



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

Gintarė Paškauskaitė

**BALSU IŠTARTŲ KODŲ ATPAŽINIMO SISTEMOS
SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Doc. dr. Kastytis Ratkevičius

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA

BALSU IŠTARTŲ KODŲ ATPAŽINIMO SISTEMOS
SUKŪRIMAS IR TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

Doc. dr. Kastytis Ratkevičius
2015-05-27

Recenzentas

Doc. dr.

Projektą atliko

Gintarė Paškauskaitė
2015-05-27

KAUNAS, 2015

G. Paškauskaitė. Balsu išstartų kodų atpažinimo sistemos sukūrimas ir tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Kastytis Ratkevičius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Kaunas, 2015. 154 psl.

SANTRAUKA

Šnekamoji kalba žmonėms yra kasdieninio bendravimo priemonė. Sparčiai tobulėjant technologijoms ir joms užimant vis svarbesnę vietą kasdieninėje žmonių veikloje, tampa labai aktualu pritaikyti technologijas taip, kad jų valdymas būtų įmanomas žmonėms pačiu priimtinausiu būdu – balsu. Todėl pagrindinis automatinio šnekos atpažinimo sistemų kūrimo ir tobulinimo tikslas yra sukurti technologijas, kurios galėtų girdėti, suprasti, kalbėti ir veikti pagal balsu gautą informaciją. Šioje srityje jau keliasdešimt metų yra atliekami tyrimai tiek užsienyje tiek Lietuvoje. Vienas iš Kauno technologijos universiteto mokslininkų 2012-2013 metais vykdytų projektų yra - „Hibridinė atpažinimo technologija balso sąsajai“. Baigiamasis magistro darbas yra tiesiogiai susijęs su šiuo projekto metu atliktais tyrimais ir gautais rezultatais.

Atikus išsamią literatūros analizę apimančią šnekamosios kalbos sampratą, atpažinimo sistemas ir taikomus metodus kalbos atpažinimo modeliavime, buvo atliekami balso komandų atpažinimo tyrimai naudojant kitakalbį ir lietuvišką, paremtą paslėptaisiais Markovo modeliais, atpažintuvus. Šnekamosios kalbos tyrimai atlikti naudojant paslėptiesiems Markovo modeliams taikomus metodus, kitakalbio atpažintuvo adaptavimo kitai kalbai atpažinti metodus, duomenų analizės sistemą WEKA, įvairius duomenų klasifikatorius ir kt.

Šiame darbe pateikiama „Balsu išstartų kodų atpažinimo sistema“, atpažįstanti diktuojamas balso komandas. Demonstracinė programa parašyta C# programavimo kalba. Taip pat atlikti balso komandų atpažinimo tikslumo tyrimai. Gauti tyrimu rezultatai išanalizuoti ir pateikiamos išvados.

Reikšminiai žodžiai (iki 8 žodžių): „Hibridinė atpažinimo technologija balso sąsajai“, paslėptieji Markovo modeliai, kitakalbis atpažintuvas, WEKA.

Paškauskaitė Gintarė. Development and Investigation of Voice-Based Codes Recognition. Final project of *Master's degree* / supervisor doc. dr. Kastytis Ratkevičius; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.

Kaunas, 2015. 154 pg.

SUMMARY

Colloquial language are common communication tool for people. Technology is improved and occupied an increasingly prominent place in people's daily activities rapidly, therefore it is very important to improve technologies control by the easiest way for people - by voice. Therefore, the main purpose of automatic speech recognition systems design and development is to develop technologies which could hear, understand, speak and act according to the information received by voice. Technology research is being carried out for years both in Lithuania and abroad. Kaunas University of Technology scientists implemented the project - "Hybrid recognition technology for voice interface" in 2012-2013. Master thesis is directly associated with researches and results of this project.

There was accomplished detailed analysis of the literature covering the colloquial language, systems of voice recognition, speech recognition modelling methods. Researches of speech recognition was accomplished by using other language recognition system and Lithuanian recognition system. There was applied various methods for researches of speech recognition – Hidden Markov Models, Data mining system WEKA, various data mining classificatory and other.

This paper presents the "Voice-Based Codes Recognition system", which recognizes dictated voice commands. Demonstration program is written in C # programming language. Also there was accomplished voice commands recognition accuracy tests. The obtained results are analysed with conclusions.

Keywords (up to 8 words): " Hybrid recognition technology for voice interface", Hidden Markov Models, other language recognizer, WEKA.

TURINYS

SANTRAUKA	3
ĮVADAS	7
1. KALBOS ATPAŽINIMO SISTEMŲ APŽVALGA.....	9
1.2 Automatinis šnekamosios kalbos atpažinimas	9
1.2.1 Šnekamosios kalbos samprata	10
1.2.2 Šnekamosios kalbos atpažinimo sistemos ir jų klasifikavimas	13
1.2.3 Markovo modelių taikymas šnekos atpažinimui	16
1.2.4 Šnekamosios kalbos analizė ir požymių išskyrimas.....	18
1.2.5 Žodžių modeliais grįstas šnekamosios kalbos atpažinimas.....	19
1.2.6 Fonemų modeliais grįstas šnekamosios kalbos atpažinimas	21
1.3 Hidden Markov Model Toolkit programinis paketas	22
1.4 Garsynai.....	24
1.5 Projekto „Hibridinė atpažinimo technologija balso sąsajai“ apžvalga	25
1.6 Kitakalbio atpažintuvo panaudojimo principai	28
1.7 Informacijos kodavimas	30
1.8 Duomenų analizės sistemų apžvalga	31
2. METODINĖ DALIS	32
2.1 Garsynų paruošimas atpažinimo tikslumo tyrimams.....	32
2.2 Gramatikų paruošimas atpažinimo tikslumo tyrimams.....	33
2.3 Kitakalbis atpažintuvas ir jo panaudojimas	35
2.3.1 Kitakalbio atpažintuvo panaudojimas vardų balso komandų atpažinimui	38
2.3.2 Kitakalbio atpažintuvo panaudojimas skaičių balso komandų atpažinimui.....	41
2.4 Hidden Markov Model Toolkit įsisavinimas.....	42
2.4.1 HTK panaudojimas šnekos atpažinimui taikant žodžių modeliais grįstus PMM.....	43
2.4.1.1 Požymių tipų įtakos atpažinimo tikslumui tyrimo metodika.....	44
2.4.1.2 Vardų atpažinimo tyrimo, naudojant žodžiais gristus modelius, metodika.....	45
2.4.1.3 Skaičių atpažinimo tyrimo, naudojant žodžiais gristus modelius, metodika.....	46
2.4.2 HTK panaudojimas šnekos atpažinimui taikant fonemų modeliais grįstus PMM	47
2.4.2.1 Skaičių atpažinimo tyrimo, naudojant fonemomis gristus modelius, metodika.....	49
2.5 Šnekos atpažintuvų sujungimo galimybių tyrimo metodika	49
3. ATLIKTŲ TYRIMŲ REZULTATŲ DALIS.....	52
3.1 Vardų atpažinimo tyrimas su ispanišku atpažintuvu	52
3.2 Skaičių atpažinimo tyrimas su ispanišku atpažintuvu	56
3.3 Kalbos signalo apdorojimo požymių tyrimas	58

3.4 Vardų atpažinimo tyrimas naudojant žodžiais gristus modelius	59
3.5 Skaičių atpažinimo tyrimas naudojant žodžiais gristus modelius	63
3.6 Skaičių atpažinimo tyrimas naudojant fonemomis gristus modelius	66
3.7 Atliktų tyrimų rezultatų palyginamoji analizė.....	67
3.8 Šnekos atpažintuvų sujungimo galimybių tyrimas.....	69
3.9 Ligų kodų atpažinimo demonstracinė programa	72
IŠVADOS IR REZULTATAI.....	73
LITERATŪROS ŠALTINIAI	74
1 PRIEDAS. Garsynai	77
2 PRIEDAS. Gramatikos.....	79
3 PRIEDAS. UPS alfabeto fonemų lentelės ispanų kalbai.....	85
4 PRIEDAS. Vardų atpažinimo tikslumo priklausomybė nuo būsenų ir Gauso mišinių skaičiaus	87
5 PRIEDAS. 2.4 Hidden Markov Model Toolkit detalus įsisavinimas.....	90
6 PRIEDAS. Fonemų deriniai	115
7 PRIEDAS. Atpažintuvų sujungimas	116
8 PRIEDAS. TLK-10-AM klasifikavimas	118
9 PRIEDAS. Komandų modelių pavyzdžiai su skirtingais būsenų ir Gauso mišinių skaičiumi	119
10 PRIEDAS. Vardų komandų atpažinimo tikslumas gautas su ispanišku atpažintuvu.....	131
11 PRIEDAS. Skaičių atpažinimo tikslumo priklausomybė nuo būsenų ir Gauso mišinių skaičiaus	134
12 PRIEDAS. „Balsu išstartų kodų atpažinimo sistema“	136
13 PRIEDAS. Konferencija ECT 2014	145
14 PRIEDAS. Konferencija E2TA 2015.....	151
15 PRIEDAS. Baigiamasis magistro darbas (elektroninė versija)	156

IVADAS

Šnekamoji kalba žmonėms yra kasdieninio bendravimo priemonė. Sparčiai tobulėjant technologijoms ir joms užimant vis svarbesnę vietą kasdieninėje žmonių veikloje, tampa labai aktualu pritaikyti technologijas taip, kad jų valdymas būtų įmanomas žmonėms pačiu priimtinausiu būdu – balsu. Todėl pagrindinis automatinio šnekos atpažinimo sistemų kūrimo ir tobulinimo tikslas yra sukurti technologijas, kurios galėtų girdėti, suprasti, kalbėti ir veikti pagal balsu gautą informaciją.

Pastaruoju metu kalbos atpažinimo technologijos yra plačiai taikomos informacinėse technologijose. Todėl kalbos atpažinimo pritaikymas informacinėse technologijose yra viena iš labiausiai tyrinėjamų sričių šiandien. Šnekamosios kalbos atpažinimo sistemos taikomos įvairiose srityse: automobilių pramonėje – laisvų rankų įranga, navigacijos bei multimedijos prietaisų valdymas balsu, mobiliuosiuose telefonuose (Siri, Nina) ir dar daugelyje kitų sričių. Sėkmingi tyrimai šnekos atpažinimo srityje reikalauja didelių finansinių išteklių ir didelės apimties duomenų kiekio apimančio sudėtingas balso komandas. Žinoma kompanija „Google“ sėkmingai vykdo tyrimus šnekos atpažinimo srityje, kurie yra paremti paslėptaisiais Markovo modeliais, ir stebina savo rezultatais net ir neplačiai vartojamų kalbų atpažinime. „Google“ sėkmę šnekamosios kalbos atpažinime nulemia surinkti didelės apimties garsynai, naudojami atpažinimo sistemų apmokymui.

Darbo tikslas – Balsu išstartų kodų atpažinimo sistemos sukūrimas ir tyrimas.

Darbo objektas – Skaičių ir vardų komandų daugiadiktoriniai garsynai, projekte „Hibridinė atpažinimo technologija“ nuodoti tyrimų duomenys.

Darbo tyrimų metodai: eksperimentinis ir kiekybinės analizės metodai. Komandų atpažinimui naudojami - ispanų kalbos atpažintuvas *Microsoft Speech Recognizer 8.0*, dirbant su *Microsoft Windows 7 Ultimate* operacine sistema ir HTK programinių įrankių rinkinys. Atpažinimo tikslumo tyrimų metu buvo taikyti įvairūs metodai – balso komandų transkripcijų korekcija naudojant kitakalbio sintezatoriaus sugeneruotas ir UPS transkripcijas, naudojant HTK programinį įrankį taikomi fonemomis ir žodžių modeliais grįsti metodai, atpažintuvų sujungimo galimybių analizė atliekama naudojant duomenis apibudinančių požymių atranką, duomenų analizės sistemą WEKA ir joje taikomus klasifikatorius.

Darbo aktualumas ir naujumas.

Darbo tema susijusi KTU universiteto mokslininkų 2012-2013 metais vykdytu projektu - „Hibridinė atpažinimo technologija balso sąsajai“. Kurio metu buvo paruošta hibridinė balso komandų atpažinimo sistema medicininio garsyno pagrindu.

Baigiamojo magistro darbo užduotis - „Balsu išstartų kodų atpažinimo sistemos sukūrimas ir tyrimas“ yra suformuluota spręsti iškilusiai problemai projekto „Hibridinė atpažinimo technologija balso sąsajai“ metu. Problema kilo sudarinėjant medicininį garsyną, kurio vienas iš

segmentų yra „Ligų“ garsynas. Ieškant dažniausiai žmones Lietuvoje kankinančių ligų pavadinimų, buvo išsiaiškinta, kad iš viso yra 15'000 ligų pavadinimų, pateiktų tarptautinės statistinės ligų ir sveikatos sutrikimų klasifikacijos leidime (TLK-10-AM), o sukurtoji projekto metu sistema geba atpažinti tik 217 ligų pavadinimų. Projekto metu sukurtai sistemai pritaikyti atpažinti tokią apimtį ligų pavadinimų, reikėtų labai gausaus medicininio garsyno, kurio paruošimas brangiai kainuotų ir labai ilgai užtruktų. Todėl, ieškant išeities kaip supaprastinti ligų atpažinimą ir pritaikyti praktikoje buvo pastebėta, kad dėl patogumo ir praktiškumo vartotojai nuo 1992 m. (gydytojai) pildant formas dažniausiai naudoja ne pilnus ligų pavadinimus, bet jų kodus, kurie yra aprašyti TLK - 10 - AM. Dėl šios priežasties iškelti ligų kodų pritaikymo praktikoje uždaviniai.

Ligos yra koduojamos raidine – skaitine kodų sistema, apimančia ligų klasifikacijas. Todėl baigiamojo magistrinio darbo metu buvo atliekami balso komandų atpažinimo tyrimai pasitelkiant įvairius praktikoje naudojamus metodus skaičių ir vardų, kurie reprezentuotų 26 koduose vyraujančias raides, garsynų pagrindu. Balsu ištartų kodų atpažinimo sistemos kūrimas taip pat yra aktualus atsižvelgiant ir į tai, kad didžioji dalis visuomenei svarbios informacijos (pašto kodai, automobilių registracijos numeriai ir kt.) yra koduojama tam tikrais skaičių ar raidžių ir skaičių deriniais, taip supaprastinant informacijos paiešką duomenų archyvuose.

Baigiamajame magistriniame darbe komandų atpažinimo tikslumas buvo tiriamas naudojant du atpažintuvus – ispanų ir lietuvių kalbų, remiantis projekto metu paruoštos hibridinės technologijos principu – kad du atpažintuvai dirba vienu metu ir papildo vienas kitą. Dėl šios priežasties buvo aktualu panagrinėti dviejų atpažintuvų sujungimo galimybes. Pirminiai atpažintuvų sujungimo galimybių tyrimai taip pat pateikiami baigiamajame darbe.

Darbo struktūra. Darbas susideda iš įvado, teorinės dalies, metodinės dalies, tiriamosios dalies, išvadų, literatūros sąrašo ir 15 priedų (vienas jų pateikiamas elektronine versija).

Teorinėje dalyje aprašomos šnekamosios kalbos samprata, atpažinimo sistemos, jų pritaikymo metodai. Metodinėje dalyje išsamiai pateikiama pagrindinė informacija apie balso komandų atpažinimo tyrimuose naudojamus metodus, tyrimų eigą, atliktus eksperimentus ir testavimus. Tiriamojoje dalyje tyrimų metu gauti rezultatai pateikiami lentelių ir diagramų pavidalu, aprašyta detali rezultatų analizė ir bandomosios kodų atpažinimo sistemos kūrimo procesas. Išvadose pateikiamas tyrimų ir gautų rezultatų apibendrinimas.

1. KALBOS ATPAŽINIMO SISTEMŲ APŽVALGA

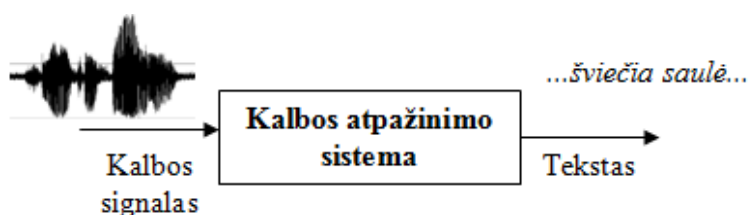
Maždaug per 15 pastarųjų metų šnekamosios kalbos technologijos (atpažinimas, sintezė, asmens patikra pagal balsą) tapo pramoninio informacinių technologijų panaudojimo objektu daugelyje žmogaus veiklos sričių, kurios vis plečiasi ir tobulėja. Todėl technologijų valdymas balsu yra labai patogus jų valdymo būdas, bet informacinėms technologijoms tai yra vis dar nelengvas uždavinys.

1.2 Automatinis šnekamosios kalbos atpažinimas

Kalbos suvokimas žmonėms atrodo natūralus reiškinys, tačiau kompiuteriui tai nėra taip paprasta. Pagrindinis automatinio šnekos atpažinimo sistemų kūrimo ir tobulinimo tikslas yra sukurti mašinas, kurios galėtų girdėti, suprasti, kalbėti ir veikti pagal balsu gautą informaciją. Įprastinės darbo su kompiuteriu priemonės, kaip klaviatūra ir pelė, būtų papildytos (ar netgi pakeistos) dar vienu komunikavimo su kompiuteriu būdu – šnekamąja kalba.[4]

Automatinis kalbos atpažinimas tai garsinio kalbos signalo vertimą į tekstą arba kitas analizuojamas kalbos signalus atitinkančias formas. Automatinių šnekos atpažinimo sistemų kūrimo ir tobulinimo pagrindinis tikslas yra sukurti technologijas, kurios galėtų suprasti žmonių šneką, kalbėti su žmonėmis jų šneka ir veikti pagal balsu gautą informaciją. Automatinė šnekos atpažinimo sistema (AŠA) – tai programinės ir aparatinės įrangos kompleksas, automatiškai atpažįstantis šnekamąją kalbą. [3]

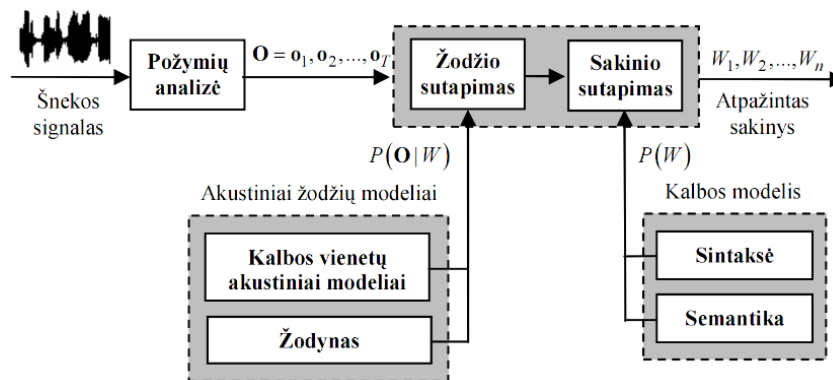
Kalbos atpažinimo procesas, pavaizduotas 1.2.1 paveiksle, prasideda į kalbos atpažinimo sistemą įvedus garsinį kalbos signalą ir baigiasi tekstinio atitikmens pateikimu.[4]



1.2.1 pav. Kalbos atpažinimo procesas [4]

Šnekamosios kalbos supratimas technologijoms yra komplikotas uždavinys, reikalaujantis perprasti sudėtingą fizikinį procesą, priklausantį nuo konkrečios kalbos, kalbinės aplinkos, diktoriaus artikuliacinių savybių ir kitų faktorių, lemiančių šnekos atpažinimo sudėtingumą.

Taigi, automatinis kalbos atpažinimas – visų pirma procesas, kuris technologijoms suteikia galimybę vertinti ir „suprasti“ natūralią žmogaus kalbą panaudojant įvairius algoritmus ir metodus.[10] Bendroji šnekamosios kalbos atpažinimo schema pateikiama 1.2.2 paveiksle.



1.2.2 pav. Šnekamosios kalbos atpažinimo sistemos schema [5]

Pagrindinės automatinės šnekamosios kalbos atpažinimo sistemos dalys:

- požymių išskyrimas iš šnekos signalo;
- garsinę informaciją atitinkančių modelių (akustinių ir kalbos) formavimas;
- nežinomo ištario klasifikavimas vienam iš atitinkančių modelių — žodžių ir sakinių atpažinimas [16].

Kalbos technologijos taip pat skirstomos pagal jų taikymo sritį ir poreikį:

- *gramatiškai apribotos šnekos* – plačiai taikomos klientų aptarnavimo sferoje;
- *kalbos atpažinimo* (garsinio kalbos signalo vertimas į tekstą) arba šnekos indentifikavimo;
- *kalbos sintezės* (tekstas verčiamas į garsinius signalus) – Tokios technologijos tik įgarsina rašytinį tekstą;
- *ištisinės arba natūralios šnekos* – sudėtingesnės formos šnekamosios kalbos atpažinimo įranga, kur vartotojas gali paaiškinti problemą, arba prašyti aptarnauti; [15]

Naudoti šnekos atpažinimo technologijas yra labai patogu, nes technologijų valdymui nebereikia skirti tiek dėmesio. Nors dar nėra sukurta AŠA sistemų, kurios nepriekaištingai atpažįsta kalbą taip, kaip žmogus, pastaraisiais dešimtmečiais mokslininkai sukūrė ir išstobulino nemažai šnekos atpažinimo sistemų, galinčių išspręsti įvairius kalbos atpažinimo uždavinius.[15]

1.2.1 Šnekamosios kalbos samprata

Šnekamoji kalba yra skirstoma į tokius tipus:

- *Izoliuoti žodžiai (isolated words)*. Izoliuotų žodžių atpažinimo sistemai reikia pateikti ištariamą (žodį ar frazę), ribojamą šnekos signalo charakteristikų (tyla, žemo lygio triukšmas).
- *Rišlios frazės (connected words)* yra tarpinis variantas tarp izoliuotų žodžių ir ištisinės šnekos.
- *Ištisinė šneka (continuous speech)* yra skaitytinė šneka.
- *Spontaniška šneka (spontaneous speech)* yra kasdienė šneka. Spontaninė šneka nėra sklandi šneka, dažnai joje yra ilgų pauzių, žodžių fragmentų (pradėtų ir nebaigtų žodžių), neteisingų ištarių, mikčiojimų, pasikartojimų, sintaksiškai neteisingų sakinių [4].

Šnekos analizės būdai naudojami projektuojant šnekos atpažinimo sistemas - *fonologija*, kuri nagrinėja fonetinius kalbos elementus ir *akustinė fonetika*, kuri spektriniais metodais tiria patį garso signalą. Fonologija atsako į klausimą kiek ir kokių fonetinių vienetų yra šnekamojoje kalboje, o akustinė fonetika pateikia šių vienetų charakteristikas spektriniais metodais.

Kiekvienai atpažinimo sistemai reikia parinkti ir sukurti tinkamiausius šnekos signalo atpažinimo modelius, todėl šie ankščiau minėti kalbos analizės būdai taikomi:

- kalbos vienetų parinkimo etape – formuojamų kalbos vienetų aibių elementai parenkami atlikus kalbos analizę fonologiniu aspektu.
- kalbos vienetus atitinkančių akustinių modelių mokymo metu –akustinės fonetikos tikslui įgyvendinti.

Kadangi šnekos atpažinimo sistemos mokymo, atpažinti reikalingas frazes, žodžius ar skiemenis, metu duotų kalbos vienetų spektriniai atitikmenys gaunami automatiškai, toliau aptariama fonologinė kalbos analizė.

Fonetinių vienetų skaičius pagal pasirinktą segmentinį elementą lemia akustinių modelių skaičių. Skirtingų kalbų automatinį šnekos atpažinimo (AŠA) sistemų naudojamos kalbos vienetų ir akustinių modelių aibės skiriasi tiek atskirų jos elementų spektrine išraiška, tiek skaičiumi, nes visos kalbos turi savas fonetinių vienetų sistemas.

Tinkamai pasirinkti kalbos vienetus yra labai svarbu. Pagrindiniai kalbos vienetų tipai yra: *fonemos, kontekstinės fonemos, skiemenys, kontekstiniai skiemenys ir žodžiai*.

- *Fonema (phoneme)* – minimalus kalbos garsas, leidžiantis atskirti vieną išstartą žodį nuo kito.
- *Skiemu*o - pereinamasis segmentinis vienetas tarp fonemos ir žodžio.
- *Žodis* – pasak A. Girdenio „leksinis vienetas, kuriam konvencijos būdu priskirta reikšmė ir kuris gali būti naudojamas įvairiose sintaksinėse kombinacijose, priklausomai nuo to, kokiai kalbos daliai jis priklauso“.

Automatinės šnekamosios kalbos atpažinimo sistemos atpažįsta smulkesnius žodžio elementus - skiemenis, kontekstines fonemas ar fonemas – atpažintuvui pateikiamas šių elementų junginys, reprezentuojantis žodį. Taip sukuriama ribos atpažinimo erdvei, o atpažinimo rezultatai pateikiami žodžių atpažinimo tikslumu [8].

Dabartinėse šnekos atpažinimo sistemose akustiniai modeliai yra kuriami segmentiniams kalbos vienetams. Kalbos vienetais gali būti ir fonemų kombinacijos, gaunamos taikant automatinį darbo su akustiniais duomenimis metodus. Taigi, kalbos vienetu bus vadinama gauta vieno ar kelių segmentinių kalbos vienetų kombinacija.

Automatinėje šnekos atpažinimo sistemoje operuojama minėtais kalbos vienetais, kurių skaičius yra baigtinis ir apibrėžiamas prieš AŠA sistemos modeliavimą. Žodžio atpažinimas yra pagrįstas atitinkamos sekos kalbos vienetams sukurtu akustinio modelio atpažinimu. Žodžiai ir juos sudarančios kalbos vienetų sekos laikomos atpažinimo sistemos žodyne. Automatinės šnekos

atpažinimo sistemos žodyno žodžio transkripcija yra kalbos vienetų nuoseklus junginys, kuris gali būti sudarytas tiek iš fonemų, tiek iš skiemenų.

Vienas iš svarbiausių akustinio modeliavimo uždavinių yra minėtų kalbos vienetų tipo ir konkretaus tipo aibės sudėtinių elementų parinkimas. Kalbos vienetų tipo parinkimo būdai galimi du - naudojant mokymo duomenų klasterizacijos metodus ir kalbos vienetų išankstinis apibrėžimas remiantis lingvistiniu kriterijumi [4].

Kalbos vienetų aibės gali būti baigtinės nesuskaičiuotos (žodžiai, skiemenys ir kt.) ir baigtinės suskaičiuotos (fonemos). Iš pastarųjų aibių atrenkami tie elementai, kuriuos galima rasti automatinėse šnekos atpažinimo sistemos kūrimui naudojamuose duomenyse – garsyne (žr. skyrių 1.4).

Kalbos vienetų aibės gali būti:

- besiremiančios lingvistiniu kriterijumi aibės (pvz.: fonemos, skiemenys);
- kompiuterio generuotos aibės.

Parentant kalbos vienetų tipą ir aibes būtina atsižvelgti į tai ar šnekos atpažinimo sistemai reikia atpažinti ištisinę šneką ar atskirus žodžius, o nuo to priklauso AŠA sistemos sudėtingumas.

Šnekos atpažinimo sistemose segmentinių kalbos vienetų akustinės charakteristikos sužinomos sistemos apmokymo metu ir užkoduojamos akustiniuose modeliuose, kurie atitinka lingvistinius kalbos vienetus. Kiekvienas kalbos žodis gali būti ištartas jungiant segmentinius kalbos vienetus įvairiomis kombinacijomis, tai akustiniai segmentinių kalbos vienetų modeliai, AŠA tikslais, leidžia modeliuoti kiekvieną kalbos žodį.

Fonemos yra pagrindiniai kiekvienos kalbos lingvistiniais vienetai. Pagal jas kuriami nuo konteksto nepriklausančių fonemų akustiniai modeliai. Fonemos priklausančios nuo konteksto, ilgesnės trukmės lingvistiniai kalbos vienetai – skiemenys, žodžiai yra alternatyvos fonemoms.

Kalbos vienetų akustinis modeliavimas yra lingvistinių kalbos vienetų atitinkančių akustinių modelių kūrimo ir mokymo (modelių parametrų įvertinimo) procesas. Šis procesas vyksta naudojant mokymo duomenis – paruoštą garsyną, kuriame minėti kalbos vienetai pasirodo pakankamą skaičių kartų. Vienas iš pagrindinių klausimų mokymo procese – mokymo imties klausimas. Mokymo imtis visada yra baigtinė, vadinasi, vieni kalbos vienetai pasikartos rečiau už kitus, todėl mokymo metu gauti akustiniai modeliai gali būti nepakankamai reprezentatyvus.

Automatinės šnekos atpažinimo sistemų apmokymo etapo pradžioje nustatoma akustinio modelio parametrų struktūra. Pavyzdžiui, paslėptasis Markovo modelis (PMM), kurio parametrai įgyja reikšmes mokymo metu pagal pateiktą mokymo medžiagą. Formuojant PMM struktūras reikalinga informacija - stebimų vektorių tipas ir dydis, būsenų skaičius, Gauso mišinių komponentų skaičius kiekvienai emituojančiai būsenai, perėjimo matricos reikšmių struktūra.

Žodynas – tai automatinės šnekos atpažinimo sistemos naudojamų žodžių sąrašas su žodžių transkripcijomis ir sudaromas tik iš apibrėžtos segmentinių kalbos vienetų aibės. Jame yra ir akustinių modelių deriniai, naudojami žodžiui atpažinti. Žodynas naudojamas tiek mokymo, tiek

atpažinimo etapuose. Automatinių šnekos atpažinimo sistemų žodynai sudaromi pagal jau esančius skaitmeninio formato kalbos transkripcijų žodynus [4].

1.2.2 Šnekamosios kalbos atpažinimo sistemos ir jų klasifikavimas

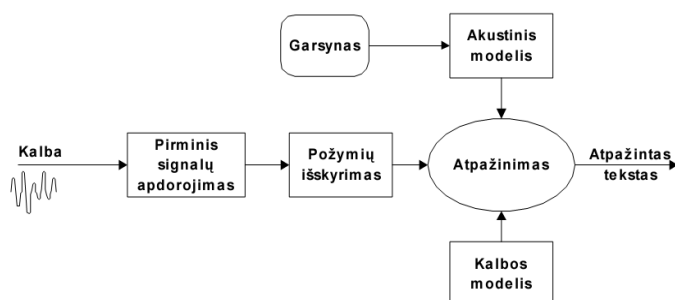
Automatinę šnekos atpažinimo sistemą dar 1989 metais Rabiner įvardijo kaip „*juodąją dėžę*“, kas nepasikeitė iki šių dienų. Taip yra todėl, kad mes negalime paaiškinti daugumos kalbos atpažinimo sistemose vykstančių procesų ir gaunamų kalbos atpažinimo rezultatų.

Prireikė net keliasdešimt metų, kol atsirado pirmosios praktiškai taikomos šnekos atpažinimo sistemos, nes šnekos atpažinimas yra sudėtingas uždavinys, o jo sudėtingumą nulemia tokios esminės priežastys:

- ne vieną kartą išstarto tam tikro garso akustinė realizacija labai skiriasi, net jei jį ištarė tas pats diktorius ir tame pačiame žodyje;
- kalbėjimo greitis dažnai būna nepastovus, todėl skiriasi kelių to paties žodžių akustinių realizacijų ilgis. Kintant žodžių ilgiui atskirų garsų ilgis kinta netiesiškai;
- kiekvieno žmogaus šnekos tarimas yra skirtingas, todėl reikia arba apmokyti sistemą darbu su konkrečiu diktoriumi, arba sistema kūrimo metu turi būti apmokyta su kuo didesniu diktorių skaičiumi;
- kalbėjimo sraute gali būti ir nekalbinių fragmentų (pvz., atodūsis), kuriuos reikia atskirti ir pašalinti [7].

AŠA sistemų veikimo kokybę vertinama pagal jų atpažįstamos šnekos tikslumą, kuris priklauso nuo sistemose techniškai naudojamų metodų. Šnekos atpažinimo sistemoje apdorojami akustiniai signalai, išskiriami požymiai, ieškoma alternatyvų, priimami sprendimai naudojant įvairius algoritmus, taisykles ir modelius.

Atpažinimo tikslumą sąlygoja kalbų įvairovė, kalbėjimo aplinkos ir perdavimo kanalo neapibrėžtumai (triukšmas, iškraipymai) ir kitos kintančios aplinkybės apsunkinančios atpažinimo procesą. 1.2.3 paveiksle pateikiama supaprastinta šnekos atpažinimo schema, kuri yra būdinga visoms AŠA sistemoms.



1.2.3 pav. Automatinio šnekos atpažinimo schema [10]

Šiuolaikinėse šnekos atpažinimo sistemose pirmiausia įeinančiam šnekos signalui taikomas pirminis apdorojimas, kurio metu signalas diskretizuojamas ir skaidomas į vienodo ilgio langus iš kurių pagal požymių apskaičiavimo algoritmus sudaromi požymių vektoriai, kuriuose yra užkoduojama informacija aprašanti akustinį kalbos signalą. Po to naudojantis kalbos ir akustiniais modeliais yra atliekamas požymių vektorių statistinis atpažinimas ir pateikiama labiausiai tikėtina fonetinių elementų transkripcijų seka [10].

Dar plačiau reikia panagrinėti šnekos atpažinimo proceso sudedamąsias dalis, be kurių neišsiverstų nei viena kalbos atpažinimo sistema.

Atpažįstant šneką reikia išanalizuoti nedidelį signalo fragmentą ir nustatyti jo priklausomybę vienai iš galimų klasių arba nustatyti jo panašumą etaloniniams signalo fragmentams ir nustatyti visos signalo fragmentų sekos (paprastai vadinamos stebėjimų seka) priklausomybę vienai iš galimų ištarimų klasių. Pirmasis uždavinys vadinamas lokalaus panašumo arba lokalių būsenų nustatymo uždaviniu, o antrasis globalaus panašumo arba tiesiog atpažinimo uždaviniu. Lokalaus panašumo nustatymas atliekamas taikant pasirinktą atstumo matą arba nustatant priklausomybės tam tikrai būsenai tikimybę, o globalaus panašumo nustatymas atliekamas naudojant dinaminės laiko skalės transformacijos algoritmą arba modeliuojant ištarimą paslėptąja Markovo grandine [15].

Šiuo metu dauguma kalbos atpažinimo sistemų yra grindžiamos paslėptųjų Markovo modelių (PMM) teorija. Paslėptųjų Markovo modelių (angl. Hidden Markov Models (HMM)) metodas gali aprašyti tam tikro laike kintančio proceso savybių kitimą ir to kitimo statistines charakteristikas. Pagrindinė PMM taikymo priežastis yra - kad šneka gali būti aprašyta kaip atsitiktinis parametrinis procesas ir šio scholastinio proceso parametrai gali būti pakankamai tiksliai nustatyti [15]. Nors PMM modeliai šiuo metu yra viena pažangiausių kalbos atpažinimo technologijų, tačiau apmokymui ir atpažinimui naudojami algoritmai turi esminių trūkumų. Pavyzdžiui, modeliuoja kalbos signalą kaip tiesinę fonemų seką su papildomais šalutiniais akustiniais įvykiais (triukšmas, atodūsis, tyła ir pan.) [10]. Tokiems trūkumams šalinti, ieškoma kitų šnekamosios kalbos modeliavimo alternatyvų.

Kaip alternatyva yra taikomi dirbtiniai neuroniniai tinklai (DNT) (angl. Artificial Neural Network (ANN)) – universalūs modeliavimo įrankiai, įgalinantys modeliuoti ypač sudėtingas funkcijas, pasižymintys didele atskiriamumo galia ir nereikalaujantys žinių apie duomenų pasiskirstymą. Bet atpažįstant šneką, turi ribotas galimybes apdoroti ilgalaikes dinamines kalbos signalų požymių vektorių sekas [17]. Nors dirbtiniai neuroniniai tinklai greitai ir plačiai paplito šnekos atpažinime, tačiau dėl šnekos atpažinimo sudėtingumo vien neuroniniais tinklais neįmanoma pasiekti puikių atpažinimo rezultatų dėl kelių DNT trūkumų - dirbtiniai neuroniniai tinklai reikalauja daug laiko apmokymo procesui ir nėra tinkami modeliuojant ilgalaikes priklausomybes, tokias kaip žodis ar sakinytis [10].

Siekiant pašalinti PMM ir DNT trūkumus sprendžiant šnekos atpažinimo problemas kai kurie mokslininkai pradėjo integruoti juos į vientisą hibridinę architektūrą [17]. Pagrindinis hibridinių DNT/PMM sistemų tikslas – pasinaudojant pranašumais, padidinti kalbos atpažinimo sistemų patikimumą ir lankstumą. Hibridinėje šių sistemų struktūroje DNT naudojami šnekos signalų akustiniam modeliavimui, o PMM – šnekos signalų kitimo laike modeliavimui.

Dar viena alternatyvi priemonė yra - dinaminio programavimo (dinaminio laiko ašies koregavimo (DLK)) (angl. Dynamic Time Warping (DTW)) algoritmo pritaikymas šnekos atpažinimui, kuris padeda rasti optimalų panašumą tarp dviejų vektorių sekų, išplečiant ar suspaudžiant atskirus vektorių sekos segmentus. DLK algoritmas netiesiškai išplečia arba sutraukia kiekvieno iš palyginamų tarpusavyje testuojamo ir etaloninio žodžių laiko ašis, kad juose apytikriai sutaptų atskirų fonemų pozicijos. Šio algoritmo pagrindinis trūkumas – didelės skaičiavimų apimtys prie didelio atpažįstamų žodžių rinkinio. Kitas trūkumas – sudėtingas atpažįstamų žodžių etalonų sudarymas. Ypač sunku automatiškai tiksliai nustatyti fonemų ribas ištarime, kad būtų galima atlikti tam tikros fonemos statistinių parametrų analizę [11].

Šnekos atpažinimo sistemose dar yra galimybė taikyti vizemų atpažinimo modelius. Šiuo metu daugiausia tiriamas vien tik garso signalas, kurio analizei taikomas paslėptųjų Markovo modelių metodas. Norint pagerinti šnekos atpažinimo tikslumą PMM metodu yra galimybė tiriant garsinius signalus naudoti ir vaizdinę informaciją – vizemas. Vizemos – burnos vaizdo informacija [18].

Šnekos atpažinimo sistemos yra klasifikuojamos pagal realizuotas funkcijas arba pagal atpažįstamo akustinio signalo sudėtingumą. Diktorių skaičius, kalbėjimo stilius ir žodyno apimtis – pagrindiniai kalbos atpažinimo sistemų klasifikavimo aspektai. Kiekvienas iš jų dar skirstomas į smulkesnes grupes.

Pagal diktorių skaičių yra skiriamos priklausančios nuo diktoriaus (dar vadinamos *viendiktorinės*), adaptuojamos ir nepriklausančios nuo diktoriaus (*daugiadiktorinės*) šnekos atpažinimo sistemos [10][14].

Pagal kalbėjimo stilių skiriamas rišlios, spontaniškos ir izoliuotos šnekos atpažinimas. Izoliuotos šnekos atpažinimo sistemos labiau pritaikytos šnekos atpažinimui paraidžiui ir pažodžiui. Jos sugeba atpažinti žodžius ar komandas, kurias ištarus daromos ilgesnės pauzės [10].

Pagal žodyno apimtį atpažinimo sistemos skiriamos į siauro žodyno sistemas (tokius žodynus sudaro keliasdešimt žodžių), vidutinės apimties (šimtai žodžių), didelės apimties (tūkstančiai žodžių) ir labai didelės apimties (dešimtytūkstančių žodžių) AŠA sistemas [10].

Taigi, didinant diktorių skaičių ir žodyno apimtį, tuo pačiu didėja ir šnekos atpažinimo sudėtingumas. Šiuo metu pagrindinis kalbos atpažinimo sistemų kūrėjų tikslas yra užtikrinti aukštą, nepriklausomą nuo diktoriaus spontaniškos šnekamosios kalbos atpažinimo tikslumą ir patikimumą. Tačiau šnekos atpažinimo tyrimai Lietuvoje vis dar labai atsilieka nuo šiuolaikinės visuomenės poreikių, tai lemia sudėtinga lietuvių kalbos fonetika ir didelė garsų įvairovė.

1.2.3 Markovo modelių taikymas šnekos atpažinimui

Paslėptieji Markovo modeliai (PMM) yra vienas iš populiariausių ir dažniausiai taikomų priemonių šnekamosios kalbos atpažinimo sistemų projektavime, todėl tikslinga apžvelgti jų veikimo principą ir taikymą šnekamosios kalbos atpažinimui.

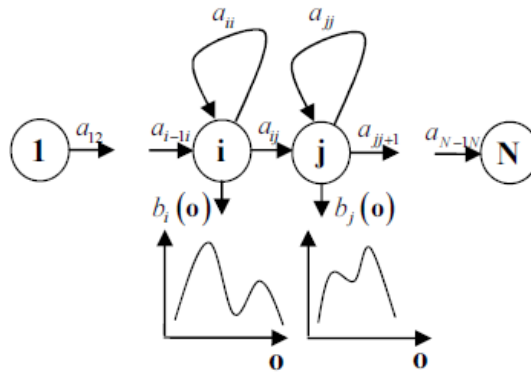
Šnekos atpažinimui naudojami tiek deterministiniai (modeliai, kuriuose paprastai naudojamos kokios nors žinomos signalo savybės (pvz. signalas yra sinusinis arba eksponenčių suma ir kt.), kurioms reikia nustatyti signalų modelio parametrus (pvz. amplitudę, dažnį, fazę ir kt.)), tiek stochastiniai (modeliai, kuriuose aprašomos statistinės signalo savybės, kurios gali būti aprašomos, kaip atsitiktinis parametrinis procesas, o jo parametrai gali būti nustatomi tiksliai kiekybiškai) signalų modeliai. Vienas iš stochastinių signalų modelių yra paslėptasis Markovo modelis.

Paslėpti Markovo modeliai pradėti taikyti šnekos atpažinime, nes gerai tiko aprašyti laike kintančio proceso (pvz. šnekos signalo) savybių kitimą ir to kitimo statistines charakteristikas. Taikant PMM šnekamosios kalbos atpažinimui daroma prielaida, kad šneka gali būti pakankamai gerai aprašyta, kaip atsitiktinis parametrinis procesas ir šio stochastinio proceso parametrai gali būti pakankamai tiksliai nustatyti [11].

Paslėptasis Markovo modelis apibrėžiamas kaip baigtinė būsenų, susietų perėjimo tikimybėmis, seka, naudojama signalų laikiniam ir spektriniam kitimui modeliuoti. PMM aprašomas nusakant penkis dydžius: būsenų skaičių; skirtingų stebėjimų būsenoje skaičių arba stebėjimų paskirstymo tankį; perėjimo iš vienos būsenos į kitą tikimybių matricą; stebėjimų tikimybinius skirstinius arba stebėjimų tikimybės tankio funkcijas; pradinio (initial) buvimo būsenoje tikimybės. Šnekos atpažinimo modeliavime pradine būseną laikoma pirmoji paslėptojo Markovo modelio būseną. Šie penki dydžiai nusako konkretų paslėptąjį Markovo modelį.

Teigiant, kad šnekos signalą ar jo dalis gali reprezentuoti PMM, šnekos signalas reprezentuojamas kaip būsenų seka. Pačios būsenos nėra stebimos, o paslėptos, bet apie jų seką galima spręsti iš kalbos signalo. Darant prielaidą, kad garsas gali būti skaidomas į tris dalis – pradžia, nekintančia dalį ir pabaigą, garsą įprasta aprašyti trimis būsenomis. Modelio pradžioje ir pabaigoje pridedama po vieną papildomą neemituojančią (angl. non-emitting) būseną, kurios sujungia kelis modelius.

Daugumoje atpažinimo sistemų naudojamų paslėptųjų Markovo modelių struktūra yra pavaizduota 1.2.4 paveiksle.



1.2.4 pav. Struktūrinis paslėptojo Markovo modelio su neemituojančiomis įėjimo ir išėjimo būsenomis vaizdas [4]

PMM tampa akustiniu modeliu tik po mokymo, kurio metu pagal stebėjimų duomenis įvertinami modelio parametrai, t. y. perėjimo iš būsenos į būseną tikimybių matrica ir stebėjimų tikimybės tankio funkcijos kiekvienai modelio būsenai. Ji yra išreiškiama stebėjimų arba požymių vektorių tikimybės tankio funkcija, nusakančia konkretaus požymio vektoriaus tikimybę.

Pirmas etapas kuriant atpažintuvą yra PMM aibės formos ir struktūros nustatymas. Antras etapas yra PMM parametrų prognozavimas, remiantis modeliuojamais duomenimis ir vadinamas apmokymu. Šnekos signalas paprastai koduojamas parametrine forma, pavyzdžiui, tiesinės prognozės kodavimo (angl. Linear Predicting Coding (LPC)) koeficientais ar melų skalės keprstiniais koeficientais (angl. Mel-Frequency Cepstral Coefficients (MFCC)).

Atpažinimo metu daroma prielaida, kad nežinomą ištarimą atitinkanti tiriamų požymių vektorių seka yra gaunama - generuojama PMM sekos. Skaičiuojamos to ištarimo atitikimo akustinių modelių deriniams tikimybės ir labiausiai tikėtina modelių kombinacija identifikuoja ištarimą. Atpažinimo procesas grindžiamas Viterbi algoritmu. Šis algoritmas gali būti vizualizuotas kaip geriausio sprendimo paieška matricoje, kurioje vertikalčiai išdėstomos PMM būsenos, o horizontalčiai – požymių vektoriai [11].

Norint atpažinti pavienius žodžius reikia kiekvienam žodyno žodžiui numatyti modelio parametrus, kurie optimizuotų apmokymo rinkinio stebėjimo vektorių panašumą žodžiui ir kiekvienam nežinomam žodžiui, kuris gali būti atpažintas, išanalizavus šnekos požymius atitinkančius tą žodį, suskaičiavus modelio panašumus į visus galimus modelius bei atrinkus panašius žodžius, matuoti stebėjimo seką.

Gana komplikauta šnekos atpažinimo problema, kuriai spręsti taikomas PMM, yra keleto rišliai išartų žodžių atpažinimas. Šiuo atveju atpažinimas pagrįstas viso žodžio modeliais. Atpažinimo problema (kai sudaromi atitinkami žodžių modeliai) – rasti optimalią žodžių modelių seką, kuri geriausiai atitiktų (būtų labiausiai panaši) į nežinomą keleto žodžių seką. Daugiausia sunkumų kyla didelės apimties žodyno atpažinimo sistemose, kur šnekos atpažinimui naudojami modeliai, sudaryti iš šnekos vienetų mažesnių nei žodis.

PMM teorija pritaikyta fonemų tipo žodžio dalims, vaizduojant žodžius ir sintaksę. Taip pat naudojami ergodiniai PMM, kur kiekviena būseną atvaizduojama, kaip akustinis fonetinis vienetas. Modelyje pritaikoma kintamos trukmės savybė kiekvienoje būsenoje, nes balsio tipo garsai turi labiau skirtingą trukmę nei priebalsiai. Naudojamas leksinis metodas, kartu su standartiniu tarimo žodynu, kad nustatyti geriausiai atitinkančias žodžio sekas iš žodžio dalies [11].

1.2.4 Šnekamosios kalbos analizė ir požymių išskyrimas

Šnekos signalas yra perteklinis - be lingvistinės informacijos signale yra informacija apie kalbantįjį, kalbėjimo aplinką ir t. t. Pirmasis etapas šnekamosios kalbos atpažinimo procese - kalbos signalo analizė, nes, šiuo atveju, mus domina tik lingvistinė informacija. Kalbos signalo analizės tikslas – išskirti požymius – signalo charakteristikas, atspindinčias lingvistinį signalo turinį. Tokių požymių naudojimas leidžia sumažinti nagrinėjamų duomenų kiekį ir padidinti jų gebėjimą atvaizduoti fonetinius skirtumus. Signalo analizę sudaro keletas etapų: signalo skaidymas į persidengiančius kadrus (kai kuriais atvejais signalas prieš skaidymą filtruojamas), dauginimas iš lango funkcijos, spektrinė analizė, reikiamų požymių išskyrimas.

Plačiau apžvelgsime signalo spektrinę analizę, kuri yra plačiausiai naudojamas kalbos signalo analizės metodas.

Kalbos signalo spektrui skaičiuoti dažniausiai naudojama klasikinė skaičiavimo schema - skaičiuojant signalo Furjė transformaciją, ją logaritmuojant ir atliekant atvirkštinę Furjė transformaciją - gaunamas signalo kepstras. Taigi signalo kepstru vadiname signalo Furjė transformacijos logaritmo atvirkštinę Furjė transformaciją.

➤ **Kepstrinė signalo analizė.**

Signalų kepstro ypatinga savybė yra tokia, kad jame išsiskiria greitai ir lėtai kintančios balso trakto komponentės.

Kepstras, kuris yra lygus atvirkštinei logaritminio spektro Furjė transformacijai, gali būti užrašytas tokia funkcija:

$$c(\tau) = F^{-1} \log |X(\omega)| = F^{-1} \log |G(\omega)| + F^{-1} \log |H(\omega)| \quad (1.2.1)$$

čia F reiškia Furjė transformaciją.

Dešinės lygybės pusės pirmas ir antras dėmenys atitinka greitai ir lėtai kintančius balso trakto komponentes. Pirmasis iš esmės atitinka periodinį žadinimą, o antrasis – balso trakto formą.

Dar viena kepstro rūšis yra - skirtuminis arba dinaminis kepstras, dar vadinamas delta-kepstras.

Skirtuminis kepstras turi informaciją tik apie kalbos signalo kepstro dinamiką, o tik jos neužtenka patikimai atpažinti šnekamąją kalbą. Todėl dažnai dinaminio kepstro analizės rezultatai apjungiami su statinio kepstro analizės rezultatais. Kartu su delta-kepstro analize kartais atliekama

ir delta-delta-kepstro analizė - skaičiuojama antroji kepstro išvestinė, rodanti kepstro kitimo pagreitį. Delta-delta-kepstro analizės rezultatai taip pat naudojami tik apjungus su statinės (ir delta-kepstro) analizės rezultatais.

Pagrindinis kepstro analizės privalumas - sugebėjimas atskirti šaltinio signalo nuo balso trakto spektrines savybes.

Šnekamosios kalbos atpažinimo sistemose didžiausią reikšmę turi melų skalės kepstro koeficientų rūšis - MFCC (angl. Mel Frequency Cepstral Coefficients). Melų skalės kepstro koeficientams apskaičiuoti yra naudojamas algoritmas:

1. Iškerpamas šnekos signalo pjūvis (apie 10-30 ms trukmės) naudojant Hemingo langą;
2. Signalo pjūviui atliekama Furjė transformacija ir gaunamas duoto pjūvio amplitudinis spektras;
3. Gautas amplitudinis spektras paverčiamas melų skalėje išdėstytais trikampaiais filtrais - nuo 20 iki 24 trikampių filtrų. Melų skalė modeliuoja - logaritmiškai aukštesniųjų dažnių srityje mažėjančią dažnių skyrimo gebą.
4. Trikampių filtrų išėjimo reikšmės logaritmuojamos ir iš logaritminių reikšmių apskaičiuojami kepstro koeficientai.

Melų dažnių skalės kepstro koeficientai (MFCC) pastaruoju metu naudojami beveik visose atpažinimo sistemose.

Atlikus kiekvieno lango analizę ir apskaičiavus kalbos signalo požymius iš jų galima sudaryti požymių vektorių, kuris bus naudojamas kaip automatinio kalbos atpažinimo sistemos įėjimas:

$$X_k = [c_k | E_k | \Delta_k | \Delta E_k | \Delta \Delta k | \Delta \Delta E_k] \quad (1.2.2)$$

Požymių vektorius gali būti sudarytas iš - kepstro koeficientų arba kitų kepstrinių, spektrinių ar tiesinės prognozės koeficientų (c_k), kalbos signalo energijos (E_k), kepstro delta koeficientų (Δ_k), energijos delta koeficiento (ΔE_k), kepstro akceleracijos koeficientų ($\Delta \Delta k$), energijos akceleracijos koeficiento ($\Delta \Delta E_k$).

Požymių vektoriaus elementų skaičius priklauso nuo naudojamo kepstro koeficientų skaičiaus, pavyzdžiui, jei naudojame 12 kepstro koeficientų, tai požymių vektorių sudarys 39 elementai.

Kartu su kepstro koeficientais naudojamas kalbos signalo energijos požymis – energija bei jos delta ir delta-delta koeficientai. Energija gali būti labai svarbus matas bandant tarpusavyje atskirti panašias fonemas [15].

1.2.5 Žodžių modeliais grįstas šnekamosios kalbos atpažinimas

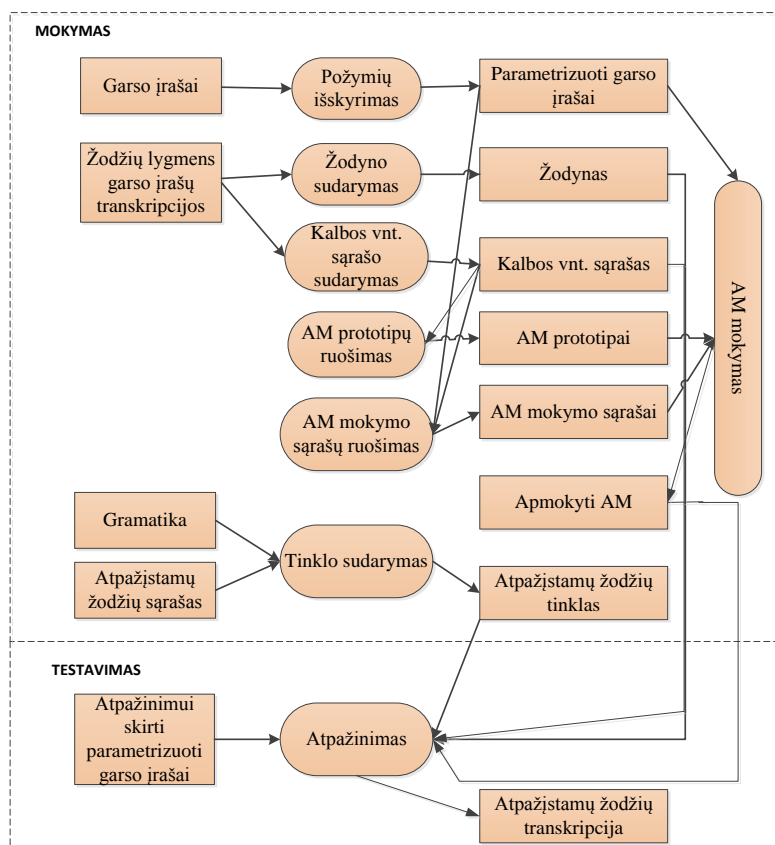
Izoliuotų žodžių atpažinimas kur kas lengvesnis uždavinys nei spontaniškos ar ištisinės šnekos atpažinimas. Izoliuotų žodžių atpažinimas vykdomas dviem metodais - tiesiogiai naudojant

spektrinę informaciją šnekos pavyzdžiams palyginti ir naudojant statistinius modelius spektrinės informacijos apibendrinimui.

Pirmosios izoliuotų žodžių atpažinimo sistemos buvo kuriamos pavyzdžiais grįsto šnekos atpažinimo metodo principu. Tokios atpažinimo sistemos tipinį modelį sudaro sprendimo priėmimo blokas, pavyzdžių palyginimas ir požymių išskyrimas. Toks modelis yra patogus dėl invariantiškumo žodynams, vartotojams, pavyzdžių palyginimo algoritmams, sprendimo taisyklėms, lengvo įgyvendinimo ir gero praktinio veikimo. Požymių pavidalo šnekos signalo spektrinė informacija yra tokių sistemų etaloniniai ir testiniai pavyzdžiai. Jiems palyginti šnekos signalo požymiai taikomi skaičiuojant atstumą dinaminio laiko skalės iškraipymo būdu (Dynamic Time Warping, DTW), taikant dinaminę programavimą. Nuo atpažinimo sistemos priklausymo ar nepriklausymo nuo kalbėtojo, skiriasi etalonų formavimo procesas – mokymas ir sprendimo taisyklė. Jei atpažinimo sistema priklausoma nuo kalbėtojo, užtenka vieno ar dviejų kalbėtojo ištarimų spektrinės informacijos. Jei atpažinimo sistema nepriklauso nuo kalbėtojo, atliekamas vektorinis kvantavimas (kalbėtojų ištarimų spektrinės informacijos pavyzdžiai suskirstomi į mažesnę pavyzdžių skaičių). Nepriklausančią nuo kalbėtojo sistemą tiesiogiai veikia - žodyno sudėtingumas, lemiantis atpažinimo tikslumą; perdavimo kanalai, darantys įtaką šnekos signalo galo taškų nustatymui ir atpažinimo tikslumui; pačios sistemos sudėtingumas, veikiantis sistemos atsako laiką [4].

