Konputazio-Zientziak eta Adimen Artifiziala Saila

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

Informatika Fakultatea

BERTSOBOT:

Gizaki-Robot Arteko Komunikazio eta Elkarrekintzarako Portaerak

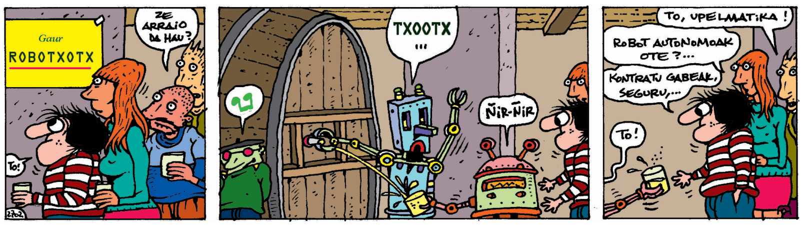
Aitzol Astigarraga Pagoagak

Elena Lazkano eta Basilio Sierraren zuzendaritzapean

Informatikan Doktore titulua eskuratzeko aurkeztutako

TESI-TXOSTENA

Donostia, 2017ko ekaina



Antton Olariaga, *Berria* (2012ko martxoak 7)

Etenik gabe ura ematen
preskua eta garbia
joaten danentzat kentzen dakizu,
berakin dun egarria.
Garbitu leike aurpegia ta
oinak bezela gerria
hitz hutsez ezin esplika leike
iturri maitagarria.

Roman Maiz "Okelar", *Okelarko iturriari*

(...) robot fikzio sortzaile bati jaten eman zion
hamar hitzeko sinopsi batekin eta, iritziz aldatu zuenean, nobela
jada argitaratuta zegoela jakin zuen.

Philip K. Dick, *Irteerako ateak barrura darama*

Eskerrik asko!

Tesi lan hau egiterakoan, gazteagotako txirrindulari garaiak biziberritu ditut. Eta ez bakarrik fakultatera bizikletaz atzera-aurrera asko egin behar izan ditudalako. Azken baten, zer da ba tesia, itzuli handi bat ez bada?

Itzuli guztietan bezala, izan dira amaierarik gabeko etapa luze aspergarriak, hasiera partean bereziki. Itsu-itsuan egindako esprint beldurgarriak ere bai, tarteko helmugara posizio onean iristeko ahalegin arnas-gabeak. Nola ez, mendiko etapatzarrak, epikoak, mendate-magaletan gora begira jarri eta tontorrak lainopean ikusten genituela. Eta amaitzeko, erlojupekoaren agonia: bakarkakoa, luzea, erlojuaren eta norbere buruaren aurka, pedalkada bakoitza, metro bakoitza, neurtuz eta sufrituz.

Noski, horrelako itzuli bati nekez egin dakioke aurre bakarrik. Talde oso bat izan du aldamenean balentria hau egin ahal izateko, eta guztiei eskerrak eman nahi nizkieke:

- RSAIT txirrindulari-taldeko kideei, emandako animo eta aholkuengatik. Batez ere Otzetari eta Igorri, zuen bultzadarik gabe oraindik helmugatik urrun nengoke.

- Elena eta Basi zuzendariei. Taldeko zuzendari orokorra bata, helburuak eta estrategiak zehazten dituena; kirol zuzendaria bestea, helburuetara iristeko norberaren onena ateratzen dakiena.

- Zuri Manex, lasterketa elkarrekin hasi genuelako eta beti, beti, laguntzeko prest agertu zarelako.

- Bertsozale Elkarteari, kilometro guzti hauek egiteko bizikleta eskaintzeagatik.

- Gurasoei, etxekoei, errepidetik kanpo emandako laguntza guztiagatik.

- Bide bazterretik animoak eman eta, epaileak begira ez zeudela, bultzatu nauzuen guztiei. Ez zarete gutxi izan!

- Dorleri: azken bolada luzean emandako laguntzagatik eta, batez ere, laguntzeko moduagatik. Hor egoteagatik.

Azken metro hauetan, guztiok zaudete nire gogoan. Begiak izerdiarekin erreta dauzkat eta ezin dezaket ondo bereizi, baina bai, hortxe behar du helmugak. Jendearen oihuak sumatzen ditut. Tentetu naiz bizikletan, mailotari tira egin eta hainbeste aldiz entsegatu bezala jaso ditut bi eskuak zerura.

Bakar bakarrik pasa dut helmuga. Ederra izan da, eskerrik asko guztioi. Zuen laguntzari esker lortu dut... kontrolez kanpo ez iristea.

Laburpena

Robot adimendunak garatzeko beharrezko diren hainbat gaitasun jartzen ditu jokoan bertsolaritzak. Sormena, komunikazio-gaitasuna, hizkuntza eta munduari buruzko jakintza eta bat-batean erantzuteko erreflexuak besteak beste. Eredu gisa bertsolaria hartu dugu eta jendaurrean aritzeko gaitasuna erakutsiko duen robot-autonomoa garatzea izan da gure ikerketa-lanaren helburu behinena. Bere egitekoa, bertsoa osatzeko instrukzioak ahoz jaso, hauek prozesatu eta ahalik eta bertsorik egokiena osatu eta kantatzea izanik, bertsolarien oholtza gaineko adierazkortasun-maila erakutsiz gorputzarekin. Bertsolaritzaz gaindi, gizaki-robot arteko komunikazioan pauso bat eman nahi genuke aurrera ikerketa-lan honekin, elkarrekintza sozialera bideratutako robot-arkitektura proposatuz. Bertsobotak, barne hartzen ditu jendarteko egoeratan beharrezko diren hizketa-bidezko komunikazioa, bat-bateko mezu-sorkuntza, gorputz adierazkortasuna eta emozioen pertzepzioa eta adierazpena.

Gaien Aurkibidea

I. atala

Sarrera

1. Kapitulua

Sarrera

1.1 Motibazioa eta eszenatokia

2012ko apirilaren 18an, Euskal Herriko Unibertsitateko errektore taldeak bertso-saio berezia antolatzeko gonbita luzatu zigun: bertsolari eta makinen artekoa, alajaina! Erronkari baiezkoa eman eta Informatika Fakultateko hainbat ikerlari elkar lanean jarri ginen, Bertsobot-aren lehen prototipoa plazaratuz. Bi robot, *Galtxagorri* eta *Tartalo*, aritu ziren saio berezi horretan bertsolariekin batera, eta saioak arrakasta handia izan zuen medioetan. Jendaurrean eta bat-batean egindako saio esperimental horretatik ikasgai garrantzitsuak atera genituen, zinez: gizaki-robot arteko elkarrekintzaz, publikoaren robotekiko enpatiaz, eta baita testu-sorkuntza automatikoari buruzkoak ere.

Baina garrantzitsuena, orduko saioak balio izan zigun jolas gisa planteatutakoari serio heldu, eta bertsolaritza, lengoaia naturalaren prozesamendua eta robotika uztartuko dituen ikerketa-lerroari hasiera emateko.

Ikerketa-lan honek biltzen du, ordutik gaur arte, RSAIT[1] taldearen baitan Bertsobotari buruz egindako lana.

Helburu praktikoa jarri diogu ikerketa-lerroari: robot-bertsolaria, *Bertsobot*a, diseinatu eta garatzea. Bere egitekoa, bertsoa osatzeko instrukzioak ahoz jaso, hauek prozesatu eta ahalik eta bertsorik egokiena osatu eta kantatzea izanik, bertsolarien oholtza gaineko adierazkortasun maila erakutsiz gorputzarekin.

[1]http://www.sc.ehu.es/ccwrobot

Helburu horretara bidean, bi faktore ezberdinek motibatu dute gure ikerketa-lana: lehenbizikoa, Adimen Artifizialaren sorrerako ametsari lotua, bertsolaria eta bere jendaurreko ariketa eredu gisa hartu eta robot-bidez gorpuztea bilatzen duena; robot adimentsuen etengabeko bilaketa berrituz. Bigarrena, teknikoagoa, gizaki-robot arteko elkarrekintza eta komunikaziorako arkitektura berri baten bilaketari lotua dago. Ahots-seinalearen ulermena, ordenagailu bidezko ikusmena bertsolariak eta oholtza gaineko beste elementuak ezagutzeko, nabigazioa, elkarrekintza publikoarekin, gorputz-espresioa eta antzeko gaitasunen diseinu eta inplementazioari dagozkionak.

Bertsobotaren muinean ondoko arloak topatuko ditugu:

Bat-bateko bertsolaritza

Bat-bateko bertsolaritza (Garzia et al., 2001) aukeratu dugu eredu gisa. Alde batetik bertsoak, bat-batekoak, lengoaia naturalaren prozesamenduaren ikuspegitik interesgarriak diren ezaugarriak biltzen dituelako. Algoritmo-bidez deskribatu eta landu daitezkeen murrizpenak dauzkalako eta, horren baitan, askatasun eremu bat esan nahi duguna esan ahal izateko. Bestetik, bertsolaria bera, robotikarako eredu kognitibo gisa interesatzen zaigu: nola mugitzen den, nola hautematen duen ingurunea eta jasotako informazio guztia nola prozesatzen duen bat-bateko sorkuntzan txertatzeko.

Poesia Sorkuntza Automatikoa

Konputagailua bere kabuz poesia sortzen jartzeko ideia Alan Turing-ek iradoki zuen 1950. urtean (Turing, 1950), makinen inteligentzia neurtzeko bere test ezaguna ilustratu asmoz. Ordutik hona ikerketa-eremu beregain bihurtu da eta egun sistema ugari aurki ditzakegu modu autonomo edo erdi-autonomoan poesia sortzeko gai direnak (Oliveira, 2015), (Gervás, 2015). Eredu horiek aztertuko ditugu arretaz eta ikusi, ea bat-bateko bertsoaren eremura ekartzeko modukoak diren.

Ikasketa Automatikoa

Ikasketa Automatikoa (*Machine Learning*, ML) (Mitchell, 1997) emandako datu-multzoetatik konputagailuek ikasteko gaitasuna garatzea helburu duen informatikaren alorrari dagokio. Datu-multzoetatik informazio ulergarria eta erabilgarria lortzeak garrantzia itzela du, makinak ere giza adituek erakusten duten trebeziarekin problemak ebazteko eta erabakiak hartzeko gai izatea nahi bada. Ondo finkatutako alor honetako algoritmoak baliatuko ditugu bertsoaren egiturari buruz ikasteko.

Semantika

Bat-batean bertsoak osatzea ez da nolanahiko zeregina. Arau formalak betetzeaz gain, sortutako testuak zentzua izan behar du, mezu jakin bat helarazi, emozioak transmititu. Horretarako beharrezkoa da bertsoaren testua, diskurtsoa, semantikoki koherentea izatea. Semantikarako bektore-espazioa (Salton et al., 1975) baliatuko dugu horretarako, bertsoetako puntuak bektore-gisa adierazi eta elkarrekiko hurbiltasun semantikoa kalkulatuz.

Konputagailu-burmuin interfazeak

Neurozientzia Kognitiboaren eta Burmuin Irudigintzaren (*Brain Imaging*) alorreko aurrerakuntzek, giza burmuinarekin zuzenean elkarreragiteko ateak zabaldu dizkigute. Azken urteotan, konputagailu-burmuin interfaze (BCI, *Brain Computer Interface*) ugari garatu dira (Guger et al., 2015), beren helburu nagusia burmuinaren seinale bidez adierazitako intentzioa jaso eta kontrol-agindu bihurtzea izanik.

Jendearen emozioak jasotzeko bide-ezberdin ugari daude, baina zinez interesgarria iruditzen zaigu automatikoki publikoaren emozio-egoera detektatu ahal izatea. Garunean kokatutako sentsore ez inbasiboekin emozioak detektatu ahal izateak, gizaki-robot elkarrekintzarako bide berriak zabaltzen lituzke.

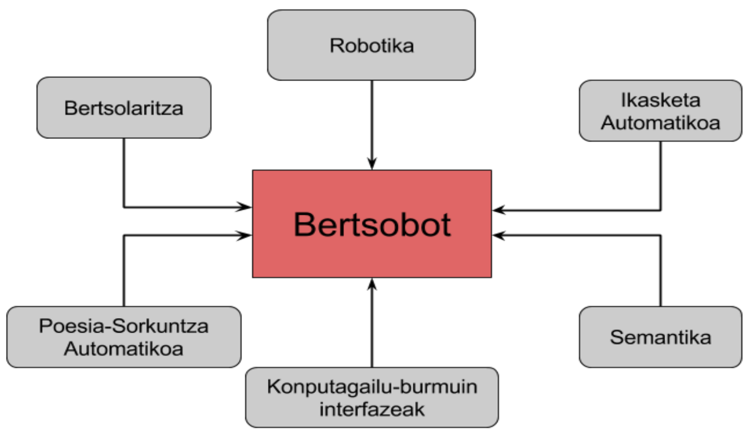
Robotika

Robotikaren zeregina izango da aurreko eremu guztiei gorputza jartzea. Norbaitek esan lezake bertsoa automatikoki sortzeko ez dagoela robotaren beharrik, edozein ordenagailutan exekutatu daitekeela. Hori da, hain zuzen ere, konputagailu-bidezko poesiaren alorrak egiten duena. Baina gu ez gatoz bat. Bertsolaria eredu kognitibo gisa hartuz gero, hura ingurunetik atera eta sorkuntza-prozesua konputagailuko programa batera mugatuz, bat-bateko bertsoaren ezaugarri garrantzitsuenetakoa albo batera uzten ariko ginateke. Alegia, bertsolariak ingurunetik estimuluak jaso eta horiek bertsotan txertatzeko erakusten duen gaitasuna.

Horrekin batera, bertsolaritzan jendaurreko aurkezpenak berebiziko garrantzia dauka, bertsolari eta ingurunearen arteko elkarrekintza testua bera bezain garrantzitsua da. Horretarako ezinbestekoa da gorpuzkera fisikoa, robota alegia.

Arlo ezberdinetako zientzialariak bat datoz funtzio kognitiboak funtzio sentsorial-eragileetan txertatzearen beharraz hitz egitean (Brooks, 1991), (Lakoff and Johnson, 1999). Kognizio eredu artifizialak sistema robotiko errealetan inplementagarriak izan behar dira beraz, portaera adimentsua agertu dadin sistemak mundu errealean txertatua egon behar du. Gorputza behar du. Honek, robotikaren ikuspegitik, eskatzen du eredu kognitiboa sentsoreetara konektatu ahal izatea eta eragileak denbora-errealean kontrolatzea, eta guzti hori begizta itxian burutzea.

1.1 irudiak grafikoki erakusten du eremu ezberdinetako ekarpena.



1.1 Irudia: Bertsobotaren muinean dauden ikerketa-eremuak

1.2 Helburuak

RSAIT ikerketa taldeak, robotikaren baitako beste alor batzuk landu ditu orain artean, batez ere: kontrol-arkitekturak, nabigazioa, ikasketa automatikoa eta ikusmen artifiziala. Laborategi inguruetan modu seguruan nabigatzea izan da gure helburu nagusietakoa, oztopoak ekidin, helburuak identifikatu, leku berriak esploratu, etab. Lan honekin, aurretik egindakoari geruza berri bat erantsi nahi genioke, gizaki-robot arteko elkarrekintzarena. Izan ere, laborategietatik atera eta ingurunea gurekin konpartituko badute, gure agindu eta mugimenduak interpretatu eta modu egokian erantzuteko gaitasuna behar dute robotek.

Elkarrekintza eta komunikazio horretan robotak portaera edo gaitasun ugari agertu behar ditu denbora errealean: hizketaren ezagutza, aginduen identifikazioa, aurpegien detekzio eta jarraipena, oztopoen detekzioa, gorputz espresioa, etab. Aldi berean eta modu koordinatuan exekutatu beharreko gaitasun ugari inondik ere.

Robotak laborategietatik atera eta jendearen zerbitzura jartzea dugu motibazio nagusi. Dela museoetan gida lanak egiteko, adineko edo ezinduei errehabilitazio lanetan laguntzeko edo, zergatik ez, astialdian lagungarri izan daitezen. Motibazio honek, bidenabar, bat egiten du Europako Batzordeak robotika mugikorraren eremuan zehaztutako 2020 estrategiarekin[2] .

Helburu orokorra beraz, ez da bertsotan aritzeko gaitasuna erakutsiko duen robot-autonomoa garatzea. Bertsolaritzaz gaindi, gizaki-robot arteko komunikazioan pauso bat egin nahi genuke aurrera ikerketa-lan honekin, mezuak sortu eta plazaratzeko orduan testuinguruko ezaugarriak kontuan hartuko dituen sistema eraikiz.

Helburu orokorra arloz-arloko beste hauetan zehaztu dezakegu: **Testu-sorkuntza automatikoari dagokionez**:

- Bertsoak osatzeko metodo automatikoen garapena. Bat-bateko bertsolaritzan ohikoak diren ariketak inplementatzeko estrategiak.
- Bertsoaren semantika eta diskurtso-egitura narratiboa aztertu eta eredu konputazionalak eraiki mezu berriak egituratu eta formulatzeko. Ikasketa Automatikoko tresnak aplikatuz.
- Testu-corpusen azterketa Ikasketa Automatikoa baliatuz.
- Bertso-sorkuntzan lagungarri izango diren tresnen garapena: errima-bilatzailea, silaba-kontatzailea, doinu hautatzailea eta abar.

Robotikari dagokionez

- Gizaki-robot arteko hizketa-bidezko elkarrizketa sistema. Bertso-saio bateko testuinguru mugatuan, gai-jartzaileak esandakoa jaso eta erantzuteko gaitasuna.
- Robot sozialen garapena, humanoide gisako roboten portaerak landuz. Bertsolariek oholtza gainean erakusten duten gorputz adierazkortasuna lantzea robotean. Emozioak pertzibitu eta erakusteko ahalmena.
- Bertso-saio batean, publikoaren erreakzioak jaso eta horren baitan erantzuteko ahalmena garatzea.

[2] http://ec.europa.eu/digital-agenda/en/robotics

1.3 Ekarpenak

Ikerketa-lan honen ekarpenak nagusiak ondokoak dira:

- Ikerketa-lan honek bertsolaria eredu kognitibo gisa hartu eta Bertsobot izeneko arkitektura aurkezten du. Bertsobota gai da bertsolariekin edo beste robotekin ohiko bertso-saio batean parte hartu eta kantatzeko. Arkitektura, bat-bateko bertsolaritzak eskatzen dituen zehaztasunen eta bertsolariak saioan erakusten duen portaeraren arabera eraikia izan da, oholtza gainean oinarrizko nabigazioa, bat-bateko mezuen sorkuntza eta emozio-egoeraren araberako gorputz adierazpidea erakutsiz. Arkitektura modularra da, gaitasun berriak inplementatu ahala aurrekoekin koordinatu eta integratzeko pentsatua. Azken ezaugarri honek arkitektura bera beste eremu batzuetara zabaldu ahal izatea dakar, zaintzara bideratutako robotika, museotarako gida edo aisialdirako robot gisa, besteak beste.

- Ikerketa-lan honek bertso-sorkuntza automatikorako bi estrategia berri proposatzen ditu: N-grama ereduak erabiltzen dituena bata eta corpusetik esaldiak atera eta horiek konbinatzen dituena bestea. Oinarrizko bi metodoetarako, bertsoaren koherentzia semantikoa neurtu eta maximizatzeko proposamenak garatu ditugu baita. Azkenik, aldarte-egoera txertatu dugu N-grama bidezko sortzailean, zehaztutako aldartearekin (positibo, negatibo edo neutroa) bat datozen bertsoak sortu ditzan. Sorkuntza-estrategiak eta semantika eredua metodologia orokor baten baitan bildu ditugu eta tresna gisa eskaini, baliabide linguistiko minimoekin hizkuntza ezberdinetan bertso edo poemak sortzeko modua eskaini asmoz.

- Ikasketa Automatikoari dagokionez, eremu honetako ekarpenik behinena, bertsoen egitura diskurtsiboa Ikasketa Automatikoko tresnen bidez atzemateko ahalegina litzateke. Batetik, agur-bertsoetako ezaugarriak adituen laguntzaz identifikatu eta, ondoren, esperientzia hori orokortu eta kasu berrietan automatikoki aplikatu ahal izateko ahalegina. Bestetik, diskurtso-antolaketaren ataza poesia edo bertsoen eremura ekarri izana. Hau da, puntuak desordenatuta eman eta ordenazio egokia identifikatzeko ahalegina. Bi atazetan gainbegiratu gabeko sailkapena erabili da, bigarrenean harago joan eta multi-sailkatzeekin probatu dugularik.

- Semantikarako Bektore-espazio ereduan oinarrituz, bertsoaren koherentzia neurtzeko metodoen inplementazioa.

- Konputagailu-burmuin interfazeen arloan, Ikasketa Automatikoa baliatuz sentsoreen kokapen optimoa aurkitzeko metodo berri baten proposamena. Aurrera begira, publikoaren emozioak detektatu eta interpertatzeko baliatu daitekeena.

- Lanik gehienak ingelesez argitaratu ditugun arren, ikerketa-txostena euskaraz idatzita dugu, euskal komunitate zientifikoari begira idatzia izan delako. Euskaraz egin dugu lan eta helburu akademikoekin batera, euskara hizkuntza handiagoen pare jartzeko ahalegina ere egon da. Esan dezakegu Bertsobota dela hizketa-bidezko elkarrekintza gaitasuna euskaraz erakutsi duen lehen robota. Ez da gure ekarpena soilik izan, ez horixe. Bertsobotak beste ikerketa-eremu batzuetako tresnak integratzen baititu. Aholab taldearen AhoTTS sistema eta IXA taldearen lematizatzailea kasu.

1.4 Txostenaren antolaketa

Esku artean duzun txostenak egindako ikerlanaren fruitu diren argitalpen garrantzitsuenak biltzen ditu. Haietan deskribatzen dira lanaren xehetasunak eta lortutako emaitzak. Nazioarteko kongresu eta aldizkarietan argitaratuak izan dira eta, beraz, komunitate zientifikoaren esku daude. Lortutako emaitzetatik, onetatik eta ez hain onetatik, ikasi eta aurrera egiteko itxaropenez ondu dugu lan hau. Gure ekarpena arloan abiatu diren (edo abiatzekotan diren) zientzialarientzat lagungarri izatea besterik ez dugu desio.

Txostena hamabi ataletan banatuta dago eta haietan biltzen diren zazpi kapituluetan jasotzen da egindako lana. Lehen atala SARRERA da, eta esku artean duzun I. kapitulu honek osatzen du. Ikerketa-lan hau abiatzearen motibazioa zein izan den azaltzearekin batera, ikerketa-lanarekin lortu nahi izan diren helburuak, ekarpenik garrantzitsuenak eta txostenaren egitura azaltzen dira. Bigarren atalak OINARRI TEORIKOA izenburupean sei kapitulu biltzen ditu. Bertsobot ikerketa-proiektuan inplikaturiko arlo guztien berri ematen da kapitulu horietan, bakoitzetik proiekturako interesgarriak diren ideia eta ekarpenak jasoz.

Hirugarren atala APLIKAZIO EREMUAK biltzen dituena da, eta lau kapitulutan antolatuta dago. VIII. Kapituluan “Bertso Sorkuntza Automatikoa”, IX. Kapituluan ”Diskurtsoaren Azterketa Ikasketa Automatikoa erabiliz”, X. Kapituluan “Bertsobot Arkitektura Orokorra”eta XI. Kapituluan “Konputagailu-burmuin interfazeak”. Lauek antzerako egitura dute: hasteko ataza deskribatzen da eta ondoren problemaren ebazpena nola planteatu den azaltzen da. Kapitulu bakoitzaren amaieran biltzen dira egindako lanarekin argitaratutako artikuluak. Esperimentuen inguruko xehetasunak eta emaitzak bertan aurkituko ditu irakurleak.

Txostenaren amaieran, ONDORIOAK ETA ETORKIZUNERAKO LANA atalean XII. kapituluak biltzen ditu ondorio nagusienak eta ikerkuntzarako zabalik geratu diren zenbait bide.

II. atala

Oinarri teorikoa

2. Kapitulua

Bat-bateko bertsolaritza

Bertsolaritza, ahozko euskal literaturaren baitako generoa da (Lekuona, 1974), (Lekuona, 1982), (Mitxelena, 1988). Bi sail nagusi ditu bertsolaritzak, bat-bateko bertsoak eta bertso jarriak edo idatziak. XX. mende hasieran idatzizkoak indar handia izan bazuen ere, mende amaierarako egungo bat-bateko bertsolaritza gailentzen hasi zitzaion. Bat-bateko eta idatzizkoaren ezaugarriak ez dira berdinak. Guk hemen bat-bateko jarduna izango dugu aztergai.

Gaur egun zabalduen dagoen definizioak, bat-bateko bertsolaritza erretorika eta komunikazioaren alorretan kokatzen du batez ere (Garzia, 2000). Garziak *erretorika*-ren ahozko genero bezala definitu zuen, ahozkoa, kantatua, inprobisatua eta bereziki literarioa ez dena. Hau da, bere esanetan, bertsoa ahozkotasunetik gertuago dago literaturatik baino, eta egoera bakoitzean entzulearengan emozioak sorraraztea da bertsoaren helburua. Bertsoaren alderdi teknikoa doinua, neurria eta errimak osatzen dute. Hala ere, egungo bertsolaritza ulertzeko erreferentziazko liburuan (Garzia et al., 2001) esaten den bezala, "Bertsoari kalitate maila bere arrazoibidearen indarrak edo bere balio poetiko-erretorikoek emango diote".

Bat-bateko ahozko inprobisazioa ez da Euskal Herriko fenomenoa soilik. Munduko herrialde askotan ezagutzen den gertakaria da. Aipagarriak dira, besteak beste, Argentinako payadoreak, Kubako dezimistak, Portugalgo kantadoreak, Eskandinaviako kantari repentistak, Mexiko, Britainia, Gales, Galizia, Andaluzia, Menorca eta besteetako inprobisatzaileak.

2.1 Ezaugarriak

Bat-bateko jardunaren ezaugarri nabarmenenak ondokoak dira:

- Bat-batekotasuna: bi une guztiz berezik eta desberdinek bat egiten dute, sortzeak eta interpretatzeak, alegia. Errepikaezina den momentua sortuz.

- Jendaurrean: jendaurrean egiten dira beti bat-bateko saioak eta publikoa ezinbesteko osagaia da, horrek dituen eskakizun guztiekin: bertsoa kantatu egin behar da, bertsoa jendarteratu egin behar da, antzeztu. Publikoarengana iritsi behar da bertsolaria, entzulea irabazi edo liluratu behar du.

- Hizkuntza: euskaraz ari da bertsolaria. Hizkuntza-ezaugarriei dagokionez, ahozko hizkeraren ezaugarriak ditu bertsoak ere. hizkera bizia eta azkarra; elipsi eta laburdurak ohikoak.

2.2 Alderdi teknikoa

Doinua, errima eta neurriak osatzen dute bertsoaren alderdi teknikoa. Kantatuz, errimatuz eta neurtuz burutzen den berbaldia baita bertsoa (Garzia et al., 2001).

Metrikari dagokionez, bertsolaritzaren unitatea bertsoa da, ahapaldia edo estrofa esaten dena.

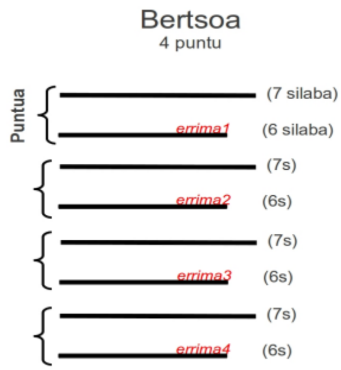
Azter ditzagun banan-banan bertsoaren alderdi formalari dagozkion hiru arau nagusiak:

Doinua

Bat-bateko bertsoa, sortu bezala, kantatu egin behar du bertsolariak eta horretarako doinua behar da. Bertsolaria musika-tresnaren laguntzarik gabe aritzen da bertsotan. (Garzia et al., 2001) liburuan esaten denez, “komunikazioaren arrakasta edo porrota ez dago bertsolariaren ahotsaren kalitatearen baitan, aukeraturiko doinuaren eta hura kantatzeko moduaren egokitasunean baizik”.

Dorronsorok, bertso-doinuak ikertzen egindako lanean (Dorronsoro, 1995) 3040 doinu bildu eta sailkatu ditu. Lan horri Bertsozale Elkarteak jarraipena eman eta egun web-bidez kontsultatu daiteke[1].

[1] http://www.bertsozale.eus/eu/bertsolaritza/bertso-doinutegia



Haiziak bidali du
handikan honuntza
geroztik hemen dabil
gure andaluza
hargiñ ona zala ta
zakurraren putza
bera eroriko dek
ez akiyok bultza

2.1 Irudia: Ezkerrean zortziko txikiaren egitura, eta eskuinean Txirritaren bertsoa

Neurria

Bertsoa puntutan zatitzen da, eta puntu bakoitzak, neurriaren arabera, silaba-kopuru jakin bat izango du. Puntu bakoitzeko azken hitzari oin deitzen zaio. Oinek elkarrekin errimatu behar dute. Neurria zehatza da eta derrigor bete beharrekoa.

Bat-batekoan gehien erabiltzen diren neurriak hauek dira:

- Zortziko txikia: lau puntu eta puntu bakoitzak 7/6 silabako erritmoa (7 silaba, etena, 6 silaba). Ikus 2.1 irudia.
- Zortziko handia: lau puntu eta puntu bakoitzak 10/8 silabako erritmoa (10 silaba, etena, 8 silaba).
- Hamarreko txikia: bost puntu eta puntu bakoitzak 7/6 silabako erritmoa.
- Hamarreko handia: bost puntu eta puntu bakoitzak 10/8 silabako erritmoa.

2.1 irudiak zortziko txikiaren egitura adierazten du grafikoki.

Errima

Esan bezala, bertso edo ahapaldiaren oinek elkarrekin errimatu behar dute. Errima, hitzen hoskidetasunean oinarritzen da, bertsoaren ardatza izanik. Oinen arteko hoskidetasun hori -ez beti- azkeneko bi silabek osatzen dute, horretan irizpide zorrotz-zorrotzik ez badago ere. Azkeneko bi silabek zenbat eta bokal eta kontsonante gehiago eduki, hainbat eta aberatsagoa izango da errima. Adibidez, kalea eta betea hoskideak dira baina errima bezela ez da oso aberatsa. Aldiz, elizan eta gerizan errima aberatsa da; (an) atzizkia errima-kide izateaz gain, aurreko kontsonantea (z) eta horren aurreko bi bokalak ere (e) eta (i), bat egiten dutelako. Amurizak, Hiztegi Errimatuak lanean (Amuriza, 2016), bertsogintzarako edo poesia errimaturako balio duen errima-hiztegia, errimen antolaketa eta erabilera azaltzen ditu.

Bertsolariak esan nahi duen hura esateko errima-multzo egokia topatu behar du, beraz. Bertso berean oina errepikatzea txartzat ematen da, eta poto egitea esaten zaio. Hitz homonimoak, jakina, salbuespen dira.

2.3 Bertsoa osatzeko estrategia

Ahozko literaturaren parte izanik, literaturaren ohiko baliabideak erabiltzen dira bat-bateko bertsolaritzan ere: onomatopeiak, hitz-jokoak, metaforak, konparazioak...

Baliabide guzti horiek erabiltzeaz gain, Egañak (Garzia et al., 2001) (Egaña, 2007) lanetan zehazten duen gisan, bertsolariak badauka entzulearen arreta bereganatzeko estrategia, bertso oro erabiliko duena: lehenbizi bukaerako arrazoi edo mezua pentsatzea. Gero, bukaera emango dion ideia nagusiaren arabera tolesten dira beste guztiak.

Lehenbizi, beraz, zer esango duen pentsatu eta ondoren nola esango duen erabakiko du bertsolariak, arrazoirik sendoena bukaerarako utziz.

Bertsoak osatzeko prozedura pausuz-pausu hauxe litzateke:

- Bertsoaren mezu nagusia aukeratu eta bukaerarako gorde.
- Mezua bertsoaren moldera (metrika) egokitu.
- Errima-multzoa lortu.
- Hasieratik bertsoa osatzen hasi: unean-unean errima bat aukeratu eta puntua osatuz.

2.4 Bertsoaren eszenaratzea

Bertsolaritza jendaurreko ekintza da, publiko aurrean sortu eta plazaratzen du bertsolariak bere lana. Bertsoaren eszenaratzea, beraz, garrantzi handikoa da. Hala ere, Olasagastik dioenez (Olasagasti, 2004), plaza gehienen oinarrizko ezaugarriak antzekoak dira: oholtza, publikoari begira lau, sei edo zortzi aulki. Aurrean, gehienetan, bi mikrofono bertsolariek bertatik kanta dezaten. Taula gaineko mugimenduei dagokionez,denak batera eta lerroan sartzen dira, nahiz ez oso formala izan. Irteera ere berdin egingo dute. Zutik mantenduko dira oholtzan agurra kantatu arte eta gero eseri egiten dira banan bana. Gai-jartzailea alde batera edo bertsolarien segidan jartzen da, eta kantatu behar duenean bertsolaria mikrora hurbiltzen da. Bertsolariak kantatzen duen bitartean bere kideei bizkarra emanez egiten du. Hori da gutxi asko saio arrunt bateko eszenaratzea.

Ikus 2.2 irudia, bertso-saio ezberdinetako eszenaratzea konparatzeko.

2.5 Ondorioak

Bertsolaritzaren ezaugarriak aletu nahi izan ditugu kapitulu honetan, bai bertsoaren ezaugarri teknikoari dagozkionak eta baita jendaurreko ekintzari buruzkoak ere. Artearen egoera aztertzean jabetu gara, akademiak bertsolaritzarekiko interesa erakutsi duela azken urteotan. Horren erakusgarri dira argitaratu diren ikerketa-lanak, ertz ezberdinetatik bertsolaritza ikertu dutenak: komunikazio-zientzien ikuspegitik (Garzia, 2000), genero ikuspegitik (Lasarte et al., 2015) (Hernandez, 2011), antropologia alorretik txapelketari buruzko analisia (Larrañaga, 2013) (Armistead and Zulaika, 2005), eta bertsolaritzaren taxonomia proposatzen duenik ere bai (Paya Ruiz, 2013), nabarmenenak aipatzearren. Konputazio-zientziek ere bihurtu dute ikerketa-helburu, horren erakusgarri (Agirrezabal, 2011), (Lizarazu, 2014) eta (Osinalde, 2013).

Bertsolaritzak, ikerketa-gai gisa, interesa piztu du beraz. Gurea ere bai. Gizaki-robot elkarrekintza aztertu eta robotika sozialean ekarpenak egiteko agertoki ederra eskaintzen baitu. Puntu honetara iritsita, batek baino gehiago galdetu izan digu: "zergatik bertsolaritza?"Izan ere, aukeratu genezakeen hizketa librean arituko den robota garatzea; bertsotan egin ordez, antzezlanetan parte hartuko duena, jendaurrean istorioak kontatuko dituena edo irakasle funtzioak beteko dituena. Horietako edozeinek balio lezake gure helburuetarako. Hala ere bertsolaritza aukeratu dugu, ondoko arrazoiengatik:

- Gizaki-robot elkarrekintza aztertzeko esparru egokia eskaintzen du bertso-saioen ikuskizunak. Izan ere, mundu errealaren konplexutasunak (dinamikoa eta ziurgabea) ikaragarri zailtzen du helburu orokorreko robota eraikitzea (Jones, 2004). Ondorioz, bertso saioak eredu gisa hartuz, ikuskizun edo gertakariaren eszena zehaztea

(a) Gpuntua 2011, Lasarte-Oria. Argazkilaria: Alberto Elosegi

(b) Bertsolari Txapelketa Nagusia 2013: Gernika-Lumo. Argazkilaria: Gari Garaialde

(c) Gipuzkoako Eskolarteko Txapelketa 2015: Orio. Argazkilaria: XDZ

2.2 Irudia: Bertso-saioetako eszenaratzeak. Iturria: XDZ gizaki-robot elkarrekintza aztertzeko baliagarria da, elementu eta mekanismo errepikakorrek roboten nabigazioa eta funtzionamendua errazten dutelako, beti ere elkarrekintza sozialak berekin duen dinamismoa galdu gabe (Fernández and Mauricio, 2016).

- Hizketa librea bere horretan larre zabalegia da. Bertsoaren betekizunek eremua mugatzen digute, eta horrek, konputazionalki bere bentajak dauzka.
- Bertsoaren betebehar tekniko diren neurria eta errima, konputazionalki tratagarriak dira.
- Hizketa arruntean ez bezala, bertso bidezko jardunean oso ondo zehaztuak daude elkarrizketaren ezaugarriak: noiz hitz egin eta noiz entzun, berbaldiaren iraupena, etab. Horrek mezu-truke eta elkarrizketa errazten du.
- Bertsoa bera komunikazio tresna gisa interesgarria da. Bat-bateko bertsoak prozedura zehatza dauka. Bertsolariak bere mekanismoak dauzka esan nahi duenari forma eman eta bertso bidez adierazteko (ideia nagusia bukaeran, oinen aukeraketa, puntuz-puntu osatzea...); beti ere betebehar tekniko zorrotzak jarraituz (neurria eta errima). Eredu hau, bertsoarena, interesgarria zaigu bai forma aldetik eta baita diskurtsoaren antolaketaren aldetik ere bertatik ikasi ahal izateko.
- Ahozko adierazpideen artean, bertsolaritzak leku berezia du gurean. Dokumentazio lan ederra ari da egiten Xenpelar Dokumentazio Zentroa[2], eta bertso-corpus hori eredu bikaina litzateke guretzat, bertatik ikasi ahal izateko.

[2] http://bdb.bertsozale.eus/

3. Kapitulua

Lengoaia Naturalaren Sorkuntza

Lengoaia Naturalaren Sorkuntza (NLG, *Natural Language Generation*), Adimen Artifizialaren eta Konputazio Linguistikoaren azpi-arloa da, helburu gisa lengoaia naturaleko testuak sortzeko gai diren konputagailu-programak garatzea duena (Paris et al., 2013). NLG arloko lehen aplikazio arrakastatsuak izan dira, besteak beste: datu meteorologikoetatik eguraldiaren iragarpen-txostenak idaztea hizkuntza ezberdinetan (Goldberg et al., 1994), proiektuen kudeaketa (White and Caldwell, 1998) eta airearen egoerari (Busemann and Horacek, 1997) buruzko txosten automatikoak idazten dituzten sistemak, eta bezeroen galderak automatikoki erantzuteko gai den gutun-sistema (Coch et al., 1995). Egun, aipatutakoez gain, NLG baitan aplikazio ugari topatu genitzake[1]. Ramos-Sotok gaiari buruzko ikuspegi orokor eta eguneratua azaltzen du bere lanean (Ramos-Soto et al., 2016). Sormenetik hurbilago kokatuko genituzkeen testuen sorkuntza automatikoa ere sartzen da NLG baitan, hala nola: istorioen narrazioa, txisteak edo poesia bera. Hala ere, azken hauek aparteko sailean, Sorkuntza Konputazioalaren (*Computational Creativity*) baitan kokatu izan dira.

3.1 Sorkuntza Konputazionala

Ikerkuntza-eremu berria da Sorkuntza Konputazionalarena, arloaren definizio-bateraturik oraindik erdietsi ez duena (Gervás, 2015). Boden-ek 1990 jada urrun hartan (Boden, 1990) proposatu zuen Adimen Artifizialeko ideiek lagun zezaketela pentsamendu sortzailea ulertzen. Sorkuntza prozesua

[1] https://www.aclweb.org/aclwiki/index.php?title=Downloadable_NLG_systems

bilaketa problema gisa formulatu zuen berak, espazio kontzeptualean barrena egindako bilaketa gisa hain zuzen ere.

Gaur egun, onarpen-maila handieneko definizioa, Colton-ek (Colton et al., 2012) emandako hauxe dugu seguru-asko: “sistema sortzaile artifizialak konputagailu-programak dira, kanpo-begirale inpartzial batek portaera sortzaile gisa identifikatuko lukeena erakusteko gai direnak”. Portaera sortzaile horiek forma ezberdinak hartu ohi dituzte: narrazioak (Gervás et al., 2005), arte bisuala (Machado and Cardoso, 2002), metaforak (Veale and Hao, 2008), neologismoak, esloganak (Tomašic et al., 2014), umorea (Valitutti, 2009), musika (Wiggins et al., 2009), edo poesia bera, arlo honetako ataza ezagun bihurtu dena azken urteotan.

Gaia geurera ekarrita, bertsotan egingo duen gailua diseinatzerako orduan, testu-sorkuntzari dagokionez, Sorkuntza Konputazionalaren eremuan poesia da erreferentziarik hurbilekoena. Izan ere, poesia eta bat-bateko bertsoa berdinak izan ez arren (Lujanbio, 2015), badituzte zenbait ezaugarri komun euren artean:

- Betekizun formalak, bai metrikari dagozkionak eta baita errimari buruzkoak ere.
- Biak ala biak onartzen dute neurri baterainoko lizentzia poetikoa, hau da, tarteka semantika eta gramatikako arauetatik urrundu ahal izatea Manurung (2003)(Oliveira, 2015).
- Emaitzak entzule edo irakurlearentzako esanguratsua izan behar du. Zehatzago esanda, emaitzak gai izan behar du jasotzailearengan emozioak sortzeko (Garzia et al., 2001).

Antzekotasun horiek gogoan, Poesia Sorkuntza Automatikoaren arloa aztertu dugu. Azken urteotan garatu diren sistema ugariek gure helburuetarako eredu egokiak eskainiko dizkigutelakoan.

3.2 Poesia Sorkuntza Automatikoa

Konputagailu-bidezko poesia sorkuntza, ikerkuntza-arlo beregaina bihurtu da Adimen Artifiziala eta Lengoaia Naturalaren Sorkuntza arloen baitan. Dena den, konputazio-zientzietako komunitatearen interesa piztu aurretik, giza-zientzietatik hurbilago zegoen jendeak ahalegin goiztiarrak egin zituen poemak automatikoki sortzeko. Aipatzeko modukoak dira, gutxienez, Queneau-ren *Cent mille milliards de poèmes*, poema lerroen bariazio edo permutazioetan oinarritutako sormen-lana (Queneau, 1961); eta, baita, Oulipo taldearen *Atlas de littérature potentielle* ere, txantiloiak erabiliz egindako poesia-sorkuntza (Oulipo, 1988).

Konputagailu-bidezko lehen poesia-sorkuntza sistema, Coltonen (Colton et al., 2012) esanetan, *Stochastische Texte* da (Lutz, 1960), txantiloi bidezko sortzea darabilena. Kafkaren *Gaztelua* obrako esaldi sorta eredu gisa hartu eta, programak, hutsuneak betetzen ditu hitzak ausaz aukeratuz.

Ordutik hona, lehenbizi tantaka eta azken aldian zarra-zarra, proposamen ugari agertu dira. Hainbeste, non hauek sailkatzeko ahaleginak egin behar izan diren. Tamalez, oraindik Poesia Sortzaile Automatikoak sailkatzeko orduan ez dago erabateko adostasunik. Ondorioz, sailkatze ezberdinak topa ditzakegu literaturan: sortutako testuaren ezaugarrien araberakoa (Manurung, 2003), eratze-prozesuan erabilitako metodoen araberakoa (Gervás, 2013), edo sorkuntza-lanari eman nahi zaion erabileraren baitakoa (Lamb et al., 2016). Sailkapenak sailkapen, proposamen berriak agertu ahala, kategorien arteko mugak lausoago bihurtzen ari dira, kategoria bakarrean sailkatzea ezinezko bihurtzeraino.

Gure interesa sorkuntza-metodoetan dagoenez, Gervasen sailkapena (Gervás, 2013), erabilitako metodoaren araberakoa, iruditzen zaigu egokiena. Ondoko 5 kategoriatan banatzen ditu poesia sortzeko sistemak:

- **Txantiloi-bidezkoa** (*Template-based*): poema-txantiloiak betetzen dituzten sistemak dira, arau sintaktiko edo eta formalak (metrika/errimari dagozkionak) betetzen dituzten hitzekin.

- **Sortu eta testatu** (*Generate and test*): ausazko hitz sekuentziak sortzen dituzte baldintza edo murrizketa erregelak jarraituz, baldintza horiek poemaren alderdi formalari edo semantikoari buruzkoa izan daitezkeelarik.

- **Eboluzioan oinarrituak** (*Evolutionary*): eboluzioan oinarritutako konputazio-teknikak dituzte oinarri.

- **Kasu-bidezko arrazoitzea** (CBR, *Case-based reasoning*): poemarekin helarazi nahi den mezuaren arabera, aurrez sortutako poemak berreskuratzen eta helburu berrirako egokitu.

- **Lengoaiaren eredu estokastikoak** (*Stochastic language modelling*): ikasitako eredu probabilistikoak erabiliz sortzen da testua.

Ondoko lerroetan, kategoria bakoitzeko lanik aipagarrienak aztertuko ditugu.

WASP (*Wishful Automatic Spanish Poet*) sistema (Gervás, 2000), problemen ebazpenerako **sortu eta berraztertu** (*Generate and Test*) paradigman oinarritua dagoena, Poesia Sortzaile Automatikoa garatzeko lehen ahalegin serio gisa kontsideratua dago . Espainiar poesia klasikoko moldeetan sortzen ditu poemak, erromantzeak eta hiruko zein lauko nagusien erako estrofak osatuz. Bertso edo poema multzo bat (corpusa) erabiltzen du sorkuntzarako eredu gisa, alde horretatik txantiloi-bidezko sistemen antza daukalarik. Baina, hutsuneak betetzerako orduan hitzak ez dira besterik gabe aukeratzen. Kontuan hartzen da bertsoko hitz kopurua, izen bakoitzeko adjektiboen ratioa, aditzen denbora, etab. Sorkuntza prozedurak bilaketa jale (*greedy*) gisa jokatzen du, poemaren hasierako hitzetik hasi eta, modu inkrementalean, baldintzak betetzen dituzten hitzak aukeratuz. Prozesu honekin, oinarrizko soluzioak garatzen dira lehenbizi, eta baldintza metrikoak gainbegiratzen dituen ebaluazio funtzio batekin parekatu segidan, bukaerako emaitza lortu aurretik. Emaitzari erreparatuz gero, metrika aldetik zuzenak diren poemak topatuko ditugu, esanahi aldetik ordea kaotiko samarrak dirudietenak (ikus adibidea 3.1 irudian).

Muérome por llamar Juanilla a Juana,
que son de tierno amor afectos vivos,
y la cruel, con ojos fugitivos,
hace papel de yegua galiciana.

3.1 Irudia: WASP sistemarekin sortutako poema

ASPERA (*Automatic Spanish Poetry Expert and Rewriting Application*) sistema (Gervás, 2001) berriz, **kasu-bidezko arrazoitze** (*case-based reasoning*, CBR) metodoan oinarritzen da espainieraz poesia sortzeko. Erabiltzaileak emandako sarrerako datuen arabera sortzen du poema: mezuaren prosa bidezko deskripzioa, amaierako poemak bete behar duen neurria (estrofa-mota), neurri horretako adibide-poema multzoa eta, azkenik, amaierako poemak eduki behar dituen hitz multzoa (3.2 irudian adibidea). ASPERA, CLIPS (Giarratano and Riley, 1998) erregela sisteman oinarritua dago eta ohiko CBR sistemen lau pausuak jarraitzen ditu (Aamodt and Plaza, 1994): Berreskuratu (*Retrieval*), Berrerabili (*Reuse*), Berrikusi (*Revise*) eta Gorde (*Retain*). Lehenik eta behin, corpusetik mezuaren esaldi bakoitzeko kasu (*case*) berezitu bat berreskuratzen da (*Case retrieval*). Honen ondoren, Aukeratu diren kasu horien *part-of-speech* (POS) etiketen egitura jarraituta, hitz osagarriak jarrita, esaldi berriak sortzen dira (*CBR Reuse Step*). Honek zirriborro bat sortzen du eta erabiltzaileari erakusten zaio, hark ebaluatu edo zuzendu dezan (*CBR Revise Step*). Amaitzeko, poema balioztatuak analizatzen ditu, informazio hori datu-basean txertatu eta hurrengo poema sorkuntzan erabili ahal izateko (*CBR Retain Step*).

COLIBRI (*Cases and Ontology Libraries Integration for Building Reasoning Infrastructures*) (Díaz-Agudo et al., 2002) ere CBR metodoan oinarritzen da, ASPERAren oso antzekoa izanik. Bien arteko alderik nabarmenena litzateke COLIBRIk CBROnto izeneko ontologia txertatua duela, bere inferentzia-ahalmena eta adierazpidea hobetzen dituena (ikus 3.3 irudia).

Ladrará la verdad el viento airado
en tal corazón por una planta dulce
al arbusto que volais mudo o helado

3.2 Irudia: ASPERA sistemarekin sortutako poema

no sólo en boca y viola ardiente
se pase mas tú y ello juntamente
en tierra en techo en suelo en sombra en nada

3.3 Irudia: COLIBRI sistemarekin sortutako poema

Levy-k (Levy, 2001), eboluzioan oinarritutako eredua garatu zuen poesia sorkuntzarako. **POEVOLVE** sistemak *limerick*[2]erako poemak sortzen ditu, gizakion poesia sortzeko modua eredu gisa hartuz. Sistema bi moduluz osatua dago: modulu-sortzailea eta modulu-ebaluatzailea. Lehenbizikoak objektu edo testu poetikoak sortzen ditu etengabe, bigarrenak jaso eta aztertzen dituenak. Bi moduluak prozesu paralelo gisa exekutatzen dira, modu honetan: abiapuntu gisa hitz multzo bat hartuta haserako populazioa sortzen da, azentuari eta fonologiari buruzko informazioa txertatuta daukatenak. Algoritmo genetikoak *limerick*-a osatzen duten hitzak aldatuko ditu pausuz-pausu. Ebaluazioa, giza-epaiekin trebatutako sare-neuronalak

[2]https://en.wikipedia.org/wiki/Limerick_(poetry)

burutzen du. Hurbilpen honek, dena den, sintaxi eta semantika ez ditu aintzat hartzen.

Manurung-ek (Manurung, 2003) aurkeztutako **McGonnagall** sistema, POEVOLVEren antzekoa da oso, biak ala biak algoritmo ebolutiboetan oinarritzen direlarik. Poesia sorkuntza prozesua, egoera espazioaren baitako bilaketa problema gisa formulatzen da, *hill-climbing* estokastikoa erabiliz. Prozesu osoa bi pausutan banatu daiteke: ebaluazioa eta eboluzioa. Ebaluazio fasean indibiduo multzo bat sortzen da (populazioa), sarrera gisa emandako datuak, semantika alorreko helburuak eta helburu fonetikoetan oinarrituta. Hasierako indibiduo multzoa ebaluatu egiten da ondoren fonetika, semantika eta azaleko egitura (*surface form*) bezalako ezaugarriak kontuan izanda. Indibiduo bakoitzak puntuazio bat jasotzen du eta, eboluzio fasean, puntuazio altuena daukan azpitaldea aukeratzen da birsortzeko. Pausuz pausu, sortuz eta mutatuz doazen indibiduoek poemaren bertsio hobeago bat osatuko dutela espero da. Sistemak helburu-egoera erdietsiko du eratutako poemak ondoko hiru ezaugarriak betetzen dituenean: adierazgarritasuna, poetikotasuna eta gramatikaltasuna. Egoera hori identifikatzeko ebaluazio-funtzioak definitzen ditu Manurung-ek.

McGonnagall-ek ere gizakien sortze-prozesua simulatzen du, testua sortzerakoan forma eta edukiaren arteko elkarrekintza estuki lotua dagoelarik. Baina inplementatzerako orduan, jakintza intentsiboko soluzioa da, fonetika, gramatika eta semantikako formalismo asko behar dituelako. Forma eta edukia, bakoitza bere aldetik tratatuz gero, soluzio optimoak topatzeko gai den arren McGonnagall sistema, egileak berak aitortzen du (Manurung et al., 2012) bi ebaluazio funtzioak aldi berean kontuan hartzeko arazoak dauzkala.

Wong-ek (Wong et al., 2008) blogetatik erauzitako testuekin haiku modernoak sortzen dituen sistema aurkeztu zuen, Bektore-espazio ereduan (VSM, *Vector Space Model*) (Turney and Pantel, 2010) oinarritua. Proposamenak, hitz gakoak erabiltzen ditu, haikuak idazterakoan erabilienak diren 50 hitzez osatua, eta lerro edo esaldi errepositorio bat blogetatik ateratako esaldiz osatua. Haiku-sortze prozesua lexiko zerrendatik 3 hitz aukeratuz hasten da, haikuaren irudi orokorra osatuko dutenak. Segidan, hitz gako horiek dituzten blogetako esaldiak bilatzen ditu sarean. Esaldi bakoitzetik 2 hitz gako ateratzen dira, *tf-idf* (Ramos, 2003) pisatze-sistema erabiliz hitz bakoitzak esaldiaren baitan daukan garrantzia edo pisua neurtzeko. Ondoren, esaldi pareak bektore-bidez konparatzen dira Bektore espazio eredua (VSM) erabiliz. Bektoreen arteko angeluaren kosinua erabiltzen da beraien arteko hurbiltasun semantikoa neurtzeko, baliorik altueneko esaldi parea aukeratzen delarik. Bukaerako haikua, bektore bikoteen artean baliorik altuena ematen duen konbinazioak osatuko du (adibidea, 3.4 irudian).

Like mirror images
Of the moon falls
At river house restaurant

3.4 Irudia: Wong-en haiku sortzailea

Full-FACE (Colton et al., 2012), corpusean oinarritutako poema-sortzailea da, eredu edo txantiloiak erabiltzen dituena baita. Hutsuneak bete eta poema berriak osatzeko errima, metrika, azentua, sentimendua eta hitzen maiztasuna eta antzekotasuna aintzat hartzen dituena. Horretaz gain, sistemak eguneko aldartea erabakitzen du egun jakin horretako egunkariak aztertuta. Eguneko aldarteak poemaren sorkuntza baldintzatzen du, txantiloi gisa erabiliko den poema-eredua aukeratzeko orduan eta baita, poemaren oinarri izango den egunkariko artikulua aukeratzeko orduan ere. Poema sortzeko algoritmoak 4 pausu dauzka: Erauzketa, Biderketa, Konbinaketa eta Instantziazioa deiturikoak: lehen pausuan, konparazioak (esaldi oso laburrak) Internetetik erauzten dira, euren sentimendua kontuan izanda. Biderketa pausoan, aukeratutako konparazio bakoitzeko bariazio edo aldaketak egiten ditu Full-FACE sistemak. Konbinazioan, konparazioak eta euren bariazioak egunkarietatik ateratako esaldi-gakoekin konbinatzen dira, txantiloiaren arabera egokiak direnak soilik aukeratuz. Amaitzeko, Instantziazioan, sortutako konbinazio berrietatik bat ausaz aukeratzen da. Poema sortzeaz batera, sistemak poemaren sortze-prozesuari buruzko komentarioa idazten du, hartutako erabakiak azaltzen dituena (aldartea, erabilitako esaldiak, estetika, eta abar). Full-FACE sistemaren ekarpenik nabarmenena, emozioetan oinarritutako poemak sortzeko zabaldu duen bidea da (ikus 3.5 irudia).

Poetryme, poesia sortzeko metodo gisa baino, plataforma bezala irudikatzen dute egileek (Oliveira, 2012) (Oliveira, 2015). Planteamendu horrekin, bertsio berritzen eta ezaugarriak gehitzen joan dira arkitektura orokorrean. Bertan, gramatika eta semantika alorreko txantiloiak aplikatzen dira eta sorkuntza estrategia ezberdinak inplementatu daitezke. Poetryme gai da bertso edo estrofako lerro kopurua, lerro bakoitzaren silaba kopurua eta errima-eskema emanda bukaerako poema osatzeko. Hizkuntza ezberdinetan eskuragarri jartzeko helburuarekin, gazteleraz erabiltzeko egokitu dute berriki (Oliveira et al., 2014), baina batik bat portugaleraz

Relentless attack
a glacier-relentless attack
the wild unprovoked attack of a snake
the wild relentless attack of a snake
a relentless attack, like a glacier
the high-level function of eye sockets
a relentless attack, like a machine
the low-level role of eye sockets
a relentless attack, like the tick of a machine
the high-level role of eye sockets
a relentless attack, like a bloodhound

3.5 Irudia: Full-Face sistemarekin sortutako poema

sortzen ditu bertsoak (ikus 3.6 irudia).

Adibide poema:
ah paixão afecto
não tem paixão nem objecto
sem na arte modos

3.6 Irudia: Poetryme sistemarekin sortutako poema

Markov kateetan oinarritutako sistemak ere ezagunak dira testu-sorkuntzan. Eredurik sinpleena eta erabiliena N-grama eredua da: N-gramek hitz sekuentzien probabilitateak esleitzen dituzte eta eraikitako ereduak hitz kate berriak sortzeko erabili daitezke, ikasitako probabilitate banaketaren arabera (Jurafsky and Martin, 2000). Beste modu batera azalduta, N-grama eredu batean, hitz jakin baten agerpen-probabilitatea, bere aurreko N-1 hitzek baldintzatzen dute.

Demagun, adibide gisa, "Eibarren egiten da eskopeta fiña"esaldia hartzen dugula. Esaldi horretako unigrama, bigrama eta trigramak ondokoak dira:

- **Unigramak** = {Eibarren, egiten, da, eskopeta, fiña}
- **Bigramak** = {Eibarren egiten, egiten da, da eskopeta, eskopeta fiña}

- **Trigramak** = {Eibarren egiten da, egiten da eskopeta, da eskopeta fiña}

Markov-en hipotesiak zera dio: sistema dinamiko baten hurrengo egoera azkenaldiko historiaren araberakoa izango dela. Zehazki esanda, k ordenako Markov ereduan, hurrengo egoera kalkulatzeko azken k egoerak bakarrik hartzen dira kontutan. Beraz, N-grama eredua (N-1) ordenako Markov eredua besterik ez da. Modu formalago batean adierazteko, demagun hitz sekuentzia bat daukagula:

$$w_1^n = w_1 ... w_n$$

Hitz kate hori gertatzeko probabilitatea, ondoko formularen bitartez kalkula dezakegu:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) = \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

Markov-en hipotesia aplikatu eta bigrama bidezko hurbilketa eginez gero (azken 2 hitzak kontuan hartuz soilik):

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-1})$$

Goiko hurbilpena lortuko genuke. Trigramak erabiliz gero berriz:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-N+1}^{k-1})$$

Orokorrean, testu bat emanda, haren N-grama probabilitate baldintzatuak kalkulatzeko, hitzen maiztasun erlatiboak besterik ez ditugu behar. Bigrama ereduaren probabilitate baldintzatua:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

N-grama ereduaren probabilitate baldintzatua:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

Funtsean, N-grama eredua esaldiak sortzeko automata probabilistiko gisa ikus eta erabili genezake. Esaldia osatzeko hurrengo hitza, aurreko N-1 hitzen probabilitate baldintzatua kontuan hartuz aukeratzen delarik.

N-grametan oinarritutako lan interesgarria Barbierik aurkeztutakoa da (Barbieri et al., 2012). Baldintzatutako N-grama eredua berria proposatzen da bertan, autore jakin baten estiloa jarraitzeaz gain, poemak sortzerakoan errima, metrika, gramatikaltasuna eta, neurri batean behintzat, semantika ere bilaketa-prozeduran txertatzeko gai dena. Sistema, sarrera gisa emandako hitz batekin hasten da, eta corpusean hitz horren hurbilekoenak diren M hitzak aukeratzen ditu. M hitz horiek barne hartzen dituen poema soilik hartuko du semantikoki zuzentzat sistemak. Finean, aberastutako edo baldintza berrietan lan egiteko moldatutako N-grama testu-sortzailea da, testu-sortze prozeduran bertan baldintzak teilakatzen dituena (ikus 3.7 irudia).

Innocence of a story I could leave today today imposed
When I go down in my hands and pray
She knocked upon it anyway
Paradise in the dark side of love it is a sin paradise
And I am getting weary looking in
Their promises of paradise
Now I want to know you would be spared this day
Wind is blowing in the light in your alleyway
Innocence in the wind it whispers to the day
Out the door but I could leave today
She knocked upon it anyway

3.7 Irudia: Barbieriren N-grama sortzailea. Dylan-en kantuetan oinarrituta sortutako poema

Corpus-erauzketan oinarritutako prozedura proposatzen digu Toivanen eta bere taldeak (Toivanen et al., 2012). Bertan, corpusetako testuak berrerabiltzen dira baina zenbait hitz ordezkatuz. Bi corpus ezberdinetan oinarritzen da, bata forma edo egitura-mailako baldintzak ezartzeko eta, bestea, edukia baldintzatzeko. Edukiari dagokionez, finlandierako Wikipedia baliatuz, hitzen arteko elkarketa edo asoziazio sareak eraikitzen ditu. Formari dagokionez berriz, finlandierako poesia klasikotik elikatzen da, bertatik testu-puskak hartuz. Sortze-prozesua honakoa da: poesia-corpusetik eredu gisa bat aukeratu eta bertako zenbait hitz sintaktikoki bateragarriak diren beste hitz batzuengatik ordezkatzen dira, asoziazio sarea baliatuz. Metodo hau txantiloietan oinarritutakoaren erkatu genezake, corpusetik eredua atera eta bertan hitzen ordezkapenak burutzen direlarik (ikus 3.8

irudian adibide-poema). Aurrerago (Toivanen et al., 2014), metodoaren hobekuntza proposatu zuten: hitzen asoziazio sarea osatzeko, sarrera dokumentuarekiko espezifikoak diren hitz-elkarketak eraikitzen dira metodo estatistikoak erabiliz eta ez, aurrekoan bezala, Wikipedian oinarritutako generikoak. Horrez gain, sare hori baliatuko da txantiloiko hitzen ordezkapenerako, segurtatuz sortuko den poema berria dokumentu bakarrean oinarritua dela. Laburbilduz, sarrera edo *input* gisa emandako dokumentuan oinarritutako poemak sortzen ditu sistema honek.

Lives got the frolic ways,
snow the home of time,
softly chimed abandoned homes,
softly got frolics beloved -
ripening crop got the snows' joys.
Waves fared the wind's ways,
sun the track of time,
slowly skied for long days,
slowly crept for long nights -
day wove the deeds of moons

3.8 Irudia: Toivanen-en sistema. Elurra gai gisa emanda sortutako poema

Agirrezabalen proposamena (Agirrezabal et al., 2013), Kategoria Gramatikalen (POS, *part-of-speech*) txantiloietan oinarritzen da. Bertso-corpus batetik POS-tag sekuentziak ikasten dira lehenbizi eta, ondoren, horiek erabiltzen dira poema berriko lerroak sortzeko txantiloi gisa. Esperimentu ezberdinak burutu zituzten txapelketetan kantatutako bertsoak analizatuz.

1. **Esperimentua**: puntuetako hitz guztiak kategoria morfologiko bereko hitzekin ausaz ordezkatzea.

   Eredua: Donostitikan Itziarrera etorritzen naiz erromes

   Kategoria Gramatikala: `IZE_ABL IZE_ABU ADI ADL IZE_ABS`

   Sistemak sortutakoa: Tributik eskera joan ditu kristo

2. **Esperimentua**: aditzak aldatu gabe

   Eredua: Donostitikan Itziarrera etorritzen naiz erromes

   Kategoria Gramatikala: `IZE_ABL IZE_ABU ADI ADL IZE_ABS`

Sistemak sortutakoa: Burutik hiriburura etorritzen naiz indar

3. **Esperimentua**: izen eta adjektiboak aldatu, semantikoki antzekoak diren hitzekin (sinonimoak, antonimoak, ...)

   Eredua: Lotsagabekeria zuregan nagusi

   Kategoria Gramatikala: `IZE_ABS IOR IZE_ABS`

   Sistemak sortutakoa: Lasaitasuna zuregan gazte

Turing-en testa (Turing, 1950) burutu eta egiaztatu zuten, emaitzak nabarmen egiten zuela hobera 3. esperimentuan. Azken honetan ebaluatzaileak ez zuen horren garbi bereizten poema-lerroa makina batek edo gizakiak egindakoa zen.

Bertsoa erabili arren eredu gisa, emaitzak ez ditu metrika eta errimaren baldintzak betetzen.

DopeLearning (Malmi et al., 2015), rap estiloan testuak sortzen dituen sistema da . Rap letrez osatutako corpusa erabiltzen du horretarako. Lehenbizi, Ikasketa Automatikoko tresnak erabiliz, iragarpen eredu bat eraikitzen dute: lerro bat eman eta ondoko edo segidako hautagai posibleen artean lerro egokia zein litzatekeen ikasteko. *RankSVM* eta *Deep Learning* teknikak erabiltzen dituzte, hurrenkera ikasteko. Bigarren pausuan, iturri ezberdinetako rap letrak nahastu eta iragarpen-sistema baliatzen dute, errima eta semantika kontuan hartuz, lerroen konbinazio berriak lortzeko.

Sistemaren online demoa eskuragarri dago[3]. Bertan, rap letrak automatikoki sortzen dituen sistema probatu daiteke. Beste aukera bat ere bada, post-edizio antzekoa, jada sortutako letrak moldatu (eta hobetzeko) tresnak eskaintzen zaizkio bertan erabiltzeari, bertsotarako Arbela Digitalean[4] egiten den bezalatsu (Agirrezabal et al., 2012). Sistemaren erantzun-denborari dagokionez, online demoa 30 segundutan testua sortzeko gai da, beraz, bat-bateko sorkuntza dela esan genezake. Dena den, 3 gai nagusitara mugatzen da rap letrak sortzeko bere gaitasuna (maitasuna, egia, urtebetetzea).

Ezin da ahaztu poesiaren helburuetakoa dela mezuak transmititzearekin batera emozioak eragitea, beti ere estetika edo edertasun printzipioak bazter utzi gabe. Testuen sorkuntza-prozesuan emozioak integratzen dituzten lanak aztertzen hasita, **Full-face** poesia sistema ezin aipatu gabe utzi (Colton et al., 2012), dagoeneko aztertu duguna. **Stereotrope** (Veale, 2013) litzateke beste adibide bat, gaia eman eta hari buruzko poema

[3]http://deepbeat.org/
[4]http://www.bertso-eskolak.eus/web/arbeldigitala

emozional eta zorrotzak sortzen ahalegintzen dena. Misztal-ek ere (Misztal and Indurkhya, 2014) poema-bidez sentipen edo emozioak adierazteko gai den sistema aurkezten digu. Azkenik, **MASTER** (Kirke and Miranda, 2013), konputagailu-bidez poemak sortzeko laguntza-tresna da. Sistema honetan, agente multzo bat, bakoitza bere umore egoera eta hasierako hitzekin, elkar baldintzatzen saiatzen da bukaerako poema lortzeko bidean.

