AhoLab taldearen sarrera Albayzin 2010 hizlarien diarizazio erronkarako

Iker Luengo, Eva Navas, Ibon Saratxaga, Inmaculada Hernáez, Daniel Erro

Elektronika eta Telekomunikazioak saila
Euskal Herriko Unibertsitatea
{iker.luengo, eva.navas, ibon.saratxaga, inma.hernaez, daniel.erro}@ehu.es

Abstract

This paper presents the speaker diarization system presented by Aholab Signal Processing Laboratory to the Albayzin Speaker Diarizarion Challenge 2010. The system was built to run on-line, without any recording of the audio data to produce its output. As a result, the whole process must be done in a single iteration, which prevents the use of many optimization processes that are usually implemented in diarization systems. In order to minimize the reduction of the accuracy in the output and to maximize computational efficiency, the applied algorithms were carefully selected and some new modifications were implemented.

Laburpena

Artikulu honek, Aholab taldeak 2010ean Albayzin hizlarien diarizazio erronkara aurkeztutako sistema deskribatzen du. Sistema hau linean lan egiteko diseinatu zen, hau da, diarizazioaren emaitza lortzeko audioaren grabaketa beharrezkoa izan barik. Beraz, prozesu osoa iterazio bakar baten egin behar da. Honen ondorioz, diarizazio sistemetan ohizkoak diren optimizazio teknika asko ezin dira erabili. Zehaztasunaren murrizketa ahal den heinean mugatzeko, eta kalkulu eraginkortasuna hobetzeko, erabilitako algoritmoak kontu handiz aukeratu dira, eta kasu batzuetan aldaketa berriak ere proposatzen dira.

Keywords: Speaker diarization, BIC, on-line audio processing.

Gako hitzak: Hizlarien diarizazioa, BIC, linean audio prozesaketa.

1. Sarrera

Audio grabaketa baten agertzen diren hizlari aldaketak antzematea, eta ondoren, lortutako zatiak hizlarien arabera sailkatzea da diarizazioaren helburua. Hau da, pertsona bakoitzak noiz hitz egin duen jakitea da zeregina.

Prozesu hau normalean azpilanetan banatu egiten da, bakoitza arazo osoaren atal batez arduratzen delarik. Gehienetan azpilanak hauek izaten dira: hizketaren antzematea, hizlarien txanden banaketa, multzokatzea eta birsegmentazioa. Bata bestearen atzetik abiatzen dira normalean, hau da, pausu bakoitza audio osoan aplikatzen da hurrengoa hasi aurretik. Egitura honek beharrezkoa egiten du audioaren grabaketa osoa erabilgarri edukitzea prozesamendu bakoitzerako. Gainera, ezinezkoa da emaitzik lortzea grabaketa osoa amaitu aurretik.

Lineaz kanpoko egitura honen ordez, linean lan egiten duen diarizazio algoritmo bat proposatzen dugu. Algoritmo honek pausu bakar baten egiten du prozesamendu osoa, eta beraz, ez du inolako audio grabaketarik behar, mikrofono sarrerarekin zuzenean lan egiteko aukera emanez. Gainera, kalkulu eraginkortasun handiagoa du, prozesamendu denbora eta memoria beharrak murriztuz. Bestalde, honelako egitura duen sistema batek ez ditu ondorengo audio laginak ezagutzen, eta aurrekoak bakarrik erabili ahal ditu erabakiak hartzeko. Beraz, zehaztasunaren murrizketa bat itxarotekoa da. Pausu anitz edo algoritmo iteratiborik ez erabiltzeak ere emaitzen zehaztasuna mugatu dezake.

