două segmente care nu au fost adiacente, putem obține discontinuități prozodice sau spectrale. Discontinuitățile spectrale apar când formații segmentelor nu se potrivesc la punctul de concatenare. Discontinuitățile prozodice se referă la nepotrivirea pitch-ului celor două unități în punctul de concatenare [7].

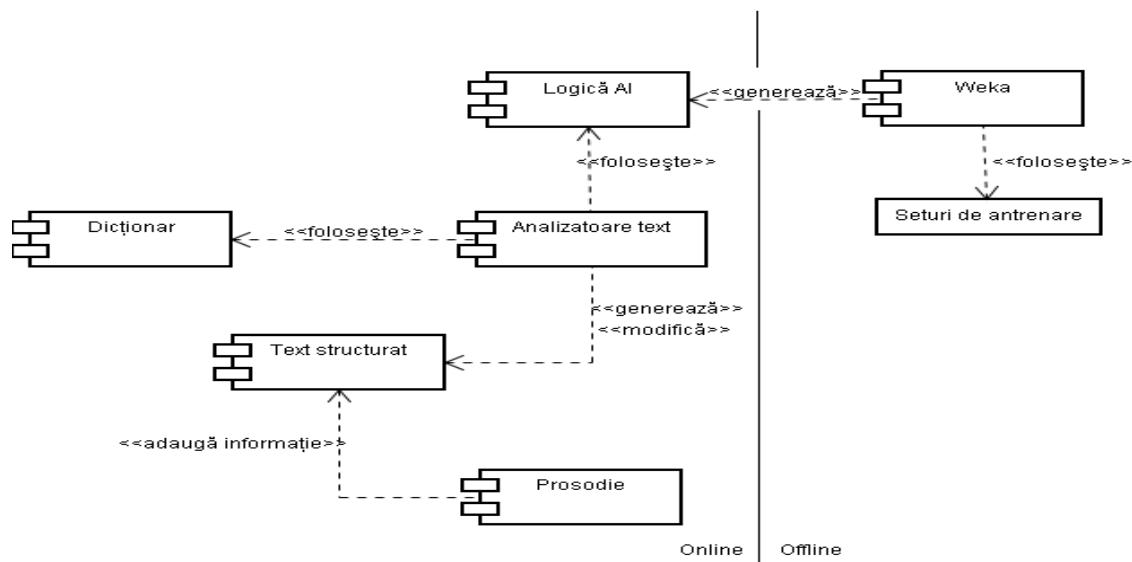


Figura 2. Componentele modului de procesare de text

Tranzițiile între sunetele vorbirii sunt definite în domeniul articulator și sunt exprimate prin intermediul unor relații neliniare. Când sunetele ce urmează să fie înlănțuite sunt suficient de “aproape” unul de altul, relațiile lineare dau totuși rezultate bune [8]. Unitățile care au fost folosite în acest domeniu sunt: foneme care sunt independente de context, difoneme, foneme dependente de context, unități subfonemice, silabe, cuvinte, propoziții și fraze. Selecția unităților se face folosind funcții de cost.

DESPĂRȚIREA ÎN SILABE

În cadrul acestei secțiuni se prezintă algoritmul de despărțire în silabe a cuvintelor din limba română. Algoritmul se aplică asupra textului de intrare după transcrierea fonetică și transformarea literă-în-sunet. Metoda utilizată pentru construirea silabelor este împărțirea în funcție de modul de pronunție. Această metodă a fost aleasă în defavoarea metodei structurate din două motive. Primul motiv este faptul că metoda structurată are nevoie de informații complexe referitoare la fiecare cuvânt, informații precum rădăcina cuvântului, prefix și sufix. Alte informații necesare ar fi fost: dacă cuvântul este compus sau nu. Al doilea motiv este faptul că împărțirea textului de intrare cu metoda structurată determină crearea mai multor unități diferite decât metoda orientată pe pronunție.

