Liet. matem. rink., T. 43, spec. nr., 292–296

Žodžių atpažinimo sistemos kūrimas

Gintautas TAMULEVIČIUS, Antanas LIPEIKA (MII)
el. paštas: lipeika@ktl.mii.lt

1. Įvadas

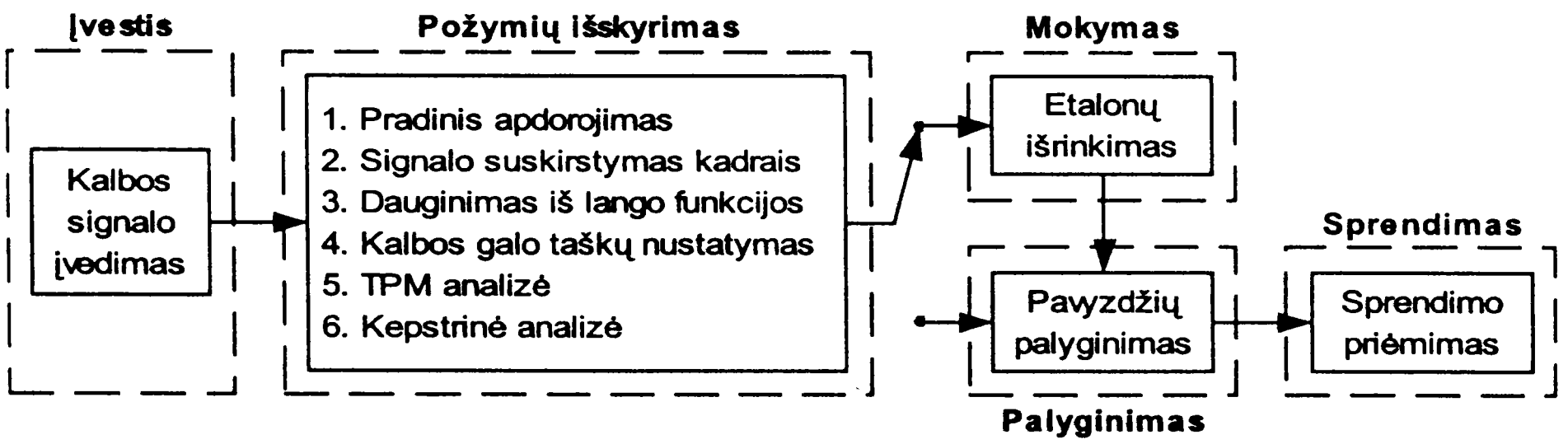
Kalbos atpažinimas – procesas, kurio metu aparatine ar programine įranga iš kalbos signalo nustatoma lingvistinė informacija. Atpažinimo objektu gali būti pavieniai žodžiai arba ištisinė kalba. Vienu iš labiausiai paplitusių pavienių žodžių atpažinimo metodu – pavyzdžių palyginimu – nagrinėjamasis kalbos signalo pavyzdys lyginamas su visais turimais žodyno etaloniniais pavyzdžiais. Tam kalbos pavyzdžiai atvaizduojami spektrinių požymių vektorių sekomis.

2. Požymių išskyrimas

Remiantis teorija sukurta pavienių žodžių atpažinimo sistema *Atpažinimas*. Kalbos pavyzdžių atvaizdavimui panaudoti tiesinės prognozės modelio (TPM) koeficientai, kepstro ir centruoti kepstro koeficientai. Sistema sudaryta iš penkių posistemių: įvesties, požymių išskyrimo, mokymo, palyginimo ir sprendimo (1 pav.). Trumpai aptarsime jas.

Pradinis apdorojimas. Kalbos signalo energija yra susikoncentravusi žemų dažnių srityje. Toks energijos netolygumas skaičiavimuose sukelia paklaidas, todėl tenka spektrą „išlyginti“. Tam tikslui sistemoje naudojamas pirmos eilės aukštų dažnių filtras su keičiamu pradinio apdorojimo koeficientu. Po pradinio apdorojimo kalbos signalas yra suskaidomas kadrais ir tolimesnis apdorojimas vykdomas kadruose.