1.2.5 paveiksle pateikta izoliuotų žodžių modeliavimo schema atpažinimui, taikant paslėptuosius Markovo modelius. Šioje schemoje stačiakampiai vaizduoja procesams pateikiamus ar po procesų gaunamus objektus, ovalai - procesus. Pradedant modeliuoti izoliuotų žodžių atpažinimo sistemą, kiekvienas garso įrašas parametrizuojamas „*Požymių išskyrimo bloke*“, o iš garso įrašų žodinių komandų transkripcijų „*Žodyno*“ ir „*Kalbos vienetų sąrašo sudarymo*“ išskiriami žodynas ir kalbos vienetų sąrašas. Po to pagal kalbos vienetų sąrašą sukuriama juos atitinkančių akustinių modelių (AM) prototipai ir paruošiamos mokymo imtys „*AM prototipų*“, ir „*AM mokymo sąrašų ruošimo*“ blokuose. Pradiniai duomenys akustinių modelių mokymui „*AM mokymo*“ bloke – yra pirmuose etapuose gautieji rezultatai ir parametrizuoti garso įrašai. Mokymo etapas pasibaigia atpažįstamų žodžių tinklo sudarymu. Testavimo bloke „*Atpažinimas*“ vyksta atpažinimui skirtų parametrizuotų garso įrašų vertimas į žodžių transkripcijas, naudojant mokymo metu gautus atpažįstamų žodžių tinklą, apmokytus akustinius modelius, kalbos vienetų sąrašą ir žodyną [4].

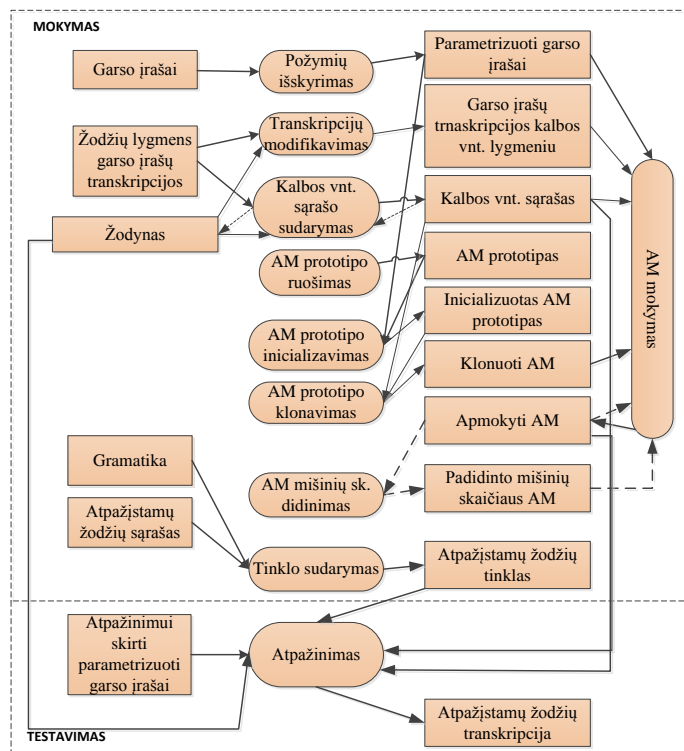
Procesų sukeisti vietomis pateiktoje izoliuotų žodžių atpažinimo sistemos modeliavimo schemoje 1.2.5 pav., negalima, nes visas modeliavimas vyksta nuosekliai ir tolesniems procesams reikia ankstesniuose paruoštų įėjimo duomenų. HTK įrankiai realizuoja visus procesus, išskyrus žodyno sudarymą, keitimą, akustinių modelių prototipų ruošimą ir garso įrašų segmentavimą [4].



1.2.5 pav. Izoliuotų žodžių atpažinimo sistemos, mokymo ir testavimo etapų detalizavimas.

1.2.6 Fonemų modeliais grįstas šnekamosios kalbos atpažinimas

Šnekos atpažinimo, grindžiant fonemomis ar skiemenimis, modeliavimo schema pateikta 1.2.6 paveiksle. Pagal šią schemą atliekant akustinį modeliavimą, garso įrašų fonemų ar skiemenų transkripcijas reikia keisti į transkripcijas fonemų ar skiemenų pagrindu „*Transkripcijų modifikavimo*“ bloke. O „*Kalbos vienetų sąrašo sudarymo*“ bloke sudaromas fonemų ar skiemenų sąrašas. Pagal anksčiau apmokytus kalbos vienetų akustinius modelius, kalbos vienetų sąrašą ir fonemų ar skiemenų sąrašą paruošiami kalbos vienetų AM, kurie yra mokomi „*AM mokymo*“ bloke. Po šio etapo kiekviena mokymo duomenyse esanti fonema ar skienuo turi savo akustinį modelį. Tada „*AM mokymo*“ ir „*AM mišinių skaičiaus didinimo*“ operacijos, atliekamos norimą skaičių kartu. Po mokymo proceso vyksta šnekos atpažinimo procesas fonemų ar skiemenų pagrindu. 1.2.6 paveiksle pateiktoje šnekos atpažinimo sistemos modeliavimo schemeje, procesų sukeisti vietomis negalima [4].



1.2.6 pav. Šnekos atpažinimo sistemos, grįstos skiemenimis ar fonemomis, mokymo testavimo etapų detalizavimas. Brūkšninės rodyklės vaizduoja AM mokymo procedūrų kartojimą.

1.3 Hidden Markov Model Toolkit programinis paketas

Paslėptaisiais Markovo modeliais pagristas HTK (angl. Hidden Markov Model Toolkit) – programinių priemonių rinkinys, sukurtas Kembridžo universitete. HTK vienas iš populiariausių programinių paketų naudojamų šnekamosios kalbos atpažinimo tyrimams, remiantis paslėptaisiais Markovo modeliais. HTK taip pat taikomas tyrimuose ir kitose srityse - kalbos sintezės, simbolių atpažinimo ir DNR sekų tyrimuose.

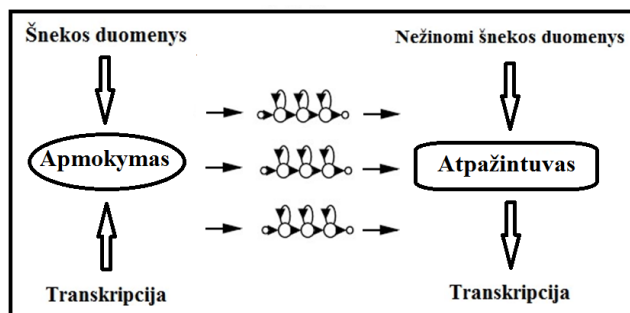
HTK biblioteką sudaro modelių rinkinys ir programiniai įrankiai parašyti C programine kalba. Šiame programiniame rinkinyje galime rasti priemones, skirtas duomenims apdoroti, mokyti ir testuoti akustinius ir šnekos modelius, priemones akustiniams modeliams adaptuoti konkrečiam kalbėtojui, bei rezultatų analizės priemones.

Naudojantis „*Hidden Markov Model Toolkit*“ programine įranga galima modeliuoti izoliuotų žodžių ir ištisinės kalbos atpažinimo sistemas, pasirinkti skirtingas paslėptųjų Markovo modelių topologijas ir parametrus, signalą reprezentuojančių požymių tipą. HTK priemonių valdymui naudojama tekstinė sąsaja. Šnekos atpažinimo rezultatai, gauti HTK programinėmis priemonėmis, pateikiami tik skaitine forma. Todėl atpažinimo klaidų analizė yra pakankamai sudėtinga, nėra paprasta išsiaiškinti klaidų priežastis. [20]

HTK programinėje įrangoje yra du pagrindiniai šnekos signalo apdorojimo etapai:

1. HTK programiniame pakete šnekamosios kalbos atpažinimui apmokymo įrankiai yra naudojami siekiant apytiksliai apskaičiuoti paslėptųjų Markovo modelių parametrus, naudojant apmokymui pateiktus komandų įrašus ir parinktas komandų transkripcijas;
2. Nežinomi komandų įrašai yra transkribuojami naudojant HTK programinius įrankius [21].

1.3.1 paveiksle pavaizduota schema, apibūdinanti šiuos du svarbiausius apdorojimo etapus.



1.3.1 pav. HTK šnekos apdorojimo schema

Naudojant HTK programinį paketą šnekos atpažinimui, kiekviename etape reikia panaudoti po atskirą programą iš HTK programinių įrankių paketo. Visos programos yra valdomos iš komandinės eilutės per papildomus failus ir eilutės komandas. [21]

Toliau pateikiami keturi pagrindiniai darbo su HTK programiniu paketu etapai žodžių atpažinimo modeliavimui:

1. Žodyno sudarymas. Žodyno sudarymas yra reikalingas tam, kad projektuojama sistema žinotų, kokios žodžių (akustinių vienetų) kombinacijos yra naudojamos atpažinimui;

2. Požymių apskaičiavimas. Atpažinimo sistemose naudojami iš balso komandų (žodžių) įrašų apskaičiuoti požymiai. Požymiams apskaičiuoti naudojama programa „Hcopy“, kuri naudodama įrašų failus apskaičiuoja požymius;

3. Modelių apmokymas. Apmokymas vyksta naudojant „HRest“ įrankį, kuris savo ruožtu naudoja Baum-Welch algoritmą apmokymo proceso vykdymui. Svarbus apmokymo etapas – modelių failų kūrimas, kuriuose nurodomi pagrindiniai PMM modelio parametrai. Modelių failai – tai specialaus formato failai, kuriuose yra surašoma atitinkamo modelio parametrai: perėjimų tikimybės ir išėjimų tikimybės. HTK operuoja tik tolydiniais PMM modeliais, todėl išėjimų tikimybės modeliuojamos Gauso skirstiniais. Vadinasi išėjimų tikimybės aprašomos vidurkiais ir standartiniais nuokrypiais;

4. Atpažinimas. Atpažinimo testavimas vykdomas naudojant „HVite“ įrankį. Po apmokymo etapo, naudojant testavimui skirtus įrašus (kitus įrašus nei apmokymui) galima atlikti atpažinimo testavimą ir patikrinti atpažinimo kokybę su esamais modeliais. Atpažinimo kokybės gerinimui iš naujo atliekami visi etapai nuo pirmojo keičiant transkripcijas ir modelių parametrus. [21]

1.4 Garsynai

Kalbos atpažinimo tyrimams yra labai svarbūs šnekamosios kalbos garsynai – surinkta struktūrizuota garso įrašų aibė konkrečiai kalbai, kur kiekvienas garso įrašas turi savo atitikmenį tekstu (žodžiais, skiemenimis ar transkripciją fenomenomis), žodyną.

Garsynai – tai didelės apimties šnekos signalų imtys, be jų nebūtų įmanomas šnekos atpažinimo sistemų modeliavimas [4]. Garsynai ne tik reprezentuoja kalbą, bet ir pasitarnauja atliekant jos tyrimus (siekiant atlikti išsamius kalbos tyrimus, tam tikslui yra kuriami gerai anotuoti, sisteminiai garsų fondai) [22]. Garsynai skiriasi transkribavimo lygmeniu, trukme, diktorių skaičiumi, renkamų garso įrašų turiniu, žodyno dydžiu ir kt.

Stebimos šios garsynų kūrimo tendencijos - didėjant taikomosios paskirties garsynų apimtims, automobilių pramonės, telefonijos kompanijos jų kūrimui skiria dideles investicijas; daugėja nacionalinių garsynų, kurie dokumentuoja kalbą, kaip tautos socialinės kultūrinės aplinkos paminklą; yra kuriami sisteminiai, gerai anotuoti garsynai, kurie skiriami išsamiesiems šnekos mokslo tyrimo darbams.

Būtina pakalbėti ir apie prototipą, į kurį yra orientuojamasi, kuriant naujus garsynus - tai TIMIT (Texas Instruments/Massachusetts Institute of Technology) šnekos duomenų bazė. Ji yra laikoma klasikiniu garsyno pavyzdžiu, kurios esmė – užkoduota pagrindinė garsyno organizacinė informacija direktorių struktūroje, o tekstiniuose failuose lentelėse – apie šnekos signalus, diktorius, jų anotacijas pateikta papildoma informacija. Visa informacija apie anotacijas, kalbėtojus ir kitus svarbius TIMIT duomenų bazės duomenis, yra saugomi atskiruose, keturių tipų failuose, kuriuos galima rasti atskirose direktorijose (visa duomenų bazė yra sukelta į CD-ROM diskus) [23].

Turint garsyną ypatingai svarbu yra nuodugnai patikrinti bei užtikrinti nepriekaištingą visų įrašų kokybę, nes įvairūs trikdžiai bei triukšmai, daro didžiulę įtaką, todėl jų neturi būti.

Pagal paskirtį garsynai yra skirstomi į sistemas atpažįstančias balsą, tačiau nepriklausančias nuo diktoriaus, ir į sistemas atpažįstančias balsą, tačiau pritaikomas (apmokomas) prie diktoriaus. Taip pat garsynus galima skirstyti į garsynus, kurių turinys yra nespontaniškos kalbos įrašai („perskaitytos šnekos“ – skaičių sekos, knygų ištraukos ir pan.), ir į garsynus, kurių turinys yra spontaniškos kalbos įrašai (dialogai ir pan.).

Kaip ir šnekos atpažinimui, taip ir garsynų anotavimui bei segmentavimui yra taikomi tie patys metodai: neuroniniai tinklai, PMM ir DLK modelis (dinaminis laiko ašies kraipymas (angl. - DTW - „Dynamic Time Wrapping“)). Šie metodai dažniausiai yra taikomi anotavimo bei segmentavimo sistemose esančiose šnekos technologijų tinklalapiuose [22].

Akivaizdu, jog garsynų įvairovė yra iš tiesų didelė, o kiekvienas jų atskirai labai savotiškas ir turintis tik jam būdingas charakteristikas. Atsižvelgiant į tai, šių laikų, efektyviai naudojamas šnekos atpažinimo sistemų kūrimui, garsynas, privalo: suteikti galimybes vartotojui manipuluoti dideliais duomenų kiekiais (atnaujinti, perkelti, naikinti) bei užtikrinti visų duomenų integralumą ir

korektiškumą; turėti efektyvią paieškos sistemą, kuri gebėtų užtikrinti šnekos signalų ir duomenų ištraukimą; turėti savybes, kurios atitiktų bendro darbo sistemos principus (daugiavartotojiškumas, duomenų kontrolė bei dalinimasis); turėti standartinę sąsają duomenų lygyje (duomenų importavimui ir eksportavimui į kitas sistemas); turėti standartinę programinę sąsają, kuomet sukūrus papildomą funkcionalumą, jis būtų lengvai prieinamas visiems vartotojams; turėti vartotojo sąsają ne tik patogiam duomenų bazės užklausimui, tačiau ir ataskaitų formavimui. Patogi vartotojo sąsaja leidžia koncentruotis į tyrimą, o ne į duomenų tyrimui paruošimą [23].

1.5 Projekto „Hibridinė atpažinimo technologija balso sąsajai“ apžvalga

2012-2013 metais buvo vykdytas projektas „Hibridinė atpažinimo technologija balso sąsajai“. Projektas buvo finansuojamas ir remiamas MITA (mokslo ir inovacijų agentūros). Jo trukmė buvo 18 mėnesių. Projekto idėja - panaudoti hibridinį atpažinimo metodą, kurį sudaro:

- tradicinis adaptuotas Lietuviškas paslėptaisiais Markovo modeliais pagrįstas atpažintuvas, kaip bazinis atpažinimo metodas daugumos balso komandų atpažinimui;
- suderinti du alternatyvūs atpažintuvai, skirti blogiau atpažįstamų komandų apmokymui, taikant šioms komandoms labiau tinkamus atpažinimo metodus ir tokiu būdu optimizuojant bendrą sistemos atpažinimo tikslumą.

Projekto tikslas – sukurti visiškai naują, šnekos atpažinimo technologiją ir ją panaudoti kuriant pirmos informacinės paslaugos, naudojančios komandų, lietuvių kalba, atpažinimą, kaip pagrindinį žmogaus ir kompiuterio sąsajos prototipą. Balso komandų atpažinimo sistema orientuojama į gydytojų/farmacinių darbuotojų darbo vietas, siekiant palengvinti ir pagreitinti norimos informacijos paiešką medicininėje duomenų bazėje neatsitraukiant nuo paciento ir nenukreipiant dėmesio į kompiuterinės informacinės sistemos sąsają.

Projekto rezultatas - hibridinis internetinis atpažintuvas (HIA) planuojamas taikyti elektroninio sveikatos įrašo informacinėje sistemoje „Foxus“. Hibridinio internetinio atpažintuvo demonstracijos paruoštos KTU (<http://193.219.154.23/infobalsas/home/html5recorder>) ir „UAB SoftDent“ (<http://infobalsas.softdent.lt/Home/HTML5Recorder>) serveriuose.

Atpažintuvo sukūrimas bei jo testavimas negalimas be balso resursų – garsynų. Projekto medicininį garsyną „Medic“ sudaro sudaro 175440 garso įrašai = 731 medicinoje vartojamų terminų x 12 diktorių (5 moterys ir 7 vyrai) x 20 tos pačios komandos ištarimų. Medicininės komandos buvo susegmentuotos į 3 garsynų grupes: VAISTAI, NUSISKUNDIMAI ir LIGOS. Visa garsyno struktūra pateikta 1.5.1 lentelėje 1 PRIEDE.

„Medic“ medicininis garsynas su 731 alternatyva dar buvo papildytas 269 komandomis atitinkamai iš vaistų, ligų ir nusiskundimų, kad medicininis garsynas sudarytų 1000 komandų, siekiant iširti Hibridinio atpažintuvo galimybes atpažinti didesnę kiekį komandų. Šios komandos nebuvo įdiktotos, buvo paruoštas tik komandų žodynas ir gramatikos. Garsyno papildymo struktūra pateikta 1.5.2 lentelėje 1 PRIEDE.

Taip pat pirminiam akustinių modelių ir fonetinių transkripcijų apmokymui ir nustatymui atpažinimo tikslumo testavimui buvo panaudoti turimi bendro pobūdžio 50 val. garsynai.

Šiame projekte buvo panaudoti šie autorių turimi garsynai, kurie padėjo kurti medicininių balso komandų atpažinimo sistemą: garsynas SKAIC (10 lietuviškų skaičių pavadinimai x 100 diktorių (50 vyrų ir 50 moterų) x 20 ištariimų = 20000 ištariimų), garsynas PAVRD (100 lietuviškų pavardžių ir vardų balso įrašai x 40 diktorių x 1 ištariimas = 4000 ištariimų), garsynas SKIEM (66 skiemenų balso įrašai x 60 diktorių x 20 ištariimų = 79200 ištariimų).

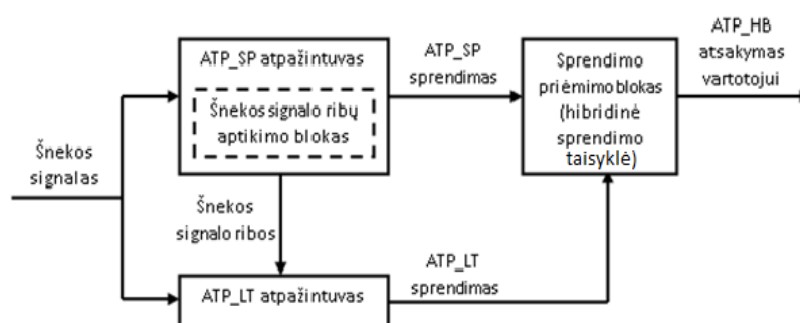
Duomenys apie vaistus buvo surinkti iš didmenininkų, apie ligonių nusiskundimus ir ligas buvo gauti iš projekto partnerio UAB „SoftDent“. Pavadinimų kirčiavimas atliktas remiantis VDU Kompiuterinės lingvistikos centro internetine kirčiavimo sistema, tačiau atsižvelgta į gydytojų naudojamą kirčiavimą.

Mokslinė prasme šiame projekte buvo siekiama išanalizuoti bent dviejų atpažintuvų panaudojimo galimybes daugiadiktoriniam lietuviškų medicininės srities balso komandų atpažinimui.

➤ Hibridinis atpažintuvas (ATP_HB)

Projekto pavadinime naudojamas žodis „hibridinis“ reiškia potencialų abiejų atpažintuvų privalumų išnaudojimą - kelių atpažintuvų panaudojimą kartu.

Sukurto Hibridinio atpažintuvo struktūra pateikta 1.5.1 paveiksle.



1.5.1 pav. Hibridinio atpažintuvo ATP_HB struktūra [24]

ATP_HB atpažintuvas sudarytas iš ATP_SP (ispanų kalbos) atpažintuvo, turinčio integruotą signalo aptikimo bloką, ATP_LT (lietuviško) atpažintuvo ir sprendimų priėmimo bloko, kuris realizuoja hibridinę sprendimų priėmimo taisyklę (-es). Šnekos signalas pirmiausia patenka į ATP_SP, kuris nustato komandos ribas ir pateikia signalo ištrauką ATP_LT atpažinimui, o nuosavą sprendimą perduoda sprendimų priėmimo blokui. ATP_LT atpažintuvas savo sprendimą taip pat perduoda sprendimų priėmimo blokui. Jei sprendimai skiriasi, sprendimų priėmimo blokas nusprendžia, kurį iš dviejų rezultatų pateikti vartotojui, kaip galutinį atsakymą.

Lygiagrečius dviejų atpažintuvų naudojimas yra prasmingas, jei, vienam atpažintuvui klystant, kitas priima teisingą sprendimą ir atvirkščiai. Nelietuviško atpažintuvo svarbiausi privalumai yra signalo aptikimo, triukšmo apdorojimo ir kt. moduliai. 2013 m. buvo atlikta išsami

atpažintuvų ATP_LT ir ATP_SP papildomumo analizė jau visiškai sukomplektuoto taikomosios srities garsyno „*Medic*“ pagrindu.

➤ **Nelietuviškas ATP_SP atpažintuvas**

Nelietuviško ATP_SP atpažintuvo panaudojimas remiasi daugiakalbio atpažinimo principais. Tinkamų fonetinių vienetų sekų (transkripcijų) lietuviškoms komandoms atpažintuvo parinkimas yra centrinis kitakalbio atpažintuvo adaptavimo („mapping“) uždavinys.

Ispaniškame atpažintuve yra 11 balsių ir 20 priebalsių fonetinių vienetų modeliai, kurie taikomi fonetinėms sekoms parinkti naudojant: ispanišką sintezatorių kai lietuviškas žodis (skiemuo ar frazė) užrašomi konkrečios kalbos atpažintuvo fonetiniais vienetais ir perklausomas sintezuotas signalas, pramoninio atpažintuvo projektavimo priemonės, kurios pasiūlomam tekstui automatiškai generuoja galimus tarimo variantus. ATP_SP lietuvių kalbai adaptuotas ispanų kalbos atpažintuvas, platinamas su Windows‘7 operacine sistema.

Ispaniško atpažintuvo ATP_SP analizei, tyrimai vykdyti, naudojant 175440 garsyno „*Medic*“ balso komandų įrašus. Atpažinimo tikslumas vertintas atskirai ligų, nusiskundimų, vaistų komandų grupėse, taip pat atliktas visų komandų „*Medic*“ atpažinimo vertinimas, kai sprendimas parenkamas iš 731 alternatyvos. Šiame tyrime gauti rezultatai leidžia padaryti svarbiausią išvadą, kad, nors „*Medic*“ garsyno komandų ispaniško atpažintuvo ATP_SP tikslumas nėra aukštas (apie 78,24%), bet mažas komandų praleidimų skaičius leidžia jį panaudoti itin svarbiai komandų aptikimo funkcijai vykdyti ir naudoti kaip papildomą sprendimo pagerinimo šaltinį hibridiniame atpažintuve.

➤ **Lietuviškas ATP_LT atpažintuvas**

Paslėptųjų Markovo modelių technologija buvo panaudota lietuviškam ATP_LT atpažintuvui, kaip vienam iš hibridinės šnekos atpažinimo technologijos komponentų, sukurti. ATP_LT atpažintuvo akustiniams modeliams sudaryti buvo naudojamas atviro kodo programinių priemonių rinkinys HTK v.3.2 (Hidden Markov Toolkit). Akustiniam modeliavimui naudotas 141 fonetinių vienetų (angl. monophones) rinkinys, atsižvelgiantis į balsių ilgumo ir kirčiuotumo savybes bei priebalsius. Fonetinių vienetų žymėjimui naudoti ASCII kodai, remiantis SAMPA-LT žymėjimu. Pradinė atpažintuvo versija apmokyta remiantis 50 val. balso įrašais.

Garsyno „*Medic*“ įrašai buvo panaudoti ir lietuviško atpažintuvo savybėms. Atlikus tyrimus buvo gauti tokie rezultatai – ligų, vaistų ir nusiskundimų komandos atpažįstamos didesniu, nei 99% tikslumu. Visų komandų iš „*Medic*“ garsyno atpažinimo tikslumas didesnis nei 99%, o ATP_LT atpažintuvo vidutinis tikslumas yra - 98,58%.

➤ **Ispaniško ir lietuviško atpažintuvų klaidų palyginimas**

Palyginus ATP_SP ir ATP_LT atpažinimo tikslumo įverčius, yra nustatyta, kad lietuviškas atpažintuvas tikslesnis, nei ispaniškas. O atlikus 1000 balso komandų atpažinimo tikslumo tyrimą palyginus „*MEDIC-731*“ ir „*MEDIC-1000*“ rezultatus stebimas proporcingas atpažinimo tikslumo sumažėjimas visiems diktoriais. Tai leidžia manyti, kad atpažintuvas ATP_LT yra

stabilus ir, didinant atpažįstamų komandų skaičių, galima tikėtis tolygaus atpažinimo tikslumo mažėjimo.

Pagrindinis atliktų tyrimų rezultatas yra ATP_LT atpažintuvo tikslus veikimas - nepadaroja nė viena atpažinimo klaida.

➤ **Atpažintuvų sujungimo galimybės**

Projekto trukmė buvo tik pusantų metų, todėl dviejų šiame projekte naudotų atpažintuvų sujungimo galimybių tyrimas liko pilnai neišnagrinėtas.

Pagal ATP_SP ir ATP_LT atpažintuvų pateiktus atpažinimo sprendimus, įrašai buvo suskirstyti į poaibius. Šie poaibiai detalai atskleidžia ATP_LT ir ATP_SP papildomumą. Jie glaudai apibūdinti 1.5.3 lentelėje pateiktoje 7 PRIEDE.

Nors atpažintuvas ATP_LT tikslumu lenkia ATP_SP, tačiau įrašų poaibyje *FT*, ATP_SP yra tikslesnis nei ATP_LT.

Projekto metu buvo suformuluota dviejų klasių – *TF* ir *FT* atskyrimo užduotis. Svarbiausias elementas hibridiniame atpažintuve - hibridinis sprendimų priėmimo blokas, sukonstruotas taikant mašininio mokymo metodą, naudojant „*Medic*“ garsyno įrašus, kai atpažintuvų ATP_SP ir ATP_LT sprendimai skyrėsi (žr. 1.5.3 lent., 7 PRIEDAS). Kiekvieną mokymo imties objektą sudarė abiejų atpažintuvų sprendimai konkrečiam garso įrašui. Mokymo imtį iš viso sudarė 35007 objektai -abiejų atpažintuvų sprendimai konkrečiam garso įrašui (*TF* ir *FT* klases atitinkantys sprendimai).

Paprasčiausia „*akla*“ sprendimo taisyklė teigianti - „*jei atpažintuvų sprendimai skiriasi, pasirink geresniojo atpažintuvo ATP_LT sprendimą*“ - veikia $33650/35007 \cdot 100\% = 96,12\%$ tikslumu. Gauta hibridinė sprendimo taisyklė bus naudinga tik tada, jei viršys šios „*aklos*“ taisyklės tikslumą.

Kiekvienas mokymo imties objektas buvo apibrėžtas 70 požymių, kurie paaiškinti 1.5.4 lentelėje pateiktoje 7 PRIEDE.

Pagrindiniai požymiai, pateikti 1.5.4 lentelėje, yra „*lt_prob*“ ir „*sp_prob*“, pavydžiui, „*sp_prob*“ – ATP_SP priimto sprendimo panašumo įvertinimas, padaugintas iš 1000, kuo panašumo įvertis artimesnis vienetui, tuo su didesne tikimybe atpažintuvas priima sprendimą. Požymiai „*lt_delta_prob*“ ir „*sp_supp*“ suformuoti remiantis tuo, kad ATP_LT pateikia nuo 1 iki 3 atsakymų, išdėstydamas juos prioriteto mažėjimo tvarka. Jei pirmasis prioritetinis sprendimas yra teisingas - frazė laikoma teisingai atpažinta. Kitų požymių kiekį nulėmė lietuviškų raidžių skaičius, jie yra papildomi požymiai. Tikslas buvo sudaryti kuo daugiau požymių, o jų įtaka bus nustatyta tolimesniuose tyrimuose. Detalesnė informacija apie projektą ir jo rezultatus pateikta [24].

1.6 Kitakalbio atpažintuvo panaudojimo principai

Automatinė šnekamosios kalbos atpažinimo sistema turi būti pritaikyta pasirinktajai kalbai atpažinti. Norit pritaikyti nelietuviškos (kitakalbės) automatinės kalbos atpažinimo sistemą

lietuvių kalbai reikia atsižvelgti į sistemos savybes, kurios palengvina kitakalbio atpažintuvo pritaikymą norimai kalbai atpažinti.

Didžiausias nelietuviškos atpažinimo sistemos privalumas yra suderinamumas su kitais programiniais standartais. Šis privalumas yra esminis jungiant į visumą šnekos atpažinimo modulius su kitomis techninėmis bei programinėmis priemonėmis. Šiuose atpažintuvuose yra įtraukta nemažai šnekos signalo apdorojimo modulių (signalo aptikimo, triukšmo apdorojimo ir kt.). Siekiant nelietuvišką šnekos atpažinimo sistemą pritaikyti lietuviškoms komandoms atpažinti, centrinis atpažintuvo adaptavimo kitai kalbai atpažinti uždavinys yra tinkamų fonetinių vienetų sekų parinkimas. Šiuo atveju atpažintuvo adaptavimo tikslas yra rasti, kokios kitos kalbos akustiniai modeliai geriausiai aprašo lietuviškas fonetinių vienetų savybes. Pavyzdžiui, plačiausiai taikomų anglų ir ispanų kalbų, adaptavimui lietuvių kalbai, fonetiniai modeliai skiriasi. Angliška atpažintuve naudojami 18 balsių (ar dvigarsių) ir 27 priebalsių turintys fonetiniai modeliai, o ispaniška atpažintuve naudojami 11 balsių ir 20 priebalsių turintys modeliai. Kitų kalbų akustinių modelių adaptavimas lietuvių šnekamosios kalbos atpažinimui vadinamas teksto transkribavimu.[12]

Transkribavimas paprastai tariant yra raidžių keitimas į garsus. S. Laurinčiukaitė išsamų transkripcijos, kaip elemento, apibrėžimą pateikia, kad „tai yra nuosekli kalbos vienetų seka, sudaroma tik iš apibrėžtos kalbos vienetų aibės elementų“ [4]. Populiariausi du transkribavimo metodai yra: transkribavimas kuris yra pagrįstas žodynu ir transkribavimas pagrįstas taisyklėmis. Šie metodai yra plačiai taikomi šnekos atpažinimo sistemose, o kuris metodas yra taikomas kuriai sistemai priklauso nuo pačios kalbos [7].

Nelietuviškos šnekos atpažinimo sistemos tinkamumui ir tikslumui įvertinti reikia sudaryti galimas fonetines sekas. Fonetinėms sekoms parinkti yra patogų taikyti kitų kalbų sintezatorius, kai įvestas lietuviškas žodis (skiemuo ar frazė) užrašomi konkrečios kalbos atpažintuvo fonetiniais vienetais ir perklausomas sintezuotas signalas. Kitų kalbų laisvai platinamus sintezatorius galima rasti internete. Dar vienas ir tiksliausias fonetinių sekų parinkimo būdas yra pramoninių atpažintuvų projektavimo priemonių panaudojimas, kurios pasiūlomam tekstui automatiškai generuoja galimus tarimo variantus, pvz. Microsoft balso serveris (*Microsoft Speech Server* (MSS'2007)).

Microsoft balso serveris yra interaktyvus telefoninis auto atsakiklis, integruotas kartu su paketu „*Visual Studio 2005*“. Pagrindinė iš „MSS'2007“ priemonių yra „VoIP“. Ji leidžia vartotojams patalpinti ir priimti užklausas per internetą. Balso serveris gali priimti „VoIP“ užklausas be jokios papildomos programinės ar techninės įrangos.

„MSS'2007“ pilnai palaiko anglų, vokiečių, ispanų, prancūzų, japonų ir kinų kalbas. „MSS'2007“ balso serveryje naudojama UPS (*Universal Phone Set*) transkribavimo sistema.

UPS - yra koduotas simbolių rinkinys, kuris yra sudarytas IPA (International Phonetic Alphabet) pagrindu ir yra papildytas SAPI (Speech Application Programming Interface) ženklais.

UPS gramatikose kai kurie žodžiai gali būti papildomi įvairiais žymėjimais žyminčiais tarimą, kirtį ir kt. UPS simbolių sekos yra aprašytos ASCII (American Standard Code for Information Interchange) ženklų eilutėmis [24]. Balso komandos atpažinimo gramatikos pateikia sistemai įvairius tikėtinus jos atpažinimo variantus, nes vienos balso komandai galima priskirti net keletą skirtingų transkripcijų.

Peržvelgus V. Rudžionio atlikto ATP_EN ir ATP_SP palyginimo tyrimo rezultatus, išsiaiškinta, kad ispaniško atpažintuvo atveju galima greičiau gauti tikslesnes transkripcijas lietuviškoms balso komandoms atpažinti, nes ispanų fonetinė sistema žymiai artimesnė lietuviškai, nei angliška, bet kartu turi savo ypatumų (nėra garso „š“ ir pan.) [24].

Apibendrinant galima teigti, kad kruopščiai ir atidžiai atlikus norimų atpažinti balso komandų transkribavimo procesą, galima tikėtis gauti daug geresnę atpažinimo tikslumą ir su kitakalbiu atpažintuvu.

1.7 Informacijos kodavimas

Didžioji dalis svarbios informacijos – banko sąskaitų numeriai, pašto kodai, asmens kodai, automobilių registracijos numeriai ir kt. – yra koduojami tam tikrais skaičių ar raidžių ir skaičių deriniais, taip supaprastinant informacijos paiešką duomenų archyvuose. Kiekvienas kodas turi savo kodavimo sistemą kuria remiantis yra koduojama reikalinga informacija. Pavyzdžiui, asmens kodas - unikali vienuolikos dešimtinių skaitmenų seka, žyminti asmenį ir naudojama duomenims apie jį kaupti valstybės tvarkomose duomenų bazėse. Asmens kodo pirmasis skaitmuo rodo gimimo šimtmetį ir asmens lytį (1 – XIX a. gimęs vyras, 2 – XIX a. gimusi moteris, 3 – XX a. gimęs vyras, 4 – XX a. gimusi moteris, 5 – XXI a. gimęs vyras, 6 – XXI a. gimusi moteris ir t.t), tolesni šeši – asmens gimimo metų du paskutiniai skaitmenys, mėnuo (du skaitmenys), diena (du skaitmenys), dar tolesni trys skaitmenys – tą dieną gimusių asmenų eilės numeris ir paskutinis – iš kitų skaitmenų išvedamas kontrolinis skaičius.

Baigiamasis magistro darbas yra tiesiogiai susijęs su projektu „*Hibridinė atpažinimo technologija balso sąsajai*“, todėl išsamesnei analizei yra pasirinkta ligų kodavimo sistema.

Iš viso yra 15 000 ligų pavadinimų, kol kas dar nėra sukurta atpažintuvo kuris sugebėtų atpažinti tokį kiekį lietuviškų ligų pavadinimų. Todėl ieškant išeities kaip supaprastinti ligų atpažinimą ir pritaikyti praktikoje buvo pastebėta, kad dėl patogumo ir praktiškumo vartotojai (gydytojai) nuo 1992 m. pildant formas naudoja ne pilnus ligų pavadinimus, bet jų kodus, kurie yra aprašyti TLK – 10 - AM – *Tarptautinė statistinė ligų ir sveikatos sutrikimų klasifikacijos dešimtasys pataisytas ir papildytas leidimas Australijos modifikacija*, kurį parengė nacionalinis medicininės klasifikacijos centras (NCCH).

TLK-10-AM ligų klasifikacijos pagrindas yra triženklis kodas, kurį pasaulio sveikatos organizacija (PSO) reikalauja pateikti įvairiuose tiriamuosiuose ir moksliniuose darbuose ir pranešimuose. Pagrindinis triženklių kodų rinkinys yra išplėstas iki keturių ir penkių ženklų. Dėl to yra galimybė nustatyti svarbius specifinius nozologinius vienetus (nozologinis vienetas yra

pavyzdžiui liga, arba sutrikimas, arba sindomas), taip pat duomenis galima pristatyti plačiose grupėse, suteikiant naudingą ir suprantamą informaciją.

Pagrindinė sisteminio ligų sąrašo dalis – pagrindinių ligų klasifikacija, kurią sudaro 22 skyriai. Pirmasis TLK-10-AM kodo simbolis yra raidė. Kiekviena raidė yra priskiriama tam tikram skyriui, išskyrus raides: D, H. Antrasis ligos kodo simbolis yra skaičius, atitinkamai pagal ligos klasifikaciją yra ir skaičių simbolių. Raidės, grupių tipai ir jas atitinkantys skyriai išvardyti 1.7.1 lentelėje pateiktos 8 PRIEDE [19].

1.8 Duomenų analizės sistemų apžvalga

Šiuo metu duomenų analizės sistemų pasirinkimas yra labai platus. Sistemą galima rinktis iš daugiau kaip 600 komercinių ir atvirojo kodo duomenų analizės sistemų. 2008-2010 metais atlikto, realiuose projektuose naudojamų duomenų, analizės paketų paklausos tyrimo, rezultatai parodė, kad dažniausiai naudojami yra atvirojo kodo paketai RapidMiner, R, KNIME, WEKA, Orange ir kt., o komerciniai paketai Excel ir MATLAB naudojami tik kaip pagalbiniai paketai [25].

Vienos išsamios šešių atvirojo kodo duomenų analizės sistemų apžvalgos metu, teigiama, kad nėra geriausios duomenų analizės sistemos, bet vartojimui, kaip alternatyvūs, siūlomi 4 duomenų analizės sistemų paketai - RapidMiner, KNIME, R ir WEKA [26]. Kitoje atvirojo kodo duomenų analizės sistemų apžvalgoje, kurioje nagrinėti 12 duomenų analizės paketų, geriausius įvertinimus gavo KNIME, YALE (senesnė RapidMiner versija), Orange, WEKA ir AlphaMiner [27]. Dar viename duomenų analizės sistemų tiriamajame darbe buvo analizuoti devynių tipų klasifikavimo objektai ir šešių tipų klasifikatoriai – tyrimų rezultate geriausiai įvertintas WEKA duomenų analizės paketas. WEKA sistemoje įgyvendintas didelis kiekis duomenų analizės metodų bei jų modifikacijų. [28]

WEKA paketas pasižymi didele klasifikatorių įvairove. Norint eksperimentiniams tyrimams pasirinkti efektyviausią klasifikatorių, reikia apžvelgti ankstesnius tiriamųjų analizių darbus. Viename tyrime kaip populiariausi nagrinėjami klasifikatorių tipai - *k* artimiausių kaimynų (angl. K-Nearest Neighbors kNN), *Bajeso* (angl. Naive Bayes NB), *daugiasluoksnis neuroninis tinklas* (angl. Multilayer Perceptron MP), *atraminių vektorių klasifikatorius* (angl. support vector machine SVM), *sprendimų medis* (angl. decision tree), kurio populiariausi klasifikatoriai - *atsitiktinis miškas* (angl. random forest RF) ir *C4.5* [29]. Renkantis klasifikatorių galima remtis tyrimu, kuriame apžvelgti dešimt pasaulyje populiariausių duomenų analizės algoritmų, tarp jų patenka *C4.5*, *NB*, *kNN*, *SVM*, *AdaBoost*, *CART* (*Classification and Regression Trees*) klasifikavimo algoritmai bei keletas klasterizavimui skirtų algoritmų [30].

2. METODINĖ DALIS

2.1 Garsynų paruošimas atpažinimo tikslumo tyrimams

Norint ištirti balso komandų atpažinimo tikslumą pirmiausia reikia turėti tinkamai paruoštus daugiadiktorinius lietuviškų skaičių ir vardų balso komandų garsynus. Skaičių ir vardų balso komandos buvo sudiktuotos ir įrašytos naudojant MS DOS operacinės sistemos aplinkoje veikiančią programą „*inp_sr16.exe*“.

Daugiadiktorinis lietuviškų skaičių garsynas yra suskaidytas į du atskirus garsynus pavadintus - „*Skaiciai_20dikt_VOC*“ ir „*Skaiciai_New_10dikt_VOC*“. Garsyną „*Skaiciai_20dikt_VOC*“, sudaro dvidešimties diktorių balso įrašai, o garsyną „*Skaiciai_New_10dikt_VOC*“ sudaro dešimties diktorių balso įrašai. Šiuos garsynus sudaro iš viso trisdešimt diktorių - dvidešimt trys moterys ir septyni vyrai. Kiekvienas diktorius diktavo skaičių pavadinimus nuo nulio iki devynių, iš viso dešimt skirtingų skaičių. Skaičių pavadinimus talpinantys katalogai pavadinti kataloge sudėtų ištariamų pavadinimu, jei kataloge yra skaičiaus „*Trys*“ ištariai, tai katalogas pavadintas „*TRYS*“ ir pan. Garsyną sudarantis katalogų sąrašas bei jame esančios skaičių balso komandos pateikiami 2.1.1 lentelėje 1 PRIEDE.

Daugiadiktorinis vardų garsynas yra pavadintas – „*Vardai_10dikt_VOC*“. Jį sudaro dešimties diktorių balso įrašai - penki moteriški ir penki vyriški balsai. Vardų garsyną sudaro 26 skirtingų vardų balso komandos - atitikmenys kiekvienai lotyniškos abėcėlės raidei. Vardus talpinantys katalogai pavadinti kataloge sudėtų ištariamų vardu, tarkime, jeigu kataloge yra „*Austėja*“ ištariai, tai katalogas pavadintas „*AUSTEJA*“ ir pan. Garsyną sudarantis katalogų sąrašas bei jame esančios vardų balso komandos pateikiami 2.1.2 lentelėje 1 PRIEDE.

Tiek skaičių tiek vardų balso komandos buvo diktuojamos po 20 kartų, pradedant skaičiuoti nuo 0 iki 19 ištariamų. Sudarant garsyną pasirenkama kataloge esanti įrašinėjimo komanda ir pradedama diktuoti komanda. Įdiktavimai buvo daromi tylioje aplinkoje, bet ne profesionaliose garso įrašų studijose, buvo naudojama skaitmeninė įrašymo įranga bei profesionalus mikrofonas. Diktuojant buvo labai svarbu pabrėžti pradžią ir pabaigą, kadangi siekiama tikslaus ir kokybiško atpažinimo, o dėl tam tikrų techninių nesklandumų ar neaiškaus ištario gali pasitaikyti netikslumų. Pasitaikius tokiems netikslumams kiekviename kataloge yra programėlė, leidžianti perrašyti blogą įrašą, nurodant jo numerį, t.y. nereikia perrašinėti 20 kartų iš naujo dėl vieno netikslumo.

Siekiant užtikrinti garsyno įrašų kokybę, įdiktavus visas komandas, buvo atliekamas pirmasis ir pakartotinis patikrinimai, ieškant klaidingai įdiktuočių įrašų, klaidos buvo taisomos panaikinant triukšmus, naujai įdiktuojant neaiškiai ištartas komandas.

Diktorių balso įrašai yra saugomi atskiruose kataloguose, kurių pavadinimai parodo asmens tapatybę bei lytį. Pirmoji katalogo raidė parodo asmens lytį, **F** nuo žodžio „*female*“ - moteris, o

M nuo žodžio „male“ – vyras. Toliau seka trys pirmosios vardo bei pavardės raidės, pavyzdžiui Kastyčio Ratkevičiaus katalogo pavadinimas - „MKASRAT“. Kiekviename iš šių katalogų yra atskirai sudiktuotų, komandų katalogai su tam tikros komandos pavadinimu. Komandos pavadinimo kataloge yra:

- „*mano.voc*“ - audio failas (16000Hz, 16 bitų mono audio formatas) ir tas pats failas „*.wav“ formate, „*mano.wav*“;
- „*mano.zgl*“ - anotacijų failas, kuriame yra pateikti kiekvieno įrašo pradžios ir pabaigos adresai;
- „*mano.txt*“ - audio failo turinį aprašantis pagalbinis failas, kuriame nurodyta kiek kartų yra išstartas tam tikras skaičius;
- kiti pagalbiniai failai, naudoti garsyno rinkimo ir tikrinimo metu;

Tiriamasis lietuviškų skaičių garsynas sudarytas net iš šešių tūkstančių (6000) skirtingų balso įrašų. Vardų garsyną iš viso sudaro penki tūkstančiai du šimtai (5200) balso įrašų.

Tiriamosios dalies 3.8 skyriuje yra aprašytas šnekos atpažintuvų sujungimo galimybių tyrimas, kuriam buvo naudojamas projekto „Hibridinė atpažinimo technologija balso sąsajai“ metu surinktas medicininis garsynas „*Medic*“, kurį sudaro sudaro 175440 garso įrašai = 731 medicinoje vartojamų terminų x 12 diktorių (5 moterys ir 7 vyrai) x 20 tos pačios komandos tarimų. Medicininės komandos buvo susegmentuotos į 3 garsynų grupes: VAISTAI, NUSISKUNDIMAI ir LIGOS. Visa garsyno struktūra pateikta 1.5.1 lentelėje 1 PRIEDE.

2.2 Gramatikų paruošimas atpažinimo tikslumo tyrimams

Balso komandų atpažinimo tyrimui naudojant kitakalbį atpažintuvą, buvo pasirinktas ispanų kalbos atpažintuvas, apmokytas anglų kalba.

Turimiems garsynams aprašytiems 2.1 skyriuje, buvo atliktas lietuviškų balso komandų fonetinių transkripcijų adaptavimas ispanų kalbos atpažintuvui.

Garsyno paruošimas analizei yra gramatikų ir transkripcijų kūrimas, kuris siejamas su įdiktuoju garsynu ir jo atpažinimo gerinimu. Gramatikų kūrimas yra svarbus tuo, kad gramatika – tai failas, kuriame nurodomos komandų transkripcijos ir tam tikros komandos, nurodančios, ką reikia išvesti į ekraną įvykus atpažinimui.

Atpažinimo tyrimams su kitakalbiu atpažintuvu paruošiama pirminė gramatika, žodinė (papildyta naujomis transkripcijomis) gramatika bei gramatika, perrašyta UPS (Universal Phone Set) transkripcijomis.

Tyrimai vykdomi vienodomis sąlygomis, visi testavimai atliekami naudojant ispanų kalbos balso atpažintuvą (Microsoft Speech recognizer 8.0 for Windows), apmokytą anglų kalba. Išsamiam tyrimui buvo išbandomi įvairūs profiliai: kompiuterio numatytasis neapmokytas profilis, diktorių balsais apmokyti profiliai.

Tam, kad paruoštas komandų garsynas būtų naudojamas atpažinime, būtina paruošti kiekvienos jį sudarančios komandos gramatiką. Be gramatikos būtų neįmanoma vykdyti kalbos atpažinimo, nes ji yra reikalinga kiekvienai kalbos atpažinimo sistemai. Gramatikos sudarymas yra labai svarbi dalis, nes tai yra specialiai paruoštas „*.grmx“ failas, kuriame aprašoma, tam tikros komandos (žodžio) ištarimas - transkripcija, ir ką reikėtų išvesti į ekraną įvykus atpažinimui.

Pirminės gramatikos sudarymui transkripcijos buvo sugalvotos, nesiremiant jokiais pagalbiniais transkripcijų generavimo įrankiais, naudota tik po vieną transkripciją, beveik visos komandos aprašytos taip, kaip tariamos įprastai, tik nenaudojami lietuviški simboliai, arba pakeitus lietuviškų simbolių kitokiais. Nosinginės ir ilgiosios raidės virto paprastomis trumposiomis atitinkamomis raidėmis, pavyzdžiui, raidė Š virto į SH, raidė J → HY, Č → CH, C → TS ir kt. Tokiu būdu siekiama patikrinti, kiek ir koku būdu galima pagerinti atpažinimo kokybę. Lietuviškų skaičių ir vardų pilni pirminės gramatikos failų turiniai pateikiami 2 PRIEDE. Žemiau pateikiama ištrauka iš pirminės Lietuviškų vardų ir skaičių balso komandų gramatikų failų:

```
1. <item>
  <one-of>
  <item>ehiyamahntahs</item>
  <item>eimantas</item>
  </one-of>
  <tag>$. _value = "eimantas"</tag>
</item>
2. <item>
  <one-of>
  <item>sheshii</item>
  <item>chechii</item>
  </one-of>
  <tag>$. _value = "6sesi"</tag>
</item>
```

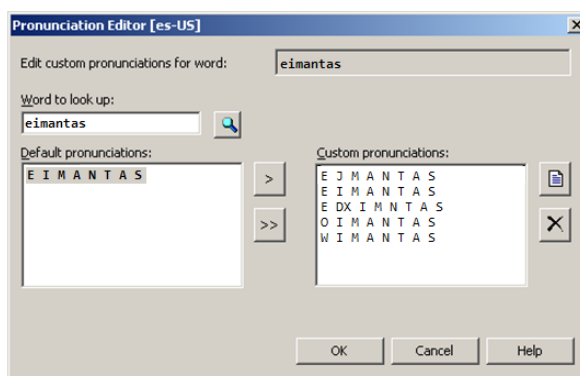
Antroji gramatika sudaroma tik vardų garsynui, pildant ją naujomis transkripcijomis. Joms sudaryti taikomas internete esantis nemokamas ispanų kalbos sintezatorius <http://text-to-speech.imtranslator.net> (kūrėjai *Smart Link Corporation*). Šio sintezatoriaus veikimas pagrįstas *text-to-speech* principu. Juo naudojantis, galima peržvelgti, tam tikrų garsų variacijų skambesį, galima nustatyti, kaip tikslingiau pakeisti fonetiškai netinkamus garsus. Šis sintezatorius suteikia galimybę pakeisti ištarimo greitį, todėl galima geriau nustatyti, ar garsai yra fonetiškai tinkami ir ar atitinka lietuvišką ištarimą. Vardų pilnas pirminės gramatikos su papildomomis transkripcijomis failo turinys pateikiamas 2 PRIEDE. Taip atrodo ištrauka iš patobulintos gramatikos:

```
1. <item>
  <one-of>
  <item>ehiyamahntahs</item>
  <item>eimantas</item>
  <item>eihmantas</item>
  <item>eigmantas</item>
  <item>eihmanhtas</item>
  <item>eihmanhtahss</item>
  <item>aeihmanthahs</item>
```

```
</one-of>  
<tag>$. _value = "eimantas"</tag>
```

```
</item>
```

Trečioji gramatika vardų garsynui buvo gauta naudojant „MS Speech Server 2007“ su integruotu „MS Visual Studio 2005 paketu“. Transkripcijos buvo automatiškai sugeneruotos UPS. MSSS serveris pats sugeneruoja UPS (Universal Phone Set) transkripcijų variantus pagal fonemų lentelę (žr. 3 PRIEDAS) ir pakeičia gramatiką, papildydamas ją PRON (angl. - pronunciation, lt. – tarimas) atributu (žr. 2.2.1 pav.).



2.2.1 pav. UPS transkripcijų įterpimas į gramatiką

2.2.1 paveiksle pateikiamas langas, kuriame kairėje pusėje matomos sugeneruotos UPS transkripcijos yra perkeliamos į dešinę, taip prie komandų priskiriant PRON atributą ir papildant gramatiką. Vardų pilną UPS sugeneruotą gramatikos failo turinį galima matyti 2 PRIEDE.

Taip atrodo trečiosios gramatikos sudarytos su „MS Speech Server 2007“ ištrauka:

```
1. <item>  
  <?MS_Grammar_Editor GroupWrap?>  
  <token sapi:pron="E J M A N T A S;E I M A N T A S;E D X I M A N T A S;O I M A N T A S;W I  
M A N T A S">eimantas</token>  
  <tag>$. _value = "eimantas"</tag>  
</item>  
<item>  
  <?MS_Grammar_Editor GroupWrap?>  
  <token sapi:pron="E I L J M A X N T A S;W I L J M A N T A S;E D X I L J M A N T A S;O I L J M  
A N T A S;E I L J M A N T A S">ehiyamahntahs</token>  
  <tag>$. _value = "ehiyamahntahs"</tag>  
</item>
```

Vardų atpažinimo tikslumo tyrimų metu naudojamos visos gramatikos atskirai ir stebimi komandų atpažinimo tikslumo rezultatai. Vardų atpažinimo tikslumo tyrimų tikslas yra parinkti tas transkripcijas su kuriomis bus išgauta pats aukščiausias atpažinimo tikslumas. Skaičių atpažinimo tikslumo tyrimų metu naudojama tik pirminė gramatika.

2.3 Kitakalbis atpažintuvas ir jo panaudojimas

Microsoft Windows operacinėje sistemoje yra įdiegta šnekamosios kalbos atpažinimo sistema - „Microsoft Speech Recognizer“. Ši sistema gali atpažinti dauguma plačiausiai pasaulyje

vartojamų kalbų – anglų, ispanų, prancūzų ir kt., tačiau lietuvių kalbos atpažinimui ji dar nepritaikyta. Pasitelkus įvairius kitakalbio atpažintuvo, pritaikymo kitai kalbai atpažinti, metodus (žr. skyrių 1.6), kitakalbį atpažintuvą galima taikyti ir lietuvių kalbos atpažinimui.

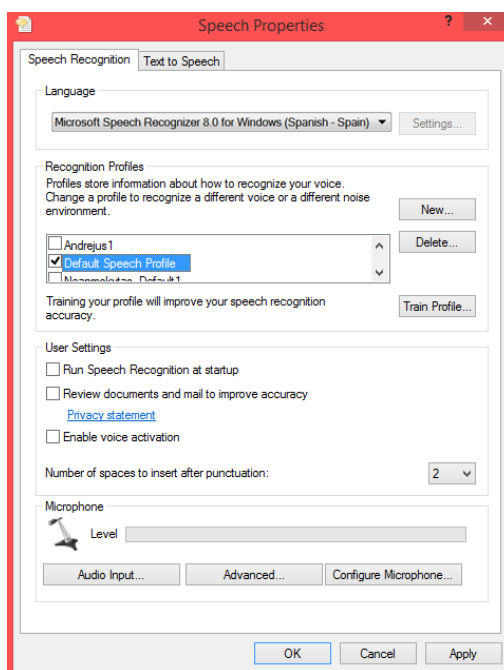
Kitakalbis atpažintuvas taikomas balso komandų atpažinimui, todėl buvo atliekami tyrimai taikant pasirinktą kitakalbį atpažintuvą, siekiant kuo geriau jį pritaikyti šiame darbe naudojamiems garsynams atpažinti ir pritaikyti praktiškai „*Ligų kodų atpažinimo programoje*“ (žr. skyr. 3.9).

Atlikus literatūros analizę ir remiantis ankstesniuose šnekamosios kalbos atpažinimo tyrimuose gautais rezultatais, naudojant ispanišką kalbos atpažintuvą yra gaunami žymiai geresni šnekamosios kalbos atpažinimo rezultatai, nei naudojant kitų kalbų atpažintuvus lietuviškoms balso komandoms atpažinti. Todėl šiame darbe lietuviškų skaičių pavadinimų ir vardų atpažinimo tikslumui tirti buvo naudojamas ispaniškas kalbos atpažintuvas – „*Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)*“ – esantis Microsoft Windows operacinėje sistemoje.

„*Microsoft Speech Recognizer 8.0 for Windows*“ šnekamosios kalbos atpažinimo sistema leidžia šnekos atpažinimui naudoti skirtingus atpažinimo profilius. Atpažinimo profiliai įtakoja atpažinimo kokybę, todėl su „*Microsoft Speech Recognizer 8.0 for Windows*“ atpažintuvu galima taikyti įvairius profilius – MS Windows numatytąjį savaime apsimokantį (*Default Speech Profile*), visai naują neapmokytą profilį (*default*) ir diktorių balsais apmokytus profilius.

