(19) OFICIUL DE STAT
PENTRU INVENŢII ŞI MĂRCI
Bucureşti

ROMÂNIA

(11) **RO 127590 A2**
(51) Int.Cl.
G10L 13/08 (2006.01);
G06F 17/20 (2006.01)

(12) # CERERE DE BREVET DE INVENŢIE

(21) Nr. cerere: **a 2010 01246**

(22) Data de depozit: **29.11.2010**

(41) Data publicării cererii:
29.06.2012 BOPI nr. **6/2012**

(71) Solicitant:
• **UNIVERSITATEA "POLITEHNICA" DIN BUCUREŞTI**, *SPLAIUL INDEPENDENŢEI NR.313, SECTOR 6, BUCUREŞTI, B, RO*

(72) Inventatori:
• **BURILEANU DRAGOŞ**, *STR. JOHANNES KEPLER NR. 2, BL. 2, SC. 2, AP. 61, SECTOR 2, BUCUREŞTI, B, RO;*
• **UNGUREAN CĂTĂLIN**, *STR. MOLDOVIŢA NR. 2, BL. M2D9/4, AP. 44, SECTOR 4, BUCUREŞTI, B, RO*

(54) # METODĂ DE PREDICŢIE A ACCENTULUI LEXICAL ÎN SINTEZA VORBIRII PORNIND DE LA TEXT ÎN LIMBA ROMÂNĂ

(57) **Rezumat:**

Invenţia se referă la o metodă de predicţie a accentului lexical, destinată utilizării în cadrul etapei de analiză prozodică, din cadrul unui etaj de prelucrare a limbajului natural într-un sistem de sinteză a vorbirii pornind de la un text în limba română, cu aplicaţii în dezvoltarea interfeţelor inteligente de comunicare om-maşină. Metoda conform invenţiei utilizează o strategie statistică, bazată pe *n*-gramele construite la nivel de caracter, şi foloseşte un algoritm de netezire de tip "Katz back-off" modificat. Metoda necesită cunoştinţe lingvistice limitate, are la bază un corpus de antrenare de dimensiuni medii şi este foarte precisă în predicţia poziţiei accentului lexical în limba română.

Revendicări: 1

RO 127590 A2

METODĂ DE PREDICŢIE A ACCENTULUI LEXICAL ÎN SINTEZA VORBIRII PORNIND DE LA TEXT ÎN LIMBA ROMÂNĂ

Invenţia se referă la o metodă de predicţie a accentului lexical, destinată utilizării în cadrul etapei de analiză prozodică din cadrul etajului de prelucrare a limbajului natural într-un sistem de sinteză a vorbirii pornind de la text în limba română, cu aplicaţii diverse în dezvoltarea interfeţelor inteligente de comunicare om-maşină.

Sunt cunoscute un număr de studii pe plan mondial în problema poziţionării automate a accentelor lexicale pentru sisteme de sinteză a vorbirii pornind de la text (TTS – *"Text-to-Speech"*), care se bazează fie pe o serie de reguli (*"rule based"*), fie sunt de tip statistic (*"data driven"*).

– De exemplu, rezultate recente foarte bune au fost raportate în [Q. Dou, S. Bergsma, S. Jiampojamarn, G. Kondrak, "A ranking approach to stress prediction for letter-to-phoneme conversion", *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, Suntec, Singapore, 2009, pp. 118-126], folosind un algoritm de tip *data driven*, prin reprezentarea fiecărui cuvânt ca subşir de caractere. Algoritmul, cunoscut sub denumirea de SVM – *"Support Vector Machine"*, este antrenat a-priori pentru a clasifica toate modelele de accent. Autorii au raportat o acurateţe de 98% pentru predicţia accentului primar (calculat la nivelul fonemelor) pentru limba engleză, 97,1% pentru germană şi 93,1% pentru olandeză.

Dezavantajul principal al acestei metode constă în faptul că ea este destinată altor limbi decât limba română, fiind bazată pe particularităţi lingvistice complet diferite şi de asemenea pe resurse lingvistice în format electronic ce nu sunt disponibile pentru limba română.

– În ceea ce priveşte limba română, singura încercare de soluţionare a problematicii plasării accentelor lexicale din perspectiva folosirii în sistemele TTS, este descrisă în [E. Oancea, A. Badulescu, "Stressed syllable determination for Romanian words within speech synthesis applications", *International Journal of Speech Technology,* vol. 5, no. 3, Kluwer,

Lucrările au fost realizate în cadrul contractului de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

2002, pp. 237-246.]. Autorii propun şi evaluează experimental o metodă bazată pe reguli prin care calculează poziţia silabelor accentuate în funcţie de mulţi parametri morfologici, fonetici şi lexicali, şi anume: numărul de caractere şi silabe, tipul silabelor, secvenţele de trei caractere de la începutul şi sfârşitul cuvintelor. Aceşti parametri sunt folosiţi în vederea antrenării sistemului prin împărţirea cuvintelor de intrare pe clase, iar în vederea recunoaşterilor accentelor cuvintele de test sunt clasificate după clasa de apartenenţă. A fost raportată o eroare maximă ("*Word Error Rate*") de 6% pentru un corpus de test de 4.500 cuvinte, bază de test care a fost folosită şi pentru generarea regulilor.

Dezavantajele metodei, în afară de rezultatele modeste, sunt un set incomplet de reguli şi lipsa unor teste mai complexe şi adecvate.