Dauginimas iš lango funkcijos. Suskirsčius signalą kadrais, dėl trūkių kadro galuose gaunasi dideli spektro iškraipymai. Siekiant jų išvengti signalą kadre tenka dauginti iš lango funkcijos. Jeigu TPM parametrų vertinimui yra naudojamas autokoreliacinis metodas, kaip lango funkcija dažniausiai naudojamas Hamming'o langas [3].



1 pav. Atpažinimo sistemos struktūra.

Kalbos galo taškų nustatymas. Ištarimo ribų nustatymas leidžia pašalinti foninio triukšmo sritis prieš ištarimą ir po. Programoje panaudoti du galo taškų nustatymo metodai. Pirmuoju jų galo taškais laikomos tos signalo vietos, kuriose energijos reikšmė viršija pasirinktą slenkstį. Slenksčiu laikomas dydis lygus didžiausiai dispersijos kadre reikšmei, padaugintai iš slenksčio koeficiento (pasirenkamo nustatymuose). Kitas mūsų pasiūlytas metodas, realizuotas dinaminiu programavimu – grindžiamas atsitiktinių sekų savybių pasikeitimo momentų nustatymu. Jame tariama, kad kadrų energijos reikšmės yra nepriklausomi atsitiktiniai dydžiai. Tuomet galima užrašyti [2]

$$A(n) = \begin{cases} A_1 = N(\mu_1, \sigma_1^2), & n = 0, 1, \ldots, u_1, \\ A_2 = N(\mu_2, \sigma_2^2), & n = u_1 + 1, \ldots, u_2, \\ A_3 = N(\mu_3, \sigma_3^2), & n = u_2 + 1, \ldots, N, \end{cases} \tag{1}$$

kur μ_i, σ_i^2 yra signalo energijos vidurkiai ir dispersijos. A_1 ir A_3 nurodo foninio triukšmo prieš ir po žodžio energijos parametrus, A_2 – žodžio energijos parametrus. $u = [u_1, u_2]$ – šuoliško parametrų pasikeitimo momentai. Juos galima rasti maksimizavus tikėtinumo funkcijos logaritmą

$$\hat{u} = \arg \max_{u} \log p(x|u). \tag{2}$$

Maksimizuoti ją sudėtinga dėl didelės skaičiavimų apimties. Tačiau yra įrodyta, kad jos maksimumo vieta sutampa su tikslo funkcijos $\Theta(u|x) = L_1(u_1|x) + L_2(u_2|x)$ maksimumu. Pastarosios dėmenis galima išreikšti rekurentinio skaičiavimo forma [2]

$$L_i(k|x) = L_i(k-1|x) - \log \sigma_i + \log \sigma_{i+1} - \frac{1}{2\sigma_i^2}[x(n) - \mu_i]^2 + \frac{1}{2\sigma_{i+1}^2}[x(n) - \mu_{i+1}]^2, \quad i = 1, 2; \;\; k = 2, \ldots, N. \tag{3}$$

$\Theta(u|x)$ maksimizuoti galima naudoti dinaminį programavimą. Tam tikslui skaičiuojamos Bellman'o funkcijos [2]

$$g_1(u_2|x) = \max_{\substack{u_1 \\ p<u_1<u_2}} L_1(u_1|x), \quad u_2 = p+2, \ldots, N; \tag{4}$$

$$g_2(u_3|x) = \max_{\substack{u_2 \\ p+1<u_2<u_3}} \left[L_2(u_2|x) + g_1(u_2|x)\right], \quad u_3 = p+3, \ldots, N. \tag{5}$$

Pasikeitimo momentai randami iš Bellman'o funkcijų

$$\hat{u}_k = \min \left[\arg \max_{\substack{n \\ p+k \leqslant n \leqslant \hat{u}_k + 1}} g_k(n|x) \right], \quad k = 1, 2, \tag{6}$$

kur $\hat{u}_3 = N$.