MS Windows operacinėje sistemoje norint susirasti, naudoti, apmokyti ir pasirinkti norimus šnekamosios kalbos atpažinimo sistemos parametrus, reikia susirasti pagrindinį atpažinimo sistemos langą tokia veiksmų seka: atidaromas „*Control Panel*“ langas >> pasirenkame „*Speech recognition*“ >> spaudžiame „*Advanced speech options*“ >> „*Speech recognition*“.

2.3.1 paveiksle pavaizduotas atsivėręs, pagrindinis šnekamosios kalbos atpažinimo sistemos parametrų nustatymo langas.



2.3.1 pav. Pagrindinis šnekos atpažinimo sistemos parametrų nustatymo langas

Pagrindiniame šnekamosios kalbos atpažinimo sistemos parametrų nustatymo lange skiltyje „*language*“ pasirenkamas ispanų kalbos atpažintuvas – „*Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)*“. Jeigu pasirenkamų kalbų skiltyje „*language*“ ispanų kalbos atpažintuvo nėra, reikia atnaujinti (*Windows update*) Windows operacinę sistemą, ir įsidiegti ispanų kalbą. Norint sukurti naują profilį, skiltyje „*Recognition profile*“ spaudžiame parinktį „*New*“, įvedame profilio pavadinimą (pvz. *Andrejus1*) spaudžiame parinktį „*OK*“, šių veiksmų rezultate sukuriamas naujas neapmokytas ispanų kalbos atpažinimo profilis. Norint apmokyti sukurtą naują atpažinimo sistemos profilį individualiam diktoriui pasirenkame parinktį „*Train profile*“. Atsivėrusiame lange spaudžiame parinktį „*next*“ ir pradedame kalbos atpažinimo sistemos apmokymą individualiam balsui – aiškiai ir taisyklingai tariame atsivėrusiame lange MS Windows sistemos pateikiamus angliškus sakinius, taip suteikdami sistemai reikiamą individualią akustinę ir vokalinę informaciją pasirinktam kalbos atpažinimo profiliui.

Iš baigiamojo bakalauro darbo metu darytų tyrimų su *Microsoft Windows Speech Recognition* profiliais, yra pastebėta, kad su skirtingais profiliais gaunami skirtingi balso komandų atpažinimo tikslumo rezultatai. Siekiant išsiaiškinti ispaniško atpažintuvo veikimo principus ir profilių įtaką atpažinimo tikslumui, pirmiausia buvo sukurti ir individualiai apmokyti, vardų ir skaičių balso komandų atpažinimo tikslumo tyrimams naudojami, atitinkami profiliai.

Vardų garsyno atpažinimo tikslumui tirti naudojami:

- Neapmokytas profilis „*Default_Neapmokytas1*“;
- Apmokytas diktorės Ivetos profilis „*Iveta1*“;

Skaičių garsyno atpažinimo tikslumui tirti naudojami:

- Apmokytas diktoriaus Andrejaus profilis „*Andrejus1*“;

Microsoft Speech Recognizer 8.0 for Windows kalbos atpažinimo sistemoje yra įdiegti du kalbos atpažinimo režimai – sinchroninis (*synchronous - single*) ir asinchroninis (*asynchronous - multiple*). Sinchroninis darbo režimas veikia žodžio galo ir pradžios adresų principu - žodis suskaidomas dalimis, kiekviena dalis turi savo adresą, pagal kurį atpažinimo testavimo metu žodis yra randamas. Asinchroninis darbo režimas paremtas pauzių tarp žodžių veikimo principu. Atpažinimo tikslumo testavimo metu programa ieško pauzių - tylos, kuri suprantama kaip kito žodžio pradžia, todėl labai svarbu, kad tarp ištarimų būtų aiškios pauzės, kitu atveju, testas gali kelis ištarimus suprasti kaip vieną, o dėl to suprastės atpažinimo kokybė.

Taigi naudojant ispanų kalbos atpažinimo sistemai *Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)* sukurtuosius profilius buvo atliktas vardų ir skaičių daugiadiktoriinių garsynų (žr. 1 PRIEDAS) atpažinimo testavimas. Pradiniam testavimui buvo naudojama šnekamosios kalbos atpažinimo testavimo programa. Programa testavo garsyno atpažinimo tikslumą su skirtingais *Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)* šnekos atpažinimo sistemos profiliais.

2.3.1 Kitakalbio atpažintuvo panaudojimas vardų balso komandų atpažinimui

Vardų atpažinimo tikslumo tyrimui naudojamas ispanų kalbos atpažintuvas kuriam turi būti parenkami atpažinimo profiliai. MS Windows atpažintuve yra galimybė naudoti visai neapmokytą profilį arba apmokytą diktorius balsu profilį. Todėl pirmiausia reikia iširti atitinkamų profilių įtaką balso komandų atpažinimo kokybei. 2.3 skyriuje nurodyta tvarka lietuviškų vardų atpažinimo tikslumo tyrimui su ispanų kalbos atpažintuvu buvo sukurti du skirtingi profiliai, vienas iš tyrimo naudojamų profilių yra neapmokytas profilis „*Default_Neapmokytas1*“, kurio taikymui nereikia atlikti jokių papildomų nustatymų, jį tereikia sukurti, kad tyrimui būtų naudojamas dar nenaudotas profilis. Turint neapmokytą profilį kuriamas kitas - apmokytas profilis. Jo kūrimas remiasi ta pačia tvarka kaip ir neapmokyto profilio kūrimas, tik jis dar yra apmokomas. Profilis pavadintas jį apmokiusio diktorius vardu – „*Iveta1*“. Atlikus minimalius profilių paruošimo darbus pradėti profilių įtakos balso komandų atpažinimui tyrimai su atitinkamais profiliais.

Vardų atpažinimo tikslumo tyrimui su skirtingais profiliais naudojami 26 skirtingų lietuviškų vardų balso komandos, įdiktotos 10 diktorių. Vardų garsyno įdiktavimo ir kūrimo procesas yra aprašytas 2.1 skyriuje.

Atlikus tyrimą ir peržvelgus rezultatus ir išrinkus penkias blogiausiai atpažintas vardų komandas, buvo atliekamas pradinių gramatikų papildymas naujomis blogiausiai atpažintų komandų transkripcijomis ir tyrimas tęsiamas toliau nurodyta tvarka.

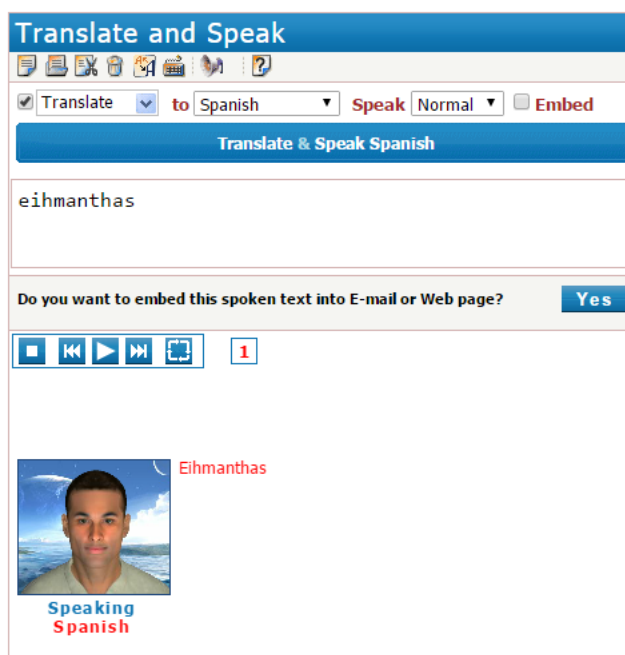
Transkripcija yra tikslus kalbinių vienetų (žodžių) fonetinis užrašymas. Kalbos atpažinime transkribavimas yra reikalingas dėl kalbų įvairovės, nes pagrindiniai atpažintuvų kūrimo principai yra skirti populiariausioms pasaulio kalboms, o kitoms pasaulio kalboms tenka tiesiog prisitaikyti prie to kas yra jau sukurta. Transkripcijos padeda kitakalbiams atpažintuvams lengviau atpažinti kitas kalbas. Parinkus tinkamas transkripcijas galima pasiekti ypač gerų atpažinimo rezultatų. Tai norime pritaikyti ir mūsų pradinėms transkripcijoms, kad galėtumėme pasiekti dar aukštesnių atpažinimo rezultatų.

Atsižvelgiant į ankstesnius tyrimus naujų transkripcijų sudarymas ir pritaikymas reikalingai kalbai yra labai naudingas. Atliekant kalbos atpažinimo tikslumo tyrimą su naujomis transkripcijomis naudojamas ispanų kalbos atpažintuvas, todėl norint pagerinti atpažinimo tikslumą, reikėjo išrinkti 5 blogiausiai atpažįstamas vardų balso komandas ir joms sugalvoti naujas fonetines transkripcijas, kurios atitinka ispanų kalbos abėcėlės fonetines ypatybes.

Norint pagerinti 5 blogiausiai atpažįstamų balso komandų atpažinimą, buvo sudaryta po 5 naujas transkripcijas kiekvienai blogiausiai atpažįstamai komandai, viso vardų garsynui buvo

pritaikytos 25 naujos transkripcijos. Nesant konkrečios metodikos transkripcijų sudarymui, buvo naudojami įvairūs metodai:

- Morfologinis – buvo nepaisoma gramatikos, priebalsių asimiliacijos. Vietoje to parenkami ženklai, kurie geriausiai atitinka ištarimą. Šiuo atveju „ch“ buvo keičiama į „j“ ar „t“, pvz.: *Zacharijus – Zajarijus, Zacharijus – Zahthaarihyuhs*.
- Fonetinis – atsižvelgiant į tam tikrus ispanų bei lietuvių kalbos fonetinius panašumus, kai kurie ženklai buvo koreguojami pagal tiksliausią fonetinį sąskambį. Tai pat reikėjo atsižvelgti ir į kalbų skirtumus – kai kurie garsai ispanų kalboje tariami visiškai kitaip negu lietuvių kalboje. Bandymai buvo atliekami naudojantis laisvai internete prieinamu sintezatoriumi (žr. 2.3.2 pav.) - <http://imtranslator.net/translate-and-speak/#>.



2.3.2 pav. Ispaniško sintezatoriaus internetinė sąsaja

Ispaniškas sintezatorius leidžia greitai ir lengvai patikrinti norimą transkripciją, su *Text-To-Speech* parinktimi leidžia perklausyti užrašytos transkripcijos ištarimą pasirinktu tarimo greičiu, taip įvertinant ar transkripcijos ištarimas tikrai tapatus su lietuvišku žodžiu. Iš to pastebėta, kad garsas *j* tariamas kaip *ch*.

- Intuityvusis – kaip jau buvo minėta, nėra konkrečios metodikos transkripcijų sudarymui, todėl apjungus morfologines ir fonetines žinias ir tikrinant ištarimus ispanų kalbos sintezatoriumi, buvo:
 - dirbtinai ilginami skiemenys pridėdant pasikartojančių raidžių (*Zahthaarihyuhs - Zacharijus*);
 - įterpiami priebalsiai *h* prieš arba po balsių siekiant juos prailginti, sukietinti arba suduslinti jų tarimą (*Iygrehkas – Ygrekas*);
 - ilgieji balsiai buvo keičiami trumpaisiais, nes kaip pastebėta, sintezatorius žodį kirčiuodavo kiek kitaip (*Igrahkas – Ygrekas*);

Beveik visos transkripcijos sudarytos jungiant visus tris metodus ir klausantis sintezatoriaus ištarimų. Taigi sudarant transkripcijas:

- garsas *y* keičiamas į *i* (pvz., žodis *Ygrekas* keičiamas į *Igrekas*)
- papildomų raidžių (*h*, *y*) įterpimas (pvz., *Zacharijus* keičiamas į *Zahthaarihyus*)
- raidžių dvigubėjimas taip pailginant garsą (pvz., *Zacharijus* keičiamas į *Zahthaarihyus*)
- papildomos *J* garso įterpimas (pvz., *Ygrekas* keičiamas į *yigrejhkas*)

Esant galimybei keisti ištarimo greitį, buvo galima lengviau įsiklausyti į tarimą, išgirsti kur koks garso trūksta, kuris panašesnis į reikiamą, kuris tinkamesnis. Visos šios žinios labai palengvino transkripcijų sudarymą ir koregavimą, norit sudaryti kuo panašesnę į lietuvišką skambesį. Sudarytos naujos transkripcijos yra įrašomos į gramatikos aprašą „*.grxml“, kiekviena nauja transkripcija yra grupuojama, ir rašoma į tą patį elementą *<item>*.

Pavyzdžiui:

```
<item>
  <one-of>
    <item>ehiymanhtahs</item>
    <item>eimantas</item>
    <item>eihmantas</item>
    <item>eigmantas</item>
    <item>eihmanhtas</item>
    <item>eihmanhtahss</item>
    <item>aeihmanthahs</item>
  </one-of>
  <tag>$_value = "eimantas"</tag>
</item>
```

Sudarytos transkripcijos bus naudojamos tyrimuose, norint išsiaiškinti ar naujų transkripcijų sudarymas ir pritaikymas naudojant kitakalbiui atpažintuvui pagerina vardų garsyno komandų atpažinimo tikslumą.

Komandos kurioms buvo pritaikytos naujos transkripcijos, kurių atpažinimo tikslumas procentais buvo prasčiausias su apmokytu profiliu „Iveta1“ ir su neapmokytu profiliu „Default_Neapmokytas1“: „Eimantas“, „Kju“, „Ulijona“, „Ygrekas“ ir „Zacharijus“. Visos gramatikos su naujomis transkripcijomis pateiktos 2 PRIEDE.

Norint nustatyti, su kokiais gramatikomis komandų atpažinimas yra tiksliausias, paskutinis vardų garsyno komandų atpažinimo tikslumo testavimas atliktas naudojant *PRON* atributu papildytą gramatiką UPS (UPS alfabeto fonemų lentelės ispanų kalbai 3 PRIEDAS). Gramatika buvo generuojama naudojantis *Microsoft Speech Server 2007*. Su *Visual Studio 2005* programine įranga buvo sugeneruotos daugybinės UPS transkripcijos kiekvienai iš vardų garsyno prasčiausiai atpažintai komandai – „Eimantas“, „Kju“, „Ulijona“, „Ygrekas“ ir „Zacharijus“. Vardų garsyno gramatika papildyta naujomis prasčiausiai atpažintamų komandų UPS transkripcijomis pateikiama 2 PRIEDE.

Pirmiausiai, kuriant UPS gramatiką sukuriamas naujas projektas („*New Project*“) „*Visual Studio 2005*“ aplinkoje. Buvo naudojama gramatiką „*.grxml“ su jau sukurtomis pradinėmis transkripcijomis. Nustatymuose pasirinkus „*Pronunciation*“, programa automatiškai sugeneruoja UPS transkripcijas.

Paimta pirminė žodžio transkripcija yra sugeneruojama į UPS transkripciją, kuri yra perkeliama į redaktoriaus langą ir gali būti papildomai redaguojama (žr. 2.3.1 pav.).

Tokiais principais parengtos transkripcijos yra saugomos „*Microsoft Speech Server 2007*“ sugeneruotose „XML“ formato atpažinimo gramatikose, dar žinomos kaip „*Grammar XML*“ (*GRXML*).

Atlikto vardų atpažinimo tikslumo tyrimo, su ispanų kalbos atpažintuvu naudojant skirtingus profilius ir gramatikas, rezultatai pateikiami 3.1 skyriuje.

2.3.2 Kitakalbio atpažintuvo panaudojimas skaičių balso komandų atpažinimui

Šis tyrimas atliekamas siekiant išsiaiškinti, kuris darbo režimas, sinchroninis ar asinchroninis, priklausantis „*Microsoft Speech Recognizer 8.0 for Windows*“ platformai, yra tikslesnis ir efektyvesnis skaičių komandų atpažinime.

Sinchroniniame darbo režime atliekamas automatizuotas kalbos atpažinimo procesas. Kalbos atpažinimui naudojama „*cWavoDalis.exe*“ programa, pridėjus papildomą gramatikos aprašą „*.grxml“. Testavimo programai reikalingi „*.wav“, „*.zgl“ ir „*.voc“ failai, o testavimo rezultatai pateikiami tekstiniu failu „*fresult.txt*“. Testuojamų skaičių – „*Skaiciai_20dikt_VOC*“ ir „*Skaiciai_New_10dikt_VOC*“ garsynų katalogai turėjo būti įkelti į specialiai paruoštą kietojo disko direktoriją, *C* → [*TEST*] → *CORPORA_11*. Programa atlieka skaičiavimus tik tuomet jeigu katalogas yra šioje direktorijoje. Gauti skaičių komandų (viso 6000, žiūrėti skyrių 2.1) atpažinimo tikslumo rezultatai surenkami rankiniu būdu, iš programos pateikto atpažinto komandų sąrašo išrenkamos klaidingai atpažintos komandos.

Asinchroninis darbo režimas nėra visiškai automatizuotas kaip sinchroninis darbo režimas, jis taikomas su programa „*Rec_UPS.exe*“, papildomai pridėjus gramatikos aprašą „*.grxml“. Programą sudaro kelios pagrindinės parinktys reikalingos programos valdymui – parinktis „*Start*“ atidaro langą, kuriame reikia pasirinkti, tam tikro diktorius susikurtą, „*.wav“ audio failą. Pasirinkus šį failą, programos lange atsiranda pasirinkto skaičiaus pavadinimo transkripcijos. Parinktis „*Valyti*“ išvalo teksto lauką, kuriame yra pateikiamos pasirinkto skaičiaus transkripcijos, o parinktis „*Baigti*“ uždaro „*Rec_UPS.exe*“ programą.

Pasirinkus norimą atpažinti skaičiaus pavadinimą programa atvaizduoja to pavadinimo teisingai atpažintas transkripcijas. Rankiniu būdu buvo suskaičiuotas teisingai atpažintų skaičių pavadinimų kiekis ir skaičiuojami tikslumo rezultatai.

Lietuviškų skaičių pavadinimų atpažinimo tikslumo tyrimą, naudojant ispanišką kalbos atpažintuvą, buvo nuspręsta atlikti naudojant visiškai kitokią metodiką nei vardų atpažinimo tikslumui tirti, siekiant kuo plačiau iširti *Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)* galimybes lietuviškoms balso komandoms atpažinti.

Skaičių atpažinimo tikslumo tyrimas su ispanų kalbos atpažintuvu buvo atliekami su vienu, apmokytu profiliu – „*Andrejus1*“, naudojant skaičių pavadinimų garsyną, kurį sudaro 6000 ištarimų, kuriam buvo taikoma tik žodinių transkripcijų gramatika. Siekiant išsiaiškinti kokią įtaką skaičių pavadinimų atpažinimo tikslumui daro ispanų kalbos atpažintuvo režimas, buvo atliktas skaičių pavadinimų tikslumo tyrimas su ispanų kalbos atpažintuvu sinchroniniame ir asinchroniniame režime.

Šiame skyriuje aprašytų ispaniško atpažintuvo darbo režimų įtakos skaičių komandų atpažinimo tikslumui tyrimo rezultatai pateikti 3.2 skyriuje.

2.4 Hidden Markov Model Toolkit įsisavinimas

Šiame skyriuje yra aprašoma darbo eiga, skaičių ir vardu pavadinimų atpažinimo tikslumo tyrimams atlikti, naudojant HTK (*Hidden Markov Model Toolkit*) programinį paketą. HTK programinio paketo pritaikymas šnekamosios kalbos atpažinimui ir pagrindiniai bruožai yra aptarti teorinėje darbo dalyje (žr. 1.3 skyrių), tačiau taip pat svarbu ir praktinis šio įrankio pritaikymas. Taigi šiame skyriuje aptariamas praktinis HTK programinio paketo pritaikymas, įsisavinimas ir reikalingų duomenų paruošimas šnekamosios kalbos atpažinimui.

HTK žodžių atpažinimas grįstas paslėptaisiais Markovo modeliais (PMM), taigi sistema pirmiausiai turi būti apmokyta, o tik paskui naudojama šnekos atpažinimui. PMM apmokymo metu visiems žodyno žodžiams pagal parinktas žodžių transkripcijas sudaromi tikimybiniai akustiniai modeliai ir apskaičiuojamas kiekvieno akustinio modelio tikėtinumo įvertis. Sistema iš žodyno pateikia kaip atpažintą tą žodį, kurį reprezentuoja modelis, kurio tikėtinumo įvertis yra didžiausias. Šnekamosios kalbos atpažinimo sistemos, kurioje kalbos atpažinimui taikomas PMM metodas, apmokymui rekomenduojama naudoti kitų diktorių garsyno įrašus nei atpažinimo kokybės testavimui.

HTK programiniam paketui reikalingų duomenų redagavimui, katalogavimui, perkėlimui, grupavimui ir kopijavimui buvo naudojamos papildomos programos - „*Total Commander*“ ir „*Far Manager 3*“, jas naudojant palengvėja duomenų tvarkymas ir padidėja darbo su duomenimis efektyvumas.

HTK programinio paketo programų aktyvavimui naudojami „*batch*“ (trumpinys „*.bat“) failai. Šių failų pagrindinė paskirtis yra automatizuoti tam tikrą procesą. Failų viduje parašoma tam tikra veiksmų seka, pvz. „*HCOPY -C CONFIG -S failai.scf*“, kuri vykdoma aktyvavus „*batch*“ failą. HTK paketo programos yra naudojamos nuosekliai, veiksmai atliekami vienas po kito.

2.4.1 HTK panaudojimas šnekos atpažinimui taikant žodžių modeliais grįstus PMM

Šnekamosios kalbos atpažinimui taikyti HTK programinių įrankių paketą galima modeliuojant balso komandų atpažinimą žodžių modelių metodu. Apie balso komandų modeliavimą taikant žodžių modelius plačiau aprašyta 1.2.5 skyriuje. Norit įvertinti šio metodo efektyvumą balso komandų atpažinimui modeliuoti, šio darbo metu balso komandų atpažinimo tikslumui gautam taikant žodžių modelius buvo atliktas tyrimas ir gauti rezultatai palyginti su balso komandų atpažinimo rezultatais gautais taikant fonemų modelių metodą.

Šiame skyriuje aprašomas HTK programinių įrankių naudojimas balso komandų atpažinimo modeliavimui taikant žodžių modelių metodą, kurio taikymą sudaro 7 nuoseklūs etapai. Pirmi šeši etapai apima žodžių akustinių modelių apmokymą, o septintasis etapas apima modelių testavimą. Trumpa visų etapų apžvalga pateikiama 2.4.1 lentelėje, o detalus žodžių modelių metodo taikymas pateiktas 5 PRIEDE.

2.4.1 lentelė. Žodžių modelių metodo taikymas

	Pavadinimas	Programinis įrankis	Programos aktyvavimo „*.bat“ failai	Reikalingi failai		
ETAPAS	1	<i>Gramatikos failo kūrimas</i>	_____	_____	<i>gram.dict</i>	MOKYMAS
	2	<i>Žodžių modelių sudarymui reikalingų failų paruošimas</i>	Hparse.exe	HParse.exe gram.dict wordnet.txt	<i>gram.dict, wordnet.txt</i>	
	3	<i>Požymių kūrimas</i>	Hcopy.exe	HCopy -C CONFIG -S failai.scp	<i>CONFIG.txt, failai.scp</i>	
	4	<i>Modelių parengimas</i>	_____	_____	<i>trans_nulis.scp, hmm_nulis, ...</i>	
	5	<i>Modelių apmokymas</i>	HRest.exe	HRest.exe -T 1 -S trans_nulis.scp hmm_nulis>> kk1 ... HRest.exe -T 1 -S trans_devyni.scp hmm_devyni>> kk10	<i>trans_nulis.scp, hmm_nulis, ..., trans_devyni.scp, hmm_devyni</i>	
ETAPAS	6	<i>Duomenų paruošimas testavimui</i>	Hcopy.exe	HCopy -C CONFIG -S test.scp	<i>CONFIG.txt, test.scp</i>	TESTAVIMAS
	7	<i>Modelių testavimas</i>	Hvite.exe	HVite -T 1 -S test_mfc.scp -i results -w wordnet.txt dict hmmlist	<i>test_mfc.scp, wordnet.txt, dict.txt, hmmlist</i>	

Naudojant HTK programinį paketą buvo atliktas tyrimas balso komandų atpažinimo tikslumui įvertinti naudojant paslėptųjų Markovo modelių metodą tiriamųjų komandų žodžių modeliams sukurti. Vardų ir skaičių garsynų atpažinimo tikslumo tyrimų rezultatai taikant žodžiais grįstus PMM aprašomi skyriuose 3.4 ir 3.5.

2.4.1.1 Požymių tipų įtakos atpažinimo tikslumui tyrimo metodika

Požymių tipų įtakos atskirai tariamų (izoliuotų) žodžių atpažinimo tikslumui tyrimas buvo atliekamas su daugiadiktoriniu lietuviškų skaičių garsynu, kurį sudaro nuo 0 iki 9 skaičių pavadinimai ištarti po 20 kartų trisdešimties diktorių balsais. Tiriamasis skaičių garsynas sudarytas iš šešių tūkstančių (6000) skirtingų balso įrašų. Plačiau garsynas aprašomas 2.1 skyriuje.

Skaičių garsyno balso komandos buvo modeliuojamos naudojant HTK programinių įrankių paketą, taikant žodžių modelių metodą aprašytą šiame skyriuje.

HTK programiniame pakete galima taikyti įvairius kalbos signalą koduojančių požymių tipus – WAVEFORM, LPC, LPCEPSTRA, LPREFC, LPDELCEP, IREFC, MFCC, FBANK, MELSPEC, USETR, DISCRETE, PLP [21].

Pirmajame, požymių įtakos šnekos atpažinimo tikslumui, tyrimo etape, atlikus literatūros analizę, tyrimui buvo pasirinkti trys plačiausiai naudojami kalbos signalų požymių tipai – LPC, LPCEPSTRA ir populiariausias MFCC. Pirmiausia buvo siekiama išsiaiškinti kuris požymių tipas nenaudojant papildomų parametrų yra efektyviausias skaičių komandų atpažinimui.

HTK programiniame pakete naudojamas „*config*“ failas, kuriame yra nustatomi keičiami ir nekeičiami parametrai kalbos signalui užkoduoti, tame tarpe yra ir požymių vektoriaus tipo parinkimas „*config*“ failo komandinėje eilutėje - „TARGETKIND =“, kur ir yra įrašomas pasirinktas požymio tipas. Taip pat kiekvienos komandos modelyje - „*hmm_komandos pav.*“ (naudojami komandų modeliai buvo pirminiai be papildomų būsenų ir Gauso mišinių) esančiose eilutėse nustatomas požymio tipas ir jam priskirti parametrai:

```
<STEAMINFO> 1 požymio kepstro koeficientas  
<VECSIZE> požymio kepstro koeficientas<NULLD><Požymio tipas>  
<MEAN> požymio kepstro koeficientas  
<VARIANCE> požymio kepstro koeficientas
```

Pavyzdžiui požymio tipui MFCC be jokių papildomų kepstro koeficientų (MFCC be papildomų koeficientų turi 12 kepstro koeficientų) šios eilutės kiekvienos komandos modeliuose atrodys taip:

```
<STEAMINFO> 1 12  
<VECSIZE> 12<NULLD><MFCC>  
<MEAN> 12  
<VARIANCE> 12
```

Pirmojo tyrimo etapo rezultatai, ištyrus MFCC, LPC ir LPCEPSTRA požymių tipų įtakos tyrimas aprašomas 3.3 skyriuje.

Antrame tyrimo etape, nustatius, efektyviausią požymių tipą komandų atpažinimui, toliau buvo didinamas keprstinis skaičius taikant koeficientų reikšmes ir tiriama skaičių atpažinimo tikslumo priklausomybė.

Su požymių tipu yra naudojami papildomi koeficientai: „*E*“ – kalbos signalo energijos, „*N*“ – absoliutinis energijos slopinimo koeficientas, „*D*“ – delta koeficientai, „*A*“ – pagreičio koeficientas, „*C*“ – suspaudimas, „*Z*“ – nulinis statinis koeficientas, „*K*“ – ciklinio perteklinio patikrinimo (*CRC*) kontrolinė suma, „*O*“ – nulines eiles keprstro koeficientas [21]. Kiekvienas koeficientas padidina keprstinio skaičiaus reikšmę.

Remiantis projekto „*Hibridinė atpažinimo technologija balso sąsajai*“ tyrimų rezultatais, kurie buvo atliekami naudojant MFCC požymio tipą ir papildomus „*D*“, „*A*“, „*E*“ koeficientus, skaičių komandų atpažinimo tikslumo priklausomybės tyrimui naudojant koeficientus, buvo pasirinkti būtent šie – projekto metu naudoti koeficientai. Kiekvienas iš jų MFCC požymio tipo turimą 12 keprstro koeficiento reikšmę atitinkamai padidina.

Požymio tipas MFCC buvo papildomas koeficientais palaipsniui – *MFCC_D* (turi 24 keprstro koeficiento reikšmę) >> *MFCC_D_A* (36) >> *MFCC_D_A_E* (39), kuri sudaro 12 MFCC tipo keprstro koeficientų, 12 iš šių koeficientų gautų delta koeficientų, 12 iš delta koeficientų gautų delta-delta koeficientų, energijos reikšmė (1), energijos delta ir delta-delta (2). Taigi iš viso galutinį požymių vektorių sudaro 39 koeficientai.

Šiame skyriuje aprašytu požymių tipų įtakos atskirai tariamų (izoliuotų) žodžių atpažinimo tikslumui tyrimo rezultatai pateikti 3.3 skyriuje.

2.4.1.2 Vardų atpažinimo tyrimo, naudojant žodžiais gristus modelius, metodika

Pagal šiame skyriuje aprašyto HTK programinių įrankių paketo taikymo balso komandų atpažinimui tirti, naudojant žodžiais gristus modelius, metodiką buvo sumodeliuoti vardų garsyno komandų modeliai, kurių atpažinimo tikslumo rezultatai, naudojant skirtingą modelių būsenų ir Gauso mišinių skaičių, aprašomi 3.4 skyriuje.

Norint gauti kuo tikslesnius vardų komandų modelius, reikia iširti komandų modelių atpažinimo tikslumą su įvairiomis parametru vertėmis, redaguojamomis HTK programiniame pakete.

Visi vardų atpažinimo tikslumo tyrimai atlikti naudojant 8 diktorių (moterų ir vyrų) vardų garsyno įrašus apmokymui, o 2 diktorių (moters ir vyro) įrašus vardų komandų modelių atpažinimo testavimui.

Kaip jau minėta ankščiau kiekviena balso komanda buvo modeliuojama viena ištisine Markovo grandine sukuriant komandoms atitinkamus PMM, kuriuose pagrindiniai redaguojami

parametrai, kurie gali įtakoti balso komandų atpažinimo tikslumą yra būsenų ir Gauso mišinių skaičius grandinėje.

- ***Papildomų būsenų įterpimas į žodžių modelius***

Atpažinimo tikslumo tyrimui, su skirtingu būsenų skaičiumi, būsenų skaičius buvo parenkamas apytiksliai lygus balso komandą sudarančių fonetinių elementų skaičiui intervale imtinai nuo 3 iki 10. Vardo komandos „*Kju*“ modeliui buvo naudotos 3 būsenos, komandų „*Boleslovas*“ ir „*Zacharijus*“ modeliams buvo naudotos 10 būsenų ir kitų balso komandų modeliams atitinkamai parenkamas būsenų skaičius. Toliau būsenų skaičius buvo parenkamas prie pirminio būsenų skaičiaus palaipsniui pridėdant 2, 4 ir 6 papildomas būsenas.

- ***Papildomų Gauso mišinių įterpimas į žodžių modelius***

Toliau tęsiant vardų atpažinimo tikslumo su skirtingu būsenų ir Gauso mišinių skaičiumi tyrimus testuojant tą patį vardų garsyną, buvo įterpiami Gauso mišiniai, ir tiriamas komandų atpažinimo tikslumas naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi - 2, 3, 6, 10 ir 16 Gauso mišinių, ir naudojant modelius su keturiomis papildomomis būsenomis ir 2, 3, 6, 10 ir 16 Gauso mišinių.

Atlikto vardų komandų atpažinimo tikslumo tyrimo, naudojant žodžių modelių metodą, rezultatai įvertinami 3.4 skyriuje.

Komandų modelių pavyzdžiai su skirtingu būsenų ir Gauso mišinių skaičiumi pateikti 9 PRIEDE.

2.4.1.3 Skaičių atpažinimo tyrimo, naudojant žodžiais gristus modelius, metodika

Pagal šiame skyriuje aprašyto HTK programinių įrankių paketo taikymo balso komandų atpažinimui tirti, naudojant žodžiais gristus modelius, metodiką buvo sumodeliuoti skaičių garsyno komandų paslėptieji Markovo modeliai, kurių atpažinimo tikslumo rezultatai, naudojant skirtingą modelių būsenų ir Gauso mišinių skaičių, aprašomi 3.5 skyriuje.

Skaičių komandų modelių atpažinimo tikslumo su įvairiomis parametru vertėmis, redaguojamomis HTK programiniame pakete tyrimas atliekamas lygiai tokia pačia tvarka kaip ir vardų komandų atpažinimo tikslumo tyrimas naudojant žodžiais gristus modelius aprašytas 2.4.1.2 skyriuje.

Visi skaičių atpažinimo tikslumo tyrimai atlikti naudojant 24 diktorių (moterų ir vyrų) skaičių garsyno įrašus apmokymui, o 6 diktorių (moterų ir vyrų) įrašus skaičių komandų modelių atpažinimo testavimui.

- ***Papildomų būsenų įterpimas į žodžių modelius***

Atpažinimo tikslumo tyrimui, su skirtingu būsenų skaičiumi, būsenų skaičius buvo parenkamas apytiksliai lygus balso komandą sudarančių fonetinių elementų skaičiui intervale imtinai nuo 2 iki 7 pridėdant 2, 4, 10 ir 16 papildomų būsenų.

- ***Papildomų Gauso mišinių įterpimas į žodžių modelius***

Toliau tęsiant, skaičių atpažinimo tikslumo su skirtingu būsenų ir Gauso mišinių skaičiumi, tyrimus testuojant tą patį vardų garsyną, buvo įterpiami Gauso mišiniai, ir tiriamas komandų atpažinimo tikslumas naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi - 2, 3, 6 ir 10 Gauso mišinių, ir naudojant modelius su keturiomis papildomomis būsenomis ir 2, 3, 6 ir 10 Gauso mišinių.

Atlikto skaičių komandų atpažinimo tikslumo tyrimo, naudojant žodžių modelių metodą, rezultatai įvertinami 3.5 skyriuje.

Komandų modelių pavyzdžiai su skirtingais būsenų ir Gauso mišinių skaičiumi pateikti 9 PRIEDE.

2.4.2 HTK panaudojimas šnekos atpažinimui taikant fonemų modeliais grįstus PMM

Šnekamosios kalbos atpažinimui taikyti HTK programinių įrankių paketą galima ir modeliuojant balso komandų atpažinimą fonemų modelių metodu. Apie foneminį balso komandų modeliavimą plačiau aprašyta 1.2.6 skyriuje. Norit įvertinti šio metodo efektyvumą balso komandų atpažinimui modeliuoti, šio tyrimo metu, balso komandų atpažinimo tikslumo rezultatai, gauti taikant fonemomis grįstus modelius, buvo palyginti su balso komandų atpažinimo rezultatais, gautais taikant žodžiais grįstus modelius.

Šis balso komandų modeliavimo metodas buvo taikomas tik skaičių garsynui atsižvelgiant į tai, kad balsu išstartų kodų sistemai atpažinti reikės daugiausia skaičius (visuose koduose daugiausia vyrauja skaičiai).

Taikant HTK programinių įrankių paketą balso komandų modeliavimui fonemų metodu reikia specialiai paruošti duomenis ir atitinkamus failus reikalingus modelių apmokymui ir atpažinimui. Balso komandų modeliavimui ir atpažinimui taikant fonemų modelius reikia nuosekliai atlikti 8 nuoseklus duomenų ir failų paruošimo etapus, pirmieji 7 etapai reikalingi fonemų PMM modelių paruošimui ir apmokymui, o aštuntasis etapas modelių testavimui. Trumpa visų etapų apžvalga pateikiama 2.4.2 lentelėje, o detalus žodžių modelių metodo taikymas pateiktas 5 PRIEDE.

2.4.2 lentelė. Fonemų modelių metodo taikymas

		ETAPAS			MOKYMAS
		Pavadinimas	Programinis įrankis	Programos aktyvavimo „*.bat“ failai	
1	Gramatikos failo kūrimas	Hparse.exe	HParse gram wdnet	gram.dict	
2	Fonemų modelių sudarymui reikalingų failų paruošimas			dict, words.mlf, monophones0 monophones1	
3	Požymių kūrimas	HLEd.exe	HLEd -l '*' -d dict -i phones0.mlf mkphones0.led words.mlf; HLEd -l '*' -d dict -i phones1.mlf mkphones1.led words.mlf	words.mlf, mkphones0.led, mkphones1.led	
4	Duomenų kodavimas	HCopy.exe	HCopy -A -D -T 1 -C CONFIG -S codetr.scp	CONFIG, codetr.scp	
5	Modelių parengimas	HCompv.exe, HERest.exe	HCompV -A -D -T 1 -C config6 -f 0.01 -m -S train.scp -M hmm0 proto; HERest -A -D -T 1 -C config6 -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones0; ...	proto, config6, train.scp, phones0.mlf, macros, hmmdefs, monophones0	
6	Tylos modelių taisymas	HHEd.exe, HERest.exe	HHEd -A -D -T 1 -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed monophones1; HERest -A -D -T 1 -C config6 -I phones1.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm5/macros -H hmm5/hmmdefs -M hmm6 monophones1 ...	macros, sil.hed, hmmdefs, monophones1, config6,train.scp, phones1.mlf, monophones1	
7	Modelių apmokymas	HVite.exe, HERest.exe	HVite -A -D -T 1 -l '*' -o SWT -b SENT-END -C config6 -a -H hmm7/macros -H hmm7/hmmdefs -i aligned.mlf -m -t 250.0 150.0 1000.0 -y lab -a -I words.mlf -S train.scp dict monophones1>>HVite_log; HERest -A -D -T 1 -C config6 -I aligned.mlf -t 250.0 150.0 3000.0 -S train.scp -H hmm7/macros -H hmm7/hmmdefs -M hmm8 monophones1	config6, macros, hmmdefs, words.mlf, train.scp, dict, monophones1, aligned.mlf	
8	Modelių testavimas	HCopy.exe, HVite.exe HResults.exe	HCopy -A -D -T 1 -C CONFIG -S codetr.scp; HVite -H hmm9/macros -H hmm9/hmmdefs -S test.scp -l '*' -i recout.mlf -w wdnet -p 0.0 -s 5.0 dict monophones1; HResults -p -I testref.mlf dict recout.mlf>>rez	CONFIG, codetr.scp, macros, hmmdefs, test.scp, recout.mlf, wdnet, dict, monophones1, testref.mlf	
TESTAVIMAS					

Naudojant HTK programinį paketą buvo atliktas tyrimas balso komandų atpažinimo tikslumui įvertinti naudojant paslėptųjų Markovo grandinių metodą tiriamųjų komandų foneminiams modeliams sukurti.

Skaičių garsyno atpažinimo tikslumo tyrimų rezultatai aprašomi 3.6 skyriuje.

2.4.2.1 Skaičių atpažinimo tyrimo, naudojant fonemomis gristus modelius, metodika

Dar vienas plačiai taikomas metodas modeliuoti šnekamąją kalbą - fonemomis gristų modelių metodas – kurio pagrindas kaip ir žodžiais gristų modelių metodo yra paslėptosios Markovo grandinės. Tik modeliuojant balso komandas fonemomis gristu metodu kiekviena skaičių garsyno balso komanda buvo suskaidoma į smulkesnius foneminius segmentus, kurie paskui išskiriami ir modeliuojami paslėpta Markovo grandine, ne taip kaip žodžiais gristų modelių metode, kai kiekviena komanda modeliuojama atskiru PMM. Norit iširti šio metodo efektyvumą skaičių komandų atpažinimo tikslumui, buvo atliekamas skaičių atpažinimo tikslumo tyrimas, vis didinant foneminių segmentų skaičių modeliuose.

Skaičių atpažinimo tikslumas buvo tiriamas pagal šiame skyriuje aprašytą metodiką. Atpažinimo tikslumo tyrimui pagal fonemomis gristus modelius, buvo pasirinkta tas pats 30 diktorių garsynas, kaip ir atpažinimo tikslumo tyrimams naudojant žodžiais gristus modelius – 24 diktorių garso įrašai naudojami apmokymui, o 6 naudojami atpažinimo tikslumo testavimui. Buvo tiriamas atpažinimo tikslumas su labiausiai balso komandų raides atitinkančiomis pasirinktomis fonemomis, kurių skaičius buvo didinamas įvairiais kiekiais nuo 17 (minimalus visas skaičių komandas atitinkantis fonemų skaičius) iki 33, norit nustatyti fonemų derinį su kuriuo atpažinimo tikslumo rezultatai bus patys geriausi. Tyrimai buvo atliekami naudojant derinius iš 17, 26, 27, 28, 30, 33 fonemų. Tyrimų rezultatai aprašyti 3.6 skyriuje. Naudojamų fonemų derinių sąrašas patiktas 6 PRIEDE.

2.5 Šnekos atpažintuvų sujungimo galimybių tyrimo metodika

Projekto „*Hibridinė atpažinimo technologija balso sąsajai*“ trukmė buvo tik pusantrų metų, todėl dviejų šiame projekte naudotų atpažintuvų sujungimo galimybių tyrimas liko pilnai neišnagrinėtas. Taip pat projekto metu buvo atlikta detali atpažintuvų ATP_SP ir ATP_LT papildomumo analizė. Kiekvienam iš atpažintuvų buvo pateikti atpažinti „*Medic*“ garsyno 175440 garso įrašai. Gautieji rezultatai buvo susisteminti ir apskaičiuota „*akla*“ sprendimo taisyklė, teigianti - „*jei atpažintuvų sprendimai skiriasi, pasirink geresniojo atpažintuvo ATP_LT sprendimą*“ veikia 96,12% tikslumu. Šnekos atpažintuvų sujungimo galimybių tyrimo tikslas gauti hibridinę sprendimo taisyklę, kuri viršytų „*aklos*“ taisyklės tikslumą. Detalesnė informacija apie patį projektą ir atliktus tyrimus pateikta literatūros šaltinyje [24].

Šnekos atpažintuvų sujungimo galimybių tyrimo metu bus naudojamosi projekto metu gautais rezultatais ir surinktuoju „*Medic*“ garsynu (žr. 2.1 skyrių).

Atlikus išsamią literatūros analizę, dviejų atpažintuvų sujungimo tyrimui pasirinkta duomenų analizės sistema WEKA, jis taip pat yra vienas iš plačiausiai naudojamų Lietuvoje.

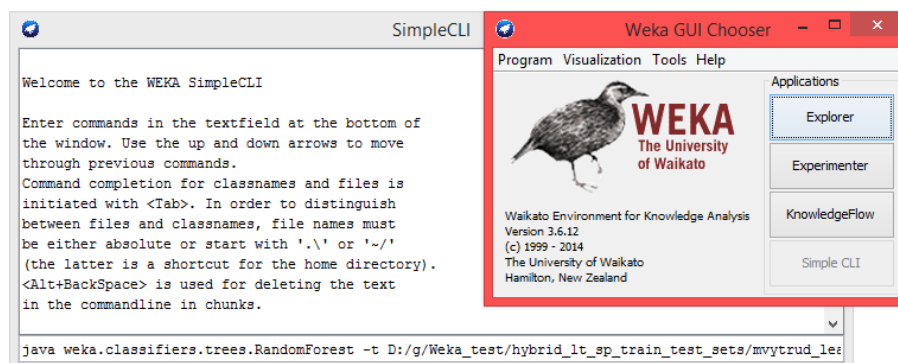
WEKA pakete įdiegta kelios dešimtys klasifikatorių, todėl iš jų reikia pasirinkti efektyviausią klasifikatorių. Taip pat, kaip ir duomenų analizės sistemos pasirinkimą lėmė literatūros apžvalga, taip ir klasifikatorių atrankos tyrimui pasirinkimą lėmė literatūros apžvalga. Efektyviausio klasifikatoriaus nustatymo tyrimui buvo pasirinkta 10 kandidatų: kNN, RIPPER, NB, RF, C4.5, ZeroR, SVM, AdaBoost, MP ir MLR.

Hibridinė sprendimo priėmimo taisyklė buvo mokoma ir testuojama 12-kartų kryžminio patikrinimo būdu. Ji buvo mokoma, naudojant 11 kalbėtojų duomenis, o dvyliktojo kalbėtojo duomenys naudojami patikrinti išmoktos taisyklės tikslumą. Šio tyrimo metu gautieji rezultatai vertinami tada, kai atpažintuvas atpažįsta vieno iš žinomų 12 diktorių šneką. Eksperimento rezultatai pateikti 3.8 skyriuje. Gautųjų rezultatų vertinimui taip pat yra svarbus ir laikas sugaištas apmokytos taisyklės testavimui. Testavimo laikas yra vienas iš kriterijų galutinei efektyviausio klasifikatoriaus atrankai.

Hibridinės sprendimo priėmimo taisyklės efektyvumo, 12-kartų kryžminio patikrinimo būdu, tyrimas buvo atliekamas naudojant WEKA duomenų analizės sistemos paketo įrankį „*SimpleCLI*“. Šio įrankio komandinėje eilutėje yra nurodoma duomenų klasifikatoriaus tipas, klasifikatorius, apmokymo duomenų direktorija ir testavimo duomenų direktorija:

```
java weka.classifiers.trees.RandomForest  
-t D:/g/Weka_test/hybrid_lt_sp_train_test_sets/fginpas_learn.arff  
-T D:/g/Weka_test/hybrid lt sp train test sets/fginpas test.arff
```

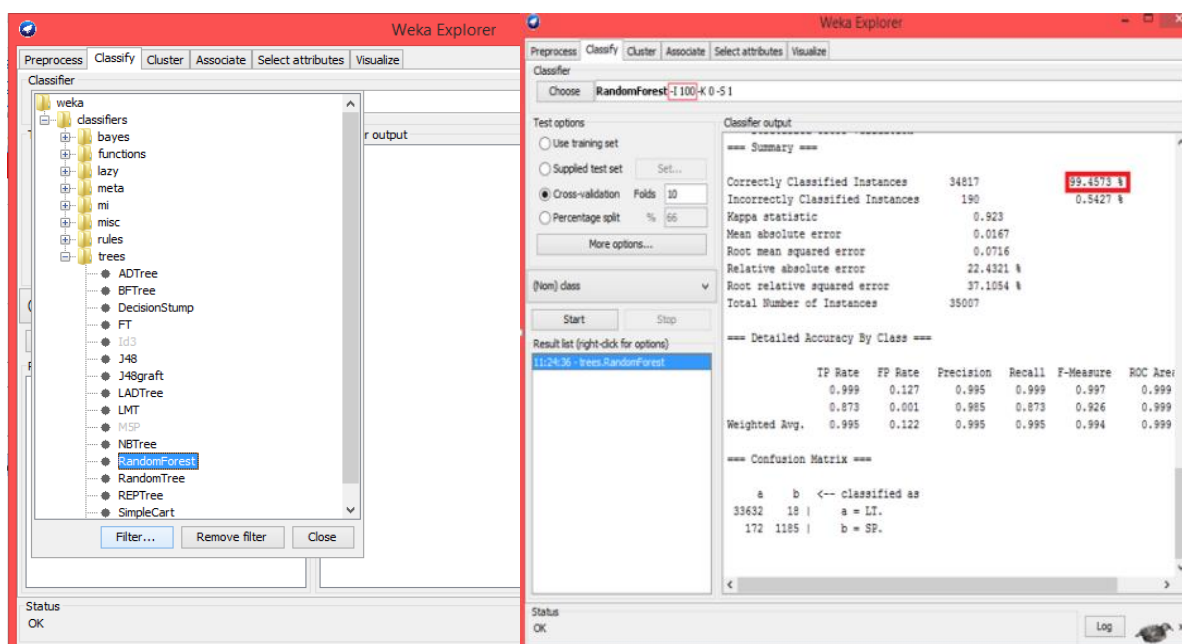
Apmokymui ir testavimui naudojami „**_learn.arff*“ ir „**_test.arff*“ failai, kuriuose yra surašyti testuojamų duomenų požymiai, failai buvo kurti projekto „*Hibridinė atpažinimo technologija balso sąsajai*“ metu. Trumpas tyrimo fragmentas pateiktas 2.5.1 paveiksle. Tyrimas atliekamas su visais tyrimui pasirinktais klasifikatoriais ir su visų 12 diktorių paruoštais duomenimis kas kartą vis nurodant skirtingo diktoriaus duomenų failus ir klasifikatorių.



2.5.1 pav. WEKA sistemos duomenų klasifikavimo tyrimas (12 – k. kryžminis)

Taip pat hibridinė sprendimo priėmimo taisyklė buvo mokoma ir testuojama 10-kartų kryžminio patikrinimo būdu. Mokoma naudojant 90% objektų, o testuojama naudojant 10% objektų. Automatiškai atliekama 10 bandymų, vis keičiant testavimo aibės objektus, rezultatai apdorojami ir pateikiami. Šio tyrimo gautieji rezultatai vertinami tada, kai atpažintuvus atpažįsta vieno iš nežinomų kalbėtojų šneką. Eksperimento rezultatai pateikti 3.8 skyriuje.

Hibridinės sprendimo priėmimo taisyklės efektyvumo, 10-kartų kryžminio patikrinimo būdu, tyrimas buvo atliekamas naudojant WEKA duomenų analizės sistemos paketo įrankį „EXPLORER“. Įrankio valdymo lange atidaromas analizuojamų duomenų failas „*.arff“, „classify“ parinktyje išsirenkamas norimas klasifikatorius, „cross-validation“ parinktyje įvedame 10 (10-k. kryžminis patikrinimas) ir pradame duomenų klasifikavimą – „Start“. Programos teksto lauke atsiranda testavimo rezultatai. Tyrimas atliekamas su visais tyrimui pasirinktais klasifikatoriais ir naudojant vieną, talpinantį visų 12 diktorių paruoštus duomenis, failą, kas kartą vis nurodant kitą klasifikatorių. Trumpas tyrimo fragmentas pateiktas 2.5.2 paveiksle.



2.5.2 pav. WEKA sistemos duomenų klasifikavimo tyrimas (10 – k. kryžminis)

Atlikus tyrimą geriausi klasifikavimo rezultatai abiem atvejais buvo gauti naudojant RF klasifikatorių (naudotas numatytasis medžių skaičius 100). RF klasifikatorius naudoja 3 numatytuosius parametrus – I – medžių skaičių, K – svarstytinų požymių skaičius, S – atsitiktinių skaičių sekų generatoriaus numeris (angl. „seed“). Todėl buvo nuspręsta atlikti tyrimą keičiant RF klasifikatoriaus medžių skaičių (I), ir paieškoti, tikslumo prasme, efektyviausio RF klasifikatoriaus medžių skaičiaus, atsižvelgiant ir į tai, kad mažinant medžių skaičių mažėja testavimui sugaištamasis laikas. Medžių skaičiaus paieškai buvo atliktas 10-kartų kryžminio patikrinimo eksperimentas, mokymui naudojant 90% objektų, o testavimui naudojant likusius 10% objektų ir

keičiant RF klasifikatoriaus medžių skaičių nuo 1 iki 400. Kiti parametrai palikti WEKA paketo numatytieji, o jų poveikio eksperimentai, šnekos atpažintuvų sujungimo galimybei tirti, gali būti atliekami ateityje. Eksperimento rezultatai pateikti 3.8 Skyriuje.

RF klasifikatoriaus tinkamiausios medžių paieškos tyrimas atliktas lygiai tokia pačia metodika kaip ir 10-kartų kryžminis patikrinimas, tik klasifikatoriai jau buvo nebekaitaliojami, paliekamas tik vienas RF, o keičiamas didėjančia tvarka jo medžių skaičius I. Trumpas tyrimo fragmentas pateiktas 2.5.2 paveiksle.

Atlikus RF klasifikatoriaus tinkamiausios medžių paieškos tyrimą ir suradus tinkamiausią medžių skaičių, yra aktualu patikrinti kokią įtaką hibridinio atpažintuvo sprendimo priėmimo tikslumui daro mokymo imties objektams aprašyti požymiai (žr. lent. nr. 1.5.4, 7 PRIEDAS), kurie plačiau aprašyti 1.5 skyriuje.

Požymių įtaka buvo tirama, iš duomenų failų pašalinant tam tikrus požymius ir naudojant duomenų analizės sistemą WEKA, kuri pateikia automatiškai apskaičiuotą tikslumą. Tyrimas buvo atliktas 10-kartų kryžminio patikrinimo eksperimentu, mokymui naudojant 90% objektų, o testavimui naudojant likusius 10%, parenkant, remiantis prieš tai darytais tyrimais, aktualius klasifikatorių RF ir jo medžių skaičių I – 9. Požymių įtakos tyrimo metu buvo keičiami duomenys su skirtingais požymiais ir tikrinamas gautas sprendimo priėmimo tikslumas procentais. Atlikto tyrimo rezultatai pateikiami 3.8 skyriuje.

3. ATLIKTŲ TYRIMŲ REZULTATŲ DALIS

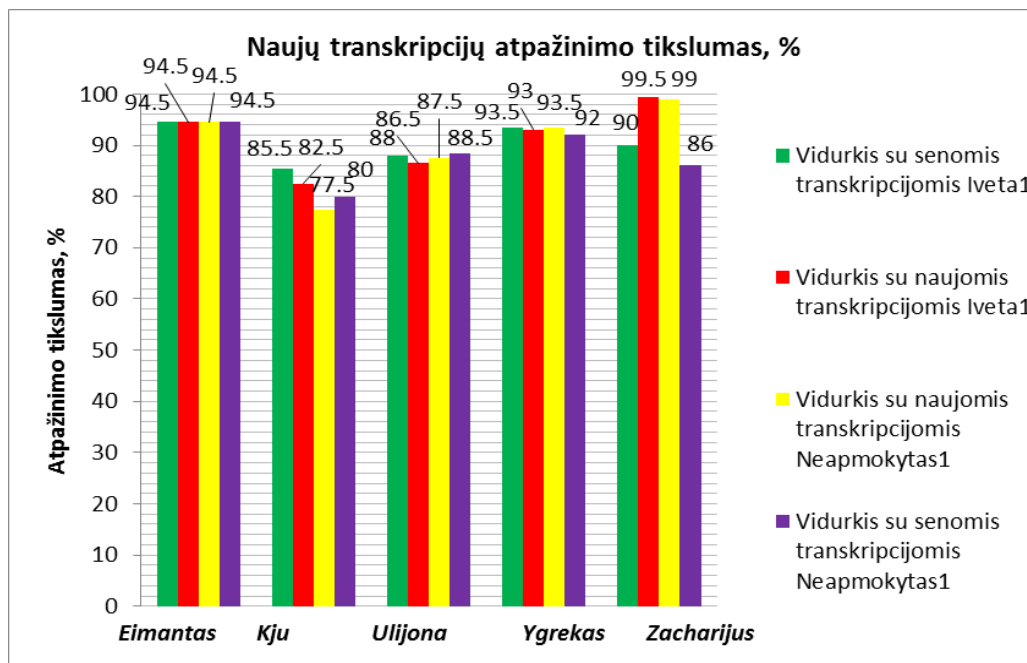
3.1 Vardų atpažinimo tyrimas su ispanišku atpažintuvu

Vardų atpažinimo tikslumo tyrimui naudojamas ispanų kalbos atpažintuvas. Tyrimo metodika aprašyta 2.3 skyriuje, o tyrimo metu gautieji, atpažinimo tikslumo su ispaniško atpažintuvo skirtingais profiliais, rezultatai pateikti 3.1.1 lentelėje 10 PRIEDE.

Iš 3.1.1 lentelėje pateiktų vardų balso komandų atpažinimo tikslumo rezultatų buvo apskaičiuotas $96,9 \pm 1,46\%$ atpažinimo tikslumas su apmokytu profiliu – „*Iveta1*“ ir $96,4 \pm 1,90\%$ tikslumas su neapmokytu profiliu – „*Default_Neapmokyta1*“. Atpažinimo tikslumas gautas su apmokytu profiliu yra 0,5% aukštesnis nei atpažinimas gautas su neapmokytu profiliu. Tiriant profilių įtaką atpažinimo kokybei taip pat buvo išrinktos vardų - komandos, kurios buvo atpažintos prasčiausiai, todėl tolimesnių tyrimų metu taikant naujų transkripcijų parinkimą bus gerinamas jų atpažinimo tikslumas. Apmokytas profilis „*Iveta1*“ prasčiausiai atpažįsta vardus - *Kju* (85,5%) ir *Ulijona* (88%). Neapmokytas profilis „*Default_Neapmokyta1*“ prasčiausiai atpažįsta, kaip ir apmokyto profilio atveju, - *Kju* (80%) ir *Ulijona* (88,5%) taip pat ir *Zacharijus* (86%).