Arloaren azterketarekin amaitzeko, hizkuntzari erreparatu nahi genioke. Sorkuntzaren emaitza zer hizkuntzatan dagoen begiratuz gero, ingelesa da jaun eta jabe. Baina badira beste hizkuntzatan egindako ahaleginak ere: espainiera (Gervás, 2000) (Gervás, 2001), portugalera (Oliveira, 2015), finlandiera (Toivanen et al., 2013) eta (Toivanen et al., 2014), txinera (Yan et al., 2013), indonesiera (Rashel and Manurung, 2014), bengalai hizkuntza (Das and Gambäck, 2014), tamilera (Ramakrishnan et al., 2009) eta baita euskara ere (Agirrezabal et al., 2012) (Agirrezabal et al., 2013).

3.3 Ondorioak

Gaiari buruzko literaturak erakusten duen bezala, konputagailu-bidezko poesia sorkuntzak ikerketa komunitatearen atentzioa lortu du azken urteetan eta sistema aski interesgarriak garatu dira. Kapitulu honen sarreran adierazi bezala, idatzizko poesia eta bertsolaritzak badituzte zenbait ezaugarri komun, baina bat-bateko bertsoa ahozko generoari dagokio (Garzia, 2000) eta bere berezitasunak dauzka:

- Idatzizko poesian ez bezala, bat-bateko bertsoa momentuan sortu eta kontsumitzen da, leku eta une berean (Egaña, 2007) (Lujanbio, 2015).
- Publikoaren aurrean, denbora mugatuan, burutu beharreko *performance* edo gertakaria da. Publikoak eta testuinguruak eragin nabarmena dute sorkuntza-prozesuan eta emaitzan (Garzia, 2007). Sarasuaren hitzetan (Sarasua, 2004) :“garrantzitsua ez ezik sorkuntzan bertan parte hartzen duen elementu funtsezkoa da. Izan ere, ikusgarria bat-batekotasunean oinarritzen denean, une-gune horretako errepikaezinean, sortzaile eta entzulearen artean gertatzen denak garrantzi berezia hartzen du, muinean berean eragiten du.”
- Bertsolaritzan helburu komunikatiboa da helburu nagusia, eta ez poetikotasuna (Lujanbio, 2015).

Bat-bateko bertsogintzatik hurbilen egon daitekeen ahozko jarduna, *freestyle* hip-hopa izango da seguru asko (Malmi et al., 2015) (Wu and

Addanki, 2015). Aipatu lanetan, **DopeLearning** eta **Freestyle** sistemak aurkezten zaizkigu, *freestyle* eran hip-hoperako testua garatzen edo sortzen duten lanak hain zuzen ere. *Challenge-response* edo puntu-erantzunerako gaitasuna erakusten dute, errimatuz aurkariari erantzuteko gaitasuna alegia. Sortu eta plazaratzeko moldean bertsoarekiko antzekotasun handiko inprobisazio-moldea izan arren, aipatu lanak ez dira ahozko jardun inprobisatu gisa aurkezten.

Orokorrean, aztertu ditugun sistemak konputagailu-programak dira. Beraien helburua idatzizko testua da, ez kantatzen ez deklamatzen ez dena. Zenbait sistematan erabiltzaileak sortze-prozesua bideratzen edo gidatzen duen arren, publikoarekin elkarrekintzarik ez daukate. Adimen Artifizialaren ikuspegi dual klasikoaren ikuspegia jarraituz garatuak izan dira. Konputagailua burmuinaren eredutzat hartuz eta sorkuntza-prozesua burmuin horretan kokatuz, inguratzen duen errealitatearekin inongo konexio eta elkarrekintzarik gabe.

Gure proposamena bestelakoa da, adimen gorpuztuan oinarritua (Brooks, 1991): bertsolariak gorputza dauka eta bere bitartez, sorkuntza-lanean eragiten duten inguruneko estimuluak jasotzen ditu (kantulagunaren doinua, publikoaren erreakzioa, txaloak, aurpegiera...). Eta, bestetik, ingurune horretan ekintzak burutzen ditu (keinuak, mugimenduak, kantua...) era berean ingurune horretan eragiten dutenak. Bertsoa, sorkuntza-lana, ez da beraz burmuinean isolatuta garatzen den prozesua, gorputzaren bitartez inguruarekin elkarrekintzan garatzen dena baizik. (Jordanous, 2015) eta (McGregor et al., 2016) lanen ildo beretik, sorkuntza-lana ez da existitzen agentearen (robot, konputagailu, web-aplikazio...) gorputzean bakarrik, agentea, publikoa eta ingurunearen arteko elkarrekintza dinamikan baizik. Hori da bertso-sorkuntza automatikorako gure proposamena eta horrek konputagailu-bidezko estrategiak robotean txertatzea baino zerbait gehiago eskatzen du. Robotikari buruzko atalean landuko dugu zehatzago gai hau.

Bestalde, bertsoaren betebehar teknikoen zailtasun-maila ere kontuan hartzekoa da. Errimatuz eta silabak neurtuz diskurtsoa eraikitzea ez da ataza erraza, gutxiago bat-batean egin behar bada. Baldintza estu horietan ari delarik, Egañak (Garzia et al., 2001) dionez, bertsolariak ez du esan nahi duen hura esaten, errima eta metrikak uzten diotena baizik. Baina, murrizpen edo betebehar horiek berak sorkuntza-lanean lagungarri ere gerta daitezke. Izan ere, hizketa librea larre zabalegia da konputazionalki tratatu ahal izateko. Testu-sorkuntza Boden-ek (Boden, 1990) planteatu zuen gisan, bilaketa-prozesu bezala, ezin genezake konputagailu-bidez eraiki. Esan daitekeenaren barruan mugak behar ditugu, eremua zehaztu. Bertsoaren soineko estuak, metrika eta errima baldintzek, horretan lagunduko digute. Askozaz ere modu ederragoan adierazi zuen Igor Stravinsky-k: “Zenbat eta murriztapen gehiago ezarri, orduan eta libreago gara izpitirua lotzen duten kateetatik”[5]. Maialen Lujanbiok beste hitz batzuekin baina ideia berdintsua adierazten du: “bertsoaren zehaztasunak libreago egiten gaitu bertsolariak. Bestela joango ez gintezkeen tokietara eramanez. Errimak bilatzean bururatzen zaizkigun hitzek ekartzen dizkigute sarri bertsoetarako ideiak” (Lujanbio, 2015).

Beraz, badirudi betekizun formalek esan nahi dugun hura esateko zailtasunak jartzen dituztela alde batetik; baina bestetik, infinituaren amildegi ertzetik salbatzen dutela sortzailea, sormenerako heldulekuak emanez.

Gaiaren beste ertz bat, zorrotz askoa gainera, sortu edo osatutako testuen ebaluazioari buruzkoa da. Nola ebaluatu automatikoki sortutako testua? Emaitza ebaluatzera mugatu behar ginateke edo, prozesua bera ere kontuan hartzea komeni da? Adibidez, Colton-ek (Colton et al., 2012) emandako Sormen Konputazionalaren definizioak kanpo-begiralea inplikatzen du, eta ondorioz, giza-ebaluatzailearen beharra dakar berarekin. Edozein modutara, konputagailu-bidez sortutako bertsoak ebaluatzearena ez da nolanahiko ataza. Automatikoki burutzeko metodoak izan badira, baina nahiko nahasiak iruditu zaizkigu eta, beraz, adituen laguntza eskatzea beste biderik ez dugu topatu gai honi dagokionez.

[5]https://quotecatalog.com/quote/igor-stravinsky-the-more-constr-V7LR2Ja

4. Kapitulua

Ikasketa Automatikoa

Adimen Artifizialaren adar bat da Ikasketa Automatikoa, esperientziatik ikasteko gai diren konputagailu programak garatzea helburu gisa daukana (Mitchell, 1997). Baina zer da ikastea? Guk erabiliko dugun testuinguruan, definizio hauxe hartuko dugu aintzat (Shalev-Shwartz and Ben-David, 2014): ikastea, esperientzia trebetasun edo jakintza bihurtzen duen prozesua da. Ikasketa algoritmoaren sarrera edo inputa entrenamendu datuak dira, esperientzia adierazten dutenak, eta irteera edo outputa jakintza edo trebetasuna litzateke, ataza jakin bat burutzeko gai den konputagailu-programa gisa adierazia.

Definizio formalagoa eskaintzen digu, Ikasketa Automatikoari buruz hitz egitean ezinbesteko erreferentzia den Mitchell-ek (Mitchell, 1997): `T` ataza-multzoa burutzeko `R` errendimendua daukan konputagailu programa batek, `S` esperientziatik ikasi duela esango dugu baldin eta `S` esperientziak `T` ataza burutzeko `R` errendimendua hobetzen badu.

Definizio hori jarraituz, hainbat metodo garatu izan dira eta, horiei esker, konputagailua gai da kalkulu-ahalmenaz harago doazen trebeziak eskatzen dituzten problemak ebazteko. Arrakastaz aplikatu dira Ikasketa Automatikoko algoritmoak eguraldiaren iragarpena, gaixotasunen diagnosia, eta antzeko ataza askotan (Libbrecht and Noble, 2015), (Witten et al., 2016), (Xingjian et al., 2015). Azken batean, metodo horien bidez ezagutza lortu nahi da, egoera berri baten aurrean (paziente berri bat datorkion medikuari, gaurko eguraldiaren deskribapena jaso duen meteorologoari) erabaki egokia hartzen laguntzeko.

Azken aldian konputagailuek izan duten konputazio-ahalmenaren gorakadak eta hainbat aplikazio-eremutan bildu diren datu-multzo erraldoiek izugarrizko bultzada eman diote Ikasketa Automatikoari (Marz and Warren, 2015), (White, 2012), (Manyika et al., 2011), (LeCun et al., 2015). *Big Data* bezala ezagun egin dena esploratu eta esplotatzeko ezinbesteko tresna bihurtzeraino (Witten et al., 2016).

Lengoaia Naturalaren Prozesamenduan ere, Ikasketa Automatikoko metodoak ataza ugaritan erabili izan dira azken urteotan. Besteak beste Dokumentuen Sailkatze Automatikoan, Informazio-Berreskuratzean, Esaldien Ordenazioan, Itzulpen Automatikoan, Hitzen Adiera-Desanbiguatzean, Korreferentzia-Ebaztean eta Galdera-Erantzun sistemetan (Jurafsky and Martin, 2000), (Manning and Schütze, 1999).

4.1 Algoritmo sailkatzaileak

Sailkatzea helburu duten IAko metodoak bitan banatu genitzake:

- **Gainbegiratutako sailkapena (*Supervised Learning*)**: aldez aurretik sailkatuak izan diren kasuetatik abiatuz eredu bat eraiki eta, eredu hori erabiltzen da kasu berriei buruzko iragarpenak egiteko. Funtsezkoa da aztertu nahi den arloa ondo adieraziko duten instantziez osatutako datu-multzoa izatea.
- **Gainbegiratu gabeko sailkapena (*clustering*)**: ikasketa algoritmoari ez zaio etiketatutako daturik ematen. Kasu honetan, sarrerako datu-multzoa zenbait azpimultzotan banatzen da, eta kategoria batzuk lortzen dira.

Aipatutako metodoetatik, gainbegiratutako sailkatzaileak erabiliko ditugu batez ere gure ikerketa-lanean. Etiketatutako kasu-multzo batetik abiatu eta esperientzia horretatik orokortzeko gaitasuna iristeko helburuarekin. Dena den, zenbait kasutan sailkatzaile bakarra erabili ordez sailkatzaile anitz erabiltzen dira erabakia hartzeko: multi-sailkatzaileak. Sailkatzaile ezberdinak entrenatu eta haien artean onena aukeratu ordez haien iragarpenak konbinatzen direlarik (Ho et al., 1994). Kasu askotan, oinarrizko sailkatzaileekin lortutako emaitzak hobetzen dira multi-sailkatzaileekin.

4.2 Ondorioak

Gizakiok gure ingurunearekin dugun hartu-emanetatik ikasten dugun bezala konputagailuek ere esperientziatik ikastea nahi badugu, esperientzia hori jasotzen duten datuak eman behar dizkiogu. Hori da, hain zuzen ere, kapitulu honetan bildu ditugun metodoen ideia nagusia: problema jakin baten inguruko datu-bilduma batetik abiatuz eta Ikasketa Automatikoko metodoak erabiliz, problemarako ereduak sortzea, eredu horiek erabiliz giza aditu batek egingo luken antzera egoera berrien aurrean erabakiak hartzeko. Horretarako funtsezkoa da aztertu nahi den arloa ondo adieraziko duten instantziez osatutako datu-multzoa izatea, hau da, gertaera espazioa egoki ordezkatzea.

Funtsezkoa da hortaz eredu gisa erabiliko ditugun testu egokiak izatea; testu-corpusak alegia. Testu-corpusak, azterketa linguistikoak egiteko balio duten testu-bildumak dira, hizkuntzalaritza-ikerketetarako ezinbestekoak. Baliagarriak dira baita ere testu-sorkuntzan edo itzulpengintzan. Gure kasuan, bertsoen corpus egoki batek bertatik ikasteko aukerak zabalduko dizkigu.

Amaitzeko, Mitchell-en definizioak Ikasketa Automatikoaren definizio operazionala ematen digu, eta ez kognitiboa. Hau da, Alan Turing-ek proposatutako galdera (Turing, 1950), "Pentsatzeko gai al dira makinak?" ondoko galderagatik ordezkatu beharko genuke, "Gai ote dira makinak guk (izaki pentsalari gisa) egiten duguna egiteko? (Harnad, 2006) Pil-pilean dagoen eztabaidari ez diogu erabateko erantzunik emango. Esango dugu, ikerkuntza-lan honetan, bigarren ikuspuntua egin dugula geure. Hau da, emaitza erreplikatzen ahalegindu eta ez prozesua bera. Ikasketa Automatikoa bertsoaren sorkuntza prozesura eramateko ahalegina egingo dugu baina ez sorkuntza-prozesua bera hobeto ulertzeko asmoz, ezpada bertsoaren zenbait komunikazio-gaitasun simulatu eta erreplikatzeko.

Ondo finkatutako ikerkuntza-arloa da IA, ikasketa-algoritmo ezagunenak ondo aztertuak ditu komunitate zientifikoak. Beraz, zein da alor honek Bertsobotari egin diezaiokeen ekarpena?

Bertsoa komunikazio-tresna gisa aztertuz gero, badirudi bertsoak, bertso bakoitzak, diskurtso beregaina osatzen duela. Baina elkarrekin partekatzen duten ezaugarririk ba ote dute? Ba al dago ezaugarri horietan oinarrituta bertso mota horren egitura narratibo jakina edo askotarikoari antzematerik?

Diskurtsoaren ezaugarriak identifikatuta, horien antolamenduari buruzko galderak sortzen zaizkigu hurrena: bertsolarien sortze-estrategia antzemateko gai ote gara IA bidez? Bertsoaren osaeran, ba ote puntuen arteko ordenazio jakinik? Ideia nagusia, azken puntua, identifikatzerik ba ote beste guztien artetik? Baiezkorik topatu ezkero, aurrerapauso nabarmena suposatuko luke mezuak bertso-bidez komunikatu behar dituen Bertsobotarentzat.

Galdera horiek eta antzekoak erantzuten ahaleginduko gara IAko teknikak baliatuz.

5. Kapitulua

Semantikarako Bektore-espazio eredua

Bektore-espazio eredua (VSM), Salton eta bere taldeak garatu zuen SMART informazio berreskuratze proiekturako (Salton et al., 1975). Bertan, gaur egungo bilatzaileek erabiltzen dituzten kontzeptu ugari finkatu zituzten Salton eta bere taldeak. VSMk testuen prozesamendu semantikoa burutzen du, ondoko ideian oinarrituta: dokumentu-multzo bat izanda, dokumentu bakoitza espazioko puntu gisa adierazi (bektore gisa bektore-espazioan). Espazio horretan elkarrengandik hurbilago dauden ountuak semantikoki ere hurbilago egongo dira eta, alderantzikoa ere bai, urrunago daudenak semantikoki ere aparteago izango dira elkarrengandik. Erabiltzailearen galdera ere puntu gisa adierazten da dokumentuen espazio berean eta, modu horretan, dokumentuak ordenatu daitezke galderarekiko hurbiltasunaren arabera. haiek egindako lanak gaur egungo Informazio-Berreskuratze (IR, *Information Retrieval*) sistemen oinarria finkatu zuen.

Bektoreen erabilera ohikoa da Adimen Artifizialean. VSM ereduaren benetako berrikuntza testu corpusetako hitzen maiztasunak erabiltzea izan zen, haiei buruzko informazio semantikoa lortzeko laguntza gisa. Semantika Estatistikoa bezala ezagutzen dena, testuetako hitzen maiztasunen azterketan oinarritzen da hitzen esanahia eta haien arteko erlazioak topatzeko. Testu multzo handiak aztertuz, hitzen agerpenen maiztasunak kalkulatu eta horiek testuaren esanahia zenbateraino jasotzen duten neurtu ahal izango da. Aipatzekoa da, Furnas eta bere taldeak (Furnas et al., 1983) Semantika Estatistikoaren alorrari egindako ekarpena.

Firth-ek esana omen da (Firth, 1957): "daukan lagunarteagatik ezagutuko duzu hitza". Hipotesi distributiboak jasotzen du ideia, antzeko testuingurutan gertatzen diren hitzek antzeko esanahia izan ohi dutela, alegia (Wittgenstein, 1953), (Deerwester et al., 1990). VSMri zor diogu hipotesi hori bektore-bidezko algoritmoekin inplementatu izana.

VSMk izandako arrakastak, metodoa semantikarekin zerikusia duten NLPko beste ataza batzuetan erabiltzea ekarri zuen (Turney and Pantel, 2010), (Erk, 2012). Ordutik gaur arte, makina bat algoritmo proposatu izan da, agerpenen maiztasunak aztertuz semantikaren azterketan sakontzeko, beti ere testu corpus handietan oinarrituz. *Latent Semantic Analysis* (LSA) bera ikerketa lan horietatik sortutako aplikazio arrakastatsu bat da (Deerwester et al., 1990).

Atal honetan, VSM eredua testuen semantika jasotzeko erabili nahi denean eman behar diren urratsak aztertuko ditugu.

Corpusak

Hemendik aurrerakoan, corpus terminoa testu-dokumentu (digital) multzo gisa ikusiko dugu. Dokumentua bera, testu-puska besterik ez da eta ez du zertan jatorrizko dokumentu osoa jaso behar. Hau da, corpusa osatzen duten dokumentuak pasarte, paragrafo, esaldi edo hitzez osatua egon daitezke. Dokumentu deituko diogu unitate gisa kontsideratzen dugunari, corpusari eman nahi diogun erabileraren baitakoa izango dena.

5.1 Aurreprozesamendu linguistikoa

Testuaren azterketa semantikoari ekin aurretik, corpuseko testuen prozesamendu linguistikoa burutzea gomendagarria izan liteke. Teknika guztien azterketarik ez dugu hemen egingo, baina aipatu nahi genituzke behintzat ikerketa-lan honetan erabili ditugunak:

- Tokenizazioa. Testu bat tokenetan banatzeari deitzen zaio tokenizatzea. Ohikoena, token gisa hitza hartzea da, hau da, zuriune edo puntuazio marken arteko karaktere-segida. Baina kontuan hartu behar da testuetan zenbakiak eta karaktere bereziak agertu ohi dira baita, hitza non hasi eta non bukatzen den zehaztea zailduz. Atazaren arabera tokenaren definizioa aldatu badaiteke ere, gure testuinguruan, karaktere segida gisa definituko dugu, zuriune, zenbaki eta puntuazio-markak ezabatuz.

- “Stop words” edo lasto-hitz zerrendak. Maiztasun altueneko hitzak eduki semantiko eskasekoak izan ohi dira eta, normalean, ezabatu egiten dira aurre-prozesamendu fasean.

- Lematizazioa edo erro-bilaketa. Corpusean erro bereko hitz desberdin asko ager daitezke, batez ere euskara bezalako hizkuntza eranskarietan. Hitz horiek guztiak funtsean esanahi berekoak izan ohi dira (*mendia*, *menditik*, *mendiraino*, *mendiaren*...). Hitzen semantika bektore-bidez adieraztea nahi dugunez, aldaera guzti horiek forma bakarrarekin jasotzea ideia ona izan daiteke. Lematizazioa edo erro-bilaketa hitzen erroa erauztea da (*mendi*), erro horrentzat adierazpen bektorial bakarra sortuko delarik. IXA taldeak garatutako *ixa-pipes* tresna erabili dugu euskarazko corpusak lematizatzeko (Agerri et al., 2014).

Hitzen zakua (bag-of-words)

Aurre-prozesamendu linguistikoa amaitu denean, dokumentuak bektore gisa adierazteari ekingo diogu. Dokumentu oro bektore gisa ordezka daiteke. Adibidez: "ez da ez makala gero"esaldia `{ez, da, ez, makala, gero}` bezala adierazi genezake. Elementuak ordenatu eta ondoko bektore-gisa erakutsiko bagenu, `<1,2,1,1>` adieraziz lehen elementua *da* terminoaren maiztasunari dagokiola, bigarrena *ez* terminoari, hirugarrena *gero* eta, azkenik, laugarrena *makala* terminoaren maiztasuna dela. Dokumentuak adierazteko era honi hitzen zakua edo bag-of-words esaten zaio. Normalean, bektorearen osagai bakoitza terminoen hiztegian dagokeen termino bakoitzari egokituko zaio, eta orobat, hitz horrek dokumentuan zehar duen agerpen kopurua adieraziko duen zenbaki bat jarriko zaio (zero balioa hartuko du dokumentuan inoiz agertzen ez bada).

5.2 Termino-dokumentu matrizea

Dokumentu multzo handia izanik, horiek jasotzen dituzten bektore edo *bag-of-words* mordoa izango dugu baita. Beraz, komenigarria izan daiteke bektore horiek matrize batean antolatzea. Matrizeko lerroak terminoak izango dira (hitz edo tokenak gure kasuan) eta zutabeak berriz, dokumentuak (corpuseko paragrafoak edo esaldiak, adibidez). Era honetako matrizeari, termino-dokumentu matrize (*term-document matrix*) deitu ohi zaio. Matrize hauetan, dokumentua *bag-of-words* gisa adierazten da.

Bektore-espazio ereduak termino-dokumentu matrizea erabiltzen du. Terminoak dokumentuetan ageri diren hitzak izango dira (hitz guztiak erabili beharrean azpimultzo bat erabili daiteke kasuaren arabera). Behin terminoak izanda, dokumentuetan duten agerpen maiztasuna aztertzen da. Hau da, n dokumentuz osatutako corpusetik m termino atera badira, $\mathbf{M} \in \mathbb{R}^{m \times n}$ matrizea eraikiko da. Matrizeak m errenkada eta n zutabe ditu, errenkada bat termino bakoitzeko eta zutabe bat dokumentu bakoitzeko. Matrizeko i. errenkadak t_i terminoaren adierazpen bektoriala emango digu $\mathbb{R}^n$ espazioan, eta j. zutabeak d_j dokumentuaren bektore-adierazpena $\mathbb{R}^m$ espazioan. Beraz, terminoak eta dokumentuak dimentsio desberdineko bektore-espazioetan adieraziak datoz. Matrizeko m_{ij} elementu bakoitzak i. terminoa j. dokumentuan zenbat aldiz agertzen den adierazten du, maiztasuna alegia.

5.2.1 Matrizeko elementuen pisatzea

Dokumentuen arteko antzekotasuna kalkulatzeko, maiztasunen termino-dokumentu matrizea bere horretan erabil dezakegun arren, ohikoa izaten da matrizearen elementuak pisatzea. Helburua, terminoen garrantzia maila neurtu eta dokumentuen semantika hobeto adieraziko duen beste matrize bat lortzea da.

Termino-dokumentu matrizeko m_{ij} elementuak informazio lokala ematen digu, t_i terminoak d_j dokumentua duen agerpen maiztasuna, corpus mailako bere pisua edo agerpena kontuan izan gabe. Hori osatzeko, pisatze eskemak erabiltzen dira bektore-espazio ereduan. Pisatze edo ponderazio eredu hauek ondoko balioetan oinarritzen dira:

- $\mathrm{tf}(t_i, d_j)$: "term frequency" edo t_i terminoaren maiztasuna d_j dokumentuan.
$$\mathrm{tf}(t_i, d_j) = m_{ij}.$$
- $\mathrm{gf}(t_i)$: "global frequency" edo maiztasun orokorra. t_i terminoaren maiztasun globala neurtzen du, hau da, corpus osoan guztira zenbat aldiz agertzen den:
$$\mathrm{gf}(t_i) = \sum_{j=1}^{n} m_{ij}.$$
- *df*(t_i): "document frequency" edo dokumentu maiztasuna. t_i terminoa corpuseko zenbat dokumentutan azaltzen den adierazten du:
$$\mathrm{df}(t_i) = \sum_{j=1}^{n} \min\{m_{ij}, 1\}.$$

Hiru faktore horietan oinarrituz eraikitzen dira ponderazio estrategia ezberdinak. Guztien artean erabiliena *tf-idf* estrategia da, segidan azalduko duguna. Bestelako ponderazio metodoen berri jakin nahi duenak jo beza (Turney and Pantel, 2010) edo (Salton and Buckley, 1988) lanetara.

tf-idf pisatze ereduak (Ramos, 2003), dokumentuko termino bakoitzaren pisua kalkulatzeko terminoaren maiztasuna (*tf*) eta dokumentuaren alderantzizko maiztasuna (*idf*, inverse document frequency) konbinatzen ditu . Hau da, ponderazio lokala (*tf*) eta globala (*idf*). *tf-idf* eskemak, dokumentu bakoitzeko t terminoari balio bat esleituko dio ondoko formulan oinarrituta:

$$tf - idf_{t,s} = tf_{t,s} \cdot idf_t$$

Non, *idf* ponderazio globalak terminoa dokumentu multzoan nola banatua dagoen adierazten duen. Zenbat eta gutxiago agertu dokumentuetan, orduan eta balio altuagoa itzuliko duelarik.

tf-idf eredua ulertzeko, kontuan izan behar da:

- Balio altua itzuliko duela t terminoa dokumentu-multzo txikian agertzen denean
- Balio baxuagoa itzuliko duela terminoa dokumentu berean gutxiagotan agertzen denean edo, dokumentu ezberdin askotan agertzen denean.
- Balio baxuena itzuliko duela terminoa dokumentu guztietan agertzen denean.

5.3 Antzekotasun semantikoa neurtzea

Ikusi dugu dokumentuak bektore bidez nola adierazi ditzakegun eta baita, elementu edo termino bakoitzaren pisua nola kalkulatu ere. Beraz, zenbaki errealez osatutako bi bektore adierazpen izanda, beraien arteko antzekotasuna neurtzeko modua behar dugu orain. Hemen ere funtzio ezberdinak topa genitzake (Turney and Pantel, 2010), guk ikerketa-lan honetan baliatu dugun kosinu-antzekotasuna azalduko dugu.

5.3.1 Kosinu-antzekotasuna

Metodo honekin, bektore espazioko bi bektoreren arteko angeluaren kosinua neurtuko dugu.

$$\cos(\theta) = \frac{V_1 \cdot V_2}{\|V_1\| \cdot \|V_2\|}$$

Non, zenbakitzaileak bektoreen arteko biderketa eskalarra adierazten duen

$$V_1 \cdot V_2 = \sum_{i=1}^{M} V1_i \cdot V2_i$$

Eta izendatzailea, distantzia euklidearren arteko biderketa izango den,

$$\|V_1\| \cdot \|V_2\| = \sqrt{\sum_{i=1}^{M} V1_i^2} \cdot \sqrt{\sum_{i=1}^{M} V2_i^2}$$

Kosinu-antzekotasuna honela interpretatu daiteke: Bi bektorek termino asko konpartitzen badituzte, antzeko noranzkoan adieraziak izango dira bektore-espazioan; beren arteko angelua txikia izanik, kosinu-antzekotasuna altua izango da. Bektoreen arteko angeluaren arabera neurtuko da haien arteko antzekotasuna. Horrela, noranzko bereko bi bektoreren arteko angelua zero izanik, haien kosinu-antzekotasuna 1 izango da. Elkarzutak diren bi bektoreren arteko angelua 90°-koa da, eta haien kosinu-antzekotasuna 0 da. 180°-ko angelua duten bi bektoreren kosinu-antzekotasuna −1ekoa da. Kosinu-antzekotasuna beti egongo da −1 eta 1 balioen artean.

Beraz, dokumentuen bektore-adierazpena terminoek baldintzatzen dute eta baita beraien arteko antzekotasuna ere. Zenbat eta termino gehiago izan komun, orduan eta txikiagoa izango da bektoreen arteko angelua, eta ondorioz, handiagoa kosinu-antzekotasuna.

Horixe da hain zuzen ere, VSM ereduaren oinarrizko planteamendua. Bektore-adierazpenak eta beren arteko antzekotasunak corpusetik erauzitako terminoek eta haien agerpen maiztasunek baldintzatuko dituzte. Tamalez, eredu honek ez du terminoen arteko ordena kontuan hartzen eta bektoreen arteko terminoen kointzidentzia hutsean oinarritzen da. Beraz, Hizkuntzalaritza Konputazionalean klasikoa den problema bati aurre egiteko gaitasunik ez du: sinonimia eta polisemia.

- Sinonimiaren arazoa aurkituko dugu bi hitz desberdinek esanahi bera dutenean, adibidez, eseri eta jarri. Bektore-espazio adierazpena ez da gai hitz sinonimoen arteko erlazioa harrapatzeko eta adierazpen desberdin bat egokituko zaio hitz bakoitzari.

- Polisemia arazoa aurkituko dugu hitz batek esanahi bat baino gehiago badu, adibidez, baso (edalontzia adieraz dezake, edo oihana). Hitzak adierazpen bakarra izango du bektore-espazio ereduan, nahiz eta dokumentu guztietan ez duen esanahi bera izango.

5.4 Latent Semantic Analysis (LSA)

VSM ereduaren ageriko muga edo arazo horiei aurre egiteko tresna gisa proposatu zen Latent Semantic Analysis (LSA) (Landauer et al., 2013). Testu idatzien semantika adierazteko gaitasuna duen tresna da LSA, terminoen arteko erlazio semantikoen neurria itzultzen duena. LSA metodoa termino, esaldi eta dokumentuen arteko erlazio ezkutukoak aurkitzen ahalegintzen da, beren agerpen maiztasunak ez ezik, agerpen horien testuingurua ere aintzat hartuz. Adibide batekin azalduta, LSA gai da A eta B terminoen arteko erlazioa aurkitzeko nahiz eta elkarren ondoan bi termino horiek agertu ez dokumentuetan eta beren arteko erlazioa C terminoaren bitartez gertatu. Hau da, A eta C dokumenturen batean elkarrekin agertzen dira eta B eta C ere bai, hortik LSAk A eta Bren arteko erlazioa inferitzen du.

Tresna bi izenez ezagutzen da ematen zaion erabileraren arabera (Zelaia, 2016): LSI esaten zaio Informazio-Berreskuratze atazarako erabiltzen denean eta erabiltzaileak egindako kontsulta bati erantzun behar zaionean. LSA (Latent Semantic Analysis) esaten zaio testuen eta hitzen semantikaren analisirako erabiltzen denean, termino-termino, termino-dokumentu edo dokumentu-dokumentu antzekotasunak neurtzea helburu denean, alegia.

LSAren mamia matrizeen aljebran dago. SVD izeneko matrize-banaketa (*Singular Value Deconposition* edo Balio Singularretan Deskonposatzea) erabiltzen du, eta hori baliatuz LSA-ak terminoen kointzidentziaz gaindi, esaldien arteko lotura semantikoaren neurria itzultzen du. SVD matrize-deskonposaketa, termino-dokumentu matrizeari aplikatzen zaio bere heina murrizteko, eta murrizte prozesu horretan, antzekoak diren bektoreak fusionatzen dira, edo bektore-espazioan hurbilago kokatzen dira. (Zelaia, 2016) eta (Zelaia et al., 2011) lanetan, SVD deskonposaketan oinarritutako dimentsio-murrizketa azaltzen da zehatz mehatz.

LSA erabili ahal izateko corpus bat behar da. Aurre-prozesamendu linguistikoa egin ostean, corpuseko dokumentuetan eta terminoetan oinarrituz termino-dokumentu matrizea osatuko du LSA metodoak. Matrizeko elementuak *tf-idf* ponderazio eskema erabiliz pisatuko ditugu eta matrizeko balioak birkalkulatu. Segidan, SVD deskonposaketa kalkulatuz p dimentsioko espazio semantikoa sortuko du. Espazio hori da antzekotasun semantikoa neurtzeko erabiliko duguna. Erabiltzaileak bi dokumentu, bi esaldi, konparatu nahi baditu antzekoak diren edo ez jakiteko, bi dokumentu horien bektore-adierazpena sortu behar da lehenbizi. Ondoren, kontsultaren koordenatuak kalkulatuko dira espazio semantikoan.

Erabiltzaileak erabaki beharrekoa da zein p dimentsiora murriztu nahi duen jatorrizko bektore-espazioa. p-ren aukeraketa oso garrantzitsua da, LSAk emandako emaitzak p dimentsio horren arabera aldatuko baitira. Hala ere, ez da ezagutzen dimentsio egokia aukeratu ahal izateko metodorik. Hori dela eta, enpirikoki aukeratu behar izaten da p, balio desberdinetarako probak eginez. Praktikan, 100-250 arteko balioak erabili ohi dira.

Hitzen eta esaldien arteko antzekotasuna antzemateko, ezinbestekoa espazio semantiko egokia eraikitzea. Horretarako tamaina handiko corpusak beharrezkoak dira, terminoen arteko erlazioak ikasi ahal izateko. Baina, bestalde, corpusa handiegia baldin bada erlazio gehiegi antzemango ditu antzekotasunak ez dira nabarmenduko. Esperientziak erakutsi digu zenbat eta corpus zehatzagoak erabili, orduan eta emaitza hobeak ematen dituela LSA metodoak. Hau da, itsasoari buruzko dokumentuen arteko antzekotasuna bilatzeko, onen onena itsasoari buruzko testu-corpusa izatea dela.

Informazio-Berreskuratze (*Information Retrieval*) eremuan oso erabilia da metodo hau. Guk, hala ere, dokumentuen arteko antzekotasuna neurtzeko metodo gisa erabili dugu lan honetan, LSA gisa alegia.

6. Kapitulua

Robot Mugikorrak

Garaian garaiko garapen teknologikoarekin estuki lotua egon da robot kontzeptua. Ez da harritzekoa, hortaz, gaur egun oraindik terminoaren definizio estandar bat ematerik ez izatea. Gaur egun, *robot laguntzailea* eta *robot soziala* bezalako bikoteak erabili ohi dira, nabigazio edo industria arloko lanak utzi eta gizakiekin batera bestelako jarduera sozialago batzuetara bideratutako robotak adierazteko (Breazeal, 2004), (Goodrich and Schultz, 2007), (Nakano et al., 2011).

Ikerlan honen esparruan, gorpuzkera fisikoa duten eta helburu zehatzak betetzeko ingurune erreal dinamikoarekin elkarreragiten duten gailu *autonomoak* dira robotak. Beraz, robotak:

- Gorputz fisikoa behar du.
- Ingurunearekin elkarreragiteko, sentsoreak eta eragileak behar ditu.
- Erabakiak modu autonomoan hartu behar ditu.

Autonomo izateak, aldi berean, zenbait ezaugarri eskatzea dakar (Arkin, 1998), hala nola denbora-epe luzean funtzionamenduan aritzeko gaitasuna, ustekabeko egoerak identifikatu eta aurre egin ahal izateko gaitasuna, autoelikatzeko gaitasuna, funtzionamenduan erroreak detektatzeko gaitasuna, autorregulazioa, ikasteko ahalmena, etab. Gaur egun, oraindik ez dago horrelako autonomia maila erakusten duen robotik. Horregatik, ikerlan honen eremuan, helbururen bat lortzeko ingurunean gutxieneko autonomia mailarekin aritzeko gaitasuna erakusten duten gailuak izango dira robotak.

6.1 Robotikaren historia laburra

Robot *modernoak* XX. mendearen bigarren erdialdean garatzen hasi baziren ere, esan bezala robotak eraikitzea aspaldiko helburua da, eta, garaian garaiko aurrerapen teknologikoek ahalbidetuta, historian hainbat saiakera aipagarri topa daitezke, mekanismo fisiko hutsetatik hasi, eta fikziozko humanoideetaraino.

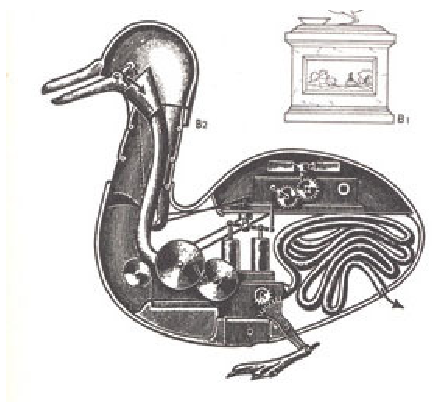
Robot terminoa bera 1921. urtean erabili zen lehenbiziko aldiz. Urte horretan Karel Čapek antzerki-zuzendari txekiarrak R.U.R (*Rossum's Universal Robots*) (Čapek, 2001) izeneko antzezlana aurkeztu zuen. *Robotnik* hitzak txekieraz morroi esan nahi du; *robota*-k, berriz, lan behartua, eta bi horietatik eratorri zuen. Antzezlanarekin, gizarte teknologikoan gizakiak pairatzen duen gizagabetasuna salatu nahi izan zuen, gizakien menpeko kopia edo klon artifizialak aurkeztuz. Robotak, alegia. Baina aurkeztu zituen robotak ez ziren mekanikoak, prozesu kimikoen ondorioa baizik, eta, beraz, gure irudimenean ditugun roboten aldean oso desberdinak.

Robotikaren historian mugarri izan diren gertakizun batzuk ekarriko ditugu hona segidan. Lehenengo kontrol-sistema automatikoak industria-aro modernoan sortu ziren arren, atzeraelikaduraren lehenengo aplikazioak K.a. 300. urtean kokatzen dira, Grezian. Aipagarriena, Alexandrian garatutako **Ktesibios-en ur-erlojua** izanik. Atzeraelikadura ezinbestekoa da robotetan ingurunea hauteman eta atazak modu eraginkorrean gauzatzeko.

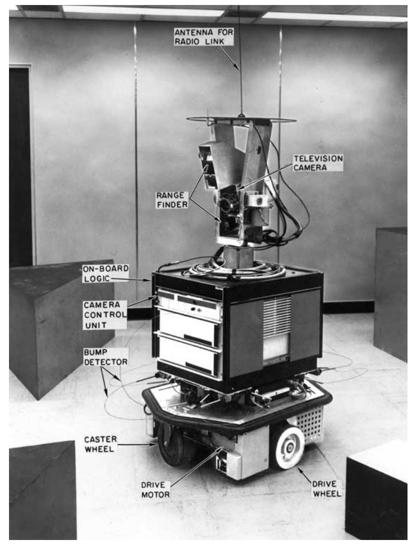
Lehenengo makina animatuak XVIII. mendearen hasieran koka daitezke. Horien artean ezagunenetako bat, Vaucanson-en ahatea da (ikus 6.1(a) irudia). Ahateak lepoa luzatzeko eta ahoan sartutakoa irensteko ahalmena zuen, eta ahateek elikatzeko egiten dituzten mugimenduak errepikatzeko gaitasuna ere bai.

Adimen Artifizial terminoa 1956. urtean jaio zen, John McCarthy-k Darmouth-eko konferentziaren gai nagusitzat "Adimen Artifiziala" proposatu zuenean. Esan bezala, AAk burmuin artifiziala garatzea izango du helburu, sistema fisiko errealarekin zerikusia duten kontuak eta ingurunearekin gertatzen den elkarrekintza erabat alboratuz. Lengoaia naturala, ezagutzaren adierazpen sinbolikoa eta ikasketa automatikoa dira, besteak beste, AAren izarrak, betiere goi-mailako adimenarekin erlazionatutako ezaugarriak, eta, ikuspuntu kartesiarretik begiratuta, gorpuzkera fisikoaren beharrik eta ingurunearen mendekotasunik ez dutenak.

AAren hastapenetan garatutako robot errealak urriak diren arren, azpimarragarriak dira *Shakey* (Fikes and Nilsson 1971) eta CART (H.



(a) Vaucanson-en ahatea



(b) Shakey robota

6.1 Irudia: Robot aitzindariak

Moravec 1965-79) robotak (ikus 6.1(b) irudia).

Bi robot aitzindari hauek bazuten ezaugarri komuna: denbora luzea behar zuten mugimendu-sekuentziak planifikatzeko eta denbora-tarte horietan geldirik iraun behar zuten, ez baitzekiten zer egin planifikazio-prozesuak iraun bitartean, itsu-itsuan mugitzeko aukera baino ez zuten, eta hori arriskutsuegia zen.

70eko hamarkadan izan zuen arrakasta handiena AAk robotak eraikitzeko hartutako hurbilketak, harik eta 80ko hamarkadaren amaieran kolokan jarri zen arte. 1986. urtean Rodney A. Brooks-ek Robotika Mugikorrarentzako paradigma berria proposatu zuen (Brooks, 1986), **Portaeran Oinarritutako Sistemak** (*Behavior-based systems*) izenarekin ezagutzera emango zena eta gaur egungo robot modernoen eragile nagusietakotzat jotzen dena. Brooks-en aburuz, sistema biologikoak izan behar dira robotak garatzeko oinarriak eta, adimena barik, lehenengo ezinbestekoa da ingurunearekin elkarrekintzan aritzeko gaitasuna erakusten duten gailuak eraikitzea. Paradigma honi jarraituz dago garatuta *iRobot* etxeko *Roomba*[1] xurgagailua, arrakasta komertziala lortu duen lehenbiziko robot mugikorra.

[1] https://www.irobot.es/

Brooks-ek robot adimendunak garatzeko hiru printzipio finkatzen ditu (Brooks, 1991):

1. **Gorpuzkera**: robotak gorpuzkera fisikoa eduki behar du; sentsore eta eragile errealak; simulazioak ez dira nahikoak. Gorpuzkerak baldintzatzen du robotak ingurumenarekin elkarreragiteko duen modua.

2. **Kokapena**: robota ingurumen errealean kokatu behar da; ezin du eragin ingurumenaren eredu sinboliko abstraktuetan. “Mundua bera da bere buruaren eredurik onena”.

3. **Portaera emergentea**: portaera adimenduna robotaren eta ingurumenaren arteko elkarrekintzaren ondorioa da.

Horrela, Brooks-ek paradigma edo filosofia berria proposatu zuen robot adimendunak lortzeko: *Portaeran Oinarritutako Sistemak*. “Portaerak” dira kontrolerako oinarrizko moduluak, eta kontrol-sistema elkarren artean komunikatzen diren portaera multzoez osatuta dago.

1997. urtean, Deep Blue makinak Gary Kasparov-i xakean irabazten zion bitartean (AA klasikoaren arrakasta), NASAk *Mars pathfinder Sojouner* ibilgailua bidali zuen Martera (Lurretik 500 milioi kilometrotara), beste planeta batean gertaera ezezagunen aurrean erreakzionatzeko gaitasuna zuen lehenengo robot adimenduna.

Aldi berean, Hondak bere lehenengo giza itxurako robota atera zuen, P3 izenekoa, bi zangoen bitartez ibili eta oreka mantentzeko ahalmenarekin. Arras ezaguna bihurtu den Asimoren aitzindaria.

Hortik aurrera, hamaika dira topa ditzakegun robot errealak, bai gurpildunak eta zilindro itxurakoak, bai eta giza edota animalia itxurakoak ere (6.2 irudia).

Gaur egun, apenas dagoen robotikak ukitu ez duen eremurik. Besteak beste:

- Etxean: *Roomba* robot-xurgagailua (Forlizzi and DiSalvo, 2006)

- Irakaskuntza eta ikerkuntzan: *Lego Mindstorms*, *Ozobot*, Arduino robotak eta *Cozmo* (García-Peñalvo et al., 2016).

- Museoetan: *KTBOT* museo gida (Susperregi et al., 2012)

- Industrian: *Kuka* (Schreiber et al., 2010), *Baxter* (Ju et al., 2014) eta *Kiva* (Guizzo, 2008).

- Ikuskizunetan: *NAO* robota eta Blanca Li (Li, 2016)
- Medikuntza eta erreabilitazioan: *Da Vinci* (Sung and Gill, 2001), *RP-VITA* (Dahl and Boulos, 2013) eta *Paro* (Wada et al., 2010)
- Erreskate-lanetan: *Versatrax* (inuktun, 2017)
- Esplorazioan: *Mars Curiosity* (Leshin et al., 2013)
- Gerra eta gatazka armatuetan: *Packbot* (Yamauchi, 2004).
- Robot sozialak: *Reeti*, *Jibo* eta *Buddy*.
- Cyborg itxurakoak: *BigDog* (Raibert et al., 2008)

6.2 Kontrol-arkitekturak

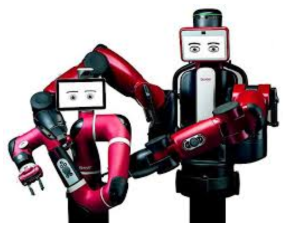
Azken urteotan, batik bat gizakion zerbitzura eta laguntzara bideratutako robotikan, sistema oso konplexuak agertu dira (Nakano et al., 2011) (Nooraei, 2012) (Fasola and Matarič, 2012) (Kortenkamp et al., 2016). Konplexutasun-maila hori, neurri handi batean, denbora errealean hainbat sentsore eta eragile kontrolatu beharrari egotzi behar zaio, sarritan ziurgabetasun eta dinamikotasun handiko inguruneetan.

Kontrol-arkitekturak robotaren kontrol-sistema (haren garuna) antolatzeko printzipioak ezartzen ditu. Robot-diseinatzaileari ataza behar bezala inplementatzen lagunduko dioten tresnak eskaintzen dizkio. Robotentzako kontrol-programak ugariak eta oso desberdinak izan daitezkeen arren, programa egokiak eta eraginkorrak sortzeko argibide eta printzipio batzuk ezagutu behar dira.

Kontrol-arkitekturak bestelako software-arkitekturekin konparatzeko eta termino baliokide gisa erabiltzeko joera izan, bada (Kortenkamp et al., 2016), baina sistema robotikoen beharrak ezberdinak izanik, ezin ditugu baliokidetzat har (Jones, 2004).

Ezberdintasunik nabarmenena, seguru asko, ondoko da: sistema robotikoek modu asinkronoan elkarreragin behar dutela, denbora errealean, ingurune ezezagun eta dinamikoan. Horrekin batera, sistema askok denbora esparru ezberdinetan erantzun behar dute: milisegundotan neurtutako *feedback*-etik hasi eta goi-mailako atazek behar izaten dituzten minutu edo orduetaraino. Beharrizan horiek kudeatzeko, robot arkitektura ugarik inplementatzen dituzte denbora-errealean erantzuteko gaitasuna, sentsore eta eragileak kontrolatzekoa, konkurrentzia edo exekuzio paraleloa,

(a) RP-vita



(b) Baxter eta Sawyer

(c) Buddy

(d) Packbot

(e) Versatrax 100

(f) Boston Dynamics

(g) ecce

(h) Lego Mindstorms

(i) Jibo eta Pepper

6.2 Irudia: Roboten irudi panoramikoa

ziurgabetasunaren kudeaketa, eta goi-mailako planifikazioa (sinbolikoa) eta behe-mailako kontrola.

Kontrol-arkitekturen diseinu eta garapenari buruzko azterketa zehatza, ikerketa-lan honen eremutik kanpo geratzen da. Informazio zehatzagoa behar duenak jo beza (Lazkano, 2004), (Astigarraga and Lazkano, 2012), (Matarić, 2007) eta (Kortenkamp et al., 2016) lanetara.

6.3 Gizaki-robot elkarrekintza

Gizaki-robot Elkarrekintza (HRI, *Human-Robot Interaction*) arloak, izenak berak adierazten duen gisan, gizaki eta roboten arteko elkarrekintza aztertzea du xede. HRI diziplinarteko arloa da, iturri ezberdinetatik edaten duena: Gizaki-konputagailu Elkarrekintza, Adimen Artifiziala, Robotika, Lengoaia Naturalaren Ulermena (*Natural Language Understanding*), Diseinua eta Zientzia Sozialak, garrantzitsuenen artean. Ikerkuntza arlo honek hartu duen garrantziaren lekuko gisa, horra azken hamarkadan HRI gaitasunak erakusten dituzten robotak (Goodrich and Schultz, 2007) (Sheridan, 2016). Hurbilagoko adibideak ere badira, (Susperregi et al., 2012) eta (Rodríguez et al., 2017) kasu.

Baina robot sozialak HRItik harago doaz. Breazeal-i kasu eginez (Breazeal, 2004), robot sozialak, gizakien eran gizartekoi agertzen diren robotak dira, eta beraiekiko elkarrekintzak gizakion artekoak bezalakoa izan behar du. Gizakion artean komunikatzeko biderik naturalena hizketa-bidezkoa dugu. Dena den, keinu-bidezko espresioa (hizketa ez dena) funtsezkoa da baita ere jendetasuna erdiesteko (Knight, 2011). Aurpegi adierazkotasunari buruzko ikerketa-lan mordoska topa genezake literaturan (Hanson, 2015) eta (Ishiguro, 2015). Breazeal-en Kismet buru robotikoak berak mugarria ezarri du gizakien ahotsak adierazkortasunean duen eraginari dagokionez. Bestalde, robot humanoideen ugaritzeak, roboten gorputz adierazkortasuna ikertu eta garatze bidean jarri du. Seguru asko, Pepper[2] robota izango da gaur egun gorputz espresiorako gaitasun handiena duen robot komertziala. Hankarik ez daukan arren, gerria eta besoak erabiltzen ditu hitz egiten duen bitartean giza-mailako adierazkortasuna erakusteko.

Robotekin egindako *performance* edo emanaldiek erakutsi dute erakusleiho egokia direla robot sozialen egungo egoera eta ahalmenak publiko orokorrari erakusteko, eta horrekin batera, baita roboten onarpen soziala neurtzeko ere. Holako antzerki-emanaldietan egin beharreko guztia

[2] Pepper Aldebaran Robotics. https://www.aldebaran.com/en/arobots/who-is-pepper

aldez aurretik prestatu eta entseatu den arren, antzerkiak inguru hobeezina eskaintzen du roboten portaera sozialak ikertu eta garatzeko, eta baita emozioen adieraztea eta lengoaia naturalaren bidezko komunikazioa hedatzeko ere (Lin et al., 2009) (Fernandez and Bonarini, 2014).

Baina esandako guztiaz gain, robot sozialek autonomoak behar dute izan. Gizakien kopia sintetikoak gehien-gehienetan aurre-programatutako edo telegidatutako robotak izan ohi dira. Aktore robotek erakusten duten autonomia-maila urrun da -bai urrun ere!- gizakiek erakutsi ohi dutenarengandik. Antzezlanetan aritzen diren roboten azterketa eta sailkapena jasotzen du Fernandezen lanak (Fernandez and Bonarini, 2014).

6.4 Ondorioak

Arloari buruzko azterketatik atera dezakegun lehen ondorio garbia hauxe litzateke: helburu orokorreko robotik ez da (Arkin, 1998), (Lazkano, 2004), (Jones, 2004), (Fernández and Mauricio, 2016). Robot bakoitzak bere ezaugarriak dauzka eta ataza jakin bat burutzeko diseinatua izan da. Beraz, aldez aurretik robotaren habia ekologikoa zein izango den eta bertan burutu beharko duen ataza ondo zehaztea komeni da. Robota, ingurunea eta ataza definituta dauzkagula, kontrol arkitekturaren bidez diseinatu eta inplementatuko dugu ataza burutzeko robotaren ezaugarriak ondoen ustiatzen dituen software-egitura.

Bertsobotari dagokionez, bertsolaritzak batean biltzen ditu robot adimenduna garatzeko beharrezkoak diren hainbat abilezia, hala nola: ahots-seinalearen ulermena, ordenagailu bidezko ikusmena bertsolariak eta oholtza gaineko beste ezaugarriak ezagutzeko, nabigazioa, elkarrekintza publikoarekin, gorputz-espresioa, etab. Bertsoaren *performancea* aztertu ostean (ikus 2.4 atala), robotaren portaera orokorra deskribatzeko moduan gara:

1. Gainerako kideekin oholtzan kokatu eta itxoin kantatzeko txanda iritsi arte.
2. Txanda iritsitakoan, mikrofonoa identifikatu eta bere aurrean kokatu.
3. Gai-jartzaileak emandako ariketa/gaia jaso.
4. Bertsoa osatu eta jendaurrean kantatu.
5. Publikoaren erreakzioa jaso eta horren arabera erreakzionatu.
6. Atzera buelta eta eseri aulkian.

Portaera orokor hori azalduko duen software-egitura diseinatzen hasi aurretik ordea, hartu beharreko erabakietan laguntzeko, kontuan izan behar genituzke ondoko galderak:

- Zeintzuk dira aipatutako atazak burutzeko beharrezko ekintza zehatzak? Beraien arteko koordinazioa nolakoa izango da?
- Ze datu behar ditu robotak ataza burutzeko? Nola lortuko ditu bai ingurunetik eta baita erabiltzaileengandik ere datu horiek?
- Nolakoa da robota, ze gorpuzkera dauka? Zeintzuk sentsore dauzka eta ze informazio jaso dezakegu bertatik? Nola mugitzen da?
- Robotaren konputazio-gaitasuna nolakoa da?
- Erabiltzaileek ze motako elkarrekintza izango dute robotarekin? Nola emango dizkiote aginduak eta nola interpretatu robotaren irteera?
- Kontrol-arkitektura inplementatzeko ze software egitura erabiliko dugu?

Galdera horiek buruan dauzkagula ekin beharko diogu Bertsobotaren kontrol-arkitektura inplementatzeari. Hala ere, robot arkitektura diseinatzea zientziatik baino artetik hurbilago dagoen langintza da (Kortenkamp et al., 2016).

Dena den, hutsetik baino eredu bat izanda abiatzea errazagoa denez, robot-gizaki elkarrekintza ikuspegitik robotekin burututako antzerkigintza izan dugu erreferente. Izan ere, azken urteetan antzerkigintzara bideratutako sistema robotiko ugari garatu dira (Fernandez and Bonarini, 2014) (Lemaignan et al., 2012) (Hoffman et al., 2008). Arrakastaren lekuko, antzerkia eta robotika uztartzen dituen udako eskola[3] dugu baita. Akademiaren mundutik aparte, ikuspegi artistikoagoa eman dionik ere bada, Blanca Li dantzariak NAO robotekin batera[4] daukan emanaldia kasu; dantza, antzerkia eta robotika uztartzen dituena.

Gure ustez, bertsolaritzak roboten gorputz espresioa eta gizaki-robot komunikazioa lantzeko testuinguru egokia eskaintzen digu, antzerkietako emanaldiekin antzekotasunak dauzkana baina, bere horretan, ezaugarri propio aski interesgarriak ere badauzkana robot sozialagoak lortze bidean.

Hona, gure ustez, antzerki emanaldi eta bertso saio edo emanaldi baten arteko ezberdintasunik nabarmenenak:

[3] https://cs.uiowa.edu/resources/robot-theater-summer-camp
[4] http://blancali.com/en/event/99/Robot

- Antzerki emanaldietan gidoia aurrez emana dago eta inprobisaziorako tartea oso mugatua da. Antzezleak esaldi baten ordena, ekintzen erritmoa eta gorputz baliabideak alda ditzake emanaldi batetik bestera baina ez esanahia. Bestetik, bertsolaritzan, oholtza gaineko mugimenduak ia beti berberak dira, aldaketa gutxirekin saio batetik bestera. Baina kantatu beharrekoak momentuan bertan sortu behar dira, gai-jartzaileak jarritako lanaren arabera, eta ondorioz, ez dira bi emanaldi berdin izango.

- Antzezlanean elkarrizketa oinarrizkoa da, aktoreak elkarri edo publikoari hitz eginez. Bertsolariak kantatu egiten du, kantulagunari (binaka ari direnean) eta baita publikoari ere. Gai-jartzailearekin hitz batzuk trukatu ditzake baina nekez elkarrizketa bat izateko adina. Bertsoa da bere komunikazio erreminta.

- Antzerki emanaldietan taula gaineko agertokia alda daiteke. Bertso saioetako egitura dezente alda daiteke saio batetik bestera, baina ia beti izango ditu bertsolariak pare bat erreferentzia: kanturako mikrofonoa eta aulkia.

- Antzerki emanaldietan, publikoaren parte hartzea emanaldiarekiko asebetetze-maila erakustera mugatzen da (antzerki klasikoan,esperimentalean publikoak gehiago eragiten du), baina bertso saioetan publikoaren erreakzioak saioa baldintzatu dezake (Sarasua, 2004).

Beraz, robotean portaera sozialak garatu eta probatzearen ikuspegitik, bertsolaritza eta antzerkia, biak ala biak dira interesgarri. Antzerkian gorputz adierazkortasunak pisu handiagoa izango du eta, aldiz, bertsolaritza, Lengoaia Naturalaren Sorkuntza lantzeko egokiagoa izango da.

Esan bezala, robot-bertsolariak berekin behar du jarduera soziala, inguruneko pertzepzioei erantzun behar die eta autonomoa izan behar du bere egitekoan. Mezuak, bertsoak, unean bertan sortu eta plazaratu behar ditu, uneko pertzepzioen baitan, eta robotak emanaldiko gertaerak jaso eta haiei erantzuteko gaitasuna behar du dudarik gabe. Eta guzti hori ahalik eta modurik naturalenean egin, jakina.

7. Kapitulua

Konputagailu-burmuin interfazeak

Neurozientzia Kognitiboaren eta Burmuin Irudigintzaren (*Brain Imaging*) alorreko aurrerakuntzek, giza burmuinarekin elkarreragiteko bide berriak urratu dituzte. Helburu nagusia burmuinaren seinale bidez adierazitako intentzioa jaso eta kontrol-agindu bihurtzea izanik. Kapitulu honetan arloko egoeraren berri emango dugu, garunean kokatutako sentsoreekin emozioak detektatu ahal izateko aukerak aztertuz.

7.1 Emozioen neurketa

Emozioak izaki guztion funtsezko parte dira, beste alor askotan bezala, beren definizioa eztabaida iturri izan da eta diziplina ezberdinetatik aztertua (Kleinginna and Kleinginna, 1981). Emozioen funtsa bera ere perspektiba edo ikuspegi ezberdinetatik landu izan da. Emozioak buru-egoeren egoera psiko-fisiologikoak dira, kanpoko edo barneko estimuluekiko erantzuna adierazten dutenak (Gratch and Marsella, 2005). Psikologiaren ikuspegitik, emozioek atentzioa aldatu edo finkatzen dute, norberaren portaera indartzen dute eta memoriarentzako egokiak diren asoziazio edo elkarketa sareak aktibatzen dituzte. Ikuspegi fisiologikoago batetik, emozioek sistema biologikoen erantzunak antolatzen dituzte, aurpegiko espresioak, muskuluak, ahotsa, nerbio sistema autonomoaren aktibitatea eta baita sistema endokrinoarena ere. Beraz, laburbilduz, emozioak estimuluekiko erantzunak dira, norbere aldarte-egoeraren arabera konexio eta intentsitatean aldakorrak direnak.

Gizakion emozioen ezaugarri anitzak kontuan izanik (Alzua-Sorzabal et al., 2012), hauek neurtzeko hiru metodo nagusi daude: portaeraren obserbazioa, neurketa psiko-fisiologikoak eta txosten subjektiboak. Lehen bi metodoak nerbio sistemaren aktibitatea neurtzen dute eta, ondorioz, tresna espezializatuak behar dira horretarako. Txosten subjektiboetan oinarritutakoak, aldiz, eskala emozionaletan oinarritutako galdetegiak erabiltzen dituzte indibiduoaren egoera jasotzeko.

Txosten subjektibo edo indibidualetan oinarritutako metodoetako bat *Pleasure-Arousal-Dominance* (PAD) izenekoa da. Bertan, emozioen espazioa hiru dimentsiotara murrizten da:

- ***Pleasure***: erabiltzailearen egoera afektibo positibo edo negatiboaren neurria jasotzen du. Bestela esanda, erabiltzailearen atsegin sentsazio subjektiboa.
- ***Arousal***: indibiduoaren aktibazio maila, lo egoeratik aktibitate frenetikoraino doan tartean.
- ***Dominance***: indibiduoaren sentipenen askapen-maila neurtzen du azken honek.

(Alzua-Sorzabal et al., 2012) lanean, PAD metodoaren inplementazioa planteatzen da testuinguru kulturan batean, Donostia 2016 Kultur Hiriburu izateko hautagai zen garaian hain zuzen ere. Hiriaren hautagaitza defendatu zuten egunean, taldeko partaideen emozioak neurtu zituzten aipatutako metodologian oinarrituz. Horretarako, mugikorretarako Android aplikazioa garatu zuten, emozio-neurketa une jakinetan aktibatzen zena eta erabiltzaileari emozio-egoeraren berri galdetzen ziona.

Bestalde, (Imbir et al., 2015) lanean, elektroentzefalografia (EGG) ikerketa burutu zuten 25 emakumerekin. Bertan, emakumeei karga emozionaldun hitz-multzoa eman zieten irakurtzeko eta beraien erantzunak neurtu zituzten. Alde batetik erantzun automatikoak, berehalako erreakzioak; eta, bestetik, hitzen bitartez zehaztu behar izan zituzten emozio erreflexiboagoak. Hitzen karga emozionala positibo, negatibo edo neutrala zen. Emaitzetan, emozioen arteko zenbait ezberdintasun neurtzeko gai izan ziren, bai jasotako seinalearen nolakotasunari buruz, eta baita seinalearen burmuin-kokapenari buruz ere.

Garunean kokatutako sentsore ez inbasiboekin emozioak detektatu ahal izateak, gizaki-robot elkarrekintzarako bide berriak zabaltzen ditu.

7.2 Konputagailu-burmuin interfazeak

Azken aldiko aurrerakuntzek, Neurozientzia Kognitiboaren eta Burmuin Irudigintzaren (*Brain Imaging*) alorrekoak batez ere, giza burmuinarekin zuzenean elkarreragiteko ateak zabaldu dizkigute, konputagailu-burmuin arteko komunikazio eta kontrolerako aukera berriak agertuz. Bide horretatik, azken urteotan, konputagailu-burmuin interfaze (BCI, *Brain Computer Interface*) ugari garatu dira (Guger et al., 2015).

BCI sistemen helburu nagusia komunikazio eredu bat ezartzea da, burmuinaren seinale bidez (elektrikoak, kimikoak edo odol presioaren aldaketa bidez antzemandakoak) adierazitako intentzio edo ataza mentalak kontrol agindu bihurtu ahal izateko. Irteera, konputagailu aplikazio batera bideratu daiteke edo neuroprotesi batera baita (muskuluen erreakzio beharrik izango ez lukeena). Asmoa paralisiren bat sufritu arren gaitasun kognitiboak mantentzen dituzten horiei komunikazio bide berri bat ezartzea izanik.

BCI sistemetako motor-imajinagintzaren oinarrian zera datza: mugimenduen imajinatze edo pentsatzeak kortexean burmuinaren aktibitatearen aldaketak eragiten dituela. Hortaz, zenbait mugimenduri lotutako patroiak antzemanez gero, hauek kontrol seinale bihurtu ahalko genituzke.

Bi eratako BCI sistemak aurki ditzakegu: teknologia inbasiboa baliatzen dutenak, sentsoreak zuzenean burmuinean bertan txertatuz, eta, bestetik, teknologia ez-inbasiboak, burmuinaren aktibitatea neurtzeko kanpo-sentsoreak erabiltzen dituztenak. BCI aplikazio gehienetan, azken mota honetako sentsoreak erabiltzen dira. Zehazki, elektroentzefalografia sentsoreak (EEG), aktibitate elektrikoaren berri jasotzen dutenak.

Ikerketa-eremu berri honek zabaltzen dituen aukerak ikaragarriak dira, baina, bidean topatuko ditugun erronkak ere ez dira nolanahikoak. Adibidez, BCI sistemak garatzerakoan, pertsona bakoitzarentzako aukeratu beharreko kanalen kopurua eta kokapena (EGG kaskoetan) funtsezko zeregina da. Gehienetan burmuin seinaleak jaso eta sailkatzeko kanal asko jartzearen aldeko hautua egiten da, baina horrek denbora aldetik kostu handia dakar berarekin eta sisteman zarata sartzea ere suposa dezake. Beraz, sistemak eskatzen duen zehaztasun maila emateko kanal kopuru minimoa aukeratzea erabakigarria da.

Aukeraketa hori automatikoki egiten duten sistema ugari aurkituko dugu literaturan, nabarmenenak aipatzearren: (Barachant and Bonnet, 2011), (Yang et al., 2012) eta (Lal et al., 2004).

Gaiari buruzko azterketa eguneratu eta osoagoa nahi duenak, jo beza ondoko lanetara: (Arvaneh et al., 2011) eta (Kee et al., 2012).

7.3 Ondorioak

Konputagailu-burmuin interfazeen ikerketan murgildu izanaren arrazoiak bi dira, oso desberdinak euren artean: lehenbizikoa eta garrantzitsuena, burmuin-aktibazio bidez erreakzioak eta emozioak jaso ahal izatea. Hau da, batetik bertsoek publikoarengan eragiten dituzten emozioak identifikatu eta kokatzea EEG bidez. Bigarrena, robotarekiko interfazea hobetu eta pentsamendu bidez oinarrizko aginduak eman ahal izatea (zutitu, etorri, gelditu, atzera, aurrera eta antzekoak).

Artearen egoeran azaldutako bi lanetatik edatea litzateke gure asmoa. Hau da, batetik bertsoek publikoarengan eragiten dituzten emozioak identifikatu eta kokatzea EEG bidez. Oinarrizko emozioak, positibo, negatibo eta neutroa adibidez. Eta, bestetik, sumatutako emozio horiek robotera bidaltzea honek emozio-egoera orokorraren berri izan dezan. (Westerink et al., 2009) lanean adibidez, egoera emozionalak eta hauen karga modu automatikoan detektatzeko sistema proposatzen dute. Erabilitako sentsoreak, elektrokardiograma sentsorea (*Electrocardiogram sensor, EGC*) eta azaleko ezaugarri elektrikoak neurtzen dituena (*Skin conducctance, SC*) dira eta bereziki prestatutako mugikor batera bidaltzen dituzte seinaleak bertan prozesatu ahal izateko.