2. Hizlarien diarizazioa: sarrera labur bat

2.1. Hizketaren antzematea

Hizketaren antzematea (ingelesez VAD, *Voice Activity Detector*) beharrezkoa da hizketarik ez duten audio zatiak baztertzeko. Honekin hurrengo urratsak erraztu egiten dira, eta baita emaitza fidagarriagoak lortu ere. Sistema zein ingurunean erabiliko den arabera, hizketa-bako zatiak gertaera akustiko desberdinak izan daitezke: isiluneak, zarata, musika, txaloak, garrasiak, eta abar. Honegatik, gehienetan aurretik entrenatutako ereduekin dabilen Viterbi segmentazio bat erabiltzen da hizketa zatiak bereizteko. GMM ereduak (Duda eta Hart, 2001) dira gehien erabiltzen direnak, baina batzuetan HMMak (Rabiner, 1989) aukeratzen dira.

Posiblee da eredu bi bakarrik erabiltzea (bata hizketarako eta bestea ez-hizketarako), baina gertaera akustiko ezberdinak espero direnean komenigarri da bakoitzeko eredu bat entrenatzea. Sarritan bost erabiltzen dira: zarata, musika, hizketa garbia, hizketa+zarata eta hizketa+musika (Reynolds eta Torres-Carrasquillo 2004). Gizonezko eta emakumezko

hizketentzako eredu desberdinak erabiltzea ere posiblea da, edota banda zabaleko eta banda murriztuko hizketa desberdintzea (Sinha eta beste, 2005). Era honetan, eta entrenamendurako behar adina datu daudela suposatuz, antzematearen zehaztasuna hobetu egiten da.

2.2. Hizlarien txanden banaketa

Pausu hau erabakigarria da diarizaziorako. Hizketarik gabeko zatiak baztertu ondoren, hizlarien aldaketak noiz eman diren aurkitu behar da. Honetarako bata bestearen ondoan kokaturiko leiho biren arteko aldea kalkulatzen da, eta atalase jakin bat baino handiagoa bada, leiho bien arteko mugan hizlari aldaketa bat eman dela suposatzen da. Erabilitako algoritmoen arteko ezberdintasuna aldea kalkulatzeko metrikan eta leihokaketa sisteman datza.

Gehien erabilitako metrika BIC (Bayesian information criterion) delakoa da (Chen eta Gopalakrishnan, 1998). Eredu baten konplexutasunagatik zigorturiko egiantza da eredu horren BIC balioa. Hau da, zenbat eta egiantza handiagoa, BIC handiagoa dauka eredu batek, datuen banaketa hobeto modelatzen duelako; baina eredu konplexu batek BIC txikiagoa dauka egiantza bera lortzen duen eredu sinple batek baino.

Beraz, BIC balioa erabili daiteke leihokatutako audioa banaketa birekin (bakoitza leiho batentzako) ala bakarrekin (leiho biak batuz) hobeto modelatzen den ikusteko. Leiho birekin hobeto modelatzen bada, hizlari aldaketa bat egon dela esan nahi du, bestela, hizlari bera da leiho bietan agertzen dena.

Dena den, BIC balioa kalkulatzeak prozesamendu karga handia suposatzen du. Honegatik inplementazio batzuetan beste metrika batzuk erabiltzen dira, nolabait zehaztasun gutxiagokoak, baina askoz azkarragoak. Adibidez, Hotelling T^2 (Zhou eta Hansen, 2000), dibergentzia gaussiarra (Sinha eta beste, 2005) eta GLR (Meignier eta beste, 2006). Posiblea da baita metrika azkarrago hauek lehen pauso bezala erabiltzea eta ondoren lortutako aldaketa uneak BIC bidez zehaztea (Delacourt eta Wellekens, 2000).

Leihokaketari dagokionez, tamaina finkoa duten eta bata bestearen ondoan kokatuta dauden leiho bi erabiltzea da sistema sinpleena. Leihoak seinalean zehar mugitu ahala, metrikaren balioak kalkulatu egiten dira, eta metrika kurbaren gailurrek finkatzen dituzte aldaketa uneak. Sarritan BIC metrikarekin beste leihokaketa landuago bat erabiltzen da, geroz eta handiagoa den leiho batekin. Leiho honen barruan dagoen trama bakoitzean BIC balioa kalkulatu egiten da. Balio hauen maximoa zero baino handiagoa bada, maximoaren tokian aldaketa bat dagoela suposatzen da, eta leihoa hasierako neurrira bueltatu eta maximoaren tokian jartzen da prozesua jarraitzeko. Balio guztiak zero baino txikiagoak badira, leihoa toki berean mantendu baina bere luzera handitu egiten da (Chen eta Gopalakrishnan, 1998).