Tiesinės prognozės modelio analizė. Šiame etape išskiriami tiesinės prognozės koeficientai. Jiems išskirti panaudotas autokoreliacinis metodas, realizuotas rekurentiniu Levinsono–Durbino algoritmu [3].

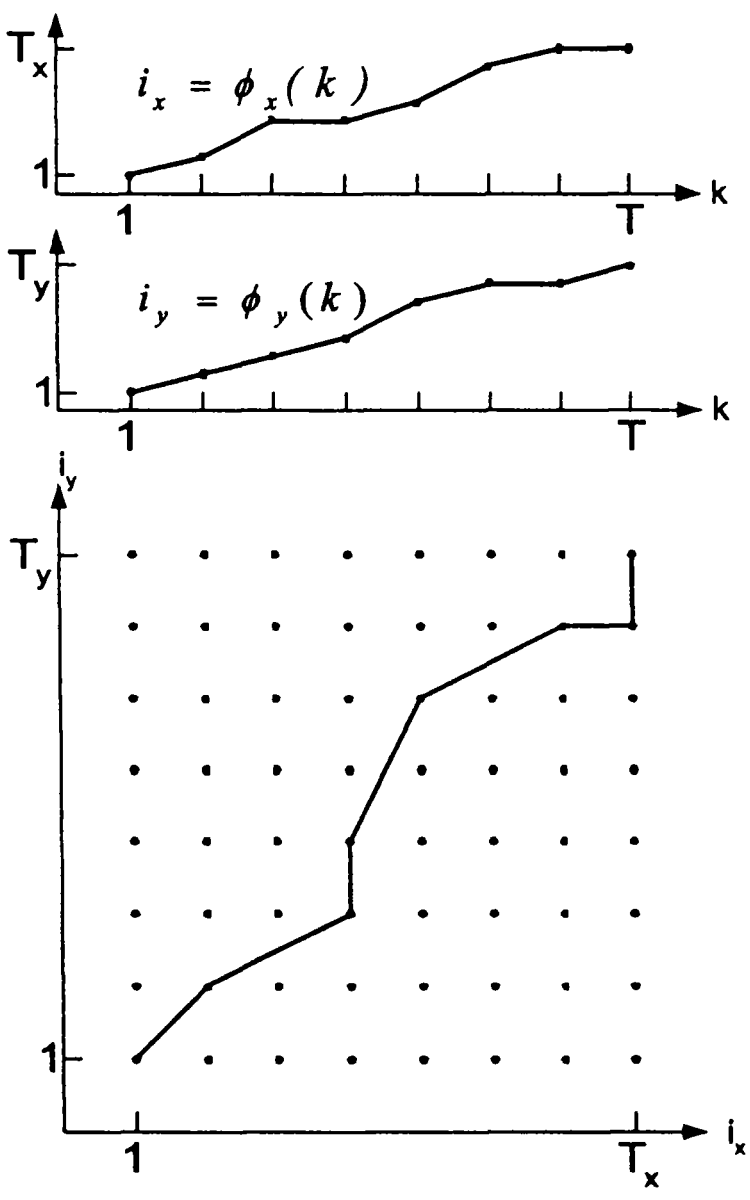
Kepstrinė analizė. Kepstro koeficientai apskaičiuojami iš tiesinės prognozės koeficientų panaudojus rekurentines formules. Siekiant padidinti programos eksperimentinį lankstumą, kepstrinės analizės eilė, t.y., elementų skaičius požymių vektoriuje padarytas keičiamu. Požymių kokybei gerinti naudojamas centravimas, t.y., iš kiekvieno požymių vektoriaus elemento atimamas to elemento vidurkis viso ištarimo atžvilgiu. Tai kompensuoja koeficientų atsitiktinę dedamąją, atsirandančią dėl įvairių garso įvedimo kanalų skirtingų savybių.