Gu, lehen pausu gisa, Emozionometro lanaren egokitzapena planteatzen ari gara, hau da, aplikazio-bidez publikoaren uneko egoeraren berri jaso eta horren berri zuzenena robotari bidaltzea.

III. atala

Ekarpenak

8. Kapitulua

Bertso Sorkuntza Automatikoa

Ikusi dugun bezala, konputagailu-bidezko poesia sorkuntzak ikerketa komunitatearen atentzioa lortu du azken urteetan, sistema ezberdin ugari eta aski interesgarriak garatu direlarik. Lanik aipagarrienak aztertu ostean (ikus 3.2 atala), bertsoak osatzeko diseinatu eta garatu ditugun metodo propioak azalduko ditugu kapitulu honetan.

Laborderen arabera, bertsolariek ondoko hiru tresnetan oinarritzen dute beren sorkuntza lana (Laborde, 2005):

1. Bat-bateko bertsoak sortzeko beharrezko teknika eta arauak, metrika eta errimaren erabilerari dagozkionak.

2. Memoria, aurretik entzundako bertsoak gorde eta sailkatzeko, eta baita informazio bisuala eta lexikala gordetzeko ere.

3. Bertsoa sortu aurreko uneetan jasotako estimulu sentsorialak.

Gure ekarpenak ere, Labordek zehaztutako hiru zutabe horien inguruan kokatu genitzake: batetik, bertsoa osatzeko estrategia orokorra eta horretarako beharrezko diren tresnak. Hor emango dugu testu berriak sortzeko gure hurbilpenen berri eta bertsoaren arau formalak betearazteko tresnen zehaztapen eta erabileraren berri ere. Bestetik memoria, Bertsobotaren memoria nola osatua dagoen eta bertatik bertso berriak sortzeko informazioa nola berreskuratzen duen azalduko dugu. Bertsolarien lan-tresna garrantzitsua da memoria, han gorde eta sailkatzen dituzte entzundako bertso, irakurritako testu eta ikusitako irudiak, abagune egokia iristean bertso-moldera ekarri ahal izateko. Bertsolarien bat-bateko jarduna ez baita momentuan sortzea bakarrik, etxean landutakoa egoki txertatzea ere bada (Garzia, 2000). Azkenik, berriz, oinarrizko emozioak testu-sorkuntzan txertatzeko moduaz arituko gara.

Aurrekoei gehitu beharreko laugarren ekarpen-eremua ere badugu, metodologiari buruzkoa. Bertsoa osatzeko estrategia eta erabilitako baliabideekin lan-modu edo metodologia zehatza garatu dugu. Metodologia horrek baliabide gutxirekin bertsoak sortzeko bidea ezartzen du eta baita gure sorkuntza-lana aztertu eta darabilgun baliabideen egokitasunari buruzko azterketa burutzeko ere.

8.1 Testu-sorkuntzarako estrategia, tresna eta baliabideak

Bertsoak automatikoki sortzeko oinarrizko bi estrategia diseinatu eta inplementatu ditugu:

- **Corpus-erauzketa metodoa**: testu-corpus batetik puntuak atera eta hauek konbinatzean oinarritzen den metodoa. Hurbilketa hau, Informazio-Erauzte arloko ataza gisa ikus liteke, helburua dagoeneko existitzen diren esaldiak atera eta hauen arteko konbinazio berriak sortzea izanik. Corpusetik ateratako esaldiek neurri eta errimak ezarritako baldintzak bete beharko dituzte. Hala ere, inplementatzeko estrategia erraza da bere horretan. Gakoa, bertsoa osatuko duten puntuen arteko konbinaketa egokian datza, noski.

- **N-grama eredu probabilistikoa**: N-grama ereduetan oinarritutako bertso-sortzailea. Metodo honek eredu gisa corpus bat jaso eta ereduaren estilo bereko testu berriak sortzen ditu, errima eta metrikaren arau formalak betez gainera. Abiapuntu edo eredu gisa emandako corpusetik bi Markov kate eraikitzen dira, bata aurrera begirakoa eta beste atzera begirakoa. Bi eredu horiek erabiltzen dira bertsoa puntuz-puntu osatzeko. Bertsoa osatzerakoan, bertsolarien estrategia bera erabiltzen du, lehenbizi azken puntua, ideia nagusia, zehaztu eta segidan puntuz-puntu bertsoa goitik behera josiz. Eraikitako sistemaren ekarpen nagusia, bertsoaren arau formalak eta puntuen arteko semantika neurtzeko sistema N-grama testu-sortzailean txertatu izana da.

Aipatutako sorkuntza-estrategiak, oinarrizko tresna eta baliabideak erabiliz eraikiak izan dira. Hona berauen azalpen laburra:

- **Silaba-kontatzailea**: sarrera gisa emandako hitz edo hitz-segidaren silaba-kopurua itzultzen du. Silabifikazio-algoritmo gisa, Jauregik proposatutako (Jauregi, 2013) metodoa inplementatu dugu adierazpen-erregularrak erabiliz.

- **Errima-bilatzailea**: errimatzea funtsezkoa da bertsoak osatzeko, errimarik gabe ez dago bertsorik. Helburu hori betetzen duen funtzioa inplementatu dugu, bertsogintzan erreferentzia den Hiztegi Errimatua (Amuriza, 1981) erabiliz gida bezala. Han zehaztutako errima-patroi edo taldeak inplementatzeko adierazpen-erregularrak baliatu ditugu baita.

- **Corpusak**: bertsoak osatzeko eta ereduak eraikitzeko testu-iturri gisa baliatu ditugu.

  - **Egunkaria**: Egunkariako 2000-2001 urteetako testuekin osatutakoa. 981888 paragrafo, 1277457 esaldi, 15609348 termino eta 839659 termino bakar.
  - **Txapelketako-bertsoak**: 1986-2009 arteko Bertsolari Txapelketa Nagusiko bertsoez osatua. Guztira 6887 bertso, 41322 puntu, 208160 termino eta 26138 termino bakar. Bertsoak sortzeko corpusean molde txikikoak (7-6 silaba) soilik erabili ditugu: 5545 bertso, 22180 puntu, 29234 termino eta 9507 termino bakar.
  - **Txirrita**: Jose Manuel Lujanbio "Txirrita" [1] bertsolariak jarritako bertsoekin osatutako corpusa[2]. 2127 puntu eta 24277 termino eta 6998 termino bakar osatua.

Aipatutako estrategiak jarraituz, Bertsobota gai da ondoko ariketak burutzeko bat-bateko bertsolaritzak eskatzen duen denbora tarteetan:

- Lau oinak emanda bertsoa osatu.

- Hasierako puntua emanda bertsoa osatu.

- Hitz bat gai gisa eman eta bertsoa osatu.

Bertsobotaren testu-sorkuntza estrategiak eta tresnak, ondoko artikuluetan zehazten dira:

[1] https://eu.wikipedia.org/wiki/Jose_Manuel_Lujanbio
[2] http://klasikoak.armiarma.eus/idazlanak/T/Txirrita.htm

- **Bertsobot: the first minstrel robot**. A. Astigarraga, M. Agirrezabal, E. Jauregi, E. Lazkano, B. Sierra. 6th International Conference on Human System Interaction (HSI), 2013ko ekaina, Sopot, Polonia.

- **Markov Text Generator for Basque Poetry**. A. Astigarraga, J.M. Martínez-Otzeta, I. Rodriguez, E. Lazkano, B. Sierra. 20th International Conference on Text, Speech and Dialogue (TSD), 2017ko abuztua. Czechia, Praga (**ONARTUTA**).

Bertsobotarekin zerikusi zuzenik izan ez arren, hemen azaldu dugun N-grama testu-sortzailea bestelako testuinguru batean ere aplikatu genuen: Fran Meana artistak Tabakalerako[3] egonaldian garatu zuen *El Laboratorio de Formas*[4][5] proiektuan hain zuzen ere. Zehazki, N-grama sortzaileak erabili genituen Díaz-Pardoren obra (*Discusión sobre organización de industrias manufactureras*) elkarrizketa gisa erakusteko, ikuslearekin elkarreraginez.

8.2 Koherentzia semantikoa

Bertsoaren puntuak sortu edo corpusetik atera eta horiek ausaz konbinatuta lortzen den bertsoak, ez du izaten ez hanka ez buru gehien-gehienetan. Ez bada, surrealistek zori objektibo bezala izendatu zuten erlazioa (Lejeune, 2012); hau da, elkarren ondoan paratutako errimei, adibidez *agurra/beldurra*, *ezina/egina* edo *bihotz/hotz*, semantikoki ere lotura topatzen diegula.

Ausazko loturez gaindi, lotura intentzionala lortu nahi genuke. Bertsoak koherentzia izatea, bai puntuen artean eta baita helarazi nahi den mezuarekiko ere. Sortutako bertsoen barne kohesioa, puntuen arteko lotura neurtzeko modua behar da horretarako. Ondoko metodoak garatu ditugu:

- Vector Space Model (VSM) (Salton et al., 1975) (Lee et al., 1997): puntuak bektore gisa adierazi, hitz bakoitza bektoreko elementu delarik, eta bektoreen, puntuen, arteko kosinu bidezko distantzia kalkulatzen du metodo honek. Interpretatzen oso erraza da: bektoreen artean zenbat eta hitz kointzidente gehiago izan, orduan eta elkarrengandik semantikoki hurbilago puntuak. Hitz edo termino guztiek, noski, ez dute pisu bera testuan. Pisuak erabakitzeko, *tf-idf* (Ramos, 2003) pisaketa-sistema erabili dugu. VSM metodoaren

[3] Tabakalera, Kultura Garaikidearen Nazioarteko Zentroa. Donostia
[4] http://www.nosotros-art.com/revista/entrevistas/fran-meana
[5] https://www.tabakalera.eu/es/la-historia-del-laboratorio-de-formas-fran-meana

eragozpenik handiena da, antzekotasuna terminoen kointzidentzian soilik oinarritzen duela. Izan ere, gai beraren bueltan aritu arren termino ezberdinak erabiltzen dituzten bi testu ez lituzke antzekotzat joko. Hau da, polisemia eta sinonimia ez ditu tratatzen.

- Latent Semantic Analysis (LSA) (Landauer et al., 2013) (Zelaia, 2016): VSMren hurbilketa berezia, aipatutako arazoak ekiditeko sortua. Funtsean, testu idatzien semantika adierazteko gaitasuna duen tresna da LSA, terminoen arteko erlazio semantikoen neurria itzultzen duena. Horretarako matrizeen aljebra erabiltzen da, SVD izeneko matrize-banaketa, eta emaitza gisa LSA-ak terminoen kointzidentziaz gaindi, esaldien arteko lotura semantikoaren neurria itzultzen du.

Aipatutako metodoak bertso-sorkuntza prozesuan txertatu ditugu eta lan ia guztietan ditugu aipagai. Dena den, ondoko artikuluak jasotzen ditu lehen aldiz:

- **Textual coherence in a verse-maker robot**. A. Astigarraga, E. Jauregi, E. Lazkano, M. Agirrezabal. Human-Computer Systems Interaction: Backgrounds and Applications 3, Advances in Intelligent Systems and Computing Series, volume 300. Springer International Publishing.

8.3 Emozioak txertatzea

Bertsoaren helburua, mezu jakin bat transmititzeaz gain, entzulearengan emozioak sorraraztea ere bada (Garzia, 2000) (Larrañaga, 2013). Bertsolariak berak ere, gaiaren eta testuinguruaren arabera, ez du beti aldarte beretik kantatzen. Gerta liteke, *etorkizuna* gai bezala jarri eta une jakin batean mezu positibo bat transmititu nahi izatea eta, hurrengo batean, gai bera jarrita ikuspegi negatibotik kantatzea.

Guk ere bertso-sorkuntza metodoetan emozioak txertatzeko modua topatu nahi izan dugu. Emozio-egoera jakin bat abiapuntu gisa emanda, aldarte-egoera horrekin bat datozen bertsoak sortu, alegia.

Garatu dugun sistemak, sarrera gisa gaia eta aldarte-egoera (positibo, negatibo edo neutroa) jaso eta bertsoa osatuko du:

1. Errima eta metrikari dagozkion arau formalak betez.

2. Gaiari dagokion edukia aukeratuz, hau da, koherentzia semantikoa erakutsiz.

3. Adierazitako aldarte-egoera erakusten duen bertsoa (testua) osatuz.

Beraz, sistemari eskatu geniezaioke *maitasuna* modu negatiboan ikusteko edo *udaberria*ri buruz kantatzeko emozio neutro batetik. Helburua publikoari emozio jakin bat transmititzea litzateke eta, horrekin batera, baita kanpotik jasotako estimuluei erantzuteko gaitasuna garatzea ere (adibidez, txalorik egiten ez badu jendeak aldartez aldatzea bertsoa sortzerakoan).

Sentimenduen analisia burutzeko, **EliXa** tresna erabili dugu (San Vicente et al., 2015). Testu motzetan, sentimendu positibo, negatibo eta neutroen balioak neurtzen ditu, sailkatze-ataza klase anitzeko SVM algoritmo bidez burutzen duelarik.

Arlo honetan egindako ekarpena, ondoko artikuluak jasotzen du:

- **Emotional Poetry Generation**. A. Astigarraga, J.M. Martínez-Otzeta, I. Rodriguez, E. Lazkano, B. Sierra. 19th International Conference on Speech and Computer (SPECOM), 2017ko iraila, Hatfield, Hertfordshire, UK (**ONARTUTA**).

8.4 Sorkuntza metodologia orokorra

Aurreko ataletan azaldutako bertso-sorkuntza estrategiak eta semantika ereduak metodologia orokor baten baitan bildu ditugu. Metodologia horren helburua bikoitza da:

- Baliabide linguistiko minimoarekin bertso edo poemak sortzeko modua eskaintzea.
- Hizkuntza ezberdinetan burututako bertso-sorkuntza konparatu ahal izatea.

Metodologia orokorra inplementatzen duen tresna garatu dugu baita, *Poet's Little Helper* (PLH) izena jarri diogu. Tresna honekin, ikertzaileak baliabide gutxi batzuekin poema edo bertsoak sortu ahal izango ditu eta, bide batez, erabilitako baliabideei buruzko egokitasun azterketa burutu baita ere. Sistemak behar dituen sarrerako baliabideak honakoak dira: testu-corpusa, silaba-kontatzailea eta errima-egiaztatzailea. Erreminta horiekin, bertso edo poemak sortzeko estrofa-mota eta errima-eredua sarrerako datu-gisa eman eta PLH tresnak ondoko ekintzak burutzen ditu:

- Bertsoak osatzeko corpusaren egokitasun-maila aztertu. Lexiko-mailan eta maila semantikoan burutzen da azterketa hori.

- Gaia eta bertso-moldea jasota, gaiari buruzko bertso-sorta itzultzen du. Arau formalak betetzeaz gain, gaiarekiko hurbiltasun semantikoa ere erakutsiz.

PLH tresnak baliabide gutxirekin poema edo bertsoak sortzeko modua ezarri nahi luke, hizkuntza txiki edo baliabide gutxikoei ateak zabalduz. Bestetik, hizkuntza ezberdinak inplementatzeko erraztasunari esker, hizkuntzaren ezaugarriek bertso-sorkuntzan nola eragiten duten aztertu nahi genuke. Hala ere, txosten hau idazteko garaian, inplementatutako hizkuntza bakarra euskara da.

Hizkuntza berri batean PLH tresna erabili ahal izateko, ondoko baliabideak beharko genituzke:

- **Errima-bilatzailea**: bi hitz jaso eta errimatzen duten ala ez itzultzen duen funtzioa.
- **Silaba-kontatzailea**: sarrera gisa jasotako karaktere-katearen silaba-kopurua itzultzen duena.
- **Corpusak**: testu-corpusak bi helbururekin erabiliko ditugu:
  - Eredu semantikoa eraiki eta bertso-lerroen antzekotasuna neurtzeko.
  - Bertatik esaldiak erauziz poemak edo bertsoak osatzeko.

  Corpus bera erabili daiteke bi zereginetarako edo, nahiago izatera, corpus berezituak. Adibidez, eredu semantikoa eraikitzeko iturri ezberdinetako dokumentu-multzo handia erabili genezake eta, bestalde, bertsoak osatzeko corpus gisa askoz txikiagoa eta espezifikoagoa den beste bat, poesia edo bertsoen hizkuntzatik hurbilago legokeena.
- **Lematizatzailea**: hitz bat jaso eta haren lema edo erroa itzultzen duena. Ez da ezinbestekoa baina bai gomendagarria. Ikerketa ezberdinek erakutsi dute (Zipitria et al., 2006) (Lifchitz et al., 2009), batez ere hizkuntza eranskarietan, eredu semantikoaren zehaztasunak behera egiten duela erabiltzen ez den kasuetan.

Aipatutako baliabideak izanda, poema edo bertsoak sortzeko prozesua hiru pausutako zikloa da: analisi lexikoaren esplorazioa, analisi semantikoaren esplorazioa eta bertso-sorkuntza.

Pausu bakoitzean ostean, sarrera gisa erabilitako datuen egokitasunari buruzko datuak jasoko ditugu, parametroak eta datuak aldatu/egokitu ahal izango ditugularik.

- **Analisi lexikoaren esplorazioa**: sarrera gisa corpusa eta sortu beharreko poemaren ezaugarriak (metrika, errima-egitura) emanda, ezaugarri horretako bertsoak sortzeko corpusak daukan egokitasuna neurtzen da. Errima-multzo bakoitzeko sor litezkeen bertso potentzialen kalkulua (sail bereko zenbat errima-hitz eta oin hori daukaten zenbat puntu) kalkulatzen da, besteak beste.
- **Analisi semantikoaren esplorazioa**: sarrera gisa jasotako corpusarekin eredu semantikoa eraiki eta eredu hori erabiliko da emandako gaiarekiko hurbiltasun handieneko esaldiak bilatzeko batetik, eta, bestetik, bertsoaren hautagai-puntuen hurbiltasun semantikoa neurtzeko.
- **Bertso-sorkuntza**: hirugarren eta azken pausuan, ezarritako gaiarekin ondoen ezkontzen diren esaldiak (puntuak) aukeratu eta bertsoa osatzen du adierazitako moldean (bertso-egituran).

Lehen bi pausuen ostean lortutako emaitzek nahikoa informazioa ematen digute aztertu ahal izateko, bertsoak kaskarrak diren kasuetan, emaitza kaskar horiek erabilitako corpusen ondorio diren (corpus txikiegia edo ez egokia errima/metrika baldintzetan), edo eraikitako eredu semantikoaren erruz (eredua ez delako gai hurbiltasuna atzemateko) edo, errima eta metrikaren murriztapenen (soineko estuegiaren) ondorio diren emaitza txarrak.

Aurkeztu berri dugun metodologia orokorra eta PLH tresnaren inplementazioa, ondoko artikuluak jasotzen du:

- **Poet's Little Helper: A methodology for computer-based poetry generation. A case study for the Basque language**. A. Astigarraga, J.M. Martínez-Otzeta, B. Sierra, I. Rodriguez, E. Lazkano. International Workshop on Computational Creativity in Natural Language Generation (CC-NLG), 2017 (**BIDALITA**).

Textual Coherence in a Verse-Maker Robot

A. Astigarraga[1], E. Jauregi[1], E. Lazkano[1], and M. Agirrezabal[2]

[1] Department of Computer Science and Artificial Intelligence,
University of the Basque Country (UPV/EHU), Donostia, Spain
{aitzol.astigarraga,e.lazkano,ekaitz.jauregi}@ehu.es
[2] Department of Language and Software systems,
University of the Basque Country (UPV/EHU), Donostia, Spain
Manex.aguirrezabal@ehu.es

Abstract. The *Bertsobot* project aims to develop an autonomous robot capable of composing and playing traditional Basque impromptu verses –*bertsoak*. The system should be able to construct novel verses according to given constraints on rhyme and meter, and to perform it in public. The *Bertsobot* project, at the intersection of Autonomous Robotics, Natural Language Generation and Human Robot Interaction, works to model the human abilities that collaborate in the process that enables a verse-maker to produce impromptu verses. This paper provides a general overview of the system, specially focusing on the description and evaluation of different semantic similarity methods for predicting the textual coherence of the generated verses.

1 Introduction

Basque, *Euskara*, is the language of the inhabitants of the Basque Country. And *bertsolaritza*, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive. Events and competitions in which improvised verses, *bertso*-s, are composed are very common. In such performances, one or more verse-makers, named *bertsolari*-s, produce impromptu compositions about topics or prompts which are given to them by an emcee (theme-prompter). Then, the verse-maker takes a few seconds, usually less than a minute, to compose and sing a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme. Melodies are chosen from among hundreds of tunes. Fig. 1 shows a picture of the national championship of *bertsolaritza*, which took place on 2009.

Xabier Amuriza, a famous verse-maker that modernized and contributed to the spread out of the *bertsolaritza* culture, defined *bertsolaritza* in a verse as:

Neurriz eta errimaz	*Through meter and rhyme*
kantatzea hitza	*to sing the word*
horra hor zer kirol mota	*that is what kind of sport*
den bertsolaritza.	*bertsolaritza is.*

The main objective of this research project is to develop an autonomous robot capable of generating and performing improvised verses in Basque. The interaction

Z.S. Hippe et al. (eds.), *Human-Computer Systems Interaction: Backgrounds and Applications 3*, Advances in Intelligent Systems and Computing 300,
DOI: 10.1007/978-3-319-08491-6_23, © Springer International Publishing Switzerland 2014

with the robot should be speech-based; thus, the system should be able to receive the instructions to compose the verse in Basque, to generate the most appropriate verse according to the given instructions and to sing it with the proper melody. The robot should also show the degree of expressiveness that Basque troubadours, *bertsolari*-s, show in their performance. And all those tasks must be accomplished concurrently in a extemporaneous performance.

We believe that the *Bertsobot* project provides a unique opportunity to join together the capabilities of autonomous robots to sense their environment and interact with it, and the natural language processing tools devoted to automatic verse generation, in an attempt to generate improvised and context-specific poetry.

We decided to decompose the task of performing a *bertso* into several smaller tasks or abilities, so that each one might be modeled correctly before attempting to combine them in the composition of the overall performance.

Fig. 1 2009 national championship

The main concern of this paper is to measure the textual coherence of a poem automatically generated under meter and rhyme constraints.

2 Related Work

Computer-based poetry has received attention in the research community in the last years, and several interesting systems have been developed. In our opinion, automatic generation of poems resembles the creation of a *bertso* in these three aspects:

1. They have to satisfy very specific technical requirements: on the one hand metric restrictions, that is, the number of syllables per line; and, on the other hand, they have to meet certain rhyme pattern.
2. They are allowed a certain poetic license which implies, sometimes, deviations from the rules of syntax and semantics.
3. The result must be meaningful for the user, more specifically, the resulting text must arouse specific emotions amongst the audience.

Taking aforementioned similarities in mind, we reviewed the existing literature on the automatic creation of poetry. A good overview can be found in [Oliveira 2009; Gervas 2013]. Most recent -and relevant- ones include:

- **Haiku Generation Using Vector Space Model.** Wong and Chun [Wong et al. 2008] presented an approach to generate “modern haikus” using text collected from blogs. The proposed approach uses a keyword and a line repository containing sentence fragments found in blogs. The haiku generation process starts by choosing three keywords from the lexicon, and then, sentence fragments in the blogosphere containing those keywords are searched. Two keywords are extracted from each sentence, using a *tf-idf* weighting scheme to evaluate how important each word is to a sentence. Afterwards, vectors are used to compare each sentence pair. The cosine of the angle between two vectors is used to measure their semantic relation. Finally, the most semantically related pairs of vectors are chosen for the resulting haiku.
- **Full-FACE Poetry Generation.** This system [Colton et al. 2012] is a corpus-based poetry generation system, aimed to be a fully automatic computer poet. The generation is driven by a four stage process that involves: simile retrieval from a simile corpus, generation of variations for each simile, combination -- they combine similes with their variations and key phrases extracted from newspaper articles-- and instantiation, which chooses one of the possible combinations randomly.
- **Computational Modelling of Poetry Generation**.In this paper [Gervás 2013] Gervás shows a review of different works done in this area and presents a redesigned version of the well known WASP poetry generator which combines different AI techniques. The poems generated by this system have been accepted for publication in a book about the possibilities of computers writing love poems.

To the best of our knowledge, there are no poetry generation systems implemented in a real robot. But *bertsolaritza* belongs to the oral genre, and the public performance is extremely important. Therefore, it is not enough the development of an automatic verse generation system, the created poem has to be part of a performance. Thus, a real body that interacts with the public and sings the improvised verse with a proper melody is needed.

The design and implementation of such an autonomous interactive robot is a challenging task. There is much to learn from the potential of the integration of these technologies, as there is from the embodied and speech-based human robot interaction [Beck et al. 2010; Scheutz et al. 2011].

3 System’s Task

Bertsos can be composed in a variety of settings and manners. At a formal competition the verse-makers are called upon to compose and sing different kinds of bertsos by the theme-prompter or emcee. For our particular challenge we have chosen the popular exercise of “Rhymes Given”: the robot is given the four rhyming words and it is required to compose the bertso in Zortziko Txikia meter “around” these rhyming words.

Zortziko Txikia (see Fig. 2) is a composition of eight lines in which odd lines have seven syllables and even ones have six. The union of each odd line with the next even line, form a strophe. Each strophe has 13 syllables with a caesura after the 7th syllable (7 + 6) and must rhyme with the others.

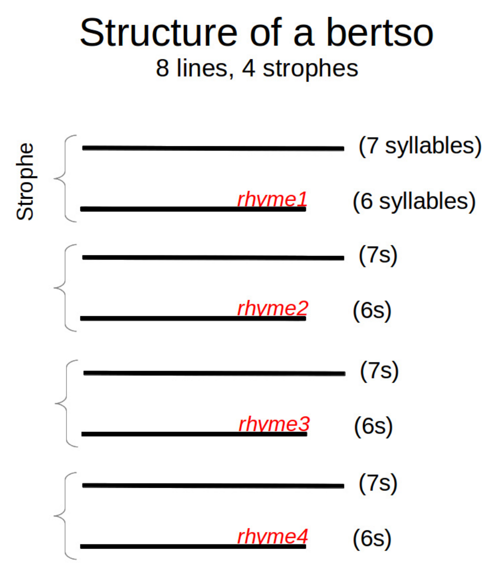


Fig. 2 Structure of the Zortziko Txikia

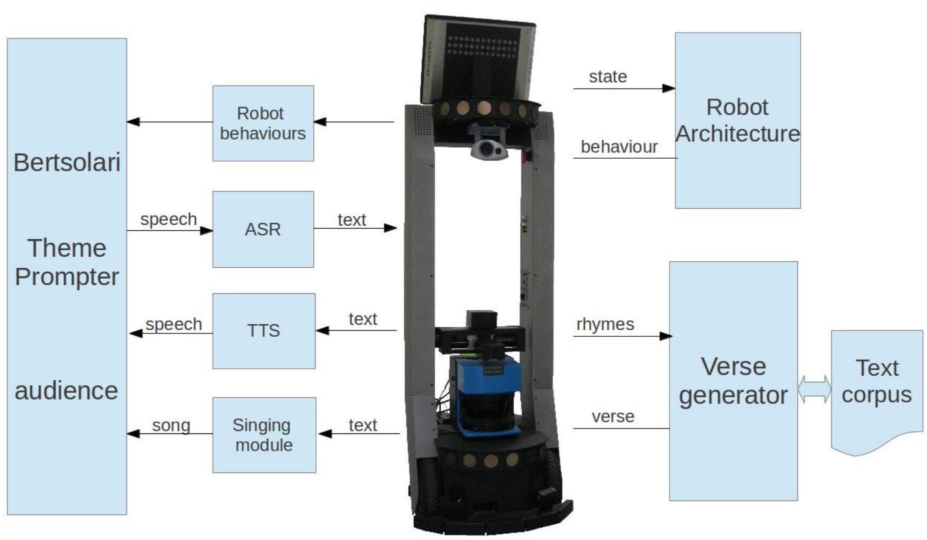


Fig. 3 Overall software architecture

It must be said that human-robot interaction is entirely speech-based: the theme-prompter (or emcee) proposes the four rhymes to compose the verse in Basque, and the system answers in Basque too. The overall performance is represented as natural as possible by the robot, interacting as much as possible with the emcee and the audience (see [Astigarraga et al. 2013] for more details).

4 Verse Generator

As it can be deduced from Bertsobot's software architecture (see Fig. 3), several research areas are involved in the project. This paper concentrates on the verse generation subpart of the project, which generates a poem text under meter and rhyme constraints and using two methods to achieve textual coherence.

In the basic scenario, the four rhymes to compose a *bertso* are received as input, and the verse generator module then should give as output a novel and technically correct verse, and (hopefully) with coherent content.

The poetry generator is based on the following principles:

- **Form**: rhyme and metric compound the technical requirements of a *bertso*. Thus, finding rhymes and counting syllables are essential abilities that the system must perform.
- **Content**: the output of the verse generator module must be meaningful. Methods to measure the semantic coherence of the generated text are needed.

It can be said that the final verse is a marriage of form and content in which each works together to convey a meaningful text with formal precision.

The verse generation process then consists of the following steps:

1. Receive as input the four rhymes to compose the verse.
2. Find sentences in the corpus that rhyme with the input words.
3. Generate the verse with the highest textual coherence.

We next describe in some more detail the background corpus construction process as well as the similarity methods used to measure textual coherence.

4.1 Text Corpus

Our approach for verse-generation is corpus based. We have chosen initially to work with documents mined from the Basque newspaper *Berria* [WWW_1] (%85) alongside verses extracted from the work of well-known *bertsolari*-s [WWW_2] (%15). The later reflects the desire to maintain the language-model of the *bertsolaritza*, and the former tends to increase quality while not appropriating text intended for poems. Next, we divided the text collection into passages with coherent meaning. For our purposes, we selected as document the minimum semantically coherent block of text, where each document deals with a single idea. Therefore, we created a corpus of individual sentences. The sentence selection and extraction process consists of the following two steps:

- **Splitting**: split the text collection into sentences.
- **Selection**: extract the sentences that fulfill the meter requirements of the *Zortziko Txikia* (13 syllables).

Once preprocessing step is finished, we have a corpus of 56274 sentences, with 20098 unique terms in it. This sentence repository will be used in the *bertso* generation process.

4.2 Semantic Relationship/Similarity Measures

The four rhymes received as input data are used to find sentences in the corpus ending with those rhymes. During the search process, a set of sentences for each initial rhyme are found. Those sentences will then be fed into the similarity measuring stage to perform the sentence matching process.

A system capable of validating its output from a metrical and rhythmical point of view shows a hint of aesthetic sensibility, but it is missing out one important aspect that humans use to value poetry: meaningfulness. The main purpose of this stage is to compare the relationship between pairs of sentences to choose the most semantically related ones for the final verse. To do that, the sentence relation system computes, for each pair of sentences, a score that evaluates how those sentence are semantically related. That is, once the technical requirements are fulfilled, sentences that maximize the internal coherence of the generated verse are selected.

The starting point for our similarity methods is the observation that related strophes in a *bertso* tend to use the same or similar words. It is true that the use of the same or similar terms does not always guarantee relatedness, but can be considered as a precondition, and we believe it should work well as a measure of relatedness. We have tested the two similarity measures described below.

Method 1: Standard Vector Space Model (VSM)

The Vector Space Model (VSM), introduced first by Salton [Salton 1971], is a model for representing text in a vector space. Each dimension of a vector corresponds to a term that is present in the text collection. This is the simplest vector-based similarity approach, and similarity calculation is based on word matching. Before any semantic processing takes place, we applied the following text preprocessing tasks:

- **Tokenization**: sentences are split into individual words. Punctuation marks are also removed from text.
- **Lemmatization**: due to its agglutinative characteristic, a given word lemma makes many different word forms in Basque language. Therefore, a lemmatization step is accomplished to reduce term quantity.

In order to represent the importance of a term within a document and within the whole document collection, the well known *tf-idf* (term frequency-inverse term frecuency) [Ramos 2003] weighting scheme has been applied, which combines local and global weighting. The *tf-idf* weighting scheme assigns to term *t* a score in sentence *s* given by

$$tf - idf_{t,s} = tf_{t,s} \cdot idf_{t} \quad (1)$$

Finally, to measure the semantic relation between a pair of sentences, we make use of the classical cosine similarity of their vector representations:

$$cos\theta = (V_1 \cdot V_2)/(||V_1|| \cdot ||V_2||) \quad (2)$$

where the numerator represents the dot product (or inner product) of the vectors, and the denominator is the product of their Euclidean lengths.

Shortcomings of this standard vector-based method are known: on the one hand, only sentences that overlap in vocabulary will be considered similar. Two documents with a similar topic but different vocabulary will not be considered similar. In other words, it cannot deal with synonymy and polysemy. And on the other hand, it assumes complete independence among the terms.

Method 2: Latent Semantic Analysis (LSA)

LSA was developed as a special vector space approach to solve the aforementioned issues [Deerwester et al. 1990]. In the standard VSM, term associations are ignored, and therefore, synonymy and polysemy are not captured. By contrast, LSA tries to find the latent semantic structure of a document collection. That is, to find hidden relations between terms, sentences or other text units [Landauer and Dumais 1997]. In other words, LSA analyzes term co-occurrence of higher orders, so that it is able to incorporate the relationship of words A, B which only co-occur in documents through word C, and never appear in the same document together directly.

LSA involves the application of the matrix algebra method Singular Value Decomposition (SVD) to the document-by-term matrix in order to reduce its rank and construct a semantic space. The matrix is then decomposed in such a way that similar vectors will be conflated, or moved closer within the space. And thus, LSA enables the calculation of semantic similarities between words, sentences or paragraphs.

To construct a semantic space to capture textual coherence, a large representative text corpus is needed. Thus, a large collection of documents collected from the newspaper *Berria* was used for the construction of the semantic space. The text collection was divided into paragraphs, that lead to a corpus of 161.113 documents and 10.121.624 words. Again, the text collection was first lemmatized in order to reduce term quantity with no loss in meaning. Thus, a corpus of 180.616 unique words was obtained. Then, the entries of the document-by-term matrix were weighted using the already presented *tf-idf* scheme. Sentence similarity in the reduced space is measured by the cosine distance between their vector representations.

We used Gensim [Řehůřek and Petr 2010] Python framework to compute the LSA and perform similarity queries. After applying the SVD, the semantic space dimension was reduced to 250, a value chosen empirically. We thus map each term/document to a 250 dimensional space. The consequence of this dimension reduction is that some dimensions are combined on more tha one term.

Alg. 1 summarizes the verse-generation process. The algorithm receives as input the four rhymes to compose the verse, extracts rhyming sentences from the corpus and generates all possible verse-combinations. For each generated verse a cosine similarity of adjacent sentences is calculated and the verse with the highest score is returned for each of the above mentioned methods.

It must be noted that for each *s1-s2-s3-s4* sentence combination, *s1-s2, s2-s3* and *s3-s4* similarity comparisons are made. That is, the overall semantic measure is calculated for each *bertso* by adding up the cosines between the vectors for all pair of adjoining sentences. This is so because we seek a progressive relationship between sentences. And for the overall coherence of the *bertso*, it is more important the relationship between *s1* and *s2* (adjacent strophes), than the relationship between *s1* and *s4* (non-contiguous strophes). Finally, the system returns the verse with the highest cosine measure.

It must be noted that for each *bertso* three comparisons are made. For example, if for each (r1-r4) rhyme group we have **s**=10 sentences in the corpus, then the cosine similarity between two vectors will be calculated *10x10x10x10x3 =30000* times. As **s** grows, computing the cosine similarities can become computationally expensive.

```
Input: r1 , r2 , r3 and r4 rhymes
Output: the verse with the highest cosine measure

group1 = sentences that have r1 as rhyme;
group2 = sentences that have r2 as rhyme;
group3 = sentences that have r3 as rhyme;
group4 = sentences that have r4 as rhyme;

for all s1 in group1 do
     for all s2 in group2 do
            for all s3 in group3 do
                   for all s4 in group4 do
                          verse = (s1, s2, s3, s4 );
                          cos_12 = cosine_similarity(s1, s2);
                          cos_23 = cosine_similarity(s2, s3);
                          cos_34 = cosine_similarity(s3, s4);
                          cos_all = cos_12 + cos_23 + cos_34
                   ;
                          store(file, verse, cos_all);
                   end for
            end for
     end for
end for
return find_max_cosine(file)
```

Alg. 1 General algorithm to generate verses

It must be noted that for each *s1-s2-s3-s4* sentence combination, *s1-s2, s2-s3* and *s3-s4* similarity comparisons are made. That is, the overall semantic measure is calculated for each *bertso* by adding up the cosines between the vectors for all pair of adjoining sentences. This is so because we seek a progressive relationship between sentences. And for the overall coherence of the *bertso*, it is more important the relationship between *s1* and *s2* (adjacent strophes), than the relationship between *s1* and *s4* (non-contiguous strophes). Finally, the system returns the verse with the highest cosine measure.

It must be noted that for each *bertso* three comparisons are made. For example, if for each (r1-r4) rhyme group we have **s**=10 sentences in the corpus, then the cosine similarity between two vectors will be calculated *10x10x10x10x3 =30000* times. As **s** grows, computing the cosine similarities can become computationally expensive.

5 Evaluation

The evaluation of computer-generated poetry is a difficult task. Taking into account that our goal is to build a robotic system that performs verses in public, like real *bertsolari*-s do, we believed that empirical validation by human subjects was the best way of evaluation. Thereby, we contacted with the *Bertsozale Elkarte* [WWW_3] (Association of the Friends of *Bertsolaritza*) and 5 *bertso*-judges participated in the evaluation.

We selected the following 4 different rhyming groups (rhymes used in a verse-campionship), where each group was composed by four rhyming words to generate the bertso:

Hor-inor-zor-gogor	There-nobody-debt-hard
gera-batera-atera-bera	we're-together-go out-he
dakar-zakar-azkar-alkar	bring-rough-fast-together
dira-tira-erdira-begira	they're-come on-center-look

For each group, 6 computer-generated verses were selected:

- 2 generated with no similarity measure. Sentences were selected randomly, without applying any cohesion measure between them
- 2 generated following VSM and cosine similarity approach
- 2 generated following LSA approach.

We mixed all the verses and a a blind experiment was carried out. Verse-judges were asked to evaluate them based on the following scoring:

0→ The verse has no meaning. There is no cohesion between the strophes.
1→ Minimal cohesion.
2→ Average cohesion (some strophes maintain meaninful relation).
3→ Good cohesion level.
4→ Really well related strophes. Very well overall cohesion.

Table 1 shows how many verses have achieved 0 scoring, how many 1 and so on, for each method used.

Table 1 Score frequencies and percentages

Scoring	Random	VSM	LSA
0	14 (35%)	7 (17.5%)	4 (10%)
1	12 (30%)	8 (20%)	8 (20%)
2	12 (30%)	13 (32.5%)	14 (35%)
3	2 (5%)	8 (20%)	6 (15%)
4	0 (0%)	4 (10%)	8 (20%)

Table 2 shows the average scoring obtained by each method.

Table 2 Average scores

	Rhyming gr. 1		**Rhyming gr. 2**		**Rhyming gr. 3**		**Rhyming gr.4**	
	Verse1	Verse2	Verse1	Verse2	Verse1	Verse2	Verse1	Verse2
Random	1,2	1	0,4	1,6	1	1	1,2	1
VSM	2,2	2,8	2,2	2,2	0,2	1,8	1,2	2,2
LSA	2	3	1,2	1	3	3,2	2	1,8

Looking to the above tables, it is notorious that there is a clear distinction between the verses generated randomly, and the verses generated applying any of the similarity measures presented. Whereas lower scores are the most frequent scores given to the randomly generated verses, highest scores are reserved to VSM and LSA verses. It is also worth noting that although VSM shows the best mean values, LSA gets highest scores.

For illustration purposes, we show a *bertso* generated by our system using LSA method. We also give an English translation, even though part of its aesthetic value is lost in translation.

Gu euskaldunak berez	*We, Basque people*
ez gerala inor	*are nothing at all*
guk inori ez diou	*we do not owe our live*
gure bizitza zor	*to anyone*
berak gogor badaude	*if they are hard*
gu ere hain gogor	*we also*
eta euren jarrera	*and that will show*
igarriko da hor	*their stance*

The objective of the evaluation was none other than finding the most suitable method for measuring the internal consistency of the verses. Once the method is implemented in the real robot, the improvised verses will be judged by the public with applause or whistles, as in the events of improvisers.

6 Integration of the Components into a Real Robot

So far we have described the tools used for creating verses and the methods to predict their textual coherence. But *bertsolaritza* belongs to the oral genre, and thus, needs to be performed in public. Therefore, the *Bertsobot* system needs additional functionalities, such as a robotic body to interact with the environment and display emotional body-language, and a synthesized voice.

In this last step we have integrated the automatic verse-making system in a pair of real autonomous robots.

- ***Galtxagorri*** is a Pioneer 3-DX robot
- ***Tartalo*** is a PeopleBot robot built on the P3-DX base.

Fig. 4 shows a picture of the robots in a public performance.

Fig. 4 Verse-duel between robots and real *bertsolari*-s

The *Bertsobot* system can be run in a simple computer, but as mentioned before, *bertsolaritza* implies interaction with the environment, such as the other *bertsolaris* of the performance and the audience. In improvised oral festivals, the *bertsolari*-s wait sitting their turn to sing, and when their turn comes, (s)he approaches the microphone that is placed in front. Once in their location, they have to look to the theme-prompter, that will prescribe a topic which serves as a prompt for the *bertso* (in our case the four rhymes to compound the *bertso*). After that, they have a specific time to create the verse and finally sing it. Once finished, they await a few seconds to perceive the audience's reaction and they go back to their chair.

The aim of the robotic behaviors is to represent as faithfully as possible *bertsolari*'s movements on stage, interacting as much as possible with the other improvisers and the audience.

Videos of the public demonstrations carried out with real improvisers can be found on our website [WWW_4].

7 Conclusions and Future Work

We have presented an embodied system, Bertsobot, capable of improvising and performing Basque traditional verses. The main contribution of this paper consists on the verse-maker module, which composes verses automatically taking into account the semantic relation of strophes within a *bertso*.

We are already working on further enhancements to each of the processes involved, including:

1. Implementing improved methods to generate phrases for templates and working with other corpora.
2. Exploring new models to measure the semantics relationship between sentences.
3. Letting the robots display facial expressions by means of a 3D avatar.
4. Enabling the robot to adequate the message of the poem based on the reactions perceived from the public.
5. Improving the emotional body language of the robots and increasing their degree of autonomy while acting the performance. For the time being, we are trying to identify the microphone using a Kinect, so the robot will know where it is.
6. Identify the faces of the participants (*bertsolari*-s and the theme-prompter). Thus, the robot will turn exactly towards its competitor, or mentioning its name in the improvised verse, as it occurs sometimes.

Acknowledgments. This work has been partially conducted under the SAIOTEK projects XIMUR and POETAUTO, the Basque Government Research Team grant and the University of the Basque Country UPV/EHU, under grant UFI11/45 (BAILab).

References

[Alegria et al. 1996] Alegria, I., Artola, X., Sarasola, K., Urkia, M.: Automatic morphological analysis of Basque. Literary and Linguistic Computing 11(4), 193–203 (1996)

[Astigarraga et al. 2013] Astigarraga, A., Agirrezabal, M., Lazkano, E., Jauregi, E., Sierra, B.: Bertsobot: the first minstrel robot. In: Proc. 6th Inter. Conf. on Human System Interaction, pp. 129–136 (2013)

[Beck et al. 2010] Beck, A., Hiolle, A., Mazel, A., Cañanamero, L.: Interpretation of emotional body language displayed by robots. In: Proc. 3rd International Workshop on Affective Interaction in Natural Environments, pp. 37–42 (2010)

[Colton et al. 2012] Colton, S., Goodwin, J., Veale, T.: Full face poetry generation. In: Proc. 3rd Inter. Conf. on Computational Creativity, pp. 95–102 (2012)

[Deerwester et al. 1990] Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. of the American Society for Information Science 41(6), 391–407 (1990)

[Gervas 2013] Gervás, P.: Computational modelling of poetry generation. AISB Convention 2013. University of Exeter (2013)

[WWW_1] `http://www.berria.info/` (accessed January 5, 2014)

[WWW_2] `http://bdb.bertsozale.com` (accessed January 5, 2014)

[WWW_3] `http://www.bertsozale.com/en` (accessed January 5, 2014)

[WWW_4] `http://www.sc.ehu.es/ccwrobot/bideoak`

[Landauer and Dumais 1997] Landauer, T.K., Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review 104(2), 211 (1997)

[Landauer et al. 2013] Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W.: Handbook of latent semantic analysis. Psychology Press (2013)

[Oliveira 2009] Oliveira, H.: Automatic generation of poetry: an overview. Universidade de Coimbra (2009)

[Ramos 2003] Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proc. 1st Instructional Conf. on Machine Learning (2003)

[Řehůřek and Petr 2010] Řehůřek, R., Petr, S.: Software framework for topic modelling with large corpora. In: Proc. of LREC 2010 Workshop New Challenges for NLP Frameworks, pp. 45–50 (2010)

[Salton 1971] Salton, G.: The SMART retrieval system: Experiments in automatic document processing. Prentice-Hall (1971)

[Scheutz et al. 2011] Scheutz, M., Cantrell, R., Schermerhorn, P.: Toward humanlike task-based dialogue processing for human robot interaction. AI Magazine 32(4), 77–84 (2011)

[Wong et al. 2008] Wong, M.T., Chun, Li, Q., Chen, S.Y., Xu, A.: Automatic haiku generation using vsm. In: WSEAS International Conference. Mathematics and Computers in Science and Engineering, no. 7. World Scientific and Engineering Academy and Society (2008)

Markov Text Generator for Basque Poetry

Aitzol Astigarraga, José María Martínez-Otzeta, Igor Rodriguez, Basilio Sierra, and Elena Lazkano

Department of Computer Sciences and Artificial Intelligence, University of the Basque Country UPV/EHU, Donostia-San Sebastian 20018, Spain
www.sc.ehu.es/ccwrobot
{aitzol.astigarraga,josemaria.martinezo,igor.rodriguez,b.sierra,e.lazkano}@ehu.eus

Abstract. Poetry generation is a challenging field in the area of natural language processing. A poem is a text structured according to predefined formal rules and whose parts are semantically related. In this work we present a novel automated system to generate poetry in Basque language conditioned by non-local constraints. From a given corpus two Markov chains representing forward and backward 2-grams are built. From these Markov chains and a semantic model, a system able to generate poems conforming a given metric and following semantic cues has been designed. The user is prompted to input a theme for the poem and also a seed word to start the generating process. The system produces several poems in less than a minute, enough for using it in live events.

Keywords: Poetry generation, Basque language, N-grams

1 Introduction

Poetry is one of the most expressive -and challenging- ways to use language. It is commonly accepted that quality of good poetry arises from an equilibrium between content and form, both content and form contributing to its aesthetic value. But to what extent each one affects the overall result is still a matter of debate. Oral poetry is considered poetry constructed without the aid of writing [17]. Oral meant that a work was composed and performed at the moment, with no prior preparation. Poets and story-tellers from many different cultures have historically used such oral improvisation in performances. Nowadays many improvisational oral practices exists around the world, such as Serbo-Croatian guslars, freestyle rap and Basque bertsolaritza.

In this work we present a novel system that is able to generate Basque poetry under the constraints of *bertsolaritza*, a form of oral and improvised poetry. Those constraints are local (the metric of the verses and the rhyming pattern) and non-local (the semantic similarity with a given theme and the time allotted to produce the finished poem). The mechanism for poetry generation relies in N-gram models [18] created from a corpus of Basque poetry previously collected. Semantic similarity between lines is measured.

An experiment where poems have been generated with a given theme and metric has been performed, and the result has been subject to evaluation by four individuals familiar with Basque poetry. We present the summary of their evaluations, along with some poems.

The rest of the paper is organized as follows: in section 2 a brief account of Basque poetry is presented. In section 3 related work is surveyed, while section 4 is devoted to present our approach. Experimental setup and results are shown in section 5, and finally the conclusions are summarized in section 6.

2 Some words about Basque language and *bertsolaritza*

Basque is the language of the inhabitants of the Basque Country. And *bertsolaritza*, an improvised contest poetry, is one of the manifestations of traditional Basque culture. Events and competitions in which improvised verses, *bertso-s*, are composed are very common. In such performances, one or more verse-makers, named *bertsolari-s*, produce impromptu compositions about topics or prompts which are given to them by an emcee (theme-prompter). Then, the verse-maker takes a few seconds, usually less than a minute, to compose and sing a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme.

Xabier Amuriza, a famous verse-maker who modernized the *bertsolari* movement, defined *bertsolaritza* in a verse as:

Neurriz eta errimaz	*Through meter and rhyme*
kantatzea hitza	*to sing the word*
horra hor zer kirol mota	*that is what kind of sport*
den bertsolaritza.	*bertsolaritza is.*

When constructing an improvised verse a number of formal requirements must be taken into account. Rhyme and meter are inseparable elements in improvised verse singing (in the above example, odd lines, which must rhyme with each other, have seven syllables and even lines six). A person able to construct and sing a *bertso* with the chosen meter and rhyme is considered as having the minimum skills required to be a *bertsolari*. But the true quality of the *bertso* does not only rely on those demanding technical requirements. The real value of the *bertso* resides on its dialectical, rhetorical and poetical value [9]. Thus, a *bertsolari* must be able to express a variety of ideas and thoughts in an original way while dealing with the mentioned technical constraints. In this balance lies the magic of a *bertso*.

3 Related Work

Computational modeling for poetry generation has become a topic in the artificial intelligence community in the last years. People with a background closer to humanities made early efforts in systematic generation of poetry. We could mention works related to generating variations over a predetermined set of verses [21], or to select a template to produce poems from it [20].

According to [12], two main strategies can be outlined in the field of computer generation of poetry:

– **Corpus-based approach**: computer is used to harvest and reuse text already formated into poem-like structure of lines. This approach can be formulated as an

information retrieval task, where the objective is to extract and select existing lines to compose new poems. The reuse and ordering of written text was introduced by Queneau [21]. Many computer-based systems rely nowadays in this method. Most relevant are: [19], [22], [23]. This procedure is adopted in [2] and [3] where two methods to ensure internal coherence of poems were presented.
- **Composition from scratch**: alternative methods rely on building a text from scratch, character by character or word by word, and establishing a distribution of the resulting text into poem lines by some additional procedure. A popular -and rather simple- method to generate text is the N-gram model, which is the simplest Markov model. N-grams assign probabilities to sequences of words and the generated model can be used to stochastically generate sequences of words based on the generated distributions [14] [18]. An N-gram probability is the conditional probability of a word given the previous N-1 words. Markov chains have been widely used as the basis of poetry generation systems as they provide a clear ans simple way to model some syntactic and semantic characteristics of language [16]. Popular and recent examples of N-gram poem generators are [4], [6] and [12]. But text poetry hold non-local properties such as rhyme and metric that cannot be modeled by an ordinary N-gram model. Therefore, the above mentioned methods need additional procedures for distributing the resulting text into poem lines with metrical and rhyming constraints.

For a more thorough review of systems related to automatic generation of poetry, we point the reader to [10] and [15].

4 Proposed Approach

The formal constraints of *bertsolaritza* (and poetry as well) can be viewed as intra-line constraints and inter-line constraints. Intra-line constraints include the number of syllables allowed on each line, that is, the metrical scheme. Inter-line constraints consist of rules that state rhyming constraints between the lines and the semantical relatedness of the text respect to the proposed theme. The *bertsolari* has a basic strategy that is used in a systematic way [8]: think up the end first. On hearing a proposed theme, the *bertsolari* turns on their mental machinery and starts thinking about what is going to say and the order in which (s)he is going to say it, keeping the main idea for the end.

Our approach implements the same strategy used by *bertsolaris* for the creation of impromptu verses, and in a few seconds - less than minute - assembles a new poem along the prescribed verse-form. Although our work focuses on *bertsolaritza*, it can be generalized as the automatic poem generation.

The main characteristics of our word-based N-gram approach are the following:

- Live generation of verses.
- Poem generation in the style of an existing author.
- Satisfy structural constraints, both, inter-line metrical constraints and between-line rhyming ones.
- Semantic similarity respect to the theme given to construct a verse and between the lines that compose the verse (internal coherence).

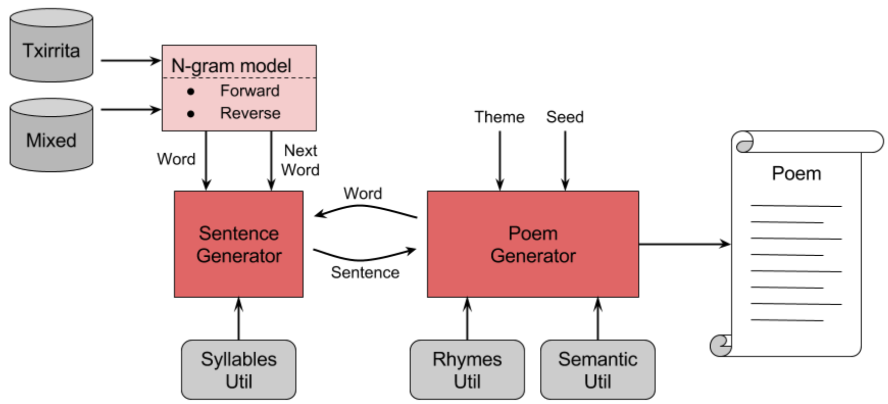


Fig. 1. System architecture.

4.1 Training the language model

As the text generation module relies on an N-gram model of language to produce sequences of text that are word to word coherent, the overall style of the resulting poems is strongly determined by the sources used to train the content generators. Therefore, different sources of text were used in the experimental setup in order to experiment with them and extract conclusions.

- Mixed: 18913 lines and 94314 words, of which them 21411 are unique. This corpus is a compilation of sentences mined from Basque newspaper Egunkaria [1](85%) alongside poetry sung in *bertsolari* contests by different performers (15%).
- Txirrita: 2127 lines and 24277 words, of which them 6998 are unique. This corpus is a compilation of poetry by a famous *bertsolari* called *Txirrita* [2].

4.2 Text applications

Several linguistic tools have been developed and used in the verse-maker project.

- **Rhyme search**: finding words that rhyme with a given word is an essential task that the verse-generation system must perform. Basque rhyme schemes are mainly consonant. The widely consulted rhyming dictionary *Hiztegi Errimatua* [1] contains a number of documented phonological alternations that are acceptable as off-rhymes. These alternations have been implemented using regular expressions.
- **Syllable counter**: counts the number of syllables present in the given text. For the syllabification itself, the approach describing the principal elements of Basque language structure [13] has been implemented.

[1] https://en.wikipedia.org/wiki/Egunkaria
[2] https://en.wikipedia.org/wiki/Txirrita

– **Similarity measure**: the main purpose of this module is to measure the semantic relationship between pairs of words and sentences. The module computes for each pair of words/sentences a score that evaluates how similar they are. Latent Semantic Analysis (LSA) method [7] [24] has been implemented to capture the semantic relatedness. This semantic model has been generated with news mined from the Egunkaria newspaper.

The aforementioned modules have been integrated into a verse-maker system for automatic poem generation, as shown in Figure 1.

5 Experimental setup and results

In the basic scenario, a topic is given by the user and the proposed method then aims to give as output a novel poem following *bertsolaris*' strategy for poem creation (think up the last line, find rhyming words and generate the poem line by line). The output satisfies the formal constraints and also shows coherent content related to the given topic.

The procedure to create a poem involves prompting the user for a *theme*, a *first word* to start the last line, a *metric* and a *corpus* from which extract the 2-grams that will make the poem. The poem will consist of four lines, each 13 syllables long, and all of them sharing a common rhyme. This is the metric of *Zortziko-txikia*, commonly used in *bertsolaritza* contests.

With these parameters the system follows these steps:

– The last line of the poem is generated using the forward-looking chain created for *corpus*, taking as starting point the *first word* given by the user and then chaining 2-grams until the *metric* is exactly fulfilled. A number of candidates (60) are generated in this way.
– The candidate lines are then ordered according to their semantic similarity with respect to the *theme*.
– Starting from the most similar candidate, the system tries to complete a poem. The feasibility of this task is dependent on the number of possible rhyming words existing in the corpus as well as in the existence of a chain of 2-grams finishing in that rhyme with the given *metric*.
– To create the remaining lines, the possible rhyming words are ordered according to their semantic similarity with respect to the *theme*.
– The N-gram generation system generates poem lines in the following manner: at each step the sequence of words are extended with new words that have a non-zero probability of appearing after the last word according to the N-gram model. At each step, word choice is made randomly.
– The first five poems built in this way are returned by the system.

By means of this procedure, the poems with the last line and rhymes more similar to the theme are returned, given the constraints faced by the system. In less than a minute the created poems are obtained, making it possible to use this approach in a live event of improvised poetry.

Table 1. 2-gram frequency table. The number of 2-grams appearing with a frequency of 1, 2, 3 and 4 is shown.

Corpus	**Txirrita**	**Mixed**
#2-grams	21665	85720
Unique	19500	81321
2-times	1320	3144
3-times	402	603
4-times	176	242

In order to test the capabilities of the above described approach, an experiment with two different corpora has been carried out. We have tested 2-gram and 3-gram models and the preliminary results have shown that the 3-gram model tended to replicate the corpus almost verbatim. Therefore we have used 2-grams, because we wanted our system to produce tentative solutions somewhat different to the original lines, and this is more likely with low order N-grams. Table 1 shows the number of total 2-grams for the two corpora along with the number of them appearing 1, 2, 3 and 4 times is shown. As the corpora are not big, no minimum threshold has been imposed over the frequency of 2-gram terms.

Table 2. Two poems created by the system. On the left with the theme 'man' and from the Txirrita corpus, and on the right with the theme 'war' and from Mixed corpus.

Basque	Nun dezu bada nere baserritar ona zabaldu eta orain zeruan dagona biartzen bada kotxe bakoitzak komona bertso birekin egin nai zuen gizona	Basque	Orain globo bat jarri dezagun betiko nik ahal izan baino lehen itzuliko gizarteak bizkarra eman eta kito eta biolentzia ez da eroriko
English	Where you have so my good farmer that has spread and is now in the sky if each car is forced a service wanted to create a man with two verses	English	Let's have a balloon forever I will be back as soon as possible society turns its back and ready and violence will not decay

In Table 2 we show two poems composed by the automated system, in its original form and in an approximate English translation.

Objective evaluation of poetry is difficult, if not impossible, to assess in an automatic way, because as the saying goes, beauty is in the eye of the beholder. As Gervás [11] and Cardoso [5] stated, human evaluators are needed to assess the degree of creativity of a computational creation. We have presented the generated poems to four people familiar with *bertsolaritza*, explicitly telling them that such poems were the product of an automated system. Each of them analyzed twenty poems, ten from each corpus. They have been asked to give their overall impression about the quality, similarity with the theme, internal coherence and style. Their impression has been positive, stating that they were well-formed poems, although not of human-produced quality. They found a general sense of semantic relationship with the given theme, even diluted, but they also found that the internal coherence of the whole poem was pretty poor. It was also stated that the poems created from the Txirrita corpus sounded more natural and closer to the style of the *bertsolaritza* than the poems from the Mixed corpus. This corroborates, as some authors suggest, that the N-gram model imposes a certain overall style on the texts that can be produced.

6 Conclusions and Further Work

In this paper we have presented an automated system to generate poetry in Basque language. As a preliminary step, two Markov chains are generated from a given corpus, with their nodes representing the words and the directed edges the existence of a 2-gram with the words in the given order. The proposed approach uses one forward and one backward Markov chain because we construct some lines from the start and other from the end. From a theme, a first word and a metric, poems are generated.

The result of a experiment with two different corpora has been shown, with the system been able to generate five poems in less than a minute, fast enough to be used in a live event. The evaluation of the poems has been made by several people familiar to *bertsolaritza* and their opinions have been reflected. The overall impression has been positive, although the quality is still far from a human composer.

As further work, we will try to speed up the generating process, in order to be able to explore more candidate lines and build higher quality poems. Other bigger or more diverse corpus will be used to generate the Markov chains. The semantic model could be learned from a corpus with a higher percentage of fiction documents, that could convey semantic relationships closer to those in poetry.

7 Acknowledgments

This paper has been supported by the Spanish Ministerio de Economía y Competitividad, contract TIN2015-64395-R (MINECO/FEDER, UE), as well as by the Basque Government, contract IT900-16. The authors gratefully acknowledge Bertsozale Elkartea [3] (Association of the Friends of Bertsolaritza), whose verse corpora has been used to test and develop the proposed method.

References

1. Amuriza, X.: Hiztegi Errimatua. Lanku (1981)
2. Astigarraga, A., Jauregi, E., Lazkano, E., Agirrezabal, M.: Textual coherence in a verse-maker robot. In: Human-Computer Systems Interaction: Backgrounds and Applications 3, pp. 275–287. Springer (2014)
3. Astigarraga, A., Agirrezabal, M., Lazkano, E., Jauregi, E., Sierra, B.: Bertsobot: the first minstrel robot. In: Human System Interaction (HSI), 2013 The 6th International Conference on. pp. 129–136. IEEE (2013)
4. Barbieri, G., Pachet, F., Roy, P., Esposti, M.D.: Markov constraints for generating lyrics with style. In: Proceedings of the 20th European Conference on Artificial Intelligence. pp. 115–120. IOS Press (2012)

[3] http://www.bertsozale.com/en

5. Cardoso, A., Veale, T., Wiggins, G.A.: Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. AI Magazine 30(3), 15 (2009)
6. Das, A., Gambäck, B.: Poetic machine: Computational creativity for automatic poetry generation in bengali. In: 5th International Conference on Computational Creativity, ICCC (2014)
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science 41(6), 391 (1990)
8. Egaña, A.: Oral tradition 22 (2007)
9. Garzia, J., Sarasua, J., Egaña, A.: The art of bertsolaritza: improvised Basque verse singing. Bertsolari liburuak (2001)
10. Gervás, P.: Computational modelling of poetry generation. In: Artificial Intelligence and Poetry Symposium, AISB Convention 2013. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, University of Exeter, United Kingdom (2013)
11. Gervás, P.: Deconstructing computer poets: Making selected processes available as services. Computational Intelligence (2015)
12. Gervás, P.: Constrained creation of poetic forms during theme-driven exploration of a domain defined by an n-gram model. Connection Science 28(2), 111–130 (2016)
13. Jauregi, O.: Euskal testuetako silaba egituren maiztasuna diakronikoki. Anuario del Seminario de Filología Vasca" Julio de Urquijo" 37(1), 393–410 (2013)
14. Jurafsky, D., James, H.: Speech and language processing an introduction to natural language processing, computational linguistics, and speech (2000)
15. Lamb, C., Brown, D.G., Clarke, C.L.: A taxonomy of generative poetry techniques. In: Bridges Finland Conference Proceedings. pp. 195–202 (2016)
16. Langkilde, I., Knight, K.: Generation that exploits corpus-based statistical knowledge. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1. pp. 704–710. Association for Computational Linguistics (1998)
17. Lord, A.B., Mitchell, S.A., Nagy, G.: The singer of tales, vol. 24. Harvard University Press (2000)
18. Manning, C.D., Schütze, H., et al.: Foundations of statistical natural language processing, vol. 999. MIT Press (1999)
19. Oliveira, H.G., Cardoso, A.: Poetry generation with poetryme. In: Computational Creativity Research: Towards Creative Machines, pp. 243–266. Springer (2015)
20. Oulipo, A.: Atlas de litterature potentielle. Collection Idees, Gallimard (1981)
21. Queneau, R.: 100.000.000.000.000 de poemes. Gallimard Series, Schoenhofs Foreign Books (1961)
22. Toivanen, J., Toivonen, H., Valitutti, A., Gross, O., et al.: Corpus-based generation of content and form in poetry. In: Proceedings of the Third International Conference on Computational Creativity. pp. 175–179. Dublin, Ireland (2012)
23. Toivanen, J.M., Järvisalo, M., Toivonen, H., et al.: Harnessing constraint programming for poetry composition. In: Proceedings of the Fourth International Conference on Computational Creativity. pp. 160–167. Sydney, Australia (2013)
24. Zelaia, A., Arregi, O., Sierra, B.: Combining singular value decomposition and a multi-classifier: A new approach to support coreference resolution. Eng. Appl. of AI 46, 279–286 (2015)

Emotional Poetry Generation

Aitzol Astigarraga, José María Martínez-Otzeta, Igor Rodriguez, Basilio Sierra, and Elena Lazkano

Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Donostia-San Sebastian 20018, Spain
www.sc.ehu.es/ccwrobot
{aitzol.astigarraga,josemaria.martinezo,igor.rodriguez,b.sierra,e.lazkano}@ehu.eus

Abstract. In this article we describe a new system for the automatic creation of poetry in Basque that not only generates novel poems, but also creates them conveying a certain attitude or state of mind. A poem is a text structured according to predefined formal rules, whose parts are semantically related and with an intended message, aiming to elicit an emotional response. The proposed system receives as an input the topic of the poem and the affective state (positive, neutral or negative) and tries to give as output a novel poem that: (1) satisfies formal constraints of rhyme and metric, (2) shows coherent content related to the given topic, and (3) expresses them through the predetermined mood. Although the presented system creates poems in Basque, it is highly modular and easily extendable to new languages.

Keywords: Poetry generation, sentiment analysis, Basque language, affective computing

1 Introduction

Writing poetry requires both creativity to construct a meaningful message and lyrical skills to produce rhyme patterns and follow metrical constraints. Furthermore, oral poetry, poetry constructed without the aid of writing [1], implies that a work has to be composed and performed at the moment, with no prior preparation. Nowadays many improvisational oral practices exists around the world, such as Serbo-Croatian guslars [1], freestyle rap [2] and Basque *bertsolaritza*. It is obvious that improvising novel poems under challenging formal constraints, transmitting an intended message and all that at once, in front of an audience requires both high technical skills and creativity.

That is exactly our main goal: to develop a system that is able to generate Basque poetry under the constraints of *bertsolaritza*, a form of oral and improvised poetry.

Basque, *euskara*, is the language of the inhabitants of the Basque Country. And *bertsolaritza*, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive. In the book The Art of Bertsolaritza [3], it is described as a sung, rhymed and metered discourse. The *bertsolari* performs without the help of any musical instrument; but the *bertsolari*s discourse is always sung.

The Basque *bertso* follows strict constraints on meter and rhyme. In the case of a metric structure of verses known as *Zortziko Txikia*, the poem is composed of 4 verses, called *puntuak*. Each verse has 13 syllables and must rhyme with others.

Rhyme is the formal quid of the *bertso*; without the rhyme there is no *bertso*. If we rhyme (although its quality may not be the best), we are creating a *bertso*. If the two rhymed words in the same verse turn out to be the same, the *bertsolari* is considered to have committed a *poto*. It is simply the act of repeating a rhyming word but is undoubtedly the most penalized mistake both for a judging panel and for the public.

