Šnekamosios kalbos identifikavimas

Deividas Brazauskas

Vytauto Didžiojo universitetas, Informatikos fakultetas,
Vileikos g. 8, LT-44404, Kaunas
brazauskas862611594@gmail.com

Santrauka. Šiame straipsnyje gilinamasi į skirtingų autorių naudojamus šnekamosios kalbos identifikavimo metodus ir pasiektus rezultatus. Tiriamos jau egzistuojančios atviro kodo sistemos. Bandoma jas optimizuoti, apjungti naudojant bendrą duomenų rinkinį, atlikti kryžminį sistemų patikrinimą ir apjungti gautus rezultatus naudojant meta-klasifikatorių [1]. Taip pat, didesniam klasifikavimo tikslumui išgauti yra panaudojamas šnekos atpažinimo servisas, gauti rezultatai yra sujungiami su atviro kodo sistemų rezultatais naudojant meta-klasifikatorių. Tyrime yra atlikti 4 eksperimentai, kurių pagalba buvo pasiektas statistiškai reikšmingas klasifikavimo tikslumo pagerinimas.

Raktiniai žodžiai: šnekamosios kalbos identifikavimas, šnekos atpažinimas, dirbtiniai neuroniniai tinklai, mašininis mokymasis, meta-klasifikatorius.

1 Įvadas

Sparčiai augant privačių, bei komercinių tarptautinių komunikacijų poreikiui dažnai kyla skirtingomis kalbomis bendraujančių asmenų komunikacijos problemos. Esamų technologijų pagalba įmanoma bendrauti su skirtingų tautybių asmenimis, nors ir jie nesupranta kito asmens vartojamos kalbos. Tokios technologijos turi savo limitus, kaip: kalbėtojų įpareigojimas jau žinoti kito kalbėtojo vartojamą kalbą, priverčiant atlikti sistemų konfigūracijos darbus prieš pradedant pokalbį, tik tekstinio vertimo palaikymas, mažas palaikomų kalbų, tokių kaip Lietuvių, kiekis. Viešai prieinamų sprendimų leidžiančių identifikuoti šnekamąją kalbą šiuo metu nėra, yra tik pavienės - uždaro kodo, dažnai neužbaigtos arba apleistos sistemos, kurios veikia tik ant savo turimų duomenų rinkinių. Šios sistemos niekada nebuvo apjungtos su kitomis sistemomis ir jų pasiektas tikslumas nebuvo optimalus. Šio darbo tikslas yra ištirti kitų autorių naudojamus šnekamosios kalbos metodus, pasiektus rezultatus, optimizuoti esamas šnekamosios kalbos identifikavimo atviro kodo sistemas, atlikti kryžminį patikrinimą, kurio rezultatus

DOI: https://doi.org/10.15388/LMITT.2021.2

galima apjungti naudojant meta-klasifikatorių, sistemų gautus klasifikavimo rezultatus sujungti su šnekos atpažinimo servisų gautomis tikimybėmis ir apjungti meta-klasifikatoriumi.

2 Analogiškų darbų apžvalga

Tyrimui buvo išanalizuoti egzistuojantys šnekamosios kalbos identifikavimui skirti sprendimai. Analizei pasirinkti 10 straipsnių [2, 3, 4, 5, 6, 7, 8, 9, 10, 11], iš kurių dauguma siūlomų sprendimų dar nėra realizuoti ne-laboratorinėje aplinkoje. Palyginimui buvo pasirinktos 3 atviro kodo sistemos: autoriaus Paul-Louis Pröve [12], Catalin Tiseanu [13] ir Nipun Manral [14], kurios buvo apleistos po jų sukūrimo, tačiau pavyko atstatyti jų veikimą. Sistemų naudojami duomenų rinkiniai nebuvo viešai pasiekiami, daugumai autorių duomenų rinkiniai buvo suteikiami tik sistemų kūrimo tikslais. Autorių naudoti požymių aptikimo ir klasifikavimo metodai matomi 1 lentelėje. Straipsniuose dažnai aptariama daugiau nei vienas metodas siekiant rasti efektyviausią iš jų. Aptariami naudoti požymių aptikimo metodai buvo: MFCC [15], SDC, LPC, LPCC, BoS, JFA, PLP, SVM, Voice tokenizer, dažniausiai sutinkamas metodas - MFCC, kuris buvo naudojamas 7 autorių, dažniausiai naudojamas klasifikavimo metodas buvo GMM [16] ir jo kombinacijos su kitais metodais. Jis buvo naudojamas 7 iš analizuojamų straipsnių. Kiti naudojami metodai buvo: DNN [17], RNN, LSA, KNN [18], Mixture Smoothing, PRLM, P-PRLM, dauguma iš šių metodų buvo jungiami su kitais klasifikavimo metodais.