Handiagotzen doan leiho honekin zehaztasun hobea lortzen da sarritan, baina kalkulu karga handiagoa suposatzen du. Inplementazio batzuetan kalkulu beharrizanak murriztu egiten dira leihoaren tamaina mugatuz eta hizlari aldaketak probabilitate gutxiko uneetan bilatzea eragotziz (Tritschler eta Gopinath, 1999; Cettolo eta Vescovi, 2003).

2.3. Multzokatzea

Behin hizlarien banaketa egin eta gero, hizlari berari dagozkion zatiak zeintzuk diren jakin behar da. Pausu hau multzokatze algoritmo batekin burutzen da, eta behetik gorako multzokatzea da metodo erabiliena. Zati guztien arteko distantzia kalkulatu egiten da, eta hurbilen dauden zati biak multzokatzen dira. Distantzia matrizea eguneratu eta beste pare bat aukeratzen da multzokatzeko, amaitzeko irizpidea bete arte. Metrika asko erabili daitezke distantzia lez: BIC, GLR, GMMen arteko distantzia euklideoa, eta abar.

2.4. Multzoen birkonbinazioa

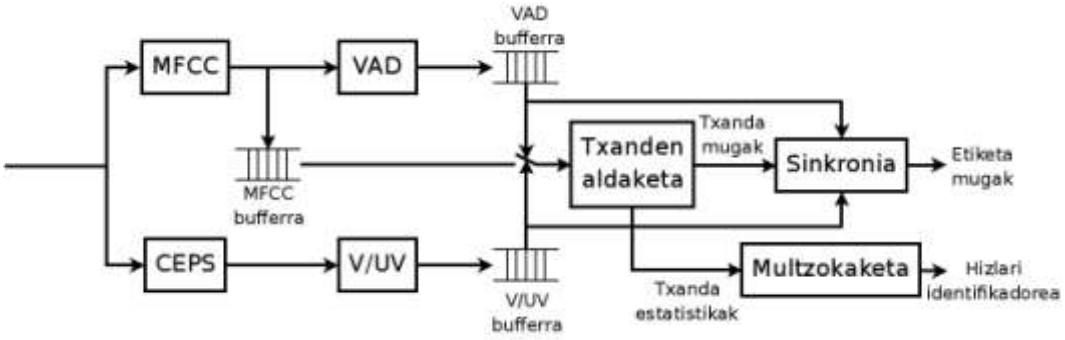
Nahiz eta pausu hau guztiz beharrezkoa ez izan, emaitzen zehaztasuna hobetu dezake (Reynolds eta Torres-Carrasquillo 2004). Ideia zera da: lehenengo multzokatzean azpi-multzokatzea (hau da, hizlari baino multzo gehiago lortzea), baina aldi berean, multzo bakoitzak hizketa denbora nahikoa daukala ziurtatzea. Multzo bakoitzeko GMM eredu bat moldatzen da UBM eredu batetik, MAP moldaketaren bidez. Ondoren GLRa kalkulatzen da eredu guztien artean, eta hurbilen dauden multzo biak batzen dira. Multzoak batzerakoan, GMM eredu berri bat sortu behar da eta distantzia matrizea eguneratu, amaitzeko irizpidea bete arte. Frogatuta dago parametroen normalizazioa beharrezkoa dela teknika honekin hobekuntzarik lortzeko (Tranter eta Reynolds, 2006).

2.5. Birsegmentazioa

Pausu honetara heltzean, multzo (edota hizlari) bakoitzeko hizketa kopuru nahikoa daukagu. Beraz, posible da hizlari bakoitzeko eredu berriak entrenatzea, eta hauek ez-hizketa ereduekin batera erabiltzea Viterbi segmentazio berri bat lortzeko. Pausu honekin txanden mugen zehaztasuna handitu daiteke, are gehiago segmentazio iteratiboa erabiltzen bada.