Pentru despărțirea în silabe prin metoda orientată pe pronunție se construiesc în primul pas expresii comune pentru potrivirea structurilor cu fonemele consecutive. S-a identificat că în limba română (similar majorității limbilor) structurile utilizate pentru despărțirea în silabe sunt: hiați, diftongi și trifongi. Un hiat este alcătuit din două vocale sonore în secvență. Diftongul este secvența alcătuită dintr-o semivocală și o vocală sonoră sau o vocală sonoră urmată de o semivocală. Trifongul este secvența a trei sunete dintre care unul este vocală sonoră iar celelalte sunt semivocale. În limba română se întâlnesc două cazuri de trifongi. Primul caz este când două sunete semivocalice sunt urmate de o vocală sonoră, iar al doilea caz când un sunet vocalic este precedat și urmat de câte o semivocală.

Prezentăm mai jos abstractizarea acestor structuri.

$V = \text{“a”} \mid \text{“ă”} \mid \text{“î”} \mid \text{“e”} \mid \text{“i”} \mid \text{“o”} \mid \text{“u”}$

$S = \text{“ș”} \mid \text{“ț”} \mid \text{“đ”} \mid \text{“ț”}$

$C = \text{“b”} \mid \text{“k”} \mid \text{“č”} \mid \text{“k”} \mid \text{“d”} \mid \text{“p”} \mid \text{“g”} \mid \text{“g”} \mid \text{“g”} \mid \text{“h”} \mid \text{“j”} \mid \text{“l”} \mid \text{“m”} \mid \text{“n”} \mid \text{“p”} \mid \text{“r”} \mid \text{“s”} \mid \text{“ș”} \mid \text{“t”} \mid \text{“ț”} \mid \text{“v”} \mid \text{“z”}$

$H = V, V$

$D = (S, V) \mid (V, S)$

$T = (S, S, V) \mid (S, V, S)$

unde

“V” – sunetele vocalice

“C” – sunetele consonantice

“H” – regula de obținere a unui hiat

“D” – regulile de obținere a unui diftong

“T” – regulile de obținere a unui trifong

Pentru grupurile de consoane consecutive am identificat patru cazuri în care despărțirea în silabe este realizată diferit de regula generală. Două dintre excepții apar la întâlnirea unui grup alcătuit din patru consoane (FCE1 – prima excepție pentru grup patru consoane; FCE2 – a doua excepție pentru grup patru consoane). O excepție apare la grupul de trei consoane consecutive (THCE - excepție pentru trei consoane) și ultimul caz pentru două consoane consecutive (TWCE - excepție pentru două consoane).

$FCE1 = \text{“stșk”} \mid \text{“ldsp”} \mid \text{“ngst”} \mid \text{“rnbl”} \mid \text{“nsfr”} \mid \text{“nsgr”} \mid \text{“nsp”} \mid \text{“rtkh”} \mid \text{“rtdr”} \mid \text{“rtsk”} \mid \text{“rtst”} \mid \text{“stpr”} \mid \text{“stsk”}$

$FCE2 = \text{“rst”}, C$

$THCE = \text{“ltč”} \mid \text{“nsč”} \mid \text{“rgș”} \mid \text{“rtț”} \mid \text{“mpț”} \mid \text{“nkș”} \mid \text{“nkț”} \mid \text{“ldm”} \mid \text{“lpn”} \mid \text{“ndb”} \mid \text{“ndk”} \mid \text{“nsb”} \mid \text{“nsk”} \mid \text{“nsd”} \mid \text{“nsf”} \mid \text{“nsh”} \mid \text{“nsl”} \mid \text{“nsm”} \mid \text{“nsn”} \mid \text{“nsp”} \mid \text{“nss”} \mid \text{“nsv”} \mid \text{“ntl”} \mid \text{“rtb”} \mid \text{“rtk”} \mid \text{“rth”} \mid \text{“rtj”} \mid \text{“rtm”} \mid \text{“rtp”} \mid \text{“rts”} \mid \text{“rtt”} \mid \text{“rtv”} \mid \text{“stb”} \mid \text{“stk”} \mid \text{“std”} \mid \text{“stf”} \mid \text{“stg”} \mid \text{“std”} \mid \text{“stn”} \mid \text{“stp”} \mid \text{“sts”} \mid \text{“stt”} \mid \text{“stv”} \mid \text{“lpt”} \mid \text{“mpt”} \mid \text{“nkt”} \mid \text{“ndv”} \mid \text{“rkt”} \mid \text{“rtf”} \mid \text{“stm”}$