Although technical aspect of a *bertso* are highly demanding, the quality of the *bertso* is reflected in its force of reasoning and in its poetical-rhetorical value [4].

Our proposed system, based on a corpus-based poetry generation approach, uses two methods to construct poems according to given constraints on rhyme, meter, semantic similarity and sentiment. Thus, the system can be asked to view a topic (eg *spring*)from a particular affective stance (eg. *negative*). In doing so, the goal is to not only convey a message in a form of a poem but also to respond to an affective target and/or to create an affective response in the audience. That is, creating a poem in an intentional way.

The rest of the paper is organized as follows: in section 2 related work is surveyed, while section 3 is devoted to present the developed tools and resources. The proposed verse-maker module is presented in section 4, and results are drawn in 5 . Finally the conclusions are summarized in section 6.

2 Related Work

Computational modeling for poetry generation has become a topic in the artificial intelligence community in the last years. Before the computer science community took an interest in the area, people with a background closer to humanities made early efforts in systematic generation of poetry. We could mention works related to generating variations over a predetermined set of verses [5], or to select a template to produce poems from it [6].

According to Gervás [7], nowadays two main strategies can be outlined in the field of computer generation of poetry: corpus-based approach and composition from scratch.

In the corpus-based approach the computer is used to harvest and reuse text already formatted into poem-like structure of lines. This approach can be formulated as an information retrieval task, where the objective is to extract and select existing lines to compose new poems. Many computer-based systems rely nowadays in this method. Most relevant include PoeTryMe [8], the poetry generation platform used for Portuguese; an approach using text mining methods, morphological analysis, and morphological synthesis to produce poetry in Finnish presented in [9]; constraint programming for poetry composition explored in [10]; WASP [11] [12], a Spanish verse-generation system developed following the generate-and-test strategy; and [13], in which an approach of poetry generation based in POS-tag is presented. This corpus-based procedure is also adopted in our previous works [14] and [15] where two methods to ensure internal coherence of poems were presented.

On the other hand, the composition from scratch approach relies on building a stream of text from scratch, character by character or word by word, and establishing a distribution of the resulting text into poem lines by some additional procedure. An example of this procedure is the evolutionary system presented by Manurung in [16]. Poetry generation in Chinese language based on recurrent neural networks has also been analyzed in [17] and [18].

A popular -and rather simple- method to generate text from scratch is the N-gram model, which is the simplest Markov model. N-grams assign probabilities to sequences of words and the generated model can be used to stochastically generate sequences of words based on the generated distributions [19] [20]. An N-gram probability is the conditional probability of a word given the previous N-1 words. Markov chains have been widely used as the basis of poetry generation systems as they provide a clear ans simple way to model some syntactic and semantic characteristics of language [21]. Popular and recent examples of N-gram poem generators are [22], [23] and [7]. But text poetry hold non-local properties such as rhyme and metric that cannot be modeled by an ordinary N-gram model. Therefore, the above mentioned methods need additional procedures for distributing the resulting text into poem lines with metrical and rhyming constraints.

Poetry tries to convey messages and to evoke emotions in an aesthetic way. The sentiment conveyed in a text is a line of research within the Natural Language Processing (NLP) area that has drawn a lot of interest recently. The goal of text sentiment analysis is to extract the affective information or writers attitude from the source text [24]. Basically the sentiments may be considered within the polarity classification (positive, negative or neutral) [25].

The computational methods for sentiment analysis are usually based either on machine learning techniques such as naive Bayes classifiers trained on labeled dataset, or use lists of words associated with the emotional value (positive-negative evaluation or sentiment score values). A survey of current techniques is presented in [26].

It is a challenging task for a system intended to interact with people, to combine in a single system the above reviewed automatic poetry generation and sentiment analysis capabilities. That is, the development of a system not only capable of creating meaningful poems, but also of creating them with emotional personality.

An overview of works that incorporate emotional affects in creation process must include Full face poetry system [27], a corpus based poetry generator that creates poems according to days mood estimated from the news of the day. Another example would be the Stereotrope system [28] which generates emotional and witty metaphors for given topic based on corpus analysis. An interesting approach is also described in [29], in which a system capable of expressing feelings in the form of a poem is presented. The emotional state is extracted from text. Finally, MASTER [30] is a computer-aided tool for poetry generation. In this approach, a society of agents with initial moods and words influences each other to create the final poem.

Our approach also tries to combine both approaches: detecting specific emotions and transmitting them through the poem.

3 Resources

Several linguistic tools and resources have been developed and used in the verse-maker project.

3.1 Corpora

It has been shown already that the use of human generated corpora (oral or text) is of common use in computational poetry. It has the advantage of avoiding the generation of text that is un-interpretable. However, it may be interpreted as plagiarism. Hence, we have chosen initially to work with phrases mined from a newpaper alongside sentences extracted from the work of well-known *bertsolaris*. The later reflects the desire to maintain the language-model of the *bertsolaritza*, and the former tends to increase quality while not appropriating text intended for poems.

- **Mixed-corpus**: 18913 lines and 94314 words, of which them 21411 are unique. This corpus is a compilation of sentences mined from Basque newspaper Egunkaria[1] (85%) alongside poetry sung in *bertsolari* contests by different performers[2] (15%).

3.2 Text applications

- **Rhyme search**: finding words that rhyme with a given word is an essential task that the verse-generation system must perform. Basque rhyme schemes are mainly consonant. The widely consulted rhyming dictionary *Hiztegi Errimatua* [31] contains a number of documented phonological alternations that are acceptable as off-rhymes. These alternations have been implemented using regular expressions.
- **Syllable counter**: counts the number of syllables present in the given text. For the syllabification itself, the approach describing the principal elements of Basque language structure [32] has been implemented.
- **Similarity measure**: the main purpose of this module is to measure the semantic relationship between pairs of words and sentences. The module computes for each pair of words/sentences a score that evaluates how similar they are. Latent Semantic Analysis (LSA) method [33] [34] has been implemented to capture the semantic relatedness. This semantic model has been generated with news mined from Egunkaria, a Basque-language newspaper.
- **Sentiment analysis**: to extract the sentiment evaluation, we use the EliXa tool, a supervised Sentiment Analysis system [35]. It estimates the negative, neutral and positive sentiment values in short texts. The polarity classification is addressed by means of a multiclass SVM algorithm which includes lexical based features such as the polarity values obtained from domain and open polarity lexicons.

[1] https://en.wikipedia.org/wiki/Egunkaria
[2] http://bdb.bertsozale.eus/en

4 Verse-maker module

The aforementioned modules have been integrated into a verse-maker architecture for automatic poem generation. In the basic scenario, a topic and a sentiment to be expressed (positive, negative or neutral) is given by the user and the system then aims to give as output a novel poem that satisfies the formal constraints, conveys a predetermined sentiment and also shows coherent content related to the given topic.

For our particular challenge, the selected stanza will be *Zortziko Txikia*, a poem consisting of four lines, each 13 syllables long and all of them sharing a common rhyme.

The general problem of poem generation can be split up into two subproblems: the generation of content and the combination of fragments into the final poem.

4.1 Sentence Generator

The Sentence Generator module is used to compose meaningful and metrically correct natural language sentences. Two corpus-based methods are used towards this end:

- **Harvesting a corpus to retrieve sentences that fulfill the constraints.** The basics of this method is to extract sentences from the corpus that meet rhyme and metric constraints. All the sentences created in this way will come from the corpus *verbatim*. This approach is useful as a way to test the corpus potential to bring meaningful poems without further processing and, therefore, as a benchmarking for other approaches.
- **Generating sentences from scratch using an N-gram model.** Starting from the rhyming word, the verse is built backwards using the selected N-gram model; extending at each step the sequence of words with new ones that have a non-zero probability of appearing after the last word. When this approach is used, it is assured that the final sentence is different from the existing ones in the original corpus.

4.2 Poem Generator

This module organizes sentences such that they suit a target template of a poetic form (*Zortziko Txikia*). After that, according to some heuristics, sentences are selected to form the final poem. The following criteria is used to select candidate sentences:

- **Semantic similarity with respect to the given theme.** It is measured according to the cosine distance of the sentence and the theme when both are represented as vectors in the space generated by a LSA model.
- **Sentiment value of the text.** It is assured that the sentiment conveyed by the poem is the desired one. Currently we demand that all the verses in the poem share the same sentiment, but it could be possible to easily adapt the system to evaluate the poem as a whole, or analyze it by chunks and tolerate some percentage of them to differ.
- **Rhyme.** The sentences that are part of a poem have to comply with some rhyming constraints, that have to be enforced.

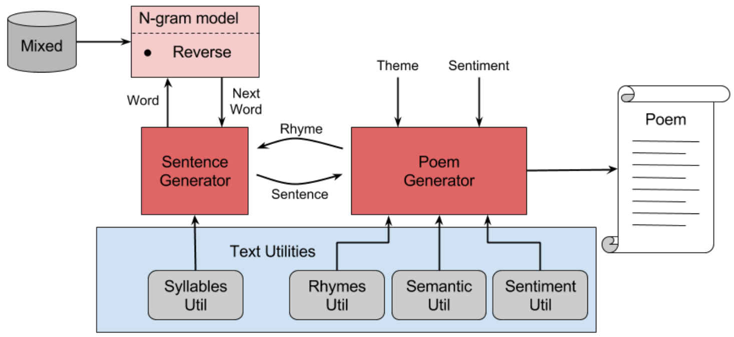


Fig. 1. System architecture.

The overall architecture, depicted in Figure 1, is modular and provides a high level of customization, depending on the needs of the user.

The procedure to create a poem involves prompting the user for a *theme*, a *sentiment* to be expressed, a *metric* and a background *corpus*. This is captured by the algorithm 1.

```
Data: topic, sentiment, corpus, numPoems
Result: poems related to the topic with conveying sentiment

wordList ← ExtractSentenceEndingWords(corpus);
simList ← OrderBySimilarity(wordList, topic);
while not generated numPoems do
    rhymeBase ← pop(simList);
    createBertso(rhymeBase);
end
Procedure createBertso(rhymeBase)
    rhymeList ← getRhymes(rhymeBase);
    while bertso not generated do
        if NgramGenerate(rhymeList) has sentiment then
            add line to bertso;
        end
    end
    return;
```

Algorithm 1: general procedure for *bertso* generation

With this process it is possible that in some step no valid sentence is found (in case of rarely occurring bigrams) or that a combinatorial explosion occurs (in the case of bigger corpus or longer number of syllables). In our case the latter situation is prevented imposing a maximum number of generated sentences. If the former problem arises, the system keeps trying another last words till the list is exhausted. An additional check ensures that no solution copies *verbatim* lines in the original corpus. In our experiments always a minimum of five poems have been generated.

5 Results

In order to test the capabilities of the above described approach, a series of experiments have been carried out. We have tested 2-gram and 3-gram models and the preliminary results have shown that the 3-gram model tended to replicate the corpus almost verbatim. Therefore we have used 2-grams, because we wanted our system to produce tentative solutions somewhat different to the original lines, and this is more likely with low order N-grams.

In order to compare the generated poems with the original lines in the corpus, another variant of the algorithm has been tested, where instead of creating verses backwards with an N-gram model, the original verses are the candidates. From those verses poems are created following the previously explained procedure.

In Tables 1 and 2 we show four poems composed by the automated system, in its original form and in an approximate English translation. Two are made with sentences mined from the original corpus and the other two produced from scratch with the 2-gram model. Some general conclusions can be drawn:

- The emotional affect of the poems can be clearly appreciated.
- The generated poems are related to the subject. This relationship is not only appreciated through the repetition of the key word or theme, but also through the incursion in the poem of words semantically similar to the theme (eg with the theme *music*, the terms *tone*, *classic*, *concert* and *public* appear).
- With respect to the content generator methods, corpus-harvesting and N-gram model, the corpus-harvesting method ensures the internal coherence of the sentences (since it extracts entire phrases from the corpus) but, on the other hand, creates more rigid poems. It can be seen sentences related to everyday news (due to the influence of the corpus used) that are hardly related to the proposed theme.
- The N-gram method is more flexible, malleable, and seems to get closer to the given topic. But, the toll to be paid is that flexibility is sometimes translated into unintelligible phrases.

Table 1. Two poems created by the system with the theme 'music' and with lines from the original corpus. On the left with negative sentiment and on the right with positive sentiment.

Basque	Gutxien ezagutzen zen musikaria hak bertsotan aurkeztu zuen jaialdia haiengandik aldendu hori nire nahia Bolibarko Txikito puntista ohia	Basque	Atzo eskaini zuten lehen kontzertua eta bertsolariok antza talentua Gorrotxategirena dugu sonatua proposizioaren bigarren puntua
English	The least known of the musicians he introduced the festival improvising verses get away from them, that's my goal Txikito de Bolivar, ex pelotari	English	Yesterday they offered the first concert and the bertsolaris, it seems to be, talent Gorrotxategi is well known the second point of the proposition

Table 2. Two poems created by the system with the theme 'music' and with lines created from scratch with 2-grams. On the left with negative sentiment and on the right with positive sentiment.

	Negative	Positive
Basque	Ez har ta ez det lortu nahi dute tonua klasiko bat da ta ez dezu zuk lekua eta beste bat izan zen denen patua jo ta orain ezin da bere sekretua	Zaintzen nahiko lan daukat emanaldiekin eta libre izan nahi dut entzuleekin zorion gehiago gaur ez gaude ezberdin nere musika ez al duzu zuk atsegin
English	Don't pick up, I haven't got their tone It's a classic and you have no room and the destiny of all was another and now also you can not your secret	I've enough taking care of myself in concerts and I want to be free with the listeners more happiness, today we are not different Don't you like my music?

6 Conclusions and Further Work

In this paper we have presented an automated system to generate poetry in Basque language. The proposed method not only generates novel poems, but also creates them conveying a certain attitude or state of mind. The system receives as an input the topic of the poem and the affective state (positive, neutral or negative) and tries to give as output a novel poem that satisfies formal constraints of rhyme and metric, shows coherent content related to the given topic and expresses them through the predetermined mood.

Two methods has been tested for creating content: on the one hand composing verses from scratch and putting together phrases already made by the other. The performed experiments have served as a basis for better insight of the proposed methods as well as to mark the way and future lines of work. We are already working on further enhancements to each of the processes involved, including:

1. We are trying to test another ways to assert the sentiment of a whole poem. That is, capturing the general sentiment conveying of the poem, instead of calculating it line by line. The main message of a *bertso* goes always at the end. Therefore, perhaps we could only take into account the sentiment of the last line, leaving free the sentiment of the others.
2. Exploring new models to measure the semantic coherence of the poem. We are currently experimenting with a word embedding approach to measure semantic similarity between sentences.
3. Implementing improved methods to generate phrases for templates and working with other corpora. To generate the Markov chain, the use of other bigger or more diverse corpus would also be interesting.
4. Improving the performance of the algorithm to create verses in real-time (as real *bertsolaris* do) and use it in live performances.

7 Acknowledgments

This paper has been supported by the Spanish Ministerio de Economía y Competitividad, contract TIN2015-64395-R (MINECO/FEDER, UE), as well as by the Basque Government, contract IT900-16.

References

1. Lord, A.B., Mitchell, S.A., Nagy, G.: The singer of tales. Volume 24 of Harvard Studies in Comparative Literature. Harvard University Press (2000)
2. Pihel, E.: A furified freestyle: Homer and hip hop. Oral Tradition **11**(2) (1996) 249–69
3. Garzia Garmendia, J., Sarasua, J., Egaña, A.: The art of bertsolaritza: Improvised basque verse singing. (2001)
4. Egaña, A.: Oral tradition. **22** (2007)
5. Queneau, R.: 100.000.000.000.000 de poemes. Gallimard Series. Schoenhofs Foreign Books (1961)
6. Oulipo, A.: Atlas de litterature potentielle. Collection Idees. Gallimard (1981)
7. Gervás, P.: Constrained creation of poetic forms during theme-driven exploration of a domain defined by an n-gram model. Connection Science **28**(2) (2016) 111–130
8. Oliveira, H.G., Cardoso, A.: Poetry generation with poetryme. In: Computational Creativity Research: Towards Creative Machines. Springer (2015) 243–266
9. Toivanen, J., Toivonen, H., Valitutti, A., Gross, O., et al.: Corpus-based generation of content and form in poetry. In: Proceedings of the Third International Conference on Computational Creativity, Dublin, Ireland (2012) 175–179
10. Toivanen, J.M., Järvisalo, M., Toivonen, H., et al.: Harnessing constraint programming for poetry composition. In: Proceedings of the Fourth International Conference on Computational Creativity, Sydney, Australia (2013) 160–167
11. Gervás, P.: WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In: Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI, Birmingham, United Kingdom (2000) 93–100
12. Gervás, P.: Engineering linguistic creativity: Bird flight and jet planes. In: Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity, Association for Computational Linguistics (2010) 23–30
13. Agirrezabal, M., Arrieta, B., Astigarraga, A., Hulden, M.: Pos-tag based poetry generation with wordnet. In: ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria. (2013) 162–166
14. Astigarraga, A., Jauregi, E., Lazkano, E., Agirrezabal, M.: Textual coherence in a verse-maker robot. In: Human-Computer Systems Interaction: Backgrounds and Applications 3. Springer (2014) 275–287
15. Astigarraga, A., Agirrezabal, M., Lazkano, E., Jauregi, E., Sierra, B.: Bertsobot: the first minstrel robot. In: Human System Interaction (HSI), 2013 The 6th International Conference on, IEEE (2013) 129–136
16. Manurung, H.: An evolutionary algorithm approach to poetry generation. PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics. (2004)
17. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. (2014) 670–680
18. Yan, R.: i, Poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), New York, United States (2016) 2238–2244
19. Jurafsky, D., James, H.: Speech and language processing an introduction to natural language processing, computational linguistics, and speech. (2000)
20. Manning, C.D., Schütze, H., et al.: Foundations of statistical natural language processing. MIT Press (1999)

21. Langkilde, I., Knight, K.: Generation that exploits corpus-based statistical knowledge. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics (1998) 704–710
22. Barbieri, G., Pachet, F., Roy, P., Esposti, M.D.: Markov constraints for generating lyrics with style. In: Proceedings of the 20th European Conference on Artificial Intelligence, IOS Press (2012) 115–120
23. Das, A., Gambäck, B.: Poetic machine: Computational creativity for automatic poetry generation in bengali. In: 5th International Conference on Computational Creativity, ICCC. (2014) 230–238
24. Pang, B., Lee, L., et al.: Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval **2**(1–2) (2008) 1–135
25. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 347–354
26. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-Based Systems **89** (2015) 14–46
27. Colton, S., Goodwin, J., Veale, T.: Full face poetry generation. In: Proceedings of the Third International Conference on Computational Creativity. (2012) 95–102
28. Veale, T.: Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In: Proceedings of the International Conference on Computational Creativity. (2013) 152–159
29. Misztal, J., Indurkhya, B.: Poetry generation system with an emotional personality. In: Proceedings of the 5th International Conference on Computational Creativity. (2014) 72–81
30. Kirke, A., Miranda, E.: Emotional and multi-agent systems in computer-aided writing and poetry. In: Proceedings of the Artificial Intelligence and Poetry Symposium. (2013) 17–22
31. Amuriza, X.: Hiztegi Errimatua. Lanku (1981)
32. Jauregi, O.: Euskal testuetako silaba egituren maiztasuna diakronikoki. Anuario del Seminario de Filología Vasca" Julio de Urquijo" **37**(1) (2013) 393–410
33. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6) (1990) 391–407
34. Zelaia, A., Arregi, O., Sierra, B.: Combining singular value decomposition and a multi-classifier: A new approach to support coreference resolution. Eng. Appl. of AI **46** (2015) 279–286
35. San Vicente, I., Saralegi, X., Agerri, R., Sebastián, D.S.: EliXa: A modular and flexible ABSA platform. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). (2015) 748–752

Poet's Little Helper: A methodology for computer-based poetry generation. A case study for the Basque language

Aitzol Astigarraga
José María Martínez-Otzeta
Igor Rodriguez
Basilio Sierra
Elena Lazkano
Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU
Donostia-San Sebastian 20018, Spain
aitzol.astigarraga@ehu.eus

Abstract

We present Poet's Little Helper (PLH), a tool that implements a methodology to generate poetry using minimal language-dependent information. The user only needs to provide a corpus with a set of sentences, a rhyme checker and a syllable-counter. From these building blocks, PLH produces: (1) an exploratory analysis of the suitability of the given corpus for poetry generation. (2) a novel and non-trivial poem grammatically correct under metrical and rhyming constraints. This poem also shows content that is coherent with a topic given by the user. The process of poetry generation is a cycle with three phases: lexical exploratory analysis, semantic exploratory analysis and stanza generation. The goal is twofold: on the one hand PLH aims to be a useful poem-generator for many languages with minimal effort; on the other hand, analizes how the particularities of each language affect in the creation of poems. The presented methodology is offered in a public repository with all the needed code. The results of an experiment with a set of sentences in Basque language is shown.

1 Introduction

Poetry is the art of creating texts structured according to predefined formal rules and whose parts are semantically related. Poetry generation is a challenging field in the area of natural language processing. When a poem is automatically created by computational means, usually the programmer takes advantage of existing general semantic knowledge from resources like WordNet for semantic validation, or of already known formalized grammars for sentence generation. Furthermore, most of the existing poetry-generation systems are devoted to English language (Oliveira, 2015). Therefore, for languages with a low profile in the natural language community, the most usual scenario is a lack of such computational resources.

We thus, have developed a methodology to help researchers in both directions:

- To generate poetry almost from scratch, using little more than a corpus of sentences, a function that evaluates if two given sentences rhyme, and another function to evaluate if a set of sentences is a valid stanza.
- A tool to compare poetry creation in different languages, rhyming patterns, syllable-construction, etc. in order to learn in which way affects each of them when generating text under constraints.

The rest of the paper is organized as follows: in section 2, the state of the art in poetry generation is summarized; in section 3 our approach is presented; section 4 is devoted to the source code organization; an application of the proposed method to the Basque language is shown in section 5; and in the final section 6, conclusions and possible future lines of research are presented.

2 Related Work

Computational modeling for poetry generation has become a topic in the artificial intelligence community in the last years. Before the computer science community took an interest in the area, people with a background closer to humanities made early efforts in systematic generation of poetry. We could mention works related to generating variations over a predetermined set of verses (Queneau, 1961), or to select a template to produce poems from it (Oulipo, 1981).

According to (Gervás, 2016), nowadays two main strategies can be outlined in the field of computer generation of poetry: corpus-based approach and composition from scratch.

In the corpus-based approach the computer is used to harvest and reuse text already formatted into poem-like structure of lines. This approach can be formulated as an information retrieval task, where the objective is to extract and select existing lines to compose new poems. Many computer-based systems rely nowadays in this method. Most relevant include PoeTryMe (Oliveira and Cardoso, 2015), the poetry generation platform used for Portuguese; an approach using text mining methods, morphological analysis, and morphological synthesis to produce poetry in Finnish presented in (Toivanen et al., 2012); constraint programming for poetry composition explored in (Toivanen et al., 2013); WASP (Gervás, 2000) (Gervás, 2010), a Spanish verse-generation system developed following the generate-and-test strategy; and (Agirrezabal et al., 2013), in which an approach of poetry generation based in POS-tag is presented.

On the other hand, the composition from scratch approach relies on building a stream of text from scratch, character by character or word by word, and establishing a distribution of the resulting text into poem lines by some additional procedure. An example of this procedure is the evolutionary system presented by Manurung in (Manurung, 2004). Poetry generation in Chinese language based on recurrent neural networks has also been analyzed in (Zhang and Lapata, 2014) and (Yan, 2016).

A popular -and rather simple- method to generate text from scratch is the N-gram model, which is the simplest Markov model. N-grams assign probabilities to sequences of words and the generated model can be used to stochastically generate sequences of words based on the generated distributions (Jurafsky and Martin, 2000) (Manning et al., 1999). An N-gram probability is the conditional probability of a word given the previous N-1 words. Markov chains have been widely used as the basis of poetry generation systems as they provide a clear ans simple way to model some syntactic and semantic characteristics of language (Langkilde and Knight, 1998). Popular and recent examples of N-gram poem generators are (Barbieri et al., 2012), (Das and Gambäck, 2014) and (Gervás, 2016).

Our poetry generation strategy is a corpus-based method (Astigarraga et al., 2013) and the overall semantic relationship has been implemented with an LSA model (Deerwester et al., 1990) (Astigarraga et al., 2014). The verse generation procedure relies in the extraction of sentences from corpora and combining them (under rhyme and metric constraints) to form the final poem. The LSA model assures the internal coherence between poem lines and the overall coherence with respect to a theme.

3 Methodology

As previously stated, the goal is twofold: firstly, to assess the potential of some given building blocks to produce stanzas, and secondly to propose a way to build stanzas related to a given theme. These two goals can be stated as exploratory analysis and poetry generation, respectively. The overall process is a cycle with three phases: lexical exploratory analysis, semantic exploratory analysis and stanza generation.

Let us start with some terminology. $\mathbb{N}$ will denote the set of natural numbers. $\mathbb{R}$ will denote the set of real numbers. A document d is a sequence of words (w) and punctuation marks (m) and spaces which follow the syntactic conventions of the language. W is the set of all possible words, M the set of all punctuation marks, and D is the set of all possible documents. A verse v is a document d under some restrictions, while the set of all possible verses will be V. The power set of a set S will be denoted with $\mathcal{P}(S)$.

We next explain the three phases of the methodology in terms of the input required by the user, the tools already provided by the system and the output generated.

3.1 Lexical exploratory analysis

This step is intended to explore the lexical dimension of the data and its suitability for poem generation.

System input:

- *A set of documents (Dv) which is used to obtain the verses.*
- *A rhyming function ($is\_rhyme : D^2 \rightarrow \{T, F\}$) that returns True if the two documents rhyme, and False otherwise.*

Tools:

- *A rhyme detector function ($rhyme\_generator : D \times \mathcal{P}(V) \rightarrow \mathcal{P}(V)$), which returns all the rhyming verses given a document.*

System output:

- *Count the number of potential verses.*
- *Find the number of verses which do not adjust to the rhyming convention, and show their endings.*
- *Find the number of verses which rhyme with a given word, and list them.*
- *Compute the number of rhyming equivalence classes of the set of verses.* A rhyming equivalence class is a set of verses which share the same rhyming pattern.
- *Compute the number of rhyming equivalence classes of the set of verses that have more elements than the minimum number of rhyming verses in a stanza.* This is the number of valid equivalence classes, in the sense that elements from the other equivalence classes cannot form part of a stanza.
- *Create a list with a verse from every equivalence class along with the number of elements in such equivalence class.*
- *Plot the number of verses in each equivalence class.*
- *Plot the logarithm of the number of verses in each equivalence class.* Useful when the distribution is very skewed.
- *Plot the histogram of the number of equivalence classes according to the equivalence class size.* Another way of exploring the distribution of equivalence classes according to their size.
- *Plot the histogram of the number of equivalence classes according to the logarithm of the equivalence class size.*

3.2 Semantic analysis

In this step the lexical data is projected onto the semantic model to gain further insight into the suitability of the data for poetry with a meaning.

System input:

- *A set of documents (Ds) which is used to infer the semantic models used later.*
- *A set of (M) of punctuation marks.* The elements of this set will be removed when performing semantic analysis.
- *A natural number $NV \in \mathbb{N}_{\neq 0}$ denoting the number of verses in a stanza.*
- *A lemmatizer function ($lemmatize : W \rightarrow W$), which returns a lemmatized word.*

Tools:

- *A function ($clean\_document : D \times M \rightarrow D$) which returns the document without punctuation marks.*
- *A function ($semantics\_extractor : D \times \mathbb{N} \times W^* \times \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{S}$) that returns a semantic model.* This function takes five parameters to generate a semantic model. The parameters are: a set of documents used to build the semantic model; the number of topics which should be used to create the semantic model; a list of words to be removed from the documents; a parameter indicating the minimum document number in which a word should appear; and a parameter indicating the maximum fraction of the total documents in which a word could appear.

- *A similarity function ($sim : D^2 \rightarrow \mathbb{R}$), which returns the similarity between two documents.*

System output:

- *Build a semantic model from the set of documents Ds provided by the user.*
- *Find the verses more similar to a given theme according to the semantic models.*
- *Find the verses more similar to a given theme according to the semantic models and that also rhyme with a sentence.*

3.3 Poetry generation

This step performs the actions needed to create a well-formed poem with respect to metrical and rhyming constraints, as well as the semantic intended meaning.

System input:

- *A sequence of rhyme patterns $RP \in \{0, 1, ..., NV\}^{NV}$.* The rhyme patterns in the stanza have to be encoded in the following manner: the rhyme pattern of the first verse corresponds to the number zero; for the pattern of a new verse, if such a rhyming pattern already exists, the number corresponding to that pattern is written, or the first natural number not yet chosen otherwise. For example, four verses with the same rhyming pattern would be written as 0, 0, 0, 0. Six verses with rhyming patterns by consecutive pairs will be 0, 0, 1, 1, 2, 2. The pattern 0, 1, 1, 2, 2, 1, 1, 0 would corresponding to eight verses where the last three patterns are the same as the first three patterns, but reversed.
- *A function ($extract\_verses : D \rightarrow \mathcal{P}(V)$) that extracts all the potential verses from a document.*

Tools:

- *Several stanza generator functions ($stanza\_generator : V \times D \times \mathbb{N} \times \mathbb{N}^{NV} \times \mathbb{S} \rightarrow \mathcal{P}(V)$), that given a set of verses, a document, a number of verses, a rhyming pattern and a semantic model, returns a stanza under the constraints imposed by the number of verses and the rhyming pattern; the stanza will be semantically related to a document under a semantic model.* The document above referred can be viewed as the theme of the stanza. These generator functions will differ in their inner implementation of another two auxiliary functions: a stanza validator function ($stanza\_validator : \mathcal{P}(V) \rightarrow \{T, F\}$), which returns True if the stanza conforms to the poetry rules, and False otherwise, and a stanza goodness function ($stanza\_goodness : \mathcal{P}(V) \rightarrow \mathbb{R}$), which returns a value for every set of verses according to its quality.

System output:

- *Ignore equivalence classes with fewer elements that the minimum needed.* In this step the equivalence classes from which a stanza cannot be created are ignored.
- *Compute the best stanzas given a theme according to a goodness function.* Several goodness functions are available to create stanzas.

Lexical and semantic richness is largely required for acceptable stanza generation, and exploring the lexical and semantic dimensions of the data could help the researcher to focus on improving it along their weaknesses.

4 Open source code

A public repository has been created with the basic code needed to implement this methodology. The code is a collection of Python modules and notebooks, which, after installation, permits people with little knowledge of the Python language to perform the analysis on their own.

The structure of the code is the following:

4.1 Modules

- *Customize.* It contains the following definitions, that have to be customized according to the needs of the researcher: the filenames where D_s and D_v are stored, *M* and *RP* as Python tuples, the natural number *NV*, and the functions *extract_verses*, *lemmatize* and *is_rhyme*.

- *General.* General functions not directly related to natural language processing or poetry generation. For instance, list manipulation or histogram drawing functions are defined here.

- *NLP.* Functions related to natural language processing. The construction and manipulation of semantic models takes place here.

- *Poetry.* Functions related to poetry generation. Rhyming and stanza construction takes place here.

4.2 Notebooks

- *Get_started.* This notebook will be called from the other two. The research does not need to open it, given that it only contains some code that has to be executed before further analysis is made.

- *Exploratory_analysis.* Here, the code that permits exploration of the lexical and semantic possibilities of the verses is located.

- *Stanza_generation.* The notebook where the last step, the poetry generation, is performed. The researcher, after analyzing the data with the *Exploratory_analysis* notebook, is ready to execute this code and create automated poetry.

5 Case study

Verse improvisation (under the name of *Bertsolaritza*) is a traditional cultural expression in the Basque Country. With ancient roots, it has undergone a revival in the last times, being widely popular.

In this section we present the experiments made with a corpus in Basque language. The results of the exploratory analysis are shown along with some automatically produced poetry. The poetry meter is *Zortziko Txikia*, a composition of eight lines in which odd lines have seven syllables and even ones have six. The union of each odd line with the next even line, forms a strophe. Each strophe has 13 syllables with a caesura after the 7th syllable (7 + 6) and must rhyme with the others [1].

[1] http://www.bertsozale.eus/en/bertsolaritza/what-is-a-bertso

5.1 Building blocks

As previously said, some building blocks are needed in order to apply the proposed methodology. In this case those blocks were defined in the following manner:

- The set of documents Ds from which the semantics models are created is the set of all the news that appeared in the Basque newspaper Egunkaria in the years 2002-2003.

- The set of documents Dv from which extract the potential verses is equivalent to Ds.

- The set of punctuation marks is M = (, . ? ! ” ’ / \).

- A $syllable\_counter$ function that counts the syllables of the input text.

- The rhyming function is_rhyme that returns all the rhyming lines given an input line. It is based on (Amuriza, 1981) and implemented using regular expressions.

- The natural number NV that denotes the number of verses in a stanza. In *Zortziko Txikia* this number is 4.

- The sequence of rhyme patterns RP. In *Zortziko Txikia* this sequence is (0, 0, 0, 0). It means that all the verses have to rhyme among themselves.

5.2 Lexical exploratory analysis

The following actions have been performed automatically with the help of our lexical exploratory software:

- Count the number of potential verses: 41659.

- Find the number of verses which do not adjust to the Basque rhyming conventions: 139. Percentage over the total: 0.33%.

- Find the last words of such verses. Analyzing these words we find the sign %, making us wonder if we should expand the set M, filter these kind of characters or make another decision. We also find interjections (”eh”, ”hi”), foreign proper names (”olaf”,

”jerusalem”, ”bush”), Basque proper names (”unanue”, ”orue”), Roman numerals (”xix”, ”xx”, ”xxi”), acronyms (”eajk”, ”ugt”, ”upn”) and other words not easily classifiable. After these step we could decide what to do in every case: for example, we could modify the rhyming function to add those Basque names, expand acronyms or Roman numerals in the original documents and repeat the verse extraction process, or remove all the no rhyming verses. We have chosen this last option, that is provided by our software.

- Compute the number of equivalence classes of the set of verses, according to the rhyme. In this example the number of partitions is 184.
- Compute the number of equivalence classes of the set of verses that have more elements than the minimum number of rhyming verses in a stanza. In Zortziko Txikia the rhyme pattern (RP) is (0, 0, 0, 0), which means any valid partition has to contain at least four elements, because a stanza is composed by four rhyming verses. The number of equivalence classes of minimum size in our example is 141. If RP would be (0, 1, 0, 1, 0, 1), the minimum equivalence class size would be three, and in the case of (0, 1, 0, 1, 2, 2), the minimum size would be two.
- Create a list with a verse of every equivalence class along with the number of elements in such equivalence class. Our list is of the form [(’a aita zenarekin joan zen bertara’, 4441), (’a arrieta hartu zituzten mendean’, 4209), ... , (’zeelandarrekin ez da ariko lomu’, 1), (’ziganda badiola zalakain gaztelu’, 1)].
- Plot the number of verses in each equivalence class (figure 1), the logarithm of the number of verses in each equivalence class (figure 2), the histogram of the number of equivalence classes according to the equivalence class size (figure 3) and the histogram of the number of equivalence classes according to the logarithm of the equivalence class size (figure 4).

These four figures can help the user in the interpretation of the distribution of the rhyming equivalence classes and their relative size, that appears to follow a power law (Piantadosi, 2014).

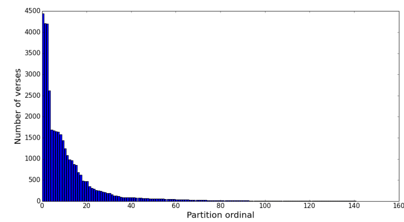


Figure 1: Number of verses in each equivalence class

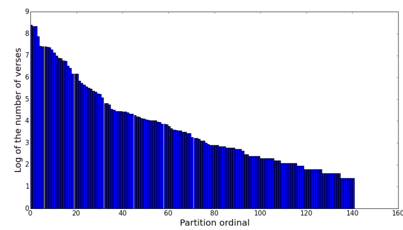


Figure 2: Logarithm of the number of verses in each equivalence class

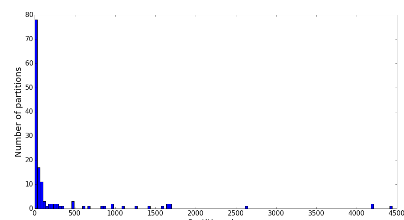


Figure 3: Histogram of the number of equivalence classes according to the equivalence class size

5.3 Semantic exploratory analysis

The following actions have been performed automatically with the help of our semantic exploratory software:

- Build a semantic model from the set of documents Ds provided by the user. The number of topics has been assigned to 100, filtering all the words that do not appear in at least 5 documents, and all the words that appear in more than 20% of the documents. We have also filtered the stopwords.

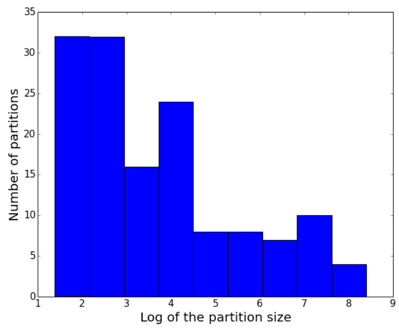


Figure 4: Histogram of the number of equivalence class according to the logarithm of the equivalence class size

- Find the verses more similar to a given theme according to the semantic models. Taking for example the theme "itsaso" (sea), we find sentences with the word "itsaso", but also sentences without that word, but other related words, as "txalupa" (small boat) or "ur" (water). We find sentences with the word "ontzi", that could be translated as "ship" but also as a "generic container", highlighting the challenging issues which polysemy implies.

- Find the verses which are more similar to a given theme according to the semantic models and that also rhyme with a sentence. Following with the "itsaso" example, the more similar verse is 'oliobideetan eta itsasoan', with a similarity value of 0.99833471). The three sentences more similar to "itsaso" that also fulfil the rhyming restrictions, along with their similarity values, are ('zenbait otzara eta ontzi hareatzan', 0.91221404), ('paper eta ontzien birziklapenean', 0.88736451), and ('delas eta izura oraindik ontzian', 0.83049005).

5.4 Poetry generation

For a stanza to be valid, it has to be composed of the number of verses given by *NV* and follow the rhyming pattern given by *RP*. No two verses are allowed to share the same final word. In the following we will refer to the document to which the stanza has to be semantically related, as the theme t. Two different *stanza_generator* strategies have been used to build a stanza.

1. Choose the verse v more similar to the theme t. Then choose the $NV - 1$ verses more similar to t that follow the rhyming pattern RP.

2. Choose the ten verses v more similar to the theme t. Then, for each of the ten verses, choose the $NV - 1$ verses more similar to t that follow the rhyming pattern RP. The stanza with highest score is chosen.

For each of these two functions, two examples are shown, using different themes t. In the first example, the theme "itsaso" (sea) has been chosen, as in the other "gurasoak" (parents) has been used. NV is equal to four, and the RP pattern is (0, 0, 0, 0), meaning that all the verses have to share the same rhyming pattern.

Table 1: Stanza created with the more similar verse (in Basque and its English translation)

Basque	oliobideetan eta itsasoan atentatu susmoak itsaso beltzean itsaso baretura itzuli nahian zenbait otzara eta ontzi hareatzan
English	in oil paths and in the sea attack rumors in the sea's darkness trying to return to calm sea several container and ships are going on

Table 2: Best verse in our opinion (in Basque and its English translation)

Basque	Kantauri itsasoa haserre zeharo ozeano haunditan egin dugu txango putz egitea ere tokatu ezkero maitatu eta negar egin baitut Bilbo
English	Cantabrian Sea very angry we have made a trip to the vast stormy ocean if we are emerged to make blow I have loved and cried, Bilbao

Example using the theme "itsaso" (sea).

The verse more similar to the theme is 'oliobideetan eta itsasoan' (in oil paths and in the sea). The stanza generated choosing the three verses more similar to to theme is shown in Table 1.

The same experiment has been performed choosing the best ten verses and then computing the best stanza among the ten ones generated. The same stanza is ranked the first with this approach, but in our opinion, the stanza in Table 2 (which ranked 8th of 10) is the best of all ten.

Table 3: Stanza created with the more similar verse (in Basque and its English translation)

Basque	nire gurasoentzat Peret kristona zen haiekin bi urteko alaba zeukaten familia osoak topa zitezkeen noizbait haur bat badator nahiz ez jakin nor den
English	Peret was very good in my parents' opinion with these people they had a two years old daughter for all the family to get together sometimes a child comes and I do not know who (s)he is

Table 4: Best stanza built from the ten more similar verses (in Basque and its English translation)

Basque	ikastolako haurren txanda izango da haurtzaindegietatik haur eskoletara haur danborradarako Easo Ederra eta gero eta haur gehiago dira
English	It will be school children turn from nursery to school children drum performance in San Sebastian there are more and more kids

The verse more similar to the theme is 'nire gurasoentzat Peret kristona zen' (Peret[2] was very good in my parents' opinion), and the stanza generated choosing the three verses more similar to the theme is shown in Table 5. When performing the same experiment with the best ten verses, another stanza, shown in Table 4 is chosen. As in the previous example, we found the stanza in Table 5 (ranked 10th of 10) the best for our liking.

Let us remember that the goal of the methodology and the associated code is to help the user to explore the possibilities of their data. In this example we find that the generated stanzas are not of high quality, and that even we do not agree with the relative ordering of them given by the code. So, which conclusions could we extract from these facts? We already have tools to find all the verses related to a theme ordered by relevance, so one first step could be to check if such ordering is suitable. If that it is not the case, it is very likely the process of the semantic model construction needs some tuning. Or maybe the problem lies in the few number of verses that rhyme with the most promising candidates. This could be also explored with our tool. In our case, it looks as if the verses with a haiku-like structure are better valued by our ear. This makes us wonder if the stanza goodness function could take into account this fact, and weight down the stanzas with all the verses very related to the theme, or those with too many repeated words between verses.

[2] A Spanish singer

Table 5: Best verse in our opinion (in Basque and its English translation)

Basque	ahizpa ere haurrei irakasten dabil aitona hola dabil makur eta ixil zuk errondak atera bai ibili trankil Londreseko Paddington geltokitik hurbil
English	(S)he is teaching to his/her sister and children the grandfather goes on bowed and silent you carry on calm proposing challenges close to London's Paddington station

6 Conclusions and further work

In this work a methodology to guide the exploration of the possibilities of a collection of documents to perform automatic poetry generation has been described. Along with it a tool and its source code written in Python has been presented. The possibilities of the system have been shown with an example in Basque language.

As further work, we intend to add more functionalities to the lexical and semantic exploratory subsystems, as well as to the poetry generation subsystem. Another ways of building stanzas, as for example using genetic algorithms or Markov chains, would be of interest. The *stanza_goodness* functions are fixed and predefined, but it would be possible to be customizable by the researcher. Another idea would be to get a feedback from the researcher or a knowledgeable user about the subjective goodness of a stanza, in order to improve the goodness functions that the system uses.

Acknowledgements

This paper has been supported by the Spanish Ministerio de Economía y Competitividad, contract TIN2015-64395-R (MINECO/FEDER, UE), as well as by the Basque Government, contract IT900-16.

References

Manex Agirrezabal, Bertol Arrieta, Aitzol Astigarraga, and Mans Hulden. 2013. POS-Tag based poetry generation with WordNet. In *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, pages 162–166.

Xabier Amuriza. 1981. *Hiztegi Errimatua*. Lanku.

Aitzol Astigarraga, Manex Agirrezabal, Elena Lazkano, Ekaitz Jauregi, and Basilio Sierra. 2013. Bertsobot: the first minstrel robot. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, pages 129–136. IEEE.

Aitzol Astigarraga, Ekaitz Jauregi, Elena Lazkano, and Manex Agirrezabal. 2014. Textual coherence in a verse-maker robot. In *Human-Computer Systems Interaction: Backgrounds and Applications 3*, pages 275–287. Springer.

Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. 2012. Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 115–120. IOS Press.

Amitava Das and Björn Gambäck. 2014. Poetic machine: Computational creativity for automatic poetry generation in bengali. In *5th International Conference on Computational Creativity, ICCC*, pages 230–238.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Pablo Gervás. 2000. WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100. Citeseer.

Pablo Gervás. 2010. Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, pages 23–30. Association for Computational Linguistics.

Pablo Gervás. 2016. Constrained creation of poetic forms during theme-driven exploration of a domain defined by an N-gram model. *Connection Science*, 28(2):111–130.

Daniel Jurafsky and James H. Martin. 2000. Speech and language processing. An introduction to natural language processing, computational linguistics, and speech.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

Christopher D. Manning, Hinrich Schütze, et al. 1999. *Foundations of statistical natural language processing*. MIT Press.

Hisar Manurung. 2004. *An evolutionary algorithm approach to poetry generation*. Ph.D. thesis, University of Edinburgh. College of Science and Engineering. School of Informatics.

Hugo Gonçalo Oliveira and Amílcar Cardoso. 2015. Poetry generation with PoeTryMe. In *Computational Creativity Research: Towards Creative Machines*, pages 243–266. Springer.

Hugo Gonçalo Oliveira. 2015. Automatic generation of poetry inspired by twitter trends. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 13–27. Springer.

Association Oulipo. 1981. *Atlas de litterature potentielle*. Collection Idees. Gallimard.

Steven T Piantadosi. 2014. Zipfs word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130.

Raymond Queneau. 1961. *100.000.000.000.000 de poemes*. Gallimard Series. Schoenhofs Foreign Books.

Jukka Toivanen, Hannu Toivonen, Alessandro Valitutti, Oskar Gross, et al. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the Third International Conference on Computational Creativity*.

Jukka Toivanen, Matti Järvisalo, Hannu Toivonen, et al. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity*, page 160.

Rui Yan. 2016. i, Poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. IJCAI.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 670–680.

9. Kapitulua

Diskurtsoaren Azterketa Ikasketa Automatikoa erabiliz

Bertsolariak, bere komunikazio-ekintza gauzatzerako orduan, beti estrategia bera erabiltzen du (Garzia et al., 2001). Ideia nagusia edo mezua pentsatzen du lehenbizi, aukeratutako metrikaren mende jarri eta bukaerarako uzten du ideia. Segidan errimak bilatu eta bertsoari ekiten dio, puntuz-puntu, bukaerarako gorde duen ideiara iritsi arte.

Estrategia hori aintzat hartuz gero, badirudi bertsoak, bertso bakoitzak, diskurtso beregaina osatzen duela. Baina elkarrekin partekatzen duten ezaugarririk ba ote dute? Ba al dago ezaugarri horietan oinarrituta bertsoaren egitura narratiboa antzematerik? Diskurtsoaren koherentzia edo barne-ordenazioa ikasterik ba ote?

Ikerketa-ildo hori landu dugu atal honetan. Helburu horretara bidean, Ikasketa Automatikoko tresnak baliatu ditugu.

9.1 Bertsoen egitura diskurtsiboaren bila

Bat-bateko ekoizpenean bertsoak antolatzeko baliatzen den egitura orokorrik ba ote den antzematen saiatu gara. 1986-2009 arteko Bertsolari Txapelketa Nagusiko bertsoak geneuzkan eskura, eta lanen zabala kontuan hartuta, agur-bertsoetara mugatu dugu gure azterketa. Izan ere, agur-bertsoekin hasi ohi dute bertsolariek saioa, eta baita amaitu ere. Inork gai edo bertso-molde jakinera mugatu gabe, libre aritzen dira eta bertsolariak nahi duena zuzenean adierazteko modua izaten du bertan. Horregatik iruditu zaizkigu agur-bertsoak egokiak bertsolarien narrazio egitura aztertzeko.

Agur-bertsoak aztergai hartuta beraz, bi urratsetan bereizi dugu egitekoa:

1. Alorreko adituen laguntzaz agurren azterketa egin dugu. Agur-bertsoetan ohiko ezaugarriak erauzi, bildu eta bateratuz. Agur-bertsoak puntuz-puntu aztertu eta ondoko sei kategoria bereizi ditugu: *Mezua*, *Lekua*, *Publikoa*, *Saioa*, *Norbera* eta *Betelana*.

2. Behin ezaugarri edo kategoriak zehaztuta, Testu Kategorizazio (*Text Classification*) gisa planteatu da ataza (Sebastiani, 2002), kategoria horien arabera bertsoetako puntuak sailkatuz.

Beraz, ikerketa-eremu honetako helburua bikoitza izan da: batetik, agur-bertsoak aztertu eta ezaugarri narratibo komunak topatzea; bestetik berriz, ezaugarri horiek IA bidez ikasi edo eskura daitezkeen egiaztatzea; eta, proposatutako metodoaren irismena balioestea.

Erabilitako corpusari dagokionez, txapelketako agur-bertsoak aukeratu ditugu. Bertatik 40 bertso bildu, eta unitatetzat puntua genuenez, 212 instantziatako corpusa osatu dugu.

Esperimentuak iturburu irekiko WEKA tresna erabilita gauzatu ditugu (Hall et al., 2009). Hauek dira erabilitako algoritmoak: Nearest Neighbour Classifier (IBk) (Dasarathy, 1991), Naive Bayes Classifier (NB) (Minsky, 1961), J48 Decision Tree Learner (Quinlan, 1993), Bayes Net eta SMO Support Vector Machine (Joachims, 1998).

Corpusaren dimentsionaltasuna murrizteko bi teknika aplikatu ditugu: batetik, sailkatze atazan lagungarri ez diren eta bereizle lanik betetzen ez dituzten atributuak detektatu eta ezabatu nahi izan ditugu (Forman, 2003): Information Gain, Chi-square eta Gain Ratio (Zipitria et al., 2012) metodoak baliatu ditugu horretarako. Bestetik, ezaugarri eraldaketa aplikatu dugu, atributuen jatorrizko zerrenda egokitu eta trinkoagoa den berri bat lortuz. Principal Component Analysis (PCA) (Wold et al., 1987) eta Latent Semantic Analysis (LSA) (Deerwester et al., 1990) (Hofmann, 2001) metodoak erabili ditugu helburu horrekin.

9.1 irudiak erakusten du kategorizazio edo sailkapen atazaren prozesu osoa.

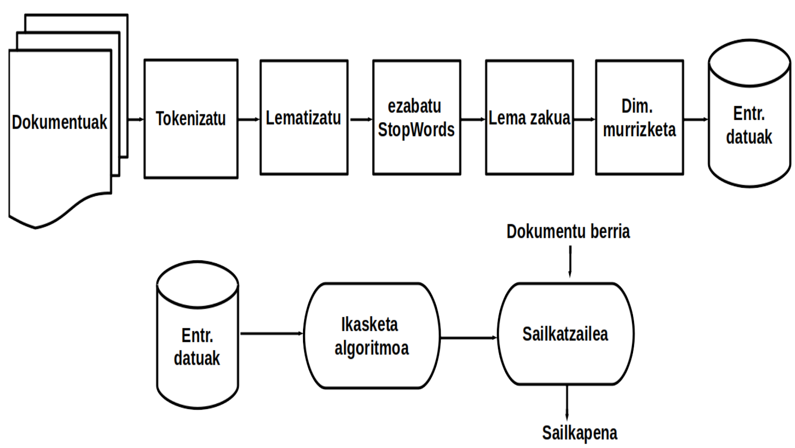
9.1 taulak jasotzen ditu ezaugarri bakoitzarekin lortutako emaitzarik onenak.

Eremu honetako ekarpenik nabarmenena, bertsoen ezaugarri diskurtsiboak Ikasketa Automatikoko tresnen bidez atzemateko ahalegina litzateke. Arloan aditu direnen esperientzia baliatuz bertso-sorta etiketatu eta, ondoren, esperientzia hori orokortu eta kasu berrietan modu

Kategoria	Algoritmoa	Atributu aukeraketa	Errendimendua	F-measure
Mezua	1-nn	None	64.62%	0.62
Lekua	SMO	InfoGain	89.62%	0.86
Publikoa	SMO	ChiSquare	83.01%	0.81
Saioa	5-nn	None	78.30%	0.76
Norbera	SMO	InfoGain	62.26%	0.60
Betelana	1-nn	GainRatio	87.74%	0.83

9.1 Taula: Kategoria bakoitzeko emaitza onenak

automatikoan aplikatu ahal izateko. Aipagarria da baita sailkapen-prozesua bestelako interbentziorik gabe burutzen dela, erregela linguistikorik eta ezaugarrien eskuzko aukeraketarik gabe.



9.1 Irudia: Testu kategorizazio atazaren prozesu osoa

Azaldutakoa ondoko artikuluak jasotzen du:

- **Towards Basque Oral Poetry Analysis: A Machine Learning Approach**. M. Osinalde, A. Astigarraga, I. Rodriguez, M. Agirrezabal. International Conference on Recent Advances in Natural Language Processing (RANLP), 2013ko iraila, Hissar, Bulgaria.

9.2 Koherentziaren azterketa. Hurbilketa sailkatzailea

Aurreko atalean diskurtsoaren ezaugarriak aztertu baditugu, oraingo honetan diskurtsoaren antolaketan jarriko dugu fokua. Diskurtsoaren unitate gisa bertsoa harturik, bertsoaren ordenazioa aztertuko dugu hain zuzen ere.

Esaldi ugariz osatutako testuak sortzerakoan, beharrezkoa da esaldiak ordena egokian kokatzea testu osoa behar bezala ulertuko bada. Honela, esaldien ordenatze-ataza, arlo ezberdinetan agertzen da, Galdera-Erantzun sistemetan eta iturburu anitzeko Laburpengintza Automatikoa arloetan besteak beste (Barzilay and Elhadad, 2002) (Hayashi et al., 2013). Sistema horiek iturri ezberdinetatik informazioa erauzi eta konbinatzen dute testu koherentea osatzeko helburuarekin. Koherentzia berebizikoa da testu-sorkuntza sistemetan. Bi eratako koherentzia bereizi genezake: koherentzia lokala, ondoz-ondoko esaldien artekoa eta globala, testuaren diskurtso-mailakoa (Asher and Lascarides, 2003) (Hobbs, 1979) (Polanyi et al., 2003).

Ikerlan honetan, esaldien ordenatze-ataza bertsoaren alorrera ekarri dugu. Gure asmoa, bertsolari profesionalek beren testua nola antolatzen duten ikastea da, bertso berriak sortzerakoan erabili ahal izateko. Ataza honela zehaztu dugu: zortziko txikiko bertso-sail batetik abiatuta, Ikasketa Automatikoko tresnak baliatuz, bertso horietako puntuen ordenazio egokia ikasten ahalegindu gara. Hau da, bertsoa (4 puntuz osatua) eta bere permutazio guztiak izanda ($4! = 24$), haien arteko ordenamendu egokia ikasteko ahalegina burutu dugu.

Gure ahaleginak, koherentzia lokal eta globala, biak uztartzen ditu. Koherentzia lokala, ondoz ondoko esaldien artekoa neurtzeko LSA (Landauer et al., 2013) (Zelaia, 2016) eta Word2vec (Mikolov et al., 2013) metodoak baliatu ditugu. Diskurtso-mailako koherentzia globalerako berriz, gainbegiratutako ikasketa algoritmoak baliatu ditugu. *Classifier Subset Selection for Stacked Generalization* (*CSS Stacking*) izeneko multi-sailkatzailea hain zuzen ere (Kotsiantis, 2007). Beraz, esaldien ordenazio ataza sailkatze ataza gisa planteatu dugu. Bertsoa esaldi edo puntuen zaku gisa adierazi eta algorimoaren helburua, puntuen arteko ordenazio egokia topatzea izanik. Sailkatze-ataza gisa planteatuta, bertsoak ondoko hiru klaseetako batean sailka ditzan eraiki dugu sistema:

- Bertsoaren ordenazioa egokia (4 puntuak ondo)

- Ordenazioak ez du bat egiten originalarekin baina azken puntua ondo

- Ordenazio ez-egokia (gainerako guztiak)

9.2 irudiak erakusten du ordenazio atazaren adibidea. Alde batean bertsoaren puntuak desordenatuta dauzkagu eta ondoan bertsolariak emandako ordenan.

Aspaldi dauka tripa	*Zer eskatzen diozu*
bapo eta lori	*zuk Olentzerori?*
Oraindik bizi al da	*Oraindik bizi al da*
gizon gizen hori?	*gizon gizen hori?*
nik ez diot eskatzen	*Aspaldi dauka tripa*
kanpoko inori.	*bapo eta lori*
Zer eskatzen diozu	*nik ez diot eskatzen*
zuk Olentzerori?	*kanpoko inori.*

9.2 Irudia: Maialen Lujanbioren bertsoa. Ezkerrean puntuak desordenatuta eta eskubian kantatutako ordenan

Ikerketa lan honen ekarpenik esanguratsuena, esaldien ordenazio-ataza poesiaren alorrera ekarri izana da. Guk dakigula behintzat, alor honetan egiten den lehenbiziko saiakera da gurea. Horrez gain, erabilitako algoritmoek emaitza aski onak erakutsi dizkigute, urratutako bidearen egokitasuna azpimarratzen dutenak.

Azaldutako ikerketa-lana ondoko artikuluak jasotzen du:

- **A Machine Learning Approach to Poem-line Ordering for Automatic Poetry Generation**. A. Astigarraga, J.M. Martínez-Otzeta, B. Sierra, I. Rodriguez, E. Lazkano. IEEE Transactions on Knowledge and Data Engineering (**BIDALITA**)

Azkenik, bigarren-mailako ekarpen gisa, azken urteetako Bertsolari Txapelketa Nagusien (1986-2009 artekoak) analisia egiten duen lana aipatuko dugu. Bertan hainbat mailatan egin dugu azterketa, betiere bertsoaren ezaugarri nagusiak kontuan izanda: errimak, neurriak, doinuak, hitzak, kategoria morfosintaktikoak eta euskara batuaren erabilera. Azterketa-lana ondoko artikuluak biltzen du:

- **Bota bertsoa, eta guk aztertuko dugu. Azken urteetako Bertsolari Txapelketa Nagusien analisia**. M. Agirrezabal, B. Arrieta, A. Astigarraga eta M. Hulden. Elhuyar aldizkaria, 2013.

Towards Basque Oral Poetry Analysis: A Machine Learning Approach

Mikel Osinalde, Aitzol Astigarraga, Igor Rodriguez and **Manex Agirrezabal**
Computer Science and Artificial Intelligence Department,
University of the Basque Country (UPV/EHU), 20018 Donostia
teagenes@hotmail.com
aitzol.astigarraga@ehu.es
igor.rodriguez@ehu.es
manex.agirrezabal@ehu.es

Abstract

This work aims to study the narrative structure of Basque greeting verses from a text classification approach. We propose a set of thematic categories for the correct classification of verses, and then, use those categories to analyse the verses based on Machine Learning techniques. Classification methods such as Naive Bayes, k-NN, Support Vector Machines and Decision Tree Learner have been selected. Dimensionality reduction techniques have been applied in order to reduce the term space. The results shown by the experiments give an indication of the suitability of the proposed approach for the task at hands.

1 Introduction

Automated text categorization, the assignment of text documents to one or more predefined categories according to their content, is an important application and research topic due to the amount of text documents that we have to deal with every day. The predominant approach to this problem is based on Machine Learning (ML) methods, where classifiers learn automatically the characteristics of the categories from a set of previously classified texts (Sebastiani, 2002).

The task of constructing a document classifier does not differ so much from other ML tasks, and a number of approaches have been proposed in the literature. According to Cardoso-Cachopo and Oliveira (2003) , they mainly differ on how documents are represented and how each document is assigned to the correct categories. Thus, both steps, document representation and selection of the classification method are crucial for the overall success. A particular approach can be more suitable for a particular task, with a specific data, while another one can be better in a different scenario (Zelaia et al., 2005; Kim et al., 2002; Joachims, 1998).

In this paper we analyse the categorization of traditional Basque impromptu greeting verses. The goal of our research is twofold: on the one hand, we want to extract the narrative structure of an improvised Basque verse; and, on the other hand, we want to study to what extent such an analysis can be addressed through learning algorithms.

The work presented in this article is organized as follows: first we introduce Basque language and *Bertsolaritza*, Basque improvised context poetry, for a better insight of the task at hand. Next, we give a general review of computational pragmatics and text classification domains, examining discourse pattern, document representation, feature reduction and classification algorithms. Afterwards, the experimental set-up is introduced in detail; and, in the next section, experimental results are shown and discussed. Finally, we present some conclusions and guidelines for future work.

2 Some Words about Basque Language and *Bertsolaritza*

Basque, *euskara*, is the language of the inhabitants of the Basque Country. It has a speech community of about 700,000 people, around 25% of the total population. Seven provinces compose the territory, four of them inside the Spanish state and three inside the French state.

Bertsolaritza, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive. Events and competitions in which improvised verses, *bertso*-s, are composed are very common. In such performances, one or more verse-makers, named *bertsolaris*, produce impromptu compositions about topics or prompts which are given to them by a theme-prompter. Then, the verse-

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 119–125,
Hissar, Bulgaria, 9-11 September 2013.

maker takes a few seconds, usually less than a minute, to compose a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme. Melodies are chosen from among hundreds of tunes.

Figure 1: *Bertsolari Txapelketa Nagusia*, the national championship of the Basque improvised contest poetry, held in 2009

When constructing an improvised verse strict constraints of meter and rhyme must be followed. For example, in the case of a metric structure of verses known as *Zortziko Txikia* (small of eight), the poem must have eight lines. The union of each odd line with the next even line, form a strophe. And each strophe, in turn, must rhyme with the others. But the true quality of the *bertso* does not only depend on those demanding technical requirements. The real value of the *bertso* resides on its dialectical, rhetorical and poetical value. Thus, a *bertsolari* must be able to express a variety of ideas and thoughts in an original way while dealing with the mentioned technical constraints.

The most demanding performance of Basque oral poetry, is the *Bertsolari Txapelketa*, the national championship of *bertsolaritza*, celebrated every four years (see Fig.1). The championship is composed by several tasks or contests of different nature that need to be fulfilled by the participants. It always begins with extemporaneous improvisations of greetings, a first verse called *Agurra*. This verse is the only one in which the poet can express directly what she/he wants. For the rest of the contest, the theme-prompter will prescribe a topic which serves as a prompt for the *bertso*, and also the verse metric and the number of iterations. For that reason, we thought the *Agurra* was of particular interest to analyse ways verse-makers use to structure their narration.

3 Related Work

3.1 Computational Pragmatics

As stated in the introduction, the aim of this paper is to notice if there is any discourse pattern in greeting verses. In other words, we are searching certain defined ways verse-improvisers in general use to structure their discourse.

If the study of the meaning is made taking into account the context, we will have more options for getting information of the factors surrounding improvisation (references, inferences, what improvisers are saying, thinking, self-state, context). The field that studies the ways in which context contributes to meaning is called pragmatics. From a general perspective, Pragmatics refers to the speaker and the environment (Searle, 1969; Austin, 1975; Vidal, 2004).

The study of extra-linguistic information searched by pragmatics is essential for a complete understanding of an improvised verse. In fact, the understanding of the text of each paragraph does not give us the key for the overall meaning of the verse. There is also a particular world's vision and a frame of reference shared with the public; and, indeed, we have been looking for those keys. We believe that the verse texts are not linear sequences of sentences, they are placed regarding a criterion and the research presented here aims to detect this intent.

Therefore, searching for the discourse facts in greeting verses led us to study their references.

3.2 Text Categorization

The goal of text categorization methods is to associate one or more of a predefined set of categories to a given document. An excellent review of text classification domain can be found in (Sebastiani, 2002).

It is widely accepted that how documents are represented influences the overall quality of the classification results (Leopold and Kindermann, 2002). Usually, each document is represented by an array of words. The set of all words of the training documents is called vocabulary, or dictionary. Thus, each document can be represented as a vector with one component corresponding to each term in the vocabulary, along with the number that represents how many times the word appears in the document (zero value if the term does not occur). This document representation is called the bag-of-words model. The major drawback of this text representation model is that the number of features in the corpus can be considerable, and thus, intractable for some learning algorithms.

Therefore, methods for dimension reduction are required. There exists two different ways to carry out this reduction: data can be pre-processed, i.e., some filters can be applied to control the size of the system's vocabulary. And, on the other hand, dimensionality reduction techniques can be applied.

3.2.1 Pre-processing the Data

We represented the documents based on the aforementioned bag-of-word model. But not all the words that appear in a document are significant for text classification task. Normally, a pre-processing step is required to reduce the dimensionality of the corpus and, also, to unify the data in a way it improves performance.

In this work, we applied the following pre-processing filters:

- **Stemming**: remove words with the same stem, keeping the most common among them. Due to its inflectional morphology, in Basque language a given word lemma makes many different word forms. A brief morphological description of Basque can be found in (Alegria et al., 1996). For example, the lemma *etxe* (house) forms the inflections *etxea* (the house), *etxeak* (houses or the houses), *etxeari* (to the house), etc. This means that if we use the exact given word to calculate term weighting, we will loose the similarities between all the inflections of that word. Therefore, we use a stemmer, which is based on the morphological description of Basque to find and use the lemmas of the given words in the term dictionary (Ezeiza et al., 1998).

- **Stopwords**: eliminate non-relevant words, such as articles, conjunctions and auxiliary verbs. A list containing the most frecuent words used in Basque poetry has been used to create the stopword list.

3.2.2 Dimensionality Reduction

Dimensionality reduction is a usual step in many text classification problems, that involves transforming the actual set of attributes into a shorter, and hopefully, more predictive one. There exists two ways to reduce dimensionality:

- **Feature selection** is used to reduce the dimensionality of the corpus removing features that are considered non-relevant for the classification task (Forman, 2003). The most well-known methods include: Information Gain, Chi-square and Gain Ratio (Zipitria et al., 2012).

- **Feature transformation** maps the original list of attributes onto a new, more compact one. Two well-known methods for feature transformation are: Principal Component Analysis (PCA) (Wold et al., 1987) and Latent Semantic Analysis (LSA) (Deerwester et al., 1990; Hofmann, 2001).

The major difference between both approaches is that feature selection selects a subset from the original set of attributes, and feature transformation transforms them into new ones. The latter can affect our ability to understand the results, as transformed attributes can show good performance but little meaningful information.

3.2.3 Learning Algorithms

Once the text is properly represented, ML algorithms can be applied. Many text classifiers have been proposed and tested in literature using ML techniques (Sebastiani, 2002), but text categorization is still an active area of research, mainly because there is not a general faultless approach.

For the work presented here, we used the following algorithms: Nearest Neighbour Classifier (IBk) (Dasarathy, 1991), Nave Bayes Classifier (NB) (Minsky, 1961), J48 Decision Tree Learner (Hall et al., 2009) and SMO Support Vector Machine (Joachims, 1998).