3. Proposaturiko algoritmoaren deskribapena

1. irudiak proposaturiko diarizazio algoritmoaren diagrama eskematiko bat erakusten du. Sistema hau hiru zutabetan oinarrituta dago. Alde batetik, hizketaren antzematea Viterbi algoritmoarekin eta GMM ereduekin egiten da. Gero, hizlarien txandak antzemateko algoritmo eraginkor eta azkar bat aukeratu da, BIC metrika erabiliz. Azkenik, linean lan egiteko prestatuta dagoen multzokatze algoritmo bat erabiltzen da.



1. irudia: Proposaturiko diarizazio algoritmoaren eskema.

Hizketaren antzematea MFCC parametroen eta lehenengo eta bigarren deribatuen bidez egiten da. Txanden banaketak, ordea, ez ditu deribatuak erabiltzen, eta gainera, trama ahostunak bakarrik erabiltzen ditu. Honetarako, trama ahostunak eta ahoskabeak bereizteko sistema bat erabiltzen da (irudian V/UV bezala agertzen dena). Algoritmo osoa linean lan egiteko prestatuta dago, audioaren grabaketa behar izan gabe. Hurrengo ataletan sistemaren pausu bakoitza zehaztasun handiagoarekin azaltzen da.

3.1. Hizketaren antzematea

GMM eredu desberdin bat entrenatu da musikarako, zaratarako, hizketa garbirako, hizketa+zaratarako eta hizketa+musikarako, Albayzin erronkaren antolatzaileek eskainitako garapen grabaketak eta etiketak erabiliz. Eredu hauek Viterbi segmentazio algoritmo baten erabiltzen dira hizketa eta hizketa-bako zatiak antzemateko.

Prozesu osoa linean egin behar denez, linean dabilen Viterbi algoritmo bat inplementatu da, Srámeken lanean aurkeztu bezala (Šrámek, 2007). Algoritmo honek aktibo dauden bide guztien erregistroa darama, eta bide guztiek nonbait bat egiten badute ala ez modu eraginkor baten antzematen du. Bide guztiak puntu baten bat egin ezkero, puntu horretaraino egindako erabakiak (hau da, tramek hizketarik daukaten ala ez) zehaztasunik galdu gabe atera daitezke. Beraz, ohizko Viterbi algoritmo baten kontra, ez dago zertan itxaron grabaketa osoa amaitu arte erabakiak ateratzeko. Gainera, bat egite punturaino erabilitako memoria guztia ezabatu daiteke. Honi esker, memoria beharrizan txikia dauka algoritmoak, eta audio oso luzeekin lan egin dezake.

Audioa 12 MFCC eta lehenengo eta bigarren deribatuak erabiliz parametrizatu egiten da. Garapen frogetan ikusi da deribatuak erabiltzean zehaztasuna nolabait hobea dela, eta honegatik erabiltzen dira.

3.2. Hizlarien txanden banaketa

Txanden banaketarako BIC metrika eta geroz eta handiagoa den leiho bat erabiltzen dira. Handitzen doan leihoak tamaina finkokoak baino emaitza hobeak ematen ditu, baina kalkulu beharrizanak askoz handiagoak dira. Beharrizan hauek ahal den guztia murrizteko, ondorengo moldaketak egin dira (Cettolo eta Vescovi, 2003):

- Sistema 5 segundoko leiho batekin hasten da.
- Ez da txanda aldaketarik bilatzen leihoaren hasierako eta amaierako 2 segundoetan. Hau honela egiten da, BIC metrikarekin nekez antzematen delako 2 segundoko txandarik. Beraz, ez da ezer galtzen tarte horiek ez aztertzearren.
- Aldaketarik ez aurkitzekotan, leihoa 2 segundoz luzatzen da.
- Leihoa 20 segundotara heltzerakoan, luzatu ordez, 2 segundoz mugitzen da, tamaina finkoa mantenduz.
- Leiho bakoitzean, 250 milisegundoero bilatzen dira aldaketak. Aldaketa bat aurkitu ezkero, leiho bera berriz aztertzen da 50 milisegundoero, zehaztasuna hobetzeko.
- Hizlari aldaketa bat antzematean, leihoa 5 segundoko luzera berreskuratzen du, eta aldaketaren unean jartzen da.