3. Palyginimas

Naudojant požymius, atsiranda galimybė išreikšti panašumą tarp kalbos signalo pavyzdžių. Vietinis atstumas nusako dviejų požymių vektorių panašumą, suminis (susumuoti vietiniai) – pavyzdžių panašumą. Kyla klausimas, tarp kurių požymių vektorių skaičiuoti atstumus, kadangi tie patys žodžiai kiekvieną kartą ištariami vis kitokiu greičiu. Kadangi netgi tie patys garsai skirtinguose žodžio ištarimuose yra skirtingos trukmės, tenka naudoti kraipymo funkcijas, susiejančias pavyzdžių kadrų numerius su nauja, normuota laiko skale k (2 pav.).

Tuomet pavyzdžių panašumas, panaudojus kraipymo funkcijas, gali būti išreikštas [3]:

$$d_\phi(X,Y) = \sum d\left(\phi_x(k), \phi_y(k)\right) m(k)/M_\phi, \tag{7}$$



2 pav. Dinaminis laiko skalės kraipymas.

čia X, Y – nagrinėjamasis ir etaloninis ištarimai, T – nagrinėjamojo ištarimo trukmė normuotoje laiko skalėje, $\phi_x(k), \phi_y(k)$ – kraipymo funkcijos, $m(k)$ – krypties svorio koeficientas, M_ϕ – normuojantis koeficientas.

Dinaminiam laiko skalės kraipymui realizuoti naudojamas dinaminis programavimas. Tuomet skaičiavimo eiga pradedant tašku (1, 1) ir baigiant (T_X, T_Y) atrodo taip [1]:

1. Pradinis nustatymas
$$D(1,1) = D_v(1,1). \tag{8}$$
2. Rekursija
$$D(i_x, i_y) = D_v(i_x, i_y) + \min_{p(i,j)} D[p(i_x, i_y)], \quad 1 < i_x < T_x, \ 1 < i_y < T_y. \tag{9}$$
3. Užbaigimas
$$d(X,Y) = D(T_x, T_y)/T_x. \tag{10}$$

$D_v(i_x, i_y)$ – dalinis atstumas, $D(i_x, i_y)$ – suminis atstumas, $d(X,Y)$ – suvidurkintas suminis atstumas, $p(i_x, i_y)$ – trajektorija iki taško (i_x, i_y).

Turint lyginamojo pavyzdžio atstumus iki visų etalonų, sprendžiama, kuris jų labiausiai atitinka lyginamąjį. Parenkamas etalonas, su kuriuo atstumas yra mažiausias. Jeigu šis atstumas viršija iš anksto pasirinktą slenksčio reikšmę, laikoma, kad ištarimas neatpažintas.

4. Mokymas

Mokymo tikslas – sukurti etalonus. Mokymui mes naudojome artimiausio kaimyno principą. Iš keleto mokomųjų pavyzdžių išrenkami turintys mažiausią vidutinį atstumą su kitais. Iš pradžių skaičiavimai atliekami su visomis įmanomomis kandidatų į vieną etaloną kombinacijomis, paskui su visais kandidatų į etalonus dvejetais, trejetais ir t.t. Analitiškai tai išreiškiama

$$\begin{aligned} I_m &= \left\{\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_m\right\} \\ &= \arg \min_{\substack{i_1,\ldots i_m \\ i_i \neq i_2 \neq \ldots \neq i_m}} \left[\frac{1}{N-m} \sum_{\substack{k=1 \\ k \neq i_1, i_2, \ldots, i_m}}^{N} \min\left\{d_{i_1 k}; d_{i_2 k}; \ldots; d_{i_m k}\right\}\right], \\ m &= 1, 2, \ldots, M, \end{aligned} \tag{11}$$

čia $\hat{i}_m$ – etalonu paskelbtas pavyzdys, N – mokomųjų pavyzdžių skaičius, m – etalonų skaičius, $d_{i_m,k}$ – atstumas tarp i_m-ojo ir k-ojo pavyzdžių.