All the experiments were performed using the Weka open-source implementation (Hall et al., 2009). Weka is written in Java and is freely available from its website [1].

In Fig.2, the graphical representation of the overall Text Classification process is shown.

4 Experimental Setup

The aim of this section is to describe the document collection used in our experiments and to give an account of the stemming, stopword deletion and dimensionality reduction techniques we have applied.

[1] http://www.cs.waikato.ac.nz/ml/weka/

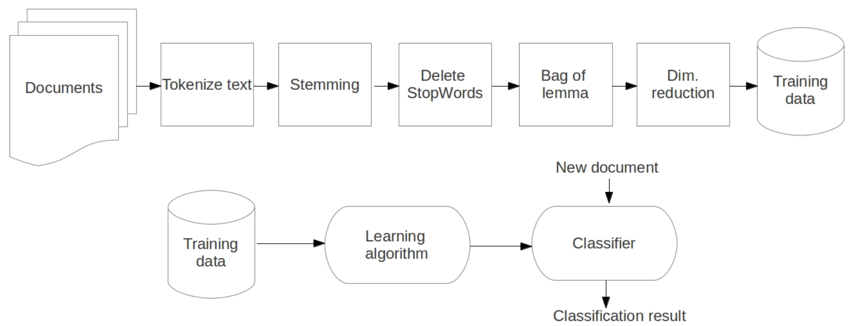


Figure 2: The overall process of text categorization

4.1 Categorization

To make a correct categorization of the verses, before anything else the unit to be studied needs to be decided. We could take as a unit of study the word, the strophes or the entire verses. Considering that we want to extract the structure that would provide information about the decisions made by the improviser and the discourse organization, we decided that the strophe[2] was the most appropriate unit to observe those ideas. Therefore, the first job was to divide the verses in strophes. After that, we began to identify the contents and features in them. The goal was to make the widest possible characterization and, at the same time, select the most accurate list of attributes that would make the strophes as much distinguishable as possible.

We sampled some strophes from the verse corpus described in section 4.2 and analysed them one by one. We had two options when categorizing the strophes: first, analyse and group all the perceived topics, allowing us to propose a realistic classification of the strophes from any verse. And second, make a hypothesis and adjust the obtained data to the hypothesis. We decided to take both paths.

After analysing each of the strophes and extracting their topics, we made the final list, sorted by the relevance of the categories. We obtained a very large list of contents and we arranged it by the importance and by the number of appearance. But that thick list did not help us in our mission as we wanted. So we agreed to try to define and limit the collection of attributes. And we decided to use the second option. Therefore, we studied the foundations of discourse analysis (Roberts and Ross, 2010; Gumperz, 1982), and the classifications proposed by critics of the improvisation field (Egaña et al., 2004; Diaz Pimienta, 2001); and then, we compared them with our predicted one. Merging both approaches we tried to build a strong set of categories.

Combining inductive and deductive paths we formed a list of six categories. So the initial big list that we gathered was filtered to a more selective classification. Therewith, we found possible to label the majority of the strophes in the analysed verses, and also get a significant level of accuracy.

Thus, these are the categories to be considered in the verse classification step:

1. Message: the main idea
2. Location: references to the event site
3. Public: messages and references relating to the audience
4. Event: messages and references relating to the performance itself
5. Oneself aim or Oneself state
6. Miscellaneous: padding, junk. Sentences with no specific meaning or intend.

As well as the five categories closely linked to the communication situation, there is another that we called Miscellaneous (padding, filling). Due to

[2] a pair of stanzas of alternating form on which the structure of a given poem is based

the demanding nature of the improvisation performances, they usually are sentences not very full of content and intent.

We have decided to consider each one of them as a separate goal, and hence six classifiers were to be obtained, one for each category. Thus, each categorization task was addressed as a binary classification problem, in which each document must be classified as being part of $category_i$ or not (for example, Location vs. no Location).

4.2 Document Collection

For the task in hands, we decided to limit our essay to greeting verses from tournaments. We selected 40 verses of a corpus of 2002 verses and divided them into strophes (212 in total). But when we began assigning categories (1-6) to each strophe, we realized we were in blurred fields. It was pretty difficult to perform that task accurately and we thought it was necessary to ask some expert for help. Mikel Aizpurua[3] and Karlos Aizpurua[4] (a well-known judge the former and verse improviser and Basque poetry researcher the latter) agreed to participate in our research, and they manually labelled one by one the 212 strophes.

In that study, we considered each binary class decision as a distinct classification task, where each document was tested as belonging or not to each category. Thus, the same sentence could effectively belong to more than one categories (1 to 6 category labels could be assigned to the same sentence).

As an example, let us have a look to an initial greeting verse composed by Anjel Larrañaga, a famous verse-maker (see Fig.3).

There we can see that each strophe (composed of two lines), was labelled in one, two or even tree different categories.

- (1) (3): Message, Public
- (5): Oneself aim
- (4) (5): Event, Oneself state
- (1) (5) (3): Message, Oneself aim, Public

The document categorization process was accomplished in two steps: during the training step, a general inductive process automatically built a classifier by learning from a set of labelled documents. And during the test step, the performance of the classifier was measured. Due to the small size of our manually categorized corpus, we used the k-fold cross-validation method, with a fold value of k=10.

> *Agur ta erdi bertsozaleak*
> *lehendabiziko **sarreran**,*
> *behin da berriro jarri gerade*
> *kantatutzeko **aukeran**,*
> *ordu ilunak izanagaitik*
> *txapelketan gora-**beheran**,*
> *saia nahi degu ta ia zuen*
> *gogoko izaten **geran**.*

As a first introduction,
greetings to all improvisation fans. (1) (3)
Many times we were ready
to sing like now! (5)
Even though there are hard times
in our championship contest, (4) (5)
We will try to make our best
and we hope you find it to your liking! (1) (5) (3)

Figure 3: A welcome verse composed by Anjel Larrañaga

4.3 Pre-processing the Data

In order to reduce the dimensionality of the corpus, two pre-processing filters were applied. On the one hand, a stopword list was used to eliminate non-relevant words. On the other hand, a stemmer was used to reduce the number of attributes.

The number of different features in the unprocessed set of documents was 851, from which were extracted 614 different stems and 582 terms after eliminating the stopwords. So finally, we obtained a bag-of-lemmas with 582 different terms.

5 Experimental Results

In this section we show the results obtained in the experiments. There are various methods to determine algorithms' effectiveness, but precision and recall are the most frequently used ones.

It must be said that a number of studies on feature selection focused on performance. But in many cases, as happened to us, the are few in-

[3] http://bdb.bertsozale.com/en/web/haitzondo/view/-Mikel-Aizpurua

[4] http://bdb.bertsozale.com/en/web/haitzondo/view/-Karlos-Aizpurua

Category	ML method	Attribute selection	Performance	F-measure
Message	1-nn	None	64.62%	0.62
Location	SMO	InfoGain	89.62%	0.86
Public	SMO	ChiSquare	83.01%	0.81
Event	5-nn	None	78.30%	0.76
Oneself	SMO	InfoGain	62.26%	0.60
Miscellaneous	1-nn	GainRatio	87.74%	0.83

Table 1: Best results for each category

stances of positive classes in the testing database. This can mask the classifiers performance evaluation. For instance, in our testing database only 22 out of 212 instances correspond to class 2 ("Location"), giving an performance of 90.045 % to the algorithm that always classifies instances as 0, and thereby compressing the range of interesting values to the remaining 9.954 %. Therefore, in text categorization tasks is preferred the F-measure, the harmonic average between precision and recall.

Table1 shows the configurations that have achieved the best results for each category.

Based on the results of the table, we can state that they were good in three out of six categories (Location, Public and Miscellaneous); quite acceptable in one of them (Event); and finally, in the remaining two categories (Message and Oneself) the results were not very satisfactory.

Regarding to the learning algorithms, it should be pointed out that SMO and k-nn have shown the best results. We can state also that in most cases best accuracy rates have been obtained using dimensionality reduction techniques. Which in other words means that the selection of attributes is preferable to the raw data.

6 Conclusions and Future Work

In this paper we shown the foundations of the automated analysis of Basque impromptu greeting verses. The study proposes novel features of greeting-verses and analyses the suitability of those features in the task of automated feature classification. It is important to note that our primary goals were to establish the characteristics for the correct classification of the verses, and so to analyse their narrative structure. And, secondly, to validate different methods for categorizing Basque greeting verses.

Towards this end, we introduced different features related to improvised greeting verses and categorized them into six groups of Message, Location, Public, Event, Oneself and Miscellaneous. Then, we implemented six different approaches combining dimensionality reduction techniques and ML algorithms. One for each considered categories.

In our opinion, the most relevant conclusion is that k-nn and SMO have shown to be the most suitable algorithms for our classification task, and also, that in most cases attribute selection techniques help to improve their performance.

As a future work, we would like to assess the problem as a multi-labelling task (Zelaia et al., 2011), and see if that improves the results.

Finally, we must say that there is still much work to do in order to properly extract discourse-patterns from Basque greeting verses. To this end, we intend to use our classifiers to label larger corpora and find regular discourse patterns in them.

7 Acknowledgements

The authors gratefully acknowledge *Bertsozale Elkartea*[5] (Association of the Friends of Bertsolaritza), whose verse corpora has been used to test and develop the *Bertsobot* project.

This work has been partially conducted under the SAIOTEK projects XIMUR and POETAUTO, the Basque Government Research Team grant and the University of the Basque Country UPV/EHU, under grant UFI11/45 (BAILab).

References

Iñaki Alegria, Xabier Artola, Kepa Sarasola, and Miriam Urkia. 1996. Automatic morphological analysis of Basque. *Literary and Linguistic Computing*, 11(4):193–203.

John Langshaw Austin. 1975. *How to do things with words*, volume 88. Harvard University Press.

Ana Cardoso-Cachopo and Arlindo Oliveira. 2003. An empirical comparison of text categorization methods. In

[5]http://www.bertsozale.com/en

String Processing and Information Retrieval, pages 183–196. Springer.

Belur V Dasarathy. 1991. Nearest neighbor ({NN}) norms:{NN} pattern classification techniques.

Scott Deerwester, Susan T. Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Alexis Diaz Pimienta. 2001. *Teoría de la improvisación: primeras páginas para el estudio del repentismo*. Ediciones Unión.

Andoni Egaña, Alfonso Sastre, Arantza Mariskal, Alexis Diaz Pimienta, and Guillermo Velazquez. 2004. *Ahozko inprobisazioa munduan topaketak: Encuentro sobre la improvisación oral en el mundo : (Donostia, 2003-11-3/8).* Euskal Herriko Bertsozale Elkartea.

Nerea Ezeiza, Iñaki Alegria, José María Arriola, Rubén Urizar, and Itziar Aduriz. 1998. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 380–384. Association for Computational Linguistics.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305.

John J Gumperz. 1982. Discourse strategies: Studies in interactional sociolinguistics. *Cambridge University, Cambridge*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142.

Sang-Bum Kim, Hae-Chang Rim, Dongsuk Yook, and Heui-Seok Lim. 2002. Effective methods for improving naive bayes text classifiers. *PRICAI 2002: Trends in Artificial Intelligence*, pages 479–484.

Edda Leopold and Jörg Kindermann. 2002. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1):423–444.

Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

W Rhys Roberts and WD Ross. 2010. *Rhetoric*. Cosimo Classics.

John R Searle. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge university press.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March.

María Victoria Escandell Vidal. 2004. Aportaciones de la pragmática. *Vademécum para la formación de profesores. Enseñar español como segunda lengua (12) 1 lengua extranjera (LE)*, pages 179–197.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52.

Ana Zelaia, Iñaki Alegria, Olatz Arregi, and Basilio Sierra. 2005. Analyzing the effect of dimensionality reduction in document categorization for basque. *Archives of Control Sciences*, 600:202.

Ana Zelaia, Iñaki Alegria, Olatz Arregi, and Basilio Sierra. 2011. A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. *Applied Soft Computing*, 11(8):4981–4990.

Iraide Zipitria, Basilio Sierra, Ana Arruarte, and Jon A Elorriaga. 2012. Cohesion grading decisions in a summary evaluation environment: A machine learning approach.

A Machine Learning Approach to poem-line ordering for Automatic Poetry Generation

A. Astigarraga, J. M. Martínez-Otzeta, B. Sierra, I. Rodríguez, I. Irigoien and E. Lazkano

Abstract—Ordering information is an essential task within the field of Natural Language Generation. Changing the order of the information not only makes it difficult to understand, but also conveys an entirely different idea to the reader. Same can be said about poetry, in which the author seeks a particular logical and topical structure for the poem. This paper proposes a supervised Machine Learning approach for poem-line ordering for Basque language. In this paper, a supervised classification paradigm called Classifier Subset Selection for Stacked Generalization (CSS Stacking) is used to deal with the poetry verse ordering. The new approach consists of an improvement of a bi-level multi-classifier system known as Stacked Generalization by means of an integration of an Estimation of Distribution Algorithm (EDA) in the first layer to select the optimal subset from the standard base classifiers. These initial CSS Stacking classifiers are compared to other multi-classifier systems and the employed standard classifiers built on the same set of textual features. Experimental results indicate that the proposed method is well suited for the task at hands and thus could be one of the latest stages of a system creating automated poetry.

Index Terms—Natural Language Processing, Machine Learning, Automatic Poetry Generation

1 Introduction

In the domain of Natural Language Generation (NLG) [1] , when generating a text that consists of multiple sentences, it is necessary to arrange the sentences in an appropriate order so that the overall text can be understood correctly [2], [3]. Thus, the task of ordering sentences arises in many fields: Multi-document Summarization, Question and Answering and concept-to-text generation systems are some of them. These systems extract information from different sources and combine them to produce a coherent text.

Coherence is essential in any text-generating system. It can be divided in local coherence, which captures text relatedness at the level of sentence-to-sentence transitions; and global coherence, dealing at the discourse-level of the text.

Poems have a series of characteristics that differentiate them from prose: metric and rhyming constraints that condition the content or message, aesthetics, continued use of ellipsis, sometimes obfuscated and not clear language... Furthermore, poetic text is not always a precise task [4] and several rules of NLG are often broken [5], [6]. All of them add a series of extra challenges to the itself difficult task of ordering lines.

For instance, taken one example of the Limerick Poetry [1]:

Original
There once was a young lady named bright Whose speed was much faster than light She set out one day In a relative way And returned on the previous night.
Reordered
And returned on the previous night Whose speed was much faster than light She set out one day In a relative way There once was a young lady named bright

A limerick is a five-line poem with the rhyme scheme a-a-b-b-a, and is intended to be funny. In can be noticed the importance of having an appropriated order. In the above text the original ordered is shown. In the below text, only two verse orders have been exchanged: the first and the last one have been swap.

In this work, we transfer the task of ordering sentences to the field of Automatic Poetry Generation. Our long standing goal is to create an automatic Basque poem generator system. A poetic machine that is able to compose poems as Basque improvisers, called *bertsolaris*, do. Our motivation for this work is to learn how Basque professional poetry writers structure their text, in order to apply this knowledge when composing novel poems.

We are interested in the automated generation of poetry, having implemented approaches as the two described below.

- N-gram based method [7]: in this approach a stream of text has to be created from scratch, character by character or word by word, and establish a distribution of the resulting text into poem lines by some additional procedure. N-grams assign probabilities to sequences of words and the generated model can be used to stochastically generate sequences of words based on the generated distributions [8], [9] and [10].
- Corpus based method [11]: the computer is used to harvest and reuse text already formated into poem-like structure of lines. the objective is to extract and select existing lines to compose new poems. The basics of this method is to extract sentences from the corpus that meet rhyme and metric constraints. Several works can be found in the literature that use also this approach, as [12], [13], [14].

The main problem that arouses from the above mentioned methods is that the overall coherence of the text is not guaranteed. Thus, a method to learn the correct ordering of the created verses is needed in order to achieve a meaningful and coherent discourse.

Our aim is to create an approach that integrates both local and global coherence. Local, sentence-to-sentence level, coher-

1. http://www.webexhibits.org/poetry/explore_famous_limerick_examples.html

ence is achieved using the methods described above. For global, discourse-level, coherence a supervised classification paradigm called Classifier Subset Selection for Stacked Generalization (CSS Stacking) is presented. Thus, the sentence ordering problem is instantiated as a classification problem. A poem is viewed as a bag of sentences and the algorithm's task is to try to find the correct ordering which maximizes coherence.

The rest of the paper is structured as it follows. Section 2 introduces related work. Section 3 explains some characteristics of the Basque language and the Basque oral poetry or *Bertsolaritza*. In Section 4 the data used in the experiments is described, while in section 5 the proposed method is fully explained. Section 6 describes how the experiments that have been carried out were performed, specifying which techniques have been used in each step of the process, as well the obtained results and discussion. Section 7 concludes the paper and presents conclusions and future work.

2 Related Work

As mentioned in the Introduction, information ordering is an essential task within NLG. Information can be ordered word by word or sentence by sentence. Finding the original word order of a shuffled sentence is treated in word ordering task [15], [16]. And, on the other hand, sentence ordering task aims to arrange a set of sentences into a coherent text in a clear and consistent manner [2], [3]. Sentence ordering can refer to the task of reordering a set of sentences that come from the same text; or, on the other hand, can refer also to the task of finding an optimal order to sentences that come from different sources.

Sentence ordering is usually discussed in literature in the contexts of multi-document summarization and concept-to-text generation tasks. In those tasks, sentences come from different sources and factors other than coherence are also considered, as sentence position in the source document and creation time [17], [18].

Linguistic theories of discourse comprehension refer to both local and global coherence of a text. Local coherence is defined as the process of determining the manner by which two consecutive discourse segments relate to one another [19], [20], [21], [22], [23], [24]. In contrast, global coherence refers to the general discourse of the text [25], [26], [27], [28], [29].

Psychological evidences show that coherence on both levels is manifested in text comprehension [30].

Both, sentence ordering and coherence modeling are closely related and have been approached by similar techniques. Most approaches propose a measure of coherence and formulate the ordering problem as finding an order with maximal coherence [31], [32].

We will mention some of the most relevant works related to the sentence ordering task.

Barzilay [17] proposed an improved version of chronological ordering by combining topic relatedness and chronological ordering to order sentences. In this approach, sentences were grouped into different themes extracted from the source documents and then ordered in each group chronologically.

Lapata [33] presented a probabilistic method for text structuring and its application in sentence ordering. Sentences, represented by vectors of linguistic features, were ordered based on conditional probabilities of sentence pairs, learned from a training corpus of domain specific corpus. Once the conditional probability of sentence pairs was computed, the global ordering was achieved with a greedy algorithm.

Bollegala [34], [18] combined four different ordering criteria in the same system: chronological ordering, probabilistic precedence and succession ordering and topic relatedness ordering. Then they learnt the weighted linear combination of those criteria using a hedge regression algorithm. Finally a greedy algorithm was used to search the optimal ordering of sentences.

Barzilay presented the Entity-Grid model [3], a popular model of coherence which captures local coherence by modeling patterns of entity distributions in the discourse. Sentences were represented by the syntactic roles of entities appearing in the document and entity transition frequencies in successive sentences were treated as features that are are used to train a ranking SVM.

More recent works that combine both, local and global coherence for text structuring include: Li and Jurafsky [32], in which open-discourse coherence is examined, training discriminative models that treating natural text as coherent and permutations as non-coherent, and Markov generative models that can predict sentences given their neighbors. Logeswaran [31] presents a Recurrent Neural Network based model to organize coherently a set of sentences. The proposed neural approach it is based on the model proposed by Vinyals [35].

Textual coherence can be assessed also by psychological models of discourse. In [36], [37] the authors use Latent Semantic Analysis (LSA) embeddings to generalize lexical cohesion. Although LSA has shown ability to model local and global discourse characteristics, as far as we know, it has not been applied to the task of sentence ordering. Nevertheless other word embedding methods have been developed and used in computational semantics. Below is a brief description of the two methods that we employ.

- LSA [38] was developed as a special Vector Space Model (VSM) [39] approach. In the standard VSM method , term associations are ignored, and therefore, synonymy and polysemy are not captured. By contrast, LSA tries to find the latent semantic structure of a document collection. That is, to find hidden relations between terms, sentences or other text units [36]. In other words, LSA analyzes term co-occurrence of higher orders, so that it is able to incorporate the relationship of words A, B which only co-occur in documents through word C, and never appear in the same document together directly. LSA involves the application of the matrix algebra method Singular Value Decomposition (SVD) to the document-by-term matrix in order to reduce its rank and construct a semantic space. The matrix is then decomposed in such a way that similar vectors will be conflated, or moved closer within the space. And thus, LSA enables the calculation of semantic similarities between words, sentences or paragraphs.
- Word2vec is a technique for computing continuous vector representations of words from very large data sets [40]. It pertains to a more general class of techniques called word embeddings [41]. Once the words are projected in a multidimensional real-valued space, it is possible to compute Euclidean distances between them. In this work we want to compute distances between verses, and thus a suitable way of defining them is needed. We have chosen the following one: find the centroid of the words that compose each sentence, and define the distance between sentences as the

Euclidean distance between those centroids.

3 Basque Language and Bertsolaritza

Basque, *euskara*, is the language of the inhabitants of the Basque Country. It has a speech community of about 700,000 people, around 25% of the total population. Seven provinces compose the territory, four of them inside the Spanish state and three inside the French state.

Basque is a language isolate, not related to the Indo-European languages with which it shares its geographic boundaries. It is an agglutinative language with about seventeen cases, with no total agreement about the precise number. It is highly inflected. For example, with respect to the lexical and grammatical resources for expressing space, there are five different locational cases and over thirty postpositions, also inflected with these cases, that allow fine and detailed descriptions of space [42].

Bertsolaritza, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive. Events and competitions in which improvised verses, *bertso*-s, are composed are very common. In such performances, one or more verse-makers, named *bertsolaris*, produce impromptu compositions about topics or prompts which are given to them by a theme-prompter. Then, the verse-maker takes a few seconds, usually less than a minute, to compose a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme. Melodies are chosen from among hundreds of tunes.

When constructing an improvised verse strict constraints of meter and rhyme must be followed. For example, in the case of a metric structure of verses known as *Zortziko Txikia* (small of eight), the poem must have eight lines. The union of each odd line with the next even line, form a strophe. And each strophe, in turn, must rhyme with the others. But the true quality of the *bertso* does not only depend on those demanding technical requirements. The real value of the *bertso* resides on its dialectical, rhetorical and poetical value. Thus, a *bertsolari* must be able to express a variety of ideas and thoughts in an original way while dealing with the mentioned technical constraints. Figure 1 shows a picture of the 2009 national championship, an annual event with a high media profile.

Different languages and realities will set up different conditions. Foley explains this idea in the following proverb: "Oral poetry works like language, only more so" [43]. Oral poetry is not a fixed product, but it is part of a living language and is thus constantly changing. Foley states that "the major difference, in comparison to everyday language, is that the specialized registers of oral poetry are characterized by greater structure and more highly coded idiomatic meaning. Idiom is the 'more so' in this proverb." *Bertsolaritza* is then part of the Basque language, but oral poetic characteristics, such as formulae, add some extra value to the art.

Bertsolaritza belongs to oral poetry and has some inherent properties that must be taken into account:

On the other hand, it must be taken into account that the *bertsolari* improvises. Thus, (s)he never says what (s)he wants to say, but rather what is permitted by the meter and the rhyming words that (s)he has stored and can, at the opportune moment, retrieve. There are no *bertsolaris* who say exactly what they want to say at the same time as rhyming and using a set meter [44]. This means that the coherence of a *bertso* is not always neither good nor clear.

4 Experimental data

A selection of professional poems made by *bertsolaris* has been used in this paper as the ground truth representing the correct ordering of a set of verses. Along this approach, we have selected stanzas in which the order of the verses is important (change on the order implies nonsense stanzas) and thus try to detect the correct orders from all the permutations of the poem lines composing the whole stanza.

We have selected *Zortziko-txikia* as preferred stanza. It is composed by 4 verses of 13 syllables (7+6). As there are 4 verses on it, 24 (4!) different orders can be done. We take also into account which of the orders maintain the last verse as it is (6 of the 24 permutations, being one of them the original one). This is interesting because the *bertsolari* has a basic strategy that is used in a systematic way [45]: think up the end first. On hearing a proposed theme, the *bertsolari* turns on their mental machinery and starts thinking about what is going to say and the order in which (s)he is going to say it, keeping the main idea for the end. This justifies the importance of stressing the correct positioning of the last line.

When presented with a classification, a suitable representation of the instances is needed. Our task consists of classifying a given stanza as belonging to one of three classes:

1) Ordering coincident with the original one. As would be expected, this is considered the most correct ordering.
2) Ordering different from the original but with the fourth verse in the right position. These ordering are nor totally correct, but still maintain the most important verse in its right place.
3) Other ordering. These orderings are deemed as incorrect ones.

Each ordering (permutation of verses) is assigned one of these three classes. As a selection of 36 stanzas has been used, this gives a total of 864 permutations (36×24). As previously stated, the verses in a stanza are semantically related. Our aim is to find if some measure of semantic similarity between verses can help us to find the correct ordering. The features of the instances to be classified would be the distances between all the possible pairs of verses.

We have chosen two different similarity measures: Latent Semantic Analysis (LSA) and the Word2vec model of Word Embedding. To construct a semantic space to capture textual coherence, a large representative text corpus is needed. Thus, a large collection of documents collected from the Basque language newspaper *Egunkaria* was used for the construction of the semantic space. The text collection was divided into paragraphs, that lead to a corpus of 161,113 documents and 10,121,624 words. Again, the text collection was first lemmatized in order to group several inflected forms under a single lemma. Thus, a corpus of 180,616 unique words was obtained. Over this corpus semantic models have been created in order to be able to compute similarity measures or distances between two sentences. When using the Word2vec method, a suitable way of defining sentences (verses) was needed. We have chosen the following one: find the centroid of the words that compose each sentence, and define the distance between sentences as the Euclidean distance between those centroids. These values are the features fed to the classifier in charge of looking for the best ordering. Given that there are six possible pair combinations between the four verses of a stanza, each instance representing them will consist of 12 features, 6

Fig. 1: National Championship final event.

for the similarity between pairs according to LSA and 6 for the centroid distance according to Word2vec.

5 Proposed Method: Classifier Subset Selection within Stacked Generalization

It is worth mentioning that one of the most important question to be taken into account in a Machine Learning problem is the adequateness of the used features. We have used a set of 12 features to describe the data, and two Feature Selection approaches to compare with them, namely Principal component Analysis (PCA) and Latent Semantic Indexing[2] (LSI).

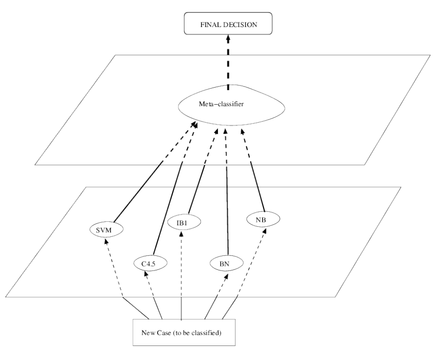


Fig. 2: Stacked Generalization schemata.

2. The same method is given two names: LSI and LSA. When it is used for the task of Information Retrieval or feature selection LSI is used and, when semantic similarity is measured, LSA is more common.

Stacked Generalization [46] is a bi-level classifier in which several single classifiers located in the first layer are combined by a so called meta-classifier, which is the one located in the second layer (a general idea is shown in Figure 2). To deal with the verse ordering problem a Classifier Subset Selection approach to improve a bi-level multi-classifier named Stacked Generalization have been used [47]. To the best of our knowledge, this is the first time that this kind of classifier has been applied to a Natural Language Processing task.

To reduce the number of classifiers to be used in the first layer an evolutionary algorithm called Estimation of Distribution Algorithms [48] is used. This approach is called Classifier Subset Selection (CSS) and a graphical example is illustrated in Figure 3.

6 Experiments

The experiments have been organized in four steps:

In the first step, single classifiers are applied to the verse ordering problem. WEKA software is used to this end [49]; used classifiers are well known standard ones.

Standard classifier combination approaches are used in the second step; Bagging, Boosting and ClassRegression are the used paradigms.

Stacked Generalization [46] is applied in the third step; it is a standard two-level classifier assembly as well (see Figure 2); it is indeed presented separately as the Classifier Subset selection process is done over its first layer.

The fourth step consists on applying a reduction technique to the number of classifiers to be used in the first layer final model [47]. This new approach is called Classifier Subset Selection (CSS) and a graphical example is illustrated in Figure 3.

In all the experiments 10-fold cross-validation [50] was applied to get a validated classification result (error Rate), and this value has been the criterion to define the fitness of an individual, inside the evolutionary algorithm.

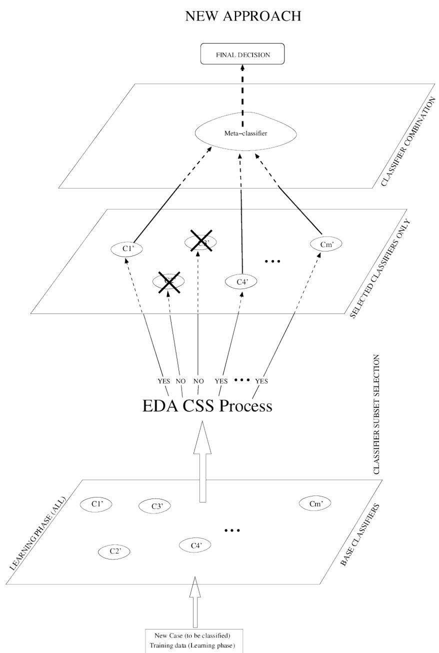


Fig. 3: Classifier Subset Selection Stacked Generalization.

6.1 Base Classifiers

The experiments of the first phase were carried out over 7 well-known Machine-Learning (ML) supervised classification algorithms. A brief description of the classifiers of the first phase is presented below.

Bayesian Networks (BN): A Bayesian network [51], belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independences via a directed acyclic graph (DAG).

C4.5: The C4.5 [52] represents a classification model by a decision tree. The tree is constructed in a top-down way, dividing the training set and beginning with the selection of the best variable in the root of the tree.

k-Nearest Neighbors (KNN): This algorithm is a case-based, Nearest-Neighbor classifier [53]. To classify a new test sample, a simple distance measure is used to find the training instance closest to the given test instance, and then it predicts the same class as this nearest training instance.

Naive Bayes (NB): The Naive-Bayes rule [54] uses the Bayes theorem to predict the class for each case, assuming that the predictive genes are independent given the category. To classify a new sample characterized by d genes $\mathbf{X} = (X_1, X_2, ..., X_d)$, the NB classifier applies the following rule:

$$c_{NB} = \arg\max_{c_j \in C} p(c_j) \prod_{i=1}^{d} p(x_i|c_j)$$

where c_{NB} denotes the class label predicted by the Naive-Bayes classifier and the possible classes of the problem are grouped in $C = \{c_1, \ldots, c_l\}$.

Random Forest (RandomF): constructs a combination of many unpruned decision trees [55]. The output class is the mode of the classes output by individual trees.

Support Vector Machines (SVM): are a set of related supervised learning methods used for classification and regression [56]. Viewing input data as two sets of vectors in an n-dimensional space, a SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets.

Multilayer Perceptron (MLP): A multilayer perceptron is a feed-forward artificial neural network model to map sets of input data onto a set of appropriate outputs [57]. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a processing element with a nonlinear activation function.

Radial Basis Function (RBF) network: A radial basis function network is an artificial neural network using radial basis functions as activation functions [58]. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

6.2 First Step: Single Classifiers

In this first step seven well known classifiers have been used to deal with the verse ordering problem. Obtained results (Error Rates) are shown in Table 1. As it can be seen, the best result is obtained by the Random Forest classifier (19.33% Error Rate) with the attributes selected by LSI. This attribute selection is the one which obtains the best mean (23.54% Error Rate).

TABLE 1: Single Classifier results

Single	New36	LSI	PCA
BN	25.00	25.00	25.00
NB	28.47	25.00	25.69
J48	30.44	24.77	27.89
KNN	33.33	21.18	30.21
SVM	25.00	25.00	25.00
MLP	26.16	24.54	24.77
RF	29.28	**19.33**	28.94
Mean	28.24	*23.54*	26.71

6.3 Second Step: Standard Multi-Classifiers

In this second Step three standard classifier combination paradigms are used: Bagging, Boosting and ClassRegression. Obtained results –Error Rates (ER)– are shown in Table 2; in this second step as well LSI attribute selector is the best one (19.71% ER in mean); the best result is obtained by Boosting (17.36% ER), which improves the best one obtained in the previous step.

TABLE 2: Obtained Results (Error Rates) with multi-classifier approaches

Multi	New36	LSI	PCA
Bagging	25.58	19.44	25.00
Boosting	26.97	**17.36**	26.27
ClassRegression	24.77	22.34	24.88
Mean	25.77	*19.71*	25.39

6.4 Third Step: Stacked Generalization

Stacked Generalization (SG) is used in this third step. As a Meta classifier has to be selected for the second layer, all the seven single classifier are used, hence seven SG paradigms are evaluated for each attribute set. Obtained results (Table 3) show an improvement with respect to the second step, both in mean (19.39%) and in individual result (16.78% using SVM as meta-classifier). Once again, LSI is the best attribute selection approach.

TABLE 3: Stacked Generalization results

Stacking	New36	LSI	PCA
BN	29.17	19.21	35.07
NB	38.19	20.72	32.75
J48	34.84	19.33	30.56
KNN	35.19	22.80	34.72
SVM	25.12	**16.78**	25.58
MLP	30.09	19.33	29.40
RF	27.31	17.59	26.50
Mean	31.42	*19.39*	30.65

6.5 Four Step: Classifier Subset Selection

Finally, a Classifier Subset selection is applied to each of the SG models of the previous step. As it can be seen in Table 4, all the results of the previous step are improved by the CSS approach. As a matter of fact, and looking to the LSI attribute selector, which again is the best one, five of the seven Meta-classifiers (BN, NB, J48, SVM and MLP) outperform the best result achieved in the third step; even the obtained average ER (16.58% is better than the precedent best result (17.58%).

The best result (15.39%) is obtained after applying a Classifier Subset Selection over the previous Stacked Generalization model which have MLP as Meta-classifier paradigm. LSI is the best attribute set.

TABLE 4: CSS

CSS	New36	LSI	PCA
BN	24.77	16.67	25.00
NB	25.00	16.09	16.20
J48	24.88	16.20	24.54
KNN	25.00	18.40	25.00
SVM	25.00	15.97	25.00
MLP	24.88	**15.39**	24.07
RF	26.62	17.36	25.00
Mean	25.17	*16.58*	23.54

TABLE 5: Number of correct guesses by user

Users	All verses right	Last verse right
User 1	1	3
User 2	0	2
User 3	2	3
User 4	1	2
User 5	1	2
User 6	0	2
User 7	4	4
User 8	4	4
User 9	3	3
User 10	1	2
Total	17	27
Accuracy	42.5%	67.5%

6.6 Obtained results analysis

Obtained results have offered a increasing accuracy measure among the four steps. Firstly, standard classifiers obtained a best result of 80.67% well classified cases (19.33% error rate). The best result is obtained by the Random Forest classification method, followed by the K-NN algorithm (21.18% Error Rate).

Obtained results are a little bit better with Standard Multi-Classifier models. Boosting paradigm gives a 82.64% accuracy result (17.36% Error Rate), followed by Bagging, with a 19.44% Error rate). This second best result is slightly worse that the one obtained by the best single classifier.

The third step consists on applying the stacked Generalization paradigm. To this end, all the single classifiers are used in the first layers of the multi-classifier, and an experiment is done using each of those single classifiers as the Meta-Classifier of the Stacked Generalization second layer. Obtained results improve the previous ones, being this time the best result that given by using SVM as Meta-Classifier; an accuracy of 83.22% is obtained (16.78% Error Rate), being the second best result 17.59% Error rate, when Random Forest is selected as the meta-Classifier.

The final step consists on applying a CSS approach to the Stacked Generalization paradigms, and now the best result is 84.61% well classified rate (15.39%) Error Rate), which has been obtained using MLP as Meta-Classifier. Obtained second (15.97%) ERR, third (16.09%), fourth (16.20%) and fifth (16.67% ERR) best results outperform the best obtained in the previous step (Figure 4).

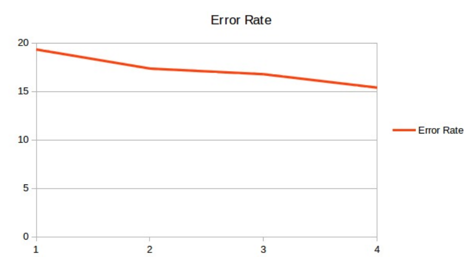


Fig. 4: Best ERR obtained in each step.

TABLE 6: Stanza in its original ordering (in Basque and its English translation)

Basque	zer eskatzen diozu zuk Olentzerori? oraindik bizi al da gizon gizen hori? aspaldi dauka tripa bapo eta lori nik ez diot eskatzen kanpoko inori
English	What do you ask Olentzero[3] for? is that fat man still alive? he has had a pot belly for a long time I do not ask to strangers

6.7 Human evaluation

Although, as it has been seen, sentence ordering is a challenging task for computer-based systems, it is supposed to be an easy task to solve for humans. Background knowledge and experience

3. Olentzero is a character in Basque Christmas tradition, similar to Father Christmas and Santa Claus.

TABLE 7: Reordered stanza (in Basque and its English translation)

Basque	aspaldi dauka tripa bapo eta lori oraindik bizi al da gizon gizen hori? nik ez diot eskatzen kanpoko inori zer eskatzen diozu zuk Olentzerori?
English	he has had a pot belly for a long time is that fat man still alive? I do not ask to strangers What do you ask Olentzero for?

are used by humans to decide the correct order among sentences. However, our intuition told us that ordering poems was not such an easy task. Taking into account that are impromptu verses, created under restrictions of metric and rhyme and with only 4 lines, we thought that the task at hands could be more complicated than we imagined.

Thus, an evaluation with humans was also conducted for further comparison to evaluate the performance of method. Ten people participated in this experiment: 3 of them were people close to verse-creation, another 3 people specialist in computational linguistics and the rest students and teachers of the Computer Science faculty. 4 poems were randomly selected and permuted from our testing dataset. And those 10 judges were asked to red the presented poems and reorder them correctly. The results show that the correct order of the whole poem was achieved in 42.5% of the cases and, last line of the poem was identified correctly in 67.5%. Detailed results are shown in Table 5. The results show that our intuition was not misguided and that the proposed task is difficult even for humans. Table 6 shows, as an example, an impromptu *bertso* created under the constraints of *Zortziko Txikia*. *Bertsoz* can be composed in a variety of manners and settings. The one that is shown corresponds to the exercise called "Point Given": the emcee or theme-prompter sings a *puntu* and the *bertsolari* has to complete it, staying within the given tune and meter. Table 7 shows the same *bertso* with random ordering.

7 Conclusions and Future Work

In this paper the verse ordering problem has been tackled. Sentence ordering is an important and unsolved problem, very closely related to Natural Language Generation. In the presented experimental setup, a set of Basque poems composed by four verses (*Zortziko Txikia*) was selected and all the permutations were calculated for each poem (4! = 24). Thus, the task was to decide the correct order of the verses. A Machine Learning approach has been used, based on a Stacked Generalization bi-level multi classifier, in which a selection of the classifiers to be used in the first layer is chosen by means of an evolutive algorithm called EDA.

Obtained results have been compared with those obtained by the single classifiers themselves (first step), with other standard multi-classifier paradigms (second step) and with the standard Stacked Generalization methods (third step). The CSS-SG approach obtains the best results among all others. Feature Subset selection has been used as well to improve the results.

The results have evolved increasingly within the experiments done. Results obtained by single classifiers are outperformed by multi-classifier models, the best of which is the Stacked Generalization; finally, a smarter version of SG is the one which obtains the best result.

Finding the optimal ordering is a difficult problem when a large number of sentences are required to be rearranged or when there is inherent ambiguity in the ordering of the sentences (as is our case). For a given set of N sentences, there are N! possible total orderings. As N grows, searching all the possible orderings to find the optimal could become computationally intractable. For this reason, the ordering problem is commonly formulated as a binary classification task: given a reference paragraph and a permuted version of it, the more coherently organized one needs to be identified [3]. Although the size of the poems used is relatively small, 4 verses or lines per poem, our approach is closer to the more harder task of open-domain full-paragraph sentence ordering. Therefore, we believe that the results obtained give a good indication of the suitability of the proposed approach.

As a conclusion, more efforts are envisaged to try to improve the results in the problem at hands; other approaches to deal with the Classifier Subset Selection problem, and the use of more sophisticated approaches such as deep learning paradigms are in the aims of the authors as next step in the research.

A future work of this research will be to perform new experiments on different corpora, and to extend the presented approach to other Natural Language Processing examples.

Acknowledgments

This research work was partially funded by the Spanish Ministry of Economy and Competitiveness under grant TIN2015-64395-R (MINECO/FEDER, UE) and by the Department of Education, Universities and Research of the Basque Government (grant IT900-16).

References

[1] E. Reiter and R. Dale, "Building applied natural language generation systems," *Natural Language Engineering*, vol. 3, no. 01, pp. 57–87, 1997.
[2] B. J. Grosz, S. Weinstein, and A. K. Joshi, "Centering: A framework for modeling the local coherence of discourse," *Computational linguistics*, vol. 21, no. 2, pp. 203–225, 1995.
[3] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach," *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008.
[4] P. Gervás, "Wasp: Evaluation of different strategies for the automatic generation of spanish verse," in *Proceedings of the AISB-00 symposium on creative & cultural aspects of AI*, 2000, pp. 93–100.
[5] H. Manurung, "An evolutionary algorithm approach to poetry generation," 2004.
[6] H. G. Oliveira, "Poetryme: a versatile platform for poetry generation," *Computational Creativity, Concept Invention, and General Intelligence*, vol. 1, p. 21, 2012.
[7] A. Astigarraga, J. M. Martínez-Otzeta, I. Rodriguez, B. Sierra, and E. Lazkano, "Markov text generator for basque poetry," to appear in Proceedings of TSD 2017: 20th International Conference on Text, Speech and Dialogue, Prague, Czechia, 2017.
[8] G. Barbieri, F. Pachet, P. Roy, and M. D. Esposti, "Markov constraints for generating lyrics with style," in *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 2012, pp. 115–120.
[9] A. Das and B. Gambäck, "Poetic machine: Computational creativity for automatic poetry generation in bengali," in *5th International Conference on Computational Creativity, ICCC*, 2014, pp. 230–238.
[10] P. Gervás, "Constrained creation of poetic forms during theme-driven exploration of a domain defined by an n-gram model," *Connection Science*, vol. 28, no. 2, pp. 111–130, 2016.
[11] A. Astigarraga, E. Jauregi, E. Lazkano, and M. Agirrezabal, "Textual coherence in a verse-maker robot," in *Human-Computer Systems Interaction: Backgrounds and Applications 3*. Springer, 2014, pp. 275–287.
[12] H. G. Oliveira and A. Cardoso, "Poetry generation with poetryme," in *Computational Creativity Research: Towards Creative Machines*. Springer, 2015, pp. 243–266.

[13] J. Toivanen, H. Toivonen, A. Valitutti, O. Gross *et al.*, "Corpus-based generation of content and form in poetry," in *Proceedings of the Third International Conference on Computational Creativity*, Dublin, Ireland, 2012, pp. 175–179.

[14] J. M. Toivanen, M. Järvisalo, H. Toivonen *et al.*, "Harnessing constraint programming for poetry composition," in *Proceedings of the Fourth International Conference on Computational Creativity*, Sydney, Australia, 2013, pp. 160–167.

[15] Y. Zhang and S. Clark, "Discriminative syntax-based word ordering for text generation," *Computational Linguistics*, vol. 41, no. 3, pp. 503–538, 2015.

[16] A. Schmaltz, A. M. Rush, and S. M. Shieber, "Word ordering without syntax," *arXiv preprint arXiv:1604.08633*, 2016.

[17] R. Barzilay and N. Elhadad, "Inferring strategies for sentence ordering in multidocument news summarization," *Journal of Artificial Intelligence Research*, vol. 17, pp. 35–55, 2002.

[18] D. Bollegala, N. Okazaki, and M. Ishizuka, "A preference learning approach to sentence ordering for multi-document summarization," *Information Sciences*, vol. 217, pp. 78–95, 2012.

[19] N. Asher and A. Lascarides, *Logics of conversation*. Cambridge University Press, 2003.

[20] J. R. Hobbs, "Coherence and coreference," *Cognitive science*, vol. 3, no. 1, pp. 67–90, 1979.

[21] A. Kehler and A. Kehler, *Coherence, reference, and the theory of grammar*. CSLI publications Stanford, CA, 2002.

[22] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," *Text-Interdisciplinary Journal for the Study of Discourse*, vol. 8, no. 3, pp. 243–281, 1988.

[23] L. Polanyi, M. Van Den Berg, and D. Ahn, "Discourse structure and sentential information structure. an initial proposal," *Journal of Logic, Language and Information*, vol. 12, no. 3, pp. 337–350, 2003.

[24] T. J. Sanders, W. P. Spooren, and L. G. Noordman, "Toward a taxonomy of coherence relations," *Discourse processes*, vol. 15, no. 1, pp. 1–35, 1992.

[25] W. Kintsch, "The role of knowledge in discourse comprehension: A construction-integration model." *Psychological review*, vol. 95, no. 2, p. 163, 1988.

[26] ——, *Comprehension: A paradigm for cognition*. Cambridge university press, 1998.

[27] P. Van den Broek, D. N. Rapp, and P. Kendeou, "Integrating memory-based and constructionist processes in accounts of reading comprehension," *Discourse Processes*, vol. 39, no. 2-3, pp. 299–316, 2005.

[28] P. Van den Broek, M. Young, Y. Tzeng, and T. Linderholm, "The landscape model of reading: Inferences and the online construction of a memory representation," *The construction of mental representations during reading*, pp. 71–98, 1999.

[29] R. A. Zwaan, *Aspects of literary comprehension: A cognitive approach*. John Benjamins Publishing, 1993, vol. 29.

[30] I. Tapiero, *Situation models and levels of coherence: Toward a definition of comprehension*. Taylor & Francis, 2007.

[31] L. Logeswaran, H. Lee, and D. Radev, "Sentence ordering using recurrent neural networks," *arXiv preprint arXiv:1611.02654*, 2016.

[32] J. Li and D. Jurafsky, "Neural net models for open-domain discourse coherence," *arXiv preprint arXiv:1606.01545*, 2016.

[33] M. Lapata, "Probabilistic text structuring: Experiments with sentence ordering," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 545–552.

[34] D. Bollegala, N. Okazaki, and M. Ishizuka, "A bottom-up approach to sentence ordering for multi-document summarization," *Information processing & management*, vol. 46, no. 1, pp. 89–109, 2010.

[35] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *arXiv preprint arXiv:1511.06391*, 2015.

[36] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[37] P. W. Foltz and T. Landauer, "Discourse coherence and lsa," *Handbook of latent semantic analysis*, pp. 167–184, 2007.

[38] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[39] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, pp. 141–188, 2010.

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[41] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.

[42] I. Ibarretxe-Antuñano, "An overview of basque locational cases: Old descriptions, new approaches," *International Computer Science Institute Technical Report*, no. 01-006, 2001.

[43] J. M. Foley, "Teaching oral traditions." Modern Language Association, 1998.

[44] J. Garzia, J. Sarasua, and A. Egaa, *The art of bertsolaritza: improvised basque verse singing*. Bertsolari liburuak, 2001.

[45] A. Egaña, "Oral tradition," vol. 22, 2007.

[46] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[47] I. Mendialdua, A. Arruti, E. Jauregi, E. Lazkano, and B. Sierra, "Classifier subset selection to construct multi-classifiers by means of estimation of distribution algorithms," *Neurocomputing*, vol. 157, pp. 46–60, 2015.

[48] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2002, vol. 2.

[49] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Fourth Edition*. Morgan Kaufmann, 2016.

[50] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 111–147, 1974.

[51] B. Sierra, E. Lazkano, E. Jauregi, and I. Irigoien, "Histogram distance-based bayesian network structure learning: A supervised classification specific approach," *Decision Support Systems*, vol. 48, no. 1, pp. 180–190, 2009.

[52] J. R. Quinlan, *C4.5: programs for machine learning*. Elsevier, 2014.

[53] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

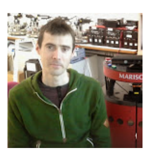
[54] B. Cestnik, "Estimating probabilities: a crucial task in machine learning." in *ECAI*, vol. 90, 1990, pp. 147–149.

[55] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[56] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1, pp. 169–186, 2003.

[57] F. Rosenblatt, *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan Books, 1961.

[58] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.



Aitzol Astigarraga received the B.Sc. Degree in Computer Science from the University of the Basque Country, Spain, in 2002, M. Sc. In Computational Engineering and Intelligent Systems (UPV/EHU) in 2007 and is pursuing the Ph.D. degree from the Department of Computer Science and Artificial Intelligence, University of the Basque Country. His current research interests include Automatic Poetry Generation, Human robot interaction and Machine Learning.

José María Martínez-Otzeta received his M.Sc. in Computer Sciences in 1993 and Ph.D. in Computer Sciences in 2008 at the University of the Basque Country. He has been a researcher at the Smart and Autonomous System Unit in IK4-Tekniker and currently is with the Robotics and Autonomous Systems Group at the University of the Basque Country. He has participated in several European and national projects dealing with robotics and machine learning. His research interests include robotics, machine learning, artificial intelligence and artificial vision.

Basilio Sierra is Full Professor in the Computer Sciences and Artificial Intelligence Department at the University of the Basque Country. He received his B.Sc. in Computer Sciences in 1990, M.Sc. in Computer Science and Architecture in 1992 and Ph.D. in Computer Sciences in 2000 at the University of the Basque Country. He is the director of the Robotics and Autonomous Systems Group in Donostia - San Sebastian. He is presently a researcher in the fields of Robotics, Computer Vision and Machine Learning, and he is working on the development of different paradigms to improve classification behaviours; he has written more than 25 journal papers in those fields, as well as more than 100 conference contributions and more than 30 book chapters.

Igor Rodríguez received the B.Sc. Degree in Computer Science from the University of the Basque Country, Spain, in 2012, M. Sc. In Computational Engineering and Intelligent Systems (UPV/EHU) in 2014 and is pursuing the Ph.D. degree from the Department of Computer Science and Artificial Intelligence, University of the Basque Country. His current research interests include Social Robotics, Human robot interaction and Machine Learning.

Itziar Irigoien is an Assistant Professor in the Computer Sciences and Artificial Intelligence Department at the University of the Basque Country. She received her BSc in Mathematics in 1996, and PhD in Computer Sciences in 2008 at the University of the Basque Country. She has published papers in referred journals and conferences.

Elena Lazkano is an Associate Professor in the Computer Sciences and Artificial Intelligence Department (UPV/EHU). She received her B.Sc. in Computer Sciences in 1992 (UPV/EHU), M.Sc. in Artificial Intelligence (Katholieke Universiteit Leuven, Belgium, 1993) and Ph.D. in Computer Sciences in 2004 (UPV/EHU). She is codirector of the Robotics and Autonomous Systems Group in Donostia-San Sebastian. Dr. Lazkano is presently a researcher in the fields of Robotics, being a member of the Program Committee of International Conferences and reviewer of International Journals.

10. Kapitulua

Bertsobot Arkitektura Orokorra

10.1 Kontrol arkitektura zehazten

Kontrol-arkitekturaren diseinua funtsezkoa da robotikan (ikus 6.2 atala). Robot eta atazaren araberako diseinu-printzipioak ezartzeaz gain, badira kontuan izan beharreko beste ezaugarriak ere (Kortenkamp et al., 2016) (Jones, 2004): garapen-azkartasuna, berrerabilpena, garapen inkrementalerako aukera, segurtasuna, denbora-errealean erantzuteko gaitasuna eta abar.

Printzipio guzti horiei erantzuten dien arkitektura garatzea ez da nolanahiko lana. (Astigarraga et al., 2003) artikuluan, nabigazio beharretara orientatutako SORGIN software-egituraren diseinu eta garapena azaltzen da. Askotan nahasi ohi diren arren, kontrol-arkitektura eta software-egitura kontzeptuak ez dira berdinak. Software-egitura tresna da, edo tresna-multzoa, robotaren kontrol-arkitektura inplementatzeko erabiliko duguna.

SORGIN izan zen sistema robotikoen konplexutasuna maneiatzeko sortu genuen lehenbiziko software-egitura, portaeratan oinarritutako sistemen ikuspuntua eta kontrola txertatzen dituena (Brooks, 1986), (Lazkano, 2004), (Matarič, 1997).

Sistema robotikoak kontrolatuko dituen softwareak heterogeneoa behar du izan, gailu, driver eta software-puska ugari bere baitan biltzen dituena. Portaeretan oinarritutako sistemen ikuspegitik, sistemaren portaera orokorra konkurrenteki exekutatzen diren entitate aktiboen sare gisa ikus liteke. SORGIN egiturak, entitate horiek gorpuzten ditu software bidez eta beraien arteko komunikazioa definitzen du, portaeren berrerabilpena sustatuz.

SORGIN arkitekturaren oinarriak ondoko argitalpenak jasotzen ditu:

- **SORGIN: a software framework for behavior control implementation**. A. Astigarraga, E. Lazkano, I. Rano, B. Sierra, I. Zarautz. 14th International Conference on Control Systems and Computer Science (CSCS), 2015eko uztaila, Bucarest, Polonia

Ondoko lanetan, SORGIN arkitekturan oinarrituz, nabigazio lokal eta globalerako estrategiak garatu genituen. Ikerketa-lan honen eremutik kanpo geratzen dira lan horietan garatutakoak, baina bigarren mailako ekarpen gisa eta SORGINen erabileraren erakusle ekarri ditugu hona:

- **Active landmark perception**. A. Astigarraga, E. Lazkano, B. Sierra, I. Raño. Proceedings of the 10th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR).

Garatu ondoko urteetan SORGIN erabili bagenuen ere, gaur egun jada baztertua daukagu eta ROS (*Robot Operating System*) erabiltzen dugu. Egun, robotak programatzeko zabalduen dagoen iturburu irekiko software-tresna da ROS (Quigley et al., 2009). Finean, elkarren artean komunikatu eta eragileak atzitzeko koordinatu behar diren moduluek portaera-sare bat osatzen dute eta etengabe paraleloan exekutatzen ari dira. Atzetik daukan komunitate zabal eta aktiboak, kode berrerabilgarriz osatutako liburutegi erraldoia eraiki eta mantentzen du: driver-ak, nabigazio-estrategiak, lokalizazio algoritmoak... deneratik aurki daiteke bertan. Horrek bultzatu gintuen SORGIN utzi eta ROS erabiltzera. Bestela, biak printzipio berberen gainean eraikiak izan dira (berrerabilgarritasuna, exekuzio konkurrentea, behetik gorako diseinua...), SORGINen inplementazioa dezentez xumeagoa izan arren. Azken honen egitura modularra ROS-eko antolaketa eta komunikazio prozeduralaren pareko da azken baten. Baina SORGINek ez du atzetik zaindu eta eguneratuko duen komunitaterik izan eta zaharkituta geratu da. 10.1 irudiak erakusten du SORGINekin garatutako programa baten itxura.

10.2 Bertsobot arkitektura orokorra

Lehenbiziko urteetan SORGINekin garatutako kontrol-arkitektura hartatik gaur egun ROSen inplementatu dugun Bertsobotera jauzi handia dago. Ataza edo robotaren egitekoa ere ez da inondik inora berbera: nabigazio

```
void main()
{
  /* Data declarations */
  io_data_t laser_readings;
  io_data_t motor_output;

  /* Behavior declaration */
  behavior_t avoid_obstacles;

  /* Data initialization */
  io_data_alloc(&laser_readings, 181, NULL, 0, 0);
  io_data_alloc(&motor_output, 2, NULL, 0 0);

  /* Behavior initialization */
  behavior_define(2, i, avoid_obstacles_start,
                        avoid_obstacles_stop,
                        avoid_obstacles_calculate);

  /* input/output connections */
  behavior_set_input(&avoid_obstacles, 0, running);
  behavior_set_input(&avoid_obstacles, 1, laser_readings);
  behavior_set_output(&avoid_obstacles, 0, motor_output);

  behavior_start(&avoid_obstacles);
  behavior_run(&avoid_obstacles);

  sleep(100);
  behavior_stop(&avoid_obstacles);
}
```

10.1 Irudia: SORGIN: adibide programa

atazak izan ditugu urteetan ikergai nagusi baina Bertsobotarekin gizaki-robot arteko elkarrekintzara hurbildu gara.

Bertsobotaren portaera orokorra ondoko pausuek deskribatzen dute:

1. Eserita zain egon kantatzeko txanda iritsi arte.
2. Txanda iritsitakoan, mikrofonoa identifikatu eta bere aurrean kokatu. Gai-jartzaileak emandako ariketa/gaia jaso.
3. Bertsoa osatu eta jendaurrean kantatu.
4. Publikoaren erreakzioa jaso eta horren arabera erantzun.
5. Atzera buelta eta eseri aulkian.

Bertsobot arkitekturak, 2012ko apirilean jendaurreko lehen agerraldia egin zuenetik 2016ko irailean UPH/EHUko uda ikastaroetan egindakora, aldaketa nabarmenak izan ditu. Software-egitura aldatu dugu, robotak ere bai, elkarreragiteko modua... ia ia esan genezake hasierako saio hartatik mantentzen den gauza bakarrenetakoa izena dela.

2012tik gaurdaino jendaurrean egindako ekitaldien errepaso azkarra egingo dugu, den denak ez baina nabarmenenak aipatuz, horrek Bertsobotak izan duen bilakaera hobeto erakutsiko duelakoan:

- **2012/04 - Hitzaren gunea inaugurazioa Gipuzkoako campusean**: lehen agerraldi publikoa eta, akaso, gure aldetik ausartena ere bai. Tartalo eta Galtxagorri - PeopleBot eta Pioneer2DX plataformak hurrenez hurren - erabili genituen aire librean egindako saioan. Robotek erakutsitako gorputz adierazkortasuna minimoa izan zen, roboten gorpuzkerak baldintzatuta eta elkarrizketa-bidezko komunikaziorik ez zen. Robotak ia erabat teleoperatuak zeuden. Bertso-sorkuntza sistema (oso oinarrizkoa) txertatuta zuten robotak ziren, besterik gabe. Ekitaldiko bideoak ikusi nahi ezkero, jo (RSAIT research group, 2012b), (RSAIT research group, 2012a)

- **2013/05 - Ikasle eta roboten arteko dema**: Informatika fakultatean egin zen ekitaldian, bertso-eskolako ikasleak aritu ziren robotekin nor baino nor. Tartaloren ondoan lehenbiziko aldiz agertu zen NAO[1] robot humanoidea. Gai-jartzaile lanak egin zituen NAOk eta oinarrizko keinuak egiteko gai zen modu erdi-autonomoan. *Choreographe*[2] izeneko softwarearekin kontrolatzen genuen orduan NAO (RSAIT research group, 2013)

- **2014/03 - Emakumeen Eguna Informatika fakultatean**: UPV/EHUk urtero ospatzen du Emakumeen Nazioarteko Eguna fakultate ezberdin batean eta 2014an Informatika fakultatearen txanda izan zen. Programaren baitan bertso-saioa izan zen eta bi goi-mailako bertsolari aritu ziren NAO eta Tartalorekin batera. NAOk hizketa-bidezko komunikazio gaitasunak erakutsi zituen eta baita bertsolarien oholtza gaineko gorputz-adierazkortasuna ere. Baina, oraindik ROSera pasa gabeak ginen eta bere funtzionamendua ez zen erabat autonomoa (RSAIT research group, 2014)

- **2014/04 - Robot Bertsolaria, zientzia ala fikzioa?** Badu Bada erakusketaren baitan hitzaldi-erakustaldia egin genuen Bilboko Azkuna zentroan. NAOk hizketa-bidez komunikatzeko gaitasuna erakutsi zuen, gai-jartzailearekin ez ezik publikoarekin ere elkarreraginez. Jendeak jarritako errimekin bertsoa osatu zuen bat-batean.

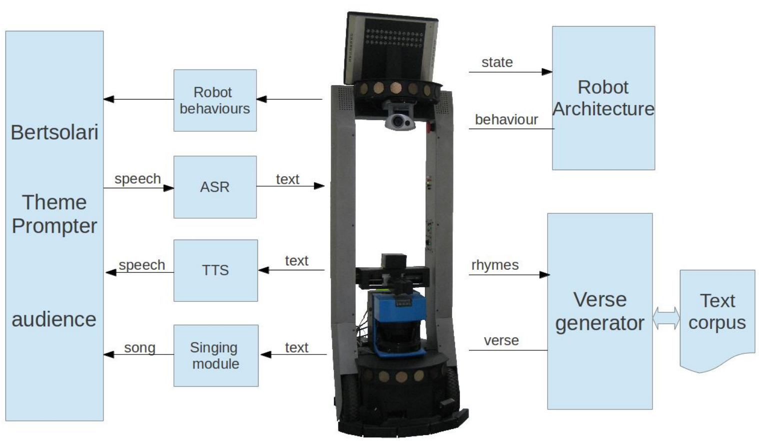
- **2016/09 - Udako ikastaroen itxiera ekitaldia**: "Irakaslegoaren ebaluazioa: gainditu gabeko ikasgaia"ikastaroaren itxiera

[1] https://www.ald.softbankrobotics.com/en/cool-robots/nao

[2] Aldebaranek sortutako aplikazio plataforma-anitza NAO robota monitorizatu eta kontrolatzeko

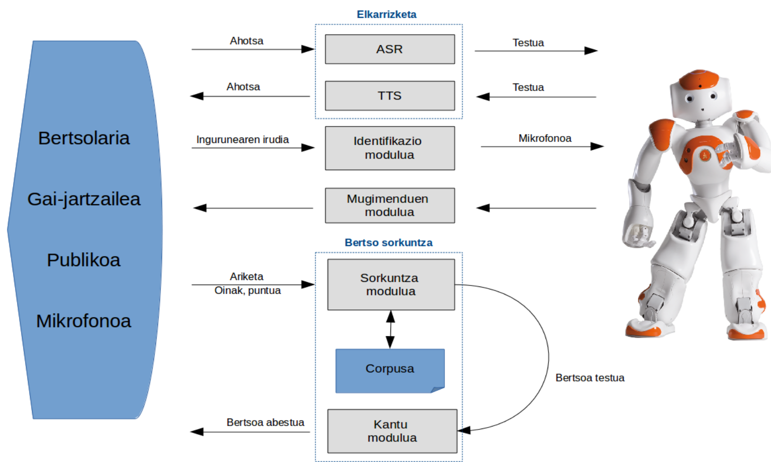
ekitaldira gonbidatu zuten Bertsobot. Ez zen bertso-saioa izan baina elkarrizketa-bidez elkarreragin eta ordura arte landutako gorputz-adierazkortasunaren aspektu guztiak erakutsi zituen. Txalo-bidezko *feedback*-a jasotzeko sistema eta egoera emozionala adierazteko keinuak garatuta zeuden dagoeneko.

Bertsobotak 5 urteotan izan duen garapenaren berri emateko, kontrol-arkitektura ezberdinen irudiak ekarri ditugu hona. 10.2 irudiak lehenbiziko kontrol-arkitektura erakusten du, Tartalo eta Galtxagorri robotetan inplementatu genuena. 10.3 irudiak berriz, NAO humanoidean garatutako lehenbiziko Bertsobot arkitektura erakusten du. Azkenik, 10.4 irudian gaur egungo Bertsobota ageri da.



10.2 Irudia: Tartalo Bertsobotaren arkitektura

Arkitektura berrienaren azalpen azkar bat emango dugu: irudian agertzen diren ataza guztiak, esan bezala, ROSen oinarritutako kontrol-arkitekturak gauzatzen ditu. *Performance State Controler* modulua da saioaren dinamika orokorra edo ordena mantentzeaz arduratzen dena. *Face and Sound Localization* eta *Chatting* moduluak gai-jartzailearekin elkarreragiteko dira, gai-jartzailearen aurpegia detektatu eta elkarrizketa burutzeko. *Find Key Objects* moduluak, berriz, inguruneko elementuekin elkarreragiteko gaitasunaz ornitzen du robota (mikrofonoa/aulkia bilatu). Elkarrekintza horiek robotaren eragileetarako aginduak itzuli ohi dituzte eta hauek *Motion Controller* moduluak kudeatzen ditu. Unean unean exekutatu beharreko ekintza robotaren uneko egoeraren baitakoa da,



10.3 Irudia: NAO Bertsobotaren arkitektura. Lehen bertsioa

Robot State Controller moduluak kudeatzen duena. Bertsoa sortu eta kantatzearen ardura berriz, *Verse Generation* prozesuarena da, eta publikoaren erreakzioak (txalo bidezkoa) *Feedback from the Audience* portaerak jasotzen ditu. Robotaren gorputz espresioa *Gesture Controller* moduluaren eskuetan dago, zeinak une bakoitzean *Performance State*, *Robot State* eta *Emotion Selector* moduluek emandako informazioaren baitan, burutu beharreko mugimendu-sorta erabakitzen duen.

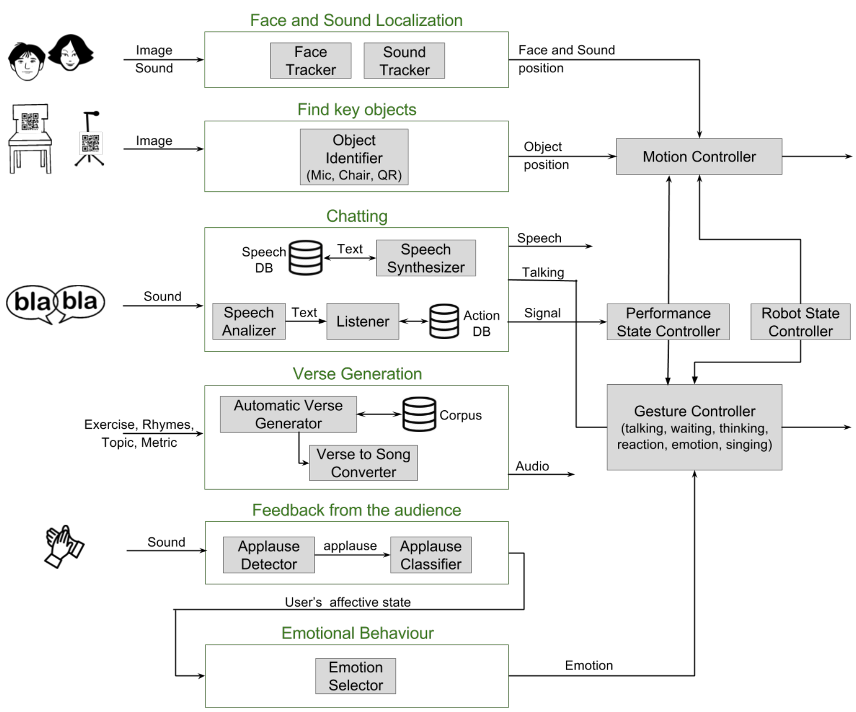
Bertsobot-aren arkitektura orokorra, beraien bertsio ezberdinetan, ondoko argitalpenek jasotzen dute:

- **Bertsobot: the first minstrel robot**. A. Astigarraga, M. Agirrezabal, E. Jauregi, E. Lazkano, B. Sierra. 6th International Conference on Human System Interaction (HSI), 2013ko ekaina, Sopot, Polonia.