Komandoms pritaikytos naujos transkripcijos, kurių atpažinimo tikslumas procentais buvo prasčiausias su apmokytu profiliu „*Iveta1*“ ir su neapmokytu profiliu „*Default_Neapmokyta1*“.

Vardų balso komandų atpažinimo tikslumo tyrimas su nauja - patobulinta gramatika atliekamas su ispanišku kalbos atpažintuvu naudojant tuos pačius – apmokytą „Iveta1” ir neapmokytą „Default_Neapmokyta1“ profilius. Tyrimo rezultatai pavaizduoti 3.1.1 paveiksle pateiktoje stulpelinėje diagramoje.



3.1.1 pav. Komandų atpažinimo tikslumas po gramatikos papildymo

Iš pateiktų rezultatų matoma, kad papildžius gramatiką naujomis transkripcijomis naudojant apmokytą profilį „Iveta1“ viena iš testuotų komandų (Zacharijus) buvo atpažinta geriau nei naudojant pradinę gramatiką. Tikslumas pakilo nuo buvusiųjų 90% iki 99,5%. Likusios komandos geriau atpažintos su pradinėmis transkripcijomis: Kju (sen. 85,5%, nauj. 82,5%), Ulijona (sen. 88%, nauj. 86,5%), Ygrekas (sen. 93,5%, nauj. 93%), o komandos Eimantas atpažinimo tikslumas lyginant naują ir pradinę transkripcijas išliko toks pats – 94,5%.

Su neapmokytu profiliumi „Default_Neapmokyta1“ atliktas tyrimas pagerino komandų Ygrekas (nuo 92% iki 93,5%) ir Zacharijus (nuo 86% iki 99%) atpažinimo tikslumą. Eimanto (94,5%) rezultatai abiem atvejais išliko tokie patys, o likusios komandos geriau atpažintos pagal senas transkripcijas: Kju (sen. 80%, nauj. 77,5%) ir Ulijona (sen. 88,5%, nauj. 87,5%). Gauti tyrimo rezultatai pateikti 3.1.2 lentelėje 10 PRIEDE.

Iš 3.1.2 lentelėje pateiktų vardų balso komandų atpažinimo tikslumo rezultatų buvo apskaičiuotas $97,1 \pm 1,6\%$ atpažinimo tikslumas su apmokytu profiliumi – „Iveta1“ ir $96,8 \pm 1,88\%$ tikslumas su neapmokytu profiliumi – „Default_Neapmokyta1“. Atpažinimo tikslumas gautas su apmokytu profiliumi yra 0,3% aukštesnis nei atpažinimas gautas su neapmokytu profiliumi.

Palyginimui 3.1.3 lentelėje pateikiami lietuviškų vardų garsyno atpažinimo tikslumo rezultatai procentais su pradine gramatika ir su papildyta gramatika naudojant skirtingus profilius – „Iveta1” ir „Default_Neapmokyta1”.

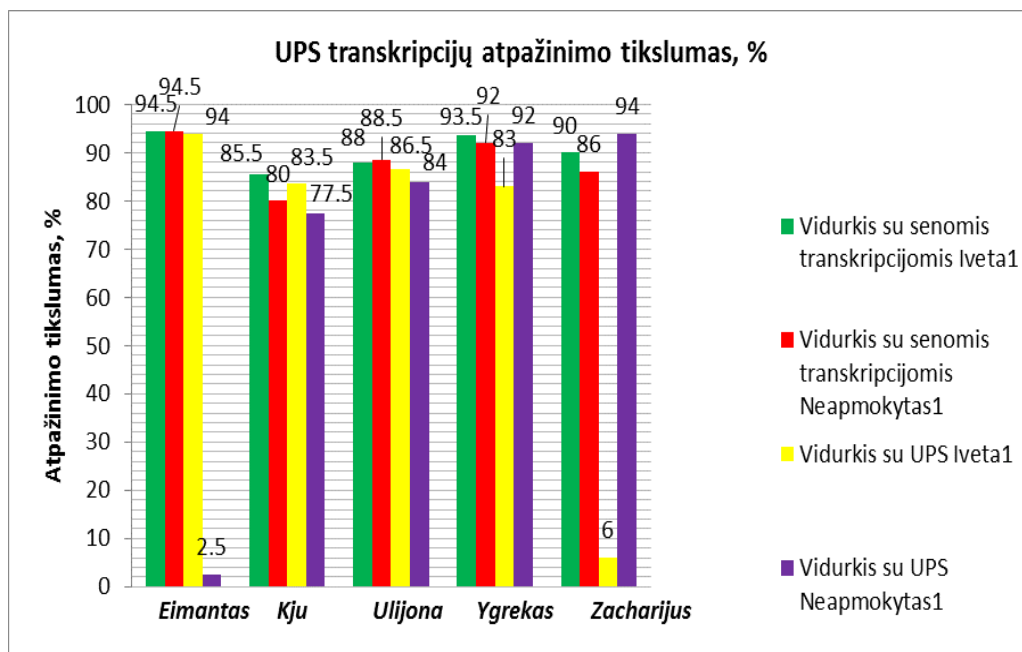
3.1.3 lentelė. Vardų garsyno atpažinimo tikslumas procentais su pradine ir su papildyta gramatikomis

Pradinė gramatika	Papildyta gramatika	Pradinė gramatika	Papildyta gramatika
„Iveta1”		„Default_Neapmokyta1”	
96,9±1,46%	97,1±1,6%	96,4±1,9%	96,8±1,88%

Iš gautų rezultatų matoma, jog po gramatikos papildymo naujomis transkripcijomis pagerėjo vardų garsyno atpažinimo tikslumas naudojant tiek apmokytą - “Iveta1” tiek neapmokytą - “Default_Neapmokyta1” profilius. Nepaisant to, kad profiliui “Iveta1” naujos transkripcijos buvo naudingos tik komandai Zacharijus, o - “Default_Neapmokyta1” komandos Zacharijus ir Ygrekas, to pakako, kad bendras lietuviškų vardų garsyno atpažinimo tikslumo vidurkis profiliui “Iveta1” pakiltų 0,2%, o profiliui - “Default_Neapmokyta1” 0,4%, lyginant pradinės ir papildytos gramatikos atpažinimo tikslumo rezultatus. Taigi gauti rezultatai leidžia daryti išvadą, kad naujų transkripcijų parinkimas gali būti vienas iš būdų, galinčių pagerinti atpažinimo rezultatus, nors ir minimaliai (bent šiuo atveju), todėl kitų kalbos atpažinimo tyrimų atveju šio būdo pritaikymas būtų tikslingas žingsnis.

Toliau siekiant dar labiau pagerinti atpažinimo tikslumą, naudojamos UPS (*Universal phone set*) transkripcijas.

Vardų garsyno atpažinimo tikslumo tyrimas su gramatika papildyta UPS transkripcijomis atliekamas su tais pačiais ispanų kalbos atpažintuvo profiliais – apmokytu „Iveta1” ir neapmokytu „Default_Neapmokyta1”. Tyrimo rezultatai pavaizduoti 3.1.2 paveiksle esančioje stulpelinėje diagramoje.



3.1.2 paveikslas. Komandų atpažinimo tikslumas po UPS transkripcijų pritaikymo

Iš pateiktos diagramos matyti, kad UPS transkripcijos pablogino 5 prasčiausiai atpažįstamų komandų atpažinimo tikslumą. Visos komandos su UPS transkripcijomis, naudojant apmokytą profilį „Iveta1“, buvo atpažintos prasčiau nei su pradine gramatika. Pritaikius UPS transkripcijas, ankstesniuose tyrimuose geriausiai iš prasčiausių atpažinta komanda *Zacharijus*, buvo atpažinta prasčiausiai (sen. 90%, UPS 6%). Likusiųjų komandų atpažinimo tikslumas su UPS transkripcijomis nuo atpažinimo tikslumo su pradinėmis transkripcijomis skyrėsi ne taip akivaizdžiai: *Eimantas* (sen. 94,5%, UPS 94%), *Kju* (sen. 85,5%, UPS 83,5%), *Ulijona* (sen. 88%, UPS 86,5%) ir *Ygrekas* (sen. 93,5%, UPS 83%).

Naudojant neapmokytą profilį „Default_Neapmokytas1“ visos komandos su UPS transkripcijomis taip pat kaip ir su profiliu „Iveta1“, buvo atpažintos prasčiau nei su pradine gramatika. „Default_Neapmokytas1“ profilis iš komandų su UPS transkripcijomis geriausiai atpažino komandą *Zacharijus* (sen. 86%, UPS 94%). Likusios komandos savo atpažinimo tikslumo rezultatais nedaug skyrėsi nuo pirminės gramatikos rezultatų: *Kju* (sen. 80%, UPS 77,5%), *Ulijona* (sen. 88,5%, UPS 84%). Pradinis komandos *Eimantas* atpažinimo tikslumo rezultatas radikaliai skyrėsi nuo atpažinimo tikslumo su UPS rezultatu – pradinė gramatika – 94,5%, o UPS – 2,5%. Vienintelės komandos *Ygrekas* pradinis ir UPS atpažinimo tikslumo rezultatai išliko tokie patys (sen. 92%, UPS 92%). Gauti atpažinimo tikslumo tyrimo rezultatai su gramatika papildyta UPS transkripcijomis pateikti 3.1.4 lentelėje 10 PRIEDE.

Iš 3.1.4 lentelėje pateiktų vardų balso komandų atpažinimo tikslumo rezultatų buvo apskaičiuotas $93,2 \pm 6,8\%$ atpažinimo tikslumas su apmokytu profiliu – „Iveta1“ ir $92,9 \pm 7,1\%$ tikslumas su neapmokytu profiliu – „Default_Neapmokytas1“. Atpažinimo tikslumas gautas su apmokytu profiliu yra 0,3% aukštesnis nei atpažinimas gautas su neapmokytu profiliu.

Norint išsiaiškinti kiek pavyko pagerinti vardų atpažinimo tikslumo rezultatus su pradine gramatika, naujomis transkripcijomis papildyta gramatika ir su UPS transkripcijų gramatika, palyginimui 3.1.5 lentelėje pateikiami vardų garsyno atpažinimo tikslumo rezultatai procentais su pradine gramatika, su papildyta gramatika ir su UPS transkripcijų gramatika, naudojant skirtingus profilius – „Iveta1“ ir „Default_Neapmokytas1“.

3.1.5 lentelė. Vardų garsyno atpažinimo tikslumas procentais su skirtingomis gramatikomis

Pradinė gramatika	Papildyta gramatika	UPS gramatika	Pradinė gramatika	Papildyta gramatika	UPS gramatika
„Iveta1“			„Default_Neapmokytas1“		
96,9±1,46%	97,1±1,6%	93,2±6,8%	96,4±1,9%	96,8±1,88%	92,9±7,1%

Iš rezultatų pateiktų 3.1.5 lentelėje matome, kad vardų garsyno atpažinimo tikslumas, tiek apmokyto „Iveta1“ (97,1±1,6%), tiek neapmokyto „Default_Neapmokytas1“ (96,8±1,88%) profilio atveju, naudojant pradinę papildytą penkių prasčiausiai atpažįstamų vardų balso komandų naujomis transkripcijomis gramatiką, gautas geriausias. Pradinės gramatikos papildymas naujomis

transkripcijomis pagerino atpažinimo tikslumą 0,2% apmokyto profilio – „*Iveta1*“ atveju ir 0,4% neapmokyto profilio „*Default_Neapmokyta1*“ atveju. Pradinės gramatikos papildymas UPS transkripcijomis nepagerino atpažinimo tikslumo rezultatų, o tik juos pablogino, nes su pradine gramatika atpažinimo tikslumo rezultatai naudojant abu profilius yra gauti geresni (apmok. - $96,9 \pm 1,46\%$ ir neapmok. - $96,4 \pm 1,9\%$) nei su gramatika papildyta UPS transkripcijomis (apmok. - $93,2 \pm 6,8\%$ ir neapmok. - $92,9 \pm 7,1\%$). Pradinės gramatikos papildytos UPS transkripcijomis atpažinimo tikslumas suprastėjo 3,7% apmokyto profilio – „*Iveta1*“ atveju ir 3,5% neapmokyto profilio „*Default_Neapmokyta1*“ atveju. Iš to galime spręsti, kad penkios prasčiausiai atpažintos komandos, su naujomis UPS transkripcijomis, įsimaišo dideliame kiekyje duomenų, o testavimo metu ispaniškas atpažintuvas pasiklysta tarp naujų transkripcijų, jas bandydamas pritaikyti ir kitų komandų atpažinimui, kurios turi tik vieną pradinę transkripciją. Norint pagerinti bendrą vardų garsyno atpažinimo tikslumą reikia rasti alternatyvių užrašymo būdų visoms garsyno komandoms, o ne tik penkioms prasčiausiai atpažintoms pirmojo tyrimo (naudojant pradinę gramatiką) metu. Taip pat nebuvo atrinktos laimėjusios vardų komandų transkripcijos - palikti visi sugalvoti transkripcijų variantai, o informacijos perteklius taip pat gali turėti neigiamos įtakos atpažinimo kokybei.

Atliekant atpažinimo tikslumo tyrimus su skirtingomis gramatikomis buvo naudojami skirtingi ispaniško atpažintuvo profiliai – apmokytas „*Iveta1*“ ir neapmokytas „*Default_Neapmokyta1*“. Iš galutinių rezultatų pateiktų 3.1.5 lentelėje, galima teigti, kad naudojant apmokytą profilį „*Iveta1*“ atpažinimo tikslumas buvo aukštesnis su pradine gramatika 0,5%, su pradine papildyta naujomis transkripcijomis ir pradine papildyta UPS transkripcijomis gramatikomis 0,3%, nei atpažinimo tikslumas naudojant neapmokytą profilį „*Default_Neapmokyta1*“.

Nors, kai kurių atpažinimo tikslumo, su papildytais gramatikomis, tyrimuose norimų pagerinti komandų atpažinimo tikslumas suprastėjo, galima teigti, kad sudarant ir pritaikant naujas transkripcijas naudojamam atpažintuvo tipui, galima pagerinti komandų atpažinimo tikslumą.

3.2 Skaičių atpažinimo tyrimas su ispanišku atpažintuvu

Atlikus skaičių pavadinimų atpažinimo tikslumo tyrimą su ispanų kalbos atpažintuvu sinchroniniame ir asinchroniniame režimuose, trisdešimties diktorių, kiekvieno skaičiaus pavadinimo (nuo nulio iki devynių) tyrimo metodika pateikta 2.3 skyriuje, o tyrimo rezultatai rezultatai pateikiami 3.2.1 lentelėje.

3.2.1 lentelė. Skaičių atpažinimo tikslumo rezultatai naudojant skirtingus ispaniško atpažintuvo darbo režimus

Skaičiaus pavadinimas	Atpažinimo tikslumas, %	
	Sinchroninis režimas, %	Asinchroninis režimas, %
Nulis	70,7	90,8
Vienas	96,7	96,7
Du	69,3	86
Trys	99	97,8
Keturi	85,7	78,2
Penki	99	98
Šeši	100	99,2
Septyni	99,2	100
Aštuoni	99,7	99,7
Devyni	72	90,7
Vidurkis:	89,1±8,32	93,7±4,47

Peržvelgus 3.2.1 lentelėje pateiktus rezultatus matome, kad ispaniškas atpažintuvas sinchroniniame režime prasčiausiai atpažįsta skaičių komandas – „Du“ (69,3%), „Nulis“ (70,7%) – šių komandų atpažinimo tikslumas nesiekia 71%, o geriausiai komandas – „Šeši“ (100%), „Aštuoni“ (99,7%) – jų atpažinimo tikslumas viršija 99%. Bendras skaičių komandų atpažinimo tikslumo vidurkis naudojant ispanų kalbos atpažintuvą sinchroniniame režime siekia 89,1±8,32%. Gerai atpažintais skaičių pavadinimais laikomi, tie, kurių atpažinimo tikslumas procentais viršijo bendrą skaičių pavadinimų atpažinimo tikslumo vidurkį. Ispaniškas atpažintuvas asinchroniniame režime prasčiausiai atpažįsta skaičių komandas – „Keturi“ (78,2%), „Du“ (86%) – šių komandų atpažinimo tikslumas nesiekia 90%, o geriausiai komandas – „Septyni“ (100%), „Aštuoni“ (99,7%) – jų atpažinimo tikslumas viršija 99%. Bendras skaičių komandų atpažinimo tikslumo vidurkis naudojant ispanų kalbos atpažintuvą asinchroniniame režime siekia 93,7±4,47%. Kaip ir skaičių atpažinimo tikslumo tyrimo su ispanišku atpažintuvu sinchroniniame režime, gerai atpažintais skaičių pavadinimais buvo laikomi, tie, kurių atpažinimo tikslumas procentais viršijo bendrą skaičių pavadinimų atpažinimo tikslumo vidurkį.

Palyginus abiem ispaniško atpažintuvo režimais gautų tyrimų rezultatus, pastebime, kad asinchroniniame režime atpažinimo rezultatai (93,7±4,47%) buvo 4,6% geresni, nei sinchroniniame režime (89,1±8,32%). Skaičiaus pavadinimo „nulis“ atpažinimo tikslumas asinchroniniame režime yra net 20,1% geresnis už sinchroniniame režime gautus skaičiaus „nulis“ rezultatus. Skaičiaus pavadinimo „devyni“ atpažinimo tikslumas asinchroniniame režime yra 18,7% geresnis už sinchroniniame režime gautus rezultatus. Tačiau asinchroniniame režime ne visi skaičių pavadinimai buvo atpažinti geriau. Skaičių pavadinimai „trys“, „keturi“, „penki“ ir „šeši“, sinchroniniame režime buvo atpažinti šiek tiek geriau, nei asinchroniniame režime. Skaičių pavadinimai „vienas“ ir „aštuoni“ abiejuose režimuose buvo atpažinti vienodai – skaičiaus „vienas“ atpažinimo tikslumas siekė 96,7%, o skaičiaus „aštuoni“ – 99,7%.

3.3 Kalbos signalo apdorojimo požymių tyrimas

Požymių tipų įtakos atskirai tariamų (izoliuotų) žodžių atpažinimo tikslumui tyrimas buvo atliekamas pagal 2.4 skyriuje pateiktą metodiką.

Pirmojo tyrimo etapo rezultatai, ištyrus MFCC, LPC ir LPCEPSTRA požymių tipų įtaką komandų atpažinimo tikslumui, pateikti 3.3.1 lentelėje.

3.3.1 lentelė. Skaičių atpažinimo tikslumas su skirtingais požymių tipais.

Požymio tipas	Atpažinimo tikslumas, %
MFCC	51,75
LPC	32,67
LPCEPSTRA	18,08

Kaip matome iš 3.3.1 lentelėje pateiktų rezultatų su MFCC požymių tipu skaičių komandų atpažinimo tikslumas yra aukščiausias (51,75%), o prasčiausias su LPCEPRSTA (18,08%).

Antrame tyrimo etape, nustačius, kad tiksliausias atpažinimas buvo pasiektas su MFCC požymių tipu, toliau buvo didinamas keptrinis skaičius taikant koeficientų reikšmes ir pagal 2.4.1.1 skyriuje pateikta metodiką tiriama skaičių atpažinimo tikslumo priklausomybė.

Tyrimo metu gautas skaičių atpažinimo tikslumas su skirtingomis MFCC keptrinio koeficiento skaičiaus reikšmėmis pateiktas 3.3.2 lentelėje.

3.3.2 lentelė. Skaičių atpažinimo tikslumas su skirtingomis požymių tipo MFCC keptrinio koeficiento skaitinėmis vertėmis

Požymio tipas	Atpažinimo tikslumas, %	Keptro koeficiento skaitinė vertė
MFCC_D	52,67	24
MFCC_D_A	47,92	36
MFCC_D_A_E	76,50	39

Iš skaičių komandų atpažinimo tikslumo rezultatų pastebime, kad tiksliausias atpažinimas buvo pasiektas naudojant MFCC_D_A_E požymių tipą su padidinta keptrinio koeficiento skaitine verte. Todėl galima teigti, kad atpažinimo tikslumas priklauso nuo panaudoto požymių tipo ir keptrinio skaičiaus. Todėl skaičių ir vardų atpažinimo tikslumo tyrimai, naudojant HTK programinį įrankį, bus atliekami naudojant MFCC_D_A_E požymių tipą su padidinta keptrinio koeficiento skaitine verte. Pasiiektas atpažinimo tikslumas nėra pakankamas, todėl kituose tyrimuose norint pasiekti geriausią atpažinimo tikslumo rezultatą, komandų modeliuose papildomai tirsime būsenų ir Gauso mišinių skaičiaus poveikį balso komandų atpažinimo tikslumui.

3.4 Vardų atpažinimo tyrimas naudojant žodžiais gristus modelius

Pagal 2.4.1.2 skyriuje aprašyto HTK programinių įrankių paketo taikymo balso komandų atpažinimui tirti, naudojant žodžiais gristų modelių metodą, buvo sumodeliuoti vardų garsyno komandų paslėptieji Markovo modeliai, kurių atpažinimo tikslumo rezultatai, naudojant skirtingą modelių būsenų ir Gauso mišinių skaičių, aprašomi šiame skyriuje.

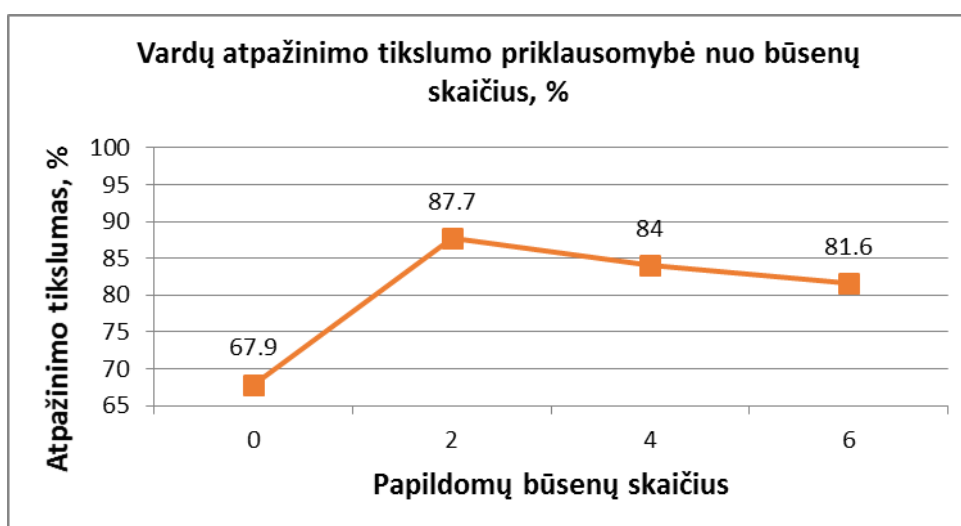
- **Papildomų būsenų įterpimas į žodžių modelius**

Atpažinimo tikslumo tyrimo, detalai aprašyto 2.4 skyriuje, naudojant modelius su skirtingais būsenų skaičiais, rezultatai procentais pateikti 3.4.1 lentelėje.

3.4.1 lentelė. Atpažinimo tikslumo tyrimų rezultatai naudojant modelius su skirtingais būsenų skaičiais

Būsenų skaičius	Pirminis (0)	+2	+4	+6
Atpažinimo tikslumas, %	67,9±13,15	87,7±9,29	84±10,58	81,6±11,98

Buvo manoma, kad didinant būsenų skaičių komandų modeliuose gerės vardų atpažinimo tikslumas, tačiau tai neįvyko. Būsenų didinimas vardų garsyno atveju nebuvo naudingas, nes tyrimui naudojami vardai nėra labai ilgi, todėl didėjantis būsenų skaičius neilgiems vardams yra per didelis. Ši tendencija buvo pastebėta atlikus tyrimus komandų modeliuose didinat būsenų skaičių. Atpažinimo tikslumo priklausomybė procentais nuo būsenų skaičiaus pateikta 3.4.1 paveiksle pavaizduotoje ašinėje diagramoje.



3.4.1 pav. Vardų atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus

Iš 3.4.1 lentelėje pateiktos vardų atpažinimo tikslumo priklausomybės nuo būsenų skaičiaus galime daryti išvadą, kad vardų garsyno komandų modeliai su 2 papildomomis būsenomis atpažinami didžiausiu tikslumu (87,7±9,29%) lyginant su atpažinimo tikslumu kai modeliuose

buvo naudojama 4, 6 būsenos ar visai nenaudojamos papildomos būsenos. Prasčiausias atpažinimo rezultatas buvo gautas kai modeliams nebuvo pridėamos papildomos būsenos ($67,9 \pm 13,15\%$). Pridėjus ir 4 ir 6 būsenas komandų modeliuose, atpažinimo tikslumas, nepaisant nuoseklaus mažėjimo didinat būsenų skaičių, vis tiek gaunamas aukštesnis nei, kad atpažinimo tikslumas visai nepridėjus papildomų būsenų komandų modeliams. Dėl to galima daryti prielaidą, kad modeliams pridėti papildomų būsenų yra naudinga, nes didėja atpažinimo tikslumas, bet reikia surasti tinkamą papildomų būsenų skaičių, nes padauginus papildomų būsenų modelyje, galima pakenkti atpažinimo tikslumo kokybei.

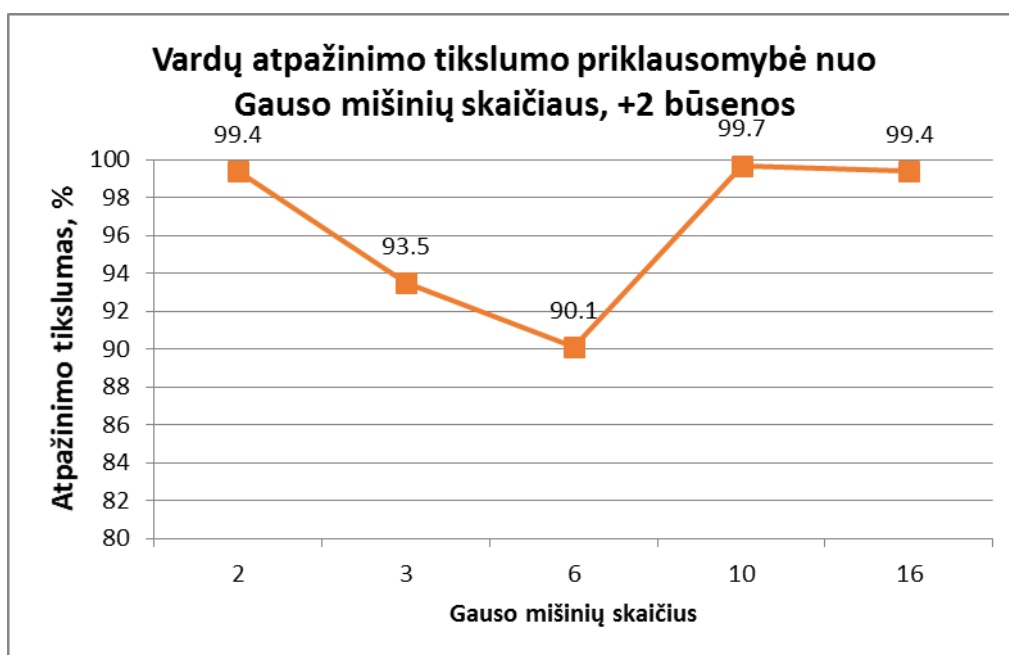
- ***Papildomų Gauso mišinių įterpimas į žodžių modelius***

Tyrimo, detaliam aprašyto 2.4.1.2 skyriuje, gautieji komandų atpažinimo tikslumo rezultatai procentais, naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi, pateikiami 3.4.2 lentelėje.

3.4.2 lentelė. Atpažinimo tikslumo tyrimo rezultatai naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi

+2 papildomos būsenos					
Gauso mišinių skaičius	2	3	6	10	16
Atpažinimo tikslumas, %	99,4±0,56	93,5±6,46	90,1±8,96	99,7±0,29	99,4±0,54

Gauso mišinių įterpimas į vardų komandų modelius akivaizdžiai pagerino atpažinimo tikslumo rezultatus. Iš 3.4.2 lentelėje pateiktos vardų atpažinimo tikslumo priklausomybės nuo Gauso mišinių skaičiaus matome, kad vardų garsyno komandų modeliai su 10 Gauso mišinių atpažįstami didžiausiu tikslumu ($99,7 \pm 0,29\%$). Prasčiausias atpažinimo rezultatas buvo gautas kai modeliams buvo įterpta 6 Gauso mišiniai ($90,1 \pm 8,96\%$). Pridėjus 10 ir 16 Gauso mišinių į komandų modelius, atpažinimo tikslumas gaunamas aukštas, tačiau tai galima būtų laikyti dirbtiniu tikslumu, nes šiais atvejais mišinių kiekis yra per didelis. Tai reiškia, kad sukuriama viendiktorinė sąlyga - kiekvienam diktoriui sukuriamas nuosavas skirstinys. Tokiu atveju atpažinimo procesas yra paviršutiniškas, nes atpažinimo programa neišanalizuoja smulkių detalių. Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.4.2 paveiksle.



3.4.2 pav. Vardų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, +2 būsenos

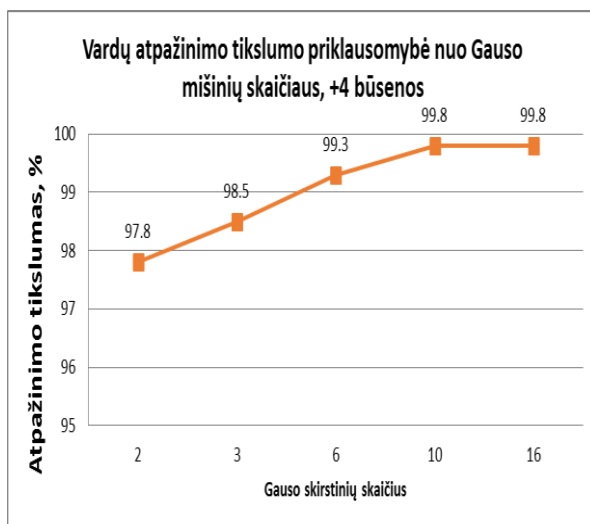
Tyrimo metu gautieji komandų atpažinimo tikslumo rezultatai procentais, naudojant modelius su keturiomis papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi, pateikiami 3.4.3 lentelėje.

3.4.3 lentelė. Atpažinimo tikslumo tyrimo rezultatai naudojant modelius su keturiomis papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi

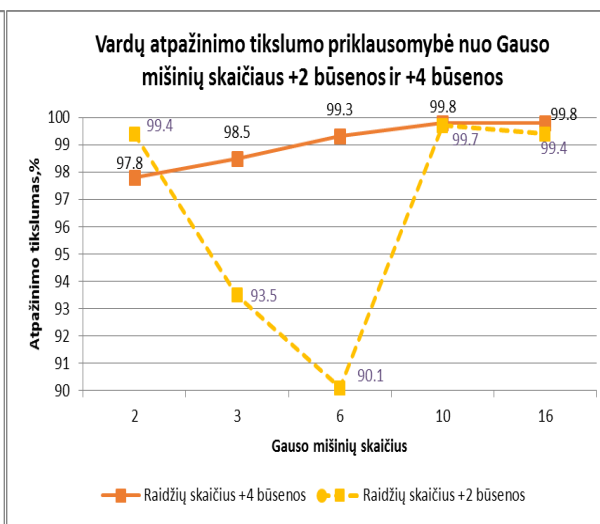
<i>+4 papildomos būsenos</i>					
Gauso mišinių skaičius	2	3	6	10	16
Atpažinimo tikslumas, %	97,8±2,15	98,5±1,46	99,3±0,16	99,8±0,17	99,8±0,17

Iš 3.4.3 lentelėje pateiktos vardų atpažinimo tikslumo priklausomybės nuo Gauso mišinių skaičiaus matome, kad vardų garsyno komandų modeliai su 10 ir 16 Gauso mišinių atpažįstami didžiausiu tikslumu (99,8±0,17%). Prasčiausias atpažinimo rezultatas buvo gautas kai modeliams buvo įterpta 2 Gauso mišiniai (97,8±2,15%). Aukšti rezultatai gauti su 10 ir 16 mišinių negali būti laikomi tiksliais, nes kaip ir pirmajame tyrimo etape, didelis kiekis skirstinių paverčia atpažinimo procesą paviršutinišku ir viendiktoriniu - kai kiekvienam diktoriui yra sukuriamas nuosavas skirstinys.

Gauso mišinių skaičiaus didinimas komandų modeliuose vardų garsyno atveju buvo naudingas, nes tai leido pasiekti aukščiausius vardų atpažinimo tikslumo rezultatus. Atpažinimo tikslumo priklausomybė procentais nuo Gauso mišinių skaičiaus pateikta 3.4.3 paveiksle pateiktoje ašinėje diagramoje.



3.4.3 pav. Vardų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, +4 būsenos



3.4.4 pav. Vardų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus +2 būsenos ir +4 būsenos

3.4.4 paveiksle pateiktoje ašinėje diagramoje pateikiamos vardų komandų modelių, su papildomomis 2 ir 4 būsenomis, atpažinimo tikslumo rezultatų priklausomybės nuo skirtingo Gauso mišinių skaičiaus. Punktyrine linija (+2 būsenos) vaizduojami rezultatai radikaliai skiriasi nuo tiesia linija vaizduojamų rezultatų, nes šie rezultatai pasižymi staigiu kritimu (staigus kritimas – menkas atpažinimas, kai didinamas mišinių skaičius). Tiesi linija rodo, jog papildomos 4 būsenos yra tinkamesnis parametras atpažinimui, todėl šia linija žymimiems rezultatams būdingas tolygus kilimas, rodantis rezultatų gerėjimą didinant mišinių skaičių. Todėl galima teigti, kad modeliams, parinkus tinkamą papildomų būsenų skaičių, įterpti tinkamą Gauso mišinių kiekį yra naudinga, nes didėja atpažinimo tikslumas. Tai galime pastebėti 3.4.4 paveiksle pateiktoje diagramoje.

Apibendrinant visus šiame skyriuje gautus, vardų balso komandų atpažinimo tikslumo tyrimų, rezultatus didinant papildomų būsenų ir Gauso mišinių skaičių komandų modeliuose, galime teigti, kad šie parametrai daro įtaką balso komandų atpažinimo tikslumui. Peržvelgus tyrimo metu gautuosius atpažinimo rezultatus, geriausiu parametru rinkiniu vardų garsyno atveju galima laikyti - modelius su papildomomis 4 būsenomis ir 6 Gauso mišiniais ($99,3 \pm 0,16\%$), neįtraukiant į rezultatų vertinimą atpažinimo rezultatų, kai būsenų ir mišinių skaičius per didelis. Šiame darbe atliktais tyrimais bus galima remtis tolimesniuose darbuose bei pratęsti tyrimą su didesniais duomenų kiekiais, kas reikalauja didelių laiko išteklių.

Vardų balso komandų sąrašas, naudojamas atpažinimo tikslumo tyrime bei duomenys, pagal kuriuos buvo apskaičiuoti vardų atpažinimo tikslumas pagal naudojamą būsenų ir Gauso mišinių skaičių yra pateikiami 3.4.4, 3.4.5, 3.4.6 lentelėse, kurios pateikiama 4 PRIEDE.

3.5 Skaičių atpažinimo tyrimas naudojant žodžiais gristus modelius

Pagal 2.4.1.3 skyriuje aprašyto HTK programinių įrankių paketo taikymo balso komandų atpažinimui tirti, naudojant žodžiais gristus modelius, buvo sumodeliuoti skaičių garsyno komandų paslėptieji Markovo modeliai, kurių atpažinimo tikslumo rezultatai, naudojant skirtingų modelių būsenų ir Gauso mišinių skaičių, aprašomi šiame skyriuje.

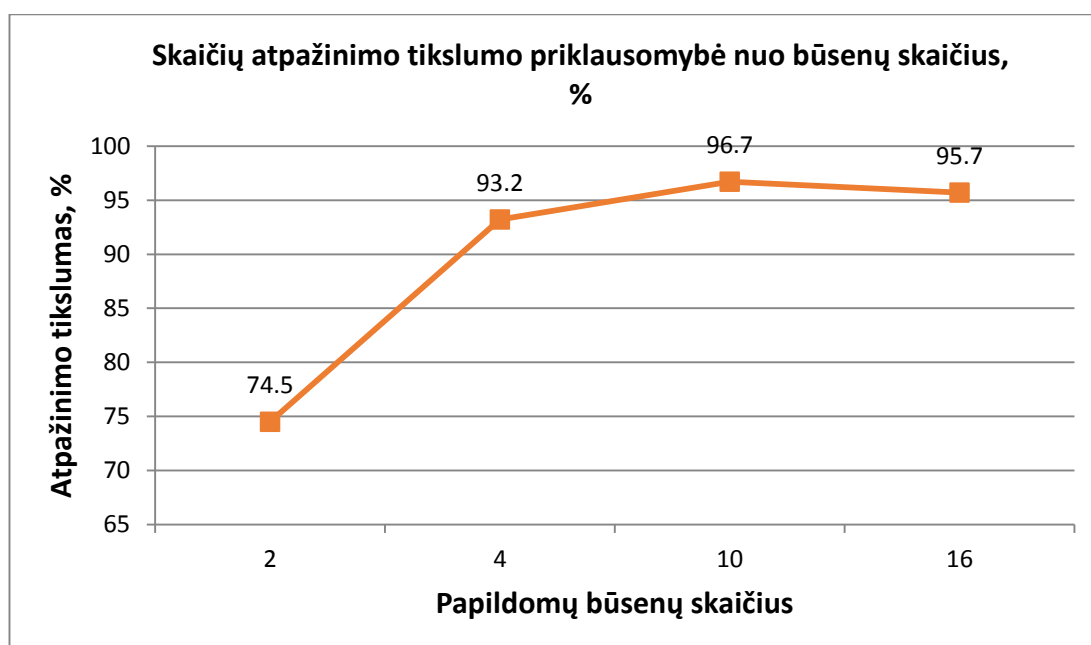
- **Papildomų būsenų įterpimas į žodžių modelius**

Tyrimo, detaliai aprašyto 2.4 skyriuje, naudojant modelius su skirtingais būsenų skaičiais, rezultatai procentais pateikti 3.5.1 lentelėje.

3.5.1 lentelė – Atpažinimo tikslumo tyrimų rezultatai naudojant modelius su skirtingais būsenų skaičiais

Būsenų skaičius	+2	+4	+10	+16
Atpažinimo tikslumas, %	74,5±18,23	93,2±6,44	96,7±1,48	95,7±1,98

Iš 3.5.1 lentelėje pateiktų tyrimo rezultatų matome, kad didinant būsenų skaičių, atpažinimo tikslumas didėja tik iki tam tikros ribos. Šią tendenciją atskleidė atpažinimo sistemos testavimas su šešiolika papildomų (+16) būsenų. Atpažinimo tikslumo priklausomybė procentais nuo būsenų skaičiaus pateikta 3.5.1 paveiksle pavaizduotoje ašinėje diagramoje.



3.5.1 pav. Skaičių atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus

Iš ašinės diagramos 3.5.1 paveiksle pateiktos skaičių atpažinimo tikslumo priklausomybės nuo būsenų skaičiaus galime daryti išvadą, kad skaičių garsyno komandų modeliai su 10 papildomų būsenų atpažįstami didžiausiu tikslumu (96,7±1,48%) lyginant su atpažinimo tikslumu kai modeliuose buvo naudojama 2, 4, 16 papildomų būsenų. Prasciausias atpažinimo rezultatas

buvo gautas kai modeliams buvo pridedamos 2 papildomos būsenos ($74,5 \pm 18,23\%$). Pridėjus ir 4 ir 10 būsenų į komandų modelius, atpažinimo tikslumas, nuosekliai didėja, tik atpažinimo tikslumas pridėjus 16 papildomų būsenų komandų modeliams jau šiek tiek nukrinta ($95,7 \pm 1,98\%$).

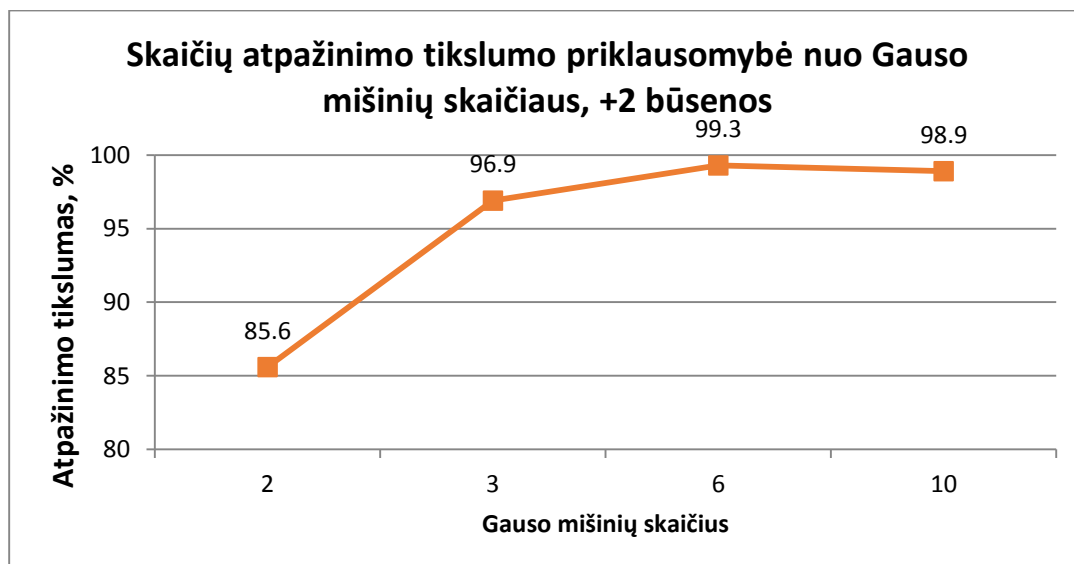
- **Papildomų Gauso mišinių įterpimas į žodžių modelius**

Tyrimo metu, detaliai aprašyto 2.4.1.3 skyriuje, gautieji atpažinimo tikslumo rezultatai procentais, naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi, pateikiami 3.5.2 lentelėje.

3.5.2 lentelė. Atpažinimo tikslumo tyrimo rezultatai naudojant modelius su dviem papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi

+2 papildomos būsenos				
Gauso mišinių skaičius	2	3	6	10
Atpažinimo tikslumas, %	$85,6 \pm 14,35$	$96,9 \pm 3,12$	$99,3 \pm 0,66$	$98,8 \pm 1,18$

Gauso mišinių įterpimas į skaičių komandų modelius pagerino atpažinimo tikslumo rezultatus. Iš 3.5.2 lentelėje pateiktos skaičių atpažinimo tikslumo priklausomybės nuo Gauso mišinių skaičiaus matome, kad Kai įterpiami 2 mišiniai gaunamas žemiausias atpažinimo tikslumas - $85,6 \pm 14,35\%$, o kai įterpiami 6 mišiniai gaunamas aukščiausias atpažinimo tikslumas – $99,3 \pm 0,66$. Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.5.2 paveiksle.



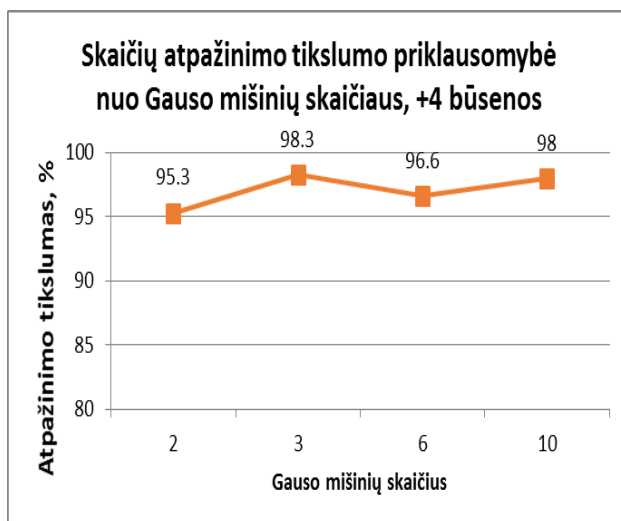
3.5.2 pav. Skaičių atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, +2 būsenos

Tyrimo metu, gautieji komandų atpažinimo tikslumo rezultatai procentais, naudojant modelius su keturiomis papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi, pateikiami 3.5.3 lentelėje.

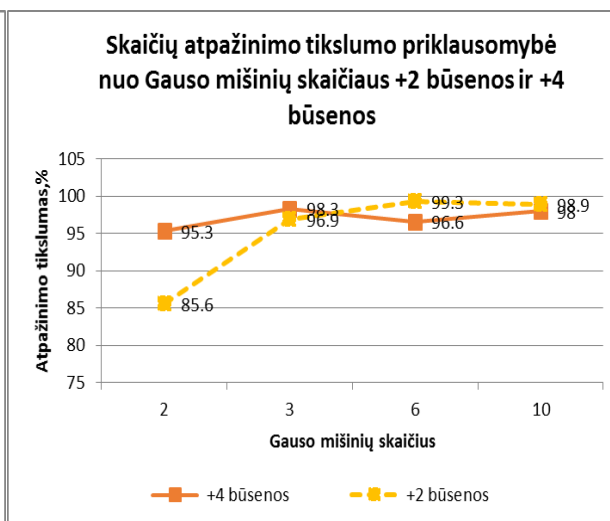
3.5.3 lentelė. Atpažinimo tikslumo tyrimo rezultatai naudojant modelius su keturiomis papildomomis būsenomis ir skirtingu Gauso mišinių skaičiumi

<i>+4 papildomos būsenos</i>				
Gauso mišinių skaičius	2	3	6	10
Atpažinimo tikslumas, %	95,3±2,35	98,3±1,45	96,6±2,59	98±1,53

Iš 3.5.3 lentelėje pateiktos skaičių atpažinimo tikslumo priklausomybės nuo Gauso mišinių skaičiaus matome, kad skaičių garsyno komandų modeliai su 3 Gauso mišiniais atpažįstami didžiausiu tikslumu (98,3±1,45%). Prasčiausias atpažinimo rezultatas buvo gautas kai modeliams buvo įterpta 2 Gauso mišiniai (95,3±2,35%). Gauso mišinių skaičiaus didinimas komandų modeliuose skaičių garsyno atveju buvo naudingas, nes tai leido pasiekti aukščiausius skaičių atpažinimo tikslumo rezultatus. Atpažinimo tikslumo priklausomybę procentais nuo Gauso mišinių skaičiaus pateikta 3.5.3 paveiksle pateiktoje ašinėje diagramoje.



3.5.3 pav. Skaičių atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, +4 būsenos



3.5.4 pav. Skaičių atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus +2 būsenos ir +4 būsenos

3.5.3 paveiksle pateiktoje ašinėje diagramoje pateikiamos skaičių komandų modelių, su papildomomis 2 ir 4 būsenomis, atpažinimo tikslumo rezultatų priklausomybės nuo skirtingo Gauso mišinių skaičiaus. Modelių atpažinimo tikslumo rezultatus su papildomomis 2 būsenomis ir Gauso mišiniais atvaizduojanti punktyrinė linija skiriasi labai nedaug nuo tiesios linijos, vaizduojančios modelių atpažinimo tikslumo rezultatus su papildomomis 4 būsenomis ir Gauso mišiniais. Iš to galime spręsti, kad tinkamas būsenų ir Gauso mišinių pasirinkimas gali pagerinti balso komandų atpažinimo tikslumą.

Apibendrinant visus šiame skyriuje darytų skaičių balso komandų atpažinimo tikslumo tyrimų rezultatus didinant papildomų būsenų ir Gauso mišinių skaičių komandų modeliuose,

galime teigti, kad šie parametrai daro įtaką balso komandų atpažinimo tikslumui. Iš atpažinimo tikslumo rezultatų analizės - geriausiu parametų rinkiniu turimo skaičių garsyno modeliuose galima laikyti - modelius su papildomomis 2 būsenomis ir 6 Gauso mišiniais (99,3±0,66%).

Skaičių balso komandų sąrašas, naudojamas atpažinimo tikslumo tyrime bei duomenys, pagal kuriuos buvo apskaičiuoti skaičių atpažinimo tikslumas pagal naudojamą būsenų ir Gauso mišinių skaičių yra pateikiami 3.5.4, 3.5.5, 3.5.6 lentelėse, kurios pateikiamos 11 PRIEDE.

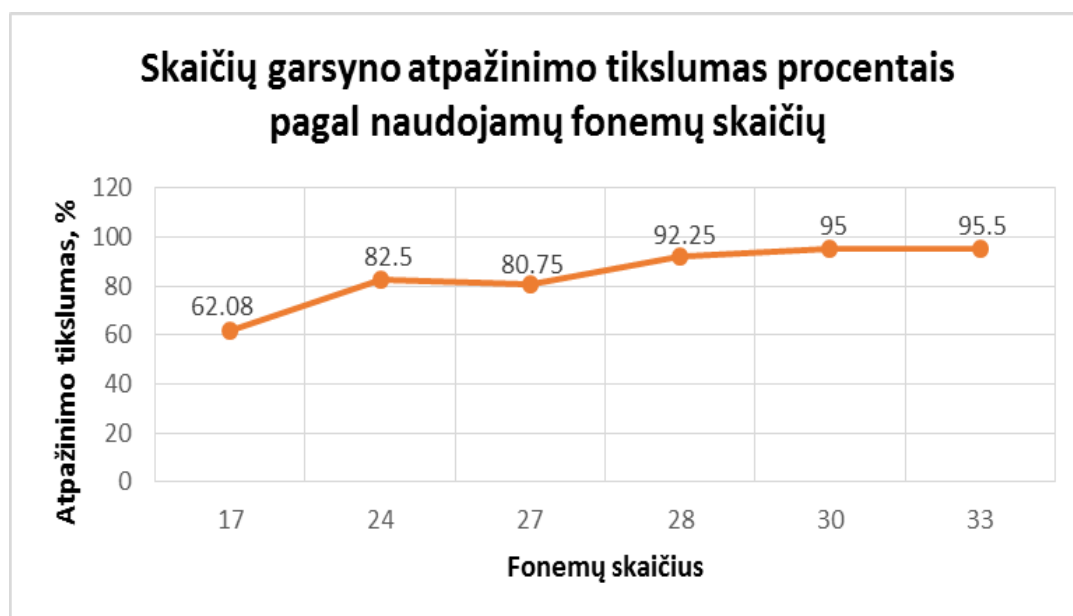
3.6 Skaičių atpažinimo tyrimas naudojant fonemomis gristus modelius

Atlikus tyrimą, detaliai aprašytą 2.4 skyriuje, gavome rezultatus pateiktus 3.6.1 lentelėje, o naudojamų fonemų sąrašas patiktas 3.6.2 lentelėje 6 PRIEDE.

3.6.1 lentelė. Skaičių Atpažinimo tikslumas procentais pagal fonemų skaičių

Fonemų skaičius	17	24	27	28	30	33
Atpažinimo tikslumas, %	62,08	82,50	80,75	92,25	95,00	95,50

Iš 3.6.1 lentelėje pateiktų rezultatų matome, kad skaičių komandų atpažinimo tikslumas yra aukščiausias (95,50%) kai naudojamos 33 fonemos, o žemiausias (62,08%) kai naudojama 17 fonemų. Didinant fonemų skaičių atpažinimo tikslumo rezultatai vis gerėjo palaipsniui, išskyrus, kai fonemų skaičius buvo padidintas iki 27, tada atpažinimo tikslumas nukrito 1,7% nuo prieš tai buvusios atpažinimo tikslumo vertės (82,50%) gautos naudojant 24 fonemas. Atpažinimo tikslumo priklausomybė nuo fonemų skaičiaus pateikta 3.6.1 paveiksle pavaizduotoje ašinėje diagramoje.



3.6.1 pav. Skaičių atpažinimo tikslumo procentais priklausomybė nuo naudojamų fonemų skaičiaus

Iš diagramos pateiktos 3.6.1 paveiksle galime spręsti apie skaičių komandų atpažinimo tikslumo priklausomybę nuo fonemų skaičiaus. Atpažinimo tikslumas po truputį augo didinant fonemų skaičių iki galutinio pasirinkto fonemų skaičiaus. Pradinis skaičių komandų atpažinimo tikslumas naudojant 17 fonemų yra - 62,08%. Padidinus fonemų skaičių iki 24 atpažinimo tikslumas pakilo 20,42% (82,5%). Padidinus fonemų skaičių iki 27 atpažinimo tikslumas nukrito nuo prieš tai buvusios vertės 1,7% (80,75%). Dar didinant fonemų skaičių atpažinimo tikslumas vis gerėja – kai naudojamos 28 fonemos atpažinimo tikslumas pakilo 11,5% (92,25%), kai 30 – pakilo 2,75% (95,00%), o kai naudojamos 33 fonemos, atpažinimo tikslumas yra aukščiausias, kuris pakilo 0,5% (95,5%) nuo prieš tai buvusios vertės.

Apibendrinant skaičių komandų atpažinimo tikslumo priklausomybės nuo tinkamo foneminių segmentų skaičiaus rezultatus, galima teigti, kad tinkamas foneminių segmentų skaičiaus parinkimas teigiamai įtakoja balso komandų atpažinimo tikslumą. Tyrimo metu fonemų skaičius buvo didinamas vis įtraukiant nauju fonemų, kurios manoma, kad geriau reprezentuos ilgesnius, duslius ar skardžius garsus, bet didinant fonemų skaičių neatsižvelgiant į kalbos ypatybes galima stipriai pabloginti atpažinimo tikslumo rezultatus, nes atpažintuvas pradės painioti teisingus garsus su neteisingais. Todėl naujų fonemų įtraukimo į komandų gramatikas yra atsakingas uždavinys reikalaujantis plačių fonetinių ir fonologinių žinių.

3.7 Atliktų tyrimų rezultatų palyginamoji analizė

Atliekant vardų ir skaičių pavadinimų atpažinimo tikslumo tyrimus, buvo naudojamos dvi skirtingos atpažinimo sistemos, tai ispaniškas kalbos atpažintuvas (*Microsoft Speech recognizer 8.0 for Windows*) ir lietuviškas realizuotas su HTK programinių įrankių paketu, pagrįstu paslėptaisiais Markovo modeliais. Naudojant HTK programinių įrankių paketą realizuotą kalbos atpažinimo sistemą buvo taikomi du kalbos atpažinimo modeliavimo metodai – žodžių modelių ir fonemų modelių (tik skaičių atpažinimo tikslumo tyrimui) metodai.

Atlikus komandų atpažinimo tikslumo tyrimus, naudojantis šiomis atpažinimo sistemomis tikslinga palyginti gautuosius rezultatus norint nuspręsti kuri atpažinimo sistema buvo pati efektyviausia komandų atpažinime.

➤ Skaičių garsyno atpažinimo tikslumo tyrimo rezultatų naudojant skirtingas atpažinimo sistemas rezultatų palyginamoji analizė

Pirmiausia yra palyginti skaičių komandų atpažinimo tikslumo rezultatai gauti su HTK programinių įrankių paketu realizuota kalbos atpažinimo sistema naudojant žodžių modeliavimo metodą ir fonemų modeliavimo metodą. Atpažinimo tikslumo tyrimo eiga naudojant šiuos metodus yra skirtinga, ir tyrimai daryti naudojant skirtingą metodiką, todėl palyginimui buvo naudojami patys geriausi skaičių komandų atpažinimo tikslumo rezultatai. Taikant žodžių modelių

metodą palyginamajai analizei buvo paimti atpažinimo tikslumo rezultatai gauti naudojant skaičių komandų modelius su papildomomis 2 būsenomis ir 6 Gauso mišiniais, taikant fonemų modelių metodą palyginamajai analizei buvo paimti atpažinimo tikslumo rezultatai gauti naudojant fonemų modelius su 33 fonemomis. Skaičių komandų, naudojant HTK programinį paketą, geriausi atpažinimo tikslumo rezultatai palyginimui, pateikti 3.7.1 lentelėje.

3.7.1 lentelė. Skaičių komandų atpažinimo tikslumo rezultatai gauti skirtingais modeliavimo metodais.