1 lentelė. Straipsnių naudoti metodai

Straipsnio autorius	Klasifikavimo metodai	Požymių aptikimo metodai
Gregoire Montavon [2]	TDNN, CNN [19]	CNN-TDNN
Haizhou Li and Bin [3]	LSA, BOS, KNN, Mixture Smoothing, SVM, GMM, P-PRLM	Voice tokenizer, SVM
Panikos Heracleous [4]	DNN, CNN, GMM-UBM, SVM	Bottleneck naudojant CNN
Ryo Masumura [5]	PA-DNN, SA-DNN, PPA-DNN, SA-LSTM-RNN, PA-LSTM-RNN, PPA-LSTM-RNN, DNN, PRLM, P-PRLM	MFCC
Eslam Mansour Mohammed [6]	WPT, ANN	MFCC, LPC

Straipsnio autorius	Klasifikavimo metodai	Požymių aptikimo metodai
Malo Grisard [7]	UBM, GMM, DNN, UBM/GMM-IV-LR	MFCC, LPC
Maarten Van Segbroeck [8]	PRLM, GMM	MFCC, SDC, PLP, JFA
Brij Mohan Lal Srivastava [9]	P-PRLM, PRLM, RNN, GMM, SVM, DNN, HMM, RNNLM	MFCC, LPCC, SDC
Rong Tong [10]	GMM, PPRLM, BOS, UBM	MFCC, SDC
Bing Jiang [11]	P-PRLM, PPRSVM, GMM-UBM, GMM-SVM, DNN, HMM, DBF-TV, PDBFTV	MFCC, SDC, PLP

Tyrime buvo dirbama su 3 šnekamosios kalbos identifikavimo atviro kodo sistemomis. Jose naudojami požymių aptikimo, bei klasifikavimo metodai matomi 2 lentelėje. Visos tiriamos sistemos naudojo MFCC metodą požymių aptikimo tikslams, klasifikavimo tikslams naudojami metodai buvo CNN, kurį naudojo Paul-Louis Pröve, GMM – Catalin Tiseanu ir jungtinis GRU/LSTM metodas, kurį pasitelkė Nipun Manral.

2 lentelė. Atviro kodo sistemų metodai

Sistemos autorius	Mokymo metodai	Požymių aptikimo metodai
Paul-Louis Pröve	CNN	MFCC
Catalin Tiseanu	GMM	MFCC
Nipun Manral	GRU/LSTM	MFCC

Kai kurių straipsnių autorių (pvz.: Malo Grisard ir Maarten Van Segbroeck) įverčių tikslas buvo kuo mažesnis kompiuterio resursų suvartojimas – Cavg (angl. Cost Average), kol kitų autorių (pvz.: Ryo Masumura) – identifikavimo laiko pagerinimas ir mažesnis kompiuterio resursų sunaudojimas. Buvo pastebėta, kad didžiausią tikslumą (mažiausią EER [20]) turintys klasifikatoriai yra hibridiniai, tiksliausias pasiektas rezultatas buvo 0.72 % EER autoriaus Malo Grisard straipsnyje. Daugelis iš analizuotų straipsnių autorių įgyvendino savo užsibrėžtus įverčių tikslus.

Šiame straipsnyje siūlomas sprendimas yra optimizuoti jau esamas atviro kodo šnekamosios kalbos identifikavimo sistemas ir sujungti su šnekos atpažinimo, t. y. įrašus į tekstą verčiančiais, servisais naudojant bendrą duomenų rinkinį, bei gautus rezultatus apjungti naudojant meta-klasifikatorių.