- **Hasi, Bertsobot**. A. Astigarraga eta I.Rodriguez. IkerGazte kongresua. 2015eko maiatza, Durango.

- **Robots en un escenario de poesıa improvisada**. I. Rodriguez Rodriguez, A. Astigarraga, E. Jauregi, D. Salinas, E. Lazkano, T. Ruiz-Vázquez. XXXVI Jornadas de automática, 2015eko iraila, Bilbo.

- **BertsoBot: Towards a Framework for Socially Interacting Robots**. A. Astigarragaa, I. Rodriguez, T. Ruiz-Vázquez, E. Lazkano. Spanish Robotics Conference (JNR), 2017ko ekaina. Valentzia.



10.4 Irudia: NAO Bertsobotaren gaur egungo arkitektura

Bertsobotaren ikerketa-eremutik kanpo, baina robotika mugikorra euskaraz landu eta azaltzeko ahaleginaren erakusgarri, ondoko argitalpena gehitu dugu bigarren-mailako ekarpen gisa:

- **Robot mugikorrak. Oinarriak**. A. Astigarraga, E. Lazkano. Udako Euskal Unibertsitatea, 2012. ISBN: 9788484383772.

SORGIN : A SOFTWARE FRAMEWORK FOR BEHAVIOR CONTROL IMPLEMENTATION

A. Astigarraga[1], E. Lazkano[1], I. Rañó[1], B. Sierra[1], I. Zarauz[2]

(1) Department of Computer Science and Artificial Intelligence
University of the Basque Country
email: {ccbaspaa, ccplaore, ccpsiarb}@si.ehu.es,
irano@miramon.es
url: http://www.sc.ehu.es/ccwrobot
(2) Department of System Engineering and Authomatics
email: ispzalei@sp.ehu.es

Abstract: This paper describes the software framework that has been developed and that it is being used to build a control architecture for the navigation of a B21 mobile robot. The subsumption architecture and the multi-agent based controller architectures are analyzed and a software framework named SORGIN is presented as a structured programming tool that allows to deal with the complexity of mobile robotic systems. We show how a task that performs safe wandering with a privileged compass orientation has been defined using different behavior bricks in the SORGIN framework.

Keywords: Robot Control, Intelligent Control, Behavior-Based Systems, Autonomous Systems, Software Framework

1. INTRODUCTION

The main goal of the project we are involved in is to design and implement a robust control architecture for mobile robot navigation. It is a known fact that a robot system is typically too complex to be developed and operated using conventional programming techniques. Rather managing complexity demands frameworks and tools that embody well-defined concepts to enable the effective realization of systems to meet high-level goals (Coste-Maniére and Simmons, 2000). This paper describes the software framework that has been developed and that it is being used to build a control architecture for the navigation of a B21 mobile robot.

Before getting deeper in details we think it is important to clarify and distinguish those two concepts that have already been mentioned and will be used further on this paper: software framework and control architecture. By software framework we mean the software components of the system and the relationships among them. Although this definition matches some of the existing definitions for software architectures, we prefer to be prudent and name it using the term *framework* because our primary goal is just to develop a tool that will allow us to define the proper control architecture for the robot. The definition of control architecture also lacks consensus in the context of robotic systems. We think that the control architecture can be seen as the overall organization of the different data processing systems in the robot that control its interaction with the environment. According to (Matarić, 1992), an architecture provides a set of principles for organizing control systems. However, in addition to providing structure, it imposes constraints on the

⋆ This work was supported the University of the Basque Country and by the Gipuzkoako Foru Aldundi Txit Gorena under OF147/2002 grant

way control problems can be solved.

In this paper we first analyze two robot control architectures and based on this study a software framework named SORGIN is presented. This framework provides a structured programming tool that allows to deal with the complexity of mobile robotic systems. We show how a task that performs safe wandering with a privileged compass orientation has been implemented using different behavior bricks in the SORGIN framework.

2. RELATED WORK

In order to develop the framework to meet the needs, it is mandatory to review the area of control architectures that form a theoretical background. Different types of control strategies for autonomous mobile robots can be found in technical literature. *Planner-based* or deliberative strategies use a centralized world model for verifying sensory information and generating actions in the world (Chatila and Laumond, 1985)(Albus et al., 1987). Deliberative approaches have been criticized for scaling poorly with the complexity of the problem and consequently not allowing for reaction in real-time (Brooks, 1990, 1991).

Reactive bottom-up approaches implement the robot's control strategy as a collection of preprogrammed condition-action pairs with minimal state (Agre and Chapman, 1987)(Brooks and Connell, 1986). Purely reactive strategies have proven effective for a variety of problems that can be well defined at design time (Bonasso et al., 1992), but are inflexible at run-time due to their inability to store information dynamically (Matarić, 1992).

In *Hybrid* strategy, the goal is to combine the best of both reactive and deliberative approaches. The challenge of this approach is bringing the two parts together, and resolve conflicts between the two. This requires a "third" part on the architecture, thereby these systems can result on "three-layered systems" (Kortenkamp and Bonasso, 1998; Arkin, 1998).

Behavior-based approaches are strictly more powerful than pure reactive systems since they have no fundamental limitations on internal state (Matarić, 1994). In behavior-based strategy, the robot controller is organized as a collection of modules, called behaviors, that receive inputs from sensor and/or other behaviors, process the input, and send outputs to actuators and/or other behaviors. All behaviors in a controller are executed in parallel, simultaneously receiving inputs and producing outputs. Unlike reactive systems, behavior-based systems are not limited in their expressive and learning capabilities: behaviors themselves can have a state (internal and particular view of the world), and can form representations when networked together. So, if a robot needs to plan ahead, it does so in a network of behaviors which talk to each other and send information around.

Below, it follows a more detailed description of two control systems. These control systems are explained identifying the two classes of elements that define the systems: behaviors (modules and agents, respectively) and type of communication among those components.

2.1 *Subsumption Architecture*

Developed by Rodney Brooks (1986). Its original implementation using Lisp, gave the basis to the later development of the Behavior Language. It is probably the most widely known robot control architecture and it is in this architecture where the behavior-based systems were founded. The two main components are:

- Modules are finite state machines that operate in parallel, achieving specific tasks. Those FSM are augmented to use time and to manage specific data structures. Sets of modules are hierarchically ordered in layers, and this layer ordering implicity defines a priority.
- Communication among modules is performed by message passing, but there is no acknowledgement nor handshaking among the processors. A suppression and inhibition communication mechanism is used to coordinate the different modules in the hierarchy. Inputs are suppressed by higher layers' outputs in the sense that any input in the same wire is replaced by the output of the higher priority behavior. Lower level outputs in the other hand can be inhibited by higher level outputs for a time period.

A big advantage of the subsumption architecture is its incremental approach to the construction of behaviors. This makes testing the system easier, provides robustness, and generally a greater chance of overall success. Subsumption succeeds in terms of robustness, additivity and simplicity. The fact that the communication between layers is simple and well defined, and the processing distributed across several CPUs means that adding more layers, sensors or processors to a system is easy in this architecture. One of the drawbacks of the Subsumption architecture is that processors cannot cooperate -each layer is built on its own "hardware" totally from scratch. The control system is hardwired directly in the structure of the behaviors and their interconnections, and can thus not be altered without redesigning the system. We find that the subsumption architecture is not sufficiently modular. Because higher layers interfere with the lower-level behaviors, they cannot be designed independently and become increasingly complex. We also believe that a shortcoming of this architecture is the lack of the possibility to combine information from different modules. Combining information may result in better control signals. However, our critiques addresses subsumption as an engineering methodology. Hartley and Pipitone (1991) critique this architecture in the same direction.

2.2 *Agent-Based Control Architecture for a Mobile Robot*

In his Ph.D. dissertation, Van Breemen (2001) critics subsumption arguing that it lacks a good design method, and presents an agent-based design method for solving complex control problems. His theoretical agent-based framework is more related to the field of control engineering than to the field of robotics. In spite of this, Astigarraga (2002) implements a software framework using C++. It describes the control tasks of the mobile robot LiAS based on Van Breemen's theoretical framework. In Astigarraga's work the main components are:

- Agents, the concept of an agent form the basis of the framework. An agent is a component that solves a particular (sub)problem. It is the basic building block. Coordinated group of agents form agencies which are treated as new coherent agents. Dependencies between groups of agents are solved by coordination objects, and all the mentioned elements are encapsulated in a MAC (Multi Agent Controller) component, which makes up the overall controller.
- Communication: The inputs and outputs are used to exchange data between components and are considered as a data flow. On the other hand, messages are sent to execute particular methods of components, and therefore can be considered as a control flow. The overall controller (MAC) is operated by three messages that come from the software environment in which it operates: *start()* to initialize (the components of) the MAC; a *tick()* message is sent periodically to indicate that a new control sample needs to be calculated. When the MAC receives the *tick()* message, subsequent messages to the coordination objects are send. The coordination object decides which agent should calculate new control samples. Finally, when control samples from agents are combined by the coordination object, the MAC sends messages to actuators; *stop()* to finalize (the components of) the MAC.

  The presented framework was applied to implement safe wandering control task in the mobile robot LiAS. Figure 1 shows how different agents are combined into one overall solution forming a hierarchically organized multi-agent controller (MAC).

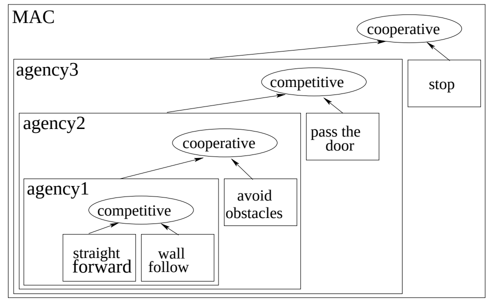


Fig. 1. Overall picture of the safe wandering task

An advantage of this framework is that it provides well structured components to describe controllers. Combining information among different modules is also possible, and this can be done in several ways and is defined by the coordination object. The control loop consists of reading the hardware sensors, calculating a control sample, and writing the control sample to the hardware actuators. But a robot system usually have multiple sensors and actuators that operate at different frequencies. One of the largest drawbacks is that it lacks specific mechanisms to implement different sample frequencies or the possibility to operate on an event-driven basis.

3. SORGIN : A BEHAVIOR-BASED SOFTWARE FRAMEWORK

In our work we use Subsumption Architecture and Van Breemen's framework as a basis, but extended them where they did not meet our requirements. The Subsumption philosophy mandates that the behaviors be relatively simple, incrementally added to the system, and executed in a parallel fashion. On the other hand, Van Bremen proposes a well structured framework to describe behaviors, providing the basis to develop a solution for complex robot control problems in a clear articulate way. Our aim is to conserve the robustness and real-time properties of behaviors and to develop a well-structured framework and a behavior representation that supports behavior reuse for multiple tasks. To develop the software framework, we must define its main components: behaviors and communication mechanism.

Behaviors consist of:

- A set if inputs and outputs that are links to a specific class of data that will be explained later on.
- A link to an initialization function that will perform everything needed before the main

loop start. This function is specially important when the module deals directly with a physical device.
- A link to the function to be performed in the main loop, thereby, the function that will be executed as an independent process (*calculate*).
- A link to a function to be performed when the behavior or module stops its execution.

Opposite to Van Breemen's framework, there is not a central component that directs the overall controller. When behaviors are launched, they are fully responsible of their execution; everyone runs at their respective working frequency. After a behavior brick is initialized and starts running, a thread is created that will execute until it stops or someone stops it. Therefore, the behavior structure definition contains a *running flag* that is always updated according to the 0-th input of the behavior – always reserved to stop the execution of behaviors. Because every behavior runs in its own thread, the specific tools used for the implementation require a delay time setting that limits the running frequency of its thread. This is mandatory to avoid behaviors to exploit all the CPU time.

Communication among the modules is performed establishing connections between the modules' inputs and output links and the data. These data (*io_data*) are more than data-messages. Each *io_data* element contains all the information needed to safely read/write the data through the net when it is required, or to avoid race conditions when the information is required locally. In this way, during the implementation of an instance of a *calculate* function the designer of the control architecture need not to worry about those details, only the offered standard functions to read/write the inputs/outputs must be called. As in the subsumption architecture, there is no handshaking among the modules when sharing data, each one runs at its own frequency and depending on that frequency data can be updated before it is processed. We do not use any inhibition or suppression mechanism but we prefer to add coordination modules for actuators signal fusion, adopting Van Breemen's style. We think this approach gives us more flexibility to change the coordination method in the sense that only the link to the *calculate* function of the coordination module needs to be changed, but not the communication net. Note that nothing prevents a behavior to delay the output generation of another behavior. Only the appropriate data and links would have to be defined.

Figure 2 shows graphically the backbone of SORGIN . Of course, SORGIN offers the appropriate interface for the user to easily define behavior and data instances and to establish the links between the data structures and the behavior input or outputs.

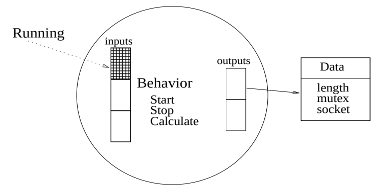


Fig. 2. Behavior and *io_data structures.*

4. A ROBOT CONTROL SYSTEM INSTANCE

The best way of explaining the use of abstract behavior representation is through demonstration. This section gives as example the definition of an obstacle avoidance behavior in SORGIN framework that uses only infrared sensors. This behavior is used in the mobile control system described afterwards that achieves the goal of safe-wandering while navigating in a privieldged orientation.

4.1 *Illustrative example*

Figure 3 shows the pseudo-code of the *calculate* function of the *avoid_obstacles* behavior. The

```
avoid_obstacles_calculate()
{
    read_input(0) /*running*/
    read_input(1) /*infrared readings*/
    If (running)
        infrared_avoid_obstacles_algorithm
    end_if
    write_output(0, trans_v, rot_w, flag);
}
```

Fig. 3. avoid_obstacle behavior implementation

communication among the specified behaviors is performed establishing connections between the different behaviors and the corresponding data elements, as shown in figure 4. Since the *io_data* type contains all the information needed to safely read/write the data, only the appropriate connections must be fixed to provide infrared readings to an avoid_obstacle behavior and connect its output to the avoid velocity data.

4.2 *Validation experiment*

The goal of the validation experiments is to demonstrate the key features of the presented software architecture – abstract behavior representation and behavior re-usability among others.

```
void main()
{
 /*Data declaration*/
 io_data_t ir_readings;
 io_data_t avoid_output;

 /*Behavior declaration*/
 behavior_t avoidObstacles;

 /*Data initialization*/
 init_data(ir_readings, 20);

/*Behavior initialization*/
 behavior_define(2, 1, avoid_obstacles_start,
                   avoid_obstacles_stop,
                   avoid_obstacles_calculate)

 /*Input/Output Connections*/
 behavior_set_input(avoid_obstacles, 0, running);
 behavior_set_input(avoid_obstacles, 1, ir_readings);
 behavior_set_output(avoid_bstacles, 0, avoid_output);
 behavior_start /*initialize the behavior*/
 behavior_run
 behavior_stop /*finalize the behavior*/
}
```

Fig. 4. main program structure example

Towards this end, we considered the task of making the robot go from our laboratory to the library, and then, come back again. Figure 5 shows the set of abstract behavior instances employed for the task and the structure of the overall control architecture. There is a small set of behaviors that are imposed by the robot hardware and must be used in every control task: *MSP-manager* (Multiple Sensor Processor) provides sonar, infrared and bumper readings in proper data objects; *Compass-manager* provides compass readings in compass data object; and *MCP-manager* (Multiple control processor) that directs the motor control and can give odometry and battery voltage information. Each one is directly related to a physical device in the robot. The rest of the behaviors are specific for the current application, but can be reused for some others also:

- The *bumper-stop* behavior stops the robot when it collides with an object and sends a special signal on its output line that shuts down the overall system.
- The *ir-avoid* behavior avoids any obvious obstacle detected by the infrared sensors.
- The *corridor-follower* behavior identifies corridors and maintains the robot at a centered position with respect to the left/right side walls.
- The *map-builder* behavior needs to be provided with odometric information and sonar reading and displays a geometric map of the environment.
- The *compass-follower* behavior takes a goal orientation (in term of compass heading) and attempts to reach that goal based on the difference between the final desired orientation and current orientation. To do this it sends headings that make the robot rotate towards the desired orientation.
- The *orientation reader* behavior receives user headings and produces a new goal orientation. This behavior pretends to simulate a landmark recognition behavior and change the orientation when the robot arrives to the library, so that it returns to the lab.
- Because each behavior is aiming at its own objective, it is essential to carry out a coordination task. A coordination mechanism (behavior) prevents conflicts when multiple outputs are sent to actuators or other behaviors (Maes, 1989; Pirjanian, 1998). The *coordination mechanism* of the example, has the responsibility of deciding the final motor commands and, depending on the chosen type of control, can be hierarchical or cooperative.

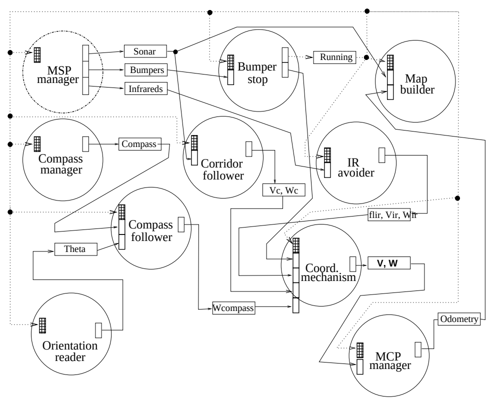


Fig. 5. Architecture instance for wandering

5. EVALUATION

The SORGIN software framework is inherently modular from a software design perspective. This enables a robotic system designer to expand robot's competency by adding new skills without redesigning or discarding the old ones. This is very useful for increasingly construct more complex robotic systems. Behavioral abstraction is provided in our software framework with the *behavior* and *io_data* components, allowing thus a uniform representation of computational objects. Code reusability is another positive aspect of SORGIN . It promotes the automatic re-usability of behaviors across different tasks, and thus, the automatic generation of behavior libraries. The decomposition and aggregation that is needed in incremental design is conceptualized in our framework. Software components can be added, modified or removed without re-programming the remaining system. Our design method allow for a bottom-up approach and incremental realization of a robot control system. This makes testing the system easier, provides robustness, and generally a greater chance of overall success, since the smaller, simple parts of the problem are already solved when attacking the larger part, and need not be worried about again. Although the portability to other platforms remains to be tested, it must be pointed out that SORGIN has been developed using the "C" programming language and Posix threads, both standard and portable elements that in principle assures the portability. The SORGIN framework has demonstrated that it is well suited to the objective it has been developed for.

6. CONCLUSIONS AND FURTHER WORK

We tried to show in this work that a critical part of the design of the control architecture of a mobile robot is the software framework in which the architecture will be embedded. Navigating mobile robot control can be viewed as a soft real-time task in the sense that the robot must act fast enough to meet its goals. On the other hand, reuse of software designed specifically to control robotic systems is difficult. Software to control robotic systems may be very heterogeneous, involving numerous devices and software. From a generic point of view, global behavior can be considered as a network of coupled concurrent active entities - threads- interacting asynchronously among them in some way. This paper describes a software framework that identifies these active entities with software components and defines their interaction, creating thus a principled method of modularity and behavior libraries. SORGIN is still at an experimental phase. A control architecture for a more complex robot behavior must still be developed to exploit the properties of the framework and to identify its weaknesses and lacks.

REFERENCES

Agre, P. E. and **Chapman, D.** (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 268–272.

Albus, J., MacCain, H., and **Lumia, R.** (1987). Nasa/nbs standard reference model for telerobot control system architecture (nasrem). Technical Report 1235, Robot Systems Division, National Bureau of Standards.

Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press.

Astigarraga, A. (2002). Agent-based control architecture for a mobile robot. Master's thesis, Euskal Herriko Unibertsitatea.

Bonasso, R. P., Antonisse, H. J., and Slack, M. G. (1992). A reactive robot system for find and fetch tasks in an outdoor environment. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 801–808.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, RA-2*, pages 14–23.

Brooks, R. A. (1990). Elephants don't play chess. *Journal of robotics and autonomous systems(1-2)*, 6:3–15.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47:139–160.

Brooks, R. A. and **Connell, J. H.** (1986). Asynchronous distributed control system for a mobile robot. In *Proceedings of the SPIE's Cambridge Symposyum on Optical and Optoelectronic Engineering*, pages 77–84.

Chatila, R. and **Laumond, J.** (1985). Position referencing and consistent world modeling for mobile robots. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 138–145.

Coste-Maniére, E. and **Simmons, R.** (2000). Architecture, the backbone of robotic systems. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 67–72.

Hartley, R. and **Pipitone, F.** (1991). Experiments with the subsumption architecture. In Press, I. C. S., editor, *Proceedings IEEE International Conference on Robotics and Automation*, volume 2, pages 1652–1658.

Kortenkamp, D. and **Bonasso, R. P., editors** (1998). *Artificial Intelligence and Mobile Robotics. Case Studies of Succesful Robot Systems*. MIT Press.

Maes, P. (1989). The dynamic of action selection. In *Proceedings of the 1989 International Joint Conference on Artificial Intelligence, Detroit*, pages 991–997.

Matarić, M. (1992). Behavior-based systems: Key properties and implications. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, pages 46–54.

Matarić, M. (1994). *Interaction and Intelligent Behavior*. PhD thesis, Massachusetts Institute of Technology.

Pirjanian, P. (1998). *Multiple Objective Action Selection and Behavior Fusion using Voting*. PhD thesis, Institute of Electronic Systems, Aalborg University, Denmark.

Van Breemen, A. (2001). *Agent-based Multi-Controller Systems: a design framework for complex control problems*. PhD thesis, University of Twente.

Bertsobot: the first minstrel robot

Aitzol Astigarraga, Manex Agirrezabal, Elena Lazkano, Ekaitz Jauregi and Basilio Sierra

***Abstract*— We describe a robot capable of composing and playing traditional Basque impromptu verses -*bertsoak*. The system, called *Bertsobot*, is able to construct improvised verses according to given constraints on rhyme and meter, and to perform it in public. Towards this end, several tools and applications have been developed and integrated in *Bertsobot*, including: speech-based communication system, text applications for verse generation, and robot behaviours to interact with the environment in a public performance. We describe the tools and processes behind our approach, present some early experimental results and illustrative verses, and finally, remark the conclusions and future steps.**

I. INTRODUCTION

The main objective of this research project is to develop an autonomous robot capable of generating improvised verses in Basque. The interaction with the robot should be speech-based; thus, the system should be able to receive the instructions to compose the verse in Basque, to generate the most appropriate verse according to the given instructions and sing it with the proper melody. The robot should also show the degree of expressiveness that Basque troubadours, *bertsolari-s*, show in their performance. And all those tasks must be accomplished concurrently in a extemporaneous performance.

This paper presents our first steps towards *Bertsobot*, the first minstrel robot. *Bertsobot* is a collaborative project between three research groups of the University of the Basque Country (UPV/EHU), each of them expertise in a different area:

- RSAIT (Robotics and Autonomous Systems Group)[1], specialized in autonomous robot navigation and machine learning techniques.
- IXA group[2], specialized in the processing of written text at different levels: morphology, syntax, semantics, IR, machine translation...
- The Aholab Signal Processing Laboratory Group[3], specialized in speech synthesis and recognition technologies.

This work has been partially conducted under the SAIOTEK projects XIMUR and POETAUTO, the Basque Government Research Team grant and the University of the Basque Country UPV/EHU, under grant UFI11/45 (BAILab).

Aitzol Astigarraga, Elena Lazkano, Ekaitz Jauregi and Basilio Sierra are with the Computer Science and Artificial Intelligence Department, University of the Basque Country (UPV/EHU), Donostia aitzol.astigarraga@ehu.es, e.lazkano@ehu.es, ekaitz.jauregi@ehu.es, b.sierra@ehu.es

Manex Agirrezabal is with the Language and Software Systems Department, University of the Basque Country (UPV/EHU), Donostia manex.aguirrezabal@ehu.es

[1]http://www.sc.ehu.es/ccwrobot/

[2]http://ixa.si.ehu.es/Ixa

[3]http://aholab.ehu.es/aholab/

The *Bertsobot* project joins together in a single system the tools that each group created independently, showing thus the potential of the integration of these technologies.

II. SOME WORDS ABOUT BASQUE LANGUAGE AND BERTSOLARITZA

Basque, *euskara*, is the language of the inhabitants of the Basque Country. It has a speech community of about 700,000 people, around 25% of the total population. Seven provinces compose the territory, four of them inside the Spanish state and three inside the French state. Therefore, the Basque speakers' community is small and finds itself in a minority in its native land.

Bertsolaritza, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive. Events and competitions in which improvised verses, *bertso-s*, are composed are very common. In such performances, one or more verse-makers, named *bertsolari-s*, produce impromptu compositions about topics or prompts which are given to them by an emcee (theme-prompter). Then, the verse-maker takes a few seconds, usually less than a minute, to compose and sing a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme. Melodies are chosen from among hundreds of tunes. Fig. 1 shows a picture of the national championship of *bertsolaritza*, which took place on 2009.

Fig. 1. *Bertsolari Txapelketa Nagusia*, the national championship of the Basque improvised contest poetry

Xabier Amuriza, a famous verse-maker that modernized the *bertsolari* movement, defined *bertsolaritza* in a verse as:

Neurriz eta errimaz	*Through meter and rhyme*
kantatzea hitza	*to sing the word*
horra hor zer kirol mota	*that is what kind of sport*
den bertsolaritza.	*bertsolaritza is.*

978-1-4673-5637-4/13/$31.00

Let's take a brief explanation about the artistic creation of a *bertso*: when constructing an improvised verse a number of formal requirements must be taken into account. Rhyme and meter are inseparable elements in improvised verse singing (in the above example, odd lines have seven syllables and even lines six. And odd lines must rhyme with each other). A person able to construct and sing a *bertso* with the chosen meter and rhyme is considered as having the minimum skills required to be a *bertsolari*. But the true quality of the *bertso* does not only rely on those demanding technical requirements. The real value of the *bertso* resides on its dialectical, rhetorical and poetical value. Thus, a *bertsolari* must be able to express a variety of ideas and thoughts in an original way while dealing with the mentioned technical constraints. In this balance lies the magic of a *bertso*.

III. Related work

To the best of our knowledge, there are no poetry generation systems implemented in a real robot. But *bertsolaritza* belongs to the oral genre, and the public performance is extremely important. Therefore, it is not enough the development of an automatic verse generation system, the created poem has to be part of a performance. Thus, a real body that interacts with the public and sings the improvised verse with a proper melody is needed. *Bertsobot* aims to join together the capabilities of an autonomous robot to sense its environment and interact with it, and the natural language processing tools devoted to automatic verse generation, in an attempt to generate improvised and context-specific poetry.

There exists some prior work related to Basque verse-making technologies that served as a foundational basis for our project:

BertsolariXa

BertsolariXa [5] is a rhyme searching tool that provides rhymes for a given word. This tool was implemented as finite-state automata using the two-level morphology formalism. It also contains other features such as semantic categorization of words, narrowing word-searches to certain themes, etc.

BAD tool

BAD [1] is a web-based assistant tool for making verses, which helps people in the verse-making process. The main functions of BAD are the following: visualization of the selected verse structure, verse-structure checking (counting syllables and evaluating the rhyme pattern), tools for rhyme and synonym searching and a verse singing module.

The above mentioned tools were specifically designed to assist Basque verse writers during the composition process. In our opinion, automatic generation of poems resembles the creation of a *bertso* in these three aspects:

1) They have to satisfy very specific technical requirements: on the one hand metric restrictions, that is, the number of syllables per line; and, on the other hand, they have to meet certain rhyme pattern.
2) They are allowed a certain poetic license which implies, sometimes, deviations from the rules of syntax and semantics.
3) The result must be meaningful for the user, more specifically, the resulting text must arouse specific emotions amongst the audience.

Taking aforementioned similarities in mind, we reviewed the existing literature on the automatic creation of poetry. A good review can be found in [19] and [10]. Most relevant ones include:

WASP

The WASP system [11] can be considered one of first serious attempts to build an automatic poetry generator system. It is based on the generate-and-test paradigm of problem solving. Simple solutions are generated and then coupled with an evaluation function for metric constraints, producing acceptable results.

ASPERA

ASPERA [12] is a case-based reasoning (CBR) system for poetry generation. It generates poetry based on the information provided by the user: a prose description of the intended message, a specific stanza for the final poem, a set of verse examples on that stanza, and a group of words that the final poem must contain.

The system was implemented using CLIPS rule-based system, and follows the four typical CBR steps: Retrieval, Reuse, Revise and Retain.

COLIBRI

The COLIBRI system [8] also uses CBR and is very similar to ASPERA. The main difference resides in the fact that COLIBRI incorporates an ontology called CBROnto, improving its inference power as well as the representation and use of more explicit and general knowledge.

POEVOLVE

Levy [17] went on to develop an evolutionary model of poetry generation. POEVOLVE creates limericks taking as a reference the human way of poetry writing. The POEVOLVE system works as follows: an initial population is created from a group of words that include phonetic and stress information. Rhymes that meet the requirements are selected and then more words are selected to fill the rest of the verse-line based on their stress information. A genetic algorithm is employed to modify the words that compose the limerick. Evaluation is performed by a neural network trained on human judgements. It must be said that this system does not take syntax and semantics into account.

McGonnagall

Manurung presented also an evolutionary approach to generate poetry [18]. The poem generation process is formulated as a state space search problem using stochastic hill-climbing. The overall process is divided in two steps: evaluation and evolution. During the evaluation phase, a group of individuals is formed based on initial information, target semantics and target phonetics. This group of initial individuals is then evaluated taking into account different aspects such as phonetics, semantics and surface form. Each individual receives a score, and in the evolution step, the subset with higher scores is selected for reproduction. The resulting mutated individuals derive, hopefully, in better versions of the poem.

Haiku generation using Vector Space Model

Wong and Chun [21] presented an approach to generate "modern haikus" using text collected from blogs. The proposed approach uses a keyword lexicon consisting of the most common 50 words used in haiku writing, and a line repository containing sentence fragments found in blogs.

The haiku generation process starts by choosing three keywords from the lexicon that will form the general picture of the haiku. Then, sentence fragments in the blogosphere containing those keywords are searched. Two keywords are extracted from each sentence, using a *tf-idf* weighting scheme to evaluate how important each word is to a sentence. Afterwards, vectors are used to compare each sentence pair. The cosine of the angle between two vectors is used to measure their semantic relation. Finally, the vector pair with the highest cosine value is selected and the combination of the most semantically related pairs of vectors are chosen for the resulting haiku.

IV. RESOURCES

Several resources and tools have been used in the *Bertsobot* project.

A. Speech applications

- AhoTTS[4] [13]: a modular Text-To-Speech conversion system for Basque and Spanish.
- Singing synthesis[5]: based on the text-to- speech synthesis engine [9] developed by Aholab, Agirrezabal [1] implemented a singing module. The application is able to sing Basque verses written by the user, with a choice of various standard melodies for *bertsolaritza*. The developed singing module is offered as a web service.
- A Basque Automatic Speech Recognition system (ASR), developed by Scansoft and adapted by Robotiker[6] [16].

B. Text applications

- **Rhyme search**: rhyme and metric compound the technical requirements of a *bertso*. We could say that without rhyme there is no *bertso*. Certain lines within a *bertso* have to rhyme with each other; thus, finding words that rhyme with a given word is an essential task that the verse-generation system must perform. Basque rhyme schemes also allows for a certain amount of leeway that *bertsolari-s* can take advantage of. The widely consulted rhyming dictionary *Hiztegi Errimatua* [4] contains a number of documented phonological alternations that are acceptable as off-rhymes. These alternations have been implemented using finite-state technology. A detailed description can be found in [2].
  For example, if we search rhymes for the word "apeta" (whim), the system returns "eta" (and), "epa" (hi), "beka" (grant) and "leka" (green bean).
  The *bertsolari* fits the content of what (s)he is going to say around the available rhyming words. Thus, the work of storing, ordering and retrieving such elements from the memory is of considerable importance.
- **Syllable counter**: the syllable counter module checks whether a given sentence meets the meter requirements of the given scheme. For the syllabification itself, the approach described in [14] has been used with some modifications to capture Basque phonology.
- **Verse structure**: as we have seen, each verse contains a pre-established number of lines and these, in turn, consist of a particular number of syllables. Although about a hundred different verse structures can be used, currently only few of them are commonly used. For this work, we used the well-known meter *Zortziko Txikia* (Fig. 2). *Zortziko Txikia* is a composition of eight lines in which odd lines have seven syllables and even ones have six. The union of each odd line with the next even line, form a strophe. Each strophe has 13 syllables with a caesura after the 7th syllable (7 + 6) and must rhyme with the others.

———————— 7
————— 6A
———————— 7
————— 6A
———————— 7
————— 6A
———————— 7
————— 6A

Fig. 2. Structure of the *Zortziko Txikia* meter

V. INTEGRATION OF THE LINGUISTIC COMPONENTS INTO A VERSE-MAKER MODULE

The aforementioned modules have been integrated into a verse-maker module for automatic poem generation. *Bertsos* can be composed in a variety of settings and manners. For our particular challenge we have chosen the popular exercise of "Rhymes Given": the system (or the *bertsolari*) is given the four rhyming words and it is required to compose the *bertso* in *Zortziko Txikia* meter "around" these rhyming words.

The verse generation process for this exercise consists of the following steps:

A Form a sentence repository.
B Prompt the user for the rhymes to be used.
C Generate a *bertso* with the given rhymes.

[4]http://aholab.ehu.es/tiki/tiki-index.php?page=AhoTTS
[5]http://aholab.ehu.es/users/manex/
[6]http://www.tecnalia.com/

A. The sentence repository

This is the pre-processing step of our verse-making module. The sentence repository is a corpus of sentences which will be used in the *bertso* generation process. The sentence selection and extraction process consists of the following two steps:

- Corpus generation: newspaper articles and a small corpora of already-created *bertsos* are used as the source material for our corpus.
- Sentence extraction: sentences that fulfil the meter requirements of the *Zortziko Txikia* (7 + 6 = 13 syllables) are extracted from the corpus.

It must be noted that the use of human generated corpora (oral or text) is of common use in computational poetry [7], [21], [10]. It has the advantage of avoiding the generation of text that is un-interpretable. That is, using human-produced sentences assure the internal coherence of the verses. However, it may be interpreted as plagiarism. Hence, we have chosen initially to work with phrases mined from the Basque newspaper *Berria*[7] alongside sentences extracted from the work of well-known *bertsolaris*[8]. The later reflects the desire to maintain the language-model of the *bertsolaritza*, and the former tends to increase quality while not appropriating text intended for poems.

B. Rhymes for verse generation

In this step, the user is prompted for the four rhymes to be used. The interaction with the system is entirely speech-based: the user gives the rhymes in Basque, and the system answers in Basque too (that occurs when it does not understand correctly what the user said).

C. Bertso generation

Sentence extraction

The four given rhymes are used to find sentences in the corpus ending with those rhymes. During the search process, a set of sentences for each initial rhyme are found. Those sentences will then be fed into the second stage to perform the sentence matching process.

Sentence relation

A system capable of validating its output from a metrical and rhythmical point of view shows a hint of aesthetic sensibility, but it is missing out one important aspect that humans use to value poetry: meaningfulness. The main purpose of this stage is to compare the relationship between pairs of sentences to choose the most semantically related ones for the final verse. To do that, the sentence relation system computes, for each pair of sentences, a score that evaluates how those sentence are semantically related. This stage consists of two main facts:

- Weight the importance of a term in a sentence, based on the statistics of occurrence of the term.
- Represent each sentence as a vector of such weights and compute a score between each pair of sentences. The Vector Space Model (VSM) [15] is used to formulate the relationship between sentences as vectors.

[7]http://www.berria.info/

[8]http://bdb.bertsozale.com

Weighting

How can we evaluate the importance of a given word in a sentence? The well-known *tf-idf* method combines the definitions of the term frequency (*tf*) and the inverse document frequency (*idf*), to produce a composite weight for each term in a document [20]. The *tf-idf* weighting scheme assigns to term t a score in sentence s given by

$$tf - idf_{t,s} = tf_{t,s} \cdot idf_t$$

Where,

- *tf* (term frequency): the number of occurrences of term t in sentence s.
- *df* (document frequency): the number of sentences in the collection that contain a term t.
- *idf* (inverse document frequency): provides information about how a term is distributed among a collection of sentences (N total number of sentences). The less a term appears in a collection of sentences, the higher its value.

$$idf_t = \log \frac{N}{df_t}$$

The assigned score by *tf-idf* is

1) Highest when t occurs many times within a small number of documents.
2) Lower when the term occurs fewer times in a document, or occurs in many documents.
3) Lowest when the term occurs in all documents.

For each sentence, we calculate the *tf-idf* score for each word in the corresponding sentence.

It must be mentioned that Basque exhibits rich morphology. Due to its inflectional morphology, a given word lemma makes many different word forms. A brief morphological description of Basque can be found in [3]. For example, the lemma *etxe* (house) forms the inflections *etxea* (the house), *etxeak* (houses or the houses), *etxeari* (to the house), etc. This means that if we use the exact given word to calculate term weighting, we will loose the similarities between all the inflections of that word. Therefore, we use a stemmer, which is based on the morphological description of Basque to find and use the lemmas of the given words in the term dictionary.

Vector generation

At this point, we may view each sentence as a vector with one component corresponding to each term in the dictionary, together with a weight for each component that is given by the *tf-idf* formula.

Terms that do not occur in a sentence are scored with zero value.

Computing the similarity

To measure the semantic relation between a pair of sentences, we make use of the classical cosine similarity of their vector representations:

$$\cos(\theta) = \frac{V_1 \cdot V_2}{\|V_1\| \cdot \|V_2\|}$$

where the numerator represents the dot product (or inner product) of the vectors,

$$V_1 \cdot V_2 = \sum_{i=1}^{M} V1_i \cdot V2_i$$

and the denominator is the product of their Euclidean lengths.

$$\|V_1\| \cdot \|V_2\| = \sqrt{\sum_{i=1}^{M} V1_i^2} \cdot \sqrt{\sum_{i=1}^{M} V2_i^2}$$

The general algorithm to find the most semantically related sentences to form the final verse is illustrated in Alg. 1.

Algorithm 1 General algorithm to find the verse with the highest cosine similarity

```
Input: r1, r2, r3 and r4 rhymes
Output: the verse with the highest cosine measure
  group1 = sentences that have r1 as rhyme;
  group2 = sentences that have r2 as rhyme;
  group3 = sentences that have r3 as rhyme;
  group4 = sentences that have r4 as rhyme;
  for all s1 in group1 do
    for all s2 in group2 do
      for all s3 in group3 do
        for all s4 in group4 do
          verse = (s1, s2, s3, s4);
          cos12 = cosine similarity(s1, s2);
          cos23 = cosine similarity(s2, s3);
          cos34 = cosine similarity(s3, s4);
          cosall = cos12 + cos23 + cos34;
          store(file, verse, cosall);
        end for
      end for
    end for
  end for
  return find max cosine(file)
```

Given the four rhymes (*r1*, *r2*, *r3* and *r4*) to compose the *bertso*, in the first place, are selected from the corpus those sentences ending with the given rhymes.

In the next step, the nested structure *for* creates all possible combinations of phrases to form a valid *bertso*. For each *s1-s2-s3-s4* combination, the following comparisons are made:

```
s1-s2
s2-s3
s3-s4
```

That is, the overall semantic measure is calculated for each *bertso* by adding up the cosines between the vectors for all pair of adjoining sentences.

This is so because we seek a progressive relationship between sentences. And for the overall coherence of the *bertso*, it is more important the relationship between *s1* and *s2*, than the relationship between *s1* and *s4*.

Finally, the system returns the verse with the highest cosine measure.

It must be noted that for each *bertso* three comparisons are made. For example, if for each (*r1-r4*) rhyme group we have *s=10* sentences in the corpus, then the cosine similarity between two vectors will be calculated *10x10x10x10x3 = 30000* times. As *s* grows, computing the cosine similarities can be computationally expensive. Our system calculates more than 400 similarities per second.

Experimental results

For illustration purposes, we compare a *bertso* composed with our implemented prototype and a *bertso* composed by a *bertsolari*, given the same four rhymes, and making the same calculation.

Rhymes: *lorea* (flower), *gordea* (hidden), *hobea* (better), *ordea* (but)

Bertsobot:

Beloki ez da egun bateko lorea (S1) irla batean zaude bakarti gordea (S2) Nik ez det ezagutzen beste bat hobea (S3) Horrek ere badauka beste bat ordea. (S4)	*Beloki is not a flower of a day you are on a lonely and hidden island I know no better but he also has another one.*

Note: Beloki is a popular Basque surname.

Score: 0.66

Human *bertsolari*:

Bera nere maitea	*He is my beloved*
ta nere lorea	*and my flower*
nere bihotz barnean	*I have saved him*
hor daukat gordea	*in my heart*
inoiz ez det bilatu	*I have found*
mutil bat hobea	*no better boy*
berak ni inoiz ez nau	*but he never*
maitatu ordea.	*loved me.*

Score: 0.57

Comparing robot-generated and human-generated verses, we found that they have similar scores, and what is really more important, both verses have a similar message. But it must be admitted that although in the second verse it is clear to whom the poem is dedicated, in the first it is not so much. But, nevertheless, *Bertsobot* has maintained a certain consistency between the sentences that make up the poem and a quite acceptable overall meaningfulness.

VI. Integration of the components into a real robot

So far we have described the tools used for creating verses. But *bertsolaritza* belongs to the oral genre, an thus, needs to be performed in public. Therefore, it is not enough to improvise poetry, the created poem has to be part of a performance. Thus, the *Bertsobot* system needs additional functionalities, such as a robotic body to interact with the environment and display emotional body-language [6], and a synthesized voice.

In this last step we have integrated the automatic verse-making system in a pair of real autonomous robots.

- ***Galtxagorri*** is a *Pioneer* 3-DX robot, a very common platform in mobile robotics. Galtxagorri is provided with a 1.6 GHz onboard computer with 1GB RAM memory. Its sensor equipment consists of 16 ultrasound sensors, a Canon PTZ colour camera, a Leuze RS4 laser sensor and a TCM2 electronic compass. For the task in hands, we added to the platform speakers and a microphone.
- ***Tartalo*** is a *PeopleBot* robot built on the P3-DX base. More specifically, *Tartalo* is a *Pioneer* robot with a chest-level extension that makes it taller and adequate for human-robot interaction. It has an 1.8 GHz computer with 1 GB RAM memory and is provided with sonar and bumper sensors, a gripper with two degrees of freedom, a SICK laser scanner, a Canon PTZ colour camera, a TCM2 compass sensor and a touchscreen on the top. Taking into account the adequateness for human-robot interaction, the robot is also equipped with speakers and a microphone.

Fig. 3 shows a picture of the robots.

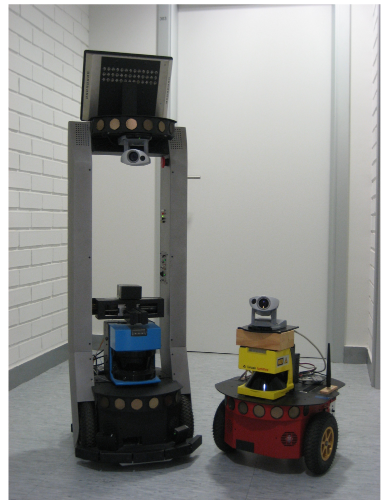


Fig. 3. *Tartalo* and *Galtxagorri* autonomous robots

The *Bertsobot* system can be run in a simple computer, but as mentioned before, *bertsolaritza* implies interaction with the environment, such as the other *bertsolaris* of the performance and the audience. In improvised oral festivals, the *bertsolari-s* wait sitting their turn to sing, and when their turn comes, (s)he has to approach the microphone that is placed in front of them. Once in their location, they have to look the theme-prompter, that will prescribe a topic which serves as a prompt for the *bertso* (in our case the four rhymes to compound the *bertso*). After that, they have a specific time to create the verse and finally sing it. Once finished, they await a few seconds to perceive the audience's reaction and they go back to their chair.

In order to represent as natural as possible the overall performance, we have developed a set of movements for the robots:

1) The robot is started by pressing a button. Then it moves towards the microphone.
2) Once in front of the microphone, it points the camera at the theme-prompter awaiting his instructions.
3) When it receives the instructions, the robot points the camera towards the audience, and starts moving the camera up and down, left and right mimicking the movements the *bertsolaris* make while thinking what to say.
4) Once the system has the text of the verse, it starts singing while balancing its body to the rhythm of the melody.
5) At the end of singing, the robot waits a while to receive the applause (or whistles) of public and retires to his initial position.

The aim of the above described movements is to represent as faithfully as possible *bertsolari*'s movements on stage, interacting as much as possible with the other improvisers and the audience.

The interaction with the system is entirely speech-based. The user (theme-prompter) gives the instructions in Basque (the four rhymes to compound the *bertso*), and *Bertsobot* answers in Basque too.

System architecture The system makes use of the following modules:

- A Basque Text-To-Speech synthesizer (TTS), developed by Aholab.
- A Basque singing synthesis system, developed between Aholab and IXA.
- A Basque Automatic Speech Recognition (ASR) system, integrated by Robotiker[9].
- A Basque verse generation system, developed by the IXA group and Robotics and Autonomous Systems group.
- A robotic behavioural system, developed by the

[9]http://www.tecnalia.com

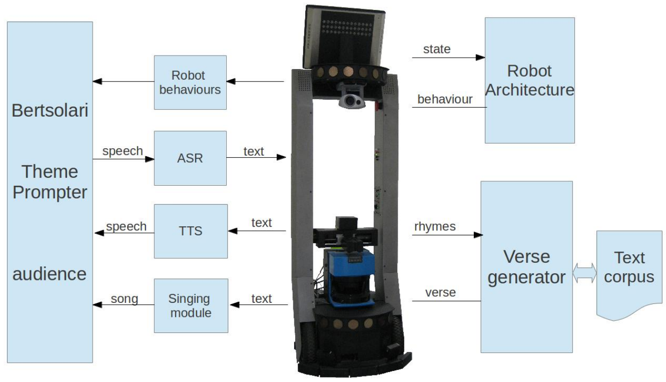


Fig. 4. System architecture

Robotics and Autonomous System group.

Fig. 4 illustrates how the different modules interact with each other and with the user.

VII. Dissemination

At the very beginning of this project, we were invited to make a public demonstration: a kind of duel between robots and real *bertsolari-s*. Four *bertsolaris* and two autonomous robots participated in the performance (see fig 5). We took the opportunity to present in public our project so as to make a didactic demonstration of what a real robot can do actually in *bertso* composition. The performance aroused great interest, and almost all newspapers, radio and television covered the event. Videos of the public performance can be found in our website [10].

We also showed the *Bertsobot* project to the general public during the Week of Science, Technology and Innovation 2012[11] and in Xare 2012[12] conference, but only the computer-based prototype was showed, because the management and transportation of robots was too complicated.

Students from the schools and members of the public in general had the chance to try it out and play with it, and they were generally surprised and interested.

VIII. Conclusions and future work

In this paper, we have explained our first steps forward the development of an autonomous minstrel robot, called *Bertsobot*. The applied verse-generation algorithm is based on VSM to model semantics and makes use of different speech and text technologies to automatically generate verses. The whole system has been integrated into a real robot, showing the degree of expressiveness that show the *bertsolaris* in their performance.

It is clear that many aspects of the system presented here are fairly rudimentary. Our contribution has been twofold: on the one hand we implemented an embodied system which can be taken seriously as a *bertsolari*, albeit in a simplistic manner. On the other hand, we presented a verse-maker module which composes verses automatically, taking into account the semantic relation of sentences within a *bertso*.

We plan further enhancements to each of the processes involved, including:

1) Implementing improved methods to generate phrases for templates.
2) Working with other corpora.
3) Exploring new models to measure the semantics relationship between sentences.
4) The development of an 3D Avatar to display facial

Fig. 5. verse-duel between Maialen Velarde and the robots

[10] http://www.sc.ehu.es/ccwrobot/videos.html
[11] http://www.zientzia-astea.org/
[12] http://www.xare.eu/albisteak/

expressions.

5) Enabling the robot to adequate the message of the poem based on the reactions perceived from the public.
6) Improve the emotional body language of the robot and increase its degree of autonomy while acting the performance. For the time being, we are trying to identify the microphone using a Kinect, so the robot will know where it is and approach the microphone.
7) Identify the faces of the participants (*bertsolari-s* and the theme-prompter). Thus, the robot will turn exactly towards their position, or mention their name in the improvised verse, as it occur sometimes.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge *Bertsozale Elkartea*[13] (Association of the Friends of Bertsolaritza), whose verse corpora has been used to test and develop the *Bertsobot* project.

REFERENCES

[1] Manex Agirrezabal, Inaki Alegria, Bertol Arrieta, and M. Hulden. BAD: An assistant tool for making verses in Basque. *EACL 2012*, page 13, 2012.

[2] Manex Agirrezabal, Iaki Alegria, Bertol Arrieta, and Mans Hulden. Finite-state technology in a verse-making tool. In *Proceeding of the 2012 conference on Finite-State Methods and Natural Language Processing*, 2012.

[3] Iñaki Alegria, Xabier Artola, Kepa Sarasola, and Miriam Urkia. Automatic morphological analysis of Basque. *Literary and Linguistic Computing*, 11(4):193–203, 1996.

[4] Xabier Amuriza. *Hiztegi errimatua [Rhyme Dictionary]*. Alfabetatze Euskalduntze Koordinakundea, 1981.

[5] Bertol Arrieta, Inaki Alegria, and Xabier Arregi. An assistant tool for verse-making in Basque based on two-level morphology. *Literary and linguistic computing*, 16(1):29–43, 2001.

[6] A. Beck, A. Hiolle, A. Mazel, and L. Cañamero. Interpretation of emotional body language displayed by robots. In *Proceedings of the 3rd international workshop on Affective interaction in natural environments*, pages 37–42. ACM, 2010.

[7] S. Colton, J. Goodwin, and T. Veale. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 2012.

[8] B. Díaz-Agudo, P. Gervás, and P. González-Calero. Poetry generation in colibri. *Advances in Case-Based Reasoning*, pages 157–159, 2002.

[9] Daniel Erro, Iñaki Sainz, Ibon Saratxaga, Eva Navas, and Inma Hernáez. Mfcc+ f0 extraction and waveform reconstruction using hnm: preliminary results in an hmm-based synthesizer. *Proc. FALA*, pages 29–32, 2010.

[10] P. Gervás. Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, pages 23–30. Association for Computational Linguistics, 2010.

[11] Pablo Gervás. Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100, 2000.

[12] Pablo Gervás. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*, 14(3):181–188, 2001.

[13] Inma Hernaez, Eva Navas, Juan Luis Murugarren, and Borja Etxebarria. Description of the ahotts system for the basque language. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[14] Mans Hulden. Finite-state syllabification. *Finite-State Methods and Natural Language Processing*, pages 86–96, 2006.

[15] D.L. Lee, H. Chuang, and K. Seamons. Document ranking and the vector-space model. *Software, IEEE*, 14(2):67–75, 1997.

[13] http://www.bertsozale.com/en

[16] I. Leturia, A. Del Pozo, K. Arrieta, U. Iturraspe, K. Sarasola, A. Ilarraza, E. Navas, and I. Odriozola. Development and evaluation of anhitz, a prototype of a basque-speaking virtual 3d expert on science and technology. In *Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on*, pages 235–242. IEEE, 2009.

[17] Robert P. Levy. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proccedings of the ICCBR-01 Workshop on Creative Systems*. Citeseer, 2001.

[18] Ruli Manurung. *An evolutionary algorithm approach to poetry generation*. PhD thesis, School of informatics, University of Edinburgh, 2003.

[19] H. Oliveira. Automatic generation of poetry: an overview. *Universidade de Coimbra*, 2009.

[20] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[21] M.T. Wong, A.H.W. Chun, Q. Li, SY Chen, and A. Xu. Automatic haiku generation using vsm. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, number 7. World Scientific and Engineering Academy and Society, 2008.

Hasi, *BertsoBot*

A. Astigarraga eta **I. Rodriguez**

Konputazio Zientziak eta Adimen Artifiziala departamendua, Informatika fakultatea (UPV/EHU)

Laburpena

Bertsotan aritzeko gaitasuna erakutsiko duen robot autonomoa garatzea da gure ikerketalanaren helburu behinena. Bere egitekoa, bertsoa osatzeko instrukzioak ahoz jaso, hauek prozesatu eta ahalik eta bertsorik egokiena osatu eta kantatzea litzateke, bertsolarien oholtza gaineko adierazkortasun maila erakutsiz gorputzarekin. Robot-bertsolariak, gizaki eta roboten arteko elkarrekintza eta komunikazioan aurrera egiteko modua jarri nahi luke, lengoaia naturala erabiliz robot-gizaki arteko bi noranzkoko komunikazioan.

Hitz gakoak: Gizaki-Robot Elkarrekintza, Lengoaia Naturalaren Sorkuntza

Abstract

The mail goal of our research project is to develop an autonomous robot capable of improvising verses in Basque. The robot should be able to receive the commands verbally, process them, generate the most appropriate verse and sing it with the proper melody. The robot should also show the expressiveness that bertsolari-s show in their performance. Our minstrel robot aims to contribute research on human-robot interaction and communication, using natural language as a two-way communication between humans and robots.

Keywords: Human-Robot Interaction, Natural Lenguage Generation

1 Sarrera eta arloko egoera

1769. urtean Von Kempelen baroiak, makinen historiako inbentziorik handiena bezala aurkeztu zuen Xake Jokalari Automata, gerora itxuragatik Turkiarra ezizenaren jabe egingo zena. 80 urtetan zehar, Europan eta Estatu Batuetan barrena xakean bere abilezia erakutsi zuen Turkiarrak, jokatutako partida gehientsuenak irabaziz. Aurkarien artean, garaiko xake jokalari onenez gain, Napoleon Bonaparte bera ere topa dezakegu.

Kontuan izan behar dugu automaten urrezko garaian gaudela, giza izatea imitatzeko ahaleginetan sortutako makinen garaian alegia. Ez da harritzekoa Turkiarraren aurkezpenak ikaragarrizko zalaparta eragin izana. Bere existentziak makinen ahalmenen inguruan aukera berriak zabaltzen zituen, adimena bera ere bai agian. Baina artefaktua zen, duda gabe: piezak modu mekanikoan mugitzen zituen, eta arreta jarriz gero, mahai azpiko engranajeen soinua adi zitekeen. Momentua iristean, *xake* hitza ahoskatzeko gai ere bazen.

Egungo talaiatik begiratuz gero, nabaria da bazela zerbait arraroa Turkiarrarekin. Xake jokoaren konplexutasunari, mugimenduak atzeman eta piezak mugitzeko gaitasuna gehitu behar zaizkio, abilezia gehiegi garai hartarako, gailu teknologiko aurreratuenak erloju mekanikoa eta automatismo txikiak zirela kontuan hartuz gero. Noski, ez zen inoiz jokalari mekanikorik existitu. Oso ondo prestatutako antzerkia izan zen. Tramankuluaren barruan, garaiko xake-jokalari onenetakoa zegoen ezkutatua, makinari bere talentu eta adimena jarriz (Standage, 2002).

Berrehun urte pasa behar izan dira halakorik benetan gerta dadin, Industri Iraultza, konputagailuen hedapena eta Adimen Artifizialaren garapena tarteko. 1997. urtean gertatu zen, Deep Blue makinak Gary Kasparov garaitu zuen urtean . Hala ere, Turkiarraren istorioak lezio garrantzitsua ekarri zigun: gure asmakuntzen abilezia eta irismena exajeratzeko joera daukagun neurri berean gutxiesten dugula zenbait ahalegin zientifikoren konplexutasuna. Robotikaren historia eta mugarriez gehiago jakin nahi duenak, (Astigarraga eta Lazkano, 2011) lanean topatuko du laburpen osoagoa.

Gure txikian, ahalegin berean murgilduta gabiltza proiektu honekin: gailu bakarrean integratu nahian goi-mailako arrazonamendua, munduaren pertzepzioa eta bertan eragiteko gaitasuna. Xakea albo batera utzi dugu bestelako erronka jarriz mahai gainean, jolastiagoa, arrazonamendu hutsaz gaindiko gaitasunak eskatzen dituena, hizkuntzaren erabilera kasu. Gure asmoa, bertsotan aritzeko gaitasuna erakutsiko duen robot autonomoa garatzea da, *BertsoBot*-a alegia. Bere egitekoa, bertsoa osatzeko instrukzioak ahoz jaso, hauek prozesatu eta ahalik eta bertsorik egokiena osatu eta kantatzea litzateke, gorputzarekin bertsolarien oholtza gaineko adierazkortasun maila erakutsiz.

BertsoBot proiektuak bi alor hartzen ditu bere baitan. Alde batetik, bertsoaren, testuaren sorkuntza; eta bestetik, ingurunearen pertzepzioa eta jendearekiko elkarrekintza. Bi eremu horietako egungo ikerketaren berri ematen ahaleginduko gara labur ondoko lerroetan.

Azken hamarkadan gizaki-robot elkarrekintza eremuan, *Human Robot Interaction* (aurrerantzean HRI) gaitasunak erakusten dituzten zenbait sistema robotiko azaldu diren arren, (Belpaeme *et al.*, 2012) eta (Fasola eta Mataric, 2012), oraindik autonomia maila txikia dute robotek. Autonomia hitz potoloa da robotikan eta teleoperazioaz haragoko portaerak barneratzen ditu (ikasteko ahalmena, autoelikadura, etab.). Teleoperazio terminoak robotaren urruneko kontrola adierazten du. Sistema telerobotiko batean, operadorek (gizakia), robotarengandik distantzia jakin batera, robotaren mugimenduak kontrolatzen ditu; alde batetik kontrol-seinaleak (aginduak) bidaliz, eta, bestetik, agindu horiek ondo bete diren edo ez seinaleak jasoz (*feedback* seinaleak). Urrunetik kontrolatu daitezkeen robotak gure gizarteko sektore askotan erabiltzen dira. Adibidez, medikuntzako ebakuntzetan (Doarn *et al.*, 2007) (Ceccarelli *et al.*, 2013), espazioaren esplorazioan (Badger *et al.*, 2012) edo zentral nuklearrak arakatzeko (Nagatani *et al.*, 2013). Gurean, Microsoft XBOX-eko Kinect sentsorea erabiliz mugimenduak jaso eta robotean denbora errealean errepikatzeko gai den sistema bat aurkezten da (Rodriguez *et al.*, 2014) lanean. Autonomia-maila areagotzeko ezinbestekotzat daukagu roboten eta gizakien arteko elkarrekintza lantzea, batez ere ahots bidezko komunikazio eta gorputz-adierazpide eremuak. Bertsolaritzak paregabeko agertokia eskaintzen du horretarako, eta robot-bertsolariak HRI arloan egindako aurrerapenak gizarteratzeko aukera ematen digu.

Bestalde, konputagailu bidezko poesia sorkuntzak ikerketa komunitatearen atentzioa lortu du azken urteetan, eta sistema aski interesgarriak garatu dira. Gaiari buruzko ikuspegi orokor baterako, (Gervás, 2013) eta (Oliveira, 2009) gomendatzen dira. Nabarmenenak aipatzearren: (Wong *et al.*, 2008) lanean, blogetatik erauzitako testuekin Haikuak eraikitzen dituen sistema azaltzen da. (Colton *et al.*, 2012) lanean berriz, testu corpusak oinarri hartu eta simil bidezko poesia eraikuntza proposatzen du egileak. Azkenik, (Gervás, 2013) lanean, egileak Adimen Artifizialeko teknika ezberdinak konbinatzen dituen WASP sistema ezaguna azaltzen du.

Aipatu berri ditugun lanek erakusten duten gisan, ertz ezberdinetatik heldu zaio poesi sorkuntza automatikoaren gaiari. Halabaina, guk dakigula behintzat, oraindik ez du beste inork poema sortzailerik robotean inplementatu. Baina, dakigun bezala, bertsolaritza ahozko generoan kokatzen da, eta jendeaurreko aurkezpenak berebiziko garrantzia dauka. Hori dela eta, bertsoak sortzeko sistema automatikoa ez da nahikoa, pertsonekin elkarreraginez, jende aurrean bertsoa sortu eta kantatuko duen gorpuzkera beharrezkoa da, robota alegia.

Artikuluan bertso-saio batean parte hartuko duen robota aurkeztuko dugu beraz, atazak eskatzen dituen portaera guztiak barne hartzen dituena: gai-jartzailearen aginduak jaso eta prozesatu, mikrofonoa detektatu eta hara hurbildu, bat-batean bertsoa sortu eta kantatu, eta gorputz adierazpidez mezua indartu publikoarekin elkarreraginez.

Esperimentuetarako, Aldebaran Robotics etxeak garatutako *NAO*[1] giza itxurako robota erabili dugu. 58 cm-ko altuera eta 4.8 kg pisatzen dituen robot humanoide programagarria. Robotaren giltzadurak kontrolatzeko 25 serbomotore ditu eta hauen kontrol estuari esker, oreka mantendu eta bi hanken gainean ibiltzeko gai da.

[1]https://www.aldebaran.com/en/humanoid-robot/nao-robot

2 Motibazioa eta ikerketaren helburuak

2.1 Motibazioa

RSAIT[2] ikerketa taldeak, robotikaren baitako beste alor batzuk landu ditu orain artean, batez ere: kontrol-arkitekturak, nabigazioa, ikasketa automatikoa eta ikusmen artifiziala. Laborategi inguruetan modu seguruan nabigatzea izan da gure helburu nagusietakoa, oztopoak ekidin, helburuak identifikatu, leku berriak esploratu, etab. Lan honekin, aurretik egindakoari geruza berri bat erantsi nahi genioke, gizaki-robot arteko elkarrekintzarena. Izan ere, laborategietatik atera eta ingurunea gurekin konpartituko badute, gure agindu eta mugimenduak interpretatu eta modu egokian erantzuteko gaitasuna behar dute robotek.

Elkarrekintza eta komunikazio horretan robotak portaera edo gaitasun ugari agertu behar ditu denbora errealean: hizketaren ezagutza, aginduen identifikazioa, aurpegien detekzio eta jarraipena, oztopoen detekzioa, gorputz espresioa, etab. Aldi berean eta modu koordinatuan exekutatu beharreko gaitasun ugari inondik ere.

Robotak laborategietatik atera eta jendearen zerbitzura jartzea dugu motibazio nagusi. Dela museoetan gida lanak egiteko, adineko edo ezinduei erreabilitazio lanetan laguntzeko edo, zergatik ez, astialdian lagungarri izan daitezen. Motibazio honek, bidenabar, bat egiten du Europako Batzordeak robotika mugikorraren eremuan zehaztutako 2020 estrategiarekin[3] .

2.2 Ikerketaren helburuak

Helburu orokorra ez da bertsotan aritzeko gaitasuna erakutsiko duen robot-autonomoa garatzea soilik. Bertsolaritzaz gaindi, gizaki-robot arteko komunikazioan pauso bat egin nahi genuke aurrera, mezuak sortu eta plazaratzeko orduan testuinguruko ezaugarriak kontuan hartuko dituen sistema eraikiz. Proiektu handinahia da, nabarmen, baina bere baitan helburu xumeago eta konkretuagoak biltzen dituena:

Bertso-sorkuntzari dagokionez:

- Bertso-sorkuntza automatikoa, testu corpus batetik abiatuta
- Bertsoaren semantika eta egitura narratiboa aztertu eta eredu konputazionalak eraiki

Robotikari dagokionez:

- Gizaki-robot arteko komunikazioa lengoaia naturala erabiliz
  - Testuinguru mugatu batean gai-jartzaileak/bertsolagunak esandakoa ulertu
  - Ahotsaren bitartez mezuaren erritmoa, entonazioa, bizitasuna jaso
- Kinect tresna erabiliz, bertsolariaren keinuak identifikatu eta bere mugimenduak jarraitu
- Robotaren portaera gidatzeko elkarrizketa-erregelen erabilera.
- Ikusmen Artifizialeko estrategiak erabiliz:
  - Kantulagunaren aurpegia identifikatu eta espresioa ezagutu
  - Publikoaren erreakzioak identifikatu
- Humanoide gisako roboten portaerak lantzea

3 Ikerketaren muina

Bertsolariek, plazan zein txapelketan, ariketa ezberdin ugari burutu behar izaten dituzte. *BertsoBot*-ak, momentuz, haietako bi baino ez ditu egiten: 4 oinak emanda bertsoa osatu eta puntuari erantzuna eman. Bakoitzak bere berezitasunak dauzkan arren, erabilitako tresna eta estrategiak oso antzekoak dira bi kasuetan. Beraz, oinak emandako ariketa hartuko dugu erreferentzia gisa ondoko azalpenetarako.

[2]http://www.sc.ehu.es/ccwrobot
[3]http://ec.europa.eu/digital-agenda/en/robotics

Ariketa honetan, gai-jartzaileak 4 oinak eman banan banan eta robotak oin horiekin bertsoa osatu eta kantatu behar du zortziko txikiko moldean (13 silabako puntuak, 7-6 egitura).

Atazaren ebazpena bitan banatu dugu: lehenbizikoan bertsoaren testua sortzeko tresna eta estrategiak izango ditugu aztergai, eta bigarrenenean, ariketak eskatzen duen gizaki-robot arteko elkarrekintza.

3.1 Bertsoaren sorkuntza

Bertsolarien lan-tresna garrantzitsua da memoria, han gorde eta sailkatzen dituzte entzundako bertso, irakurritako testu eta ikusitako irudiak, abagune egokia iristean bertso-moldera ekarri ahal izateko. Bertsolarien bat-bateko jarduna ez baita momentuan sortzea bakarrik, etxean landutakoa egoki txertatzea ere bada (Gartzia *et al.*, 2001).