Teknika honek handitzen doan leiho baten zehaztasuna ematen du, eta aldi berean, kalkulu beharrizanak txikiagotu egiten ditu. Gainera, BIC balioen kalkulua ere modu eraginkor baten egiten da, parametroen metatze buferren bidez, Cettolo eta Vescoviren lanetan erakutsi bezala (Cettolo eta Vescovi, 2003).

Garapen frogetan ikusi da kasu honetan parametroen deribatuak erabiltzea ez dela komenigarria, eta beraz, 12 MFCCko parametrizazioa erabili da, deribaturik gabe. Gainera, trama ahoskabeak baztertu eta ahostunak bakarrik erabiltzean akatsak %12 bat jaisten direla ikusi da.

3.3. Multzokatzea

Zatien multzokatzerako BIC metrika darabilen eta linean lan egiteko prestatuta dagoen multzokatze

algoritmo bat aukeratu da (Tritschler eta Gopinath, 1999). Txanden banaketa antzemateko algoritmoak hizlari aldaketa bat antzematen duen bezain laster, atera berria den audio zatia multzokatze algoritmora bidaltzen da. Algoritmo honek BIC metrika kalkulatu egiten du zati honen eta ordura arte aurkitutako multzo guztien artean. Multzo guztietatik metrika txikiena duena aukeratzen da, eta metrika hau zero baino txikiagoa bada, zati berria multzo horretara sartzen da, multzoaren estatistikak eguneratuz. Metrika zero baino handiagoa bada, multzo berri bat eratzen da zati berriarentzako. Era honetan zati bakoitza eskuragarri egon bezain laster zein hizlariari dagokion jakin daiteke.

Multzokatze sistema honek behetik gorako algoritmo batek daukan baino zehaztasun txikiagoa dauka teorikoki. Izan ere, behetik gorako algoritmoak zati guztiak ikusten ditu hasieratik, eta beraz, informazio osoa du eskuragarri erabakiak hartzeko. Linean dabilen sistemak, ordea, aurreko zatiak bakarrik ikusten ditu, eta ezin du hurrengo zatien informaziorik erabili multzokatzea egiteko.

Praktikan ordea, lineazko algoritmoak emaitza hobeak ematen ditu diarizaziorako. Arrazoia honako hau da: txanden banaketa egiten duen algoritmoak hizlari aldaketa gehiegi antzematen ditu, hau da, ezarritako muga batzuk faltsuak dira. Beraz, oso erraza da bata bestearen atzean dauden zati bi hizlari berdinarenak izatea. Lineazko algoritmo honek erabaki lokalagoak egiten ditu, bata bestearen ondoan dauden zatien multzokatzea areagotuz.

3.4. Ahostun eta ahoskabeen antzematea

Lehenago aipatu denez, trama ahostunak bakarrik kontuan hartuta egiten da txanden banaketa. Honetarako, trama ahostunak ahoskabeetatik banatzeko sistema bat ere inplementatu da, PTHCDP algoritmoa erabiliz (Luengo eta beste, 2007). Algoritmo honek cepstrum parametroak eta programazio dinamikoa erabiltzen du F_0 eta VUV informazioa lortzeko. Linean lan egiteko algoritmoa moldatu egin da, linean dabilen Viterbia erabili dezan eta bide aktiboak bat egin bezain laster VUV informazioa eskuragarri egon dadin.