Modeliavimo metodas	Žodžių modeliai	Fonemų modeliai
Atpažinimo tikslumas, %	99,3±0,66	95,5

Apibendrinant 3.7.1 lentelėje pateiktus rezultatus matome, kad skaičių komandų modeliavimui naudojant tiek žodžių modelius tiek fonemų modelius atpažinimo tikslumo rezultatai yra aukšti – viršija 95%, bet atpažinimo tikslumas, gautas taikant žodžių modelius buvo aukštesnis – 3,8% (99,3±0,66%), nei atpažinimo tikslumas gautas taikant fonemų modelius (95,5%).

Taip pat geriausi rezultatai, gauti su HTK programinių įrankių paketu realizuota kalbos atpažinimo sistema taikant skaičių komandoms žodžių modelių metodą, kai komandų modeliai buvo papildomi 2 papildomomis būsenomis ir 6 Gauso mišiniais, buvo palyginami su geriausiais rezultatais gautais naudojant ispanišką kalbos atpažinimo sistemą asinchroniniame režime. Atliekant skaičių komandų atpažinimo tikslumo testavimus su ispanų kalbos atpažintuvu, naudojami visų trisdešimties diktorių balso įrašai, o testuojant su HTK programiniu paketu – šešių diktorių. Su skirtingais atpažintuvais gauti skaičių atpažinimo tikslumo rezultatai pateikti 3.7.2 lentelėje.

3.7.2 lentelė. Skirtingų atpažinimo sistemų rezultatų palyginimas

Atpažinimo sistema	HTK	Kitakalbis atpažintuvas
Atpažinimo tikslumas, %	99,3± 0,66	93,7 ±4,47

Lyginant skirtingas šnekos atpažinimo sistemas, naudojami geriausi gauti skaičių atpažinimo rezultatai. Šiuo atveju, aukščiausias skaičių atpažinimo tikslumo, naudojant HTK programinių įrankių paketu realizuota kalbos atpažinimo sistema, vidurkis siekia 99,3±0,66%. Naudojant ispanišką kalbos atpažintuvą, geriausi skaičių atpažinimo rezultatai – 93,7±4,47% - gauti, kai sistema veikė asinchroniniame darbo režime. Iš 3.7.2 lentelėje pateiktų atpažinimo tikslumo rezultatų, matome, kad atpažinimo tikslumo rezultatai gauti 5,6% aukštesni su HTK realizuota atpažinimo sistema.

➤ Vardų garsyno atpažinimo tikslumo tyrimo rezultatų naudojant skirtingas atpažinimo sistemas rezultatų palyginamoji analizė

Taip pat yra palyginami ir vardų komandų atpažinimo tikslumo rezultatai gauti su HTK programinių įrankių paketu realizuota kalbos atpažinimo sistema naudojant žodžių modeliavimo metodą. Taikant žodžių modelių metodą palyginamajai analizei buvo paimti atpažinimo tikslumo rezultatai gauti naudojant skaičių komandų modelius su papildomomis 4 būsenomis ir 6 Gauso mišiniais, buvo palyginami su geriausiais rezultatais gautais naudojant ispanišką kalbos atpažinimo sistemą naudojant diktorius balsu apmokytą atpažinimo profilį „Iveta1“ ir pradines, papildytas naujomis penkių prasčiausiai atpažįstamų balso komandų transkripcijomis, gramatikas. Atliekant vardų komandų atpažinimo tikslumo testavimus su ispanų kalbos atpažintuvu, naudojami visų dešimties diktorių balso įrašai, o testuojant su HTK programiniu paketu – dviejų diktorių. Su skirtingais atpažintuvais gauti vardų atpažinimo tikslumo rezultatai pateikti 3.7.3 lentelėje.

3.7.3 lentelė. Skirtingų atpažinimo sistemų rezultatų palyginimas

Atpažinimo sistema	HTK	Kitakalbis atpažintuvas
Atpažinimo tikslumas, %	99,3 ±0,16	97,1±1,6

Lyginant skirtingas šnekos atpažinimo sistemas, naudojami geriausi gauti vardų atpažinimo rezultatai. Šiuo atveju, aukščiausias atpažinimo tikslumo, naudojant HTK programinių įrankių paketu realizuota kalbos atpažinimo sistema, vidurkis siekia 99,3±0,16%. Naudojant ispanišką kalbos atpažintuvą, geriausi vardų atpažinimo rezultatai – 97,1±1,6% - gauti, kai pradinės komandų transkripcijos buvo papildytos naujomis, 5 prasčiausiai atpažįstamų vargų komandų, transkripcijomis. Iš 3.7.3 lentelėje pateiktų atpažinimo tikslumo rezultatų, matome, kad atpažinimo tikslumo rezultatai gauti 2,2% aukštesni su HTK atpažinimo sistema.

3.8 Šnekos atpažintuvų sujungimo galimybių tyrimas

Duomenų analizės sistemoje WEKA esančių klasifikatorių įtakos tyrimas atpažinimo hibridinės sprendimo taisyklės efektyvumui, buvo atliekamas analizuojant - RIPPER, C4.5, MLR, RF, ZeroR, kNN, SVM, AdaBoost, NB ir MP klasifikatorių efektyvumą.

Hibridinė sprendimo priėmimo taisyklė buvo mokoma ir tikrinama 12-kartų kryžminio patikrinimo būdu ir įprastu 10-kartų kryžminio patikrinimo būdu. Šio eksperimento rezultatai pateikti 3.8.1 lentelėje. 10-k. kryžminio patikrinimo testavimo laiko WEKA duomenų analizės paketas nepateikia.

3.8.1 lentelė. Dviejų klasių klasifikavimo tikslumas, % ir testavimui sugaištamas laikas, s

Klasifi- katorius	12-kartų kryžminis		10-k. kryžminis
	Atpažinimo tikslumas, %	Testavimo laikas, s	Atpažinimo tikslumas, %
RIPPER	98,37	0,46	98,75
C4.5	98,44	0,49	98,87
MLR	98,26	0,53	98,31
RF	99,16	1,31	99,46
ZeroR	96,79	0,48	96,13
kNN	98,26	278	99,32
SVM	98,52	0,59	98,53
AdaBoost	97,60	0,47	97,22
NB	87,55	1,65	88,07
MP	98,54	2,20	99,33

Apžvelgus 3.8.1 lentelėje pateiktus rezultatus matosi, kad išskeltam uždaviniui netinka kNN klasifikatorius dėl didelio testavimui sugaištamo laiko. Geriausi klasifikavimo rezultatai abiem atvejais gauti naudojant RF klasifikatorių (naudotas numatytasis medžių skaičius 100). Klasifikatorius suteikia galimybę keisti medžių skaičių, todėl, buvo nuspręsta atlikti RF klasifikatoriaus RF efektyviausio medžių skaičiaus paieškos tyrimą. Tam buvo atliktas 10-kartų kryžminio patikrinimo eksperimentas, mokymui naudojant 90% objektų, o testavimui naudojant 10% objektų ir keičiant RF klasifikatoriaus medžių skaičių nuo 1 iki 400. Dalis eksperimento metu gautų rezultatų pateikta 3.8.2 lentelėje.

3.8.2 lentelė. Klasifikavimo tikslumo priklausomybė nuo RF klasifikatoriaus medžių skaičiaus

Medžių skaičius	2	9	25	69	100
Tikslumas, %	98,51	99,26	99,39	99,46	99,46

Atlikus eksperimentą ir išanalizavus gautuosius rezultatus pastebėta, kad didinant medžių skaičių klasifikavimo tikslumas palaipsniui didėja ir pasiekia maksimumą, kai medžių skaičius lygus 69. Toliau didinant medžių skaičių tikslumas nedidėja.

Apskaičiavus 10-kartų kryžminio patikrinimo metodu gautus rezultatus gavome, kad RF klasifikatoriaus išmokta hibridinė sprendimo priėmimo taisyklė veikia 99,46% tikslumu ir viršija „aklos“ sprendimo priėmimo taisyklės tikslumą (96,12%). Tai reiškia, kad hibridinis atpažintuvas teisingai atpažįsta visus T=T poaibio įrašus, (135898), visus T- poaibio įrašus (3398) ir 99,46% tikslumu atpažįsta TF + FT poaibių įrašus (34817 iš 35007) (žr. 1.5.3 lent., 7 PRIEDAS). Vidutinis hibridinio atpažintuvo veikimo tikslumas - 99,24% ((135898+3398+34817)/175440). Šis rezultatas galioja tuo atveju, kai atpažintuvas atpažįsta vieno iš žinomų 12 diktorių šneką.

Apskaičiavus 12-kartų kryžminio patikrinimo metodu gautus rezultatus gavome, kad RF klasifikatoriaus išmokta sprendimo priėmimo taisyklių aibė veikia 99,16% tikslumu. Atsižvelgiant į tai, kad sprendimo taisyklė iškviečiama tik tada kai ATP_SP ir ATP_LT sprendimai skiriasi, vidutinis hibridinio atpažintuvo veikimo tikslumas - 99,18%. Šis rezultatas galioja tuo atveju, kai atpažintuvas atpažįsta nežinomo kalbėtojo šneką.

Pagal 2.5 skyriuje aprašytą metodiką atliktas mokymo imties objektams aprašytų požymių įtakos hibridinio atpažintuvo sprendimo priėmimo tikslumui tyrimas. Tyrimo metu gautieji tikslumo rezultatai pateikti 3.8.1 lentelėje.

3.8.1 lentelė. Požymių derinių įtaka sprendimo priėmimo tikslumui

Klasifikatorius – RandomForest (RF), medžių skaičius (I) - 9	
Požymių deriniai	Tikslumas, %
visi	99,26
visi(no lt_prob)	99,30
visi(no sp_prob)	99,30
visi(no_delta)	99,23
visi(no_sil)	99,27
visi(no_supp)	99,19
visi(no_supp_no_delta))	98,80
visi(no_supp_no_delta_no_sil)	98,80
visi(no_gender)	99,3
visi(no_gender_no_delta)	99,21
visi(no_gender_no_letters)	97,48
visi(no_gender_no_letters_no_sil)	97,68
visi(no_gender_no_supp)	99,23
visi(no_gender_no_supp_no_delta)	98,73
visi(no_gender_no_supp_no_delta_no_sil)	98,76
visi(no_letters_no_space)	97,50
visi(no_letters_no_space_no_gender)	97,46
visi(no_letters_no_space_no_gender_no_sil)	97,27
visi(no_letters_no_space_no_gender_no_sil_no_delta)	97,07
visi(no_letters_no_space_no_gender_no_sil_no_supp)	96,76
visi(no_letters_no_space_no_gender_no_sil_no_supp_no_delta)	95,4
visi(tik lt_prob)	94,8
visi(tik sp_prob)	96,06

Peržvelgus 3.8.1 lentelėje pateiktus sprendimo priėmimo tikslumo rezultatus keičiant požymių derinius – juos mažinant, buvo pastebėta kad didžiausią įtaką tikslumui daro - „tik lt_prob“ (sumažėjo 4,46 %), „tik sp_prob“ (sumažėjo 2,66 %) ir „no_letters_no_space_no_gender_no_sil_no_supp_no_delta“ (sumažėjo 3,86 %) požymių deriniai. Naudojant šiuos požymių derinius atpažinimo tikslumas sumažėjo. Taip pat matosi, kad yra ir požymių derinių su kuriais tikslumas beveik nepakito - „no sp_prob“, „no lt_prob“,

„no_sil“, „no_delta“, „no_gender“, „no_gender_no_delta“ ir „no_gender_no_supp“.
Naudojant kitus požymių derinius atpažinimo tikslumas krito, bet nežymiai.

Išanalizavus tyrimo metu gautuosius rezultatus, galime teigti, kad kai kurių požymių įtraukimas aprašant duomenų požymius nėra esminis, nes visai neįtakoja atpažinimo rezultatų, bet neįtraukus visai jokių požymių ar palikus labai nedaug, stebimas atpažinimo tikslumo sumažėjimas.

3.9 Ligų kodų atpažinimo demonstracinė programa

Buvo parengta demonstracinė komandų atpažinimo programa susieta su tyrime naudotomis vardų ir skaičių garsynų balso komandomis. Programa buvo pavadinta - „Ligų kodų atpažinimo programa“. Ši programa naudoja 10 skaičių garsyno balso komandų ir 26 vardų garsyno balso komandas. Programos užduotis atpažinti sakomas vardų ir skaičių komandas ir atpažinti ligos kodą į ekraną išvedant kodui priskirtą ligos pavadinimą. Ligų kodai ir pavadinimai naudojami iš TLK-10-AM sisteminio ligų sąrašo.

Programa buvo projektuojama *Windows 7* aplinkoje, programinis kodas buvo rašomas *C#* (*Sy – Sharp*) programavimo kalba, naudojant *Visual Studio 2010* programinį paketą. Programos kodo esminiai fragmentai su komentarais ir pavyzdžiais, pilnas programos veikimo kodas, naudojamų komandų sąrašai ir gramatikos pateikiami 12 PRIEDE.

IŠVADOS IR REZULTATAI

1. Apibendrinus atpažinimo tikslumo tyrimo su ispanišku atpažintuvu rezultatus, buvo nustatyta, kad izoliuotų komandų atpažinimo tikslumas priklauso nuo atpažintuvo darbo režimo (skaičių garsyno atpažinimo tikslumas asinchroniniame darbo režime yra 4,6% aukštesnis nei sinchroniniame);
 2. Atlikus šnekos signalo atpažinimui skirtų požymių įtakos tyrimą, buvo nustatytas efektyviausias atpažinimo tikslumui požymių tipas – melų skalės kepstro koeficientai ir energija bei pirmosios ir antrosios koeficientų ir energijos išvestinės;
 3. Atlikus vardų ir skaičių garsynų atpažinimo tikslumo tyrimą, taikant HTK paketą ir žodžiais grįstus PMM, aukščiausias skaičių garsyno atpažinimo tikslumas buvo gautas naudojant PMM su 2 papildomomis būsenomis ir 6 Gauso mišiniais ($99,3\pm 0,66\%$), o vardų - su 4 papildomomis būsenomis ir 6 Gauso mišiniais ($99,3\pm 0,16\%$);
 4. Atlikus skaičių garsyno atpažinimo tikslumo tyrimą, taikant HTK paketą ir fonemomis grįstus PMM, aukščiausias skaičių garsyno atpažinimo tikslumas buvo gautas naudojant 33 fonemas ($95,5\%$);
 5. Naudojant WEKA duomenų analizės sistemą ir projekto INFOBALSAS metu gautus duomenis nustatyta, kad tiksliausias klasifikatorius, taikytinas dviejų atpažintuvų sujungimui, yra “atsitiktinis miškas” (random forest - RF).
 6. Gauti vardų ir skaičių pavadinimų garsynų atpažinimo rezultatai:
 - 97,1±1,6% - vardų garsynas su ispanų kalbos atpažintuvu;
 - 93,7±4,47% - skaičių pavadinimų su ispanų kalbos atpažintuvu;
 - 99,3±0,16% - vardų garsynas su žodžiais grįstais PMM;
 - 99,3±0,66% - skaičių pavadinimų garsynas su žodžiais grįstais PMM;
 - 95,5% - skaičių pavadinimų garsynas su fonemomis grįstais PMM.
- Galima tikėtis rezultatų pagerėjimo apjungiant 2 arba 3 atpažintuvus naudojant mašininio mokymo metodiką.
7. Paruošta ir išbandyta demonstracinė ligų pavadinimų atpažinimo pagal kodą programa.

LITERATŪROS ŠALTINIAI

- 1. Lileikytė, R. (2012).** Šnekos atpažinimo požymių kokybės vertinimas: daktaro disertacija: Technologijos mokslai, informatikos inžinerija (07T) / Vilniaus universitetas. Interaktyvus [žiūrėta 2015-04-09]. Prieiga per internetą: http://www.mii.lt/files/mii_dis_2012_lileikyte.pdf
- 2. Fosberg, M. (2003).** Why is speech recognition difficult? Chalmers university, Departement of Computer science. Interaktyvus [žiūrėta 2015-03-17] Prieiga per internetą: http://www.speech.kth.se/~rolf/gslt_papers/MarkusForsberg.pdf
- 3. Jurafsky, D.; Martin, J. H. (2000).** Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall, Upper Saddle River, New Jersey 07458. Inc, 934 p.
- 4. Laurinčiukaitė, S. (2008).** Lietuvių šnekos atpažinimo akustinis modeliavimas: daktaro disertacija: technologijos mokslai, informatikos inžinerija 07T / Vilniaus Gedimino technikos universitetas, Vilnius: Technika. 134 p. Interaktyvus [žiūrėta 2015-03-29]. Prieiga per internetą: www.mii.lt/files/mii_dis_08_laurinciukaite.pdf
- 5. Vaičiukynas, E. (2009).** Balso dialogų technologijos (kompiuteriniai dialogai balsu): Informacinių technologijų metodai: Kauno technologijos universitetas (T120D004). Interaktyvus [žiūrėta 2015-04-05]. Prieiga per internetą: http://www.elen.ktu.lt/studentai/lib/exe/fetch.php?media=balso_dialogu_tech.pdf
- 6. Voroneckij, M. (2007).** Kompiuterio valdymas balsu: magistrinis darbas / Vilniaus pedagoginis universitetas. Vilnius. 63 p. Žr. 24p. [žiūrėta 2015-04-28]. Prieiga per internetą: http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2007~D_20070816_175147-23409/DS.005.0.01.ETD
- 7. Kasparaitis, P. (2005).** Kompiuterinės lingvistikos paskaitų konspektai. Interaktyvus [žiūrėta 2015-04-04]. Prieiga per internetą: <http://www.mif.vu.lt/~pijus/CL/cl.htm>
- 8. Vaičiūnas, A. (2006).** Lietuvių kalbos statistinių modelių ir jų taikymo šnekos atpažinimui tyrimas, kai naudojami labai dideli žodynai: daktaro disertacija: fiziniai mokslai, informatika 09P / Vytauto Didžiojo universitetas, Kaunas. Interaktyvus [žiūrėta 2015-04-19]. Prieiga per internetą: http://donelaitis.vdu.lt/disertacijos/Vaiciunas_2006.pdf
- 9. Kemėšis P., Stasiūnas A. (2001).** Klausos neuroninis tinklas: monografija. Kaunas: Technologija. 155 p.
- 10. Driaunys K. (2006).** Lietuvių šnekamosios kalbos segmentavimo ir fonetinio atpažinimo tyrimas naudojant LTDIGITS garsyno įrašus: daktaro disertacija: fiziniai mokslai, informatika / Vilniaus universitetas. Vilnius. 171 p.
- 11. Maskeliūnas R. (2009).** Lietuviškų balsu komandų atpažinimas daugybinių transkripcijų pagrindu: daktaro disertacija: technologijos mokslai, informatikos inžinerija – 07T / Kauno technologijos universitetas. Kaunas. 159 p.

- 12. Rudžionis, V., Ratkevičius, K., Rudžionis, A., Raškinis, G., Maskeliūnas, R. (2013).** “Recognition of voice commands using hybrid approach”, In: Proc. Of 19th International Conference, ICIST 2013, Kaunas, Lithuania, Lecture Notes Communications in Computer and Information Science, vol. 403, p. 186-197, Springer.
- 13. G. Bartišiūtė, G. Paškauskaitė (2015).** „Šnekos atpažintuvų sujungimo galimybių tyrimas“, E2TA-2015 konferencija, Kaunas, Lietuva. ISBN 978-609-02-1133-5. 20-23p.
- 14. Rudžionis A., Ratkevičius K., Rudžionis V., Kasparaitis P., Šalna B. (2001).** Balso technologijų taikymo lietuvių kalbai analizė ir perspektyvinių veiklos krypčių pagrindimas: ataskaita / Kauno technologijos universitetas, Vilniaus universitetas, Teismo ekspertizės centras. Kaunas, Vilnius. 84 p., [žiūrėta 2015-04-11]. Prieiga per internetą: http://www.likit.lt/frames/balso_tech/balsotech_st.htm.
- 15. Pranevičius H., Raudys Š., Rudžionis A., Rudžionis V., Ratkevičius K., Sakalauskaitė J., Makackas D. (2008).** Agentinių sistemų modeliai: mokomoji knyga. Vilnius: Mokslo aidai. 255 p. ISBN 978-9955-591-55-9.
- 16. Kasparaitis P. (2013).** Kompiuterinė lingvistika. Kalbos signalų apdorojimas. 12 p. [žiūrėta 2015-04-09]. Prieiga per internetą: <http://www.mif.vu.lt/~pijus/CLF/SignApd.pdf>.
- 17. Filipovič M. (2005).** Atskirai tariamų lietuvių šnekos žodžių atpažinimo, grindžiamo dirbtiniais neuroniniais tinklais ir paslėptais Markovo modeliais, tyrimai: daktaro disertacijos santrauka: fiziniai mokslai, informatika / Vytauto Didžiojo universitetas. Kaunas. 138 p.
- 18. Balbonas D., Daunys G. (2005).** Fonemų klasifikavimas panaudojant garso ir vaizdo informaciją // Elektronika ir elektrotechnika. Nr. 5(61). ISSN 1392-1215. Interaktyvus [žiūrėta 2015-04-15]. Prieiga per internetą: www.ee.ktu.lt/journal/2005/5/Balbonas.pdf
- 19. Tarptautinės statistinės ligų ir sveikatos sutrikimų klasifikacijos dešimtasys pataisytas ir papildytas leidimas Australijos modifikacija (TLK-10-AM) – Sisteminis ligų sąrašas ir Abėcėlinė ligų rodyklė.** Išleido Nacionalinis medicininės klasifikacijos centras (Sidnėjus), Sveikatos mokslų fakultetas, Sidnėjaus universitetas, NSW 1825 Australija. Tekstą surinko „NCCH Publications Division“, Australija. Interaktyvus [žiūrėta 2015-04-15]. Prieiga per internetą: <http://ebook.vlk.lt/e.vadovas/index.jsp>
- 20. Ž. Ringelienė, M. Filipovič (2011).** Žodžių atpažinimo, grįsto paslėptaisiais Markovo modeliais, vizualizavimo ir analizės programinė įranga. 10 p.
- 21. Young S., Evermann G., Kershaw D., Moore G., Odell J., Ollason D., Valtchev V., Woodland P.(2006)** The HTK Book (version 3.4). Microsoft Corporation. 386p.
- 22. Raškinis, A., Raškinis, G., Kazlauskienė, A. (2003).** Lietuvių kalbos kompiuteriniai tyrimai (IX sekcija): VDU bendrinės lietuvių šnekos universalus anotuotas garsynas / Vytauto Didžiojo universitetas. Interaktyvus [žiūrėta 2015-04-12]. Prieiga per internetą: http://www.ktu.lt/lt/apie_renginius/konferencijos/2006/k6_02/it2003/sekcija09.pdf

- 23. Balvočius, B., Telksnys, L. (2003).** Lietuvių kalbos kompiuteriniai tyrimai (IX sekcija): Garsynų duomenų modeliai ir programinės įrangos architektūros / Vytauto Didžiojo universitetas, Matematikos informatikos institutas. Interaktyvus [žiūrėta 2015-03-23]. Prieiga per internetą: http://www.elibrary.lt/resursai/Konferencijos/KTU_01/IT_2003/Sekcija09.pdf
- 24. Rudžionis, V., Ratkevičius, K., Rudžionis, A., Raškinis, G., Maskeliūnas, R. (2013).** “Hibridinė atpažinimo technologija balso sąsajai (INFOBALSAS)”. Aukštųjų technologijų plėtros 2011–2013 metų programa. Mokslo, inovacijų ir technologijų agentūra. Baigiamoji ataskaita. Kaunas, Lietuva, 80 p.
- 25. Y. Wang, H. Wang, Z.-G. Gu, „A Survey of Data Mining Softwares Used for Real Projects“ (2011).** *International Workshop on Open-Source Software for Scientific Computation (OSSC)*, Beijing, pp.94-97.
- 26. A. Jovic, K. Brkic, N. Bogunovic, „An Overview of free software tools for general data mining“ (2014).** *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, pp.1112-1117.
- 27. X. Chen, G. Williams, X. Xu, „A Survey of Open Source Data Mining Systems“ (2007),** *Emerging Technologies in Knowledge Discovery and Data Mining*, Springer, vol. 4819, pp.3-14.
- 28. A. H. Wahben, Q.A. Al-Radaideh, M.N. Alkabi, E.M. Shawakfa (2011).** „A Comparison Study between Data Mining Tools over some Classification Methods“, *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, vol. 0(3), p. 18–25.
- 29. K. Paulauskienė, O. Kurasova (2013).** „Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti“, *Informacijos mokslai*, 65, pp.85-94.
- 30. X. Wu, V. Kumar, J.R. Quinlan etc. (2007).** „Top 10 algorithms in data mining“, *Knowledge and Information Systems*, Springer, vol. 14, issue 1, pp. 1-37.

1 PRIEDAS. Garsynai

1.5.1 lentelė. Garsyno „Medic“ struktūra

BENDRAS GARSYNAS		
Katalogas	Komandų skaičius	Sinonimų skaičius
VAISTAI	46	0
VAISTAI2	53	3
VAISTAI3	50	2
VAISTAI4	51	1
VAISTAI5	50	0
VAISTAI6	50	0
LIGOS	49	7
LIGOS2	50	3
LIGOS3	50	7
LIGOS4	51	0
NUSISK	50	2
NUSISK2	50	6
NUSISK3	50	0
NUSISK4	50	0
Iš viso:	700	31

1.5.2 lentelė. Garsyno „Medic“ papildymo struktūra

GARSYNO PAPILDYMAS	
Katalogas	Komandų skaičius
NUSISK5	83
LIGOS5	92
VAISTAI7	94
Iš viso:	269

2.1.1 lentelė. Lietuviškų skaičių garsynas

Skaičius	Tarimas	Katalogas
0	<i>Nulis</i>	NULIS
1	<i>Vienas</i>	VIENAS
2	<i>Du</i>	DU
3	<i>Trys</i>	TRYS
4	<i>Keturi</i>	KETURI
5	<i>Penki</i>	PENKI
6	<i>Šeši</i>	SESI
7	<i>Septyni</i>	SEPTYNI
8	<i>Aštuoni</i>	ASTUONI
9	<i>Devyni</i>	DEVYNI

2.1.2 lentelė. Vardų garsynas

Nr.	Raidė	Vardas	Katalogas	Nr.	Raidė	Vardas	Katalogas
1.	A	<i>Austėja</i>	AUSTEJA	14.	N	<i>Nojus</i>	NOJUS
2.	B	<i>Boleslovas</i>	BOLESLOV	15.	O	<i>Oskaras</i>	OSKARAS
3.	C	<i>Cecilija</i>	CECILIJA	16.	P	<i>Patrikas</i>	PATRIKAS
4.	D	<i>Donatas</i>	DONATAS	17.	Q	<i>Kju</i>	KJU
5.	E	<i>Eimantas</i>	EIMANTAS	18.	R	<i>Ričardas</i>	RICARDAS
6.	F	<i>Fausta</i>	FAUSTA	19.	S	<i>Sandra</i>	SANDRA
7.	G	<i>Gražvydas</i>	GRAZVYDA	20.	T	<i>Teodoras</i>	TEODORAS
8.	H	<i>Hansas</i>	HANSAS	21.	U	<i>Ulijona</i>	ULIJONA
9.	I	<i>Izaokas</i>	IZAOKAS	22.	V	<i>Vacys</i>	VACYS
10.	J	<i>Jonas</i>	JONAS	23.	W	<i>Vašington</i>	VASINGTO
11.	K	<i>Karolis</i>	KAROLIS	24.	X	<i>Iksas</i>	IKSAS
12.	L	<i>Laima</i>	LAIMA	25.	Y	<i>Ygrekas</i>	YGREKAS
13.	M	<i>Martynas</i>	MARTYNAS	26.	Z	<i>Zacharijus</i>	ZACHARIJ

2 PRIEDAS. Gramatikos

2.1 Skaičių gramatika

```
<rule id="Rule" scope="public">
<one-of>
<item>
<one-of>
<item>nulejs</item>
<item>mulis</item>
<item>nuhlejs</item>
<item>nuhlejs</item>
</one-of>
<tag>$_value = "0nulis"</tag>
</item>
<item>
<one-of>
<item>vienas</item>
<item>vihehnahs</item>
</one-of>
<tag>$_value = "1vienas"</tag>
</item>
<item>
<one-of>
<item>doo</item>
<item>dou</item>
<item>douh</item>
<item>duo</item>
<item>dua</item>
</one-of>
<tag>$_value = "2du"</tag>
</item>
<item>
<one-of>
<item>tryis</item>
<item>trris</item>
<item>trriis</item>
<item>trriis</item>
<item>trys</item>
<item>trris</item>
<item>trris</item>
</one-of>
<tag>$_value = "3trys"</tag>
</item>
<item>
<one-of>
<item>kewturih</item>
<item>keaturih</item>
<item>keaturii</item>
<item>keturej</item>
<item>kevturii</item>
```

```
</one-of>
<tag>$_value = "4keturi"</tag>
</item>
<item>
<one-of>
<item>penkii</item>
<item>penkiih</item>
<item>penki</item>
</one-of>
<tag>$_value = "5penki"</tag>
</item>
<item>
<one-of>
<item>sheshii</item>
<item>chechii</item>
</one-of>
<tag>$_value = "6sesi"</tag>
</item>
<item>
<one-of>
<item>septinii</item>
<item>septiinii</item>
<item>septinii</item>
<item>septinej</item>
</one-of>
<tag>$_value = "7septyni"</tag>
</item>
<item>
<one-of>
<item>astuonii</item>
<item>ashtuhohnii</item>
<item>ashtuhohniih</item>
</one-of>
<tag>$_value = "8astuoni"</tag>
</item>
<item>
<one-of>
<item>deviinii</item>
<item>debini</item>
<item>dewini</item>
<item>deviiniih</item>
<item>dejjinese</item>
</one-of>
<tag>$_value = "9devyni"</tag>
</item>
</one-of>
</rule>
</grammar>
```

2.2.1 Vardų pirminė gramatika

```
<one-of>
<item>
<one-of>
<item>ahwstehyeh</item>
<item>austeja</item>
</one-of>
<tag>$_value = "austeja"</tag>
</item>
<item>
<one-of>
<item>baolehsloavahs</item>
<item>boleslovas</item>
</one-of>
<tag>$_value = "boleslovas"</tag>
</item>
<item>
<one-of>
<item>tsehtsihlihyeh</item>
<item>cecilija</item>
</one-of>
<tag>$_value = "cecilija"</tag>
```

```
</item>
<item>
<one-of>
<item>daonaatahs</item>
<item>donatas</item>
</one-of>
<tag>$_value = "donatas"</tag>
</item>
<item>
<one-of>
<item>ehiyamahntahs</item>
<item>eimantas</item>
</one-of>
<tag>$_value = "eimantas"</tag>
</item>
<item>
<one-of>
<item>fahwstah</item>
<item>fausta</item>
</one-of>
<tag>$_value = "fausta"</tag>
```

```

</item>
<item>
<one-of>
<item>graazhvyydahs</item>
<item>grazvydas</item>
</one-of>
<tag>$.value = "grazvydas"</tag>
</item>
<item>
<one-of>
<item>haansahs</item>
<item>hansas</item>
</one-of>
<tag>$.value = "hansas"</tag>
</item>
<item>
<one-of>
<item>ihzahaokahs</item>
<item>izaokas</item>
</one-of>
<tag>$.value = "izaokas"</tag>
</item>
<item>
<one-of>
<item>yaonahs</item>
<item>jonas</item>
</one-of>
<tag>$.value = "jonas"</tag>
</item>
<item>
<one-of>
<item>kahraolihs</item>
<item>karolis</item>
</one-of>
<tag>$.value = "karolis"</tag>
</item>
<item>
<one-of>
<item>laaiymah</item>
<item>laima</item>
</one-of>
<tag>$.value = "laima"</tag>
</item>
<item>
<one-of>
<item>mahrtiynahs</item>
<item>martynas</item>
</one-of>
<tag>$.value = "martynas"</tag>
</item>
<item>
<one-of>
<item>naoyuhs</item>
<item>nojus</item>
</one-of>
<tag>$.value = "nojus"</tag>
</item>
<item>
<one-of>
<item>aoskahrahs</item>
<item>oskaras</item>
</one-of>
<tag>$.value = "oskaras"</tag>
</item>
<item>
<one-of>
<item>paatrihkahs</item>
<item>patrikas</item>
</one-of>
<tag>$.value = "patrikas"</tag>
</item>
<item>

```

```

<one-of>
<item>kyuh</item>
<item>kju</item>
</one-of>
<tag>$.value = "kju"</tag>
</item>
<item>
<one-of>
<item>rihchahrdahs</item>
<item>ricardas</item>
</one-of>
<tag>$.value = "ricardas"</tag>
</item>
<item>
<one-of>
<item>saandrah</item>
<item>sandra</item>
</one-of>
<tag>$.value = "sandra"</tag>
</item>
<item>
<one-of>
<item>tehaodaorahs</item>
<item>teodoras</item>
</one-of>
<tag>$.value = "teodoras"</tag>
</item>
<item>
<one-of>
<item>uhlihyaonah</item>
<item>ulijona</item>
</one-of>
<tag>$.value = "ulijona"</tag>
</item>
<item>
<one-of>
<item>vahtsiys</item>
<item>vacys</item>
</one-of>
<tag>$.value = "vacys"</tag>
</item>
<item>
<one-of>
<item>vahshihngktaon</item>
<item>vasington</item>
</one-of>
<tag>$.value = "vasington"</tag>
</item>
<item>
<one-of>
<item>ihksahs</item>
<item>iksas</item>
</one-of>
<tag>$.value = "iksas"</tag>
</item>
<item>
<one-of>
<item>iygrehkahs</item>
<item>ygrekas</item>
</one-of>
<tag>$.value = "ygrekas"</tag>
</item>
<item>
<one-of>
<item>zahthaarihyuhs</item>
<item>zacharijus</item>
</one-of>
<tag>$.value = "zacharijus"</tag>
</item>
</rule>
</grammar>

```


2.2.2 Vardų pirminė gramatika papildyta penkių prasčiausiai atpažintų komandų naujomis transkripcijomis

```
<one-of>
<item>
<one-of>
<item>ahwstehyeh</item>
<item>austeja</item>
</one-of>
<tag>$_value = "austeja"</tag>
</item>
<item>
<one-of>
<item>baolehslaovahs</item>
<item>boleslovas</item>
</one-of>
<tag>$_value = "boleslovas"</tag>
</item>
<item>
<one-of>
<item>tsehtsihlihyeh</item>
<item>cecilija</item>
</one-of>
<tag>$_value = "cecilija"</tag>
</item>
<item>
<one-of>
<item>daonaatahs</item>
<item>donatas</item>
</one-of>
<tag>$_value = "donatas"</tag>
</item>
<item>
<one-of>
<item>ehiyamahntahs</item>
<item>eimantas</item>
<item>eihmantas</item>
<item>eigmantas</item>
<item>eihmanhtas</item>
<item>eihmanhtahss</item>
<item>aeihmanthahs</item>
</one-of>
<tag>$_value = "eimantas"</tag>
</item>
<item>
<one-of>
<item>fahwstah</item>
<item>fausta</item>
</one-of>
<tag>$_value = "fausta"</tag>
</item>
<item>
<one-of>
<item>graazhviydahs</item>
<item>grazvydas</item>
</one-of>
<tag>$_value = "grazvydas"</tag>
</item>
<item>
<one-of>
<item>haansahs</item>
<item>hansas</item>
</one-of>
<tag>$_value = "hansas"</tag>
</item>
<item>
<one-of>
<item>ihzahaokahs</item>
<item>izaokas</item>
</one-of>
<tag>$_value = "izaokas"</tag>
</item>
<item>
<one-of>
<item>yaonahs</item>
<item>jonas</item>
</one-of>
```

```
<tag>$_value = "jonas"</tag>
</item>
<item>
<one-of>
<item>kahraolihs</item>
<item>karolis</item>
</one-of>
<tag>$_value = "karolis"</tag>
</item>
<item>
<one-of>
<item>laaiymah</item>
<item>laima</item>
</one-of>
<tag>$_value = "laima"</tag>
</item>
<item>
<one-of>
<item>mahrtiynahs</item>
<item>martynas</item>
</one-of>
<tag>$_value = "martynas"</tag>
</item>
<item>
<one-of>
<item>naoyuhs</item>
<item>nojus</item>
</one-of>
<tag>$_value = "nojus"</tag>
</item>
<item>
<one-of>
<item>aoskahrahs</item>
<item>oskaras</item>
</one-of>
<tag>$_value = "oskaras"</tag>
</item>
<item>
<one-of>
<item>paatrihkahs</item>
<item>patrikas</item>
</one-of>
<tag>$_value = "patrikas"</tag>
</item>
<item>
<one-of>
<item>kyuh</item>
<item>kju</item>
<item>qhju</item>
<item>khju</item>
<item>kjhju</item>
<item>kjjhju</item>
<item>kjjhju</item>
</one-of>
<tag>$_value = "kju"</tag>
</item>
<item>
<one-of>
<item>rihchahrdahs</item>
<item>ricardas</item>
</one-of>
<tag>$_value = "ricardas"</tag>
</item>
<item>
<one-of>
<item>saandrah</item>
<item>sandra</item>
</one-of>
<tag>$_value = "sandra"</tag>
</item>
<item>
<one-of>
<item>tehaodaorahs</item>
<item>teodoras</item>
```

```

</one-of>
<tag>$. _value = "teodoras"</tag>
</item>
<item>
<one-of>
<item>uhlihyaonah</item>
<item>ulijona</item>
<item>ulihjonah</item>
<item>uhlihjonah</item>
<item>ullihjonah</item>
<item>ullihjona</item>
<item>hulihjonah</item>
</one-of>
<tag>$. _value = "ulijona"</tag>
</item>
<item>
<one-of>
<item>vahtsiys</item>
<item>vacys</item>
</one-of>
<tag>$. _value = "vacys"</tag>
</item>
<item>
<one-of>
<item>vahshihngktaon</item>
<item>vasington</item>
</one-of>
<tag>$. _value = "vasington"</tag>
</item>
<item>
<one-of>
<item>ihksahs</item>

```

```

<item>iksas</item>
</one-of>
<tag>$. _value = "iksas"</tag>
</item>
<item>
<one-of>
<item>ygrehkahs</item>
<item>ygrekas</item>
<item>igrekas</item>
<item>ygrehkak</item>
<item>yigrejkhkas</item>
<item>igrahkas</item>
<item>yigrahqas</item>
</one-of>
<tag>$. _value = "ygrekas"</tag>
</item>
<item>
<one-of>
<item>zahthaarihyuhs</item>
<item>zacharijus</item>
<item>zajarijus</item>
<item>zagharijus</item>
<item>zsagharhijus</item>
<item>zakarhijus</item>
<item>zthajarijus</item>
</one-of>
<tag>$. _value = "zacharijus"</tag>
</item>
</one-of>
</rule>
</grammar>

```

2.2.3 Vardu UPS gramatika

```

<grammar
xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/S
RGSEExtensions" xml:lang="es-US" tag-format="semantics-
ms/1.0" version="1.0" mode="voice"
xmlns="http://www.w3.org/2001/06/grammar"
sapi:alphabet="x-microsoft-ups">
<lexicon uri="..\Lexicons\Laikinas.lex"/>
<rule id="test" scope="public">
<one-of>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="A W S T E X A;A U S T E X
A">austeja</token>
<tag>$. _value = "austeja"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="A G W E S T I I E;A W S T E I E;A B S T
E I E;A G S T E I E;A G W S T E I E">ahwstehyeh</token>
<tag>$. _value = "ahwstehyeh"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B O L E S L O B A S">boleslovas</token>
<tag>$. _value = "boleslovas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B A O L E A S L O B A S;B A O L E W S
L O B A S;B A O L E X S L O B A S;B A O L E S L O B A
S">baolehslovas</token>
<tag>$. _value = "baolehslovas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="S E S I L I X A;S E S I L J X
A">cecilija</token>
<tag>$. _value = "cecilija"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>

```

```

<token sapi:pron="T S E T S J A L I I E;T S E T S J W L I I
E;T S E T S J L I I E;T S E S J L J I E;T S E T S J L J I
E">tsehtsihlihyeh</token>
<tag>$. _value = "tsehtsihlihyeh"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="D O N A T A S">donatas</token>
<tag>$. _value = "donatas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="D A O N A T A S;D E J O N A A T A
S">daonaatahs</token>
<tag>$. _value = "daonaatahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="E J M A N T A S;E I M A N T A S;E D X I
M A N T A S;O I M A N T A S;W I M A N T A
S">eimantas</token>
<tag>$. _value = "eimantas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?><token sapi:pron="E I
L J M A X N T A S;W I L J M A N T A S;E D X I L J M A N T
A S;O I L J M A N T A S;E I L J M A N T A
S">ehiyahntahs</token>
<tag>$. _value = "ehiyahntahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="F A W S T A">fausta</token>
<tag>$. _value = "fausta"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="F A W S T A;F A B S T A;F A G S T A;F
A G W E S T A;F A G W S T A">fahwstah</token>
<tag>$. _value = "fahwstah"</tag>
</item>

```

</item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="G DX A S B I D A S;G DX A S B J D A S">grazvydas</token>
<tag>\$.value = "grazvydas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="G DX A A S W B I D A S;G DX A A S A B I A D A S;G DX A A S A B I D A S;G DX A A S A B J A D A S;G DX A A S A B J D A S;G DX A A S W B J A D A S;G DX A A S W B J D A S;G DX A A S W B I A D A S">graazhviydahs</token>
<tag>\$.value = "graazhviydahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="X A N S A S;A N S A S">hansas</token>
<tag>\$.value = "hansas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="A A N S A S;X A A N S A S;A A N S A W S">haansahs</token>
<tag>\$.value = "haansahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="I S A O K A S;I S A K A S">izaokas</token>
<tag>\$.value = "izaokas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="I S A X A O K A S;I S A W A O K A S;I A S A A O K A S;I W S A A O K A S;I S A A O K A S">ihzahaokahs</token>
<tag>\$.value = "ihzahaokahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="X O N A S">jonas</token>
<tag>\$.value = "jonas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="CJ A O N A S;CJ A O N A S;CJ A O N A W S">yaonahs</token>
<tag>\$.value = "yaonahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="K A DX O L I S;K A DX O L J S">karolis</token>
<tag>\$.value = "karolis"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="K A DX A O L I S">kahraolihs</token>
<tag>\$.value = "kahraolihs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="L A J M A;L A I M A">laima</token>
<tag>\$.value = "laima"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="L A E M A;L A E I M A;L A E C J M A;L A E J M A;L A E A M A">laaiymah</token>
<tag>\$.value = "laaiymah"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="M A DX T I N A S;M A DX T A J N A S;M A DX T J N A S">martynas</token>
<tag>\$.value = "martynas"</tag>
</item>
<item>

<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="M A RR T I J N A S;M A X DX T I J N A S;M A DX T I I N A S;M A RR T I I N A S;M A DX T I J N A S">mahrtiynahs</token>
<tag>\$.value = "mahrtiynahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="N O X U S;N O X W S">nojus</token>
<tag>\$.value = "nojus"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="N A O C J U S;N A O L J U S;N A O C J U W S;N A O C J W S;N A O C J U A S">naoyuhs</token>
<tag>\$.value = "naoyuhs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="O S K A DX A S">oskaras</token>
<tag>\$.value = "oskaras"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="A O S K A DX A S">aoskahrahs</token>
<tag>\$.value = "aoskahrahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="P A T DX J K A S;P A T DX I K A S">patrikas</token>
<tag>\$.value = "patrikas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="P A A T DX I A K A S;P A A T DX I W K A S;P A A T DX J W K A S;P A A T DX J A K A S;P A A T DX I K A S">paatrihkahs</token>
<tag>\$.value = "paatrihkahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="K X U;K X W;K A X U">kju</token>
<tag>\$.value = "kju"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="K I U;K J U;K L J U;K C J U;K A L J U">kyuh</token>
<tag>\$.value = "kyuh"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="RR I K A DX D A S;DX I K A DX D A S">ricardas</token>
<tag>\$.value = "ricardas"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="RR J A CH A DX D A S;RR J CH A DX D A S;RR J W K A DX D A S;RR J A K A DX D A S;RR J W CH A DX D A S">rihchahrdahs</token>
<tag>\$.value = "rihchahrdahs"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="S A N D DX A">sandra</token>
<tag>\$.value = "sandra"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="S A A N D DX A;E S E A A N D DX A;S A A N D RR A;S A A N D DX A W;S A A N D DX A A">saandrah</token>
<tag>\$.value = "saandrah"</tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="T E O D O DX A S">teodoras</token>

```

<tag>$_value = "teodoras" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="T E A O D A O DX A
S">tehaodaorahs</token>
<tag>$_value = "tehaodaorahs" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="U L I X O N A">ulijona</token>
<tag>$_value = "ulijona" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="U L J I A O N A;U L I A L J A O N A;U L
I A C J A O N A;U L I W L J A O N A;U L I W C J A O N A;U
L I I A O N A">uhlihyaonah</token>
<tag>$_value = "uhlihyaonah" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B A S I S;B A K I S;B A K J
S">vacys</token>
<tag>$_value = "vacys" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B A T S I I S;B A T S J J S;B A T S J E J
S;B A T S I J S;B A T S J I S">vahtsiys</token>
<tag>$_value = "vahtsiys" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B A S I N T O N;B A S I N G T O
N">vasington</token>
<tag>$_value = "vasington" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="B A S I N X K T A O N;B A C H I N G K T
A O N;B A S I N K T A O N;B A S I N C H K T A O N;B A S
I N G K T A O N">vahshihngktaon</token>

```

```

<tag>$_value = "vahshihngktaon" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="I K S A S">iksas</token>
<tag>$_value = "iksas" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="I S A S;I K S A W S;I S A W S;I K E S E A
S;I K S A S">ihksahs</token>
<tag>$_value = "ihksahs" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="C J G DX E K A S;I G DX E K A S;A G DX
E K A S;G DX E K A S;C J G DX I K A S">ygrekas</token>
<tag>$_value = "ygrekas" </tag>
</item><item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="I J G DX E K A S;I L J G DX E K A S;I G
DX E K A S;I A G DX E K A S;I I G DX E K A
S">iygrehkahs</token>
<tag>$_value = "iygrehkahs" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="S A C H A DX J X U S;S A C H A DX I X
U S;S A C H A DX J X W S">zacharijus</token>
<tag>$_value = "zacharijus" </tag>
</item>
<item>
<?MS_Grammar_Editor GroupWrap?>
<token sapi:pron="S A W T A A DX J I U S;S A T A A DX J
A C J U S;S A T A A DX J W C J U S;S A T A A DX J C J W
S;S A T A A DX J I U S">zahthaarihyuhs</token>
<tag>$_value = "zahthaarihyuhs" </tag>
</item>
</one-of>
</rule>
</grammar>

```

3 PRIEDAS. UPS alfabeto fonemų lentelės ispanų kalbai

2.2.1 lentelė. UPS alfabeto balsių lentelė ispanų kalbai

Balsiai		
UPS Label	Example Word	UPS transcription of example word
A	vaca	B A K A
A J	aire	A J D X E
A W	causa	K A W S A
E	pero	P E D X O
E J	Abeijón	A B E J X O N
E W	Europa	E W D X O P A
I	pico	P I K O
O	toro	T O D X O
O J	sois	S O J S
O W	Souto	S O W T O
U	duro	D U D X O

2.2.2 lentelė. UPS alfabeto priebalsių lentelė ispanų kalbai

Priebalsiai		
UPS Label	Example Word	UPS transcription of example word
B	vino; cabra	B I N O ; K A B D X A
CH	mucho	M U C H O
CJ	yegüa	C J E G W A
D	nada	N A D A
DX	puro	P U D X O
F	fácil	F A S I L
G	gata	G A T A
J	viudo	B J U D O
K	casa	K A S A
L	largo	L A D X G O
LJ	cballo	K A B A L J O
M	masa	M A S A
N	noche	N O C H E
NJ	año	A N J O
P	padre	P A D D X E
RR	torre	T O R R E
S	cielo; isla	S J E L O ; I S L A
T	antes	A N T E S
W	suave	S W A B E
X	mujer	M U X E D X

2.2.3 lentelė. UPS alfabeto prožodijos lentelė ispanų kalbai

Prožodija			
UPS Label	Description	Example Word	UPS transcription of example word
S1	primary stress	señor	S E I . S1 NJ O DX
S2	secondary stress	señorita	S2 S E I . NJ O . S1 DX I . T A
.	syllable break	Isla	S1 I S . L A
_!	intonation - exclamation		
_&	intonation - continue		
_,	intonation - incomplection rise		
_.	intonation – fall		
_?	intonation - question rise		
_s	silence		

4 PRIEDAS. Vardų atpažinimo tikslumo priklausomybė nuo būsenų ir Gauso mišinių skaičiaus

3.4.4 lentelė. Atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus

Atpažinimo tikslumo priklausomybė nuo būsenų skaičius, %				
Vardas	Būsenų skaičius atitinka raidžių sk.			
	+0 busenų	+2 busenos	+4 busenos	+6 busenos
Austėja	97,5	97,5	97,5	97,5
Boleslovas	100	100	100	97,5
Cecilija	100	100	100	100
Donatas	52,5	100	47,5	52,5
Eimantas	100	100	100	97,5
Fausta	42,5	97,5	100	100
Gražvydas	72,5	72,5	52,5	97,5
Hansas	72,5	100	100	100
Izaokas	100	100	100	15
Jonas	30	100	100	100
Karolis	87,5	97,5	100	10
Laima	22,5	80	62,5	100
Martynas	92,5	100	100	100
Nojus	37,5	82,5	100	100
Oskaras	97,5	82,5	100	27,5
Patrikas	95	90	65	95
Kju	0	5	95	95
Ričardas	45	67,5	15	62,5
Sandra	30	100	100	100
Teodoras	100	100	100	100
Ulijona	100	100	100	80
Vacys	40	100	97,5	100
Wašington	52,5	87,5	62,5	100
Iksas	0	20	85	85
Ygrekas	100	100	5	10
Zacharijus	100	100	100	100
Vidurkis, %	67,9±13,15	87,7±9,29	84±10,58	81,6±11,98

3.4.5 lentelė. Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus

Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, %					
Vardas	Būsenų skaičius atitinka raidžių sk. + 2				
	2 mišiniai	3 mišiniai	6 mišiniai	10 mišiniai	16 mišiniai
Austėja	97,5	97,5	97,5	100	100
Boleslovas	100	100	100	100	100
Cecilija	100	100	100	100	100
Donatas	100	97,5	100	100	100
Eimantas	100	100	100	100	100
Fausta	97,5	100	95	100	100
Gražvydas	100	100	100	100	100
Hansas	100	100	100	100	100
Izaokas	100	100	100	100	100
Jonas	100	100	52,5	100	100
Karolis	100	97,5	100	100	100
Laima	100	100	97,5	97,5	95
Martynas	100	100	100	100	100
Nojus	100	100	100	100	100
Oskaras	100	100	100	100	100
Patrikas	100	100	100	100	100
Kju	95	95	90	95	90
Ričardas	95	50	67,5	100	100
Sandra	100	100	100	100	100
Teodoras	100	100	100	100	100
Ulijona	100	100	100	100	100
Vacys	100	82,5	5	100	100
Wašington	100	12,5	100	100	100
Iksas	100	100	40	100	100
Ygrekas	100	100	100	100	100
Zacharijus	100	100	100	100	100
Vidurkis	99,4±0,56	93,5±6,46	90,1±8,96	99,7±0,29	99,4±0,54

3.4.6 lentelė. Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus

Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, %					
Vardas	Būsenų skaičius atitinka raidžių sk. + 4				
	2 mišiniai	3 mišiniai	6 mišiniai	10 mišiniai	16 mišiniai
Austėja	97,5	97,5	100	100	100
Boleslovas	100	100	100	100	100
Cecilija	100	100	100	100	100
Donatas	100	100	100	100	100
Eimantas	100	100	100	100	100
Fausta	100	100	100	100	100
Gražvydas	100	100	100	100	100
Hansas	100	100	100	100	100
Izaokas	100	100	100	100	100
Jonas	95	100	100	100	100
Karolis	100	97,5	100	100	100
Laima	100	100	100	100	100
Martynas	100	100	100	100	100
Nojus	100	100	100	100	100
Oskaras	100	100	100	100	100
Patrikas	100	100	100	100	100
Kju	95	95	90	95	95
Ričardas	67,5	92,5	100	100	100
Sandra	100	100	100	100	100
Teodoras	100	100	100	100	100
Ulijona	100	100	100	100	100
Vacys	97,5	80	92,5	100	100
Wašington	100	100	100	100	100
Iksas	92,5	100	100	100	100
Ygrekas	100	100	100	100	100
Zacharijus	100	100	100	100	100
Vidurkis	97,8±2,15	98,5±1,46	99,3±0,16	99,8±0,17	99,8±0,17

5 PRIEDAS. 2.4 Hidden Markov Model Toolkit detalus įsisavinimas

2.4.1 HTK programų rinkinio panaudojimas taikant žodžių modelių metodą šnekos atpažinimui

Šiame skyriuje aprašomas HTK programinių įrankių naudojimas balso komandų atpažinimo modeliavimui taikant žodžių modelių metodą, kurio taikymą sudaro 7 nuoseklūs etapai.

1 ETAPAS - gramatikos failo kūrimas. Sukuriamas, HTK programų paketui reikalingas, gramatikų failas „*gram.dikt*“. Jame aprašomos norimos atpažinti balso komandos ir jų tvarka atpažinimo procese. Šiame tyrime naudojamų lietuviškų vardų ir skaičių pavadinimų komandų aprašymas „*gram.dikt*“ faile atrodo taip:

Vardų komandų „*gram.dikt*“:

```
$word=austeja/boleslovas/cecilija/donatas/eimantas/fausta/grazvydas/hansas/izaokas/jonas/karolis/laima/martynas/nojus/oskaras/patrikas/qju/ricardas/sandra/teodoras/ulijona/vacys/washington/xsas/ygrekas/zacharijus;  
($word)
```

Taip pat „*gram.dikt*“ faile gramatikos gali būti pateikiamos ir kitaip. Jeigu vieno ištarimo metu norima pasakyti iškart du žodžius iš pateikto sąrašo - „Austėja, Boleslovas“, gramatikos failo turinys atrodo taip:

```
$word=austeja/boleslovas/cecilija/donatas/eimantas/fausta/grazvydas/hansas/izaokas/jonas/karolis/laima/martynas/nojus/oskaras/patrikas/qju/ricardas/sandra/teodoras/ulijona/vacys/washington/xsas/ygrekas/zacharijus;  
$word $word
```

Dar „*gram.dikt*“ faile gramatikos gali būti pateikiamos ir kitaip. Jeigu vieno ištarimo metu norima pasakyti iškart du žodžius iš pateikto sąrašo ir juos sujungti jungtuku - „Austėja ir Boleslovas“, gramatikos failo turinys atrodo taip :

```
$word=austeja/boleslovas/cecilija/donatas/eimantas/fausta/grazvydas/hansas/izaokas/jonas/karolis/laima/martynas/nojus/oskaras/patrikas/qju/ricardas/sandra/teodoras/ulijona/vacys/washington/xsas/ygrekas/zacharijus;  
$word1 = ir  
$word $word $word
```

Šiame darbe balso komandų atpažinimo tyrimuose naudojamas pirmasis gramatikos variantas ir lietuviškų skaičių ir vardų garsynams.

Skaičių pavadinimų komandų „gram.dikt“:

\$word=nulis/vienas/du/trys/keturi/penki/sesi/septyni/astuoni/devyni;
(*\$word*)

Faile „*gram.dikt*“ komandos pavadinimai yra atskiriami specialiu simboliu (/). Taip yra nurodoma, kad bus tariami atskiri žodžiai: *nulis, vienas, du, trys, eimantas, ulijona, zacharijus ir t.t.*

2 ETAPAS - žodžių modelių sudarymui reikalingų failų paruošimas. HTK programinių įrankių pagalba automatiškai sukuriamas tekstinis failas „*wordnet.txt*“. Šiam failui sukurti naudojama programa „*Hparse.exe*“, kurios *batch* faile įrašyta komandinė eilutė: *HParse.exe gram.dict wordnet.txt*. Pirmoji komanda eilutėje - „*Hparse.exe*“ nurodo, kad bus naudojama ši programa, taip pat, bus naudojamas failas „*gram.dict*“ – prieš tai jau sukurtas žodžių sąrašas, o „*wordnet.txt*“ - programos automatiškai sukuriamas failas.