BertsoBot-arentzat ere funtsezkoa da memoria. Bi testu-corpusek osatzen dute berea: Egunkariako 2000-2001 urteetako testuekin osatutakoa batetik (%85eko pisua); eta, bestetik, 1986-2009 arteko Bertsolari Txapelketa Nagusiko bertsoez osatutakoak (%15eko pisua). Corpusak esalditan (bertsoak puntutan) gorde dira. Zergatik esalditan? Bada barne koherentzia bermatzen duen egitura minimoa delako. Horrela, miloi bat esalditik gorako corpusa baliatzen du *BertsoBot*-ak, testu masa horrek osatzen du bere memoria.

Bertsoa osatzeko estrategiari dagokionez, bertso-sorkuntza moduluak sarrera gisa bertsoa osatzeko oinak jaso eta ondoko lanak burutzen ditu segidan:

1. Oin horiekin amaitzen diren esaldiak corpusetik erauzi
2. Baldintza metrikoak betetzen dituztenak (7+6 silaba) aukeratu
3. Ausaz oin-multzo bakoitzetik puntu bat aukeratu bertsoa osatzeko

Horra 4 oinak emanda teknikoki zuzena den bertsoa osatzeko estrategia. Bi tresna garatu ditugu zeregin horretarako: silaba-kontatzailea eta errima-bilatzailea (Astigarraga *et al.*, 2013). Baina tamalez, gehien-gehienetan, osatutako bertsoak ez du ez hanka eta ez buru. Puntuen arteko lotura, kohesioa, falta izaten du bertsoak. Zenbaki bidez azaltzeko, oin oso bereziak ez badira behintzat, sistemak gutxienez 10 puntu itzultzen ditu sarrera gisa emandako oin bakoitzeko. Beraz, guztira, *10 x 10 x 10 x 10 = 10000* bertso-konbinazio ezberdin! Guzti horietatik ausaz bat aukeratzen du sistemak, eta espero litekeen bezala, emaitza elkarrekin loturarik ez daukaten puntuez osatutako bertsoa izan ohi da.

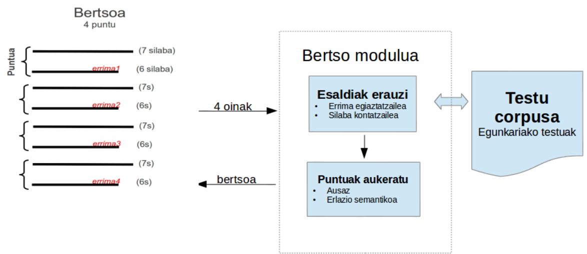
Horixe gure hurrengo urratsa, 10000 bertso-konbinazio ezberdin horietatik puntuen arteko kohesiorik handienekoa hautatzea. Nola egin? Bi metodo inplementatu ditugu horretarako:

Vector Space Model (VSM) (Lee *et al.*, 1997): Puntuak bektore gisa adierazi, hitz bakoitza bektoreko elementu delarik, eta bektoreen, puntuen, arteko kosinu bidezko distantzia kalkulatzen du metodo honek. Interpretatzen oso erraza da: Bektoreen artean zenbat eta hitz kointzidente gehiago izan, orduan eta elkarrengandik semantikoki hurbilago puntuak. Hitz edo termino guztiek, noski, ez dute pisu bera testuan. Pisuak erabakitzeko, *tf-idf* (Ramos, 2003) pisaketa-sistema erabili dugu. VSM metodoaren eragozpenik handiena, antzekotasuna terminoen kointzidentzian soilik oinarritzen duela da. Izan ere, gai beraren bueltan aritu arren termino ezberdinak erabiltzen dituzten bi testu ez lituzke antzekotzat joko. Hau da, polisemia eta sinonimia ez ditu tratatzen.

Latent Semantic Analysis (LSA) (Landauer *et al.*, 2013): VSMren hurbilketa berezia, aipatutako arazoak ekiditeko sortua. Funtsean, testu idatzien semantika adierazteko gaitasuna duen tresna da LSA, terminoen arteko erlazio semantikoen neurria itzultzen duena. Horretarako matrizeen algebra erabiltzen da, SVD izeneko matrize-banaketa, eta emaitza gisa LSA-ak terminoen kointzidentziaz gaindi, esaldien arteko lotura semantikoaren neurria itzultzen du.

Azaldu berri ditugun bi metodoekin bertsoaren barne kohesioa nabarmen hobetzen da, Gipuzkoako Bertsozale Elkarteko epaileekin egindako ebaluazioan agertu zen bezala (Astigarraga *et al.*, 2014). Bertsoaren sortze-prozesua ulertzen lagungarri izango delakoan, 1a irudian sorkuntza moduluaren adierazpen grafikoa topatuko dugu. Horrekin batera, adibide gisa (ikus 1b), *dakar, zakar, azkar* eta *alkar* oinekin osatutako bi bertso ageri dira. Ezkerraldekoan, puntuen arteko lotura ausazkoa da, eta eskuineko bertsoan aldiz, LSA metodoa erabili da puntuen arteko lotura semantikoa neurtzeko.

Egun, LSA metodoaren doitze-lanetan gabiltza. Praktikak erakutsi digu zenbat eta corpus konkretuagoak erabili (gaiari dagokionez), erlazio semantikoaren neurriak orduan eta zehatzagoak direla. Hortik abiatuta, corpus zehatzagoekin gai jakinei buruzko bertsoak sortzeko ildoa landu nahi dugu.



(a) Sorkuntza modulua

Horrek aberastasun
kulturala dakar,
anaiak alkarrekin
gabiltz oso zakar,
eta ospitalean sartu
ziren oso azkar,
nahiz ta bertsolariak
hartu degun alkar

Inbiriak bihurtzen
gaitu nahiko zakar,
galdu gera ez badeu
guk hartutzen alkar,
eta gure etsaiak
bidaltzea azkar,
ondoren onik horrek
guretzat ez dakar

(b) Adibidea: 4 oinak emanda

1 Irudia: Bertsoaren sorkuntza

3.2 Bertsoaren antzezpena

Bertsoa automatikoki sortzen duen sistemak ez du robotaren beharrik, edozein ordenagailutan exekutatu daiteke. Baina bertsolaritzan jendaurreko aurkezpenak berebiziko garrantzia dauka, bertsolari eta ingurunearen –gai-jartzailea, mikrofonoa, publikoa, gainontzeko bertsolariak, etab.– arteko elkarrekintza testua bezain garrantzitsua da. Horretarako ezinbestekoa da gorpuzkera fisikoa, robota gure kasuan. Bertso-saioen dinamika orokorra jarraitu behar du *BertsoBot*-ak. Bertsolariek egiten duten legez, lehenik, aulkian eserita bere txanda itxaron behar du. Txanda iristean aurrean kokaturik duen mikrofonora gerturatu behar da eta gai-jartzaileari begira jarri, bera izango baita ariketa jarriko diona. Ondoren, denbora tarte arrazoizko batean (1min. gehienez) bertsoa sortu eta kantatuko du. Publikoaren erreakzioaz busti eta, azkenik, bere aulkira bueltatuko da. Gutxi-asko horixe da jendaurreko antzezpena. Itxuraz sinplea dirudien zeregina burutzeko, kontrol-sistemak hainbat ataza exekutatu behar ditu, aldiberean batzuetan (paraleloan), modu sekuentzialean bestetan, robotaren portaera global koherentea gertatzeko. Modulu edo ataza horiek ondo koordinatu eta komunikatu behar dira, eta hori da azken batean kontrol-arkitekturen zeregina.

BertsoBot-aren kontrol-arkitekturak Portarean Oinarritutako Sistemen ereduari jarraitzen dio (Brooks, 1999). Finean, elkarren artean komunikatu eta eragileak atzitzeko koordinatu behar diren moduluek portaera-sare bat osatzen dute (ikus 2. irudia) eta etengabe paraleloan exekutatzen ari dira. Modulu edo portarera horiek honakoak dira:

- Ulermen modulua (Automatic Speech Recognizer, ASR)
- Ahots modulua (Text-To-Speech, TTS)
- Identifikazio modulua
- Sorkuntza modulua
- Kantu modulua
- Mugimendu modulua

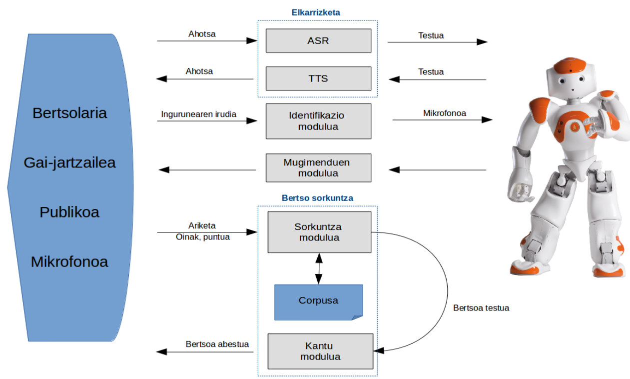
Inplementazioari dagokionean, ROS (Robot Operating System) *framework*-a (Quigley *et al.*, 2009) erabili da, moduluen arteko komunikazioa eta kudeaketa erabilterrazagoa bihurtzen duena. Egun, robotak programatzeko zabalduen dagoen iturburu irekiko software-tresna da ROS.

3.2.1 Ahots (TTS) eta Ulermen (ASR) moduluak

Gizaki eta roboten arteko harremana nabarmen hobetuko litzateke robotekin lengoaia naturala erabiliz, ahots bidez, komunikatu ahal bagina. Horretarako, robotek, entzuteko zein hitzegiteko gaitasuna erakutsi behar dute. Zer esanik ez entzuten dutena ulertu eta erantzuna arrazoitzeko, baina azken hau zaila da oraindik.

ASR eta TTS robotarekin hizketa bidez komunikatu ahal izateko moduluak dira. ASRak ahotsa testu

2 Irudia: Sistemaren arkitektura



bihurtzen du eta TTSak testua ahots. ASR sistema gisa *Google Speech API*-a[4] erabiltzen dugu SOX[5] tresnarekin batera. Sistemak sarrera gisa audio fitxategi bat espero du, irteera bezala audioari dagokion hipotesia testu formatuan eta dagokion konfidentzia mailarekin itzuliz. Bestetik, robotari ahotsa jartzeko Aholab-eko *AhoTTS* ahots sintetizadorea erabili da (Hernaez *et al.*, 2001). Ahots sintetizadore honek euskaraz, gazteleraz zein ingeleraz ahotsa sortzeko aukera eskaintzen du.

3.2.2 Bertso sorkuntza eta kantu moduluak

Gai-jartzaile eta *BertsoBot*-aren arteko elkarrekintza, ahots bidezko elkarrizketa-erregela bidez gidaturikoa da: Gai-jartzaileak ariketa-mota (4 oinak emanda) zehaztu ostean, banan banan bertsoa osatzeko oinak ematez dizkio. Robotak hitzen bat ondo ulertu ez badu errepikatzeko eskatuko dio. Bestela, Sorkuntza moduluari 4 oinak bidali eta honek bertsoaren testua itzuliko du, 3.1 atalean azaldu bezala. Bertsoa **Kantu moduluari** pasako zaio segidan (Agirrezabal *et al.*, 2012), zeinak dagokion moldeko bertso-doinua aukeratu eta kantatzen duen.

3.2.3 Identifikazio modulua

Bertsolariek oholtza gaineko elementu desberdinetan zentratzen dute atentzioa, uneko helburuak eskatzen duenaren arabera: gai-jartzailea, beste bertsolariak, esertzeko lekua, mikroa, etab. Mikroaren kokapena errererentziazko lekua da, bertara gerturatu behar baitira beraien lana egitera. Robotak berdin egin beharko du. Ikusmen artifizialeko teknikak erabili ditugu mikroa non dagoen finkatu eta bere jarraipena egiteko robota mugimenduan dagoenean. Mikroa, noski, robotaren morfologiara egokituta daukagu (ikus 3 irudia).

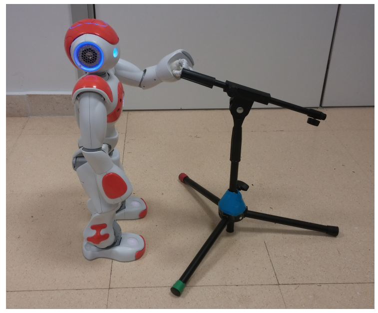
3.2.4 Mugimenduen modulua

Bertso saio batean bertsolariek erakusten duten keinu bidezko lengoaia erreproduzitzea da modulu honen ataza nagusia. Horretarako, bertsolari batek oholtza gainean egin ohi dituen mugimenduak aztertu (publikoari begiratu, bertsoa pentsatzerakoan eskuak atzean jarri, gora ta behera begiratu, kantatzerakoan gorputzarekin egiten dituzten mugimenduak, etab) eta modulu honetan inplementatu dira. Gainontzeko moduluekin komunikatzen da, mugimendu bakoitza dagokionean exekutatuz.

Azaldutako moduluen exekuzio koordinatuak, robotaren portaera orokorra sortzen du: txanda itxoin, gaia jaso, bertsoa sortu eta publikoari begira jarriz kantatzeko gai den robota. Aipatu beharra dago, 2012an proiektuari ekin genionetik jendaurreko 5 saio egin ditugula, bertsolariekin bi aldiz eta gainontzekoetan zientzia eta teknologiaren dibulgaziorako prestatutako ekitaldi gisa. RSAITen webgunean[6] topa ditzakezue saioei buruzko aipamen eta bideoak.

[4]https://www.google.com/speech-api/v2/recognize
[5]https://sox.sourceforge.net
[6]RSAIT webgunea. http://www.sc.ehu.es/ccwrobot/seccion/sarrera

3 Irudia: NAO mikrofonoaren aurrean



4 Ondorioak eta etorkizunerako norabidea

Artikulu honetan *BertsoBot* sistema aurkeztu dugu, jendaurrean bertsoak sortu eta kantatzeko gai dena. Ikerketa-lanaren ekarpenei dagokionez, testu-sorkuntzaren arloan barne kohesiorako VSM eta LSA metodoen erabilera azpimarratuko genuke. Gizaki-robot arteko elkarrekintzan aldiz, euskaraz hutsez gidatutako elkarrizketa-sistema eta ROS-en oinarritutako kontrol-arkitektura orokorra lirateke gure ekarpenak. Etorkizunerako lanei dagokienez, dagoeneko lanean gabiltza ondoko hobekuntzak inplementatzeko:

1. Internet erabili bertsoak osatzeko testu corpus gisa
2. Esaldien arteko erlazio semantikoa neurtzeko eredu berriak aztertu
3. Publikoaren erreakzioaren arabera (txalo/txistu), bertsoaren mezua aldatu
4. Ahotsaren iturburua identifikatuz, gai-jartzailearen kokapena zehaztu
5. Kinect bidezko teleoperazioa erabilita, robotari portaera berriak irakatsi

Noski, hobekuntza bakoitza ROS-eko kontrol-arkitektura orokorrean integratu behar da, gainerako moduluekiko koordinazioa eta erantzun-denbora egokia bermatuz. Moduluz modulu, puntuz puntu osatu beharreko ariketa.

Hasi, *BertsoBot*.

Erreferentziak

Agirrezabal, Manex, Inaki Alegria, Bertol Arrieta, eta Mans Hulden. 2012. Bad: An assistant tool for making verses in basque. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 13–17. Association for Computational Linguistics.

Astigarraga, A, E Jauregi, E Lazkano, eta M Agirrezabal. 2014. Textual coherence in a verse-maker robot. In *Human-Computer Systems Interaction: Backgrounds and Applications 3*, 275–287. Springer.

Astigarraga, Aitzol, Manex Agirrezabal, Elena Lazkano, Ekaitz Jauregi, eta Basilio Sierra. 2013. Bertsobot: the first minstrel robot. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, 129–136. IEEE.

——, eta Elena Lazkano. 2011. *Robot mugikorrak. Oinarriak.*. Udako Euskal Unibertsitatea (UEU).

Badger, Julia, Myron Diftler, Stephen Hart, eta Charles Joyce. 2012. Advancing robotic control for space exploration using robonaut 2.

Belpaeme, Tony, Paul E Baxter, Robin Read, Rachel Wood, Heriberto Cuayáhuitl, Bernd Kiefer, Stefania Racioppa, Ivana Kruijff-Korbayová, Georgios Athanasopoulos, Valentin Enescu, eta others. 2012. Multimodal child-robot interaction: Building social bonds. *Journal of Human-Robot Interaction* 1.33–53.

Brooks, Rodney Allen. 1999. *Cambrian intelligence: the early history of the new AI*. MIT press.

Ceccarelli, Graziano, Alberto Patriti, Alberto Bartoli, Alessandro Spaziani, eta Luciano Casciola. 2013. Technology in the operating room: the robot. In *Minimally Invasive Surgery of the Liver*, 43–48. Springer.

Colton, Simon, Jacob Goodwin, eta Tony Veale. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Doarn, Charles, Kevin Hufford, Thomas Low, Jacob Rosen, eta Blake Hannaford. 2007. Telesurgery and robotics. *Telemedicine and e-Health* 13.369–380.

Fasola, Juan, eta Maja J Mataric. 2012. Using socially assistive human–robot interaction to motivate physical exercise for older adults. *Proceedings of the IEEE* 100.2512–2526.

Gartzia, J, A Egaña, eta J Sarasua. 2001. *Bat bateko bertsolaritza: Gakoak eta azterbideak*. Bertsozale Elkartea.

Gervás, Pablo. 2013. Computational modelling of poetry generation. In *Artificial Intelligence and Poetry Symposium, AISB Convention*.

Hernaez, Inma, Eva Navas, Juan Luis Murugarren, eta Borja Etxebarria. 2001. Description of the ahotts system for the basque language. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

Landauer, Thomas K, Danielle S McNamara, Simon Dennis, eta Walter Kintsch. 2013. *Handbook of latent semantic analysis*. Psychology Press.

Lee, Dik L, Huei Chuang, eta Kent Seamons. 1997. Document ranking and the vector-space model. *Software, IEEE* 14.67–75.

Nagatani, Keiji, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, eta Shinji Kawatsuma. 2013. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics* 30.44–63.

Oliveira, H. 2009. Automatic generation of poetry: an overview. *Universidade de Coimbra* .

Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, eta Andrew Y Ng. 2009. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, p. 5.

Ramos, Juan. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

Rodriguez, Igor, A Astigarraga, E Jauregi, T Ruiz, eta E Lazkano. 2014. Humanizing nao robot teleoperation using ros.

Standage, Tom. 2002. *The Turk: The life and times of the famous eighteenth-century chess-playing machine*. Walker & Company.

Wong, Martin Tsan, Andy Hon Wai Chun, Qing Li, SY Chen, eta Anping Xu. 2008. Automatic haiku generation using vsm. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, number 7. World Scientific and Engineering Academy and Society.

5 Eskerrak eta oharrak

Gure eskerrik beroenak UPV/EHU-ri, ikerketa lan hau burutzeko jasotako laguntzagatik. Baita IXA[7], Aholab[8] eta Bertsozale Elkarteari[9] ere, lankidetza sustatu eta proiektua bultzatzeko egindako ahaleginagatik. Eta, batez ere, eskerrak Manex Agirrezabali, egindako ekarpen guztiengatik.

[7]http://ixa.si.ehu.eus
[8]http://aholab.ehu.es/aholab/
[9]http://www.bertsozale.eus

Actas de las XXXVI Jornadas de Automática, 2 - 4 de septiembre de 2015. Bilbao
ISBN 978-84-15914-12-9 © 2015 Comité Español de Automática de la IFAC (CEA-IFAC)

Robots en un escenario de poesía improvisada

Igor Rodriguez Rodriguez
igor.rodriguez@ehu.eus

Aitzol Astigarraga, Ekaitz Jauregi, David Salinas, Elena Lazkano, Txelo Ruiz
aitzolete@gmail.com, ekaitz.jauregi@ehu.eus, davidignacio.salinas@ehu.eus,
e.lazkano@ehu.eus, txelo.ruiz@ehu.eus

Resumen

En este artículo se presenta un sistema denominado BertsoBot*, capaz de tomar parte en un espectáculo de poesía improvisada. La arquitectura de control desarrollada para el sistema permite al robot, entre otras cosas, generar y cantar poesía improvisada (* bertsos*), interactuar con el presentador e identificar elementos del escenario.*

Palabras clave: interacción entre humanos y robots, lenguaje natural, robot humanoide, HRI, ROS.

1. Introducción

La interacción entre humanos y robots, *Human Robot Interaction* (HRI) en inglés, es una disciplina relativamente joven que recibe contribuciones de otros campos como la inteligencia artificial, procesamiento del lenguaje natural, robótica, etc., y que ha atraído mucha atención en los últimos años debido a la introducción de los robots de servicio. A pesar de que a día de hoy lo más común es ver a los robots realizando tareas industriales, cada vez podemos encontrar más robots de servicio a nuestro alrededor, ayudando a las personas, siendo éste su cometido. Son robots que están diseñados para interactuar con las personas, comunicarse con ellas y atender sus necesidades. Además, deben ser capaces de adaptarse a nuestro entorno y "convivir" con las personas con las que comparten el espacio.

La morfología de un robot es un factor importante, más aún cuando hablamos de HRI. En la actualidad los robots humanoides son los más utilizados para la interacción entre personas y robots. Debido a su apariencia humana, nos resulta más natural comunicarnos con el robot, de igual a igual. Y su adaptación al entorno es más sencilla, ya que a priori el entorno está adaptado a nuestra forma, habilidades y límites. *NAO* [14], *REEM* [18], *Pepper* [16] son ejemplos de robots humanoides ya desarrollados y preparados para la interacción con humanos.

Actualmente son muchas las aplicaciones de la robótica de servicio en las que los robots muestran habilidades sociales: se han desarrollado robots que ayudan a pacientes autistas en los hospitales [20], robots realizando el papel de guía en museos [21], asistiendo a personas mayores [7] que viven solas en sus hogares, e incluso robots que ayudan a hacer la compra [15]. A pesar de las habilidades HRI que muestran estos robots, no disfrutan todavía de un elevado nivel de autonomía. Para poder progresar en esa línea aún hay trabajo por hacer en el área de la interacción entre personas y robots, en sus diferentes ámbitos, especialmente en comunicación oral y expresión corporal. Los *Geminoids* del profesor Ishiguro del centro ATR de Japón son un referente en cuanto a desarrollo de expresión facial en androides [10].

El teatro es un ámbito ideal para la investigación y desarrollo de habilidades de HRI en robots, siendo un escenario adecuado, por ejemplo, para trabajar la expresión de emociones o la comunicación entre personas y robots. Ambos conceptos implican interactividad. Algunas personas puede que no consideren el teatro como una actividad interactiva, ya que existe un guión predefinido y todo está ensayado de antemano. Sin embargo, otros creen que sí existe una comunicación entre los participantes, puesto que sus acciones están basadas en las acciones de la otra persona, hablan entre ellos, improvisan, e incluso a veces interactúan con el público. No hay que olvidar que el término robot tiene su origen en una obra de teatro titulada RUR (*Rossum's Universal Robots*) [3]. En este área se han desarrollado trabajos como los de Chyi-Yeu Lin et al. [12] en el que cuatro robots, dos de ellos humanoides, escenifican diferentes actos en los que cuentan historias, cantan, bailan, tocan la batería, e incluso se besan. Fernandez y Bonarini [6] presentan una arquitectura que permite a los robots ejercer de actores en representaciones teatrales.

El versolarismo (*bertsolaritza*), una de las manifestaciones de la cultura vasca, es el arte de cantar en verso de manera improvisada para pronunciar un discurso, rimando y con una métrica establecida, disponiendo para la improvisación de un tiempo máximo de un minuto, aproximadamente. A diferencia de la poesía en general, en el versolarismo un *bertso* se define como una estrofa compuesta por un número determinado de versos. A su vez, se entiende por punto (*puntua* en euskera) la agrupación de dos versos, siendo el segundo el que va a rimar con los siguientes puntos.

Esta técnica o arte de la cultura vasca comparte algunas características con el teatro: público, cierto uso del lenguaje corporal, interacción entre los diferentes participantes, etc., por lo que también es un ámbito ideal para investigar en HRI.

El trabajo que se presenta en este artículo guarda cierta relación con el trabajo realizado por Fernandez y Bonarini [6]. Lo que se propone es una arquitectura que permite a los robots actuar como versolaris. Es decir, se ha desarrollado un sistema robótico, llamado *BertsoBot*, capaz de generar *bertsos* improvisados en euskera en tiempo real y cantárselos al público escenificando la expresividad que muestra un versolari cuando se sube a un escenario. El *BertsoBot* es capaz de tomar parte en un espectáculo de poesía improvisada (*bertso saio*), en el que debe entender y contestar a los ejercicios que el presentador (*gai-jartzaile*) le propone, además de identificar algunos de los elementos importantes que forman parte del escenario, como puede ser el micrófono, público, presentador, asiento o compañero.

La idea o el objetivo principal del trabajo aquí propuesto va más allá del versolarismo. Mediante este trabajo se pretende avanzar en la interacción entre personas y robots, conseguir que el robot exprese mensajes de la forma más natural posible y que sea capaz de dar una respuesta teniendo en consideración lo que ocurre en el entorno.

Para llevar a cabo la "representación"de un *bertso saio*, se han utilizado dos robots humanoides *NAO* que harán el papel de versolari, y un humano que hará el papel de presentador.

El artículo está dividido en dos bloques principales: el primer bloque describe brevemente cómo es la generación del *bertso*, el segundo en cambio describe cómo se realiza la representación o escenificación del *bertso*. En lo que a la implementación de la arquitectura se refiere, los módulos que componen el sistema *BertsoBot* han sido desarrollados utilizando ROS (Robot Operating System) [17], *framework* basado en una estructura modular y utilizado para programar robots que ofrece herramientas para la comunicación y gestión de módulos de una forma sencilla.

2. Generación del *bertso*

Sobre la creación de texto/poesía mediante computador, se pueden mencionar trabajos como el de Wong et al. [22], que presentan un sistema que genera Haikus (un tipo de posía japonesa) a partir de textos extraídos de blogs. En el trabajo de Colton et al. [4], en cambio, los autores proponen creación de poesía por símil tomando como base los corpus de texto. La generación automática de poesía se ha tratado desde ángulos diferentes, pero a nuestro conocer nadie lo ha implantado en un robot.

Son tres las herramientas principales de las que un versolari dispone para la generación del *bertso*:

1. Técnicas de improvisación aprendidas, indispensables para generar *bertsos* de métricas correctas.
2. Memoria, donde almacenan y clasifican *bertsos* anteriormente oídos, información visual y léxico.
3. Estímulos sensoriales que se reciben en los instantes previos a la generación del *bertso*.

BertsoBot dispone de las dos primeras herramientas. Es decir, como técnica de improvisación dispone de un conjunto de reglas que, dada una métrica, le permite generar *bertsos* que se ajusten a ella. Y, por supuesto, tiene una gran memoria, un corpus almacenado, compuesto por textos extraídos del periódico Egunkaria, el único disponible en euskera, ordenado por frases. Esta ordenación se debe a que las frases son las estructuras mínimas que garantizan la coherencia. De este modo, *BertsoBot* tiene un corpus de más de un millón de frases almacenado en su memoria.

Aunque los versolaris son capaces de afrontar distintos tipos de ejercicios, el *BertsoBot* por ahora solo es capaz de realizar dos de esos ejercicios: crear el *bertso* dadas las 4 palabras que riman (*oina* en euskera) o contestar a un punto dado, es decir, formar un *bertso* completo a partir del punto inicial. En el trabajo descrito en este artículo nos limitaremos al ejercicio de crear el *bertso* dadas las 4 rimas. Así, el presentador le dará las rimas al robot de una en una, y el robot tiene que componer el *bertso* y cantarlo en "zortziko txikia"(estructura de un *bertso* 7-6, con puntos de 13 sílabas).

Resumiendo, *BertsoBot* genera el *bertso* a partir de cuatro rimas, de esta manera:

1. Se obtienen del corpus las frases que finalizan con esas rimas.
2. Se seleccionan aquellas frases que cumplen la métrica (7 + 6 sílabas)
3. Aleatoriamente se selecciona una frase de cada uno de los conjuntos obtenidos para cada

Actas de las XXXVI Jornadas de Automática, 2 - 4 de septiembre de 2015. Bilbao
ISBN 978-84-15914-12-9 © 2015 Comité Español de Automática de la IFAC (CEA-IFAC)

rima. Estas frases van a constituir los cuatro puntos que formarán el *bertso* completo.

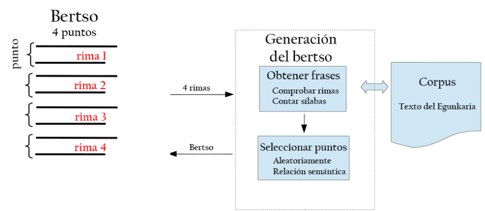
Para esta tarea se han desarrollado dos herramientas: un contador de sílabas y un buscador de rimas. Si no son rimas muy especiales, el sistema devuelve del orden de 10 frases o puntos por cada conjunto obtenido. Con esta estrategia obtenemos 10x10x10x10 = 10000 combinaciones de *bertsos* diferentes y técnicamente correctos, pero que en su inmensa mayoría carecen de sentido, ya que no existe ningún tipo de cohesión entre los puntos.

El siguiente paso es obtener de las 10000 combinaciones de *bertsos*, el *bertso* que mayor cohesión entre puntos tenga. Para tratar de medir esa cohesión se han tratado dos aproximaciones:

- *Vector Space Model* (VSM) [13]: describe los puntos como vectores, y calcula la distancia por coseno entre los puntos. Es sencillo de interpretar: cuantas mas palabras coincidentes existan entre los vectores, mayor es la cercanía semántica. Para determinar el peso de cada palabra o término en la frase se ha utilizado el esquema de ponderación denominado tf-idf (*Term Frequency-Inverse Document Frequency*). El problema principal de este método es que la similitud se basa en la coincidencia de términos. Si tuviéramos dos textos sobre el mismo tema pero términos diferentes, este método indicaría que los textos no son similares. Es decir, no trata la polisemia y la sinonimia.

- *Latent Semantic Analysis* (LSA) [5] [9]: se trata de una aproximación del método VSM, que evita los problemas anteriormente mencionados. Básicamente, se trata de una herramienta capaz de expresar la semántica de textos escritos y devuelve la medida de la relación semántica entre los términos. Para eso se utiliza el álgebra matricial (una distribución matricial denominada SVD), y como resultado, además de la coincidencia de los términos, también devuelve la medida de la relación semántica entre las frases.

Ambos métodos y los resultados obtenidos en los experimentos realizados se describen mas extensamente en los trabajos de Astigarraga et al [2, 1]. La figura 1 nos ayuda a comprender mejor lo explicado en esta sección. Por un lado se muestra el proceso de la generación automática de un *bertso* (figura 1(a)), y por el otro se muestran dos ejemplos generados por el sistema para el ejercicio de las 4 rimas (figura 1(b)); el *bertso* de la izquierda corresponde a un ejemplo generado con el método aleatorio, y el de la derecha muestra otro generado con el método LSA.



(a) Proceso de generación del *bertso*

Horrek aberastasun	*Inbiriak bihurtzen*
kulturala dakar	*gaitu nahiko zakar*
anaiak alkarrekin	*galdu gera ez badeu*
gabiltz oso zakar	*guk hartutzen alkar*
eta ospitalean	*eta gure etsaiak*
sortu ziren oso azkar	*bidaltzea azkar*
nahiz ta bertsolariak	*ondoren onik horrek*
hartu dugun alkar	*guretzat ez dakar*

(b) Dos versos distintos generados por el sistema dadas las 4 rimas

Figura 1: Generación del *bertso*

3. Escenificación del *bertso*

El versolarismo, al igual que el teatro, tiene como fin transmitir un mensaje (en el caso del versolarismo, mediante *bertsos*), a la vez que se entretiene al público. En el versolarismo no basta con crear los *bertsos*, el versolari además tiene que situarse delante del micrófono y cantar los *bertsos* al público, por lo que la interacción entre el versolari y el entorno –presentador, micrófono, público y el resto de versolaris– es tan importante como la propia composición del *bertso*.

El *BertsoBot* debe seguir la dinámica general que siguen los versolaris en un *bertso saio*. Primero, espera sentado el turno para cantar. Cuando llega su turno, se acerca al micrófono y escucha el ejercicio que el presentador le propone. Después, compone el *bertso* y lo canta. Y, por último, observa y recibe la reacción del público y vuelve a sentarse en su asiento. Aunque a simple vista no parezca complicado, son varias las tareas que el sistema de control debe ejecutar para que el comportamiento global del robot sea coherente y se asemeje al de un versolari. En ocasiones se ejecutarán de forma paralela; en otras, de forma secuencial.

La arquitectura de control desarrollada para el sistema *BertsoBot* (véase la figura 2) se define como una red de comportamientos o módulos que están en ejecución constante, que se activan cuando es necesario y ejecutan la tarea o tareas a realizar

ISBN 978-84-15914-12-9

(moverse, componer el *bertso*, hablar, cantar, etc.). Aunque esos módulos son independientes, se coordinan mediante una máquina de estados que sigue la dinámica general de un *bertso saio*, con lo que se consigue la coherencia del sistema. Los módulos que componen el sistema *BertsoBot* son los siguientes:

- Módulo de análisis de voz
- Módulo de síntesis de voz
- Módulo de control de estados
- Módulo de identificación
- Módulo de expresión corporal
- Módulo de generación y canto del *bertso*

3.1. Módulos de análisis y síntesis de voz

Del mismo modo que ocurre en la comunicación entre los seres humanos, uno de los objetivos del HRI es lograr que la comunicación se desarrolle de una forma natural. Es decir, por voz. Para eso, es necesario que los robots sean capaces de escuchar y hablar, además de entender lo que escuchan, y emitir una respuesta coherente. Este aspecto tiene aún un gran margen de mejora.

Para que los robots puedan entender lo que oyen, es necesario, por un lado, un sistema de percepción auditiva y, por otro lado, un sistema de reconocimiento automático de voz. En el trabajo aquí propuesto, el módulo de análisis de voz desarrollado combina estos dos sistemas. Su funcionamiento es sencillo: primero, se captura el audio; después, el audio se convierte en texto, y finalmente se analiza el texto. Para la captura de audio se ha utilizado la herramienta SOX, software para el procesamiento de audio, que, entre otras opciones, permite reproducir y grabar audio. La conversión de audio a texto, en cambio, se realiza utilizando la herramienta de reconocimiento automática de voz de Google, *Google Speech*. Una vez se obtiene el texto que corresponde al audio grabado, se realiza un análisis del mismo. El análisis consiste en verificar si alguna de las palabras del texto obtenido coincide con la base de datos de conocimiento del robot, que no es más que una lista de palabras predefinidas (con varios sinónimos por cada una). Si existe coincidencia, el robot realizará la acción que corresponda a esa "orden". En cambio, si no existe, el robot se lo transmite al presentador indicándole que no entiende lo que se le ha dicho.

Asimismo, para que un robot pueda comunicarse con el presentador u otros versolaris, necesita poder hablar (en euskera). El *BertsoBot* tiene esa habilidad gracias al módulo de síntesis de voz desarrollado por Aholab, *AhoTTS* [8]. Este módulo se encarga de convertir en voz el texto que se quiere transmitir, y posteriormente la voz es reproducida por el robot.

3.2. Módulo de control de estados

Para conseguir que *BertsoBot* siga la dinámica general de un *bertso saio* se ha desarrollado un módulo encargado de controlar en qué estado o momento del *bertso saio* se encuentra el robot. El modelo que describe la dinámica general se resume en los siguientes comportamientos:

1. Esperar su turno para cantar.
2. Acercarse al micrófono y escuchar el ejercicio que el presentador le propone.
3. Generar el *bertso* y cantarlo.
4. Observar la reacción del público para alimentarse de ello en futuros *bertsos*.
5. Volver a su asiento.

Estos comportamientos se controlan mediante la máquina de estados implementada en el trabajo realizado. Se han definido estados diferentes en los que el robot muestra un comportamiento u otro. El cambio de un estado a otro se realiza mediante diferentes señales que la máquina de estados recibe. Estas señales pueden ser órdenes dadas por el presentador, o bien señales relacionadas con la consecución del objetivo del comportamiento correspondiente. No se contemplan los posibles errores que puedan ocurrir en cada estado, lo que significa que el robot no cantará si anteriormente no ha localizado el micrófono o no consigue alcanzarlo, si no se le ha dado el ejercicio, etc. La única excepción de detección de errores es en el caso de las rimas, ya que si no las ha entendido, el robot solicita que le sean repetidas.

En el estado actual de desarrollo, no existe aún retroalimentación sobre el estado anímico del público ante la actuación. La figura 3 describe más detalladamente los estados que se han definido para un *bertso saio*. El sistema de control que se propone tiene en cuenta el número de robots que toman parte en un *bertso saio*. Como dos robots no pueden cantar a la vez, es el presentador quien decide el turno del robot que va a cantar, eligiendo aleatoriamente uno de ellos y llamándole por su nombre.

3.3. Módulo de identificación

Cuando los versolaris están en un escenario, dependiendo de la acción que vayan a realizar, cen-

ISBN 978-84-15914-12-9

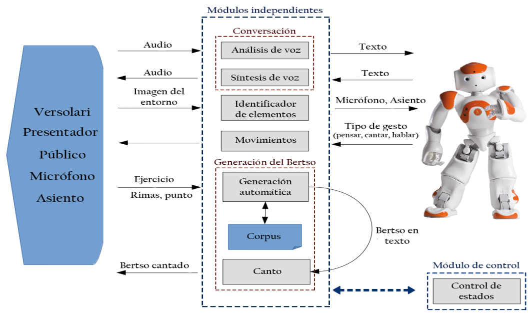


Figura 2: Arquitectura del sistema

tran su atención en uno de los diferentes elementos que forman parte del mismo: presentador, otros versolaris, asiento, micrófono, público, etc. En el trabajo desarrollado hasta el momento el *BertsoBot* es capaz de identificar el micrófono, su asiento, y a su compañero robot versolari.

La ubicación del micrófono y de su asiento son dos puntos de referencia para los versolaris (véase figura 4). Uno es el punto de referencia de inicio, la herramienta con la que va a realizar su trabajo, y el otro, el de fin, lugar donde termina su trabajo y descansa esperando su siguiente turno. Por supuesto, estos elementos están adaptados a la morfología del robot. La identificación de estos elementos se realiza en base a una etiqueta de color predeterminada que permite distinguirlos; así, las sillas de cada robot se distinguen por un color y, de la misma manera, el micro tiene una etiqueta de color azul en la base para poder localizarlo. Dado que la posición de estos elementos con respecto del robot varía según la localización del robot, el comportamiento de aproximación a ellos se implementa utilizando un tracking de color basado en un filtro de Kalman que produce un comportamiento más robusto frente al bamboleo del robot al andar y a las variaciones de luz.

Además del micrófono y el asiento, *BertsoBot* también es capaz de identificar a su compañero robot versolari. El modo de identificación es simi-

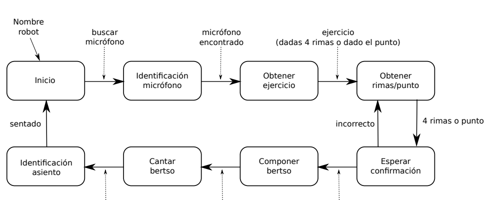
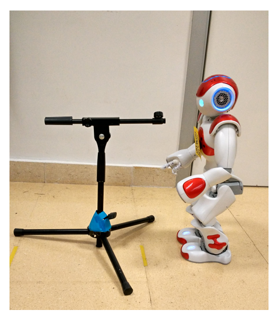
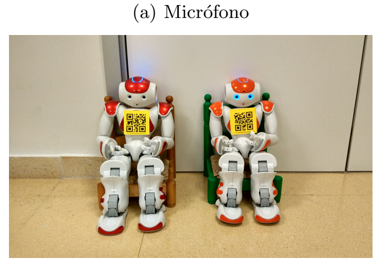


Figura 3: Descripción de los estados posibles



(a) Micrófono



(b) Robots en sus respectivas sillas

Figura 4: Elementos del escenario

lar al aplicado con los elementos anteriores. Los robots, morfológicamente iguales, también se distinguen por una etiqueta de color, y éstas, a su vez, contienen un codigo QR (figura 4(b)), que al traducirlo proporciona el nombre del robot.

3.4. Módulo de expresión corporal

La tarea principal de este módulo es reproducir los gestos y movimientos que realizan los versolaris durante un *bertso saio*. Para ello, se han analizado los gestos más habituales que suelen hacer los versolaris cuando están sobre el tablado (mirar al público, colocar las manos atrás y mover la cabeza arriba y abajo mientras están pensando el *bertso*, etc.), en definitiva, su expresión corporal en el escenario.

Se han implementado cuatro tipos de gestos, según el estado en el que se encuentren. Por una parte, los gestos considerados estándar, como son levantarse de la silla, sentarse, avanzar hacia el micro, regresar a la silla, etc. Por otra parte, los gestos particulares relacionados con la representación del *bertso*:

- Gestos al pensar: aquellos que puede realizar mientras está pensando el *bertso*, como por ejemplo poner las manos atrás, balancearse de un lado a otro en actitud pensativa, rascarse la cabeza, etc. (véase la figura 5)
- Gestos al cantar: aquellos que realiza a la hora de cantar, como por ejemplo, erguirse, carraspear, etc.
- Gestos al hablar: aquellos movimientos que puede realizar al hablar, como por ejemplo gesticular con las manos.

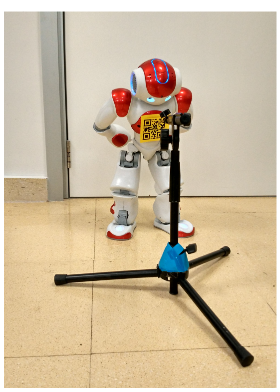


Figura 5: *NAO* pensando un *bertso*

Se han definido librerías de movimientos asociadas a las fases. Estos movimientos, en su mayoría, han sido generados mediante *Choregraphe*, aunque también se han aprovechado y modificado algunos ya disponibles en la libreria de movimientos del propio robot. Aleatoriamente, se eligen los gestos a realizar en cada una de las fases. La duración de la fase de hablar varía según la duración del audio. De ahí que, para aumentar la naturalidad y la expresividad, en definitiva la empatía del robot, se elija un número aleatorio de movimientos a ejecutar, en un orden también aleatorio, que sea inferior a la duración del habla.

3.5. Módulo de generación y canto del *bertso*

Este módulo recibe el tipo de ejercicio a realizar: dadas las cuatro rimas, generar el *bertso*, o dado el punto, finalizarlo. En el caso de las rimas, el presentador le da las cuatro rimas una a una, con pausas entre ellas. En el caso del punto, el presentador recita la frase que corresponde al punto. En ambos casos, el robot repite lo que ha entendido y se queda a la espera de una confirmación. Si la confirmación es negativa, el presentador debe proceder a repetir las cuatro rimas o el punto. Este proceso se repite hasta que el robot haya entendido correctamente la información proporcionada por el presentador y la confirmación sea, por tanto, positiva. En dicho caso comienza un proceso de comprobación en el que, si el ejercicio es el de las rimas, se comprueba que el número de rimas comprendidas es 4, o si el ejercicio es el del punto, se comprueba que el mismo es de 13 sílabas. Una vez confirmado que se cumplen los requisitos para el ejercicio el robot comienza con el proceso de generación del *bertso* explicado en el apartado 2. Además de hablar, el robot en este caso ha de cantar. Para convertir el *bertso* en canto, se escoge aleatoriamente una melodía de la base de datos disponible, acorde con la métrica del verso y se utiliza una adaptación del AhoTTS que, modificando entre otros detalles la duración y la entonación de las sílabas, produce el audio que contiene el *bertso* listo para ser cantado por el robot.

3.6. Conclusiones y trabajo futuro

En este artículo se ha presentado un sistema denominado *BertsoBot*, capaz de tomar parte en un espectáculo de poesía improvisada. La arquitectura de control desarrollada para el sistema permite al robot, entre otras cosas, generar y cantar *bertsos*, interactuar con el presentador e identificar elementos del escenario.

En el canal de Youtube del grupo RSAIT [19] puede verse un video del funcionamiento del sistema en un experimento en el que un tercer robot (*Galtxagorri*, modelo Pioneer 3DX) hace las labores de presentador [1]. Por otra parte, distintas partes del sistema han sido probadas durante las diferentes fases del desarrollo en actuaciones reales. Por ejemplo, la capacidad de improvisación y comunicación oral en una actuación en directo titulada "*BertsoBot*, ciencia o ficción"[2]. En la iniciativa llamada *ZientziaClub*[3] o Club de las ciencias se mostraron algunos avances en cuanto a la interacción entre una persona y un robot.

En lo que a las contribuciones del trabajo de investigación se refiere, por un lado se propone utilizar los métodos VSM y LSA en el ámbito de la generación automática de texto. Por otro lado, se han desarrollado varios módulos que aportan novedades en el área de HRI: un sistema de diálogo en euskera y una arquitectura de control basada en ROS que coordina todos los comportamientos del robot para que éste pueda improvisar *bertsos* en un *bertso saio*.

Queda pendiente la mejora de la máquina de estados, en la cual se pretende añadir nuevos estados que contemplen los diferentes errores que puedan ocurrir al realizar cada tarea. Por ejemplo, establecer tiempos límites para cada tarea y pedir ayuda al presentador en caso de no poder alcanzar el objetivo.

En [11] se subraya la importancia de la audiencia en la robótica social, como medio para proveer *feedback* al comportamiento del robot. La retroalimentación al robot con la respuesta emocional del público ofrece un nuevo dominio de experimentación. Como trabajo futuro inmediato, pretendemos que la reacción del público afecte al proceso de generación de los *bertsos* en tiempo real y, además, el lenguaje gestual del robot refleje el estado de ánimo percibido.

Agradecimientos

El presente trabajo ha sido parcialmente financiado por el Gobierno Vasco en la ayuda a grupos consolidados IT313-10 y por la Universidad del País Vasco UPV/EHU

Referencias

[1] A Astigarraga, M. Agirrezabal, E. Lazkano, E. Jauregi, and B. Sierra. Bertsobot: the first minstrel robot. In *6th International Conference on Human System Interaction*, pages 129–136, Sopot, Poland, June 2013. IEEE.

[2] A. Astigarraga, E. Jauregi, E. Lazkano, and M. Agirrezabal. Textual coherence in a versemaker robot. In *Human-Computer Systems Interaction: Backgrounds and Applications 3*, pages 275–287. Springer International Publishing, Cham, Switzerland, 2014.

[3] K. Čapek. *R.U.R (Rossum's Universal Robots). 1920.* Trans. Paul Selver and Nigel Playfair. Dover Pub. Inc., Mineola, NY, 2001.

[4] S. Colton, J. Goodwin, and T. Veale. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, pages 95–102, Dublin, Ireland, 2012.

[5] S. Deerwester, Susan T. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[6] J. Fernandez and A. Bonarini. Theatrebot: A software architecture for a theatrical robot. In *Towards Autonomous Robotic Systems*, pages 446–457. Springer, Birminghan, UK, 2014.

[7] D. Fischinger, P. Einramhof, K. Papoutsakis, W. Wohlkinger, P. Mayer, P. Panek, S. Hofmann, T. Koertner, A. Weiss, A. Argyros, et al. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, 2014.

[8] I. Hernaez, E. Navas, J. Murugarren, and B. Etxebarria. Description of the ahotts system for the basque language. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, Perthshire, Scotland, 2001.

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001.

[10] Hiroshi Ishiguro. ATR. `http://www.geminoid.jp/en/index.html`, [Fecha de consulta: 13 Julio 2015].

[11] H. Knight. Eight lessons learned about nonverbal interactions through robot theater. *Social Robotics*, pages 42–51, 2011.

[12] Chyi-Yeu Lin, Chang-Kuo Tseng, Wei-Chung Teng, Wei-Chen Lee, Chung-Hsien Kuo, Hung-Yan Gu, Kuo-Liang Chung, and Chin-Shyurng Fahn. The realization of robot theater: Humanoid robots and theatric performance. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6, Munich, Germany, 2009. IEEE.

[1] https://www.youtube.com/watch?v=3zh9Uy6GFWY
[2] http://www.badubada.com/badubadatzen/es/robot-bertsolaria-zientzia-ala-fikzioa/
[3] https://www.youtube.com/watch?v=NEiDwJBER9M

[13] C. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press, Cambridge, 2008.

[14] NAO. Aldebaran Robotics. https://www.aldebaran.com/en/humanoid-robot/nao-robot, [Fecha de consulta: 13 Julio 2015].

[15] OSHbot. Fellow Robots. http://fellowrobots.com/robots, [Fecha de consulta: 13 Julio 2015].

[16] Pepper. Aldebaran Robotics. https://www.aldebaran.com/en/a-robots/who-is-pepper, [Fecha de consulta: 13 Julio 2015].

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, Kobe, Japan, 2009.

[18] REEM. Pal Robotics. http://pal-robotics.com/es/products/reem/, [Fecha de consulta: 13 Julio 2015].

[19] RSAIT research group. RSAIT-youtube channel. https://www.youtube.com/channel/UCT1s6oS21d8fxFeugxCrjnQ, [Fecha de consulta: 13 Julio 2015].

[20] S. Shamsuddin, H. Yussof, L. Ismail, F. Hanapiah, S. Mohamed, H. Piah, and N. Zahari. Initial response of autistic children in human-robot interaction therapy with humanoid robot nao. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pages 188–193, Malacca, Malaysia, 2012. IEEE.

[21] L. Susperregi, I. Fernandez, A. Fernandez, S. Fernandez, I. Maurtua, and I. Lopez de Vallejo. Interacting with a robot: a guide robot understanding natural language instructions. In *Ubiquitous Computing and Ambient Intelligence*, pages 185–192. Springer-Verlag, Berlin, 2012.

[22] M Tsan Wong, A Hon Wai Chun, Qing Li, SY Chen, and Anping Xu. Automatic haiku generation using vsm. In *WSEAS International Conference. Mathematics and Computers in Science and Engineering*, number 7, Hangzhou, China, 2008. World Scientific and Engineering Academy and Society.

Jornadas Nacionales de Robótica
Spanish Robotics Conference
8-9 Junio 2017 http://jnr2017.ai2.upv.es

BertsoBot: Towards a Framework for Socially Interacting Robots

A. Astigarraga[a,1,2*], I. Rodriguez[a], T. Ruiz[b], E. Lazkano[a]

[a] *Computer Sciences and Artificial Intelligence, University of Basque Country (UPV/EHU), Manuel Lardizabal 1, 20018 Donostia.*
[b] *Computer Architecture and Technology, University of Basque Country (UPV/EHU), Manuel Lardizabal 1, 20018 Donostia.*

Abstract

The objective of this article is to compile the work carried out by the RSAIT[3] research group on the Bertsobot project. The Bertsobot project aims to develop an autonomous robot capable of composing and playing traditional Basque impromptu verses – bertsoak. The developed system is able to construct novel verses according to given constraints on rhyme and metric that also show semantical coherence, and to perform it in public. The Bertsobot project, at the intersection of Autonomous Robotics, Natural Language Generation and Human Robot Interaction, works to model the human abilities that collaborate in the process of creating and performing impromptu verses in front of an audience. This paper brings together the steps taken in the design and implementation of robot's individual behaviors and the overall control architecture.

Palabras Clave:
Robotics, Natural Language Processing, Machine Learning.
Datos del Proyecto:
Denominación del proyecto: Análisis de Personas con Biometría Blanda para Servicios Inteligentes Multilingües
Referencia: TIN-2015-64395-R
Investigador/es responsable/es: Basilio Sierra y Elena Lazkano
Tipo de proyecto (internacional, nacional, autonómico, transferencia): Nacional
Entidad/es financiadora/s: Ministerio de Economía y Competitividad
Fecha de inicio/fin: 01/012/2016 – 31/12/2018

1. Introduction

Until recently, it was impossible to consider humans and robots living together. But now, robots start to become companions or co-workers of humans, opening an important research domain to build robots that are able to intuitively interact with humans. A considerable number of robotic systems have been developed in the last decade showing Human Robot Interaction (HRI) capabilities [Fong et al., 2003][Goodrich and Schultz, 2007].

However, social robots are beyond HRI. According to Breazeal [Breazeal, 2004], sociable robots are socially intelligent robots in a human like way, and they need to show the "human social" characteristics like the expression of emotions, the ability to conduct high-level dialogue, to learn, to develop personality, and to develop social competencies. In consonance with FeilSeifer and Mataric [Feil-Seifer and Mataríc, 2011] social robots can be categorized as assistive robots (AR), socially interactive robots (SIR) and socially assistive robots (SAR). Regardless of the applications, in the last years, research in the field of social robotics has grown. Several robots have been designed in this area, to support development of self-efficacy and emotional well-being in diabetic children [Cañamero and Lewis, 2016], as interactive teachers in a collaborative learning class with infants [Kanda et al., 2012] and as shopping mall guide, designed for customer navigation, information providing and enjoyment [Chen et al., 2015], to mention some.

Entertainment is an area in which social robots can have high impact. Public performances using robots have shown to be a great setting for disclosing the state of the art of social robots to the general public. Theatre, a live entertainment activity, offers an invaluable field to research and develop social skills in robots. Although everything is rehearsed beforehand, theatre offers an invaluable sphere to research and develop social behaviours in robots, to work and extent the expression of emotions and the natural communication among humans and robots [Lin et al.2009] [Fernandez and Bonarini, 2014]. A review of robot performances can be found in [Murphy et al., 2011]. Little by little robots are bursting into

1 Aitzol Astigarraga.
2* Autor en correspondencia.
Correos electrónicos: aitzol.astigarraga@ehu.eus (A. Astigarraga),
igor.rodriguez@ehu.eus (I. Rodriguez),
txelo.ruiz @ehu.eus (T. Ruiz),
e.lazkano@ehu.eus (E. Lazkano)
URL: http://www.sc.ehu.es/ccwrobot/seccion/members-2/subseccion/aitzol-astigarraga-2 (A. Astigarraga)
3http://www.sc.ehu.es/ccwrobot/seccion/home/lang/en

theatres motivated by researchers as a means, but also by artists [Li].

The *BertsoBot* project described in this paper resumes the work done by RSAIT in the last years. This project provides a huge opportunity to develop social robot capabilities in the context of *bertsolaritza*, a traditional Basque improvised singed poetry manifestation. It aims to develop minstrel robots that, beyond generating verses automatically, are able to sense the environment and interact with it, and show proper body language and robot communication skills, like real troubadours do. Aldebaran's NAO humanoid robotic platforms are being used that, although faceless, allow for body language development and have multisensory capabilities.

2. Bertsolaritza and Automatic Verse Generation

Basque, *Euskara*, is the language of the inhabitants of the Basque Country. And *bertsolaritza*, Basque improvised contest poetry, is one of the manifestations of traditional Basque culture that is still very much alive.

Events and competitions are very common (see Figure 1), which usually consist of a rather formal flow of poetry recitations, *bertso*-s. In such performances, several verse-makers, compete with each other singing improvised verses about topics or prompts which are given to them by an emcee (theme-prompter). They compose the verses on the fly, normally in less than one minute, and sing a poem along the pattern of a prescribed verse-form that also involves a rhyme scheme. Melodies are chosen from among hundreds of tunes. Xabier Amuriza, a famous verse-maker defined *bertsolaritza* in a verse as:

Neurriz eta errimaz	**Through meter and rhyme**
kantatzea hitza	**at singing a word**
horra hor zer kirol	**could bertsolaritza be**
mota den bertsolaritza.	**seen as sport.**

Figure 1: 2015 National bertsolari's championship.

Different poetry disciplines similar to *bertsolaritza* can be found around the world, such as Catalan glossators, Argentine payadors or Italian bards to mention some. However, the closest example is the American poetry slam (Somers-Willett, 2009) in which poets read or recite poems and are sometimes judged by selected members of the audience and sometimes, like in *bertso* contests, by a panel of judges.

The art of composing improvised verses requires a number of prerequisites that must be taken into account. We can say that any person with the capabilities to construct and sing a *bertso* with the chosen meter and rhyme has the minimum skills to be a *bertsolari*. But the real value of the *bertso* goes beyond composing a verse according to those demanding technical requirements. Its real value resides on its dialectical, rhetorical and poetical value [Garzia et al. 2001]. Thus, a *bertsolari* must be able to express a variety of ideas and thoughts in an original way while dealing with the mentioned technical constraints. In this balance lies the magic of a *bertso.*

Bertso-s can be composed in a variety of settings and manners. For instance, *Zortziko Txikia* (see Figure 2) is a composition of eight lines in which odd lines have seven syllables and even ones have six. The union of each odd line with the next even line form a strophe. Each verse has 13 syllables with a caesura after the 7th syllable (7 + 6) and must rhyme with the others.

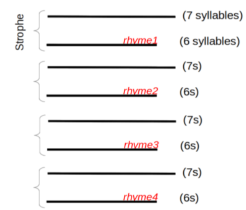


Figure 2: Structure of a verse in the Zortziko txikia meter.

2.1. Automatic Verse Generation

Computer-based poetry has been paid attention to in the research community for the last years (see [Gervás, 2013] and [Oliveira, 2009] for a review), but among the several differences that exist between poetry and *bertsolaritza*, mainly the later belongs to the oral genre, and the public performance is extremely important.

According to Laborde [Laborde, 2005], human verse makers have three main tools for improvising verses:

1. Learned techniques and rules for improvisation, mandatory for generating verses metrically correct.
2. Memory to store and classify previously listened verses, visual and lexical information.
3. The sensorial stimuli that are input in the instants prior to the generation of the verses.

The improvisation process is then the result of a set of rules that, given a metric, produce a technically sound verse with content obtained from a huge memory.

We have developed two poem generation strategies that respond to popular exercises in *bertsolaritza*:

- 4 rhymes given: four rhyming words are given and is required to compose the bertso "around" these rhyming words.
- Theme given: a bertso must be composed on the given subject.

In this work we will focus on the first strategy, and thus, the automatic verse generation process consists of the following steps:

1. Receive as input the four rhymes to compose the verse.
2. Find sentences in the corpus that rhyme with the input words and have the correct number of syllables.
3. Generate the verse with the highest textual coherence.

3. The BertsoBot Architecture

The aim of any *bertsolari* when she/he sings an improvised verse is to convey a message, according to the requirements imposed by the emcee, while entertaining the public. But the interaction with the environment – stand in front of the microphone, obtain rhymes given by the emcee, perceive audience's reaction – is as important as the composition of the verse.

The *BertsoBot* system endows the robots with some of the *bertsolari*-s' capabilities that allows this social robots to take part in a public performance. For that purpose, the system must follow the dynamic of real events, *bertso-saio*-s, as troubadours do.

1. Wait sitting for its turn.
2. When it is its turn, place itself in front of the microphone and listen to the exercise proposed by the emcee.
3. Compose and sing the verse to the public.
4. Observe and receive audience's feedback and react accordingly.
5. Go back to its sitting place.

All these tasks are accomplished and managed by a ROS[4] based control architecture, composed by different behaviours or modules that make the robot act in a consistent manner and resemble to a real *bertsolari*. Figure 3 shows the global system architecture.

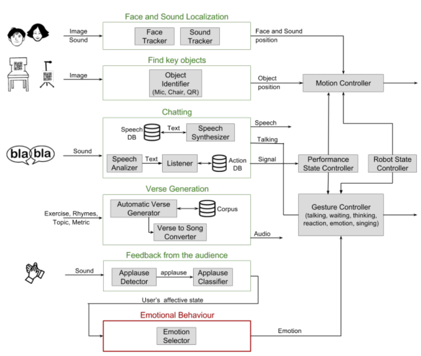


Figure 3: ROS based control architecture

Summarizing Figure 3, the "Performance State Controller" is the module that brings the coherence necessary to the system in order to follow the dynamic of a performance. The "Face and Sound Localization" as well as the "Chatting" behaviours allow the interaction with the emcee, while "Find key objects" provides the robot with necessary skills to interact with environmental key objects. These interactions, usually executed as motion actions, are managed by the "Motion Controller". The action to be executed will depend on robot current state, managed by "Robot State Controller". The verse is composed and sung by the "Verse Generation" process, and audience applauses, which affect the robot's emotional state, are captured and classified by "Feedback from the audience" behaviour. The robot body expression is managed by the "Gesture controller" which decides the gestures to be applied from the appropriate gesture set at each state according to the "Performance state", the "Robot state", and the "Emotion selector".

4. Behaviours

4.1. Chatting

In order to generate the verse, the robot needs to identify the proposed exercise and the given rhymes first. The audio is captured via SOX[5] and afterwards, the Google Speech service is used as speech recognizer to convert the audio to text. Once the text is received, it is analysed to verify whether the words are available in a local dictionary (list of words with synonyms). If, as a consequence of the analysis no word is recognized, then the robot tells the emcee that it has not understood the sentence and asks to repeat the exercise.

To be able to communicate with the emcee, the robot makes use of AhoTTS tool, a speech synthesizer for Basque Language developed by AhoLab [Hernaez et al., 2001].

4.2. Verse Generation and Singing

In the basic scenario, the four rhymes to compose a *bertso* are received as input, and the verse generator module then should give as output a novel and technically correct verse, and with coherent content. This process can be mainly split up into two steps; first, find the sentences in the corpus that rhyme with the input words and have the correct number of syllables, and then, generate the verse with coherence. See [Astigarraga et al., 2013] [Astigarraga et al., 2014] for a more detailed explanation.

For the first step, two tools have been developed: the rhymes finder and the syllables counter. The rhymes finder outputs the sentences obtained from a corpus that match with the rhymes given by the emcee. The corpus is composed with documents mined from the Basque newspaper Egunkaria[6] (%85) alongside verses extracted from the work of well-known *bertsolari*-s (%15). While the syllables counter filters the sentences according to the given metric.

The next step is to obtain the verse with highest coherence or meaningfulness. Latent Semantic Analysis (LSA) approach has been used to compare the relationship between pairs of sentences to choose the most semantically related ones for the final verse. To do that, the sentence relation system computes, for each pair of sentences, a score that evaluates how those sentence are semantically related.

As final result, the output poem must be translated to a song in an audio file that will afterwards be reproduced by the robot. To get such audio, first the utilized metric is analysed and, then, a melody is randomly chosen from an available database and, using a modified version of the AhoTTS that changes the duration and intonation of the syllables, among other features, produces the audio file with the singed verse.

4www.ros.org

5Sound eXchange, a cross-platform command line utility to process audio files

6https://en.wikipedia.org/wiki/Egunkaria

4.3. Interaction with the environment

4.3.1. Face and Sound localization

A natural reaction when we want to interact with someone is to direct our gaze towards the interested agent. The gaze feeds the communication, and conveys interest or attention to the interlocutor. It requires positioning the robot to make the most out of its sensors and to let the human talker know what the robot is actually paying attention to. Spontaneity during verbal communication involves two main behaviours, face and sound localization.

Face localization is done applying OpenCV's Haar feature-based cascade classifiers [Viola and Jones, 2001] to the images taken by the upper camera on the NAO's head. Once the face is detected within an image, the center of the face in the image is obtained, and the head joint angles to track the face, with respect to the center of the image are calculated.

Sound localization allows a robot to identify the direction of sound, and it is done using Aldebaran's "ALSoundDetection" algorithm based on TDOA (Time Difference of Arrival) approach [Bensky, 2016]. The sound wave emitted by a source is received at slightly different times on each of the NAOs four microphones, from the closest to the farthest. These differences are related to the current location of the emitting source. By using this relationship, the robot is able to retrieve the direction of the emitting source (azimuth and elevation angles) from the TDOAs measured on the different microphone pairs.

4.3.2. Find key objects

The robot pays attention to different elements at different states. The robot can be requested to reach the microphone to start its singing turn or it may need to go to rest to its chair. For the time being, those elements, as well as being adapted to the robot's morphology, they have labels to make it easier the identification and recognition processes. They all have colour tags that make them distinguishable; chairs have been painted with different colours and, similarly, the microphone has a blue tag on its base. Every key object has a QR code to make it recognizable. A colour tracking procedure enhanced with a Kalman Filter is used to produce a more robust behaviour against illumination conditions and balancing produced during walking. No location information in form of odometry or frame of reference is used because the location of those elements with respect to the robots varies depending on the scenario.

4.3.3. Feedback from the audience

Audience plays an important role in any type of performances, specially in *bertsolaritza*. Despite the thrilled state of the audience, the need for concentration of human poets is very much respected by them, and thus, the crowd waits until the actor finishes to show how pleasant the verses have been, usually clapping as well as laughing when they have found it amusing.

Perceiving and showing emotions is essential to convey interaction. The Affective Loop is the interactive process inwhich the user of the system first expresses her/his emotionsthrough some physical interaction involving her body, and the system responds by generating affective expression, which in turn affects the user making her/him respond and step-by-step feel more and more involved with the system [Höök, 2009][Paiva et al., 2015].

Developing an approach to react to the audience's feedback covers multiple fields, such as applause detection, classification and selection of the robots appropriate reaction in the context of the performance.

The presented approach uses audience applause as feedback to the robot system [Kraemer et al., 2016]. Applauses are captured and translated into a response from the public by means of energy (E) and duration (d) of the applause. The addressed strategy can be split up into a straight-forward workflow (see Figure 4). In the initial step, audio processing and machine learning techniques prepare the input audio stream by first chunking it, and then classifying each chunk as being applause or not. Next, the incoming stream of classified chunks is segmented into sections of consecutive applauses, leading to a small descriptor ([E, d]) for every evaluated applause. Based on all previous applauses of the event, the most recent one can subsequently be classified. The applauses are coarsely categorized as belonging to one of the following classes: *Negative*, *Neutral*, *Positive* and *Very Positive*.

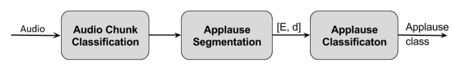


Figure 4: Approach workflow

As all events are different, it is difficult to make a comparison between them. Thus, in our approach we use unsupervised online learning techniques. We use k-means to do clustering with a variable number of classes. The first applause is always classified as *Neutral*, after that, the number of available classes is increased.

4.4. Motion and Gestures controller

4.4.1. Robot State

"Robot State Controller" provides information about the posture in which the robot is, and whether the robot is moving or not. The posture classification is made by a C4.5 decision tree [21] that classifies robot posture as being sat on floor, sat on chair, crouched (rest position) and standing up. Postures were represented using 150 variables obtained from the TF[7] tree of each robot joint (x, y, z, roll, pitch, yaw) values of the 25 DoF). To train the C4.5 classifier a total of 240 data entries were collect, 60 entries of each class. In order to determine if the robot is moving or not a comparison of all joints is done in two different times.

4.4.2. Motion

There are several behaviours that output a motion action. For instance, when the emcee's face or a sound is detected the robot must direct its gaze and body to her/him, when it is its singing turn it must walk to the microphone, or when receiving

7http://wiki.ros.org/tf

a specific order in the context of the performance. All these motion actions are managed by a robot motion controller which makes the robot react accordingly to the current situation, and which is also conditioned by the robot actual state.