3 Problemos sprendimas

Norint pradėti problemos sprendimą pirma yra paruošiamas duomenų rinkinys, naudojamas visose analizuojamose sistemose. Jis paruoštas iš 5 kalbų: Vokiečių (DE), Anglų (EN), Ispanų (ES), Prancūzų (FR) ir Lietuvių (LT), kiekvienai kalbai turint po lygiai 10 valandų įrašų. Kalbų DE, EN, ES, FR įrašai surinkti iš VoxForge [21], viešai prieinamo įrašų šaltinio. LT kalbai viešai prieinamų įrašų rinkinių nerasta, todėl nuspręsta naudoti 40 audio-knygų pavyzdžius. Tai leido sudaryti 8 iš 10 valandų įrašų rinkinį, likę įrašai gauti iš akademiniams tikslams skirtų šaltinių. Įrašų ilgiai yra suvienodinti po 10 sekundžių, ilgesnius įrašus padalijant į atskiras dalis ir pašalinant trumpo ilgio įrašus. Surinktiems LT kalbos įrašams atliktas kokybės suvienodinimas, sutapatinant įrašų kokybę su kitų kalbų įrašais. Tai buvo pasiekta atsitiktinai keičiant įrašų garso ir aido lygius, kalbėjimo greitį, panaudojami „highpass“, bei „bandpass“ signalo filtrai. Buvo suvienodintos įrašų direktorijos, takelių skaičius, dokumentų tipai, bitų sparta (angl. bit-rate). Gauti įrašų rinkinių kalbėtojų kiekiai matomi 4 lentelėje.

4 lentelė. Duomenų rinkinio kalbėtojų kiekiai

Kalba	Kalbėtojai
DE	303
EN	513
ES	232
FR	259
LT	102

Kryžminiam patikrinimui pasiruošti yra reikalingas duomenų išskirstymas į tais pačiais kalbėtojais nepersidengiančias aibes. Tam atlikti duomenų rinkinys indeksuojamas, suliejami visų kalbų įrašai į bendrą aibę, ji išmaišoma atsitiktiniu būdu, įrašai padalijami į 5 dalis, taip, kad aibių įrašų, bei kalbėtojų kiekių skirtumas nebūtų didelis. Paruošti duomenys matomi 5 lentelėje.

Turint paruoštą duomenų rinkinį šią problemą buvo bandoma spręsti keliais alternatyviais eksperimentiniais būdais. Pirmas eksperimentas buvo sutvarkyti sistemų [12, 13, 14] atvirą programinį kodą, suvienodinant jų naudojamus duomenų rinkinius, ištaisant pastebėtas daromas klaidas ir

5 lentelė. Duomenų rinkinio kalbėtojų kiekiai

Aibės nr.	Kalbėtojų kiekis	Įrašų kiekiai					
		DE	EN	ES	FR	LT	Bendras
1	278	780	701	660	667	351	3159
2	277	833	679	759	653	873	3797
3	281	516	706	625	717	796	3360
4	282	716	775	785	782	776	3834
5	291	755	739	771	781	804	3850

optimizuojant jų įverčius, tokius kaip sistemų modelio sluoksniuose neuronų kiekius. Išbandyti sistemas individualiai naudojant bendrą duomenų rinkinį. Antras eksperimentas buvo atlikti sistemų kryžminį patikrinimą ir apjungti gautus rezultatus išbandant keletą meta-klasifikatorių. Meta-klasifikatoriaus realizavimui buvo naudojamas WEKA [22] programinės įrangos paketas. Trečias – panaudoti šnekos atpažinimo servisus, norint gauti statistinį kiekvieno įrašo priklausomumą nuo tam tikros kalbos tikimybės įverčio. Naudotas įrašus į tekstą verčiantis šnekos atpažinimo servisas Google Cloud Speech-to-Text [23]. Sukurtas programinis kodas, kuris nusiunčia kiekvieną įrašą į nuotolinį servisą kiekvienai tikrinamai kalbai. Kartu su įrašu į servisą yra paduodamas kalbos sutrumpinimas. Tai leidžia servisui nuspėti duoto įrašo kalbos klasių tikimybes specifinėms kalboms. Iš gautų įverčių kalbų tikimybių kiekvienam įrašui yra išrenkami po 5 požymius – kiekvienos kalbos tikimybės procentinis įvertis, kurio pavyzdys matomas 1 pav. Gautų duomenų aibė buvo panaudota meta-klasifikatoriuje tikslumo įverčiui išgauti.

file	language	sys_4_de	sys_4_en	sys_4_es	sys_4_fr	sys_4_lt
en_3438	en	0.214	0.241	0.173	0.255	0.116
en_895	en	0.207	0.275	0.2	0.111	0.206
de_2475	de	0.273	0.246	0.261	0.117	0.103
en_1414	en	0.221	0.167	0.228	0.261	0.123