3 ETAPAS - požymių kūrimas. Turint „*gram.dikt*“ ir „*wordnet.txt*“ failus toliau sukuriamas, požymiams sukurti ir apskaičiuoti reikalingas failas - „*failai.scf*“. Tai failas, kuriame bus nurodomos kompiuterio direktorijos, kur bus sukuriami ir talpinami požymių failai. Žemiau pateikiama lietuviškų vardų ir skaičių garsynams naudojamų „*failai.scf*“ failų turinių dalys:

Vardų garsyno naudojamo failo „*failai.scf*“ turinio dalis:

C:\HTK\HTK_VARDAI\irasai2\8A00.wav C:\HTK\HTK_VARDAI\irasai2\8A00.mfc
C:\HTK\HTK_VARDAI\irasai2\8A01.wav C:\HTK\HTK_VARDAI\irasai2\8A01.mfc
C:\HTK\HTK_VARDAI\irasai2\8A02.wav C:\HTK\HTK_VARDAI\irasai2\8A02.mfc
.....
C:\HTK\HTK_VARDAI\irasai2\8A17.wav C:\HTK\HTK_VARDAI\irasai2\8A17.mfc
C:\HTK\HTK_VARDAI\irasai2\8A18.wav C:\HTK\HTK_VARDAI\irasai2\8A18.mfc
C:\HTK\HTK_VARDAI\irasai2\8A19.wav C:\HTK\HTK_VARDAI\irasai2\8A19.mfc

Skaičių garsyno naudojamo failo „*failai.scf*“ turinio dalis:

C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA000.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA000.mfc
C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA001.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA001.mfc
C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA002.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA002.mfc
.....
C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA017.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA017.mfc
C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA018.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA018.mfc
C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA019.wav C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA019.mfc

Pavyzdžiuose matoma, kad kiekviena komandinė eilutė susideda iš direktorijos, kurioje yra garsyno diktorių balso įrašas (*C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA019.wav*) ir direktorijos, kurioje bus sukuriamas to įrašo *.mfc požymių failas (*C:\HTK\HTK_SKAICIAI\irasai\FAGNGRA019.mfc*). Kiekvienas požymių failas yra sukuriamas pagal to paties diktoriaus, tą patį, garsinį failą. Požymių failai yra reikalingi, nes atpažinimo sistema naudoja ne pačius *.wav formato garso įrašus, o tik jų požymius. Sukurti požymių failai buvo patalpinti atskirame kataloge „pozymiai“. Požymių failams sukurti naudojama programa „Hcopy.exe“. Jos vykdymui reikia turėti paruošus tokius failus - *failai.scf*“, jau minėtas failas kuriame nurodomos direktorijos ir „CONFIG“, kuris yra informacinis failas su pateikta informacija apie požymių skaičiavimo būdus. „CONFIG“ failo turinys atrodo taip:

```
SOURCEKIND=WAVEFORM
SOURCEFORMAT=WAVE
TARGETKIND=MFCC_D_A_E
SAVEWITHCRC=F
SOURCERATE=625
TARGETRATE=100000.0
WINDOWSIZE=250000.0
USEHAMMING=T
PREEMCOEF=0.96
NUMCEPS=12
NUMCHANS=20
```

4 ETAPAS – modelių failų parengimas. Kai jau turime požymius, tolimesniame etape yra parengiami ir pildomi modelių failai. Šie modelių failai yra reikalingi atlikti šnekamosios kalbos atpažinimo sistemos apmokymą. Modelių failų yra sukuriama tiek kiek yra komandų - kiekvienam komandos pavadinimui po vieną:

Dešimties skaičių modelių failai:

```
hmm_nulis
hmm_vienas
hmm_du
hmm_trys
hmm_keturi
hmm_penki
hmm_sesi
hmm_septyni
hmm_astuoni
hmm_devyni
```

Dvidešimt šešių vardų modelių failai:

```
hmm_austeja      hmm_nojus
hmm_boleslovas  hmm_oskartas
hmm_cecilija    hmm_patrikas
hmm_donatas     hmm_qju
hmm_eimantas    hmm_ricardas
hmm_fausta      hmm_sandra
hmm_grazvydas   hmm_teodoras
hmm_hansas      hmm_ulijona
hmm_izaokas     hmm_vacys
hmm_jonas       hmm_washington
hmm_karolis     hmm_xsas
hmm_laima       hmm_ygrekas
hmm_martynas    hmm_zacharijus
```

Šių failų pavadinimai turi būti tikslūs, nes juose yra griežtos struktūros informacija, apie tam tikrą komandos pavadinimą. Kiekviename modelių faile nurodoma kiek būsenų ar Gauso mišinių bus naudojama modeliuojant atitinkamą žodį. Aprašomi vidurkių bei dispersijų vektoriai, perėjimų tikimybių reikšmės ir kiti parametrai. 2.4.1 paveiksle pateikiamas skaičiaus pavadinimo „du“, o 2.4.2 paveiksle pateikiamas vardo „Laima“ tokio tipo modelių failo, turinys.

```

->D
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
-h "hmm_du"
<BEGINHMM>
<NUMSTATES> 4
<STATE> 2
<NUMMIXES> 2
<MIXTURE> 1 0.5
<MEAN> 39
-1.693078e+001 9.293804e-001 -1.197502e+000 -3.922675e+000 -1.842438e+000 -3.959891e+000 -7.416571e+000 -4.536137e+000 -2.235704e+000 -2.050065e-002 -
3.593387e+000 -2.249446e+000 6.605968e-001 2.296624e-001 -1.080772e+000 -6.491815e-001 -1.526795e+000 -3.981741e-001 -5.044248e-001 -1.312936e+000 -
6.110157e-001 2.748346e-001 9.368593e-001 1.882942e-001 4.985206e-001 6.018994e-002 1.613519e-001 5.653471e-002 -1.649433e-001 -1.098936e-001 -8.851676e-002
1.005222e-001 -1.934532e-002 -1.115357e-001 2.065790e-002 5.478030e-002 2.487857e-002 -4.674675e-002 2.060280e-004
<VARIANCE> 39
1.933895e+001 3.414952e+001 2.321033e+001 5.308133e+001 2.499531e+001 3.532112e+001 6.246143e+001 3.448954e+001 2.230908e+001 2.813749e+001 2.164686e+001
1.607351e+001 7.356837e-002 3.288550e+000 3.387602e+000 2.309940e+000 3.839294e+000 3.213022e+000 3.743657e+000 5.309355e+000 4.316780e+000 2.804044e+000
2.490945e+000 2.125204e+000 1.650401e+000 4.077548e-003 5.389854e-001 5.965810e-001 3.183646e-001 6.952607e-001 5.822270e-001 7.959337e-001 1.156594e+000
7.497113e-001 4.862667e-001 4.870841e-001 3.661313e-001 3.270547e-001 9.162700e-004
<GCONST> 1.032259e+002
<MIXTURE> 2 0.5
<MEAN> 39
-1.328975e+001 4.723204e-001 -6.764323e+000 -7.637876e+000 -2.986039e+000 -1.564041e+000 -7.665658e+000 -6.564466e+000 -3.390937e+000 4.021169e-001 -
3.057339e+000 -2.548885e+000 6.698649e-001 -2.311842e-001 1.190231e-001 -4.009480e-002 4.113052e-001 2.384509e-001 3.043577e-001 5.570447e-001 2.792146e-001
-8.294746e-002 -2.238914e-001 9.495310e-002 -7.362264e-002 -2.541608e-002 -8.851899e-002 -2.963891e-002 5.919562e-002 4.478659e-002 2.214781e-002 -
3.288339e-002 2.115004e-002 5.820182e-002 7.897646e-003 9.882805e-003 6.612533e-003 3.198449e-002 -9.731298e-004
<VARIANCE> 39
1.012305e+001 3.258541e+001 3.868000e+001 7.418177e+001 2.561711e+001 3.664819e+001 5.534353e+001 3.326296e+001 3.093116e+001 4.136096e+001 3.400393e+001
2.397844e+001 6.802116e-002 4.409781e-001 5.272187e-001 8.486349e-001 1.526153e+000 1.109780e+000 1.372538e+000 1.394547e+000 1.431562e+000 1.401681e+000
1.494637e+000 1.247534e+000 9.777467e-001 7.278427e-004 5.301289e-002 1.065826e-001 1.259250e-001 1.956495e-001 1.902312e-001 2.091872e-001 2.312097e-001
2.437838e-001 2.247177e-001 2.481797e-001 2.065370e-001 1.651506e-001 8.858099e-005
<GCONST> 7.401907e+001
<STATE> 3
<NUMMIXES> 2
<MIXTURE> 1 0.5
<MEAN> 39
-2.271428e+001 -1.213378e+000 -6.510975e+000 -1.314898e+000 -2.566651e+000 -7.382380e-001 -1.036461e+000 -2.431119e+000 -2.255749e+000 -2.085052e+000 -
2.421253e+000 -1.658399e+000 2.414010e-001 -8.200081e-002 -1.336100e-002 1.963885e-002 3.487277e-002 8.236416e-003 8.186469e-003 2.748451e-002 2.127678e-002
-3.374891e-003 -4.504006e-002 -1.767734e-002 2.316262e-002 -7.696571e-004 1.153937e-002 1.646368e-003 1.671143e-003 1.190071e-002 2.906975e-002 3.457592e-002
3.625388e-002 4.320081e-002 5.375963e-002 6.349387e-002 5.591448e-002 3.462375e-002 3.097488e-004
<VARIANCE> 39
1.727345e+000 2.898428e+000 3.864456e+000 4.778569e+000 5.944373e+000 7.714757e+000 8.221948e+000 1.073362e+001 9.812807e+000 8.982080e+000 8.492149e+000
7.200127e+000 1.133422e-002 7.250682e-002 1.501579e-001 2.569709e-001 4.013907e-001 5.218737e-001 6.414114e-001 7.378351e-001 8.119459e-001 8.816100e-001
8.718670e-001 8.218667e-001 6.733612e-001 1.349819e-005 1.226704e-002 2.761240e-002 4.738049e-002 7.537142e-002 1.011376e-001 1.244754e-001 1.386513e-001
1.527112e-001 1.654747e-001 1.608668e-001 1.353774e-001 1.285447e-001 2.563321e-006
<GCONST> 2.587003e+001
<MIXTURE> 2 0.5
<MEAN> 39
-2.200596e+001 3.400850e-002 -4.770247e+000 4.670423e-001 -4.079904e-001 1.287861e+000 5.626875e-001 -1.477438e-001 -6.852416e-002 9.556989e-001 2.322826e-
001 -1.770133e-001 2.773655e-001 -2.281878e-001 -5.590088e-002 4.790937e-002 8.600929e-002 -4.247938e-002 2.150088e-003 9.337831e-002 1.098972e-001

```

2.4.1 pav. Balso komandos „hmm_du“ modelių failo turinys

```

"0
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
-h "hmm_laima"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
-1.286633e+001 -1.000192e+001 -5.730053e+000 -6.778081e+000 -5.627978e+000 4.508104e+001 -3.644324e+000 -4.040222e+000 -2.983810e+000 -1.880689e+000 -3.296680e+000
<VARIANCE> 39
2.301101e+001 6.835938e+001 6.560416e+001 5.488258e+001 6.053754e+001 5.060648e+001 4.746502e+001 4.825924e+001 3.618487e+001 2.879947e+001 2.363195e+001 1.516160e
<GCONST> 8.995935e+001
<STATE> 3
<MEAN> 39
-1.918106e+001 -6.541372e+000 -7.989298e+000 -1.814019e+000 2.986596e+000 2.384126e+000 -2.388093e+000 -5.185410e-001 1.576877e+000 -2.188964e+000 -3.267574e+000 -
<VARIANCE> 39
2.864562e+001 1.125613e+001 2.923951e+001 1.602693e+001 1.452585e+001 2.181686e+001 4.593960e+001 2.146532e+001 3.274050e+001 3.594651e+001 2.120875e+001 1.224947e
<GCONST> 7.732952e+001
<STATE> 4
<MEAN> 39
-2.793663e+001 -4.837818e+000 -3.172441e+000 -3.916919e-001 -9.841992e-001 -5.103589e-001 5.859421e-001 -5.175027e-001 -1.114067e+000 1.762865e+000 -1.593701e+000
<VARIANCE> 39
1.960183e+000 2.022394e+000 4.462610e+000 4.677469e+000 6.316516e+000 7.011181e+000 7.180126e+000 9.351744e+000 8.139812e+000 8.368270e+000 7.147951e+000 6.348445e
<GCONST> 2.500908e+001
<TRANSP> 5
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.831133e-001 1.688674e-002 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.250656e-001 7.493439e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 9.675385e-001 3.246157e-002
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

2.4.2 pav. Balso komandos „hmm_laima“ modelių failo turinys

Modelių faile esančią informaciją galima išskirti į dvi dalis: bendrą informaciją apie modelių failą ir modeliavimo duomenis. Kiekviename šių failų reikia nurodyti, kiek būsenų bus naudojama žodžiui modeliuoti, kiek požymių bus naudojama požymių vektoriuje,

nurodomi pradiniai vidurkių ir dispersijų vektoriai bei perėjimų tikimybių reikšmės. Bendroji informacija apie modelių failą pateikiama pirmose failo eilutėse.

Modelių failo informacija yra aprašoma tokiose eilutėse:

~o – pradžios simbolis;

<STREAMINFO> 1 39 – skaičius 39 nurodo, kad bus naudojamas 39 požymių rinkinys, o skaičius 1 – kad jie bus išdėstyti vienoje eilutėje;

<VECSIZE> 39<NULLD><MFCC_E_D_A> – šioje eilutėje nurodomas vektoriaus dydis - tie patys 39 požymiai, MFCC_E_D_A reiškia, kad bus naudojami MFCC požymis ir E – energijos, D – delta ir A – delta delta požymio koeficientai;

*~h "hmm_**"* – tai modelių failo pavadinimas. Norint atlikti tikslų pasirinktosios komandos modeliavimą šis pavadinimas turi sutapti su pagrindiniu komandos modelių failo pavadinimu;

<BEGINHMM> – modelio pradžios žyma;

<NUMSTATES> - nurodomas modelių faile naudojamas būsenų skaičius. Buvo atliekami atpažinimo

tikslumo tyrimai naudojant modelius su skirtingu būsenų skaičiumi. Būsenų skaičius buvo parenkamas atsižvelgiant į komandos raidžių skaičių ir pridėdant papildomą būsenų skaičių (+1, +2, ir t.t.). Pavyzdžiui, jeigu skaičių “du” sudaro dvi raidės ir atliekamas atpažinimo tikslumo tyrimas naudojant dešimt papildomų būsenų, tai „*<NUMSTATES>*“ eilutėje bus rašoma – 12 (2+10). Tyrimo rezultatai naudojant skirtingą komandos modelio būsenų skaičių aprašomi HHH skyriuje.

<STATE> 2 – modelis visada prasideda nuo antros būsenos;

<NUMMIXES> - nurodomas modelių daile naudojamų Gauso mišinių skaičius. Buvo atliekami atpažinimo tikslumo tyrimai naudojant modelius su skirtingu Gauso mišinių skaičiumi . Tyrimo rezultatai naudojant skirtingą komandos modelio Gauso mišinių skaičių aprašomi HHHK skyriuje;

<MIXTURE> - nurodomas pirmasis Gauso mišinys ir jo vertė;

<MEAN> 39 -1.942506e+001 -3.543667e+000 ... -1.191551e-003 – požymių vidurkiai, parenkami intuityviai, kurie modelių mokymo metu bus perskaičiuojami automatiškai;

<VARIANCE> 39 6.814474e+001 3.582195e+001 ... 2.968909e-004 – požymių dispersijos, parenkamos intuityviai, kurios apmokymo metu bus perskaičiuotos automatiškai;

<GCONST> 9.449938e+001 – būsenos stiprinimo koeficientas, parenkamas nuo 0 iki 1.

Tokie veiksmai atliekami su visomis būsenomis iki paskutiniosios. Tokiu būdu apskaičiuojamos tikimybės būti pirmoje, antroje, trečioje ir t.t. būsenose.

<TRANSP> - „transponentė“ - paskutinė būsena. Šioje būsenoje yra sudaroma tikimybių matrica, kuri atrodo taip:

```
<TRANSP> 4
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.261925e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.219534e-001 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
```

Prie nuorodos į transponentę esantis skaičius žymi paskutinę būseną. Šioje tikimybių matricoje, eilučių ir stulpelių skaičius turi sutapti. Tikimybės šioje matricoje juda į priekį - kiekvienoje eilutėje yra aprašoma tikimybė pereiti į sekančią būseną arba likti toje pačioje. Vykdamas paruoštų modelių apmokymą, šios tikimybių vertės yra automatiškai perskaičiuojamos. Požymių vidurkių (<MEAN>) bei dispersijų (<VARIANCE>) vertės taip pat yra automatiškai perskaičiuojamos modelių apmokymo metu;

<ENDHMM> modelio pabaigos žyma;

Susiredagavus vieną modelio failą, patogų jį taikyti kaip šabloną kitiems modeliams atitinkamai jį modifikuojant - keičiant pavadinimą, būsenų skaičių ir pridėdant arba pašalinant atitinkamas būsenas ir tikimybių matricos eilutes bei stulpelius, keičiant Gauso mišinių skaičių.

Atliekant tyrimus, naudojant skirtingą būsenų ir Gauso mišinių kiekį, šiais modelių failais buvo įvairiai varijuojama. Atitinkamai paruošus vieną modelių failą, jį galima pritaikyti skirtingose situacijose (atliekant testavimus su skirtingu skaičiumi būsenų ir pan.), pašalinant

arba pridėdant būsėnų ar Gauso mišinų kiekį, tikimybių matricos stulpėlių bei eilėčių kiekį, bei pakeičiant modelių failo pavadinimą.

Paruošus modelių apmokymui reikalingus duomenis, prieš apmokymą, reikia susikurti apmokymui reikalingus papildomus *script* (*.scp) failus, kuriems reikia nurodyti direktorijas, kuriose yra saugomi požymių (*.mfc) failai. Kiekvienos komandos pavadinimu yra sukuriamas atskiras *script* failas, pavadinimu „trans_*.scp“. Tokie failai buvo sukurti visoms atpažinimui reikalingoms komandoms:

Dešimties skaičių modelių failai:

trans_nulis.scp
trans_vienas.scp
trans_du.scp
trans_trys.scp
trans_keturi.scp
trans_penki.scp
trans_sesi.scp
trans_septyni.scp
trans_astuoni.scp
trans_devyni.scp

Dvidešimt šešių vardų modelių failai:

trans_austeja.scp	trans_nojus.scp
trans_boleslovas.scp	trans_oskaras.scp
trans_cecilija.scp	trans_patrikas.scp
trans_donatas.scp	trans_qju.scp
trans_eimantas.scp	trans_ricardas.scp
trans_fausta.scp	trans_sandra.scp
trans_grazvydas.scp	trans_teodoras.scp
trans_hansas.scp	trans_ulijona.scp
trans_hansas.scp	trans_vacys.scp
trans_izaokas.scp	trans_washington.scp
trans_karolis.scp	trans_xsas.scp
trans_laima.scp	trans_ygrekas.scp
trans_martynas.scp	trans_zacharijus.scp

Skaičių komandų failo - „trans_du.scp“ turinio fragmentas atrodo taip:

C:\HTK\HTK_VARDAI\pozymiai\8A00.mfc
C:\HTK\HTK_VARDAI\pozymiai\8A01.mfc
C:\HTK\HTK_VARDAI\pozymiai\8A02.mfc
.....
C:\HTK\HTK_VARDAI\pozymiai\8A19.mfc

Vardų komandų failo - „trans_laima.scp“ turinio fragmentas atrodo taip:

C:\HTK\HTK_SKAICIAI\pozymiai\FAGNGRA200.mfc
C:\HTK\HTK_SKAICIAI\pozymiai\FAGNGRA201.mfc
C:\HTK\HTK_SKAICIAI\pozymiai\FAGNGRA202.mfc
.....
C:\HTK\HTK_SKAICIAI\pozymiai\FAGNGRA219.mfc

Kiekvienas iš šių failų susideda tik iš jo pavadinime parašytų komandų pavadinimų požymių. Pavyzdžiui „trans_šeši.scp“ susideda tik iš skaičiaus pavadinimo „šeši“, visų apmokymui naudojamų požymių.

5 ETAPAS – *modelių apmokymas*. Paruošus šiuos failus atliekamas žodžių modelių apmokymas, naudojant „*HRest.exe*“ programą. Kuri naudoja jau paruoštus „*trans_*.scp*“ script failus, kuriuose išvardinti žodžių junginio požymių failai. Pasinaudoję „*HRest.exe*“ programa gauname apmokytą modelio failą, kurį galima naudoti atpažinimo metu. Taip apmokomi visų žodinių komandų modeliai.

„*HRest.bat*“ faile užrašomos tokios komandinės eilutės atitinkamai komandai atskirai:
Skaičių garsynui :

```
HRest.exe -T 1 -S trans_nulis.scp hmm_nulis>>kk1
```

Vardų garsynui:

```
HRest.exe -T 1 -S trans_karolis.scp hmm_karolis>>kk1
```

Kiekvienam *script* (*trans_*.scp*) failui yra priskiriamas modelių failas „*hmm_**“, ir yra nurodoma sukurti atitinkamą, jau apmokytą failą „*kk**“. Pavyzdžiui, programa „*HRest.exe*“ pasinaudojusi *trans_nulis.scp* *hmm_nulis* sukurs skaičiaus pavadinimui „*nulis*“ apmokytą failą - „*kk1*“ ir t.t.. Atlikus žodžių modelių apmokymą, automatiškai sukuriama tiek „*kk**“ failų kiek yra komandų. Šie failai reikalingi norint atlikti balso komandų atpažinimo tikslumo testavimą.

6 ETAPAS – *duomenų paruošimas testavimui*. Atlikus modelių apmokimą reikia atitinkamus failus paruošti ir apmokytų modelių taikymui balso komandoms atpažinti testavimui. Tam reikia susikurti papildomus *script* failus, kurie bus reikalingi kuriant naujus požymius testavimui. Visų pirma, sukuriamas *script* failas pavadinimu „*test.scp*“, kurio turinys atrodo taip pat, kaip pačioje pradžioje apmokymui kurto failo „*failai.scp*“ turinys, skirtumas tik tas, kad šiuo atveju požymių failus reikia sukurti, testavimui paskirtiems (kitiems nei naudojamiems apmokyme) diktorių, garso įrašams. Sukūrus šį failą ir suredagavus direktorijas, yra dar kartą naudojama programa „*Hcopy.exe*“, kuri sukuria naujus, testavimui skirtus požymių failus. Sukurti požymių failų pavadinimai yra surašomi į dar vieną *script* failą „*test_mfc.scp*“. Atliekamas tas pats procesas kaip ir aprašant „*trans_*.scp*“ failus, skirtumas tik tas, kad dabar visi požymių failų pavadinimai yra surašomi į vieną *script* failą.

Toliau sukuriamas failas „*hmmlist*“, kuriame surašomi visi modelių failų pavadinimai:

Dešimties skaičių modelių failai:

Dvidešimt šešių vardų modelių failai:

hmm_nulis
hmm_vienas
hmm_du
hmm_trys
hmm_keturi
hmm_penki
hmm_sesi
hmm_septyni
hmm_astuoni
hmm_devyni

hmm_austeja
hmm_boleslovas
hmm_cecilija
hmm_donatas
hmm_eimantas
hmm_fausta
hmm_grazvydas
hmm_hansas
hmm_izaokas
hmm_jonas
hmm_karolis
hmm_laima
hmm_martynas

hmm_nojus
hmm_oskartas
hmm_patrikas
hmm_qju
hmm_ricardas
hmm_sandra
hmm_teodoras
hmm_ulijona
hmm_vacys
hmm_washington
hmm_xsas
hmm_ygrekas
hmm_zacharijus

Sukūrus „*hmm*list“ failą, toliau reikia sukurti tiriamų žodžių tekstinį failą, „*dikt.txt*“. Šio failo paskirtis yra nurodyti atpažinimo sistemai, kokius žodžius reikia atvaizduoti kompiuterio ekrane, kai bus atpažintas žodžio panašiausias modelis. Šio failo turinys atrodo taip:

Dešimties skaičių modelių failai:

<i>Nulis</i>	<i>hmm_nulis</i>
<i>Vienas</i>	<i>hmm_vienas</i>
<i>Du</i>	<i>hmm_du</i>
<i>Trys</i>	<i>hmm_trys</i>
<i>Keturi</i>	<i>hmm_keturi</i>
<i>Penki</i>	<i>hmm_penki</i>
<i>Šeši</i>	<i>hmm_sesi</i>
<i>Septyni</i>	<i>hmm_septyni</i>
<i>Aštuoni</i>	<i>hmm_astuoni</i>
<i>Devyni</i>	<i>hmm_devyni</i>

Dvidešimt šešių vardų modelių failai:

<i>Autėja</i>	<i>hmm_austeja</i>	<i>Nojus</i>	<i>hmm_nojus</i>
<i>Boleslovas</i>	<i>hmm_boleslovas</i>	<i>Oskaras</i>	<i>hmm_oskartas</i>
<i>Cecilija</i>	<i>hmm_cecilija</i>	<i>Patrikas</i>	<i>hmm_patrikas</i>
<i>Donatas</i>	<i>hmm_donatas</i>	<i>Qju</i>	<i>hmm_qju</i>
<i>Eimantas</i>	<i>hmm_eimantas</i>	<i>Ričardas</i>	<i>hmm_ricardas</i>
<i>Fausta</i>	<i>hmm_fausta</i>	<i>Sandra</i>	<i>hmm_sandra</i>
<i>Gražvydas</i>	<i>hmm_grazvydas</i>	<i>Teodoras</i>	<i>hmm_teodoras</i>
<i>Hansas</i>	<i>hmm_hansas</i>	<i>Ulijona</i>	<i>hmm_ulijona</i>
<i>Izaokas</i>	<i>hmm_izaokas</i>	<i>Vacys</i>	<i>hmm_vacys</i>
<i>Jonas</i>	<i>hmm_jonas</i>	<i>Wašington</i>	<i>hmm_washington</i>
<i>Karolis</i>	<i>hmm_karolis</i>	<i>Xsas</i>	<i>hmm_xsas</i>
<i>Laima</i>	<i>hmm_laima</i>	<i>Ygrekas</i>	<i>hmm_ygrekas</i>
<i>Martynas</i>	<i>hmm_martynas</i>	<i>Zacharijus</i>	<i>hmm_zacharijus</i>

7 ETAPAS – Modelių testavimas. Atlikus visus paminėtus veiksmus yra atliekamas atpažinimui apmokytų žodžių modelių testavimas. Testavimui atlikti yra naudojama programa „*Hvite.exe*“. Programa atlikusi testavimą, automatiškai sukuria gautųjų atpažinimo rezultatų „*results*“ failą. Šiame faile, gautieji atpažinimo tikslumo rezultatai, yra pateikiami eilutėmis, kur kiekviena eilutė parodo diktorius, tam tikro skaičiaus pavadinimo atpažinimo rezultatą. Gautieji rezultatai yra surašomi tokia tvarka, kokia buvo pateikti faile „*test_mfc.scf*“.

Pavyzdžiui „*results*“ failo turinys su skaičių atpažinimo testavimo rezultatais atrodo taip:

"C:/HTK/HTK_GB1/pozymiai2/FIEVVIS003.rec" // gerai atpažintas
0 8500000 nulis -4376.542480

"C:/HTK/HTK_GB1/pozymiai2/FIEVVIS004.rec" // blogai atpažintas
0 9300000 vienas -4814.442383

FIEVVIS004 – atitinkamo diktoriaus išartosios balso komandos žymėjimas ir ištarimo (nuo 0-19) eilės numeris.

0 – nurodo kokią reikėjo atpažinti komandą – „Nulis“, 04 ištarimo numeris garsyne. Sistema atpažino komandą – „Vienas“ nurodytam ištarimui. Pvz. : 103 – komanda – “Vienas”, trečias ištarimas, 615 – komanda „Šeši“, penkioliktas ištarimas it t.t

Šie rezultatai yra suskaičiuojami rankiniu būdu, skaičiuojant klaidas arba teisingai atpažintus skaičių pavadinimų vienetus.

Naudojant HTK programinį paketą buvo atliktas tyrimas balso komandų atpažinimo tikslumui įvertinti naudojant paslėptųjų Markovo grandinių metodą tiriamųjų komandų žodžių modeliams sukurti.

2.4.2 HTK programų rinkinio panaudojimas taikant fonemų modelių metodą šnekos atpažinimui

Šnekamosios kalbos atpažinimui taikyti HTK programinių įrankių paketą galima ir modeliuojant balso komandų atpažinimą fonemų modelių metodu. Norit įvertinti šio metodo efektyvumą balso komandų atpažinimui modeliuoti, šio darbo metu balso komandų atpažinimo tikslumui gautam taikant fonemų modelius buvo atliktas tyrimas ir gauti rezultatai palyginti su balso komandų atpažinimo rezultatais gautais taikant žodžių modelių metodą.

Šis balso komandų modeliavimo metodas buvo taikomas tik skaičių garsynui atsižvelgiant į tai, kad balsu išartų kodų sistemai atpažinti reikės daugiausia skaičius (visuose koduose daugiausia vyrauja skaičiai).

Taikant HTK programinių įrankių paketą balso komandų modeliavimui fonemų metodu reikia specialiai paruošti duomenis ir atitinkamus failus reikalingus modelių apmokymui ir atpažinimui. Balso komandų modeliavimui ir atpažinimui taikant fonemų modelius reikia nuosekliai atlikti aštuonis duomenų ir failų paruošimo etapus, pirmieji septyni etapai reikalingi fonemų PMM modelių paruošimui ir apmokymui, o aštuntais etapas modelių testavimui.

1 ETAPAS - Gramatikos failo kūrimas. Faile aprašomos norimos atpažinti balso komandos ir jų tvarka atpažinimo procese. Šiame tyrime naudojamų lietuviškų skaičių pavadinimų komandų aprašymas „*gram*“ faile atrodo taip (jis lygiai su tokiu pačiu turiniu kaip ir žodžių modelių metodui naudojamas „*gram.dikt*“):

\$word=VIENAS | DU | TRYS | KETURI | PENKI | SHESHI | SEPTYNI | ASHTUONI | DEVYNI | NULIS;
((\$word))

Panaudojant HTK programinį įrankį - „*Hparse.exe*“, kuris naudoja prieš tai sukurtą „*gram*“ – žodžių sąrašo failą, automatiškai sukuriamas tekstinis failas „*wdnet.txt*“.

2 ETAPAS – fonemų modelių sudarymui reikalingų failų paruošimas. Reikalingi failai – „*dict.txt*“, „*monophones1.txt*“, „*monophones0.txt*“ paruošiami rankiniu būdu, nenaudojant specialių automatinių priemonių, nes naudojamas skaičių komandų sąrašas susideda tik iš dešimties komandų.

Faile „*dict.txt*“ pateikiamos skaičių komandų žodynas, kuris atitinkamai suskaidytas į smulkesnius foneminius segmentus – fonemas. Fonema „*sp*“ – trumpą pauzę reiškianti fonema kuri prirašoma prie kiekvienos komandos tam kad jos būtų viena nuo kitos atskirtos. „SENT-END [] sil“ ir „SENT-START [] sil“ yra naudojamos aprašyti tylos modeliui. Faile „*dict.txt*“ skaičių komandų žodynas yra išdėstytas abėcėlės tvarka, šio failo turinys atrodo taip:

ASHTUONI	a s h t u o n i	sp
DEVYNI	d e v y n i	sp
DU	d u	sp
KETURI	k e t u r i	sp
NULIS	n u l i s	sp
PENKI	p e n k i	sp
SENT-END	[]	sil
SENT-START	[]	sil
SEPTYNI	s e p t y n i	sp
SHESHI	s h e s h i	sp
TRYS	t r y s	sp
VIENAS	v i e n a s	sp

Iš failo „*dict.txt*“ išrenkamos visos pasikartojančios fonemos ir surašomos į „*monophones1.txt*“ failą įtraukiant ir trumpos pauzės „*sp*“ ir tylos „*sil*“ fonemas. Failo „*monophones1.txt*“ turinys atrodo taip:

v
ie
n
a
s
d
u
t
r
y
k
e
i
p
sh
uo
l

sp
sil

Iš failo „*monophones1.txt*“ nukopijuojamos visos fonemos ir įkeliamos į „*monophones0.txt*“ failą neįtraukiant trumpos pauzės „*sp*“ fonemos. Failo „*monophones0.txt*“ turinys atrodo taip:

v
ie
n
a
s
d
u
t
r
y
k
e
i
p
sh
uo
l
sil

3 ETAPAS – transkripcijų failo kūrimas. Failai „*words.mlf*“ ir „*mkphones.led*“ paruošiami „rankiniu“ būdu. „*Words.mlf*“ faile yra surašomas pilnas visų komandų transkripcijų rinkinys. „*Words.mlf*“ paruošiamas pagal „*dict*“ faile esančias komandas ir duomenų, skirtų apmokymui, failų pavadinimus (FAGNGRA001: F - female/AGNGRA - diktorius/0 - komandos pabadinimas - NULIS/00 – ištaramo eilės numeris). „*Words.mlf*“ failo fragmentas atrodo taip:

```
#!MLF!#
"/FAGNGRA000.lab"
NULIS
.
"/FAGNGRA001.lab"
NULIS
.
"/FAGNGRA002.lab"
NULIS
.
...
"/FAGNGRA019.lab"
NULIS
.
"/FAGNVIN000.lab"
NULIS
.
```

```
"*/FAGNVIN001.lab"
NULIS
.
...
"*/FAUSNEM019.lab"
NULIS
.
"*/FAGNGRA100.lab"
VIENAS
.
"*/FAGNGRA101.lab"
VIENAS
.
```

Faile „*mkphones0.led*“ surašoma atitinkamos komandos kurios bus taikomos aktyvavus programą „*Hled.exe*“. Failo „*mkphones0.led*“ turinys atrodo taip:

```
EX
IS sil sil
DE sp
```

Komanda „*EX*“ pakeičia kiekvieną ištirimą „*words.mlf*“ faile pagal atitinkamą fonemą iš fonemų sąrašo aprašyto žodyno „*dict.txt*“ faile.

Komanda „*IS*“ įterpia tylos fonemą „*sil*“ kiekvieno ištirimą pradžioje ir pabaigoje.

Komanda „*DE*“ ištrina visas trumpų pauzių - „*sp*“ žymes, kurios yra nereikalingos šiame etape.

Susiredagavus „*words.mlf*“ ir „*mkphones0.led*“ failus aktyvuojamas HTK programinis įrankis - „*HLED.exe*“, kuris pasinaudojęs failuose „*words.mlf*“ ir „*mkphones0.led*“ nurodytai parametrai apdoroja kalbos signalą ir išskiria požymio vektorius - sukuria failą „*phones0.mlf*“, kurio turinio fragmentas atrodo taip:

```
#!MLF!#
"*/FAGNGRA000.lab"
sil
n
u
l
i
s
sil
.
"*/FAGNGRA001.lab"
sil
n
u
l
i
s
sil
.
```

Turint „*phones0.mlf*“ failą, visame jo turinyje ženklai - ‘*’, pakeičiami į „*features*“ (“*features/FAGNGRA001.lab*”). Po to aktyvuojama programa „*HLEd2.bat*“, kuri atitinkamai pasinaudojusi failuose „*words.mlf*“ ir „*mkphones1.led*“ esančia informacija sukuria failą „*phones1.mlf*“, kurio turinio fragmentas atrodo taip:

```
#!MLF!#
"*"/FAGNGRA000.lab"
sil
n
u
l
i
s
sp
sil
.
"*"/FAGNGRA001.lab"
sil
n
u
l
i
s
sp
sil
.
```

Kaip matome „*phones0.mlf*“ ir „*phones1.mlf*“ failai yra beveik identiški, tik „*phones0.mlf*“ ištarimų transkripcijos yra be trumpųjų pauzių fonemų „*sp*“. Taip pat kaip ir „*phones0.mlf*“ taip ir „*phones1.mlf*“ visame failo turinyje ženklai - ‘*’, pakeičiami į „*features*“ (“*features/FAGNGRA001.lab*”).

4 ETAPAS – Duomenų kodavimas.

Garsynų duomenys audio formatu yra laikomi duomenų kataloge *data*, o šiame etape sukurtus požymius laikysime kataloge – *features*.

Duomenų kodavimas vyks aktyvavus HTK programinį įrankį „*HCopy.exe*“, kuris automatiškai pagal nurodytus parametrus „*config.txt*“ faile, apdoroja kalbos signalą ir išskiria MFCC požymio vektorius. MFCC (Mel Frequency Cepstral Coefficients) - Melų dažnių kepstriniai koeficientai, kurie skaičiuojami apibrėžiant L trikampių filtrų, tolygiai išdėstytų melų skalėje, ir taikant kosinusinę transformaciją energijos kiekvieno filtro išėjime logaritmui. Plačiau apie MFCC aprašyta BBB BBB skyriuose.

Duomenų kodavimui reikalingus tekstinius „*config.txt*“ ir „*codetr.txt*“ failus sukuriame „rankiniu“ būdu, kur surašome programos „*HCopy.exe*“ aktyvavimui reikalingus parametrus ir informaciją. Parametrai „*config.txt*“ faile aprašomi taip:

```
SOURCEFORMAT=WAVE
HIFREQ=8000
TARGETKIND=MFCC_E
TARGETRATE=100000
SAVECOMPRESSED=T
SAVEWITHCRC=T
WINDOWSIZE=200000
USEHAMMING=T
PREEMCOEF=0.97
NUMCHANS=26
CEPLIFTER=22
NUMCEPS=12
```

Kai kurie iš šių parametrų yra nekeičiami ir paliekami pagal nutylėjimą tokie kokie siūlomi HTK programinio paketo.

Faile „*codetr.txt*“ surašome visų programos „*HCopy.exe*“ naudojamų duomenų (daugiadiktorinio garsyno komandų ištarių) direktorijas – iš kur paimti pradiniai duomenis ir kur patalpinti duomenis po apdorojimo, nes HTK paketas negali tiesiogiai apdoroti garsyno žodžių rinkinio. Todėl jis rūšiuojamas nurodant kelią iki garso rinkmenos ir išdėstant į atskiras eilutes tai rinkmenai priklausančius žodžius. Failo „*codetr.txt*“ turinio fragmentas atrodo taip:

```
data\FAGNGRA000.wav features\FAGNGRA000.mfc
data\FAGNGRA001.wav features\FAGNGRA001.mfc
data\FAGNGRA002.wav features\FAGNGRA002.mfc
.....
data\FAGNGRA003.wav features\FAGNGRA007.mfc
data\FAGNGRA004.wav features\FAGNGRA008.mfc
```

Paruošus „*config.txt*“ ir „*codetr.txt*“ failus aktyvuojamas „*HCopy.exe*“ duomenų kodavimo programinis įrankis, kuris apdoroja duomenis pagal nurodytus „*config.txt*“ faile parametrus ir sukuria foneminių atpažinimo modelių apmokymui naudojamų požymių failus ir patalpina juos kataloge „*features*“.

5 ETAPAS – foneminių modelių kūrimas. Šiame etape pirmiausiai „rankiniu“ būdu yra sukuriami du tekstiniai failai „*train.txt*“ ir „*config6.txt*“. Failas „*config6.txt*“ iš esmės yra lygiai toks pat, kaip ketvirtajame etape sukurtas „*config.txt*“ tik melų dažnių skales koeficientų požymis turi didesnę reikšmę 39 (ta reikšmė nurodo fonemos modelio vektoriaus dydį). Vektoriaus dydis 39 gaunamas prie MFCC parametrų vektoriaus ilgio (13) pridėjus delta

koeficientus (+13) ir pridėjus akceleracijos koeficientus (+13). Failo „*config6.txt*” turinys atrodo taip:

```
TARGETKIND=MFCC_E_D_A_Z
TARGETRATE=100000
SAVECOMPRESSED=T
SAVEWITHCRC=T
WINDOWSIZE=200000
USEHAMMING=T
PREEMCOEF=0.97
NUMCHANS=26
CEPLIFTER=22
NUMCEPS=12
ALLOWXWRDEXP=T
```

Taip pat sukuriamas „*train.txt*“ failas, kuriame yra surašomos nuorodos iš kur programa „*HCompv.exe*“ pasiims jai reikalingus visų komandų ištarimų požymių failus. Failo „*train.txt*“ fragmentas atrodo taip:

```
features\FAGNGRA000.mfc
features\FAGNGRA001.mfc
.....
features\FAGNGRA004.mfc
```

Turint paruoštus „*train.txt*“ ir „*config6.txt*“ failus sukuriamas naujas katalogas ir pavadinamas „*hmm0*“, po to aktyvuojame HTK programinį įrankį „*HCompv.exe*“. Programai reikalingas yra dar vienas pradinis modelio failas „*proto*“, kuriame yra išdėstyti pradiniai modelio vektoriaus reikšmių koeficientai. Pradinio „*proto*“ failo turinys atrodo taip:

```
~o <VecSize> 39 <MFCC_E_D_A_Z>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
```

```

1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 4
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<Transp> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>

```

~o – pradžios simbolis;

<VECSIZE> 39<NULLD><MFCC_E_D_A_Z> – šioje eilutėje nurodomas vektoriaus dydis - tie patys 39 požymiai, MFCC_E_D_A_Z reiškia, kad bus naudojami MFCC požymis ir E – energijos, D – delta, A – delta delta ir Z – nulinės reikšmės, naudojamas vektoriaus normalizavimui, požymio koeficientai;

~h "proto" – tai modelių failo pavadinimas;

<BEGINHMM> – modelio pradžios žyma;

<NUMSTATES> - nurodomas modelių faile naudojamas būsenų skaičius;

<STATE> 2 – modelis visada prasideda nuo antros būsenos;

<MEAN> 39 – požymių vidurkiai, kurie modelių mokymo metu bus perskaičiuojami automatiškai;

<VARIANCE> 39 – požymių dispersijos, kurios apmokymo metu bus perskaičiuotos automatiškai;

<TRANSP> - „transponentė“ - paskutinė būsena. Šioje būsenoje yra sudaroma tikimybių matrica;

<ENDHMM> - modelio pabaigos žyma;

Programa „HCompv.exe“ išanalizuoja požymio vektorius ir suskaičiuoja vidurkį bei kovariaciją. Šie duomenys pakeičia pirminiame modelyje esančias nulines reikšmes ir kataloge

„hmm0“ sukuria du naujus failus „proto“ (naujame proto atsiranda vektoriaus koeficientų reikšmės pagal naudojamus duomenis) ir „vFloors“.

Failo „proto“ su perskaičiuotomis vektoriaus koeficientų reikšmėmis turinys atrodo taip:

```
~0
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A_Z><DIAGC>
~h "proto"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
<STATE> 3
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
<STATE> 4
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
```

```
<TRANSP> 5
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 7.000000e-001 3.000000e-001
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

Faile „*vFloors*“ su vektoriaus koeficientų reikmėmis turinys atrodo taip:

```
~v varFloor1
<Variance> 39
3.894049e-001 3.416032e-001 4.127912e-001 6.067739e-001 5.985034e-001 3.724602e-001 4.952216e-001
3.666311e-001 3.656544e-001 3.557177e-001 2.991481e-001 2.102285e-001 9.218900e-004 1.725736e-002
1.837272e-002 1.816840e-002 2.467288e-002 2.953259e-002 2.371335e-002 2.637159e-002 2.656657e-002
2.403944e-002 2.692570e-002 2.171899e-002 1.745611e-002 2.106186e-005 2.576748e-003 2.700861e-003
2.725722e-003 3.543172e-003 4.782551e-003 4.058538e-003 4.528526e-003 4.776322e-003 4.292024e-003
4.714570e-003 3.876490e-003 3.251144e-003 2.808729e-006
```

Taip pat kataloge „*hmm0*“ reikia „rankiniu“ būdų susikurti du naujus failus – „*macros*“ ir „*hmmdefs*“. Failas „*macros*“ sukuriamas prie „*vFloors*“ faile esančios informacijos pridėdant kelias papildomas parametrų eilutes:

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A_Z><DIAGC>
~v varFloor1
<Variance> 39
3.894049e-001 3.416032e-001 4.127912e-001 6.067739e-001 5.985034e-001 3.724602e-001 4.952216e-001
3.666311e-001 3.656544e-001 3.557177e-001 2.991481e-001 2.102285e-001 9.218900e-004 1.725736e-002
1.837272e-002 1.816840e-002 2.467288e-002 2.953259e-002 2.371335e-002 2.637159e-002 2.656657e-002
2.403944e-002 2.692570e-002 2.171899e-002 1.745611e-002 2.106186e-005 2.576748e-003 2.700861e-003
2.725722e-003 3.543172e-003 4.782551e-003 4.058538e-003 4.528526e-003 4.776322e-003 4.292024e-003
4.714570e-003 3.876490e-003 3.251144e-003 2.808729e-006
```

Toliau naujai sukurtas modelis „*proto*“ pridėdama prie kiekvienos fonemos naudojamos garsyne, tam sukuriamas failas „*hmmdefs*“ prie „*monophones0.txt*“ faile esančių fonemų sąrašo, kiekvienai fonemai pridėdant „~h“ žymę ir „*hmm0*“ esančio failo „*proto*“, su perskaičiuotomis reikšmėmis, fragmentą prasidedantį <BEGINHMM> ir pasibaigiantį <ENDHMM> .

```
~h "v"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
```

```

8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
<STATE> 3
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
<STATE> 4
<MEAN> 39
1.561484e-008 6.831492e-009 5.508789e-009 -2.040531e-008 -9.085538e-009 -3.433085e-009 9.660195e-009
2.665223e-009 -1.574364e-008 -6.781953e-009 -2.598345e-009 1.991487e-009 5.088044e-001 -3.026696e-002
-3.461873e-002 -1.391887e-002 1.820096e-003 2.929172e-002 2.267688e-002 2.742604e-002 2.578177e-002
1.248765e-002 1.116670e-002 1.267690e-002 9.401998e-003 -1.331803e-003 -1.144430e-003 -1.543300e-003 -
8.233905e-004 -2.339742e-004 8.822206e-004 1.175808e-003 1.953176e-003 5.046910e-004 6.960688e-006
3.109000e-004 1.251589e-003 9.757830e-004 -7.066995e-005
<VARIANCE> 39
3.894049e+001 3.416032e+001 4.127912e+001 6.067739e+001 5.985033e+001 3.724602e+001
4.952216e+001 3.666311e+001 3.656544e+001 3.557177e+001 2.991481e+001 2.102285e+001 9.218901e-002
1.725736e+000 1.837272e+000 1.816840e+000 2.467288e+000 2.953259e+000 2.371335e+000 2.637160e+000
2.656657e+000 2.403944e+000 2.692570e+000 2.171899e+000 1.745611e+000 2.106186e-003 2.576748e-001
2.700861e-001 2.725722e-001 3.543172e-001 4.782552e-001 4.058538e-001 4.528526e-001 4.776323e-001
4.292024e-001 4.714570e-001 3.876490e-001 3.251144e-001 2.808729e-004
<GCONST> 9.670569e+001
<TRANSP> 5
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 7.000000e-001 3.000000e-001
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
~h "ie"
<BEGINHMM>
.....

```

Toliau tęsiamas paslėptųjų Markovo modelių apmokymas naudojant HTK programinį įrankį „*HERest.exe*“, kuris, taikydamas *Baum-Welch* algoritimą, patikslina sukurto modelio

reikšmes. „*HERest.exe*“ įrankis bus aktyvuojamas tris kartus sukurtų fonemų modelių esančių „*phones1.mlf*“ perskaičiavimui, atsižvelgiant į požymio vektorius.

Pirmajam sukurtų fonemų modelių perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm1*“ . Tada aktyvuojamas programinis įrankis „*HERest.exe*“, kuris naudodamas „*hmm0*“ kataloge esančiais pirminiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm1*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

Antrajam sukurtų fonemų modelių perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm2*“ . Tada aktyvuojamas programinis įrankis „*HERest2.bat*“, kuris naudodamas „*hmm1*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm2*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

Trečiajam sukurtų fonemų modelių perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm3*“ . Tada aktyvuojamas programinis įrankis „*HERest3.bat*“, kuris naudodamas „*hmm2*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm3*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

6 ETAPAS – Tylos modelių taisymas. Šiame etape sukuriamas naujas katalogas ir pavadinamas „*hmm4*“. Į šį katalogą nukopijuojami failai „*hmmdefs*“ ir „*macros*“ iš „*hmm3*“ katalogo. Į kataloge „*hmm4*“ esantį failą „*hmmdefs*“ įtraukiamas trumpų pauzių „*sp*“ modelis, gautas iš tylos „*sil*“ modelio pašalinant antrą (2) ir ketvirtą (4) būsenas, paliekant tik trečią „3“ būseną, pernumeruotą antra būseną (3>>2), ir pakeičiant „*transp*“ pradinę 5x5 matricią į 3x3 matricią. Papildyto, trumpų pauzių modeliu „*sp*“, failo „*Hmmdefs*“, „*sp*“ modelis atrodo taip:

```
~h "sp"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
-2.259850e+000 2.275288e+000 -2.316948e+000 5.132991e+000 3.406356e+000 2.333344e+000
2.294427e+000 7.006237e-001 1.771211e+000 -5.445279e-002 8.948746e-001 -8.662137e-002 3.109235e-001
-2.967556e-002 -2.528537e-002 7.466571e-002 8.343749e-003 -6.718785e-002 3.671031e-002 -4.218653e-002
-9.004824e-002 -2.945648e-002 -3.920250e-002 -1.254476e-002 -3.754782e-002 8.468692e-003 -2.724073e-002
-1.155186e-001 6.829675e-002 -1.420055e-001 -9.289731e-002 -1.292715e-002 -1.575024e-002 -6.341466e-002
-1.632910e-002 4.378007e-002 3.664609e-002 2.599909e-002 9.815449e-003
<VARIANCE> 39
2.931908e+001 2.841906e+001 2.183179e+001 2.024068e+001 2.456464e+001 2.373295e+001
2.371367e+001 2.406507e+001 1.999465e+001 2.205825e+001 1.906113e+001 1.606240e+001 6.775062e-002
7.206373e-001 7.193661e-001 8.375908e-001 9.316252e-001 1.426072e+000 1.599122e+000 1.672637e+000
1.742921e+000 1.645044e+000 1.613045e+000 1.452183e+000 1.301615e+000 7.302568e-004 1.875999e-001
1.522206e-001 1.900400e-001 2.184787e-001 2.739959e-001 3.011622e-001 2.983601e-001 3.151499e-001
3.057576e-001 3.173104e-001 2.946637e-001 2.580854e-001 3.126432e-004
<GCONST> 7.713361e+001
<TRANSP> 3
0.0 1.0 0.0
0.0 0.9 0.1
```

Toliau sukuriamas naujas katalogas ir pavadinamas „*hmm5*“. Tada aktyvuojamas „*HHEd.exe*“, failas, kuris prideda reikalingų papildomų perėjimų perkeliant trumpų pauzių „*sp*“ modelį į tylos modelio „*sil*“ centrą. Programa „*HHEd.exe*“ veikia panašiai kaip „*HLEd.exe*“, naudodama paruoštus duomenis perskaičiuoja fonemų paslėptuosius Markovo modelius. „*HHEd.exe*“ programa naudoja failus „*hmmdefs*“ ir „*macros*“ iš katalogo „*hmm4*“ ir dar vieną failą „*sill.hed*“, kuriame esanti informacija yra HTK numatytoji ir nekeičiama. Failo „*sil.hed*“ turinys atrodo taip:

```
AT 2 4 0.2 {sil.transP}
AT 4 2 0.2 {sil.transP}
AT 1 3 0.3 {sp.transP}
TI silst {sil.state[3],sp.state[2]}
```

Aktyvavus „*HHEd.exe*“ programą, kataloge „*hmm5*“ sukuria „*hmmdefs*“ ir „*macros*“ failus.

Po to toliau tęsiamas paslėptųjų Markovo modelių apmokymas naudojant HTK programinį įrankį „*HERest.exe*“. Šiame etape „*HERest.exe*“ įrankis bus aktyvuojamas du kartus faile „*hmmdefs*“ sukurtų fonemų modelių, papildytų trumpų pauzių modeliu „*sp*“, perskaičiavimui, atsižvelgiant į požymio vektorius.

Pirmajam sukurtų fonemų modelių, papildytų trumpų pauzių modeliu „*sp*“, perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm6*“. Tada aktyvuojamas programinis įrankis „*HERest4.bat*“, kuris naudodamas „*hmm5*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm6*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

Antrajam sukurtų fonemų modelių, papildytų trumpų pauzių modeliu „*sp*“, perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm7*“. Tada aktyvuojamas programinis įrankis „*HERest5.bat*“, kuris naudodamas „*hmm6*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm7*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

7 ETAPAS – Fonemų modelių apmokymas. Fonemų modelių apmokymui naudojamas „*HVite.exe*“ programa, kuris pagal jau sukurtus fonemų modelius sukuria naujas komandų transkripcijas fonemų modelių apmokymui balso komandoms atpažinti. „*HVite.exe*“ programa naudoja fonemų paslėptuosius Markovo modelius failuose „*hmmdefs*“ ir „*macros*“, kurie yra kataloge „*hmm7*“, sumodeliuoti „*words.mlf*“ faile surašytą pilną visų komandų

transkripcijų rinkinį, naudodama „*dict.txt*“ faile esančius komandų foneminius modelius, į naujas transkripcijas, kurioms programa sukuria failą „*aligned.mlf*“. Failo „*aligned.mlf*“ turinys lygiai toks pat kaip failo „*phones1.mlf*“. Sukurtame „*aligned.mlf*“ visame failo, kaip ir „*phones1.mlf*“ failo, turinyje ženklai - „*“, pakeičiami į „features“ („features/FAGNGRA001.lab“).

Toliau tęsiamas paslėptųjų Markovo modelių apmokymas naudojant HTK programinį įrankį „*HERest.exe*“. Šiame etape „*HERest.exe*“ įrankis bus aktyvuojamas du kartus sukurtų fonemų modelių esančių „*aligned.mlf*“ perskaičiavimui, atsižvelgiant į požymio vektorius.

Pirmajam, šiame etape, naujai sumodeliuotų fonemų modelių perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm8*“. Tada aktyvuojamas programinis įrankis „*HERest6.bat*“, kuris naudodamas „*hmm7*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm8*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus.

Antrajam, šiame etape, naujai sumodeliuotų fonemų modelių perskaičiavimui sukuriamas naujas katalogas ir pavadinamas „*hmm9*“. Tada aktyvuojamas programinis įrankis „*HERest7.bat*“, kuris naudodamas „*hmm8*“ kataloge esančiais „*hmmdefs*“ ir „*macros*“ failais, „*hmm9*“ kataloge sukuria naujus „*hmmdefs*“ ir „*macros*“ failus. Kataloge „*hmm9*“ esantys fonemų paslėptieji Markovo modeliai failuose „*hmmdefs*“ ir „*macros*“ yra naudojami balso komandų atpažinimo testavimui pagal fonemų modelius.

8 ETAPAS – Atpažinimo testavimas pagal fonemų modelius. Šiame etape pirmiausia sukuriamas naujas katalogas ir pavadinamas „*Step11_9*“. Kataloge „*hmm9*“ esantys apmokyti fonemų paslėptieji Markovo modeliai failuose „*hmmdefs*“ ir „*macros*“ yra naudojami balso komandų atpažinimo testavimui pagal fonemų modelius, todėl failai „*hmmdefs*“ ir „*macros*“ yra nukopijuojami į naujai sukurtą „*Step11_9*“ katalogą. Atpažinimo testavimo etape, apmokytų fonemų modelių atpažinimo tikslumo testavimui naudojami kiti garsyno duomenys audio formatu. Katalogas „*data*“ su naujais testavimui skirtais duomenis įkeliamas į katalogą „*Step11_9*“. Naujų duomenų kodavimui reikalingus tekstinius „*config.txt*“ ir „*codetr.txt*“ failus sukuriame „rankiniu“ būdu, kur surašome programos „*HCopy.exe*“ aktyvavimui reikalingus parametrus ir informaciją. Parametrai „*config.txt*“ faile aprašomi taip:

```
SOURCEFORMAT=WAVE
TARGETKIND=MFCC_E_D_A_Z
TARGETRATE=100000
SAVECOMPRESSED=T
SAVEWITHCRC=T
```


WINDOWSIZE=200000
USEHAMMING=T
PREEMCOEF=0.97
NUMCHANS=26
CEPLIFTER=22
NUMCEPS=12
FORCECXTXP=T
ALLOWXWRDEXP=F

Testavimui naudojamas „*codetr.txt*“ failas kuriamas taip pat, kaip apmokymui naudojamas „*codetr.txt*“ failas, tik nurodant naujų testavimui naudojamų duomenų direktorijas.

Paruošus „*config.txt*“ ir „*codetr.txt*“ failus aktyvuojamas „*HCopy.exe*“ duomenų kodavimo programinis įrankis, kuris apdoroja duomenis pagal nurodytus „*config.txt*“ faile parametrus ir sukuria foneminių atpažinimo modelių testavimui naudojamų požymių failus ir patalpina juos naujai sukurtame kataloge „*features*“.

Fonemų modelių testavimui naudojamas „*HVite.exe*“ programa. „*HVite.exe*“ programa naudoja fonemų paslėptuosius Markovo modelius failuose „*hmmdefs*“ ir „*macros*“, kurie yra kataloge „*hmm9*“, sumodeliuoti „*testref.mlf*“ faile surašytą pilną visų komandų transkripcijų rinkinį, naudodama „*dict.txt*“ faile esančius komandų foneminius modelius, į naujas transkripcijas, kurioms programa sukuria failą „*recout.mlf*“.

Šiame testavimo etape „*HVite.exe*“ programa naudoja papildomus failus kurie yra sukuriami „rankiniu“ būdu naudojant naujus testavimui skirtus duomenis, informaciją apie juos, ir juos talpinančias direktorijas. Failas „*test.scp*“ kuriamas taip pat, kaip *train.scp* failas, failas „*testref.mlf*“ kuriamas, kaip „*words.mlf*“ failas. Taip pat reikalingi failai yra „*wdnet*“, „*dict.txt*“ ir „*monophones1*“, kurie yra jau paruošti pirmuose modelių apmokymo paruošimo etapuose ir nukopijuojami į katalogą „*Step11_9*“.

Apmokyti fonemų paslėptųjų Markovo modelių testavimo rezultatams gauti naudojamas HTK programinis įrankis „*HResults.exe*“. Jį aktyvavus, pasinaudodamas failuose „*testref.mlf*“, „*dict.txt*“, „*recout.mlf*“ paruošta informacija, jis rezultatų sumaišymo matricą pateikia į failą „*rez*“. Komandų atpažinimo tikslumo, taikant foneminį modeliavimą, rezultatų failo „*rez*“ turinys atrodo taip:

```
===== HTK Results Analysis =====  
Date: Tue Sep 16 17:30:51 2014  
Ref : testref.mlf  
Rec : recout.mlf  
----- Overall Results -----  
SENT: %Correct=62.08 [H=745, S=455, N=1200]  
WORD: %Corr=62.08, Acc=62.08 [H=745, D=0, S=455, I=0, N=1200]
```

```

----- Confusion Matrix -----
      A   D   K   N   P   S   S   T   V
      S   E   E   U   E   E   H   R   I
      H   V   T   L   N   P   E   Y   E
      T   Y   U   I   K   T   S   S   N
      U   N   R   S   I   Y   H           A Del [ %c / %e]
ASHT 120  0   0   0   0   0   0   0   0   0
DEVY  0  111  0   0   0   9   0   0   0   0 [92.5/0.8]
  DU  0   9  89   6  15   0   1   0   0   0 [ 0.0/10.0]
KETU  3   0  30   0   0  87   0   0   0   0 [25.0/7.5]
NULI  0   0   0  120  0   0   0   0   0   0
PENK  0   0   0   0  15  105  0   0   0   0 [12.5/8.8]
SEPT  0   2   0   0   0  118  0   0   0   0 [98.3/0.2]
SHES  17  0   0   0   0   0  103  0   0   0 [85.8/1.4]
TRYS  0   0  54  19   0   7   0  40   0   0 [33.3/6.7]
VIEN  0   0   0  31   0   1   0   0  88   0 [73.3/2.7]
Ins   0   0   0   0   0   0   0   0   0
=====

```

Atpažinimo tikslumas procentais pateikiamas faile „rez“ prie žymės **Acc="**"**.

Taip pat priede HGH pateikiami ir visų, minėtuose aštuoniuose etapuose, HTK programinių įrankių, naudojamų balso komandų atpažinimo tikslumo tyrimui fonemų modelių metodu, „*.bat“ failų turiniai.

Naudojant HTK programinį paketą buvo atliktas tyrimas balso komandų atpažinimo tikslumui įvertinti naudojant paslėptųjų Markovo grandinių metodą tiriamųjų komandų foneminiams modeliams sukurti.

6 PRIEDAS. Fonemų deriniai

3.6.2. lentelė. Fonemų derinių sąrašas

Fonemų skaičius					
17	24	27	28	30	33
Fonemos					
v	vm	vm	vm	vm	vm
ie	ie	ie	ie	ie	ie
n	n	n	n	n	n
a	a	a	a	a	a
s	s	s	s	s	s
d	d	d	d	d	d
u	u	u	u	u	u
t	tm	tm	tm	tm	tm
r	rm	rm	rm	rm	rm
y	y	y	y	y	y
k	km	km	km	km	km
e	e	e	e	e	e
i	i	i	i	i	i
p	pm	pm	pm	pm	pm
sh	shm	shm	shm	shm	shm
uo	uo	uo	uo	uo	uo
l	lm	lm	lm	lm	lm
sp	sil	sil	sil	sil	sil
sil	t	t	t	t	t
	ii	ii	ii	ii	ii
	nk	nk	nk	nk	nk
	sh	sh	sh	sh	sh
	nm	nm	nm	nm	nm
	dm	dm	dm	dm	dm
	sm	sm	sm	sm	sm
	sp	ik	ik	ik	ik
		sp	ud	ud	ud
			sp	esh	esh
				ish	ish
				sp	ek
					ut
					ir
					sp

7 PRIEDAS. Atpažintuvų sujungimas

1.5.3 lentelė. Atpažintuvų ATP_SP ir ATP_LT rezultatų papildomumas

Poaibis	Aprašymas	Koman dų sk.
T=T	Atpažintuvų siūlomi sprendimai sutampa ir yra teisingi	135898
F=F	Atpažintuvų siūlomi sprendimai sutampa ir yra neteisingi	178
T-	ATP_LT siūlo teisingą sprendimą, atpažintuvas SP sprendimo nesiūlo	3398
F-	ATP_LT siūlo neteisingą sprendimą, atpažintuvas SP sprendimo nesiūlo	48
-T	ATP_SP siūlo teisingą sprendimą, atpažintuvas LT sprendimo nesiūlo	7
-F	ATP_SP siūlo neteisingą sprendimą, atpažintuvas LT sprendimo nesiūlo	1
--	Abu atpažintuvai nesiūlo sprendimo	1
TF	Atpažintuvų siūlomi sprendimai nesutampa, ATP_LT sprendimas teisingas	33650
FT	Atpažintuvų siūlomi sprendimai nesutampa, ATP_SP sprendimas teisingas	1357
FF	Atpažintuvų siūlomi sprendimai nesutampa, abu sprendimai neteisingi	902
Iš viso		175440

1.5.4 lentelė. Požymiai, kuriais aprašoma mokymo imtis, hibridiniam sprendimų priėmimo blokui sudaryti

Požymio pavadinimas	Paaiškinimas
sp_prob	SP atpažintuvo pateikto sprendimo patikimumo įvertis [0,...,1000]
sp_supp	Jei SP atpažintuvo pateiktas sprendimas sutampa su LT atpažintuvo pateikta 2-ąja (arba 3-ąja) alternatyva, šis parametras nurodo, kiek ši 2-oji (arba 3-oji) LT alternatyva yra prastesnė už 1-ąją (prioritetinį) LT sprendimą (logaritminės tikimybės prasme). Jei LT atpažintuvas nepateikia alternatyvių sprendimų, arba jei SP atpažintuvo siūlomas sprendimas nesutampa su LT atpažintuvo alternatyvomis, šiam požymiui priskiriama reikšmė 10.
lt_prob	LT atpažintuvo pateikto sprendimo patikimumo įvertis, matuojamas vidutine logaritmine tikimybe signalo analizės langui. Frazės pradžioje ir pabaigoje galimai esančios tylos atkarpos į šį įvertinimą neįtraukiamos
lt_delta_prob	Patikimumo įverčių skirtumas tarp prioritetinės ir 2-osios LT atpažintuvo alternatyvos. Jei LT atpažintuvas nepateikia alternatyvių sprendimų, šiam požymiui priskiriama reikšmė 10
sil_prob	Patikimumo įvertis apibūdinantis, kaip frazės pradžia ir pabaiga atitinka „tylos“ PMM (logaritminės tikimybės prasme)
gender	dvireikšmis požymis, nusakantis kalbėtojo lytį (m, f)
lt_a	Proporcija (%), kurią LT atpažintuvo pateiktame prioritetiniame sprendime (frazėje) sudaro „a“ raidės. Pvz. jei LT atpažintuvas pateikia prioritetinį sprendimą „bronchų astma“, tai šis požymis lygus 16,7% (2 raidės iš 12)
lt_b	Proporcija (%), kurią LT atpažintuvo pateiktame prioritetiniame sprendime (frazėje) sudaro „b“ raidės. Pvz. jei LT atpažintuvas pateikia prioritetinį sprendimą „bronchų astma“, tai šis požymis lygus 8,3% (1 raidė iš 12)
...	
lt_ž	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
lt_space	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_a	Toliau tokiu pat būdu transformuojamas SP atpažintuvo sprendimas, žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_b	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
...	
sp_ž	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_space	žr. požymių lt_a ir lt_b paaiškinimus aukščiau

8 PRIEDAS. TLK-10-AM klasifikavimas

1.7.1 lentelė. Ligų klasifikacijai priskiriamos raidės ir tipai

Skyrius	Skyriaus pavadinimas	Prieš kodą rašoma raidė	Tipas
1 skyrius	<i>Kai kurios infekcinės ir parazitinės ligos</i>	A, B	Ypatingas
2 skyrius	<i>Navikai</i>	C, D	Ypatingas
3 skyrius	<i>Kraujo ir kraujodaros organų ligos bei tam tikri sutrikimai susiję su imuniniais mechanizmais</i>	D	Pagal sritį
4 skyrius	<i>Endokrininės, mitybos ir medžiagų apykaitos ligos</i>	E	Pagal sritį
5 skyrius	<i>Psichikos ir elgesio sutrikimai</i>	F	Pagal sritį
6 skyrius	<i>Nervų sistemos ligos</i>	G	Pagal sritį
7 skyrius	<i>Akies ir jos priklausinių ligos</i>	H	Pagal sritį
8 skyrius	<i>Ausies, nosies, burnos ir gerklės ligos</i>	H	Pagal sritį
9 skyrius	<i>Kraujotakos sistemos ligos</i>	I	Pagal sritį
10 skyrius	<i>Kvėpavimo sistemos ligos</i>	J	Pagal sritį
11 skyrius	<i>Virškinimo sistemos ligos</i>	K	Pagal sritį
12 skyrius	<i>Odos ir poodžio ligos</i>	L	Pagal sritį
13 skyrius	<i>Jungiamojo audinio ir skeleto bei raumenų sistemos ligos</i>	M	Pagal sritį
14 skyrius	<i>Urogenitalinės sistemos ligos</i>	N	Pagal sritį
15 skyrius	<i>Nėštumas, gimdymas ir laikotarpis po gimdymo</i>	O	Ypatingas
16 skyrius	<i>Tam tikros perinatalinio periodo ligos</i>	P	Ypatingas
17 skyrius	<i>Įgimtos formavimosi ydos, deformacijos ir chromosomų anomalijos</i>	Q	Ypatingas
18 skyrius	<i>Simptomai, požymiai ir nenormalūs klinikiniai bei laboratoriniai radiniai neklasifikuojami kitur</i>	R	Netaikoma
19 skyrius	<i>Traumos, apsinuodijimai ir kiti išorinių priežasčių padariniai</i>	S, T	Ypatingas
20 skyrius	<i>Išorinės sergamumo priežastys</i>	U, V, W, X, Y	Netaikoma
21 skyrius	<i>Sveikatos būklę veikiančys faktoriai ir kontaktai su sveikatos tarnyba</i>	Z	Netaikoma
22 skyrius	<i>Ypatingų tikslų kodai</i>	U	Netaikoma

9 PRIEDAS. Komandų modelių pavyzdžiai su skirtingu būsenų ir Gauso mišinių skaičiumi

Skaičiaus „*nulis*“ modelis be papildomų būsenų ir Gauso mišinių:

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_nulis"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
-1.901997e+001 1.560556e+000 -1.774561e+000 2.421496e-001 -3.639415e+000 -2.854162e+000 -
4.346484e+000 -5.623057e+000 -2.414689e+000 -1.254046e+000 -4.055236e+000 -1.911147e+000
3.550545e-001 8.811274e-001 1.983937e-001 6.702835e-001 -1.182329e-001 -5.168684e-001 -
6.920862e-001 -9.938358e-001 -9.456979e-001 -3.544414e-001 6.312389e-002 -4.575621e-001 -
2.118120e-001 3.610023e-002 2.801377e-002 -9.071910e-002 2.609866e-002 -9.318825e-002 -
2.045594e-002 -2.656152e-002 -8.294988e-002 3.611838e-004 -8.503236e-003 6.372534e-002 -
2.874717e-002 4.794284e-003 4.444513e-003
<VARIANCE> 39
3.071770e+001 1.477026e+001 2.250223e+001 1.140425e+001 1.877273e+001 2.483272e+001
3.668488e+001 4.387595e+001 2.323556e+001 2.453082e+001 1.880439e+001 1.131520e+001 3.783294e-
002 1.749561e+000 1.218295e+000 1.341627e+000 8.983074e-001 1.604713e+000 2.192991e+000
2.631782e+000 2.262852e+000 1.671081e+000 1.967330e+000 1.637923e+000 1.174842e+000 1.834478e-
003 3.226228e-001 2.174498e-001 2.031977e-001 1.481956e-001 2.708266e-001 3.548790e-001
4.414195e-001 4.131571e-001 2.600133e-001 3.112520e-001 2.412909e-001 2.017353e-001 2.608459e-
004
<GCONST> 8.083940e+001
<STATE> 3
<MEAN> 39
-1.372821e+001 -3.729793e+000 -8.800527e-001 -7.259613e+000 -7.086869e+000 -7.174390e+000 -
1.306363e+001 -5.066110e+000 -2.701850e+000 -2.749123e-001 -6.670803e+000 -7.242477e-001
8.013315e-001 -3.476667e-002 -1.597483e-001 -1.703839e-001 -1.777720e-001 -4.978981e-002 -
1.704696e-001 -3.807238e-003 2.641870e-001 1.342109e-001 -5.678174e-002 -1.254449e-002
1.624089e-001 4.346405e-003 -2.986467e-002 6.189175e-002 -2.281057e-002 5.098714e-002
3.356664e-002 1.990484e-002 4.433486e-002 -1.348968e-002 9.457828e-003 -2.471972e-002
1.262858e-002 -1.027823e-002 -2.140006e-003
<VARIANCE> 39
6.178508e+000 2.059113e+001 4.763426e+001 5.306515e+001 3.882137e+001 5.146424e+001
3.923051e+001 5.768887e+001 3.635483e+001 3.424862e+001 3.747098e+001 3.176550e+001 1.662688e-
002 2.867131e-001 1.081594e+000 2.671461e+000 2.356773e+000 1.921422e+000 2.121924e+000
1.952770e+000 2.808614e+000 1.884219e+000 1.651934e+000 1.596778e+000 1.359137e+000 7.681514e-
004 6.314972e-002 1.685290e-001 3.274954e-001 3.568201e-001 3.335121e-001 3.718762e-001
3.445843e-001 4.999219e-001 3.318075e-001 2.971268e-001 2.825662e-001 2.313023e-001 9.668715e-
005
<GCONST> 8.348557e+001
<STATE> 4
<MEAN> 39
-2.027365e+001 2.312796e+000 -4.704744e+000 -2.531956e+000 -4.684047e+000 -9.801590e+000 -
5.432903e+000 -2.585647e+000 9.922757e-001 -2.189220e+000 -3.965700e+000 1.834216e+000
7.243886e-001 -2.313533e+000 1.574218e+000 -5.900860e-001 1.270094e+000 1.099012e+000
1.212665e+000 1.677734e+000 2.176636e-001 1.012306e-001 4.123354e-001 1.121954e+000 -
2.914392e-001 1.596921e-002 -1.440323e-001 -8.922359e-002 -5.627838e-002 -7.699709e-002 -
7.970725e-002 1.086868e-001 1.510285e-002 1.868139e-001 -3.067373e-002 1.569747e-001
5.138303e-002 5.629309e-002 1.231860e-002
<VARIANCE> 39
3.400875e+001 2.144253e+001 1.801310e+001 2.338734e+001 3.070754e+001 2.495817e+001
3.658425e+001 1.898655e+001 1.902155e+001 3.128481e+001 2.494522e+001 2.006646e+001 1.526070e-
002 1.586916e+000 1.711656e+000 1.275771e+000 2.699040e+000 1.951800e+000 2.116981e+000
2.313807e+000 2.032935e+000 1.608248e+000 2.716352e+000 2.388335e+000 1.905416e+000 2.326505e-
003 6.031662e-001 2.586293e-001 2.144947e-001 3.512403e-001 3.426210e-001 3.639666e-001
5.648019e-001 3.456514e-001 3.166753e-001 4.935391e-001 5.026808e-001 2.948453e-001 3.071198e-
004
<GCONST> 8.803616e+001
<TRANSP> 5
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 8.516096e-001 1.483904e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.708599e-001 2.914009e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 8.289599e-001
1.710401e-001
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
9.583444e-001
<ENDHMM>
```

Skaiciaus „nulis“ modelis su 2 papildomom būsenom ir 6 Gauso mišiniais:

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_nulis"
<BEGINHMM>
<NUMSTATES> 7
<STATE> 2
<NUMMIXES> 6
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 3 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 4 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
```


3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 5 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<MIXTURE> 6 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<STATE> 3
<NUMMIXES> 6
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000

2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
 5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-004
 <GCONST> 9.072143e+001
 <MIXTURE> 3 1.432905e-001
 <MEAN> 39
 -2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001 6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001 1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001 2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -6.489957e-002 -1.206123e-001 -3.823178e-003
 <VARIANCE> 39
 3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001 2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000 2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001 5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-004
 <GCONST> 9.072143e+001
 <MIXTURE> 4 1.432905e-001
 <MEAN> 39
 -2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001 6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001 1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001 2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -6.489957e-002 -1.206123e-001 -3.823178e-003
 <VARIANCE> 39
 3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001 2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000 2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001 5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-004
 <GCONST> 9.072143e+001
 <MIXTURE> 5 1.067023e-001
 <MEAN> 39
 -2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001 5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000 2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002 4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002 1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002 4.884647e-003 5.284023e-002 9.121084e-003
 <VARIANCE> 39
 1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000 7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001 6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001 1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-004
 <GCONST> 3.421765e+001
 <MIXTURE> 6 1.067023e-001
 <MEAN> 39
 -2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001 5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000 2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002 4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002 1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002 4.884647e-003 5.284023e-002 9.121084e-003
 <VARIANCE> 39
 1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000 7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001 6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001 1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-004
 <GCONST> 3.421765e+001

```

<STATE> 4
<NUMMIXES> 6
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 3 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 4 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 5 1.067023e-001
<MEAN> 39

```

```

-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<MIXTURE> 6 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<STATE> 5
<NUMMIXES> 6
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 3 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001

```

```

1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 4 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-002 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 5 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<MIXTURE> 6 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<STATE> 6
<NUMMIXES> 6
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002

```

1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 3 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 4 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 5 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-

```

002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<MIXTURE> 6 1.067023e-001
<MEAN> 39
-2.839135e+001 -4.912174e+000 -2.847301e+000 -2.732145e-001 -1.548433e+000 -5.120650e-001
5.122653e-001 -7.041649e-001 -9.774616e-001 1.849443e+000 -1.658283e+000 -1.484717e+000
2.716397e-001 -2.599044e-002 -2.150631e-002 5.144155e-003 4.007615e-003 -1.429657e-002
4.007244e-003 2.336780e-002 6.348765e-003 6.795088e-002 1.643856e-002 -9.604830e-002
1.249331e-002 2.460481e-004 5.125458e-002 -1.381173e-001 4.906508e-002 -6.483736e-002 -
4.409497e-002 -3.097026e-002 -1.066664e-002 -9.287441e-003 1.260006e-002 1.472495e-002
4.884647e-003 5.284023e-002 9.121084e-003
<VARIANCE> 39
1.336169e+000 2.001651e+000 3.931962e+000 5.149906e+000 6.094798e+000 6.064711e+000
7.203526e+000 1.085728e+001 9.393357e+000 7.083497e+000 7.298682e+000 4.829781e+000 1.155231e-
002 1.018538e-001 2.243948e-001 3.758467e-001 4.222659e-001 5.245087e-001 6.029956e-001
6.831428e-001 7.711675e-001 8.427354e-001 8.209996e-001 7.117875e-001 5.063524e-001 4.023646e-
005 3.904774e-002 1.167716e-001 1.126789e-001 1.056295e-001 1.232100e-001 1.325971e-001
1.581802e-001 1.623913e-001 1.730168e-001 1.657937e-001 1.472857e-001 9.034006e-002 2.409866e-
004
<GCONST> 3.421765e+001
<TRANSP> 7
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
0.000000e+000 8.516096e-001 1.483904e-001 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
0.000000e+000 0.000000e+000 9.708599e-001 2.914009e-002 0.000000e+000 0.000000e+000
0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 8.289599e-001 1.710401e-001 0.000000e+000
0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.583444e-001 4.165558e-002
0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.617460e-001
3.825400e-002
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>

```

Vardo „Kju“ modelis be papildomų būsenų ir Gauso mišinių:

```

~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_qju"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.216269e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<TRANSP> 3
0.000000e+000 1.000000e+000 0.000000e+000
0.000000e+000 9.611866e-001 3.881339e-002
0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

Vardo „Kju“ modelis su 2 papildomom būsenom ir 3 Gauso mišiniais:

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_qju"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<NUMMIXES> 3
<MIXTURE> 1 3.567238e-001
<MEAN> 39
-2.208882e+001 -1.679512e+000 1.427382e+001 -3.887221e+000 -1.129381e+001 -9.740973e-001 -
5.993125e+000 -6.069992e+000 -8.078162e+000 -5.747728e+000 -4.729771e+000 -4.851270e+000
8.799027e-001 7.464997e-001 -4.346555e-001 2.249230e-001 7.116883e-001 -1.175757e+000 -
5.300723e-001 -5.673549e-002 5.277705e-002 -9.068720e-001 -2.211461e-001 1.132994e-001 -
4.601816e-001 1.085712e-002 -6.251267e-002 -1.863380e-001 -3.847475e-001 6.109593e-002
1.684407e-001 -1.958551e-001 4.761548e-002 2.500084e-001 1.562560e-001 1.096423e-001
1.112303e-001 1.506525e-001 1.153084e-005
<VARIANCE> 39
1.252672e+001 3.212629e+001 2.317605e+001 3.590876e+001 3.343881e+001 1.744928e+001
2.120577e+001 3.475600e+001 2.275229e+001 1.875223e+001 2.307976e+001 2.354601e+001 5.769957e-
003 6.996154e-001 8.996776e-001 1.332792e+000 1.095654e+000 2.067003e+000 1.886188e+000
1.555288e+000 1.571658e+000 2.375639e+000 1.899912e+000 1.382069e+000 1.312588e+000 4.898355e-
004 1.031535e-001 1.164022e-001 1.461228e-001 1.336237e-001 3.901750e-001 2.887692e-001
3.825051e-001 2.922307e-001 3.503464e-001 3.264502e-001 2.216707e-001 1.973490e-001 7.218438e-
005
<GCONST> 7.365504e+001
<MIXTURE> 2 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<MIXTURE> 3 1.432905e-001
<MEAN> 39
-2.715676e+001 -6.553366e+000 2.502063e+000 -4.613472e+000 -3.385529e+000 1.307645e-001 -
1.771398e-001 -1.709582e+000 5.534387e-002 1.163168e+000 -1.767773e+000 5.518001e-001
6.777486e-001 1.674616e-001 1.847601e-001 1.162947e+000 -4.955001e-001 -3.771032e-001
1.157087e-001 -3.292393e-001 -2.778467e-001 -1.739140e-001 -4.967059e-001 -1.111424e-001
2.986315e-002 5.249029e-002 1.306574e-001 1.279537e-001 1.369752e-001 9.163513e-002 -
1.164889e-001 -3.407399e-002 -1.143174e-001 -5.817175e-002 -1.773376e-001 -1.583198e-001 -
6.489957e-002 -1.206123e-001 -3.823178e-003
<VARIANCE> 39
3.315788e+001 3.113266e+001 3.795301e+001 2.874761e+001 1.767942e+001 1.962831e+001
2.700044e+001 2.335842e+001 1.721119e+001 1.329817e+001 1.188975e+001 1.104182e+001 8.403476e-
002 2.567299e+000 4.028238e+000 2.568973e+000 2.278710e+000 1.946224e+000 1.917244e+000
2.855995e+000 2.102230e+000 2.252542e+000 1.660169e+000 1.460764e+000 1.210396e+000 4.988674e-
003 4.199556e-001 7.151353e-001 3.811444e-001 3.722256e-001 3.785366e-001 3.639985e-001
5.189360e-001 3.886850e-001 4.352961e-001 2.677757e-001 2.748035e-001 2.013021e-001 8.774247e-
004
<GCONST> 9.072143e+001
<STATE> 3
<NUMMIXES> 3
<MIXTURE> 1 3.448086e-001
<MEAN> 39
-1.050859e+001 -3.126868e+000 -5.493579e+000 -7.941609e+000 -6.128641e+000 4.208409e-001 -
1.004891e+001 -1.071766e+001 -1.154191e+001 -2.768691e+000 6.011572e-001 -5.375907e-001
8.368217e-001 5.019748e-001 1.263116e+000 -8.705856e-001 -2.310600e+000 -6.998971e-001
1.332063e+000 2.795660e+000 1.817757e-001 -1.638237e+000 -1.235077e+000 2.365951e-001
1.430265e+000 -1.045969e-002 -1.631255e-001 -1.161267e-001 4.778418e-001 3.978609e-001 -
4.711601e-001 -7.415102e-001 1.961823e-001 9.542721e-001 4.207480e-001 -5.356190e-001 -
6.113527e-001 -1.197513e-001 2.273990e-003
<VARIANCE> 39
```



```

2.544418e+000 1.939818e+001 2.160168e+001 4.533728e+001 1.500395e+001 3.201537e+001
3.793359e+001 3.044609e+001 3.448021e+001 2.742553e+001 5.619424e+001 2.084089e+001 4.857773e-
003 1.202902e-001 3.867133e-001 5.898636e-001 7.097461e-001 1.042447e+000 1.372856e+000
1.084396e+000 1.958249e+000 1.344265e+000 1.715522e+000 1.915564e+000 7.616985e-001 7.879779e-
005 1.289598e-002 6.090763e-002 7.660357e-002 1.103812e-001 1.823199e-001 1.729286e-001
4.937509e-001 2.553721e-001 3.114281e-001 3.618970e-001 2.042797e-001 2.572026e-001 2.119711e-
005
<GCONST> 5.987701e+001
<MIXTURE> 2 1.187025e-001
<MEAN> 39
-1.642425e+001 -8.286872e+000 9.318871e+000 3.959207e+000 -1.553265e+001 -9.591373e+000 -
6.762008e+000 5.194609e-001 -1.064489e+001 -8.140637e+000 -1.500119e+000 -3.201714e+000
9.444200e-001 1.222669e+000 -5.372041e-001 -2.773373e+000 4.549758e-001 1.819621e+000 -
1.760396e+000 -2.682778e+000 8.728188e-001 1.316333e+000 -3.888764e-001 -7.530218e-001
5.606793e-001 -8.559619e-003 4.381256e-002 4.338953e-001 -4.009614e-001 -7.656138e-001
7.438683e-001 9.708534e-001 -5.134764e-001 -9.579337e-001 5.274279e-001 6.169433e-001 -
2.612696e-001 -1.523139e-001 -8.568480e-003
<VARIANCE> 39
6.632800e+000 1.474649e+001 3.950234e+001 4.361419e+001 2.166779e+001 2.802688e+001
3.609658e+001 2.976417e+001 2.802444e+001 2.960429e+001 3.323796e+001 2.442964e+001 3.039280e-
003 2.512416e-001 7.462267e-001 1.058304e+000 1.214052e+000 1.792952e+000 2.740855e+000
2.228956e+000 3.189018e+000 2.139513e+000 1.806955e+000 1.744977e+000 1.001212e+000 3.381036e-
004 2.696447e-002 9.144951e-002 1.438748e-001 2.261326e-001 4.022224e-001 5.237110e-001
6.093573e-001 4.348228e-001 4.788461e-001 3.323811e-001 3.899272e-001 2.515565e-001 5.726003e-
005
<GCONST> 7.452876e+001
<MIXTURE> 3 1.187025e-001
<MEAN> 39
-1.642425e+001 -8.286872e+000 9.318871e+000 3.959207e+000 -1.553265e+001 -9.591373e+000 -
6.762008e+000 5.194609e-001 -1.064489e+001 -8.140637e+000 -1.500119e+000 -3.201714e+000
9.444200e-001 1.222669e+000 -5.372041e-001 -2.773373e+000 4.549758e-001 1.819621e+000 -
1.760396e+000 -2.682778e+000 8.728188e-001 1.316333e+000 -3.888764e-001 -7.530218e-001
5.606793e-001 -8.559619e-003 4.381256e-002 4.338953e-001 -4.009614e-001 -7.656138e-001
7.438683e-001 9.708534e-001 -5.134764e-001 -9.579337e-001 5.274279e-001 6.169433e-001 -
2.612696e-001 -1.523139e-001 -8.568480e-003
<VARIANCE> 39
6.632800e+000 1.474649e+001 3.950234e+001 4.361419e+001 2.166779e+001 2.802688e+001
3.609658e+001 2.976417e+001 2.802444e+001 2.960429e+001 3.323796e+001 2.442964e+001 3.039280e-
003 2.512416e-001 7.462267e-001 1.058304e+000 1.214052e+000 1.792952e+000 2.740855e+000
2.228956e+000 3.189018e+000 2.139513e+000 1.806955e+000 1.744977e+000 1.001212e+000 3.381036e-
004 2.696447e-002 9.144951e-002 1.438748e-001 2.261326e-001 4.022224e-001 5.237110e-001
6.093573e-001 4.348228e-001 4.788461e-001 3.323811e-001 3.899272e-001 2.515565e-001 5.726003e-
005
<GCONST> 7.452876e+001
<STATE> 4
<NUMMIXES> 3
<MIXTURE> 1 5.159193e-001
<MEAN> 39
-9.858402e+000 9.143341e-001 -6.664698e+000 -1.506905e+001 -1.106503e+001 -7.228572e-001 -
2.455084e+000 -4.098136e+000 -1.189528e+001 -6.970455e+000 -5.883337e+000 -1.511328e-001
7.974994e-001 1.344540e-002 2.963305e-001 1.934945e-002 -4.127010e-001 -3.874897e-001 -
4.444826e-001 6.171618e-001 8.756399e-001 5.265664e-001 -1.432426e-001 -4.618731e-001 -
1.886856e-001 -9.456052e-003 -6.593828e-002 -1.411517e-001 3.500742e-002 2.621418e-001
1.773582e-001 -8.415265e-002 -2.741351e-001 -7.697765e-002 1.373615e-001 2.161472e-001
1.052800e-001 -1.203258e-001 -1.831135e-003
<VARIANCE> 39
1.838048e+000 1.998083e+001 1.996949e+001 4.219237e+001 1.462290e+001 3.086727e+001
2.511192e+001 2.481679e+001 3.624630e+001 2.351124e+001 5.225810e+001 2.736989e+001 6.640933e-
003 8.921771e-002 4.716771e-001 2.893594e-001 7.078879e-001 7.779482e-001 8.291999e-001
1.523142e+000 1.273484e+000 9.768979e-001 1.312346e+000 1.016774e+000 8.379387e-001 1.413979e-
004 1.622649e-002 4.561487e-002 6.313898e-002 9.867713e-002 1.239830e-001 1.504877e-001
2.626512e-001 2.079266e-001 1.717156e-001 2.244640e-001 1.725297e-001 1.780109e-001 1.961695e-
005
<GCONST> 5.350391e+001
<MIXTURE> 2 1.031969e-001
<MEAN> 39
-1.074286e+001 6.781657e-001 -6.270209e+000 -1.270227e+001 -1.009062e+001 -2.683011e+000 -
6.366626e-001 -6.338654e-001 -7.404989e+000 -5.365322e+000 -4.714336e+000 -1.583874e+000
6.788688e-001 -6.934912e-001 -1.713974e-001 -2.114369e-002 1.575354e+000 1.129783e+000
3.134390e-001 4.704967e-001 -3.364090e-002 2.702072e-001 2.883338e-001 4.882534e-002 -
3.881102e-001 -4.270354e-002 -2.817715e-001 -4.043951e-002 4.167823e-002 3.001076e-001
3.410932e-001 2.986816e-001 1.233862e-001 -1.457110e-001 -4.485578e-002 4.036235e-002 -
7.327753e-002 -1.734768e-002 -8.001210e-003
<VARIANCE> 39
2.499485e+000 2.495955e+001 2.299720e+001 4.636172e+001 1.458770e+001 2.693869e+001
2.339853e+001 3.933677e+001 3.554038e+001 2.081016e+001 3.251838e+001 2.469900e+001 1.028209e-
002 3.652351e-001 7.628086e-001 7.551258e-001 1.342311e+000 2.011446e+000 1.612413e+000

```

```

2.270261e+000 2.783773e+000 1.397477e+000 2.606770e+000 2.260560e+000 1.479250e+000 4.181588e-
004 4.984400e-002 1.264164e-001 1.399508e-001 2.027602e-001 1.851461e-001 1.836706e-001
3.090564e-001 3.551123e-001 2.703148e-001 3.368646e-001 2.977968e-001 2.208835e-001 4.485863e-
005
<GCONST> 7.142472e+001
<MIXTURE> 3 1.031969e-001
<MEAN> 39
-1.074286e+001 6.781657e-001 -6.270209e+000 -1.270227e+001 -1.009062e+001 -2.683011e+000 -
6.366626e-001 -6.338654e-001 -7.404989e+000 -5.365322e+000 -4.714336e+000 -1.583874e+000
6.788688e-001 -6.934912e-001 -1.713974e-001 -2.114369e-002 1.575354e+000 1.129783e+000
3.134390e-001 4.704967e-001 -3.364090e-002 2.702072e-001 2.883338e-001 4.882534e-002 -
3.881102e-001 -4.270354e-002 -2.817715e-001 -4.043951e-002 4.167823e-002 3.001076e-001
3.410932e-001 2.986816e-001 1.233862e-001 -1.457110e-001 -4.485578e-002 4.036235e-002 -
7.327753e-002 -1.734768e-002 -8.001210e-003
<VARIANCE> 39
2.499485e+000 2.495955e+001 2.299720e+001 4.636172e+001 1.458770e+001 2.693869e+001
2.339853e+001 3.933677e+001 3.554038e+001 2.081016e+001 3.251838e+001 2.469900e+001 1.028209e-
002 3.652351e-001 7.628086e-001 7.551258e-001 1.342311e+000 2.011446e+000 1.612413e+000
2.270261e+000 2.783773e+000 1.397477e+000 2.606770e+000 2.260560e+000 1.479250e+000 4.181588e-
004 4.984400e-002 1.264164e-001 1.399508e-001 2.027602e-001 1.851461e-001 1.836706e-001
3.090564e-001 3.551123e-001 2.703148e-001 3.368646e-001 2.977968e-001 2.208835e-001 4.485863e-
005
<GCONST> 7.142472e+001
<TRANSP> 5
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.611866e-001 3.881339e-002 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.322118e-001 6.778825e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
9.011618e-001
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

10 PRIEDAS. Vardų komandų atpažinimo tikslumas gautas su ispanišku atpažintuvu

3.1.1 lentelė. Vardų atpažinimo tikslumo rezultatai su pradine gramatika

Raidė	Vardas	Atpažinimas, %	
		„Iveta1“	„Default_Neapmokytas1“
A	Austėja	99,5	100
B	Boleslovas	97	96,5
C	Cecilija	99	99
D	Donatas	98,5	97
E	Eimantas	94,5	94,5
F	Fausta	99	100
G	Gražvydas	98	99
H	Hansas	100	100
I	Izaokas	96,5	98
J	Jonas	96,5	97
K	Karolis	100	100
L	Laima	99,5	99
M	Martynas	97,5	98,5
N	Nojus	96,5	94,5
O	Oskaras	100	100
P	Patrikas	100	98,5
Q	Kju	85,5	80
R	Ričardas	98	93
S	Sandra	99,5	98,5
T	Teodoras	100	99,5
U	Ulijona	88	88,5
V	Vacys	99	99
W	Vašington	97,5	99
X	Iksas	97,5	99,5
Y	Ygrekas	93,5	92
Z	Zacharijus	90	86
Vidutinis atpažinimo tikslumas		96,9±1,46	96,4±1,9

3.1.2 lentelė. Vardų atpažinimo tikslumo rezultatai su papildyta pradine gramatika

Raidė	Vardas	Atpažinimas, %	
		„Iveta1“	„Default_Neapmokytas1“
A	Austėja	99,5	100
B	Boleslovas	97	96,5
C	Cecilija	99	99
D	Donatas	98,5	97
E	Eimantas	94,5	94,5
F	Fausta	99	100
G	Gražvydas	98	99
H	Hansas	100	100
I	Izaokas	96,5	98
J	Jonas	96,5	97
K	Karolis	100	100
L	Laima	99,5	99
M	Martynas	97,5	98,5
N	Nojus	96,5	94,5
O	Oskaras	100	100
P	Patrikas	100	98,5
Q	Kju	82,5	77,5
R	Ričardas	98	93
S	Sandra	99,5	98,5
T	Teodoras	100	99,5
U	Ulijona	86,5	87,5
V	Vacys	99	99
W	Vašington	97,5	99
X	Iksas	97,5	99,5
Y	Ygrekas	93	93,5
Z	Zacharijus	99,5	99
Vidutinis atpažinimo tikslumas		97,1±1,6	96,8±1,88

3.1.4 lentelė. Vardų atpažinimo tikslumo rezultatai su papildyta UPS pradine gramatika

Raidė	Vardas	Atpažinimas, %	
		„Iveta1“	„Default_Neapmokytas1“
A	Austėja	99,5	100
B	Boleslovas	97	96,5
C	Cecilija	99	99
D	Donatas	98,5	97
E	Eimantas	94	2,5
F	Fausta	99	100
G	Gražvydas	98	99
H	Hansas	100	100
I	Izaokas	96,5	98
J	Jonas	96,5	97
K	Karolis	100	100
L	Laima	99,5	99
M	Martynas	97,5	98,5
N	Nojus	96,5	94,5
O	Oskaras	100	100
P	Patrikas	100	98,5
Q	Kju	83,5	77,5
R	Ričardas	98	93
S	Sandra	99,5	98,5
T	Teodoras	100	99,5
U	Ulijona	86,5	84
V	Vacys	99	99
W	Vašington	97,5	99
X	Iksas	97,5	99,5
Y	Ygrekas	83	92
Z	Zacharijus	6	94
Vidutinis atpažinimo tikslumas		93,2±6,8	92,9±7,1

11 PRIEDAS. Skaičių atpažinimo tikslumo priklausomybė nuo būsenų ir Gauso mišinių skaičiaus

3.4.4 lentelė. Atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus

Atpažinimo tikslumo priklausomybė nuo būsenų skaičius, %				
Skaičius	Būsenų skaičius atitinka raidžių sk.			
	+2 būsenos	+4 būsenos	+10 būsenų	+16 būsenų
Nulis	69,2	100	96,7	87,5
Vienas	83,3	92,5	99,2	97,5
Du	5,8	65	99,2	98,3
Trys	44,2	90,8	94,2	95
Keturi	98,3	97,5	96,7	95,8
Penki	73,3	98,3	97,5	95,8
Šeši	84,2	95,8	95,8	95,8
Septyni	100	100	100	99,2
Aštuoni	96,7	98,3	95	96,7
Devyni	90	93,3	92,5	95
Vidurkis, %	74,5±18,28	93,2±6,44	96,7±1,48	95,7±1,98

3.4.5 lentelė. Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus

Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičius, %				
Skaičius	Būsenų skaičius atitinka raidžių sk. +2			
	2 mišiniai	3 mišiniai	6 mišiniai	10 mišinių
Nulis	86,7	100	100	99,2
Vienas	92,5	100	100	100
Du	11,7	75	97,5	94,2
Trys	95	95	95	95
Keturi	97,5	100	100	100
Penki	92,5	100	100	100
Šeši	98,3	100	100	100
Septyni	100	100	100	100
Aštuoni	90,8	100	100	100
Devyni	90,8	99,2	100	99,2
Vidurkis, %	85,6±14,35	96,9±3,12	99,3±0,66	98,8±1,18

3.4.6 lentelė. Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus

Atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičius, %				
Skaičius	Būsenų skaičius atitinka raidžių sk. +4			
	2 mišiniai	3 mišiniai	6 mišiniai	10 mišinių
Nulis	100	100	99,2	99,2
Vienas	95	99,2	97,5	99,2
Du	97,5	98,3	98,3	98,3
Trys	94,2	93,3	93,3	94,2
Keturi	95	95	98,3	96,7
Penki	94,2	99,2	89,2	99,2
Šeši	98,3	100	100	100
Septyni	100	100	100	100
Aštuoni	88,3	100	100	100
Devyni	90,8	98,3	90	93,3
Vidurkis, %	95,3±2,35	98,3±1,45	96,6±2,59	98±1,53

12 PRIEDAS. „Balsu išstartų kodų atpažinimo sistema“

3.9.1 Pilnas demonstracinės programos kodas

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Speech.Recognition;
using System.Speech.Recognition.SrgsGrammar;
using System.Diagnostics;
using System.IO;

namespace Demo
{
    public partial class Demo : Form
    {
        private SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();
        string kodas = "", komanda = "", kodas_old = "";
        int num = 0, n = 10;
        string [] lig_kod = new string[100];
        string [] lig_pav = new string[100];

        public Demo()
        {
            InitializeComponent();
            label2.Text = "Spauskite Vardai";
            lig_kod[0] = "E0123"; lig_pav[0] = "Ausies uždegimas";
            lig_kod[1] = "S0031"; lig_pav[1] = "Nosies paviršinis sužalojimas, nubrozdinimas";
            lig_kod[2] = "C9201"; lig_pav[2] = "?min? mieloidin? leukemija, kai yra remisija";
            lig_kod[3] = "F2320"; lig_pav[3] = "?minis panašus ? šizofrenij? psichozinis sutrikimas";
            lig_kod[4] = "I2511"; lig_pav[4] = "Aterosklerozin? širdies liga, vainikin?s arterijos pažeidimas";
            lig_kod[5] = "M0517"; lig_pav[5] = "Reumatoidin? plau?i? liga, ?iurna ir p?da";
            lig_kod[6] = "K8041"; lig_pav[6] = "Tulžies latak? akmuo, su cholecistitu, su obstrukcija";
            lig_kod[7] = "C9201"; lig_pav[7] = "?min? mieloidin? leukemija, kai yra remisija";
        }

        private void Start_Click(object sender, EventArgs e)
        {
            recognizer.SetInputToDefaultAudioDevice();
            label2.Text = "Sakykite";
            recognizer.SpeechRecognized += new EventHandler<SpeechRecognizedEventArgs>(recognizer_SpeechRecognized);
            recognizer.RecognizeAsync(RecognizeMode.Multiple);
        }

        void recognizer_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
        {
            num++;
            string word = e.Result.Text.ToString();
            // textBox1.Text = word;
            komanda = e.Result.Semantics.Value.ToString();
            kodas_old = kodas; kodas = kodas + komanda;
            if (num == 1) { textBox1.Text = komanda; label2.Text = "Spauskite Skaiciai"; }
            if (num == 2) textBox2.Text = komanda;
            if (num == 3) textBox3.Text = komanda;
            if (num == 4) textBox4.Text = komanda;
            if (num == 5) { textBox5.Text = komanda; label2.Text = "Spauskite Liga"; }
        }
    }
}
```



```

// SendKeys.Send(komanda);
}

private void Vardai_Click(object sender, EventArgs e)
{
    Grammar gramatika = new Grammar("vardai.grxml", "Rule");
    recognizer.LoadGrammar(gramatika);
    label2.Text = "Spauskite Start";
    richTextBox1.Clear();
    richTextBox1.LoadFile("vardai.rtf");
}

private void Stop_Click(object sender, EventArgs e)
{
// recognizer.RecognizeAsyncStop();
    richTextBox1.Clear();
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
    textBox6.Clear();
    label2.Text = "Pasirinkite s?raš?";
    recognizer.UnloadAllGrammars();
}

private void Skaiciai_Click(object sender, EventArgs e)
{
    Grammar gramatika = new Grammar("skaiciai.grxml", "Rule");
    recognizer.LoadGrammar(gramatika);
    label2.Text = "Sakykite";
    richTextBox1.Clear();
    richTextBox1.LoadFile("skaiciai.rtf");
}

private void Liga_Click(object sender, EventArgs e)
{
    label2.Text = "Spauskite Stop";
    for (int i = 0; i < n; i++)
    {
        if (kodas == lig_kod[i])
        {
            textBox6.Text = lig_pav[i];
            break;
        }
        textBox6.Text = "Liga neatpažinta";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    switch (num)
    {
        case 1: textBox1.Clear(); label2.Text = "Sakykite";
            break;
        case 2: textBox2.Clear();
            break;
    }
}

```

```

        case 3: textBox3.Clear();
        break;
        case 4: textBox4.Clear();
        break;
        case 5: textBox5.Clear(); label2.Text = "Sakykite";
        break;
    }
    num--; kodas = kodas_old;
}

}
}

```

3.9.2 Demonstracinės programos esminių fragmentų iš programinio kodo aprašymas

Pirmiausiai esamos bibliotekos papildomos naujomis:

➤ *using System.Speech.Recognition;*

using System.Speech.Recognition.SrgsGrammar;

Programos kodo veikimo trumpas paaiškinimas:

➤ *namespace Demo {}* – sukuriamą programą, kurios aplanko pavadinimas „Demo“.

Pačios programos pavadinimas „Ligų kodų atpažinimo programa“

➤ *private SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();* - sukuriamas kalbos atpažinimo variklis – atpažintuvas;

➤ *InitializeComponent();* - komanda, aktyvuoja programos komponentus.

label2.Text = "Spauskite Vardai";

lig_kod[0] = "E0123"; lig_pav[0] = "Ausies uždegimas";

lig_kod[1] = "S0031"; lig_pav[1] = "Nosies paviršinis sužalojimas, nubrozdinimas";

...

Priskiriame gramatikas, kurios bus naudojama atpažinimo metu. Naudojama pirminė gramatika, kuri įnešama į programos katalogą – „debug“.

Vardų gramatikos įtraukimas aprašomas tokiu kodu:

➤ *private void Vardai_Click(object sender, EventArgs e)*

{

Grammar gramatika = new Grammar("vardai.grxml", "Rule");

recognizer.LoadGrammar(gramatika);

label2.Text = "Spauskite Start";

richTextBox1.Clear();

richTextBox1.LoadFile("vardai.rtf");

}

Analogiškai aprašoma ir skaičių gramatikos įterpimas į programą.

Dar vienas svarbus šios programos kodo fragmentas:

➤ *private void Start_Click(object sender, EventArgs e)* – nurodoma, kokiam mygtukui aprašinėjama komanda, kurią programa turės vykdyti;

Taip aprašomi visi programoje esami mygtukai ir aprašomos jų funkcijos.

➤ `recognizer.SetInputToDefaultAudioDevice();` - nurodoma, kad darbas vyks su mikrofonu, t.y. komanda atpažinimui bus paduodama iš mikrofono, o ne iš failo;

➤ `Label2.Text = "Sakykite";` - laukelis, kuriame išvedamas pranešimas, ką vartotojui daryti po to, kai paspaudžiamas kai įvykdomas prašymas, šiuo atveju tai paliepimas „Sakykite“;

➤ `recognizer.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(recognizer_SpeechRecognized);` - atpažintuvas sukuria komandos apdorojimo metodą;

➤ `recognizer.RecognizeAsync(RecognizeMode.Multiple);` - atpažintuvas paleidžiamas asinchroniniame darbo režime. Asinchroninis darbo režimas pasirenkamas, nes jam neturi įtakos kitų programų veikimas tuo pačiu metu, o sinchroniniam – reikalinga priežiūra tam, kad veiktų tik viena programa tuo metu, nes jos veikimui efektyvumą gali trikdyti aplinkiniai veiksniai.

Kai įvykdomas komandos atpažinimas programa vykdo tokią programinio kodo dalį:

➤ `void recognizer_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)`

➤ `string word = e.Result.Text.ToString();` - išveda rezultatą po komandos atpažinimo į teksto lauką;

➤ `komanda = e.Result.Semantics.Value.ToString();` - nurodoma, kad į ekraną išvestų iš gramatikos failo semantinę reikšmę, o ne patį žodį, kurį atpažino („tagas“ gramatikos faile);

➤ `textBox1.Text = komanda;` - atpažinta komanda atvaizduojama būtent šioje eilutėje/laukelyje.

Programos kodo fragmentas yra mygtuko „Baigti“ veikimo aprašymas. Jis aprašomas tokiu kodu

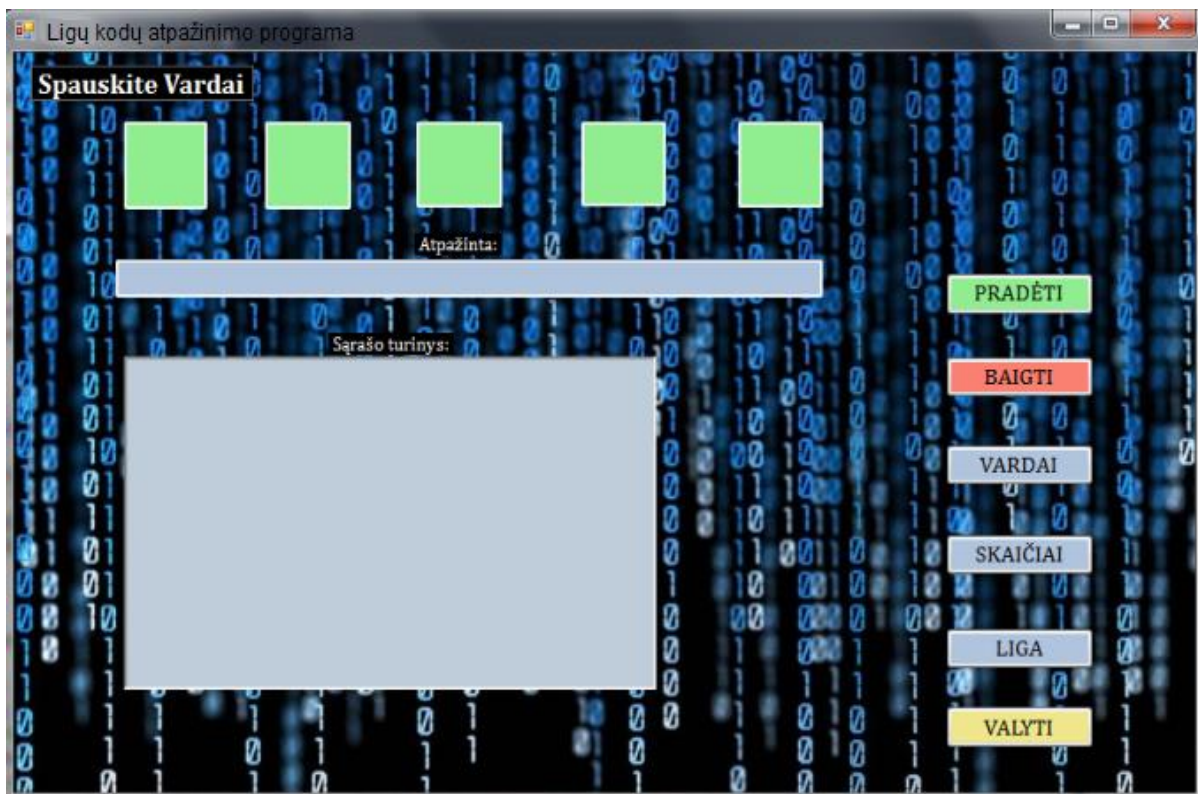
```
➤ label2.Text = "Spauskite Baigti";  
private void button1_Click(object sender, EventArgs e)  
private void Stop_Click(object sender, EventArgs e)  
{  
    richTextBox1.Clear();  
    textBox1.Clear();  
    textBox2.Clear();  
    textBox3.Clear();  
    textBox4.Clear();  
    textBox5.Clear();  
    textBox6.Clear();  
    label2.Text = "Pasirinkite sąrašą";  
    recognizer.UnloadAllGrammars();  
}
```

Paspaudus šį mygtuką išvalomas sąrašo laukas, nebenaudojamos darbo metu naudotos gramatikos, išvedamas pranešimas „Pasirinkite sąrašą“ ir programa yra pasiruošusi dirbti iš naujo.

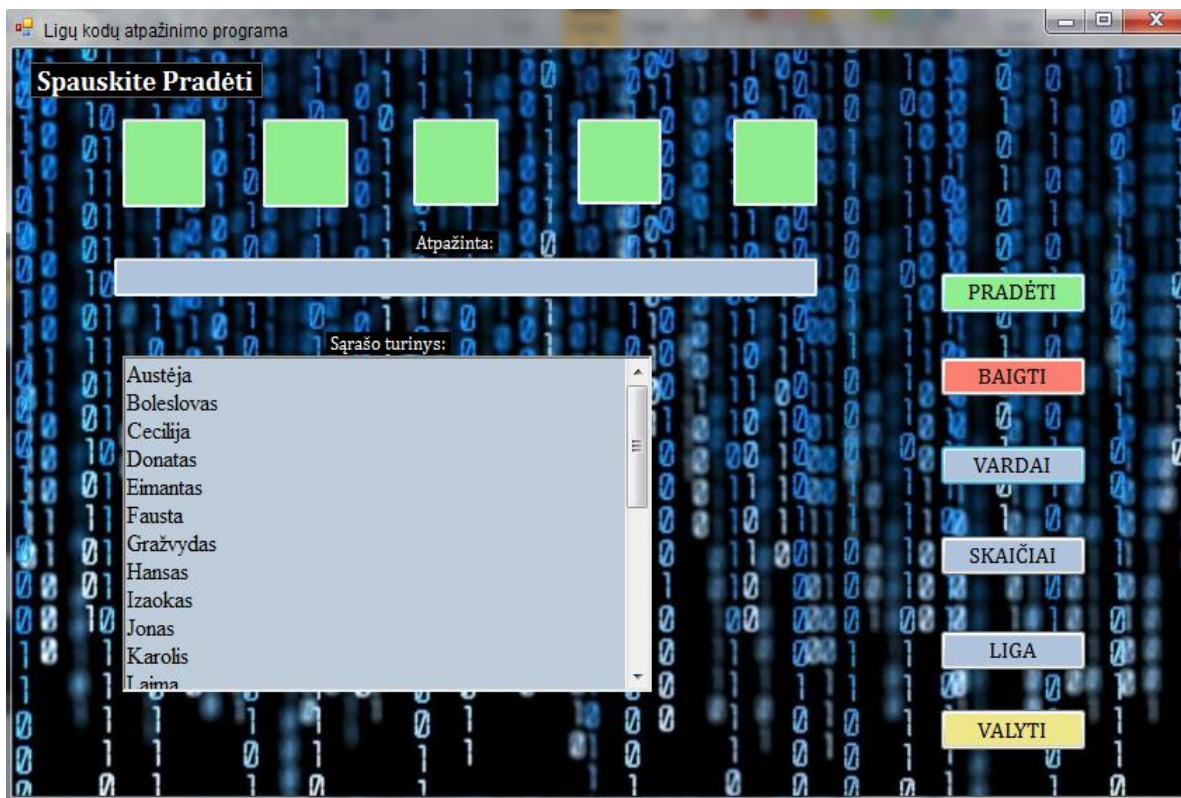
Šnekamosios kalbos atpažinimo programos – „Ligų kodų atpažinimo programa“-pradinio lango vaizdas pateiktas 3.9.1 paveiksle. Vartotojui siūloma pasirinkti norimų atpažinti balso komandų sąrašą (*Vardai arba Skaičiai*). Pasirinkus „Vardų“ parinktį – programos lange

atvaizduojamas vardų komandų sąrašas (žr. 3.9.2 pav.). Pasirinkus parinktį „Pradėti“ atsiranda užrašas „Sakykite“, o į mikrofoną ištarus vardo komandą – į ekraną išvedamas atpažinto vardo pirmosios raidės simbolis. Kadangi sisteminis ligų sąrašas TLK-10-AM koduojamas pirmiausia Raidės simboliu, o paskui skaičių simboliais, todėl kai įvyksta raidės simbolio atpažinimas programa siūlo – „Spauskite Skaičiai“. Paspaudus programos siūlomą parinktį „Skaičiai“ sąrašo turinio lauke atsiranda skaičių sąrašas ir atpažintuvas yra pasiruošęs atpažinti diktuojamas skaičių komandas, kurios, kai yra atpažintos atsiranda iš eilės sekančiuose laukuose po raidės simbolio. Kai subedamas pilnas ligos kodas, programa siūlo „Spauskite Liga“, pasirinkus siūlomą parinktį, Programos lango srityje - „Atpažinta“, atsiranda atpažintam kodui pagal TLK-10-AM priskirta liga (žr. 3.9.3 pav.). Jei kodas neteisingas, toje pačioje programos srityje atsiranda užrašas „Liga neatpažinta“. Programa dar turi papildomą parinktį „valyti“, kuria pasirinkus pašalinimas paskutinis atpažintas simbolis kode, ir komanda sakoma iš naujo, taip suklydus galima pasitaisyti, ir nereikia viso kodo diktuoti iš naujo.

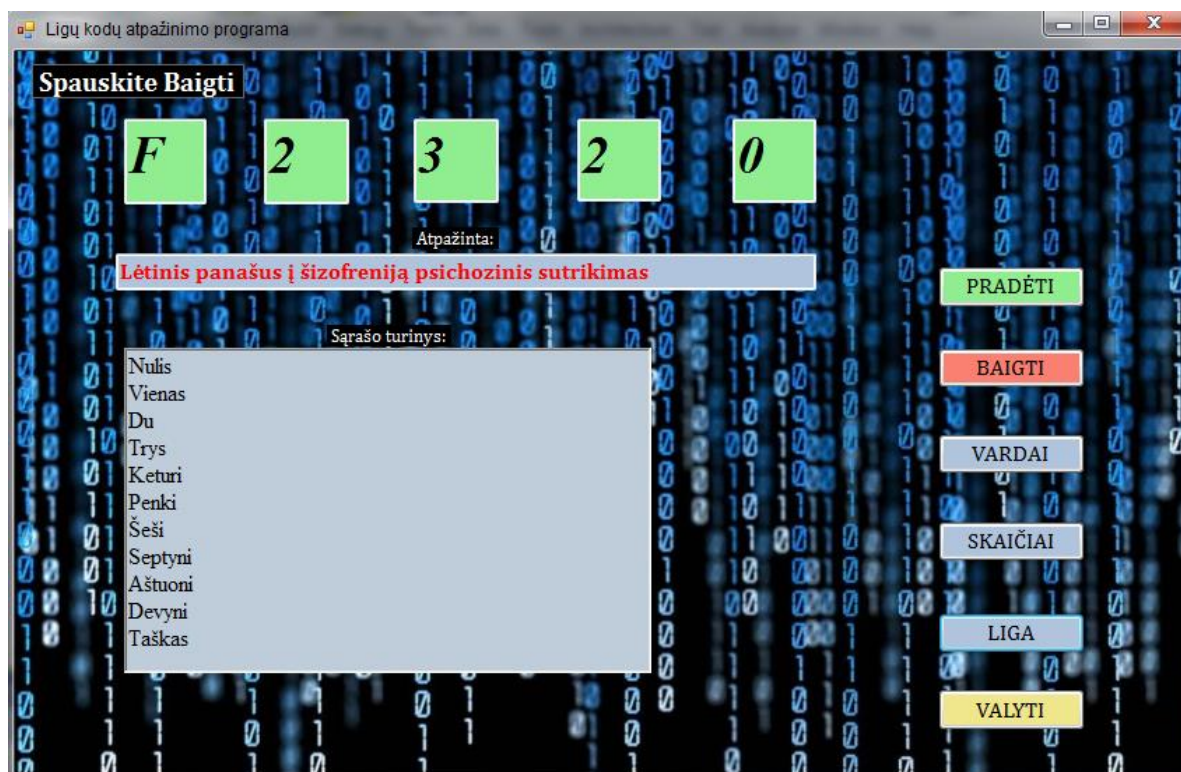
Gavus reikiamą informaciją spaudžiamas mygtukas „Baigti“, ir programa iš naujo siūlo rinktis sąrašą ir visa išvardinta veiksmų seka kartojama iš naujo. Ši programa veikia su individualiai apmokytu ispanų kalbos atpažinimo profiliu. Profilį galima pasirinkti laisvai, šiai programai buvo naudojamas baigiamojo darbo vadovo balsu apmokytas profilis „Kastytis“.



3.9.1 pav. Pradinis programos vaizdas



3.9.2 pav. Sąrašo pasirinkimas



3.9.3 pav. Atpažintas ligos kodas

3.9.3 Programoje naudojamų vardų komandų sąrašas

Austėja	Nojus
Boleslovas	Oskaras
Cecilija	Patrikas
Donatas	Kju
Eimantas	Ričardas
Fausta	Sandra
Gražvydas	Teodoras
Hansas	Ulijona
Izaokas	Vacys
Jonas	Wašington
Karolis	Iksas
Laima	Ygrekas
Martynas	Zacharijus

3.9.4 Programoje naudojamų skaičių komandų sąrašas

Nulis	Šeši
Vienas	Septyni
Du	Aštuoni
Trys	Devyni
Keturi	
Penki	

3.9.5 Programoje naudojamų vardų gramatikos

<one-of>	</one-of>
<item>	<tag>\$_value = "E"</tag>
<one-of>	</item>
<item>ahwstehyeh</item>	<item>
<item>austeja</item>	<one-of>
</one-of>	<item>fahwstah</item>
<tag>\$_value = "A"</tag>	<item>fausta</item>
</item>	</one-of>
<item>	<tag>\$_value = "F"</tag>
<one-of>	</item>
<item>baolehslaovahs</item>	<item>
<item>boleslovas</item>	<one-of>
</one-of>	<item>graazhviydahs</item>
<tag>\$_value = "B"</tag>	<item>grazvydas</item>
</item>	</one-of>
<item>	<tag>\$_value = "G"</tag>
<one-of>	</item>
<item>tsehtsihlihyeh</item>	<item>
<item>cecilija</item>	<one-of>
</one-of>	<item>haansahs</item>
<tag>\$_value = "C"</tag>	<item>hansas</item>
</item>	</one-of>
<item>	<tag>\$_value = "H"</tag>
<one-of>	</item>
<item>daonaatahs</item>	<item>
<item>donatas</item>	<one-of>
</one-of>	<item>ihzahaokahs</item>
<tag>\$_value = "D"</tag>	<item>izaokas</item>
</item>	</one-of>
<item>	<tag>\$_value = "I"</tag>
<one-of>	</item>
<item>ehiymahntahs</item>	<item>
<item>eimantas</item>	<one-of>