Problema tehnică pe care şi-a propus să o rezolve invenţia poate fi formulată pornind de la constatarea că modelarea prozodică a vorbirii generate de un sistem TTS este o etapă fundamentală în vederea obţinerii unei vorbiri sintetizate cât mai naturale plecând de la un text de intrare oarecare. Realizarea unei prozodii corecte este esenţială pentru un sistem TTS deoarece oferă informaţii referitoare la semnificaţiile posibile ale mesajului vocal şi la structura vorbirii, care nu rezultă implicit dintr-o simplă rostire segmentată. Pot fi date numeroase exemple în care mesajul vocal sintetizat devine ambiguu sau chiar neinteligibil din cauza faptului că informaţia prozodică fie lipseşte, fie este incorect poziţionată în fişierul audio generat. Pe de altă parte, trebuie menţionat că prozodia este dificil de prezis în sistemele TTS din cauza faptului că textul de intrare conţine puţină informaţie referitoare la structura şi semnificaţia frazelor şi că asemenea informaţii sunt greu de dedus automat. În acest context, este cunoscut faptul că *accentul lexical*, concept care apare la nivelul silabelor cuvintelor, are o influenţă majoră asupra conturului intonaţional generat de un sistem TTS: silabele accentuate sunt pronunţate cu un efort mai mare decât celelalte silabe din acelaşi cuvânt şi sunt caracterizate de un nivel mai înalt al frecvenţei fundamentale şi deseori de o durată mai mare a fonemelor.

Metoda care face obiectul prezentei invenţii rezolvă problema tehnică menţionată anterior prin faptul că utilizează o strategie statistică, bazată în principal pe *n*-gramele construite la nivel de caracter, destinată poziţionării automate a accentelor lexicale pentru cuvintele scrise în limba română; abordarea statistică a fost aleasă ţinând cont de faptul că poziţia accentului lexical în limba română este variabilă, fiind practic imposibil de prezis pe bază de reguli.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

În vederea rezolvării problemelor generate de insuficienţa datelor de antrenare, metoda foloseşte un algoritm de netezire de tip *Katz back-off* modificat. În plus, în vederea reducerii efectelor nedorite cauzate de poziţionarea incorectă a accentului lexical în vorbirea sintetizată, se utilizează un prag de decizie variabil. Cuvintele monosilabice sunt considerate ca fiind fără accent lexical şi sunt puse în evidenţă cu ajutorul unui algoritm de despărţire automată în silabe. Metoda necesită cunoştinţe lingvistice limitate, este bazată pe un corpus de antrenare de dimensiuni medii şi este foarte precisă în predicţia accentului lexical în limba română.

Se prezintă în continuare în detaliu obiectul invenţiei, un exemplu de aplicare a sa constând din utilizarea metodei de către autorii invenţiei în cadrul etajului de prelucrare a limbajului natural pentru un sistem complet de sinteză a vorbirii pornind de la text în limba română (dezvoltat în cadrul colectivului din care fac parte autorii).

Se poate spune că realizarea unei prozodii corecte este practic cea mai mare provocare pentru sistemele TTS actuale. În acest sens, un aspect foarte important, puternic legat de prozodie şi de naturaleţea percepută a vorbirii sintetizate, este *accentul lexical*, concept care apare la nivelul silabelor cuvintelor şi care are o influenţă majoră asupra conturului intonaţional generat de un sistem TTS: silabele accentuate sunt pronunţate cu un efort mai mare decât celelalte silabe din acelaşi cuvânt şi sunt caracterizate de un nivel mai înalt al frecvenţei fundamentale şi deseori de o durată mai mare a fonemelor. Astfel, semnalul vocal sintetizat la ieşirea unui sistem TTS este substanţial îmbunătăţit prin prezenţa accentului lexical prin faptul că vorbirea sună mult mai natural.

Există mai multe definiţii, teorii şi modele de accent lexical. Deşi unii autori apreciază că există mai multe niveluri de accent lexical (*primar, secundar* etc.) în interiorul aceluiaşi cuvânt, diferenţierea fiind făcută după nivelul de accentuare, metoda propusă ia în considerare cea mai comună abordare, şi anume aceea că **un cuvânt poli-silabic are o singură silabă accentuată (corespunzătoare accentului lexical primar)**. În multe alte limbi vorbite ca engleza, germana, spaniola sau româna, **accentul lexical are o poziţie variabilă**, fiind extrem de dificil de prezis.

Înainte de a descrie metoda care face obiectul invenţiei, vom explica succint câteva particularităţi ale accentului lexical pentru limba română.

După cum am menţionat anterior, accentul în limba română este liber (sau dinamic) şi teoretic poate să fie poziţionat pe oricare dintre silabele unui cuvânt (Tabelul 1).