TWCE = ("b" | "k" | "d" | "f" | "q" | "h" | "p" | "t" | "v" | "g"), ("l" | "r")

unde

FCE1 – prima excepție patru consoane

FCE2 – a doua excepție patru consoane

THCE – excepție trei consoane

FWCE – excepție două consoane

În continuare prezentăm regulile generale de despărțire în silabe. Aceste reguli au fost generate utilizând informațiile din DOOM II [1].

Pentru grupurile de consoane există cinci cazuri, pornindu-se de la existența unei singure consoane până la existența a cinci consoane. Pentru grupurile de vocale și semivocale consecutive există patru cazuri, pornindu-se de la o singură vocală până la patru vocale și semivocale consecutive.

C, C, C, C, C → "CC-CCC"

FCE1 → "CCC-C"

FCE2 → "CC-CC"

C, C, C, C → "C-CCC"

THCE → "CC-C"

C, C, C → "C-CC"

TWCE → "-CC"

C, C → "C-C"

C → "-C"

S, V, S, V → "SV-SV"

V, T → "V-T"

T → nicio acțiune

V, D → "V-D"

D → no action

H → "V-V"

"[i]" → "[i]" și legătură

V → nicio acțiune

unde

"→" – acțiunea care va avea loc la depistarea secvenței respective după ce textul de intrare a fost transformat în foneme.

CONCLUZII

În cadrul acestui articol s-a descris arhitectura unui sistem de sinteză text-vorbire pentru limba română, componentele acestui sistem și despărțirea în silabe.

Acest sistem TTS utilizează o abordare concatenativă a unităților de selecție după cum este recomandat în [2].

Textul de intrare este inițial procesat de componenta de analiză a textului care are rolul de a determina structura și de a realiza normalizarea textului (expandarea numerelor, dezambiguarea, etc.). Modulul de selecție a unităților caută în dicționarul extins și selectează unitățile corespunzătoare și modul de pronunție. În cazul în care nu s-a găsit cuvântul în dicționar, are loc analiza fonetică care realizează transformare literă-în-sunet prin utilizarea utilitarului Weka [5]. Transcrierea fonetică este succedată de despărțirea în silabe prin care se determină unitățile de formare a sunetului. Aceste unități sunt selectate din baza de date acustică și trimise modulului de procesare de sunet.

Articolul s-a axat pe componentele de procesare de text și pe despărțirea în silabe.

REFERINȚE

1. Romanian Academy, Iorgu Jordan - Al. Rosetti Lingvistic Institute, Dicționarul Ortografic, Ortoepic și Morfologic al Limbii Române, 2nd Edition. Univers Enciclopedic București, 2005
2. Huang, X., Acero, A., Hsiao-Wuen, H.: Spoken Language Processing: A Guide to Theory, Algorithms, and System Development. Prentice Hall (2001)
3. 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), Czech Republic
4. D. Burileanu, "Basic research and implementation decisions for a text-to-speech synthesis system in romanian," International Journal of Speech Technology, vol. 5, no. 3, pp. 211–225(15), 2002
5. A. Saupe, M. A. Ordean, L. R. Teodorescu, M. Ordean, and G. C. Silaghi, "Efficient Parsing of Romanian Language for Text-to-Speech Purposes", Proceedings of the 12th International Conference on Text, Speech and Dialogue, Pilsen, Czech Republic, 2009, to appear in LNAI, Springer Verlag
6. W. Daelemans and A. van den Bosch, "Language-independent data-oriented grapheme-to-phoneme conversion," in Progress in Speech Processing. Springer Verlag, 1997, pp. 77–89
7. Sproat, R.: Multilingual text analysis for text-to-speech synthesis. Natural Language Engineering 2(4) (1996)
8. Hunt, A.J., Black, A.W.: Unit selection in a concatenative speech synthesis system using a large speech database. In: ICASSP '96: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE Computer Society (1996)