3.5. Sistemaren integrazioa

Algoritmo hauek guztiak integratzeko, eta sistema osoa linean pausu bakarrean ibiltzeko, beharrezkoa da zenbait puntu kontutan hartzea. Batez ere, azpi-algoritmo bakoitzaren emaitzak audio sarrerarekin sinkronizatu behar dira. Adibidez, multzokatze sistemak hizlari aldaketa bat aurkitu arte itxaron behar du. Aldi berean txanden banaketa bilatzeko sistema zain egon behar da hizketaren antzemate eta ahostunen antzemate algoritmoek erabakiren bat hartu arte, eta hauek asinkronoki hartzen dituzte erabakiak, bakoitzaren Viterbi bideek bategite unearen arabera. Prozesu guzti hauek martxan dauden bitartean, gainera, audio trama berriak heltzen ari dira. Audio sarrera hau eta azpisistema guztiak sinkronizatzeko zenbait buffer eta kontrol puntu erabili dira.

Audio sarreratik trama berri bat jasotzen den bezain laster, parametrizatu egiten da. Cepstrum parametroak VUV sistemara bidaltzen dira, eta MFCC parametroak VAD sistemara. Gainera, MFCC parametroak buffer baten gorde egiten dira geroago erabiltzeko.

Hizketaren antzemate eta ahostunen antzemate sistemen asinkronotasuna dela eta, algoritmo hauen irteerak beste bi bufferretan sartzen dira Viterbiak erabaki bat hartu bezain laster. Esan dezagun VAD irteeraren bufferrean N tramen erabakiak ditugula eskuragarri, eta VUV sistemaren bufferrean beste M. Orduan $O=min\{N,M\}$ tramentzako erabaki guztiak harturik dauzkagu, eta euren prozesamendua jarraitu daiteke. O trama hauek MFCC bufferretik atera eta hizketarik ez daukatenak eta ahostunak ez direnak bota egiten dira, besteak txanden banaketa sistemara sartzen dira. Aldi berean sinkronizatze prozesuak baztertutako tramen kokapena gorde egiten du, gero etiketak ondo jartzeko.

Txanden banaketa sistemak hizlari aldaketa bat aurkitzean, lortutako zatiaren mugekin batera txandaren estatistikak ere atera eta multzokatze sistemara bidaltzen ditu. Estatistika hauek erabilita, multzokatze sistemak zati horretan dagoan hizlariari dagokion identifikadorea ematen du.

Txanden banaketa sistemak ez ditu hizketa-bako tramak ezta ahoskabeak erabiltzen. Beraz, aurkitutako aldaketa uneak moldatu egin behar dira, baztertutako tramak kontuan hartuz. Honetaz guztiaz sinkronizazio azpisistema arduratzen da, txanden antzemateak lortutako zatien mugak bihurtuz.

4. Frogak eta emaitzen analisia

Proposaturiko algoritmoa Albayzin 2010 hizlari diarizazio erronkan (Zelenák eta beste, 2010) aurkeztu zen Aholab taldearen sistema nagusi bezala. Hala ere, beste sistema bi aurkeztu ziren erreferentzia bezala. Hauetako lehenengoa algoritmo nagusiaren lineaz kanpoko bertsioa da, eta bertan azpisistema bakoitza bata bestearen atzean abiarazten da. Bigarrena ere lineaz kanpoko bertsioa da, baina trama ahoskabe eta ahostunak kontuan hartuz txanden banaketa egiteko. Beraz, ez du VUV sistemarik behar.

Erronkan erabilitako datu baseak 3/24 Kataluniako telebistako iragarki emanaldien grabaketak ditu, 16 kHz wav formatuan, lagin bakoitzak 16 bit dituelarik. Guztira 88 ordu dira, eta zati bitan banatuta dago datu-basea: 58 ordu sistemaren garapen eta entrenamendurako eta 30 ordu frogetarako. Audioa lau orduko fitxategietan banatuta dago, fitxategi bakoitza gutxi gorabehera 90 hizlari eta mila txanda dituelarik.

Atal honetan hiru sistema hauek konparatu egiten dira zehaztasunaren eta prozesua burutzeko behar den

Denbora (s)	Lineaz kanpo ahoskabe-ekin	Lineaz kanpo ahoskabe barik	Linean ahoskabe barik
Ezabatuak	2,8	2,8	4,9
Faltsuak	1,2	1,2	1,5
Nahastuak	26,9	23,1	23,9
Guztira	31,0	27,2	30,4

1. taula: Sistema bakoitzak Albayzin 2010 diarizazio erronkan lortutako emaitzak.