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Model	Exemplu
–⊥	**ca-fea**
⊥–	**ca-să**
––⊥	**pi-ja-ma**
–⊥–	**pe-re-te**
⊥––	**re-pe-de**
–––⊥	**pa-ta-la-ma**
––⊥–	**pro-fe-soa-ră**
–⊥––	**pe-re-te-le**
⊥–––	**la-cu-ri-le**
––––⊥	**(a) în-tre-bu-in-ţa**
–––⊥–	**as-cu-ţi-toa-re**
––⊥––	**li-pi-to-ri-le**
–⊥–––	**fa-mi-li-i-le**
⊥–––––	**(al) şap-te-spre-ze-ce-lea**

Tabelul 1. Diferite situaţii de prezenţă a accentului lexical în limba română

Pe de altă parte, în limba română există multe cuvinte care sunt neaccentuate lexical. În această categorie sunt introduse o serie de *cuvinte funcţionale* ca: articolele, prepoziţiile şi conjuncţiile, dar şi cuvinte din alte categorii ca verbe şi substantive. Deoarece există excepţii de la aceste reguli, nu poate fi trasă concluzia că toate cuvintele funcţionale sunt neaccentuate iar restul sunt accentuate. În urma unui studiu aprofundat, autorii invenţiei au găsit totuşi o regulă care poate fi enunţată astfel: ***cuvintele monosilabice ale limbii române pot fi considerate ca fiind neaccentuate***; această regulă este consistentă cu definiţia accentului dată anterior şi a fost introdusă în metoda propusă. În plus (ca o consecinţă directă a regulii anterioare), am considerat că fiecare cuvânt polisilabic, poartă un singur accent lexical. Accentul acţionează de fapt la nivelul unei singure vocale dintr-o silabă, iar caracterele vecine creează un context limitat în care efectul acestuia poate fi pus în evidenţă. După găsirea vocalei accentuate, cu ajutorul unui modul de despărţire automată în silabe (care nu face obiectul prezentei invenţii), evidenţierea silabei accentuate este imediată.

Metoda de predicţie a poziţiei accentului lexical pentru limba română ce face obiectul invenţiei are la bază o abordare statistică. Metoda este de tip *data driven*, bazată pe *n*-grame, statisticile de limbaj fiind realizate la nivel de caracter.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Modelele sunt construite pe baza unui corpus de antrenare. Din datele de antrenare au fost extrase iniţial toate *n*-gramele, după care modelele de *n*-grame au fost antrenate prin calcularea distribuţiilor de probabilitate. În implementarea propusă în cadrul invenţiei, numărul total de caractere este de 34, şi anume: 31 de litere ale alfabetului limbii române (cinci litere fiind scrise cu diacritice), semnul (*markerul*) de accent ', simbolurile de start şi de stop, "<" şi respectiv ">".

În mod evident, numărul total de *n*-grame posibile este foarte mare (pentru o lungime de 7 caractere acest număr este 34^7). Una dintre problemele cele mai importante care apare în astfel de implementări statistice este cea cauzată de insuficienţa datelor de antrenare care necesită realizarea unui compromis între istoria *n*-gramelor (lungimea acestora) şi dimensiunea corpusului de antrenare. Cu o istorie mai lungă, deci o lungime mai mare a *n*-gramelor, rezultatele pot fi sensibil mai bune, dar poate să apară o insuficientă antrenare din cauza fenomenului de *data sparseness* tradus prin lipsa unui număr mare de *n*-grame. Soluţiile cele mai folosite sunt cele de utilizare a tehnicilor de netezire (*smoothing*) pentru o re-estimare a probabilităţilor *n*-gramelor care lipsesc sau care sunt întâlnite foarte rar în corpusul de antrenare. Este general acceptată ideea că folosind *n*-grame de lungime mai mare, împreună cu tehnici de tip netezire, se obţin rezultate sensibil mai bune decât folosind *n*-grame de lungime mai mică dar pentru care modelul de limbaj este mai consistent, dacă se ia în calcul o aceeaşi dimensiune a corpusului de antrenare.

Pentru descrierea metodei să presupunem că fiecare cuvânt **w** este compus din N litere:

$$w = l_1 l_2 ... l_N \tag{1}$$

Prin adăugarea caracterelor de start şi stop, **w** devine:

$$\underline{\mathbf{w}} = < l_1 l_2 ... l_N > \tag{2}$$

Vom exemplifica aceste notaţii cu cuvântul **casă**, pentru care $N = 4$ (cuvântul are 4 caractere) şi $M = 2$ (cuvântul are două vocale: **a** şi **ă**): **w** = **casă** devine $\underline{\mathbf{w}}$ = <**casă**>.

După cum menţionam anterior, accentul lexical acţionează la nivelul vocalelor unei silabe, iar un cuvânt are un singur accent lexical. În abordarea propusă dacă **w** are M vocale, fiecare dintre aceste M caractere poate avea marca accentului asociată cuvântului de start. Mai mult, o singură vocală poate fi accentuată. Soluţia de predicţie a poziţiei accentului lexical constă în găsirea celei mai bune variante dintre cele M posibile. Aceste variante sunt obţinute prin plasarea markerului de accent în faţa fiecărei vocale.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Prin urmare, pentru un cuvânt pentru care luăm decizia că poate avea accent lexical (fie acesta $\underline{\mathbf{w}}'$), generăm toate variantele posibile $\mathbf{g}_j$ prin plasarea markerului de accent pe poziţiile j, cu $j = 1,...,M$; pentru exemplul anterior vom avea două variante:

$\mathbf{g}_1$ = <**c'asă**> cu $g_{11} = <$, $g_{12} = \mathbf{c}$, $g_{13} = \mathbf{'}$, $g_{14} = \mathbf{a}$, $g_{15} = \mathbf{s}$, $g_{16} = \mathbf{ă}$, $g_{17} = >$, şi

$\mathbf{g}_2$ = <**cas'ă**> cu $g_{21} = <$, $g_{22} = \mathbf{c}$, $g_{23} = \mathbf{a}$, $g_{24} = \mathbf{s}$, $g_{25} = \mathbf{'}$, $g_{26} = \mathbf{ă}$, $g_{27} = >$

Se poate observa că numărul total de caractere pentru $\underline{\mathbf{w}}'$ este $N+3$ (pentru un cuvânt care are N caractere), dar pentru a simplifica notaţiile, pentru restul explicaţiilor vom considera că $\underline{\mathbf{w}}'$ are N caractere. Soluţia găsirii celei mai bune variante dintre cele M posibile este dată prin folosirea modelelor de n-grame.