```

<item>yaonahs</item>
<item>jonas</item>
</one-of>
<tag>$. _value = "J"</tag>
</item>
<item>
<one-of>
<item>kahraolihs</item>
<item>karolis</item>
</one-of>
<tag>$. _value = "K"</tag>
</item>
<item>
<one-of>
<item>laaiymah</item>
<item>laima</item>
</one-of>
<tag>$. _value = "L"</tag>
</item>
<item>
<one-of>
<item>mahrtiynahs</item>
<item>martynas</item>
</one-of>
<tag>$. _value = "M"</tag>
</item>
<item>
<one-of>
<item>naoyuhs</item>
<item>nojus</item>
</one-of>
<tag>$. _value = "N"</tag>
</item>
<item>
<one-of>
<item>aoskahrahs</item>
<item>oskaras</item>
</one-of>
<tag>$. _value = "O"</tag>
</item>
<item>
<one-of>
<item>paatrihkahs</item>
<item>patrikas</item>
</one-of>
<tag>$. _value = "P"</tag>
</item>
<item>
<one-of>
<item>kyuh</item>
<item>kju</item>
</one-of>
<tag>$. _value = "K"</tag>
</item>
<item>
<one-of>
<item>rihchahrdahs</item>
<item>ricardas</item>
</one-of>
<tag>$. _value = "R"</tag>

```

```

</item>
<item>
<one-of>
<item>saandrah</item>
<item>sandra</item>
</one-of>
<tag>$. _value = "S"</tag>
</item>
<item>
<one-of>
<item>tehaodaorahs</item>
<item>teodoras</item>
</one-of>
<tag>$. _value = "T"</tag>
</item>
<item>
<one-of>
<item>uhlihyaonah</item>
<item>ulijona</item>
</one-of>
<tag>$. _value = "U"</tag>
</item>
<item>
<one-of>
<item>vahtsiys</item>
<item>vacys</item>
</one-of>
<tag>$. _value = "V"</tag>
</item>
<item>
<one-of>
<item>vahshihngktaon</item>
<item>vashington</item>
<item>vohshihngktaon</item>
<item>voshington</item>
</one-of>
<tag>$. _value = "W"</tag>
</item>
<item>
<one-of>
<item>ihksahs</item>
<item>iksas</item>
</one-of>
<tag>$. _value = "X"</tag>
</item>
<item>
<one-of>
<item>iygrehkahs</item>
<item>ygrekas</item>
</one-of>
<tag>$. _value = "Y"</tag>
</item>
<item>
<one-of>
<item>zahthaarihyuhs</item>
<item>zacharijus</item>
</one-of>
<tag>$. _value = "Z"</tag>
</item>
</one-of>

```

3.9.6 Programoje naudojamų skaičių gramatikos

```

<one-of>
<item>
<one-of>
<item>nulejs</item>
<item>mulis</item>

```

```

<item>nuhlejs</item>
<item>nuhlejs</item>
</one-of>
<tag>$. _value = "0"</tag>
</item>

```

```

<item>
<one-of>
<item>vianas</item>
<item>vihehnahs</item>
</one-of>
<tag>$. _value = "1"</tag>
</item>
<item>
<one-of>
<item>doo</item>
<item>dou</item>
<item>douh</item>
<item>duo</item>
<item>dua</item>
</one-of>
<tag>$. _value = "2"</tag>
</item>
<item>
<one-of>
<item>trys</item>
<item>trris</item>
<item>trriis</item>
<item>trriis</item>
<item>trys</item>
<item>trys</item>
<item>tris</item>
</one-of>
<tag>$. _value = "3"</tag>
</item>
<item>
<one-of>
<item>kewturih</item>
<item>keaturih</item>
<item>keaturii</item>
<item>keturej</item>
<item>kevturii</item>
</one-of>
<tag>$. _value = "4"</tag>
</item>
<item>
<one-of>

```

```

<item>penkii</item>
<item>penkiih</item>
<item>penki</item>
</one-of>
<tag>$. _value = "5"</tag>
</item>
<item>
<one-of>
<item>sheshii</item>
<item>chechii</item>
</one-of>
<tag>$. _value = "6"</tag>
</item>
<item>
<one-of>
<item>septinii</item>
<item>septiinii</item>
<item>ceptinii</item>
<item>septinej</item>
</one-of>
<tag>$. _value = "7"</tag>
</item>
<item>
<one-of>
<item>astuonii</item>
<item>ashtuhohnii</item>
<item>ashtuhohniih</item>
</one-of>
<tag>$. _value = "8"</tag>
</item>
<item>
<one-of>
<item>deviinii</item>
<item>debini</item>
<item>dewini</item>
<item>deviiniih</item>
<item>debjinese</item>
</one-of>
<tag>$. _value = "9"</tag>
</item>
</one-of>

```


INVESTIGATION OF DISEASE CODES RECOGNITION ACCURACY

Gintarė BARTIŠIŪTĖ, Gintarė PAŠKAUSKAITĖ, Kastytis RATKEVIČIUS
Kaunas University of Technology, Lithuania

Abstract: Paper deals with the recognition accuracy of disease codes. It is impossible so far to recognize about 15000 various diseases, but each disease can be identified by its code, consisting of one letter and some digits. The appropriate Lithuanian names were selected for each letter and their recognition accuracy together with Lithuanian digit names recognition accuracy were investigated. Two types of recognizers were used for the above mentioned purpose: a) the adapted foreign language recognizer used in Windows'7 or Windows'8 operation systems; b) Lithuanian speech recognizer based on HMMs (Hidden Markov Models) modeling. The efficiency of both recognizers was evaluated using the speech corpora.

Keywords: isolated words recognition, hybrid recognizer, recognition accuracy.

1. Introduction

People are interested in being able to access the information when they are on the move—anytime, anywhere, and in their native language. A promising solution to this problem, especially for small, handheld devices where a conventional keyboard and mouse can be impractical, is to impart human-like capabilities onto machines so that they can speak and hear, just like the users with whom they need to interact. Spoken language is attractive because it is the most natural, efficient, flexible, and inexpensive means of communication among humans.

During last 15 years speech technologies became integral part of industrial applications of information technologies in various areas

human activities. The examples of successes in the deployment of spoken language systems are conversational weather information system JUPITER [1] and transportation information system “Let’s Go” [2]. JUPITER is a conversational interface that allows users to obtain worldwide weather forecast information over the telephone using spoken dialogue. During only a two year period since coming on line in May 1997, JUPITER has received, via a toll-free number in North America, over 30 000 calls (totaling over 180 000 utterances), mostly from naive users. “Let’s Go” gives bus schedule information to the Pittsburgh population at hours when the Port Authority phones are not manned by operators (7pm to 6am on weekdays and 6pm to 8am on weekends). It collected about 20,000 calls in the year [2].

The project “Hybrid recognition technology for voice interface INFOBALSAS” was ended in Lithuania in 2013. The main goal of the project was to develop hybrid voice commands recognition technology and implement it in the first practical informative service using recognition of Lithuanian voice commands. The informative service is oriented to the workplace of physician/ pharmacist and seeks to support/ to speed up the search of the information in pharmaceutical data base. The possibilities to fill the internet forms by voice were investigated during the project too. It has been decided that voice user interface must be able to deal with the names of the drugs, with the names of the illnesses and diseases as well as with some other words and phrases often met in medical workers practice [3].

The developed Lithuanian medical information system is able to recognize the names of most often met disease names, the most popular drug

names and most often met complaints in medical practice. The total number of voice commands implemented in the system is about 1000, which is probably the biggest number of Lithuanian voice commands implemented in practically oriented Lithuanian speech recognition system so far. The recognition accuracy of 1000 voice commands is better than 98% [3].

The main point of a hybrid recognizer is a parallel usage of two different recognizers expecting that at least one of the recognizers will give the right result. Another important factor – quite good voice commands end-pointing using, for example, Windows'7 OS recognizer. A usage only of a Lithuanian recognizer would lead to additional solving of this problem. So Microsoft Spanish speech engine as the foreign language speech engine for the adaptation purposes as one kind of recognizer was selected and the CD-HMM based speech recognizer was used as the preferred selection for Lithuanian speech recognizer [3].

The main problem of diseases recognition was defined during the project INFOBALSAS – the list of diseases, **approved by the Ministry of Health, contains 14179 diseases and disorders. It is composed of more than 88000 lexical tokens (not all are of medical origin) and has 10955 unique lexical types. It may be surprising that specific medical terms are used much less often than general terms (e.g. switches).** 5991 lexical types cover 75 percent of the whole list. So far we are not able well recognize such big list of diseases [4].

The solution of the problem of diseases recognition is to use the codes of diseases and to recognize the letters and several digit names.

2. International Classification of Diseases

The International Classification of Diseases (ICD) is the standard diagnostic tool for epidemiology, health management and clinical purposes [4]. A classification of diseases can be defined as a system of categories to which morbid entities are assigned according to established criteria. The purpose of the ICD is

to permit the systematic recording analysis, interpretation and comparison of mortality and morbidity data collected in different countries or areas and at different times. The ICD is used to translate diagnoses of diseases and other health problems from words into an alphanumeric code, which permits easy storage, retrieval and analysis of the data.

The basic ICD is a single coded list of three-character categories, each of which can be further divided into up to 10 four-character subcategories. In place of the purely numeric coding system of previous revisions, the Tenth Revision (ICD-10) uses an alphanumeric code with a letter in the first position and a number in the second, third and fourth positions. The fourth character follows a decimal point. Possible code numbers therefore range from A00.0 to Z99.9. Codes U00–U49 are to be used for the provisional assignment of new diseases of uncertain etiology. Codes U50–U99 may be used in research, e.g. when testing an alternative sub-classification for a special project. The fifth and subsequent character levels are usually sub-classifications along a different axis from the fourth character [4].

3. Speech corpora

For our experiments with Lithuanian digit names, speech corpus SKAIC30 was collected, which contain utterances from 30 different speakers (7 speaker-men, 23 speaker-women). Each speaker pronounced 10 digit names 20 times at sampling rate 16 kHz. One part of this corpus (utterances from 20 speakers) was used for selecting the transcriptions of Lithuanian digit names, suitable for the Spanish recognizer.

The features of speech corpora, used for letters recognition, are presented in the table 1. Speech corpora LETTERS and NATO were used to verify their possibilities of letters recognition. Speech corpus LETTERS consists of Lithuanian letters pronunciations, for example, the letter “m” is pronounced as “em” and so on. NATO alphabet is the most widely used spelling alphabet [5]. The final choice of code words for the letters of the NATO alphabet and

for the digits was made after hundreds of thousands of comprehension tests involving 31 nationalities. The qualifying feature was the likelihood of a code word being understood in the context of others [5].

Table 1. Features of speech corpora used for letters recognition (M-male, F-female)

Speech corpus	Number of words	Number of speakers	Number of utterances
LETTERS	26	2 (1M,1F)	50
NATO	26	2 (1M,1F)	50
NAMES1	250	2 (1M,1F)	20
NAMES2	70	10 (5M,5F)	20
NAMES3	26	10 (5M,5F)	20

Speech corpus NAMES1 consists of the utterances of about 10 Lithuanian names for each letter and was used during the first step of the appropriate Lithuanian names selection procedure. When the best recognized Lithuanian names were determined, the second speech corpus NAMES2 was prepared consisting of from 2 to 3 Lithuanian names for each letter. Speech corpus NAMES3 consists of the utterances of 22 Lithuanian names and 4 words (kju, Wasington, iksas, ygrekas), which represent 26 letters, used in the disease codes.

4. The adapted foreign language recognizer

Our earlier experiments showed, that *Microsoft Speech Recognizer 8.0 (Spanish-US)* provides significantly better results of Lithuanian digit names recognition comparing with English and other recognizers, implemented in Windows'7 operation system [6]. It will be marked as REC_SP. The main problem of adaptation of foreign language recognizer is to find the best phonetic sequences (transcriptions) to write Lithuanian voice commands. Our previous activities showed that the use of Spanish synthesizer, automatic generation of a wide set of possible transcription sequences and experimental evaluation of their efficiency may be useful. So Lithuanian digits names were synthesized with Spanish synthesizer and the transcriptions providing the best sounding realization were selected. Then the transcribed digit names were fed to the lexicon design tool

of Microsoft Speech Server'2007 [7] and all possible UPS-based transcriptions [8] were generated. After that two or three the best transcriptions were left for each digit name after additional testing experiments [9].

5. Lithuanian speech recognizer

The majority of speech recognition systems prefer the phoneme-based HMM recognition, however in the case of isolated word recognition, word-based HMM recognition should be investigated first, because co-articulation and prosody phenomenon is more likely to be reflected in units of longer duration. The least error rates were achieved with word-based speech recognition systems when small vocabularies were used [10].

For the development of Lithuanian speech recognizer well-known continuous density hidden Markov models (CD-HMM) approach has been used. The core of the Lithuanian recognizer is HMM models of Lithuanian words, which are trained in offline mode.

Speech recordings were transformed to the feature vectors. The recordings were sampled at 16 kHz, split into 20 ms duration frames using 10 ms displacement. Then 12 mel-frequency cepstrum coefficients (MFCC) were obtained out of 26 outputs of triangular filters linearly spaced in the nonlinear mel-frequency scale using discrete cosine transform. In addition to 12 cepstral coefficients 12 first and second order cepstrum change rate coefficients (derivatives) were calculated. These coefficients together with energy, first and second order energy derivatives were packed to make a single vector of 39 coefficients for each 20 ms signal frame. Lithuanian HMM speech recognizer was trained following the sequence of processing steps described in [11]. It will be marked as REC_LT.

6. Investigation of Lithuanian digits names recognition accuracy

Digits recognition by REC_SP recognizer. The testing experiments with SKAIC30 speech corpus were done using REC_SP recognizer. The averaged recognition accuracy of ten

Lithuanian digit names is shown in the table 2. Two modes of recognizer were used. The average accuracy of digit names recognition by REC_SP recognizer in asynchronous mode is 93.7%. The worst recognized digit is “du”.

Table 2. Accuracy of digit names recognition by REC_SP recognizer

Digit	Accuracy, %	
	Synchronous mode	Asynchronous mode
Nulis	70.7	90.8
Vienas	96.7	96.7
Du	69.3	86.0
Trys	99.0	97.8
Keturi	85.7	78.2
Penki	99.0	98.0
Šeši	100	99.2
Septyni	99.2	100
Aštuoni	99.7	99.7
Devyni	72.0	90.7
Average	89.1	93.7

Digits recognition by REC_LT recognizer.

HMM models of Lithuanian digit names were prepared using one part of speech corpus SKAIC30 (24 speakers) and the training procedure described in [11]. 20% of speech corpus SKAIC30 (6 speakers) was used during the testing procedure. It was investigated how the important HMM parameters – the number of states and the number of Gaussians per state have to be chosen for achieving a good recognition performance. The accuracy of digit names recognition by REC_LT recognizer for a varying number of states is shown in the Fig.1. The average accuracy of digit names recognition using HTK-based recognizer states is 75.1% when the number of states used in the HMMs of digit names was equal to the number of letters plus 2 additional states. The average accuracy increased to 90.9% when 4 additional states were used. The worst recognized digit is “du”, the possibility of that might be the fact that it that the lowest amount of states.

The results of accuracy of digit names recognition by REC_LT recognizer for a varying number of Gaussians are presented in the table 3. The best average accuracy of digit names recognition was achieved with 6 Gaussians – 99%. Doubling the number of

Gaussians comes along with also doubling the demand on memory and doubling the computational expense.

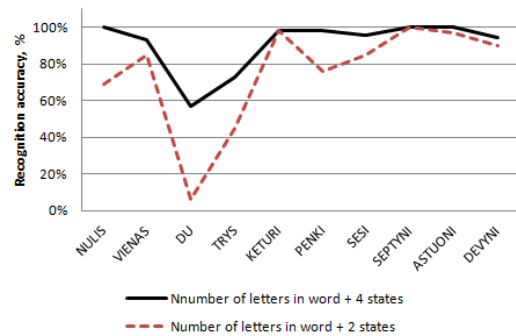


Fig. 1. Accuracy of digit names recognition by REC_LT recognizer for a varying number of states

Table 3. Accuracy of digit names recognition by REC_LT recognizer for a varying number of Gaussians

Digit	Accuracy, %		
	2 Gaussians	4 Gaussians	6 Gaussians
Nulis	91.7	100	100
Vienas	100	100	100
Du	47.5	21.7	96.7
Trys	95	92.5	95
Keturi	99.2	100	100
Penki	99.2	100	100
Šeši	97.5	98.3	100
Septyni	100	100	100
Aštuoni	100	100	100
Devyni	97.5	99.2	98.3
Average	92.8	91.2	99

The digit name “trys” is better recognized by REC_SP recognizer.

Accuracy of the sequence of digits should be equal to the accuracy of the single digit recognition raised to the power of the length of the digits sequence, for example, if the accuracy of the single digit recognition is equal 99%, the accuracy of 3 digits sequence recognition should be equal 97% (three or four digits are used in ICD-10).

7. Investigation of letters recognition accuracy

Letters recognition by REC_SP recognizer.

The average accuracy of speech corpus LETTERS recognition by REC_SP recognizer was only 25.9%, it indicates that spelled letters could not be used for the recognition of disease codes.

NATO	Acc., % REC_SP	Names	Accuracy, %	
			REC_SP	REC_LT
Alfa	94	Austėja	99.5	97.5
Bravo	20	Boleslovas	97	100
Charlie	100	Cecilija	99	100
Delta	66	Donatas	98.5	97.5
Echo	0	Eimantas	94.5	100
Foxtrott	100	Fausta	99	100
Golf	92	Gražvydas	98	97.5
Hotel	82	Hansas	100	100
India	96	Izaokas	96.5	100
Juliet	80	Jonas	96.5	82.5
Kilo	0	Karolis	100	97.5
Lima	34	Laima	99.5	100
Mike	98	Martynas	97.5	100
Nowember	90	Nojus	96.5	100
Oskar	100	Oskaras	100	87.5
Papa	98	Patrikas	100	92.5
Quebec	0	Kju	85.5	95
Romeo	82	Ričardas	98	77.5
Sierra	96	Sandra	99.5	100
Tango	100	Teodoras	100	100
Uniform	28	Ulijona	88	100
Victor	18	Vacys	99	100
Whiskey	74	Wašington	97.5	97.5
X-ray	100	Iksas	97.5	97.5
Yankee	100	Ygrekas	93.5	100
Zulu	0	Zacharijus	90	85
Average	67.2	Average	96.9	96.4

The accuracy of speech corpora NATO and NAMES3 recognition by REC_SP recognizer is presented in the

Table 4. Accuracy of NATO alphabet and names recognition

table 4. Four words of NATO alphabet (echo, kilo, Quebec, zulu) were not recognized at all, so this alphabet is not suitable for the recognition of disease codes too.

It is obviously, that the recognition accuracy of letters could be considerably improved using the appropriate set of words, as in the Morse alphabet. So the appropriate Lithuanian name was chosen for each letter: „Austėja” for letter ”a”, „Boleslovas” for letter „b” and so on. Selection of appropriate names was done experimentally by testing the recognition accuracy of ten Lithuanian names for each letter and taking the best recognized name. Only two transcriptions for each name were used, i.e., the optimization procedure of transcriptions is not finished. The average accuracy of speech corpus NAMES3 recognition by REC_SP recognizer is 96.9%. This result was achieved using recognizer in synchronous mode, therefore possibly better results could be got in asynchronous mode.

Letters recognition by REC_LT recognizer. HMM models of Lithuanian names were prepared using one part of speech corpus NAMES3 (8 speakers) and the training procedure described in [11]. 20% of speech corpus NAMES3 (2 speakers) was used during the testing procedure. The results of accuracy of names recognition by REC_LT recognizer are presented in the table 4. The

number of states used in the HMMs of names was equal to the number of letters plus 4 additional states and 6 Gaussians per state were used.

The hybrid approach of recognizer could be especially useful in the case of Lithuanian names recognition, because the recognition results are not correlated.

8. Conclusions

The problem of diseases recognition was defined during the project INFOBALSAS – the list of diseases **contains 14179 diseases and disorders and is too big to recognize it. It is proposed to recognize disease codes, consisting** of one letter and some digits. The appropriate Lithuanian name was chosen for each letter:

The experiments showed that Lithuanian digit names and the set of Lithuanian names could be surely recognized by two investigated recognizers.

The error analysis showed that errors made by two recognizers aren’t exactly correlated and the performance of one recognizer may be used for the improvement of performance of the other recognizer.

7. References

1. Zue V., Seneff S., Glass J., Polifroni J., Pao C., Hazen T.J., Hetherington L. JUPITER: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1), 2000. p. 85-96.
2. Raux A., Bohus D., Langner B., Black A., Eskenazi M. Doing research on a deployed spoken dialogue system: One year of Let’s Go! experience. In *Proceedings of INTERSPEECH-ICSLP*, 2006. p. 65-68.
3. Rudžionis V., Raškinis G., Maskeliūnas R., Rudžionis A., Ratkevičius K. Comparative analysis of adapted foreign language and native Lithuanian speech recognizers for voice user interface // *Electronics and Electrical Engineering*. Kaunas : KTU, 2013, Vol. 19, no. 7, p. 90-93.
4. TLK-10-AM sisteminis ligų sąrašas. [Online].

- Available:<http://ebook.vlk.lt/e.vadovas/index.jsp>.
5. NATO phonetic alphabet. [Online]. Available: http://en.wikipedia.org/wiki/NATO_phonetic_alphabet.
 6. Maskeliunas R., Rudzionis A., Ratkevicius K., Rudzionis V. Investigation of Foreign Languages Models for Lithuanian Speech Recognition. Electronics and Electrical Engineering, No. 3(91), Kaunas, Technologija, 2009. p. 37–42.
 7. Dunn M., Pro Microsoft Speech Server 2007: Developing Speech Enabled Applications with .NET. New York: Apress, 2007, p. 275.
 8. Universal Phone Set (UPS). [Online]. Available: <http://msdn.microsoft.com/en-us/library/hh361647.aspx>.
 9. Bartišiūtė G.; Ratkevičius K. Speech server based Lithuanian voice commands recognition // Electronics and electrical engineering. Kaunas : KTU, 2012, Vol. 18, no. 10, p. 53-56.
 10. Laurinčikaitė S. On different kinds of speech units based isolated words recognition of Lithuanian language. // Proc. of the First Baltic Conference on Human Language Technologies. – Riga, 2004. p.139-143.
 11. Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P. The HTK book – version 2.2, Entropic, 1999.

Šnekos atpažintuvų sujungimo galimybių tyrimas

*Gintarė Bartišiūtė, Gintarė Paškauskaitė
Kauno technologijos universitetas, Elektros ir elektronikos fakultetas,
Gintare.Bartisiute@gmail.com*

ĮVADAS

Maždaug 15 pastarųjų metų laikotarpyje šnekos technologijos (atpažinimas, sintezė, asmens patikra pagal balsą) tapo pramoninio informacinių technologijų panaudojimo objektu daugelyje žmogaus veiklos sričių. Viena iš sėkmingiau plintančių šnekos atpažinimo technologijų taikymo sričių yra medicininio dokumentavimo automatizavimas. Taip taupomas aukštos kvalifikacijos medicinos personalo darbo laikas rutininėms dokumentavimo operacijoms atlikti, palengvinamas ir pagreitinamas kompiuterinis informacijos pateikimas bei jos paieška. Šnekos atpažinimas pakeičia įprastinį dokumentų ar jo elementų registravimą paprastu diktavimu balsu. Derinant tai su moderniomis kompiuterinėmis - interneto - telefonijos priemonėmis sudaromos galimybės dokumentavimą atlikti bet kurioje vietoje ir bet kuriuo metu. Tam tikrais atvejais kai kurie dokumentai gali tapti prieinamais ir pacientams, suteikiant jiems galimybę juos koreguoti bei papildyti.

Vienas iš sėkmingų medicininio dokumentavimo automatizavimo žingsnių buvo 2012-2013 metais vykdytas Aukštųjų technologijų plėtros 2011-2013 metų programos projektas „Hibridinė atpažinimo technologija balso sąsajai (INFOBALSAS)“. Projekto rezultatas - hibridinis internetinis atpažintuvas, planuojamas diegti elektroninio sveikatos įrašo informacinėje sistemoje „Foxus“. Detalesnė informacija apie projektą ir jo rezultatus pateikta [1]. Projekto trukmė buvo tik pusantrų metų, todėl dviejų šiame projekte naudotų atpažintuvų sujungimo galimybių tyrimas liko pilnai neišnagrinėtas. Darbe bus naudojamosi projekto INFOBALSAS vykdymo metu gautais rezultatais ir bus ieškoma būdų kaip sujungti lietuvišką atpažintuvą ATP_LT ir lietuvių kalbai adaptuotą ispanų kalbos atpažintuvą ATP_SP.

Lygiagrečius dviejų atpažintuvų naudojimas yra prasmingas, jei, vienam atpažintuvui klystant, kitas priima teisingą sprendimą ir atvirkščiai. Nelietuviško atpažintuvo svarbiausi privalumai yra suderinamumas su kitais programiniais standartais, kas aktualu integruojant atpažinimo modulius su kitomis techninėmis bei programinėmis

priemonėmis. Šiuose atpažintuvuose įterpta eilė šnekos signalo apdorojimo modulių (triukšmo apdorojimo, signalo aptikimo, „barg-in“ ir kitos procedūros), tačiau patys statistinio atpažinimo modelio parametrai gauti, naudojant kitos kalbos (anglų, ispanų ar pan.) garsynus.

Dažniausiai šiuo metu naudojamas matematinis balso atpažinimo modelis yra paslėptos Markovo grandinės (PMM). Jose randama labiausiai tikėtina išstartoji balso komanda (atskiras žodis ar žodžių seka) pagal tam tikrus užduotus parametrus bei tenkinant tam tikrus apribojimus [2]. Atpažinimo modelio parametrai gaunami mokymo metu, o mokymas atliekamas pagal didelio skaičiaus įvairių diktorių balso įrašus (garsynus), kurie turi būti tinkamai anotuoti (sužymėti fonetiniai, skiemeniniai, kirčio, leksiniai bei sintaksiniai vienetai).

PMM technologija buvo panaudota lietuviškam ATP_LT atpažintuvui sukurti. Pradinė atpažintuvo versija apmokyta remiantis 50 diktorių (po lygiai vyrų ir moterų) skaitomų lietuviškų tekstų balso įrašais (iš viso 50 val. įrašų). ATP_LT atpažintuvo akustiniams modeliams sudaryti buvo naudojamas atviro kodo programinių priemonių rinkinys HTK v.3.2 (*Hidden Markov Toolkit*), kurio detalų aprašymą galima rasti [3]. Pirmiausia garsyno įrašai buvo transformuoti į požymių vektorių sekas. Tuo tikslu garso įrašai buvo dikretizuoti 16 kHz dažniu ir suskaidyti 20 ms trukmės analizės langais 10 ms paslinktais vienas kito atžvilgiu (persidengiantys analizės langai). Kiekviename analizės lange buvo įvertinama šnekos signalo energija ir signalo spektras. Spektro reikšmės buvo grupuojamos su 26 „filtrais“, kurie buvo išdėstyti netiesinėje (melų) dažnio skalėje. Remiantis filtrų išėjimais toliau buvo apskaičiuojama 12 melų dažnio kepstro koeficientų. Signalo energijai ir kepstro koeficientams buvo papildomai apskaičiuojami jų pirmosios ir antrosios eilės skirtumai laiko atžvilgiu. Taigi, vieną 20 ms trukmės signalo analizės langą atitiko 39 komponentes turintis požymių vektorius. Galiausiai, požymių vektoriai buvo normalizuojami, iš kiekvienos komponentės atimant tos komponentės reikšmių vidurkį. Akustiniam modeliavimui naudotas 141 fonetinių vienetų (angl. monophones) rinkinys, atsižvelgiantis į balsių ilgumo ir kirčiuotumo savybes

bei priebalsių palatališkumo savybę. Dvibalsiai ir afrikatos buvo modeliuojami kaip savarankiški fonetiniai vienetai. Apmokymo rezultatas yra maždaug 75 000 galimų kontekstinių fonemų (angl. triphones), kurias atspindi arti 40 000 skirtingų paslėptų Markovo grandinių modelių [1].

Nelietuviško atpažintuvo panaudojimas remiasi daugiakalbio atpažinimo principais, t.y. tikintis, kad vienos kalbos (paprastai, mažesnės) fonetinės savybės nemaža dalimi atsispindi kitos kalbos (paprastai, didesnės) akustiniuose-fonetiniuose modeliuose. Mokymo tikslas šiuo atveju yra rasti, kurie kitos kalbos akustiniai modeliai geriausiai aprašo lietuvių šnekos fonetinių vienetų savybes. Tinkamų fonetinių vienetų sekų (transkripcijų) lietuviškoms komandoms atpažinti parinkimas yra centrinis adaptavimo („mapping“) uždavinys. 2009-2010 m. atlikti eksperimentai, kurių aprašymas pateiktas [4] straipsnyje, parodė, kad ispanų kalbos fonetinė sistema žymiai artimesnė lietuviškai, nei angliška, todėl kaip kitakalbis lietuvių kalbai adaptuotas atpažintuvas pasirinktas ispanų kalbos atpažintuvas, platinamas su Windows'7 operacine sistema.

TYRIMO OBJEKTAS

2013 m. buvo atlikta išsami atpažintuvų ATP_LT ir ATP_SP papildomumo analizė taikomosios srities garsyno (MEDIC) pagrindu (išsamesnė informacija apie MEDIC garsyną pateikta [1]). Kiekvienam iš atpažintuvų ATP_SP ir ATP_LT buvo pateikti atpažinti 175440 garso įrašai (175440 = 731 komanda x 12 diktorių x 20 tos pačios komandos tarimų). Pagal abiejų atpažintuvų pateiktus atsakymus (sprendimus), įrašai buvo suskirstyti į poaibius. Šie poaibiai detalai atskleidžia ATP_LT ir ATP_SP papildomumą. Jie glaustai apibūdinti 1 lentelėje.

Atpažintuvų ATP_SP ir ATP_LT rezultatų papildomumas

Poaibis	Aprašymas	Failų (frazijų) kiekis
T=T	Atpažintuvų siūlomi sprendimai sutampa ir yra teisingi	135898
F=F	Atpažintuvų siūlomi sprendimai sutampa ir yra neteisingi	178
T-	Atpažintuvas LT siūlo teisingą sprendimą, atpažintuvas SP sprendimo nesiūlo	3398
F-	Atpažintuvas LT siūlo neteisingą sprendimą, atpažintuvas SP sprendimo nesiūlo	48
-T	Atpažintuvas SP siūlo teisingą sprendimą, atpažintuvas LT sprendimo nesiūlo	7
-F	Atpažintuvas SP siūlo neteisingą sprendimą, atpažintuvas LT sprendimo nesiūlo	1
--	Abu atpažintuvai nesiūlo sprendimo	1
TF	Atpažintuvų siūlomi sprendimai nesutampa, LT sprendimas teisingas	33650
FT	Atpažintuvų siūlomi sprendimai nesutampa, SP sprendimas teisingas	1357
FF	Atpažintuvų siūlomi sprendimai nesutampa, abu sprendimai neteisingi	902
Iš viso		175440

Lentelė leidžia apskaičiuoti, kad LT atpažintuvo vidutinis tikslumas yra 98,58% (T=T, T-. TF poaibiai), o SP atpažintuvo vidutinis tikslumas – 78,24% (T=T, -T, FT poaibiai). Nors atpažintuvas ATP_LT tikslumu smarkiai lenkia ATP_SP, tačiau įrašų poaibis FT yra ta sritis, kurioje situacija yra priešinga: ATP_SP veikia tiksliau nei ATP_LT.

Svarbiausias hibridinio atpažintuvo komponentas yra hibridinis sprendimų priėmimo blokas, sukonstruotas taikant mašininio mokymo metodiką. Mokymui bus naudojami MEDIC garsyno atvejai (įrašai), kuomet atpažintuvų ATP_SP ir ATP_LT sprendimai skyrėsi (žr. 1 lentelę). Dar projekto INFOBALSAS vykdymo metu suformuluotas dviejų klasių – TF ir FT atskyrimo (atpažinimo) uždavinys. Kiekvieną mokymo imties objektą (pavyzdį) sudarė abiejų atpažintuvų sprendimai konkrečiam garso įrašui. Mokymo imtį iš viso sudarė 35007 objektai (TF ir FT įrašus atitinkantys sprendimai). Ši mokymo imtis pasižymėjo labai didele mokymo objektų kiekiu skirtingose klasėse disproporcija. Klasė TF turi 33650 objektų, o klasė FT – tik 1357 objektus. Tai reiškia, kad paprasčiausia „nemokyta“ („akla“) sprendimo taisyklė teigianti „jei atpažintuvų sprendimai skiriasi, pasirink geresniojo atpažintuvo ATP_LT sprendimą“ veikia $33650/35007 \cdot 100\% = 96,12\%$ tikslumu. Mokymo rezultate gauta hibridinė sprendimo taisyklė bus naudinga tik tada, jei sugebės viršyti šios „aklos“ taisyklės tikslumą.

Kiekvienas mokymo imties objektas buvo aprašytas 70 požymių, kurie paaiškinti 2 lentelėje.

Požymiai, kuriais aprašoma mokymo imtis, hibridiniam sprendimų priėmimo blokui sudaryti

Požymio pavadinimas	Paaiškinimas
sp_prob	SP atpažintuvo pateikto sprendimo patikimumo įvertis [0,...,1000]
sp_supp	Jei SP atpažintuvo pateiktas sprendimas sutampa su LT atpažintuvo pateikta 2-aja (arba 3-aja) alternatyva, šis parametras nurodo, kiek ši 2-oji (arba 3-oji) LT alternatyva yra prastesnė už 1-ąją (prioritetinį) LT sprendimą (logaritminės tikimybės prasme). Jei LT atpažintuvas nepateikia alternatyvių sprendimų, arba jei SP atpažintuvo siūlomas sprendimas nesutampa su LT atpažintuvo alternatyvomis, šiam požymiui priskiriama reikšmė 10.
lt_prob	LT atpažintuvo pateikto sprendimo patikimumo įvertis, matuojamas vidutine logaritmine tikimybe signalo analizės langui. Frazės pradžioje ir pabaigoje galimai esančios tylos atkarpos į šį įvertinimą neįtraukiamos
lt_delta_prob	Patikimumo įvertis skirtumas tarp prioritetinės ir 2-osios LT atpažintuvo alternatyvos. Jei LT atpažintuvas nepateikia alternatyvių sprendimų, šiam požymiui priskiriama reikšmė 10
sil_prob	Patikimumo įvertis apibūdinantis, kaip frazės pradžia ir pabaiga atitinka „tylos“ PMM (logaritminės tikimybės prasme)
gender	dvireikšmis požymis, nusakantis kalbėtojo lytį (m, f)

lt_a	Proporcija (%), kurią LT atpažintuvo pateiktame prioritetiniame sprendime (frazėje) sudaro „a“ raidės. Pvz. jei LT atpažintuvas pateikia prioritetinį sprendimą „bronchų astma“, tai šis požymis lygus 16,7% (2 raidės iš 12)
lt_b	Proporcija (%), kurią LT atpažintuvo pateiktame prioritetiniame sprendime (frazėje) sudaro „b“ raidės. Pvz. jei LT atpažintuvas pateikia prioritetinį sprendimą „bronchų astma“, tai šis požymis lygus 8,3% (1 raidė iš 12)
...	
lt_ž	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
lt_space	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_a	Toliau tokiu pat būdu transformuojamas SP atpažintuvo sprendimas, žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_b	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
...	
sp_ž	žr. požymių lt_a ir lt_b paaiškinimus aukščiau
sp_space	žr. požymių lt_a ir lt_b paaiškinimus aukščiau

Pagrindiniai du požymiai yra lt_prob ir sp_prob, pvz., sp_prob – ispanų kalbos atpažintuvo priimto sprendimo panašumo įvertis, padaugintas iš 1000, kuo panašumo matas artimesnis vienetui, tuo su didesne tikimybe atpažintuvas priima sprendimą. Dar du požymiai (lt_delta_prob, sp_supp) suformuoti remiantis tuo, kad LT atpažintuvas pateikia nuo 1 iki 3 atsakymų, išrikiuodamas juos prioriteto mažėjimo tvarka. Frazė laikoma teisingai atpažinta, jei pirmasis prioritetinis sprendimas yra teisingas. Kiti požymiai yra pagalbiniai, o jų skaičių nulėmė lietuviškų raidžių skaičius. Buvo stengiamasi suformuoti kuo daugiau požymių, o jų įtaka bus nustatyta tolimesniuose tyrimuose.

Dviejų atpažintuvų sujungimui tinkamą duomenų tyrybos sistemą galima rinktis iš daugiau kaip 600 komercinių ir atvirojo kodo duomenų tyrybos sistemų [5]. 2008-2010 metais buvo atliktas tyrimas, kurio metu duomenų tyrybos paketų vartotojai balsavo kokius paketus jie naudoja realiai vykdomuose projektuose [5]. Paaiškėjo, kad dažniausiai naudojami yra atvirojo kodo paketai RapidMiner [6], R [7], KNIME [8], Weka [9], Orange [10] ir pan. Tuo tarpu gana dažnai minimi komerciniai paketai Excel ir MATLAB naudojami tik kaip pagalbiniai paketai, naudojami kartu su jau minėtais atvirojo kodo duomenų tyrybos paketais [5].

Išsami 6 atvirojo kodo duomenų tyrybos sistemų apžvalga pateikta [11]. Išvadose teigiama, kad nėra geriausios duomenų tyrybos sistemos, bet siūloma rinktis iš 4 duomenų tyrybos paketų: RapidMiner, R, Weka ir KNIME [11]. Panašūs rezultatai gauti [12] darbe, kuriame nagrinėti 12 duomenų tyrybos paketų, o geriausius įvertinimus gavo YALE (senesnė RapidMiner versija), KNIME, AlphaMiner, Weka ir Orange. Devynių tipų klasifikavimo objektai ir šešių tipų klasifikatoriai nagrinėti [13] darbe – geriausiai įvertintas Weka duomenų tyrybos paketas. Remiantis šia apžvalga dviejų atpažintuvų apjungimo tyrimams pasirinktas Weka paketas [9], tuo labiau, kad jis yra

vienas iš plačiausiai naudojamų Lietuvoje [14].

Weka pakete įdiegta kelios dešimtys klasifikatorių, todėl iš jų reiktų pasirinkti efektyviausią klasifikatorių. Pradinę atranką atliksime remdamiesi literatūros apžvalga. [14] darbe kaip populiariausi nagrinėjami Bajeso (angl. Naive Bayes NB), k artimiausių kaimynų (angl. K-Nearest Neighbour kNN), sprendimų medis (angl. decision tree), daugiasluoksnis neuroninis tinklas (ang. Multilayer Perceptron MP), atraminių vektorių klasifikatorius (angl. support vector machine SVM). Sprendimų medžio tipo populiariausi klasifikatoriai yra C4.5 ir atsitiktinis miškas (angl. random forest RF). Kitame darbe [13] nagrinėjami jau minėti NB, C4.5, SVM, ir kNN klasifikatoriai ir dar du klasifikatoriai: OneR ir ZeroR. Renkantis klasifikatorių galima remtis [15] darbu, kuriame apžvelgti 10 pasaulyje populiariausių duomenų tyrybos algoritmų, tarp jų patenka C4.5, NB, kNN, SVM, AdaBoost, CART (Classification and Regression Trees) klasifikavimo algoritmai bei keletas klasterizavimui skirtų algoritmų. Jei remsimės [11] apžvalga ir kitu tų pačių autorių darbu [16], tai kartu su jau minėtais C4.5, RF, NB klasifikatoriais reiktų nagrinėti ir RIPPER klasifikavimo algoritmą. Weka pakete nėra regresinio CART algoritmo, todėl vietoje jo pasirinktas regresinis MLR algoritmas (angl. Multinomial Logistic Regression), o iš dviejų panašių OneR ir ZeroR klasifikatorių pasirinktas ZeroR klasifikatorius.

Klasifikatoriaus atranka bus vykdoma renkantis iš 10 kandidatų: RIPPER, C4.5, MLR, RF, ZeroR, kNN, SVM, AdaBoost, NB ir MP.

EKSPERIMENTINIAI TYRIMAI

Hibridinė sprendimo priėmimo taisyklė buvo mokoma ir tikrinama 12-kartų kryžminio patikrinimo būdu. Ji buvo mokoma, naudojant 11 kalbėtojų duomenis, o dvyliktojo kalbėtojo duomenys naudojami patikrinti išmoktos taisyklės tikslumą (vėliau visų 12 bandymų rezultatai suvidurkinami). Šio eksperimento rezultatai galioja tuo atveju, kai atpažintuvas atpažįsta vieno iš žinomų 12 diktorių šneką. Eksperimento rezultatai pateikti 3 lentelėje kartu su laiku, sugaištamam apmokytos taisyklės testavimui. Testavimo laikas yra vienas iš kriterijų galutinei efektyviausio klasifikatoriaus atrankai.

Hibridinė sprendimo priėmimo taisyklė taip pat buvo mokoma ir tikrinama įprastu 10-kartų kryžminio patikrinimo būdu, nekreipiant dėmesio į mokymo objektų ir diktorių sąsajas (mokoma naudojant 90% objektų, tikslumas vertinamas naudojant likusius 10% objektų, atliekama 10 bandymų, vis keičiant testavimo aibės objektus, rezultatai suvidurkinami). Mokymo ir testavimo imtyse buvo tų pačių diktorių balsų pavyzdžių. Šio eksperimento rezultatai leidžia įvertinti, kokį tikslumą gali pasiekti hibridinis atpažintuvas, kai jis taikomas nežinomam kalbėtojui. Eksperimento rezultatai taip pat pateikti 3 lentelėje (šiam

eksperimente Weka paketas nepateikia testavimo laiko).

Dviejų klasių klasifikavimo tikslumas, % ir testavimui sugaištamas laikas, sec

Klasifikatorius	12-kartų kryžminis patikrinimas	Testavimo laikas, sec (12-kartų kryžminis patikrinimas)	10-kartų kryžminis patikrinimas
RIPPER	98,37	0,46	98,75
C4.5	98,44	0,49	98,87
MLR	98,26	0,53	98,31
RF	99,16	1,31	99,46
ZeroR	96,79	0,48	96,13
kNN	98,26	278	99,32
SVM	98,52	0,59	98,53
AdaBoost	97,60	0,47	97,22
NB	87,55	1,65	88,07
MP	98,54	2,20	99,33

Analizuojant 3 lentelėje pateiktus rezultatus matosi, kad iškeltam uždaviniui netinka kNN klasifikatorius dėl didelio testavimui sugaištamo laiko. Geriausi klasifikavimo rezultatai abiem atvejais gauti naudojant RF klasifikatorių (naudotas numatytasis medžių skaičius 100). Remiantis rezultatais, pateiktais [16] darbe, tikslinga paieškoti efektyviausio klasifikavimo tikslo prasme RF klasifikatoriaus medžių skaičiaus, be to, mažinant medžių skaičių mažėja testavimui sugaištamas laikas. Tam buvo atliktas 10-kartų kryžminio patikrinimo eksperimentas, mokymui naudojant 90% objektų, o testavimui naudojant likusius 10% objektų ir keičiant RF klasifikatoriaus medžių skaičių nuo 1 iki 400. Dalis gautų rezultatų pateikta 4 lentelėje.

Klasifikavimo tikslo priklausomybė nuo medžių skaičiaus RF klasifikatoriuje

Medžių skaičius	2	9	25	69	100
Tikslumas,%	98,51	99,26	99,39	99,46	99,46

Didinant medžių skaičių klasifikavimo tikslumas nežymiai auga ir pasiekia maksimumą, kai medžių skaičius lygus 69, o dar didinant medžių skaičių tikslumas nedidėja.

10-kartų kryžminio patikrinimo metodas parodė, kad RF klasifikatoriaus išmokta hibridinė sprendimo priėmimo taisyklė veikia 99,46% tikslumu ir viršija „aklos“ sprendimo priėmimo taisyklės tikslumą (96,12%). Taigi, hibridinis atpažintuvas teisingai atpažįsta visus T=T poaibio įrašus, (135898), visus T-poaibio įrašus (3398) ir 99,46% tikslumu atpažįsta TF + FT poaibių įrašus (34817 iš 35007). Vadinasi vidutinis hibridinio atpažintuvo veikimo tikslumas siekia 99,24% $((135898+3398+34817)/175440)$. Šis rezultatas galioja tuo atveju, kai atpažintuvas atpažįsta vieno iš žinomų 12 diktorių šneką.

12-kartų kryžminio patikrinimo eksperimentas parodė, kad RF klasifikatoriaus išmokta sprendimo priėmimo taisyklių aibė veikia 99,16% tikslumu. Atsižvelgiant į tai, kad sprendimo taisyklė iškviečiama tik tada kai ATP_SP ir ATP_LT sprendimai skiriasi,

vidutinis hibridinio atpažintuvo veikimo tikslumas siekia 99,18%. Šis eksperimentas leidžia įvertinti, kokį tikslumą gali pasiekti hibridinis atpažintuvas, kai jis taikomas nežinomam kalbėtojiui.

Išvados

Darbe naudojamosi projekto INFOBALSAS vykdymo metu gautais rezultatais ir ieškoma būdų kaip sujungti lietuvišką atpažintuvą ir lietuvių kalbai adaptuotą ispanų kalbos atpažintuvą.

Remiantis literatūros analize dviejų šnekos atpažintuvų sujungimui pasirinktas atviro kodo Weka programinis paketas bei dešimt šiuo metu populiariausių duomenų klasifikavimo algoritmų. Geriausi dviejų klasių atskyrimo rezultatai gauti naudojant RF (angl. Random Forest) klasifikatorių, kai medžių skaičius lygus 100.

RF klasifikatoriaus išmokta hibridinė sprendimo priėmimo taisyklė veikia 99,46% tikslumu ir viršija „aklos“ sprendimo priėmimo taisyklės tikslumą (96,12%). Vidutinis hibridinio atpažintuvo veikimo tikslumas siekia 99,24%, kai atpažintuvas atpažįsta vieno iš žinomų 12 diktorių šneką ir lygus 99,18%, kai jis taikomas nežinomam kalbėtojiui.

Padėka

Dėkojame vadovui K.Ratkevičiui už pagalbą ruošiant straipsnį.

Literatūra

- V. Rudžionis, G. Raškiniš, R. Maskeliūnas; A. Rudžionis, K. Ratkevičius; G. Bartišiūtė, „Web services based hybrid recognizer of Lithuanian voice commands“. *Elektronika ir elektrotechnika*, vol. 20, no. 9, pp. 50-53, 2014.
- L. Rabiner, „A Tutorial on Hidden Markov Models“. *Proceedings of IEEE*, vol. 77, issue 2, 1989, pp. 257-282.
- S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev and P. Woodland (2000), *The HTK Book*. Available: <http://htk.eng.cam.ac.uk/docs/docs.shtml>.
- R. Maskeliūnas, A. Rudžionis, K. Ratkevičius, V. Rudžionis, „Investigation of Foreign Languages Models for Lithuanian Speech Recognition“, *Elektronika ir Elektrotechnika*, no. 3, pp. 37-42, 2009.
- Y. Wang, H. Wang, Z.-G. Gu, „A Survey of Data Mining Softwares Used for Real Projects“, *International Workshop on Open-Source Software for Scientific Computation (OSSC)*, Beijing, 2011, pp.94-97.
- M. Hofmann and R. Klinkenberg, „RapidMiner: Data Mining Use Cases and Business Analytics Applications“, Boca Raton: CRC Press, 2013.
- Y. Zhao, „R Reference Card for Data Mining“, Available: <http://www.rdatamining.com/docs/r-reference-card-for-data-mining>.
- M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, et al., „KNIME: The Konstanz Information Miner“, in *Data Analysis, Machine Learning and Applications (Studies in Classification, Data Analysis, and Knowledge Organization)*, Springer, Berlin, Heidelberg, pp. 319-326, 2008.
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, „The WEKA data mining software: an update,“ *SIGKDD Explorations*, vol. 11, no. 1, pp. 10-18, 2009.
- J. Demšar, T. Curk, and A. Erjavec, „Orange: Data Mining Toolbox in Python,“ *Journal of Machine Learning Research*, vol. 14, pp. 2349-2353, 2013.
- A. Jovic, K. Brkic, N. Bogunovic, „An Overview of free software tools for general data mining“, *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2014, pp.1112-1117.

- X. Chen, G. Williams, X. Xu, „A Survey of Open Source Data Mining Systems“, *Emerging Technologies in Knowledge Discovery and Data Mining*, Springer, vol. 4819, 2007, pp.3-14.
- A. H. Wahben, Q.A. Al-Radaideh, M.N. Alkabi, E.M. Shawakfa, „A Comparison Study between Data Mining Tools over some Classification Methods“, *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, 2011, vol. 0(3), p. 18–25.
- K. Paulauskienė, O. Kurasova, „Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti“, *Informacijos mokslai*, 2013,65, pp.85-94.
- X. Wu, V. Kumar, J.R. Quinlan etc. „Top 10 algorithms in data mining“, *Knowledge and Information Systems*, Springer, vol. 14, issue 1, 2007, pp. 1-37.
- A. Jovic., N. Bogunovic, „Feature Set Extension for Heart Rate Variability Analysis by Using Non-linear, Statistical and Geometric Measures“ *Proceedings of the 31st International Conference on ITI*, 2009, pp. 35-40.

Abstract

Gintarė Bartišiūtė, Gintarė Paškauskaitė.
Investigation of the possibilities to combine two speech recognizers.

This article deals with the possibilities to combine two speech recognizers by using the results of project INFOBALSAS that took place in 2012-2013. Two recognizers were prepared during the project - a Lithuanian recognizer and a Lithuanian language adapted Spanish recognizer. The recognition accuracy of 731 Lithuanian phrases was equal to 98.58% (the result of Lithuanian language recognizer) and to 78.24% (the result of Spanish language recognizer). Speech corpus MEDIC (12 announcers, 731 phrases, 20 pronunciations) was used for testing experiments. A hybrid recognizer was created by applying machine-learning technique. An open source Weka software package and ten most popular data classification algorithms were selected for machine-learning experiments. The best recognition results were obtained using a Random Forest (RF) classifier. The average recognition accuracy of hybrid recognizer is equal to 99.24%, when the recognizer is applied for the known speaker and is equal to 99.18% when it is applied for the unknown speaker.

15 PRIEDAS. Baigiamasis magistro darbas (elektroninė versija)

Elektroninė baigiamojo darbo versija pateikiama kompaktiniame diske. Kompaktinio disko turinį sudaro:

- Elektroninė baigiamojo darbo versija išsaugota PDF ir docx formatu;
- Paruoštos demonstracinė programa;