Okerrak (%)	Lineaz kanpo ahoskabe-ekin	Lineaz kanpo ahoskabe barik	Linean ahoskabe barik
VAD	36,0	36,0	---
VUV	---	81,0	---
Diarizazioa	47,1	44,2	---
Guztira	83,1	161,2	126,1

2. taula: Sistema bakoitzak grabaketa ordu bat prozesatzeko erabilitako batezbesteko CPU denbora.

denboraren arabera, erronkan lortutako emaitzak erabiliz. 1. taulan sistema bakoitzaren okerrak aurkezten dira, eta 2. Taulan, ordea, prozesamenduaren abiadura.

Linean eta lineaz kanpoko sistemen arteko desberdintasunik aipagarriena post-tratamendua da. Kasu bietan erabilitako algoritmoak berdinak dira, baina lineaz kanpoko egiturak azpisistema bakoitzaren irteera post-tratatzeko aukera ematen digu, hurrengo pausoa abiatu aurretik. Adibidez, hizketaren antzematea egin eta gero, 500 ms baino laburragoak diren isiluneak kendu egin ziren. Honekin hizketaren antzemate hobea lortzen da, eta honegatik hizlari ezabatu gutxiago agertzen dira emaitzetan, 1. taulan ikusi daitekeenez.

Trama ahoskabeak alde batera uztearen eragina ere ikusi daiteke 1. taulan. Trama ahoskabeak erabiltzerakoan hizlarien arteko nahasmena %16 igo zen. Hizketaren antzemate sistemak bai trama ahoskabeak eta bai ahostunak beti erabiltzen dituelako, hizlari faltsu eta ezabatuetan ez da inolako desberdintasunik ikusten.

Oso interesgarria da baita ere sistemak abiaduraren aldetik konparatzea. 2. taulan sistema bakoitzak ordu beteko audioa aztertzeko behar duen batezbesteko denbora aurkezten da. Lineaz kanpoko egituran azpisistema bakoitzak behar duen denbora ere adierazten da. Neurketa guztiak 6 Gigako memoria duen quad-core Intel Xeon 2.27 GHz konputagailu baten egin dira. Dena dela, balio hauen helburua ez da sistemen konplexutasuna zehazki neurtzea, konparaketa erraztea baizik.

Linean dabilen sistemak iterazio bakarrean aztertzen du audio osoa, lineaz kanpoko egituran hainbat iterazio eta post-tratamendu egiten diren bitartean. Ondorioz, denbora beharrizanak oso desberdinak dira. Lineaz kanpoko egiturak 161 segundo CPU behar ditu audio ordu bat aztertzeko, lineazko sistemak 126 segundo behar dituen bitartean. Lineazko sistema %22 azkarragoa dela esan nahi du honek, gehien bat iterazio kopuruaren murrizketagatik.

Aipagarria da baita denboraren erdia ahostunen antzematean ematen dela. Trama ahoskabeak baztertzen ez badira, VUV sailkapena ez da beharrezkoa, eta prozesamendu denbora 83 segundotara jaisten da. Nahiz eta neurketarik ez izan, lineazko sisteman ere VUV sailkapena kentzekotan denbora beharrizana txikiagotuko dela espero da.

5. Ondorioak

Gaur egungo hizlari diarizazio sistema gehienak lineaz kanpoko egitura baten oinarrituta daude, hainbat azpisistema bata bestearen atzetik exekutatzen direlarik. Artikulu honetan linean lan egiteko gai den sistema bat aurkeztu da. Algoritmo honek iterazio bakar bat egiten du audioa prozesatzeko, eta beraz, mikrofonotik zuzenean hartutako audioarekin lan egiteko gai da. Audioa grabatzea ezinezkoa den ingurune baten ere erabili daiteke.