5. Eksperimentai

Sukurtąja sistema atlikti priklausomo nuo kalbėtojo atpažinimo eksperimentai klaidų kiekiui stebėti. Eksperimentams parametrai pasirinkti tokie: pradinio apdorojimo koeficientas – 0,95, analizės kadro ilgis – 250 atskaitų, kadro žingsnis – 125 (signalo diskretizavimo dažnis – 11025 Hz), atpažinimc slenkstis – 1, kepstrinės analizės eilė – 15. Žodžio

1 lentelė. Eksperimento rezultatai

Analizės tipas	Klaidų kiekis, proc.							
	I eksperimentas				II eksperimentas			
	Be mokymo		Su mokymu		Be mokymo		Su mokymu	
	DP	Slenkstis	DP	Slenkstis	DP	Slenkstis	DP	Slenkstis
TPM	10	13,3	6	8	28,7	50	4	27,3
Kepstrinė	8	5,3	4,7	3,3	12	38	0,6	23,3
Centruota kepstrinė	8	9,3	4,7	4,7	13,3	40	2	24,7

ribos nustatinėtos dviem metodais: naudojant energijos slenkstį ir atsitiktinių sekų pasikeitimo momentų radimo metodą (lentelėje jie pavadinti Slenkstis ir DP atitinkamai). Etalonų žodyną sudarė 50 žodžių. Testinę aibę sudarė trys kiekvieno iš 50 žodžių ištarimai. Atpažinimo eksperimentai palyginimui atlikti du kartus – su skirtingų žmonių ištarimais, įrašytais skirtinguose kompiuteriuose. Atpažinimo rezultatai pateikti 1 lentelėje.

Kaip matyti iš 1 lentelės, mokymas ir DP panaudojimas žodžio riboms nustatyti leido sumažinti klaidų skaičių. Didžiausią tikslumą duoda kepstrinė analizė, nuo jos nedaug atsilieka centruotoji kepstrinė analizė. Skirtumai tarp eksperimentų rezultatų atsirado dėl skirtingų garso įvedimo kanalų bei nevienodai tikslaus energijos slenksčio parinkimo.

6. Rezultatų apibendrinimas

Remiantis analitinėmis dinaminio laiko skalės kraipymo metodo išraiškomis sukurta atskirų žodžių atpažinimo sistema *Atpažinimas*. Tai programa, skirta pavienių žodžiams atpažinimui stebėti. Sistemoje panaudoti originalūs žodžio ribų nustatymo ir mokymo metodai, leidžiantys padidinti atpažinimo tikslumą. Atlikus žodžių atpažinimo tyrimus nustatyta:

- tikslus žodžių ribų aptikimas turi lemiamą įtaką atpažinimo tikslumui;
- mokymas ir kepstro požymių naudojimas padidina atpažinimo sistemos tikslumą.

Literatūra

[1] B. Gold, Morgan, *Speech and Audio Signal Processing*, John Wiley & Sons Inc. (2000).
[2] A. Lipeika, J. Lipeikienė, Žodžių pradžios ir galo taškų nustatymas atskirai sakomų žodžių atpažinime, in Tarptautinė konferencija „*Biomedicininė inžinerija*“, Kaunas (2002), pp. 178–181.
[3] L.R. Rabiner, B.H. Juang, *Fundamentals of Speech Recognition*, Prentice–Hall (1993).

Isolated word recognition system development

G. Tamulevičius, A. Lipeika

Isolated word recognition system, based on dynamic time warping, was developed. Speech patterns are represented by LPC, cepstral and weighted cepstral coefficients. Two endpoint detection, two speech input and two pattern training methods are implemented in the system. The system performance was investigated.