Etapa de antrenare

Pentru antrenarea modelelor de n-grame am folosit un corpus $T1$, compus din cuvinte având accentul lexical poziţionat manual iar delimitatorii de start şi stop poziţionaţi automat. Un corpus de tip *held-out*, denumit $T2$ a fost folosit în etapa de antrenare în vederea optimizării parametrilor aleşi în etapa de netezire.

Modelele au fost construite folosind datele de antrenare. Toate n-gramele posibile au fost mai întâi extrase din $T1$ după care modelele de n-grame au fost antrenate prin calculul distribuţiilor de probabilitate.

Pentru fiecare cuvânt din $T1$ am extras toate variantele posibile de n-grame şi am calculat probabilităţile acestora. De exemplu, pentru cuvântul <**c'asă**>, n-gramele extrase (de lungime maximă 7) sunt figurate în Tabelul 2.

Lungime *n*-grame	*n*-grame
1	**<, c, a, ', s, ă, >**
2	**<c, c', 'a, as, să, ă>**
3	**<c', c'a, 'as, asă, să>**
4	**<c'a, c'as, 'asă, asă>**
5	**<c'as, c'asă, 'asă>**
6	**<c'asă, c'asă>**
7	**<c'asă>**

Tabelul 2. n-gramele extrase pentru cuvântul **<c'asă>**

În general, pentru un cuvânt $\mathbf{w} = l_1 l_2 ... l_N$, un model de limbaj cu n-grame poate fi extras ca o distribuţie de probabilitate $p(\mathbf{w})$, obţinută pe baza probabilităţilor condiţionate astfel:

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

$$p(\mathbf{w}) = p(l_1)p(l_2 \mid l_1)p(l_3 \mid l_1 l_2)...p(l_N \mid l_1 l_2 ... l_{N-1}) = \prod_{t=1}^{N} p(l_t \mid l_1 l_2 ... l_{t-1}) = \prod_{t=1}^{N} p(l_t \mid l_1^{t-1}) \quad (3)$$

unde l_1^{t-1} este o notaţie pentru $l_1 l_2 ... l_{t-1}$.

În modelul *n*-gramelor facem aproximarea că probabilitatea unui caracter depinde numai de *n*–1 caractere precedente:

$$p(\mathbf{w}) \approx \prod_{t=1}^{N} p(l_t \mid l_{t-n+1} l_{t-n+2} ... l_{t-1}) = \prod_{i=1}^{N} p(l_t \mid l_{t-n+1}^{t-1}) \quad (4)$$

unde l_{t-n+1}^{t-1} reprezintă *istoria* caracterului l_t, iar lungimea *n*-gramelor (*n*) mai este cunoscută şi ca *ordinul modelului*. De exemplu, pentru $n = 3$ se obţine modelul *trigramelor*:

$$p(\mathbf{w}) \approx p(l_3 \mid l_1 l_2) p(l_4 \mid l_2 l_3) ... p(l_N \mid l_{N-2} l_{N-1}) \quad (5)$$

În ecuaţia (4), probabilitatea condiţionată $p(l_t \mid l_{t-n+1}^{t-1})$ poate fi estimată pe baza principiului probabilităţii maxime (*"maximum likelihood"*), prin numărarea frecvenţelor de apariţie ale *n*-gramelor în corpusul de antrenare:

$$p(l_t \mid l_{t-n+1}^{t-1}) = \frac{c(l_{t-n+1}^{t})}{\sum_{l_t} c(l_{t-n+1}^{t})} \quad (6)$$

unde $c(l_{t-n+1}^{t})$ numără de cate ori *n*-grama $l_{t-n+1} l_{t-n+2} ... l_t$ a apărut în corpusul de antrenare, iar $\sum_{l_t} c(l_{t-n+1}^{t})$ este numărul total de apariţii ale istoriei $l_{t-n+1} l_{t-n+2} ... l_{t-1}$.

Totuşi, o modelare care se ghidează numai după principiul ML nu este o alegere suficient de bună. Problema datelor insuficiente de antrenare obţinute din *T*1 va fi echivalentă cu un număr nul de apariţii pentru *n*-gramele neobservate în timpul antrenării, iar emisia de probabilitate va fi $p(\mathbf{w}) = 0$, ceea ce va conduce la o probabilitate finală nulă, semnificând faptul că întregul cuvânt nu apare în datele de antrenare, deşi doar o singură *n*-gramă nu a fost găsită. Rezultatul final va fi că la ieşire cuvântul de test va fi fără accent lexical (algoritmul nu va putea decide care este varianta corectă), sau va avea accentul poziţionat greşit (ceea ce este şi mai rău pentru sarcina TTS de care ne ocupăm) din cauza faptului că s-a obţinut o probabilitate ne-nulă pentru o variantă eronată.