1 pav. Meta-klasifikatoriaus duomenys

Ketvirtas eksperimentas – sujungti sistemų [12, 13, 14] sprendimus su šnekos atpažinimo gautais sprendimais ir juos apjungti naudojant meta-klasifikatorių. Į metą-klasifikatorių yra paduodami įrašų kalbų tikimybių įver-

tinimo duomenys, kurie susideda iš 20 požymių - 5 požymiai iš kiekvienos sistemos. Meta-klasifikatoriaus realizavimui buvo pasitelktas įrankis WEKA. Atlikus kiekvieną eksperimentą tyrimų sprendimams buvo patikrinamas statistinis reikšmingumas, skaičiuojant pasikliautinius intervalus [24], su 95 % pasikliautinumo lygiu. Pasikliautinumo intervalų formulės yra:

$$\overline{x} - Z^* \left(\frac{\sigma}{\sqrt{n}}\right), \overline{x} + Z^* \left(\frac{\sigma}{\sqrt{n}}\right)$$

4 Pasiūlytų metodų tyrimai

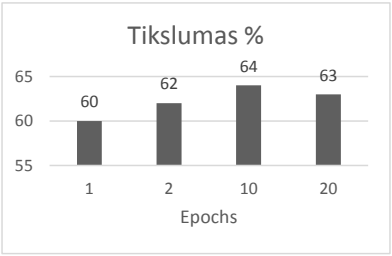
Sistemos buvo vertinamos naudojant tikslumo įvertį. Tikslumas yra skaičiavimo, matavimo arba klasifikavimo rezultatų tikslinimo su atskaitos verte kokybinis matas, kuris apskaičiuojamas su šia formule:

$$tikslumas = \frac{TN + TP}{TP + FP + TN + FN}$$

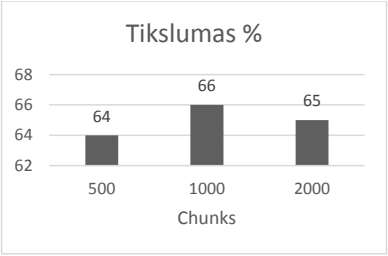
Paruošus duomenų rinkinį toliau buvo atliekami sistemų [12, 13, 14] optimizavimo darbai. Paul-Louis Pröve sistemai buvo keičiamos parametrų Epochs ir Chunks reikšmės, geriausios reikšmės, kurios matomos 2 ir 3 pav., buvo nustatytos: Epochs - 10, Chunks - 1000. Pakeitus Catalin Tiseanu sistemos parametro NUM_MIXTURES reikšmę, kuri yra matoma 4 pav., geriausias tikslumo rezultatas buvo pasiektas naudojant 2048 reikšmę. Nipun Manral keičiant Epochs parametro reikšmę, kuri matoma 5 pav., atrasta geriausia parametro reikšmė - 20. Buvo bandomi optimizuoti ir kiti sistemos rasti parametrai, tačiau juos keičiant žymaus sistemos tikslumo pokyčio nebuvo. Atlikus sistemų optimizavimo darbus gautas vidutiniškai 7 % tikslumo padidėjimas. Atsižvelgus į pasikliautinumo intervalus, galime teigti, jog optimizuotų sistemų tikslumo pagerėjimas yra statistiškai reikšmingas. Daugiau informacijos matoma 6 lentelėje.

Pritaikius 5-kartų kryžminio patikrinimo procedūrą ir jos rezultatus apjungus meta-klasifikatoriumi, buvo gautas 72.63 % tikslumas, naudojant 100 iteracijų Bagging [25] klasifikatorių. Meta-klasifikatoriuje buvo išbandyti visi galimi algoritmai su optimizuotomis algoritmų parametrų variacijomis. Apskaičiavus pasikliautinumo intervalą buvo pastebėta, kad šis tikslumo pagerėjimas nėra statistiškai reikšmingas. Pritaikius šnekos atpažinimo sistemą ir gautus rezultatus apjungus meta-klasifikatoriumi buvo gautas 82,72 %