Prozesu guztia iterazio bakarrean egin behar denez, ezinezkoa da azpisistema bakoitzaren irteera post-tratatzea hurrengora sartu baino aurretik. Beraz, zehaztasunaren nolabaiteko murrizketa bat ezinbestekoa da. Egindako frogak erakusten dute lineazko sistema honekin zehaztasuna %12 bat jaitsi dela lineaz kanpokoarekin konparatuta. Murrizketa hau batez ere hizlari ezabatuen kopurua igo dalako izan da, eta hau hizketaren antzematearen irteera post-tratatu ez izanagatik gertatu da.

Dena den, sistema iterazio bakar baten abiarazteak bere abantailak ditu prozesamendu abiaduran. Lineazko egitura %22 azkarragoa da lineaz kanpokoa baino.

Txanden antzematean trama ahoskabeak kontuan hartu behar diren ala ez ere aztertu da. Alde batetik, trama hauek baztertzeak zehaztasuna hobetu egiten du. Beste alde batetik, trama hauek baztertzeko VUV azpisistema bat beharrezkoa da, eta honek sistemaren abiadura asko moteltzen du. Adibidez, frogetan erabilitako VUV sistemak (Luengo eta beste, 2007) prozesamendu denboraren erdia suposatzen du. Diarizazioa prozesamendu ahalmen murriztua duten gailuetan erabili nahi izanez gero, VUV sistema azkarragoren bat inplementatu beharko litzateke, edota guztiz baztertu eta bai trama ahostunak eta ahoskabeak erabili.

6. Esker onean

Lan honeku Espainiako Ministerio de Ciencia e Innovaciónen finantziaketa jaso du, Buceador proiektuaren barruan (TEC2009-14094-C04-02) eta baita Eusko Jaurlaritzarena BERBATEK proiektuaren barruan (IE09-262).

7. Aipamenak

Reynolds, D. A.; Torres-Carrasquillo, P.(2004). The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations. *NIST Rich transcription Workshop*.

Sinha, R.; Tranter, S. E.; Gales, M. J. F.; Woodland, P. C. (2005). The cambridge university march 2005 speaker diarisation system. *Interspeech*: 2437–2440.

Chen, S. S.; Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. *DARPA speech recognition workshop*: 127–132.

Zhou B.; Hansen, J. (2000). Unsupervised audio stream segmentation and clustering via the bayesian information criterion. *ICSLP*: 714–717.

Meignier, S.; Moraru, D.; Fredouille, C.; Bonastre, J. F.; Besacier, L. (2006). Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech and Language* 20: 303–330.

Delacourt, P.; Wellekens, C. J. (2000) DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication* 32: 111–126.

Tritschler, A.; Gopinath, R. A. (1999). Improved speaker segmentation and segments clustering using the bayesian information criterion. *Eurospeech*: 679–682.

Cettolo, M.; Vescovi, M. (2003). Efficient audio segmentation algorithms based on the bic. *International Conference on Acoustics, Speech, and Signal Processing (ICCASP)*: 537–5340.

Šrámek, R. (2007). The on-line Viterbi algorithm. Master tesia, Comenius University, Bratislava.

Luengo, I.; Saratxaga, I.; Navas, E.; Hernáez, I.; Sánchez, J.; Sainz, I. (2007). Evaluation of pitch detection algorithms under real conditions. *International Conference on Acoustics, Speech, and Signal Processing (ICCASP)*: 1057–1060.

Zelenák, M.; Schulz, H.; Hernando, J. (2010). Albayzin 2010 Evaluation Campaign: Speaker Diarization. *VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*: 301–304.

Duda, R. O.; Hart, P. E. (2001). *Pattern Classification*. John Wiley and Sons.

Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77: 257–286.

Tranter, S. E.; Reynolds, D. A. (2006). An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech and Language Processing* 14: 1557–1565.

Chen, S. S.; Gopalakrishnan, P. S. (1998) Speaker, Environment And Channel Change Detection And Clustering Via The Bayesian Information Criterion. *DARPA speech recognition workshop*: 127–132.