Pentru reducerea efectelor nedorite cauzate de insuficienţa datelor de antrenare, am implementat în metoda propusă o tehnică de netezire de tip *Katz backoff*, care extinde algoritmul *Good-Turing* de estimare a probabilităţilor prin combinarea modelelor de *n*-grame de ordin mai mare cu cele de ordin mai mic. Frecvenţele de apariţie pentru *n*-gramele generice l_{t-n+1}^{t} sunt modificate după formula:

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

$$c_{katz}(l_{i-n+1}^{i}) = \begin{cases} d_r r & \text{dacă} \quad r > 0 \\ \alpha(l_{i-n+1}^{i-1}) c_{katz}(l_{i-n+2}^{i}) & \text{dacă} \quad r = 0 \end{cases} \tag{7}$$

unde r este numărul de n-grame observate:

$$r = c(l_{i-n+1}^{i}), \tag{8}$$

$\alpha(l_{i-n+1}^{i-1})$ sunt coeficienţi *backoff* de ponderare care asigură că suma probabilităţilor este egală cu 1, iar d_r este o rată de reducere dată de formula:

$$d_r = \begin{cases} 1 & \text{dacă} \quad r > k \\ \dfrac{\dfrac{r^*}{r} - \dfrac{(k+1)n_{k+1}}{n_1}}{1 - \dfrac{(k+1)n_{k+1}}{n_1}} & \text{dacă} \quad r \le k \end{cases} \tag{9}$$

unde k este o valoare de prag.

Trebuie specificat faptul că pentru $r \le k$ metoda Katz este obţinută din algoritmul de estimare Good-Turing, iar pentru $r > k$ numărul de apariţii este cel obţinut prin estimarea nemodificată a probabilităţii maxime. De asemenea, termenul r^* din ecuaţia (9) reprezintă numărul de n-grame, modificat (netezit) prin algoritmul *back-off*, şi este dat de formula:

$$r^* = (r+1) \cdot \frac{n_{r+1}}{n_r} \tag{10}$$

unde n_r este numărul total de n-grame diferite care apar de r ori în datele de antrenare.

Noua probabilitate este calculată după formula:

$$p_{katz}^*(l_i \mid l_{i-n+1}^{i-1}) = \frac{c_{katz}(l_{i-n+1}^{i})}{\sum_{l_i} c_{katz}(l_{i-n+1}^{i})} \tag{11}$$

Filozofia metodei de tip Katz *backoff smoothing* este aceea că încearcă să menţină eficienţa de rezoluţie a n-gramelor, atunci când nu există un corpus de antrenare consistent, prin reducerea lungimii istoriei unei n-grame de ordin n pentru caracterul l_i la o istorie mai scurtă l_{i-n+2}^{i-1} pentru care există suficiente date de antrenare. Algoritmul coboară recursiv, dacă este nevoie, până la nivelul unigramelor.

În metoda *Katz backoff*, evenimentelor neobservate li se alocă o probabilitate diferită de zero prin extragerea acesteia din evenimentele nenule (prin intermediul lui d_r). Extragerea probabilităţii se face numai din n-gramele cu o frecvenţă de apariţie sub ordinul k de prag.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Apariţiile extrase sunt distribuite către *n*-gramele neobservate de lungime *n* prin intermediul distribuţiilor de ordin *n*–1, dar cu păstrarea numărului total de apariţii (*counts*).

În algoritmul Katz original, coeficienţii de ponderare $\alpha(l_{i-n+1}^{i-1})$ sunt calculaţi pentru fiecare istorie l_{i-n+1}^{i-1} astfel încât numărul total de apariţii să se conserve, ceea ce se traduce prin $\sum_{l_i} c_{katz}(l_{i-n+1}^{i}) = \sum_{l_i} c(l_{i-n+1}^{i})$.

Autorii acestei invenţii **propun un algoritm simplificat** dar eficient care pleacă de la ipoteza că pentru aceeaşi lungime a *n*-gramelor, coeficienţii de ponderare $\alpha(l_{i-n+1}^{i-1})$ pot fi aceeaşi. Alegerea lor se face pe baza corpusului de tip *held-out* *T*2 prin alegerea acelor coeficienţi $\alpha_{held-out}(n)$ care minimizează eroarea de predicţie a accentului lexical. Se va observa totuşi că, deşi algoritmul nu garantează o sumă a distribuţiilor de probabilitate egală cu 1, acest lucru nu afectează performanţa globală a metodei propuse.

Formula de calcul a emisiilor de probabilitate finale va fi dată de formula:

$$p_{katz}(l_{i-n+1}^{i}) = \begin{cases} p_{katz}^{*}(l_{i-n+1}^{i}) & \text{dacă} \quad c(l_{i-n+1}^{i}) > 0 \\ \alpha_{held-out}(n) \cdot p_{katz}(l_{i-n+2}^{i}) & \text{dacă} \quad c(l_{i-n+1}^{i}) = 0 \end{cases} \tag{12}$$

Într-un asemenea model recursiv probabilităţile de ordin *n* sunt calculate pe baza probabilităţilor de ordin *n*–1. Algoritmul se opreşte la nivelul unigramelor prin alegerea distribuţiei de tip ML, astfel încât în această etapă probabilităţile bigramelor sunt calculate direct din probabilităţile unigramelor de tip *"Maximum Likelihood Estimation"*:

$$p_{katz}(l_{i-1}^{i}) = \begin{cases} p_{katz}^{*}(l_{i-1}^{i}) & \text{dacă} \quad c(l_{i-1}^{i}) > 0 \\ \alpha_{held-out}(n) \cdot p_{MLE}(l_i) & \text{dacă} \quad c(l_{i-1}^{i}) = 0 \end{cases} \tag{13}$$

Pentru a putea realiza o recursie completă trebuie numărate toate apariţiile *n*-gramelor, până la nivelul unigramelor. Frecvenţele de apariţie sunt salvate în fişiere separate şi apoi re-estimate prin algoritmul Good-Turing.