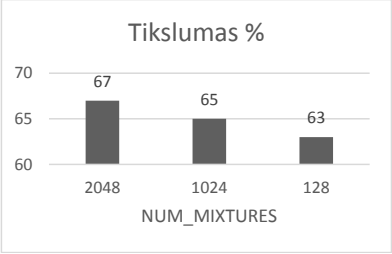
tikslumas, naudojant 100 iteracijų Bagging klasifikatorių. Atlikus pasikliautinumo intervalo skaičiavimus, buvo matoma, kad tikslumo pokytis buvo statistiškai reikšmingas. Sujungus atviro kodo sistemų, bei šnekos atpažinimo sistemos rezultatus ir pritaikius jiems 100 iteracijų Bagging meta-klasifikatorių buvo gautas 89,95 % klasifikavimo tikslumas. Apskaičiavus jungtinės sistemos pasikliautinumo intervalą, buvo pastebėtas statistiškai reikšmingas tikslumo pagerėjimas.



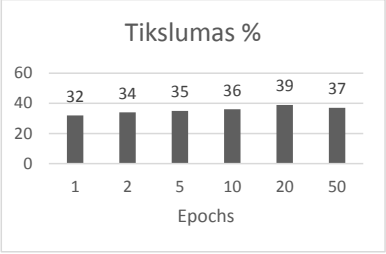
2 pav. Paul-Louis Pröve parametras Epochs



3 pav. Paul-Louis Pröve parametras Chunks



4 pav. Catalin Tiseanu parametras NUM_MIXTURES



5 pav. Nipun Manral sistemos Epochs

6 lentelė. Atviro kodo sistemų optimizavimas

Sistemos autorius	Neoptimizuotas tikslumas %	Optimizuotas tikslumas %
Paul-Louis Pröve	60	66
Catalin Tiseanu	63	71
Nipun Manral	32	39

5 Išvados

Atlikus straipsnių analizę, pastebėta, kad dažniausiai naudojamas metodas požymių aptikimui yra MFCC, o mokymui – GMM. Galime teigti, kad šie metodai buvo pasirinkti dėl jų efektyvumo. Išanalizavus 3 atviro kodo sistemų veikimą, pastebėta, kad šių sistemų veikimas yra ribotas, neoptimizuotas ir ne pilnai įgyvendinta sistemų kūrėjų vizija, todėl jų topologijas toliau galima optimizuoti. Daroma prielaida, kad tai galėtų padėti gauti didesnį klasifikatoriaus tikslumą.

Įrašų kiekis turi didelę įtaką galutiniam sistemų tikslumui, bei mokymo trukmei. Naudojamo duomenų rinkinio praplėtimas, bei optimizavimas yra galimas tolimesnis tikslumo padidinimo būdas. Pastebėtina, kad didinant kalbų kiekį esantį duomenų rinkinyje gali daryti negatyvią įtaką algoritmo tikslumui, todėl didinant kalbų kiekį yra patartina padidinti esamų kalbų įrašų kiekius.

Apibendrinant atliktus eksperimentus pastebėta, kad pirmo eksperimento sistemų optimizavimo darbai suteikė statistiškai reikšmingą pagerėjimą, vidutiniškai 7 % tikslumo. Panaudojus bendrą duomenų rinkinį visoms atviro kodo sistemoms buvo galima palyginti kiekvienos sistemos klasifikatoriaus tikslumą. Buvo pastebėta, kad tiksliausiai klasifikuojanti sistema buvo autoriaus Catalin Tiseanu, sugebanti klasifikuoti 72 % tikslumu. Iš antro eksperimento rezultatų pastebėtas, kad gautas 0,6 % papildomas tikslumas nebuvo statistiškai reikšmingas, tačiau pamatyta, kad meta-klasifikatoriaus gautas tikslumas yra didesnis nei visų 3 atviro kodo sistemų tikslumo įverčių vidurkis. Atlikus trečią eksperimentą pastebėta, kad šnekamosios kalbos identifikavimo tikslumui padeda šnekos atpažinimo sistemos panaudojimas, patartina apjungti klasifikatorių su šiais, bei jau turimais atviro kodo sistemų kalbų tikimybių duomenimis. Atlikus ketvirtą eksperimentą matoma ryški, kad ir blogesnių, sistemų apjungimo naudojant meta-klasifikatorių nauda. Tikėtina, praplečiant meta-klasifikatoriaus naudojamą duomenų rinkinį būdų galima pasiekti didesnio tikslumo nei 89,95 %, tačiau tam reikalinga atliktų papildomų tyrimų su meta-klasifikatoriais.