Pentru a însuma explicaţiile anterioare, **etapa de antrenare** are următorii paşi:

1. Din corpusul de antrenare *T*1 (compus din cuvinte cu accentele poziţionate corect şi cu delimitatorii de start şi stop pentru fiecare cuvânt), sunt extrase toate *n*-gramele, la nivel de caracter. Pentru o istorie de lungime *n*, sunt extrase de asemenea *n*-gramele de ordin *n*–1, *n*–2,...,1.
2. Se numără toate apariţiile *n*-gramelor de ordin *n*, *n*–1,...,1 şi apoi se calculează şi se salvează probabilităţile de tip ML (sau numărul de apariţii) rezultate.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

3. Pe baza corpusului *held-out* $T2$ se estimează coeficienţii de ponderare $\alpha_{held-out}(n)$ care minimizează eroarea de predicţie a accentului lexical în $T2$ (care în această etapă joacă rolul *de facto* al corpusului de test).
4. Folosind coeficienţii de ponderare estimaţi în etapa a 3-a, se recalculează probabilităţile potrivit metodei Katz *backoff*. Se renunţă la datele de antrenare anterioare.

Etapa de testare

În etapa de test fiecare cuvânt din textul de intrare este comparat mai întâi cu toate intrările dintr-un dicţionar de forme omografe; dacă un cuvânt este găsit în dicţionar, acesta va fi lăsat nemodificat şi trimis către un alt modul al sistemului TTS pentru o procesare ulterioară. În cazurile rare în care un cuvânt intră în algoritmul de poziţionare a accentelor, rezultatul poate să nu fie o formă corectă semantic.

După cum am menţionat anterior, modulul de despărţire în silabe din sistemul TTS este folosit pentru a identifica dacă un cuvânt este de tip monosilabic, situaţie în care acesta va fi lăsat nemodificat (deoarece astfel de cuvinte nu prezintă accent). Dacă un cuvânt are cel puţin două silabe, atunci acesta este procesat mai departe, după următorul algoritm:

1. Se pleacă de la forma **w**, şi se obţine forma $\underline{\mathbf{w}}$ prin plasarea delimitatorilor de start şi stop:
$$\underline{\mathbf{w}} = l_1 l_2 .. l_i .. l_N \text{, în care } l_1 = < \text{ şi } l_N = > .$$
2. Pentru fiecare formă $\underline{\mathbf{w}}$, se generează toate variantele posibile accentuate $\mathbf{g}_j$, $j = 1,...,M$ (în care M este numărul de vocale din $\underline{\mathbf{w}}$), prin inserarea unui caracter de accent înaintea fiecărei vocale.
3. Din fiecare variantă $\mathbf{g}_j$ se extrag n-gramele de ordin n, n–1,…,1 şi se calculează emisiile de probabilitate finale după formula generală din algoritmul Katz *backoff*, în care probabilităţile condiţionate sunt cele obţinute din datele salvate în timpul etapei de antrenare.
4. Varianta $\mathbf{g}_{j,\max}$ care are probabilitatea maximă este furnizată la ieşirea sistemului.

Datele folosite în **experimentele de evaluare** ale metodei ce face obiectul invenţiei au fost extrase din texte literare scrise în limba română, totalizând peste 10 milioane de cuvinte, iar corpusul T de start conţine peste 330.000 de cuvinte diferite (cuprinzând multe forme flexionate ale cuvintelor de bază). Acest dicţionar a fost procesat mai departe prin plasarea

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

manuală a semnelor corespunzătoare accentelor lexicale înaintea vocalelor accentuate. După această etapă au fost înlăturate cuvintele omografe prin eliminarea automată a cuvintelor care au două sau mai multe forme accentuate, din aceste cuvinte rezultând un dicţionar omografic. În plus, cuvintele monosilabice nu au fost introduse în algoritm, după cum am explicat anterior. În final, T a fost împărţit în trei părţi egale (şi diferite): $T1$, $T2$ şi $T3$, potrivit Tabelului 3: $T1$ a fost folosit pentru antrenarea modelelor n-gramelor, $T2$ a fost folosit ca dicţionar *held-out* pentru estimarea coeficienţilor de ponderare în algoritmul de tip Katz-*backoff*, iar $T3$ a fost folosit pentru testare. Evaluarea a fost făcută în mod automat, prin compararea rezultatelor obţinute în diferite condiţii de testare, pe corpusul $T3$ cu şi fără markerii de accente la intrare.

Corpus	**Denumire**	**Mărime din T(%)**	**Mărime în cuvinte**
Antrenare	$T1$	75	234.700
Held-out	$T2$	10	31.300
Testare	$T3$	15	47.000

Tabelul 3. Corpusurile extrase din dicţionarul T

Ţinând cont de faptul că principiile de funcţionare ale etapelor de antrenare respectiv de testare ale metodei de poziţionare ale accentelor au fost discutate anterior în detaliu, vom adăuga doar două aspecte particulare.

Astfel, în algoritmul de netezire *Katz backoff*, valoarea iniţială propusă de Katz pentru pragul k a fost egală cu 5. În cazul metodei noastre, am testat rezultatele de ieşire pe corpusul $T2$, pentru k = 1,...,20 şi am ales în final k = 10 ca valoare optimă. Pe de altă parte vrem să subliniem ideea amintită anterior, şi anume că o abordare originală propusă de algoritmul nostru este cea de calculare a coeficienţilor de ponderare $\alpha_{held-out}(n)$: **pentru fiecare *n*, cu *n* = 1,...,8 este evaluată valoarea optimă care minimizează erorile de predicţie a accentelor pe corpusul *held-out*.** Această abordare duce la un algoritm mai eficient (adică mult mai rapid), cu o descreştere a acurateţei rezultatelor de numai 0,02% faţă de algoritmul Katz original.

Ca măsură a performanţei metodei propuse, am folosit ***acurateţea la nivel de cuvânt***, definită ca raportul dintre numărul de cuvinte accentuate corect şi numărul total de cuvinte de test (cu alte cuvinte, fiecare cuvânt cu un accent lexical poziţionat incorect este considerat ca

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

fiind eronat). Această măsură de performanţă este relevantă şi consistentă cu definiţia dată în literatura de specialitate, deoarece decizia (la nivel de cuvânt) poate avea doar două variante de ieşire, care se referă la poziţionarea corectă sau incorectă a accentului. Metoda decide din mai multe variante, acestea fiind, după cum am menţionat deja, egale cu numărul de vocale pentru cuvintele polisilabice.

Metoda a fost testată mai întâi cu diferite lungimi ale *n*-gramelor extrase la nivel de caracter, până la lungimea maximă de 8 caractere. Peste această lungime, potrivit datelor experimentale, nu am obţinut o creştere de performanţă. Principalele rezultate sunt ilustrate în Tabelul 4. Se poate observa că acurateţea creşte rapid pentru *n*-gramele de ordin mic şi se saturează pentru cele de ordin mare.

Lungime *n*-grame	**Acurateţe (%)**
2	50,61
3	75,57
4	90,45
5	95,42
6	97,92
7	99,04
8	**99,11**

Tabelul 4. Acurateţea pentru diferite lungimi ale *n*-gramelor

Am obţinut astfel o acurateţe maximă de **99,1%** pentru sarcina de predicţie a accentului lexical, pentru *n*-gramele de lungime 8. Este evident că lungimea 8, aleasă ca ordinul maxim de mărime pentru *n*-grame, este mai mare decât dimensiunea unor cuvinte şi se poate naşte întrebarea modului în care sunt abordate aceste situaţii. Soluţia a fost deja amintită şi constă în folosirea unui algoritm de tip Katz *backoff*, care va reduce recursiv dimensiunea *n*-gramelor în cazul formelor care lipsesc din dicţionarul de antrenare. Un alt aspect care merită amintit este acela că lungimea medie, exprimată în caractere, pentru corpusul de antrenare este de aproximativ 10,3 – valoare destul de apropiată de maximul de 8 pentru care am obţinut cele mai bune rezultate.

De asemenea, pentru o mai bună integrare a metodei cu celelalte componente ale sistemului TTS pentru limba română, am efectuat un **al doilea set de teste**.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Deoarece etajul de predicţie a accentelor lexicale este o parte componentă a modulului de analiză şi generare prozodică, trebuie menţionat faptul că o poziţionare greşită a accentului lexical va duce la generarea unui contur prozodic eronat, ceea ce va conduce în final la producerea unei vorbiri care sună nenatural (sau chiar neinteligibil). Din această cauză, după o analiză a erorilor generate de metoda de predicţie a accentelor lexicale, am decis să tratăm cazurile de cuvinte ambigue ca fiind neaccentuate, deoarece sistemul TTS, în ansamblul său, a dovedit că funcţionează mai bine cu cuvintele neaccentuate decât cu cele având accentul poziţionat eronat.

Am decis să analizăm în profunzime erorile de predicţie pentru cuvintele din corpusul *T*3, analiză care a arătat faptul că pentru aproape 60% dintre cuvintele cu accent incorect poziţionat, probabilităţile de emisie sunt comparabile până la un ordin de mărime. Pentru a aborda aceste cazuri, am propus un algoritm de testare uşor modificat, prin introducerea unui *prag de confidenţă* (nivel de decizie), β. Astfel, algoritmul va lăsa nemodificate cuvintele pentru care nivelul de decizie nu este trecut. Dezavantajul acestui tip de testare cu prag este că eroarea de predicţie raportată va fi puţin mai mare deoarece cuvintele lăsate voit neaccentuate vor fi cuantificate ca fiind eronate.

Am efectuat o serie de experimente cu diferite valori pentru nivelul de prag β, constatând, aşa cum era de aşteptat, că la o creştere a lui β se obţine o creştere a numărului de cuvinte furnizate neaccentuat la ieşirea algoritmului, prin netratarea situaţiilor ambigue, şi rezultând totodată o uşoară descreştere a acurateţei. Se poate pune întrebarea: care ar putea fi cel mai bun criteriu de alegere a nivelului de prag? Deşi răspunsul corect ar putea fi dat numai pe baza unei analize subiective a vorbirii generate (eventual printr-o estimare de tip *Mean Opinion* Score), o soluţie teoretică imediată şi rezonabilă (deşi pot fi alese şi altele) ar fi aceea de oprire a creşterii lui β atunci când numărul de cuvinte corecte, lăsate neaccentuat ca urmare a modificării pragului de decizie, îl egalează pe cel al cuvintelor eronate lăsate neaccentuat deoarece sunt clasificate ca fiind ambigue. În aceste condiţii, am obţinut o valoare de prag $\beta = 30$, pentru o acurateţe de aproximativ 97%. În opinia noastră această valoare nu trebuie tratată ca fiind prea mică deoarece ieşirea sistemului va fi foarte echilibrată, multe dintre cazurile de erori de predicţie fiind de fapt clasificate ca fiind ambigue şi lăsate neaccentuat. Rezultatele testelor sunt prezentate în Tabelul 5 .