Literatūra

[1] A Meta Classifier by Clustering of Classifiers https://link.springer.com/chapter/10.1007/978-3-319-13650-9_13 [žiūrėta 2021 03 22].

[2] Gregoire Montavon (2009). Deep learning for spoken language identification. Machine Learning Group Berlin Institute of Technology.

[3] Haizhou Li and Bin (2005). A Phonotactic Language Model for Spoken Language Identification. Ma Institute for Infocomm Research.

[4] Panikos Heracleous, Kohichi Takai, Keiji Yasuda, Yasser Mohammad, Akio Yoneyama (2018). Comparative Study on Spoken Language Identification Based on Deep Learning. KDDI Research, Inc.

[5] Ryo Masumura, Taichi Asami, Hirokazu Masataki, Yushi Aono (2017). Parallel Phonetically Aware Dnns And Lstm-Rnns For Frame-By-Frame Discriminative Modeling Of Spoken Language Identification. NTT Media Intelligence Laboratories.

[6] Eslam Mansour mohammed, Mohammed Sharaf Sayed , Abdallaa Mohammed Moselhy, Abdelaziz Alsayed Abdelnaiem (2013). LPC and MFCC Performance Evaluation with Artificial Neural Network for Spoken Language Identification. Department of Electrical and Computer Engineering.

[7] Malo Grisar, Petr Motlicek, Wissem Allouchi, Michael Baeriswyl, Alexandros Lazaridis, Qingran Zhan (2019). Spoken language identification using language bottleneck features. EPFL, Department of Electrical Engineering, Lausanne.

[8] Maarten Van Segbroeck, Ruchir Travadi, Shrikanth S. Narayanan (2015). Rapid Language Identification. IEEE/ACM Transactions On Audio, Speech, And Language Processing 2015.

[9] Brij Mohan Lal Srivastava, Hari Vydana, Anil Kumar Vuppala, and Manish Shrivastava Language Technology Research Center (2017). Significance of neural phonotactic models for large-scale spoken language identification. International Institute of Information Technology.

[10] Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li and Eng Siong Chng (2014). Integrating Acoustic, Prosodic And Phonotactic Features For Spoken Language Identification. Institute for Infocomm Research.

[11] Bing Jiang, Yan Song , Si Wei, Jun-Hua Liu, Ian Vince McLoughlin, Li-Rong Dai (2014). Deep Bottleneck Features for Spoken Language Identification. University of Science and Technology of China.

[12] Github – Sistema nr. 1 „Spoken Language Recognition“, Paul-Louis Pröve https://github.com/pietz/language-recognition/ [žiūrėta 2021 03 14].

[13] Github – Sistema nr. 2 „Spoken language identification“, Catalin Tiseanu https://github.com/CatalinTiseanu/spoken-language-identification/ [žiūrėta 2021 03 14].

[14] Github – Sistema nr. 4 „Spoken-Language-Identification“, Nipun Manral https://github.com/nipunmanral/Spoken-Language-Identification [žiūrėta 2021 03 14].

[15] Cepstrum and MFCC https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC [žiūrėta 2021 03 22].

[16] Gaussian Mixture Models Explained https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95 [žiūrėta 2021 03 22].

[17] Deep Neural Network https://www.sciencedirect.com/topics/engineering/deep-neural-network [žiūrėta 2021 03 22].

[18] Machine Learning Basics with the K-Nearest Neighbors Algorithm https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [žiūrėta 2021 03 22].

[19] Convolutional Neural Network https://www.sciencedirect.com/topics/engineering/convolutional-neural-network [žiūrėta 2021 03 22].
[20] EER - equal error rate https://www.webopedia.com/TERM/E/equal_error_rate.html [žiūrėta 2021 03 15].
[21] VoxForge http://www.voxforge.org/ [žiūrėta 2021 03 18].
[22] WEKA https://www.cs.waikato.ac.nz/ml/weka/ [žiūrėta 2021 03 18].
[23] Goolge Cloud Speech-to-Text https://cloud.google.com/speech-to-text/ [žiūrėta 2021 03 18].
[24] Pasikliautinasis intervalas https://www.vle.lt/straipsnis/pasikliautinasis-intervalas/ [žiūrėta 2021 03 30].
[25] ML | Bagging classifier https://www.geeksforgeeks.org/ml-bagging-classifier/ [žiūrėta 2021 03 22].