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

β	Acurateţe(%)
1	99,11
1,5	98,71
2	98,32
3	97,99
5	97,76
10	97,47
15	97,27
20	97,12
30	**96,92**
40	96,74
50	96,60

Tabelul 5. Rezultatele testării după varierea pragului de decizie

În acest punct al experimentelor, prin analiza erorilor obţinute din corpusul *T*3, am încercat să realizăm o clasificare a acestora în scopul găsirii unor soluţii de reducere a acestor erori.

Am putut identifica patru tipuri importante de erori:

1. Erori de poziţionare a accentului lexical, cauzate de metodă. Reprezintă aproape 97,5% din numărul total de erori (0,86%).

Exemplu: **căm'ăşi** a fost accentuat incorect ca fiind **cămăş'i**.

Astfel de situaţii sunt cauzate de existenţa în corpusul de antrenare a unui spectru larg de *n*-grame asemănătoare, provenind de regulă de la diverse categorii gramaticale (în exemplul menţionat de la substantive şi verbe).

Am efectuat, totodată, o analiză a modului de variaţie a acurateţei metodei cu dimensiunea corpusului de antrenare. Concluzia a fost că acurateţea creşte odată cu mărimea corpusului de antrenare, după cum rezultă din Tabelul 6. Rezultatele au fost obţinute folosind ca bază de test corpusul *T*3, iar ca bază de antrenare diferite segmente extrase din *T*1, în mod aleator.

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

Dimensiune *T1* (cuvinte)	Acurateţe (%)
10.161	91,61
54.818	94,42
99.526	95,53
144.234	96,62
188.942	97,37
234.700	**99,11**

Tabelul 6. Creşterea acurateţei cu variaţia dimensiunii corpusului de antrenare

2. O decizie incorectă referitoare la cuvintele monosilabice inserată de modulul de despărţire în silabe. Aceste erori însumează aproximativ 1,7% din numărul total de erori (0,015% din totalul cuvintelor).

Exemplu: **urci** a fost introdus în algoritm, deşi are o singură silabă, fiind incorect despărţit în silabe (adică **ur-ci)**, la ieşirea algoritmului fiind furnizată varianta **urc'i**.

3. Forme omografe incorecte introduse în corpusul de test. Aceste erori totalizează aproximativ 0,4% din numărul total de erori (ceea ce înseamnă 0,004% din totalul cuvintelor).

Exemplu: **p'oza** nu a fost găsit în dicţionarul de forme omografe şi a fost accentuat ca fiind **poz'a** . Aceste tipuri de erori sunt provocate de faptul că dicţionarul de forme omografe este deocamdată incomplet.

4. Introducerea manuală incorectă a formelor accentuate în corpusul de test

Exemplu: Cuvântul **biciuie** a fost introdus în corpusul de test ca **bici'uie** (cuvânt lipsit de semnificaţie). Algoritmul a inserat corect markerul corespunzător accentului adică **b'iciuie** însă datorită comparaţiei cu cuvântul din baza de test, a fost raportat eronat ca fiind o eroare. Aceste tipuri de erori apar de multe ori în corpusurile mari adnotate manual, şi duc la o mică scădere a performanţelor (în cazul nostru scăderea a fost de 0,3% (0,0024% din totalul cuvintelor).

Lucrările au fost realizate în cadrul contractului
de cercetare UPB-CNCSIS, IDEI-782

RECTOR,
Prof. dr. ing. Ecaterina Andronescu

REVENDICĂRI

Metodă de predicţie a accentului lexical în sinteza vorbirii pornind de la text în limba română, caracterizată prin aceea că ajută la modelarea corectă a conturului intonaţional pentru sinteza automată a vorbirii de bună calitate prin poziţionarea corectă a accentelor lexicale la nivelul cuvintelor, asigurând astfel o prozodie corespunzătoare vorbirii generate.

Metoda utilizează o strategie statistică, bazată în principal pe *n*-gramele construite la nivel de caracter, fiind destinată poziţionării automate a accentelor lexicale pentru cuvintele scrise în limba română; abordarea statistică a fost aleasă ţinând cont de faptul că poziţia accentului lexical în limba română este variabilă, fiind practic imposibil de prezis pe bază de reguli. În vederea rezolvării problemelor generate de insuficienţa datelor de antrenare, metoda foloseşte un algoritm de netezire de tip *Katz back-off* modificat, după următorul principiu: din fiecare variantă posibilă se extrag *n*-gramele de ordin n, n–1,…,1 şi se calculează emisiile de probabilitate finale, probabilităţile condiţionate fiind cele obţinute în timpul etapei de antrenare. În plus, în vederea reducerii efectelor nedorite cauzate de poziţionarea incorectă a accentului lexical în vorbirea sintetizată, se utilizează un prag de decizie variabil. Cuvintele monosilabice sunt considerate ca fiind fără accent lexical şi sunt puse în evidenţă cu ajutorul unui algoritm de despărţire automată în silabe. Metoda necesită cunoştinţe lingvistice limitate, este bazată pe un corpus de antrenare de dimensiuni medii şi este foarte precisă în predicţia poziţiei accentului lexical în limba română.

RECTOR,
Prof. dr. ing. Ecaterina Andronescu