4.4.3. Gestures

If our robots are meant to participate in such verse contests, beyond singing capabilities, they must show the same degree of expressiveness Basque troubadours do. Improvising a verse is a very hard mental process that requires extreme concentration and that is reflected in the body language of the imprompters. When the *bertsolari*-s are on the stage they are continuously conveying information, through facial expressions, body postures, movements or gestures, intentionally or not, about their emotional state.

The robot needs to move, needs to reproduce some gestures but it cannot be continuously gesturing like a puppet. After identifying the main different states of the global behaviour, a gesture library composed by five different gesture sets have been defined to mimic troubadours' emotional behaviour on the stage [Rodriguez et al. 2016]. At each state of the performance appropriate gesture set is selected, and in order to avoid to see the robot doing exactly the same thing once and again, the gesture to be reproduced, the interval between gestures, the execution order and the number of gestures are always randomly selected. The five gesture sets are:

- *Thinking gestures*: Those gestures that, unconsciously, humans make while standing up in front of the microphone and thinking the verse. They are movements to unstress, to relax tension like put one's hands back, swing the hip, scratch one's head, etc. There is one gesture extremely important while thinking: reach and maintain a neutral pose.
- *Talking gestures*: Humans do not stay still while talking, we naturally gesticulate moving the hands or nodding. NAO accompanies its speech moving its arms too. In this case, the number of gestures to perform varies according to the duration of the speech. The robot also nods to make visible that it has understood something. It does not mean that it knows what has been said, but it makes the interlocutor realize that the robot has successfully processed the captured audio.
- *Singing preamble gestures*: Just after the improvisation process finishes and before the *bertsolari* starts singing, he/she needs to accommodate the body and/or clear the throat, look around and probably stare off into space, above the public. Oddly, and probably due to the extreme concentration effort that must be maintained, the troubadours stands still while singing. Of course, not everyone maintains the same pose, sometimes they keep the hands on their pockets, or on their back, or just have their arms down, but that pose does not vary significantly from one *bertsolari* to the other. Thus, no gesture is reproduced while singing.
- *Waiting gestures*: Humans are not designed to be motionless while being awake, and so, it is not appropriate to have a robot sat inert or stand up paralyzed in the stage. Humans stretch or cross their legs, drink water or move the head to change the gaze while being sat. No need to say that our robots' movements are very limited in that position, and that most of the mentioned moves cannot be replicated. But they can change their arms' position and make movements with their heads. Again, the neutral pose is often required to be maintained.
- *Emotional reaction gestures*: After the *bertsolari* sings a verse the audience responds applauding to express their opinion, and this reaction is reflected in the robot as emotion gesture. Each applause feedback class has been represented with an emotion; in the next order *Sad*, *Calm*, *Joy* and *Excited* emotions correspond to *Negative*, *Neutral*, *Positive* and *Very Positive* applause classes. Two different behaviours were implemented. The diffident behaviour only makes use of the first three classes. The more exaggerated behaviour makes use of all classes.

5. Public Performances

The robots' performance capabilities have been demonstrated in different events in a 4 years period. These public performances show the evolution of the *BertsoBot* project since its start up, when no humanoid platform was available and up to now.

- **2012/04 - First public appearance**: Inauguration of the speaker's corner of our Campus. Paradoxically the most audacious one, due to the importance of the event and the preliminary state of the project.
  Tartalo and Galtxagorri – PeopleBot and Pioneer2DX platforms – were brought out and acted outdoor. No significant body language was shown, neither chatting was possible. Robots were mainly teleoperated and control software was Player/Stage. Only the automatic verse generation system was embedded in wheeled robots. Video available[8].
- **2013/05 - Robots against bachelor students**: An event hold in the Faculty of Informatics where robots competed against some *bertso*-amateur students.
  Tartalo was accompanied by NAO for the first time. Primary gestures were shown by NAO, that acted as the emcee semi-autonomously. NAO was controlled using Choregraphe. Video available[9].
- **2014/03 - Women's day at the Faculty**: The UPV/EHU annually celebrates the women's international day in a different center and in 2014 it was held at our Faculty. The program included a *bertso* event where two big professionals and two robots (NAO and Tartalo) took part.
  NAO showed improved chatting abilities, but still "unROSified". Primary gestures in NAO were shown, which guided the event but semi-autonomously[10].

8https://www.youtube.com/watch?v=OpQBVmkzRWg
9http://www.eitb.eus/eu/kultura/bertsolaritza/osoa/1350970/robotbertsolariak-ixa-taldea-eta-ehuko-robotika-saila/
10http://ehutb.ehu.es/es/video/index/uuid/531ec65f964be.html

- **2014/11 - ScienceClub**: Club of Sciences events aim to disclose science and technologies to the society. A dialogue with NAO entitled "Chatting with NAO" of approximately 10 minutes was presented. NAO acted alone and it was its first performance after being "ROSified". However, it still acted semi-autonomously.
- **2015/11 - ScienceClub**: Next year the title of the event was "NAO, an empathetic or just amusing robot?". Body gestures were integrated and chatting abilities were shown. The key object recognition was tested together with the face and sound localization behaviours. Video available[11].
- **2016/02 - Discrete event at the Faculty**: A discrete event was organized at the Faculty in order to be able to evaluate the applause classification and emotional state gesture reproduction modules.
  Thinking and singing preamble gestures were used. Video available[12].
- **2016/09 - Closing of a Summer University Course**: *BertsoBot* was invited to the closing of a course entitled "Educational assessment: unresolved matter" (organized by the University of Basque Country). It was not a *bertso-saio* event but it covered all aspects of the interaction. It was a short exhibition in which NAO sang only one verse and thus, the applause feedback only allowed to reflect a Calm state. Unfortunately, we have no media of the event.
- **Lab demonstration**: A rehearsal without audience recorded at our laboratory[13] exhibits the global behaviour of the *BertsoBot* system in a performance similar to *bertsolari*-s events, in which two NAOs act as troubadours and the roll of the emcee is performed by Galtxagorri. The robotic emcee establishes the rules of the duel: who starts, the exercises and the flux of the performance.

6. Further Work

The work carried out during this project has revealed many promising areas of further research, such as in computer-based poetry and social robotics fields.

Regarding to the further research in computer-based poetry, we are working to improve the verse generation module by generating impromptu verses using Markov chains, and applying sentiment analysis to build higher quality poems.

On the other hand, we are considering to dynamically adapting the different DoFs to express emotions instead of precompiled gestures.

Acknowledgements.

This work has been partially supported by the Basque Government (IT900-16), the Spanish Ministry of Economy and Competitiveness MINECO/FEDER (TIN 2015-64395-R). Author Rodriguez has received the UPV/EHU research Grant PIF13/104.

11https://www.youtube.com/watch?v=IMMXHWB2mZA
12https://www.youtube.com/watch?v=SdxNgmV3CzA
13https://www.youtube.com/watch?v=UNhvd2qbuaY

References

Astigarraga, A., Agirrezabal, M., Lazkano, E., Jauregi, E., and Sierra, B., 2013. Bertsobot: the first minstrel robot. In 6th International Conference on Human System Interaction, 129–136.
Astigarraga, A., Jauregi, E., Lazkano, E., and Agirrezabal, M., 2014. Textual coherence in a verse-maker robot. In Human-Computer Systems Interaction: Backgrounds and Applications 3, 275–287.
Bensky, A., 2016. Wireless positioning technologies and applications. Artech House.
Breazeal, C., 2004. Designing sociable robots. Intelligent Robotics and Autonomous Agents. MIT Press.
Cañamero, L. and Lewis, M., 2016. Making "new ai" friends: Designing a social robot for diabetic children from an embodied AI perspective. Social Robotics, 8:523–537.
Chen, Y., Wu, F., Shuai, W., Wang, N., Chen, R., and Chen, X., 2015. Kejia robot–an attractive shopping mall guider. In International conference on Social Robotics, 145–154. Springer.
Feil-Seifer, D. and Matari c, M. J., 2011. Socially assistive robotics. IEEE Robotics & Automation Magazine, 18(1):24–31.
Fernandez, J. and Bonarini, A., 2014. Theatrebot: A software architecture for a theatrical robot. In Towards Autonomous Robotic Systems, 446–457. Springer.
Fong, T., Nourbakhsh, I., and Dautenhahn, K., 2003. A survey of socially interactive robots. Robotics and autonomous systems, 42(3):143–166.
Garzia, J., Sarasua, J., & Egaña, A., 2001. *The art of bertsolaritza: improvised Basque verse singing*. Bertsolari liburuak.
Gervás, P., 2013. Computational modelling of poetry generation. In Artificial Intelligence and Poetry Symposium, AISB Convention.
Goodrich, M. A. and Schultz, A. C, 2007. Human-robot interaction: a survey. Foundations and trends in human-computer interaction, 1(3):203–275.
Hernaez, I., Navas, E., Murugarren, J., and Etxebarria, B., 2001. Description of the ahotts system for the basque language. In 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis.
Höök, K, 2009. Affective loop experiences: designing for interactional embodiment. Philosophical transactions of the royal society B, 364:3585–3595.
Kanda, T., Shimada, M., and Koizumi, S., 2012. Children learning with a social robot. In Human-Robot Interaction (HRI), 351–358. IEEE.
Kraemer, F., Rodriguez, I., Parra, O., Ruiz, T. and Lazkano, E., 2016. Minstrel robots: Body language expression through applause evaluation. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on* (pp. 332-337). IEEE.
Laborde, D., 2005. La M emoire et l'instant. Les improvisations chant ees du bertsulari basque. Elkar.
Li, B. Robot. http://www.blancali.com/en/event/99/robot.
Lin, C., Tseng, C., Teng, W., Lee, W., Kuo, C., Gu, H., Chung, K., and Fahn, C., 2009. The realization of robot theater: Humanoid robots and theatric performance. In International Conference on Advanced Robotics (ICAR), 1–6.
Murphy, R., Shell, D., Guerin, A., Duncan, B., Fine, B., Pratt, K., and Zourntos, T., 2011. A midsummer nights dream (with flying robots). Autonomous Robots, 30(2):143–156.
Oliveira, H. G., 2009. Automatic generation of poetry: an overview. Technical report, Universidad de Coimbra.
Paiva, A., Leite, I., and Ribeiro, T., 2015. The Oxford Handbook of Affective Computing, chapter Emotion Modelling for Social Robots. Oxford University Press.
Quinlan, J. R., 2014. C4. 5: programs for machine learning. Elsevier.
Rodriguez, I., Astigarraga, A., Ruiz, T., and Lazkano, E., 2016. Singing minstrel robots, a means for improving social behaviors. In *Robotics and Automation (ICRA),* (pp. 2902-2907).
Somers-Willett, S. B., 2009. The cultural politics of slam poetry: race, identity, and the performance of popular verse in America. University of Michigan Press.
Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001, volume 1, I–511.

11. Kapitulua

Konputagailu-burmuin interfazeak

Arlo honetako gure bi interes-eremu nagusiak hauek izan dira: burmuineko emozio-kokalekuak identifikatu eta publikoaren erreakzioak burmuin-aktibazio bidez jaso ahal izatea eta, robotarekin pentsamendu-bidez komunikatzea. Urrun geratu zaizkigu bi helburuak. Ikergai eremu zabalak dira eta beren horretan tesi lan osoa beharko lukete helburuetara hurbildu ahal izateko.

Oinarrizko landa-ikerketara mugatu da beraz gure egitekoa. Egindako ekarpena, hala ere, nabarmentzeko modukoa iruditzen zaigu eta hargatik gehitu dizkiogu ikerketa lan orokorrari.

Lehenbiziko helburua gogoan, (Astigarraga et al., 2016) lanean, motor imajinagintza landu genuen. Motor imajinagintza prozesu mentala da, non erabiltzaileak agindu motoreak (mugitu, gelditu eta antzekoak) imajinazio-bidez aktibatzen dituen. Era honetako atazetan, funtsezkoa izaten da EEG sentsoreen kokapena eta kanalen aukeraketa. Gure ikerketa-lanean, EEG sentsoreak erabiliz, erabiltzailearen araberako kanal aukeraketa automatikorako metodoa proposatzen da, bi pausutako prozesuan: lehenbizi, *greedy* motako bilaketa algoritmoak bilaketa-eremua zehazten du, eta, segidan, EDA (Estimation of Distribution Algorithm) metodoa aplikatzen da aukeratutako eremuan kanal azpimultzo optimoa lortzeko. Emaitzek proposatutako metodoaren egokitasuna islatzen dute.

Ondoko artikuluak jasotzen du aipatutako esperimentua:

- **User adapted motor-imaginary brain-computer interface by means of EEG channel selection based on estimation of distributed algorithms**. A. Astigarraga, A. Arruti, J. Muguerza,

R. Santana, J.I. Martin, B. Sierra. Journal of Mathematical Problems in Engineering, vol. 2016, article ID 1435321, 12 pages.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 1435321, 12 pages
http://dx.doi.org/10.1155/2016/1435321

Research Article

User Adapted Motor-Imaginary Brain-Computer Interface by means of EEG Channel Selection Based on Estimation of Distributed Algorithms

Aitzol Astigarraga,[1] Andoni Arruti,[2] Javier Muguerza,[2] Roberto Santana,[1] Jose I. Martin,[2] and Basilio Sierra[1]

[1] *Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Computer Science Faculty, 20018 Donostia-San Sebastian, Spain*
[2] *Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU, Computer Science Faculty, 20018 Donostia-San Sebastian, Spain*

Correspondence should be addressed to Andoni Arruti; andoni.arruti@ehu.es

Received 4 June 2014; Accepted 19 November 2014

Academic Editor: Yudong Zhang

Brain-Computer Interfaces (BCIs) have become a research field with interesting applications, and it can be inferred from published papers that different persons activate different parts of the brain to perform the same action. This paper presents a personalized interface design method, for electroencephalogram- (EEG-) based BCIs, based on channel selection. We describe a novel two-step method in which firstly a computationally inexpensive greedy algorithm finds an adequate search range; and, then, an Estimation of Distribution Algorithm (EDA) is applied in the reduced range to obtain the optimal channel subset. The use of the EDA allows us to select the most interacting channels subset, removing the irrelevant and noisy ones, thus selecting the most discriminative subset of channels for each user improving accuracy. The method is tested on the IIIa dataset from the BCI competition III. Experimental results show that the resulting channel subset is consistent with motor-imaginary-related neurophysiological principles and, on the other hand, optimizes performance reducing the number of channels.

1. Introduction

Recent advances in Cognitive Neuroscience and Brain Imaging technologies provide us with the ability to interact directly with the human brain, offering thus an alternative to natural communication and control. In recent years, many Brain-Computer Interfaces (BCIs) have been built [1].

The aim of a BCI system is to establish a communication method that translates human intentions, mental tasks reflected by suitable brain signals (e.g., electric, chemical, and blood flow changes), into a control signal for an output device such as a computer application or a neuroprosthesis, not requiring any muscular response. The idea is to provide a new communication method to people who are paralysed but are cognitively intact. Motor-Imaginary BCI systems are based on the fact that imagination of movement changes brain activity in the cortex. Therefore, the recognition of patterns associated with certain movements could be used to generate control signals.

Two general BCI systems can be found in literature: invasive technologies, in which sensors are implanted directly in the brain, and noninvasive technologies, which measure brain activity using external sensors. In most BCI applications, brain signals are measured by electroencephalography (EEG) sensors, recording electrical activity from the scalp with electrodes. They are inexpensive, portable and their temporal resolution is very good.

However, there are several challenges. For example, when developing new BCI systems, the selection of specific channels to be used for each subject is a crucial aspect. However, finding optimal channel number and their positions is still a challenging task [2]. In such cases, the most common way is to use a large number of channels for signal classification. But applying a large number of channels is time consuming and may include noisy and redundant signals that degrade the BCI system performance.

Therefore, selecting the least number of channels that yield the best or required accuracy is crucial. Channel selection can be performed manually [3], based on neurophysiological knowledge, but this approach does not always guarantee optimal results [4]. Moreover, automated channel selection algorithms can provide optimal channel positions without any prior knowledge about the task at hand. Two types of automatic channel selection approaches can be found in literature: subject-independent methods, in which the electrode subset is shared between all subjects, and subject-dependent methods, in which the channel subset is customized for each subject in order to improve the individual BCI performances. This paper focuses on subject-dependent channel selection, and its main goal is the automatic selection of a reduced number of channels (and consequently their location) adapted to each subject, maintaining, or even improving, the classification accuracy in EEG-based BCI.

Our proposed method decomposes the channel selection problem into two steps: estimation of a suitable channel range by means of a backward greedy algorithm and optimal channel selection applying an evolutionary algorithm. In the first step, we propose to use a computationally inexpensive greedy algorithm to estimate a suitable channel range—this is made in a similar way as in [5]. In the second step, the optimal channels are selected applying an evolutionary algorithm. In particular, we investigate the adequateness of an Estimation of Distribution Algorithm (EDA) [6], an evolutionary algorithm based on probabilistic modelling of the search space, to exploit the channel's relationship and pick out the subset that yields the best performance. The ability of Estimation of Distribution Algorithms to maintain the quality of the joint probability distribution among the selected features is remarked by Yin and Wu [7]. In order to evaluate the performance of the proposed approach, public data available from BCI competition has been used. The main goal of this paper is to adapt BCI-EEG interfaces to each individual by selecting those channels which better discriminate among the different actions to be performed.

The paper is structured as follows: Section 2 discusses the related work. In Section 3 the two-step channel selection method (greedy algorithm + EDA) is described. Section 4 shows the experimental setup, the data acquisition procedure, and the characteristics of the EDA implementation. Experimental results are given in Section 5 and Section 6 presents the conclusions and future work.

2. Related Work

Several channel selection methods have been proposed in the literature [8–10]. An updated overview can be found in [2, 4]. Most relevant methods include common spatial patterns (CSP) [11], Support Vector Machines (SVM) [12], and methods based on the mutual information (MI).

The classical backward greedy strategy has been also applied in many research works [8, 13]. In this method the subset is built iteratively. It starts with a full set of electrodes and, at each step, the electrode that maximizes the accuracy of the complementary subset is removed.

More closely related to our approach, previous researches have addressed the channel selection problem using evolutionary algorithms [4, 9, 14–17]. However, the application of EDAs has been constrained to the analysis of Magnetoencephalography (MEG) data in the context of multiobjective optimization [18], an approach with important differences with the one introduced in this paper. Nevertheless, we briefly review this related work to emphasize differences and similarities with our work.

Wei et al. [17, 19] apply genetic algorithms (GAs) for the analysis of multichannel electrocorticogram (ECoG) recordings in a ECoG-based BCI. They combine the application of the CSP method for feature extraction, Fisher discriminant analysis as classification method, with the use of a GA for feature selection. The authors acknowledge the ability of the GA-based approach to reduce the number of features without losing classification accuracy.

In [9], a GA is combined with a multilayer neural network (ML-NN) to find a subset of channels that maximize the ML-NN's classification accuracy. They use EEG and ECoG recordings. A particular characteristic of this work is that channel selection is not directly accomplished by the GA by optimizing the accuracy. Instead, the fitness function used in the GA is the training error when an MLP is trained for a limited number of epochs. Channels are selected a posteriori by analysing the best solutions of the GA, which includes a nonautomatic part that cannot be analysed.

In [14], a multiobjective evolutionary algorithm (EA) is applied to channel selection. The idea of simultaneously optimizing two or more objectives, including the selection of an optimal set of channels, is very promising for classification of brain recordings. However, a difficulty is that the multiobjective search can be much more complex than single-objective optimization, and this difficulty rapidly increases with the number of objectives involved. Also in [18], multiobjective EAs are applied for channel selection in MEG recordings. In this case, each objective corresponds to maximizing the classification accuracy for a different individual. A regularized logistic regression method is used as a classifier, and a posteriori analysis of the best solutions obtained serves to estimate the relevance of each channel. There is not a sound experiment concerning the real contribution of the evolutionary algorithm.

Kee et al. [4] propose the combination of a Bayesian linear discriminant analysis classifier with a GA for automatic channel selection for a P300 BCI. The authors recognize that a "GA exhibits great potential to study the correlation and the joint effect of various channel combination as the fitness value is not biased to the performance of individual channel." However, the limitations of classical EAs like GAs to capture and respect the dependencies between interacting variables are not discussed. EDAs are conceived to automatically identify and represent these types of dependencies.

More recently, Bhattacharyya et al. [20] suggest the use of a combination of a Learning Automata (LA) and Differential Evolution (DE) [21] for feature extraction in the analysis of

EEG data in BCI experiments. Here, the LA-DA is not used for channel selection since features are extracted from only two channels (C3 and C4). Instead, it is applied to select the features that are passed to the classifier. The selection of those two channels is based on the authors experience, but the generalization of this approach is not clear.

There are other evolutionary computation algorithms to have in mind [22]. Nevertheless, the benefits of using EDAs that model the interactions between the variables over evolutionary algorithms that do not take correlations into account have been documented in previous work [6, 23].

To summarize, previous works have proved that channel selection by means of evolutionary algorithms can improve classification accuracy across a variety of brain signals, that is, EEG, ECoG, and MEG. However, these research works have been focused on the application of GAs and other standard EAs that do not explicitly capture and represent the dependencies between the channels. As recognized by some of the authors, the potential of the wrapper approach [24] is its capacity for considering the synergies between the features during the classification process. But in order to exploit these synergies, the EAs should be able to identify the most important interacting features and use this information at the time of generating new solutions. This is exactly what EDAs do.

While filter-based methods could in principle be applied for channel selection, these methods cannot cope with intricate and higher order dependencies between the channels, as our wrapper-type EDA approach does. In addition, the analysis of the probabilistic models generated by the EDA can provide information about what the relevant interactions for the problem are. However, in this paper, we use probabilistic models only to make a more efficient generation of solutions and leave the analysis of the potential information captured by the model as a topic of future research.

3. Proposed Approach

Having to personalize BCI-EEG interfaces to each user as main goal, this paper proposes a novel two-step method for personalized optimal channel selection. Firstly, the system selects an adequate range of number of channels to be explored in the second step. This selection is done by applying a simple greedy algorithm. Thereafter, a more computationally expensive EDA is applied to find the final channel selection. The main reason to choose EDA for the second step is its ability to capture the relevant relationships between the variables of the optimization problem, and, thus, the selection of the channels will be more accurate, reducing the complexity of the BCI system.

Both the greedy algorithm and the EDA will work with a population of candidate solutions to the problem. A solution $\mathbf{x} = (x_1, \ldots, x_{60})$ will be defined as a 60-tuple of binary 0, 1 values—the so-called Binary Encoding. Each position in the tuple refers to a concrete EEG channel, and the value indicates whether this channel is used (1 value) or not (0 value). Furthermore, the two algorithms require the use of a fitness function to evaluate the quality of the explored solutions. In this work we used the common spatial patterns (CSP) method for feature extraction from the EEG raw signals, and the accuracy of a Support Vector Machine classifier as fitness function, because of its good performance in comparison with other paradigms [25] (see Section 4.2 for more details).

In the next two subsections we explain the main characteristics of the proposed greedy algorithm and the EDA.

3.1. Greedy Search Approach. For the first step we designed a simple greedy search algorithm to reduce the search space (see Algorithm 1). The aim is to ease the search to the EDA making a rough estimation of the adequate number of channels.

In this method, we start with a solution $\mathbf{x}_{60}$ that uses the full set of electrodes (the 60 elements set to one). At each iteration l, a population D_l of solutions is generated by removing one electrode of the best solution selected in the previous iteration ($60 - l + 1$ possibilities). The solution that gives the best score when evaluated with the fitness function is chosen as the best distribution $\mathbf{x}_{60-l}$ for that number of electrodes.

Simplicity of the greedy algorithm makes a good choice for a rough selection of the range of channels to be used in the more refined search.

3.2. Estimation of Distribution Algorithms. Estimations of distribution algorithms have successfully been developed for combinatorial optimization [6]. They combine statistical learning with population-based search in order to automatically identify and exploit certain structural properties of optimization problems.

EDAs typically work with a population of candidate solutions to the problem, starting with the population generated according to the uniform distribution over all admissible solutions.

The population is then scored using a fitness function. This fitness function gives a numerical ranking for each candidate, with the higher the number the better the solution. From this ranked population, a subset of the most promising solutions are selected by the selection operator. An example selection operator is truncation selection with threshold τ = 50%, which selects the 50% best solutions. The algorithm then constructs a probabilistic model which attempts to estimate the probability distribution of the selected solutions. Once the model is constructed, new solutions are generated by sampling the distribution encoded by this model. These new solutions are then incorporated back into the old population, possibly replacing it entirely. The process is repeated until some termination criteria are met (usually when a solution of sufficient quality is reached or when the number of iterations reaches some threshold), with each iteration of this procedure usually referred to as one generation of the EDA.

4. Experimental Setup

The performed experiment is explained in this section. First of all, we present the used dataset and how it has been obtained, then details about how the fitness function is

(1) $\mathbf{x}_{60}$ ← solution with all the components (60) set to one.
(2) $l = 1$
(3) **do** {
(4) D_l ← Generate $60 - l + 1$ different solutions removing one channel from $\mathbf{x}_{60-l+1}$
(5) Compute the fitness function for all the solutions in D_l
(6) $\mathbf{x}_{60-l}$ ← Select the solution with maximum fitness function value
(7) $l = l + 1$
(8) } **until** $l = 60$

Algorithm 1: Population-based greedy search.

calculated are given, and, finally, the EDA implementation is commented.

4.1. Data Acquisition. In this work we have selected the IIIa dataset from the BCI competition III [26], because it is publicly available and has been widely used for benchmark evaluation. It contains data from 3 subjects: K3b, K6b, and L1b, collected as follows [27]:

(1) Each subject, sitting in front of a computer, was asked to perform imaginary movements of the left hand, right hand, tongue, or foot during a specified time interval according to a cue. The order of cues was random.

(2) 60 electrodes were placed on the subject's scalp (see Figure 1(a)) recording a signal sampled at 250 Hz and filtered between 1 and 50 Hz using a Notch filter.

(3) As shown in Figure 1(b), each trial started with a blank screen. At $t = 2$ s, a beep was generated and a cross "+" was shown to attract the subject's attention. At $t = 3$ s an arrow pointing to the left, right, up, or down was shown for 1 s and the subject was asked to try one of four imaginary movements until the cross disappeared at $t = 7$ s. This was followed by a 2 s break, and then the next trial began.

The dataset contains 360 instances (cases) for subject K3b, 240 for K6b, and 240 for L1b. Each instance was labelled as belonging to one of the four classes. Each subject contains a balanced distribution of the classes. Two data files are available for each subject: training and testing. The number of instances in the training and testing datasets was equal for all subjects and was 180 for K3b, 120 for K6b, and 120 for L1b. The distribution of the classes was equal in both training and testing data.

The three subjects had different amounts of experience in BCI training. K3b was the most experienced, L1b had less experience, and K6b was a beginner. This has a great influence in classification results (K3b presents the highest accuracy and K6b the lowest) as can be seen in the work of AlZoubi et al. [25] or in [28].

In this work, being our goal the study of the channel selection problem, we will maintain the methodology used in one of these studies [25] as a reference.

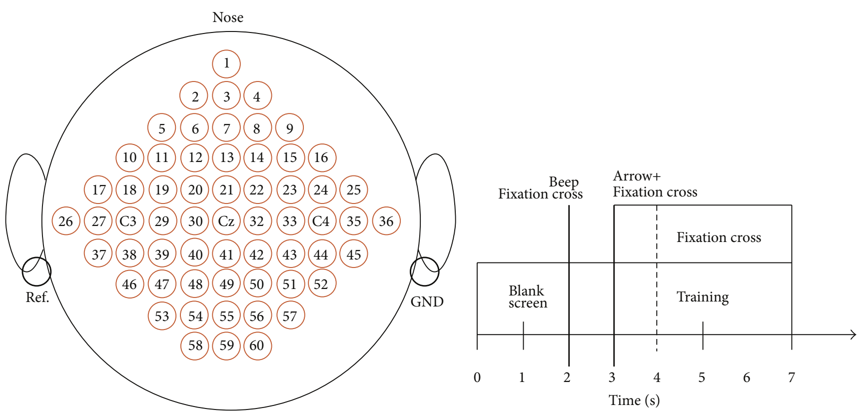
4.2. Fitness Function. As said before, both algorithms (greedy and EDA) need to evaluate each individual in the population of possible solutions. For this, a classifier using the corresponding subset of channels is built and its mean accuracy is used as a fitness function. For comparison reasons, we have applied the same methodology used by AlZoubi et al. in [25]. These are the steps followed to compute the fitness function for a given channel selection:

(i) *Feature Extraction*. Firstly, the CSP method is applied to the channel-reduced raw EEG data. For each class, applying CSP for that class versus the others, a reduced set of 5 projection signals is obtained. Then 3 frequency band filters (for 8–12 Hz, 21-20 Hz, and 20–30 Hz) are applied, and, finally, 7 features are extracted: max, min, and mean voltage values, voltage range, number of samples above zero volts, zero voltage crossing rate, and average signal power. This process gives 420 (4 classes × 5 projections × 3 filters × 7 features) numeric features for each case of the dataset.

(ii) *Training and Cross-Validation*. In [25] twelve classification paradigms are compared, and for this research we have chosen Support Vector Machines (SVM) [29, 30] because their results are among the best. The value of the fitness function is the mean accuracy obtained with 10-fold cross-validation applied to the training dataset. This has been calculated by the data mining software package WEKA [31] (default parameters are used).

4.3. Characteristics of the EDA Implementation. One distinguished feature of EDAs is the type of probabilistic model they use. They should be able to capture the relevant relationships between the variables of the optimization problem. These relationships are encoded in terms of statistical dependencies. Complex models are able to represent higher order dependencies between the variables but they are also more computationally costly due to the requirements of the learning and sampling procedures. As the probabilistic model of choice for the EDA, we use a tree, a model that exhibits a fine balance between the power of representation and the computational cost.

Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote a vector of discrete random variables. We will use $\mathbf{x} = (x_1, \ldots, x_n)$ to denote



(a) EEG electrode placement

(b) Data acquisition paradigm

Figure 1: Data acquisition schemata (dataset IIIa, BCI competition III).

an assignment to the variables. A probability distribution $p_{\mathcal{T}}(\mathbf{x})$ that is conformal with a tree is defined as $p_{\mathcal{T}}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i \mid \mathrm{pa}(x_i))$, where $\mathrm{pa}(x_i)$ is the parent of x_i in the tree, and $p(x_i \mid \mathrm{pa}(x_i)) = p(x_i)$ when $\mathrm{pa}(x_i) = \emptyset$; that is, x_i is the root of the tree. The distribution $p_{\mathcal{T}}(\mathbf{x})$ itself will be called a tree model when no confusion is possible.

Trees are able to represent bivariate marginal distributions and they can be learned from data with a computational complexity of n^2. In order to learn the structure from a dataset, the univariate and bivariate probabilities are, respectively, calculated for every variable and pair of variables. Using the marginal probabilities, the mutual information between each pair of variables is computed. The tree structure itself is determined using the Chow-Liu algorithm [32]. It works by computing the maximum weight spanning tree from the matrix of mutual information between pairs of variables. We set a threshold on the minimal mutual information value required to connect two variables. This allows for representing disconnected trees, that is, a forest.

Algorithm 2 shows the different steps of Tree-EDA. The algorithm starts by generating the random population of solutions D_0. In each iteration l, a set of solutions D_{l-1}^s is selected from population D_{l-1} and from this population the probabilistic model is learned.

The Tree-EDA shown in Algorithm 2 incorporates the method for learning the tree structure. To sample new solutions from a tree, probabilistic logic sampling [33] is applied to sample. In this method, the roots of each tree are initially sampled according to their univariate probabilities. The rest of the variables are sampled following the order determined by the trees structures and using the conditional probability distributions. As a selection method, Tree-EDA uses truncation selection in which the $T = 50\%$ best percentages of the population (highest objective values) are selected. The new sampled solutions are combined with the set of best solutions (elitist solutions) selected from the previous iteration.

EDAs that used trees were originally introduced in [34] and those that employ forests were introduced in [35]. The scheme of Algorithm 2 corresponds to the algorithm introduced in [36]. It has been implemented in MATLAB using the MATEDA software [37], a highly modular implementation in which each EDA component (either added by the user or already included in the package) is implemented as an independent program.

In this paper we use as stop criterion a fixed number of generations. The Tree-EDA parameters used in our experiments are presented in Section 5.1.

4.4. Limiting the Number of Channels of EDA Population. EDA is an evolutionary algorithm that can change the number of channels in any direction while trying to optimize the fitness function. In this work, being our goal to reduce the number of channels, we will need to fix an upper limit for number of channels at each experiment. We will achieve this with two actions:

(i) setting a unitation constraint when generating the first population: we will generate solutions uniformly distributed but with a fix number of channels;

(ii) repairing the sampled population at each generation: solutions with a number of channels greater than limit are truncated randomly before evaluating them.

5. Experimental Results

In this section we analyse the overall performance of the proposed approach in the channel selection problem. We evaluate the behaviour of the system in terms of the accuracy

```
D_0 ← Generate M solutions randomly
l = 1
do {
    D^s_{l−1} ← Select N ≤ M solutions from D_{l−1} according to a selection method
    Compute the univariate and bivariate marginal frequencies p^s_i(x_i | D^s_{l−1}) and p^s_{i,j}(x_i, x_j | D^s_{l−1}) of D^s_{l−1}
    Calculate the matrix of mutual information using bivariate and univariate marginals.
    Calculate the maximum weight spanning tree from the matrix of mutual information.
    Compute the parameters of the model.
    D_l ← Sample M solutions (the new population) from the tree and add elitist solutions.
    l = l + 1
} until A stop criterion is met
```

Algorithm 2: Tree-EDA.

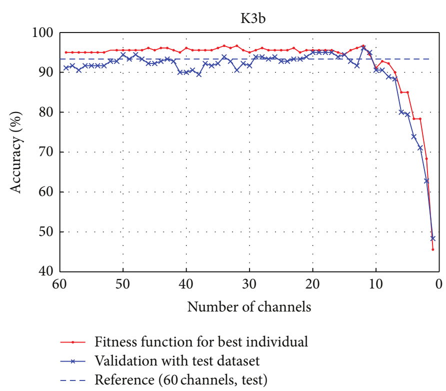


Figure 2: Results obtained with greedy search algorithm for subject K3b.

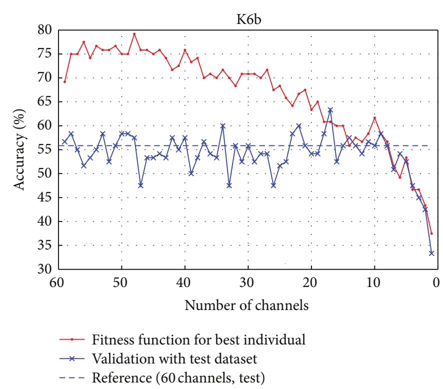


Figure 3: Results obtained with greedy search algorithm for subject K6b.

obtained using a standard classification paradigm. Furthermore, we compare the results of the complete system with those obtained by the greedy algorithm.

Obtained results applying the greedy algorithm to the three subjects are shown in Figures 2, 3, and 4. The figures represent the fitness function for the best solution at each iteration, the validation with testing dataset also for the best solution at each iteration; the results obtained with the testing dataset when all channels are used for classification are also shown as baseline reference.

As it can be seen, for the three subjects, the results obtained with 60 electrodes are similar to those obtained after the channel number has been decreased, and the obtained accuracy only decreases when less than 10 electrodes are used.

It can be seen, as said before, that the performance of the classifier depends strongly on the subject, very good for K3b, and poor for the untrained K6b and L1b. Surprisingly, this simple search algorithm shows clearly that, for the three subjects, the number of channels can be reduced drastically without losing accuracy.

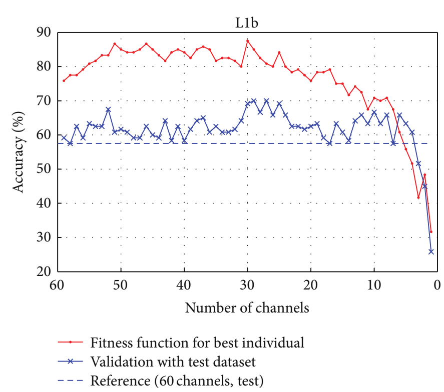


Figure 4: Results obtained with greedy search algorithm for subject L1b.

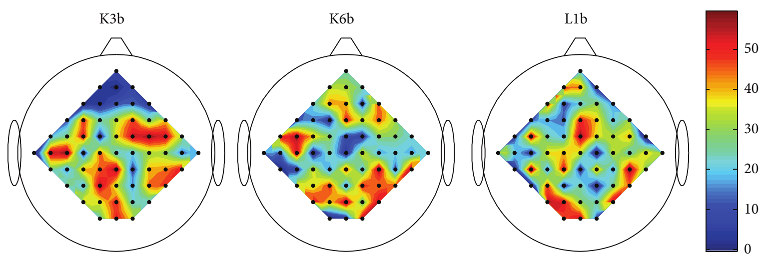


Figure 5: Channels position and frequency using greedy search algorithm.

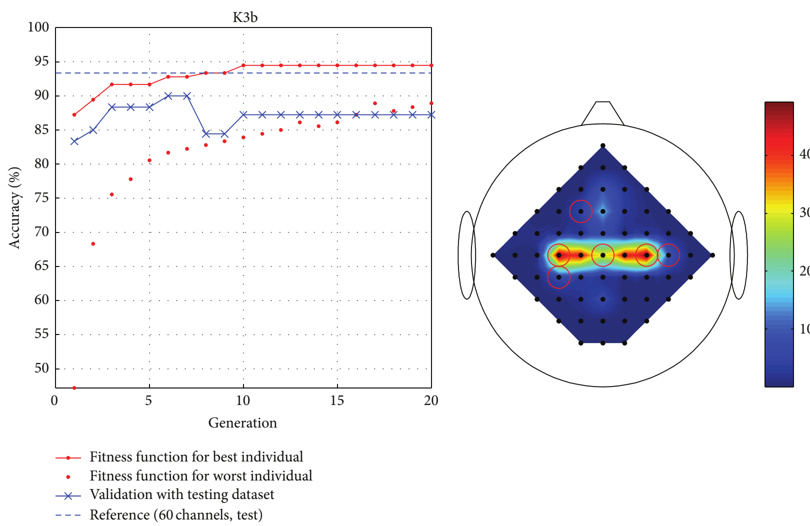


Figure 6: Results obtained with EDA search algorithm with a maximum of 6 channels for subject K3b.

In Figure 5 a topographical map for each subject shows the frequency of occurrence of the channels in the 60 best solutions (one for each iteration). The low values correspond to first discarded channels and the highest to the most significant for the classification task. It can be seen that the distribution is subject-dependant, thus justifying the development of an automatic method for searching the optimal subset of channels, making the adaptation of the system to each subject automatic.

5.1. EDA Approach. Seeing the results obtained with the greedy algorithm, we decided to carry out a sequence of experiments with the EDA, changing the top channel number limit in the interval from 16 to 1, wider than the minimum range (from 10 to 1) observed with the greedy algorithm. This focuses on the work of the EDA to a range where doing a more refined search could be interesting.

With respect to EDA parameters, and based on preliminary experiments, the population size is 200 and the number of generations 20.

To show a sample of the evolution of the EDA, Figures 6, 7, and 8 show intermediate results for the 3 subjects using 6 channels. The figures show the evolution of fitness function. A validation value obtained for the best solution of each generation, with the testing dataset, is also shown, but these values are not used by EDA. A topographic map shows the presence of channels in 50 solutions (the best ten in the last 5 generations). The results show that, for the subject L1b, with only 6 channels the EDA is able to find a solution even better than using all the channels. For subjects K3b and K6b the found solutions are near to the 60-channel based solution.

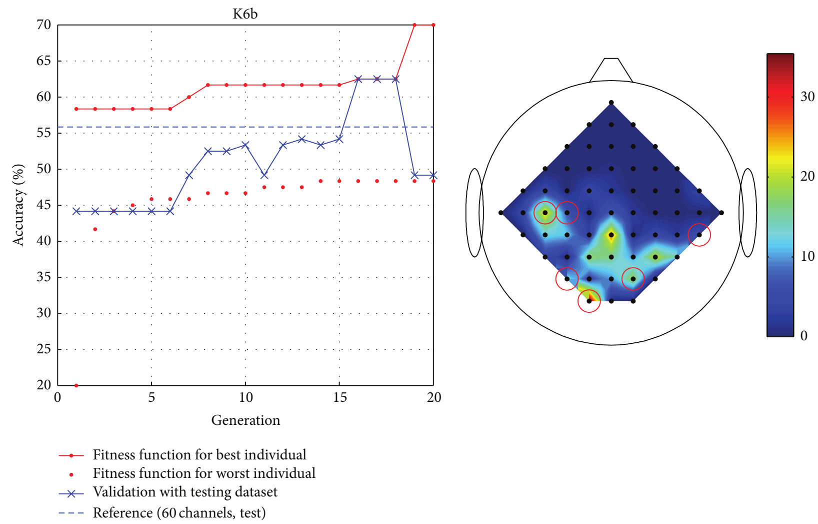


Figure 7: Results obtained with EDA search algorithm with a maximum of 6 channels for subject K6b.

Figures 9, 10, and 11 show results obtained with EDA in the last generation with different number of channels for the 3 subjects. Values obtained with the greedy search algorithm are also shown. As it can be seen, the EDA finds a better solution for the subjects K6b and L1b than the solution based on all the channels (with only 4 channels) and finds a similar solution with 10 channels for the K3b subject.

In Figure 12 channel position and frequency as a percentage, in best solutions (the best ten in the last 5 generations of 13 experiments), are shown for the tree subjects. This figure shows again the different distribution of the selected channels for the three subjects, making a specific selection of channels for each subject necessary.

Finally, the results obtained with the EDA have been compared with the results obtained with the greedy algorithm. We applied a Wilcoxon signed ranks test (0.05 confidence level) to determine if statistically significant differences are between the results of both algorithms. The test shows that the differences are statistically significant (p value = 0.037).

Therefore, the combination of the two algorithms in a two-step system is a good option to achieve a simpler user adapted interface maintaining or, even, improving the accuracy of the system.

5.2. Final Discussion. Experimental results show the adequacy of the proposed approach. On the first step, a greedy approach has been used in order to show that channel number reduction is appropriate in a 60-electrode EEG-BCI model. It can be seen in the obtained results that electrode number is decreased to about ten/sixteen channels without losing accuracy for the three subjects, focusing on the search space in the second step. On the second step, the evolutionary algorithm is used to select a fixed number of electrodes (from 1 to 16). The obtained results of the EDA based approach outperform those achieved by using both the 60 channels and the greedy approach alone as well, with statistically significant differences. In addition, our experiments show that the combination of the two algorithms in a two-step system is a good option to achieve a simpler user adapted interface maintaining or, even, improving the accuracy of the system.

It has to be noticed that, among the different brain zones, selected channels appear to be consistent with MI-related neurophysiological principles [3]. In this sense, the contribution of the performed research can be seen as a step towards a personalized EEG-based BCI interface, in which a person is first trained with a general 60-electrode system, and then the relevant ones are selected to improve the human-machine communication process.

6. Conclusions and Future Works

In this paper a two-step system for channel selection by means of EDA has been presented, aiming at maintaining or even improving the classification accuracy with a few EEG channels automatically selected for each subject. The motivation behind our research work is that electrode signals received by an EEG-BCI interface are not independent among them and that selecting an optimal subset of the electrodes can improve the results when the goal is to identify motor imagery states. In this way, the fine selection

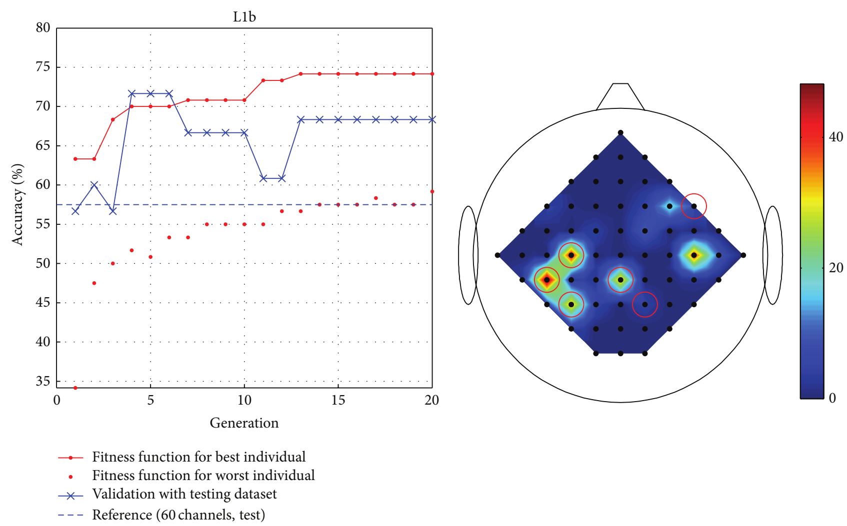


Figure 8: Results obtained with EDA search algorithm with a maximum of 6 channels for subject L1b.

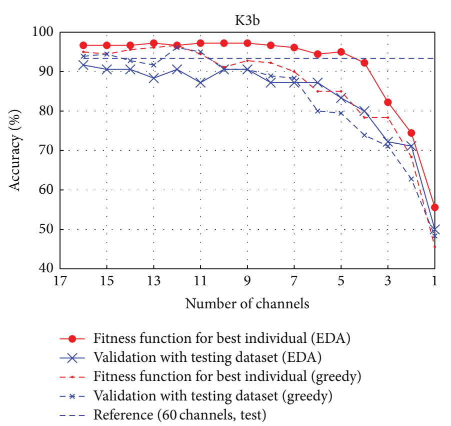


Figure 9: Results obtained with EDA search algorithm in the last generation with different number of channels for subject K3b.

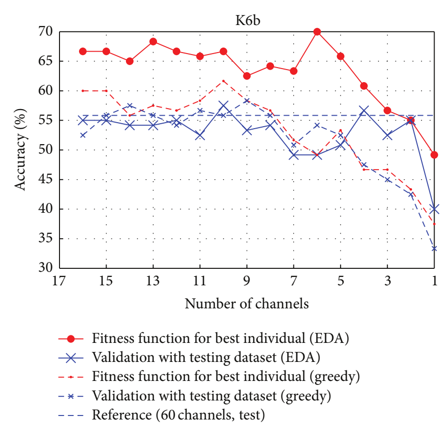


Figure 10: Results obtained with EDA search algorithm in the last generation with different number of channels for subject K6b.

is performed using an evolutionary computing paradigm—EDA—which looks for, and maintains, a relationship among the selected input channels, thus reducing or suppressing existing redundancies among the channels. The obtained results show that the combination of the greedy and EDA approaches, in a two-step system, is a good option to achieve a simpler and more accurate personalized system.

As future work, along with a comparison with other methods, we are planning to apply the presented approach taking more base classifiers or multiclassifier systems [38] in the channel selection process. Other joint probability models such as Bayesian Networks could be used to extract the relationships of the channels [39]. Fitness-scaling methods will be explored to improve the EDA searching process [40].

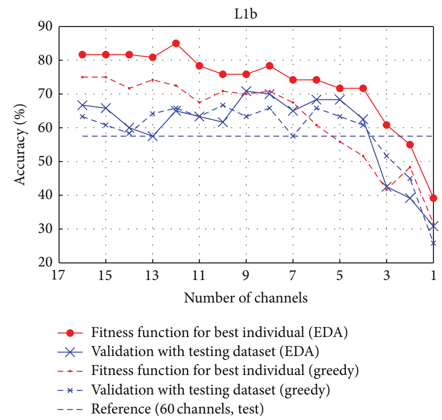


Figure 11: Results obtained with EDA search algorithm in the last generation with different number of channels for subject L1b.

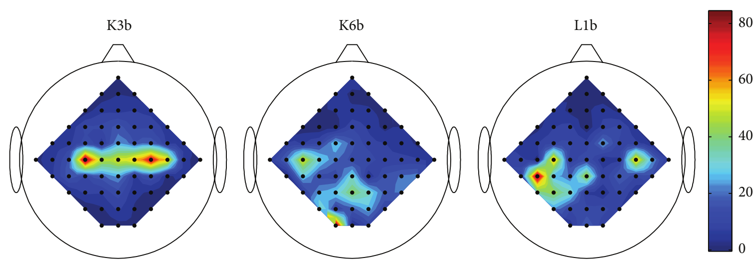


Figure 12: Channels position and frequency using EDA search algorithm.

On the other hand, in order to avoid high dimensionality, certain features must be selected before classification starts [23]. A more in-depth analysis of the channel signal is to be done, in order to detect parts of the brain activity that have more influence. A (temporal) clustering among the received signals [41] is also a line for future research which is being considered by the authors. Finally, the inclusion of nonintentional control state patterns—as fifth class—in the experiments could be a first step towards an asynchronous system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work described in this paper was partially supported by the Basque Government under Grants IT395-10, IT313-10, and IT609-13, the University of the Basque Country UPV/EHU under Grant UFI11/45, and the Ministry of Science and Technology of Spain under Grant TIN-41272P.

References

[1] C. Guger, B. Z. Allison, and G. Edlinger, *Brain-Computer Interface Research: A State-of-the-Art Summary 2*, Springer, New York, NY, USA, 2014.

[2] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek, "Optimizing the channel selection and classification accuracy in EEG-based BCI," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1865–1873, 2011.

[3] M. Thulasidas, C. Guan, and J. Wu, "Robust classification of EEG signal for brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 1, pp. 24–29, 2006.

[4] C. Y. Kee, R. K. Chetty, B. H. Khoo, and S. Ponnambalam, "Genetic algorithm and bayesian linear discriminant analysis based channel Selection method for P300 BCI," in *Trends in Intelligent Robotics, Automation, and Manufacturing*, vol. 330 of *Communications in Computer and Information Science*, pp. 226–235, Springer, Berlin, Germany, 2012.

[5] T. Shibanoki, K. Shima, T. Tsuji, A. Otsuka, and T. Chin, "A quasi-optimal channel selection method for bioelectric signal classification using a partial Kullback-Leibler information measure," *IEEE Transactions on Bio-Medical Engineering*, vol. 60, no. 3, pp. 853–861, 2013.

[6] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Press, 2001.

[7] P.-Y. Yin and H.-L. Wu, "Cyber-EDA: estimation of distribution algorithms with adaptive memory programming," *Mathematical Problems in Engineering*, vol. 2013, Article ID 132697, 11 pages, 2013.

[8] A. Barachant and S. Bonnet, "Channel selection procedure using riemannian distance for BCI applications," in *Proceedings of the 5th International IEEE/EMBS Conference on Neural Engineering (NER '11)*, pp. 348–351, Cancun, Mexico, May 2011.

[9] J. Yang, H. Singh, E. L. Hines et al., "Channel selection and classification of electroencephalogram signals: an artificial neural network and genetic algorithm-based approach," *Artificial Intelligence in Medicine*, vol. 55, no. 2, pp. 117–126, 2012.

[10] T. N. Lal, M. Schröder, T. Hinterberger et al., "Support vector channel selection in BCI," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1003–1010, 2004.

[11] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," *Clinical Neurophysiology*, vol. 110, no. 5, pp. 787–798, 1999.

[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Statistics for Engineering and Information Science, Springer, New York, NY, USA, 2nd edition, 2000.

[13] M. Xu, H. Qi, L. Ma et al., "Channel selection based on phase measurement in P300-based brain-computer interface," *PLoS ONE*, vol. 8, no. 4, Article ID e60608, 2013.

[14] B. A. S. Hasan, J. Q. Gan, and Q. Zhang, "Multi-objective evolutionary methods for channel selection in brain-computer interfaces: some preliminary experimental results," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–6, Barcelona, Spain, July 2010.

[15] J.-Y. Kim, S.-M. Park, K.-E. Ko, and K.-B. Sim, "Optimal EEG channel selection for motor imagery BCI system using BPSO and GA," in *Robot Intelligence Technology and Applications 2012*, vol. 208 of *Advances in Intelligent Systems and Computing*, pp. 231–239, Springer, Berlin, Germany, 2013.

[16] M. Schröder, M. Bogdan, T. Hinterberger, and N. Birbaumer, "Automated EEG feature selection for brain computer interfaces," in *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, pp. 626–629, IEEE, March 2003.

[17] Q. Wei, Z. Lu, K. Chen, and Y. Ma, "Channel selection for optimizing feature extraction in an electrocorticogram-based brain-computer interface," *Journal of Clinical Neurophysiology*, vol. 27, no. 5, pp. 321–327, 2010.

[18] R. Santana, C. Bielza, and P. Larrañaga, "Regularized logistic regression and multiobjective variable selection for classifying MEG data," *Biological Cybernetics*, vol. 106, no. 6-7, pp. 389–405, 2012.

[19] Q. Wei and W. Tu, "Channel selection by genetic algorithms for classifying single-trial ECoG during motor imagery," in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '08)*, pp. 624–627, IEEE, Vancouver, Canada, August 2008.

[20] S. Bhattacharyya, A. Sengupta, T. Chakraborti, A. Konar, and D. N. Tibarewala, "Automatic feature selection of motor imagery EEG signals using differential evolution and learning automata," *Medical & Biological Engineering & Computing*, vol. 52, no. 2, pp. 131–139, 2014.

[21] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer, Berlin, Germany, 2005.

[22] X. Zhang, L. Jiao, A. Paul, Y. Yuan, Z. Wei, and Q. Song, "Semisupervised particle swarm optimization for classification," *Mathematical Problems in Engineering*, vol. 2014, Article ID 832135, 11 pages, 2014.

[23] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra, "Feature subset selection by Bayesian network-based optimization," *Artificial Intelligence*, vol. 123, no. 1-2, pp. 157–184, 2000.

[24] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[25] O. AlZoubi, I. Koprinska, and R. A. Calvo, "Classification of brain-computer interface data," in *Proceedings of the 7th Australasian Data Mining Conference (AusDM '08)*, J. F. Roddick, J. Li, P. Christen, and P. J. Kennedy, Eds., vol. 87 of *CRPIT*, pp. 123–131, Australian Computer Society, Adelaide, Australia, November 2008, http://dblp.uni-trier.de/db/conf/ausdm/ausdm2008.html #AlZoubiKC08

[26] B. Blankertz, K.-R. Müller, D. J. Krusienski et al., "The BCI competition III: validating alternative approaches to actual BCI problems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 153–159, 2006.

[27] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller, "Characterization of four-class motor imagery EEG data for the BCI-competition 2005," *Journal of Neural Engineering*, vol. 2, no. 4, pp. 14–22, 2005.

[28] F. Lee, R. Scherer, R. Leeb, C. Neuper, H. Bischof, and G. Pfurtscheller, "A comparative analysis of multi-class EEG classification for brain-computer interface," in *Proceedings of the 10th Computer Vision Winter Workshop (CVWW '05)*, Zell an der Pram, Austria, February 2005.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.

[31] M. Hall, E. Frank, G. Holmes, B. P. P. Reutemann, and I. Witten, "The weka data mining software: an update," *SIGKDD Exploration Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[32] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[33] M. Henrion, "Propagating uncertainty in bayesian networks by probabilistic logic sampling," in *Proceedings of the 2nd Annual Conference on Uncertainty in Artificial Intelligence*, J. F. Lemmer and L. N. Kanal, Eds., pp. 149–164, Elsevier, 1988.

[34] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, D. H. Fisher, Ed., pp. 30–38, Morgan Kaufmann, San Francisco, Calif, USA, July 1997.

[35] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing—Engineering*

Design and Manufacturing, R. Roy, T. Furuhashi, and P. Chawdhry, Eds., pp. 521–535, Springer, London, UK, 1999.

[36] R. Santana, A. Ochoa, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '01)*, L. Spector, E. Goodman, A. Wu et al., Eds., pp. 543–550, Morgan Kaufmann, San Francisco, Calif, USA, July 2001.

[37] R. Santana, C. Bielza, P. Larrañaga et al., "Mateda-2.0: a MATLAB package for the implementation and analysis of estimation of distribution algorithms," *Journal of Statistical Software*, vol. 35, no. 7, pp. 1–30, 2010, http://www.jstatsoft.org/v35/i07.

[38] L. Susperregi, A. Arruti, E. Jauregi et al., "Fusing multiple image transformations and a thermal sensor with kinect to improve person detection ability," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1980–1991, 2013.

[39] B. Sierra, E. Lazkano, E. Jauregi, and I. Irigoien, "Histogram distance-based Bayesian Network structure learning: a supervised classification specific approach," *Decision Support Systems*, vol. 48, no. 1, pp. 180–190, 2009.

[40] Y. Zhang, L. Wu, and S. Wang, "UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.

[41] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013.

IV. atala

Ondorioak eta etorkizunerako lana

12. Kapitulua

Ekarpenak, ondorioak eta etorkizunerako lana

Tesi-lan honetan, *BertsoBot* sistema aurkeztu dugu, jendaurrean bertsoak sortu eta kantatzeko gai dena. Bertsobot sistema edo arkitektura, hura osatzen duten eremuetatik aztertu dugu, bakoitzaren ekarpena zehaztuz eta arkitektura orokorrean nola txertatzen den azalduz. Helburu orokorra, bertso bidez mezuak sortu eta plazaratzen dituen sistema garatzea izan da, gizaki-robot arteko komunikazio eredu gisa. Azken kapitulu honetan, egindako lanaren analisitik ateratako ondorio nagusiak eta etorkizunerako ikerlerroak zerrendatzen dira.

2012tik gaur arte, Bertsobot arkitektura aldatuz eta garatuz joan da. Lehen bertsio hartan, Bertsobota batik bat teleoperatutako robota zen, autonomiarik gabea, eta jendaurreko bere bertso-jarduna memoriarik bertsoak berreskuratu eta lau oinak emanda bertsoa osatzera mugatzen zen. Berarekiko komunikazioa ordenagailuaren bitartez burutzen genuen, hizketa-bidezko ulermen modulurik ez zeukan artean.

Gaur egungo Bertsobota robot humanoidea da, gizaki-robot elkarrekintzarako prestatua. Ohiko bertso-saio bateko *performancea* modu autonomoan burutzeko gai da: aulkian eseri, mikrofonora hurbildu, gaia jaso eta kantatu... saioaren testuinguruan, hizketa-bidez jasotako aginduak interpretatu eta ulertzeko gai da; baita soinuaren iturburua bilatu eta atentzioa gai-jartzailean edo bertso-lagunarengan jartzeko ere. Gorputz adierazpidea baliatzen du unean unean egiten ari dena adierazteko. Bestalde, bertsoak osatzeko gaitasunari dagokionez, lau oinak emanda, puntua jarrita eta gaia emanda osatu ditzake bertsoak, puntuen arteko hurbiltasun semantikoa kontuan hartuz. Estilo jakin bat imitatzeko gai da N-grama ereduak erabiliz eta baita emozio-egoera jakin bat transmititzen duten bertsoak sortzeko ere.

Bertsolaria izan dugu eredu kognitibo gisa, eta ikerketa-eremu bakoitzeko ekarpenak Bertsobot arkitektura orokorrean integratu ditugu, aurrerapausoak emanez gizaki-robot elkarrekintza eta komunikaziorako portaeren garapenean.

Hala ere, ibili beharreko bidea luzea da oraindik. Segidan, arlo bakoitzeko ondorio nagusiak eta etorkizunerako lanak aipatuko ditugu.

Bertso-sorkuntza automatikoa

Bertso-sorkuntza automatikoari buruz, azpimarratzeko puntuak hauek dira:

- Baliabide linguistiko minimoekin (testu-corpusa, silaba-kontatzailea eta errima-egiaztatzailea), bertsoak sortzeko bi estrategien diseinu eta inplementazioa: corpus-erauzketa metodoa N-grama eredu probabilistikoa.

- Koherentzia semantikoa neurtzeko metodoak (oinarrizko VSM, LSA eta Word2vec) proposatu eta testu-sorkuntzan txertatzea, bai ondoz ondoko esaldien arteko koherentzia neurtzeko eta baita diskurtso mailako koherentzia orokorra neurtzeko ere.

- Emozio egoeraren transmisioa bertso-bidez. Sarrera gisa aldarte-egoera eman eta haren araberako bertsoak sortzeko metodoa.

Ondorio gisa, esango dugu testu-sorkuntzako bi metodoak corpusean oinarrituak direla eta emaitza, neurri handi batean, corpusaren ezaugarrien araberakoa dela. Corpus ezberdinekin gehiago esperimentatzea behar dugu sarrerako testuaren ezaugarriek amaierako testua zer neurritan bideratzen duten zehatz jakin ahal izateko. N-grama sortzaileari dagokionez, algoritmoaren inplementazioaren hobetze-lanetan gabiltza exekuzio-denbora laburragoa izan dadin eta horrela, bilaketa espazioa zabaldu ahal izan dezagun.

Emozioak testu-bidez adierazteko metodoari buruz, emaitzek erakutsi digute aurrera begira lanean jarraitzeko moduko bidea dela. Positibo, negatibo eta neutral, 3 oinarrizko aldarteetatik harago joan nahi genuke eta aldarte-egoera bakoitzaren aktibazio-maila neurtu. Horrekin batera, testu bidez adierazitako sentimenduak, bat egin behar du robotaren keinu eta ahots-tonuarekin. Sentimenduaren araberako bertso-doinua aukeratzen erakutsi behar diogu baita Bertsobotari.

Semantika eta koherentzia

Beste horrenbeste esan dezakegu eredu semantikoari buruz ere. Ikasi dugun bezala, LSA metodoaren emaitzak erabilitako "kalibrazioaren" (*topic* kopurua, lematizatu bai/ez, *stopword*-ak...) araberakoak dira neurri handi batean, baina baita eredu gisa erabilitako corpusaren araberakoak ere. Corpus espezifikoek (gai jakin bati buruzkoek) emaitza hobeak eman dizkigute esperimentuetan eta bide hori aztertzen jarraitu nahi dugu. LSA metodoaz gain, semantika neurtzeko word2vec erabili dugu baita, hitz/esaldien adierazpen distribuzionalak (*word embeddings*) erabiltzen dituena. Azken honen ekarpenak eta hutsuneak hobeto identifikatzeko esperimentazio gehiago beharrezkoa da, baita bi metodoaren konbinazio posibleen azterketa ere.

Ikasketa Automatikoa

Ikasketa Automatikoa baliatuz bertsoaren azterketa diskurtsiboa burutu dugu. Alor honetako ekarpen aipagarrienak hauek izan dira:

- Agur-bertsoetako diskurtso-ezaugarriak identifikatu eta algoritmo bidez ikasteko ahalegina.

- Bertsoaren diskurtso-antolaketa, puntuen ordenazio egokia sailkatze-algoritmoen bidez.

Agur-bertsoen atazan emaitzak ez ziren espero bezain onak izan. Esperimentu gehiago burutu nahi genituzke eta, multi-sailkatzaileekin proba egin bidea erabat baztertu aurretik. Emaitzak hobetuko balira, hurrengo pausua, agur-bertsoetako puntuak ezaugarri edo kategoriatan automatikoki sailkatzen dituen sistema eraikitzea litzateke. Honekin, agur-bertso berriak osatu ahal izango genituzke, puntuz-puntu geuk adierazitako ezaugarriak betez. Adibidez: Publikoari eta herriari/lekuari erreferentzia egiten dion bertsoa osatzeko eskema posible bat honakoa litzateke: Betelana-Publikoa-Lekua-Mezua.

Puntuen ordenazio atazan, azpimarratzekoak dira lortutako emaitzak. Egindako esperimentuetan, gizakien asmatze-tasa hobetzea lortu du konputagailuak. Atazaren zailtasuna kontuan izanda (esaldi motzak, elipsi ugari, esanahi lausoa batzuetan...), bide onetik gabiltzalako seinale gisa interpretatu dugu. Hemen ere, beste corpus batzuekin esperimentatzea dugu etorkizun hurbileko lana: diskurtso antolaketa ikasteko idatzizko bertsoak erabiltzea egokia izan daitekeela uste dugu. Idatzikoetan puntuen arteko lotura, diskurtso orokorra... zainduagoak izaten direlako gehienetan. Horrekin batera, garatu dugun sailkatzailea integratu nahi dugu bertso-sorkuntza moduluan. Corpusetik puntuak ateratzean, hauen ordenazio egokia erabakitzen lagundu dezan.

Konputagailu-burmuin interfazeak

Arlo honetako gure helburu nagusia, burmuineko emozio-kokalekuak identifikatu eta publikoaren erreakzioak burmuin-aktibazio bidez jaso ahal izatea zen. Horrekin batera, robotarekin pentsamendu-bidez komunikatzea ere bai, oinarrizko aginduak bidaliz.

Helburuetatik urrun geratu gara eta oinarrizko landa-ikerketara mugatu da gure egitekoa. Aurrera begira, aplikazio-bidez publikoak bere uneko egoeraren berri eman eta informazio hori zuzenean robotari bidaltzeko modua jarri nahi genuke. Epe luzerako helburua, ataza automatizatu eta publikoaren erreakzioak burmuin-aktibazio bidez jasotzea litzatekeelarik.

Bertsobot arkitektura

Bestalde, aurkeztutako Bertsobot arkitekturari buruz azpimarratzekoak dira:

- Gizaki-robot elkarrekintza eta komunikaziorako eredu orokorra. Bertsolaria eredu-gisa hartuz, hura simulatuko duen arkitektura kognitiboaren diseinu eta inplementazioa. Bat-bateko bertsolaritzak eskatzen dituen zehaztasunen eta bertsolariak saioan erakusten duen portaeraren arabera eraikitakoa, oholtza gainean oinarrizko nabigazioa, bat-bateko mezuen sorkuntza eta emozio-egoeraren araberako gorputz adierazpidea erakutsiz. Portaera horiek garatzeko, objektuen errekonozimendua, Lengoaia Naturalaren Prozesamendua, hizketa-bidezko komunikazioa eta emozioak detektatu eta erakusteko metodoak inplementatu dira arkitekturan.

- ROS software-egitura erabiliz, inplementatutako arkitekturaren modulartasuna. Gaitasun berriak txertatu eta aurrekoekin konbinatzea errazten duena.

Robotika arloan, aurrera begirako lanen artean aipagarrienak ondokoak dira: emozioak detektatu eta erakustea funtsezkotzat daukagu, robot sozialak garatze bidean. Bertsobota publikoarengandik oinarrizko emozioak jaso (txaloen bidez) eta horien arabera erreakzionatzeko gai da. Detektatutako emozio-egoera hori testu-sorkuntza moduluarekin bat-egitea falta zaigu, biekin gorputz eta testu, emozioa egoera adierazi eta azpimarratzeko.

Ibili beharreko beste bide bat Lengoaia Naturalaren Ulermenarena dugu. Gaur gaurkoz, robotean Google Speech ASRa erabiltzen dugu. Sistemaren errekonozimendu ahalmena hobetzeko azterketa sakona behar dugu egin. Ikasi dugunaren arabera, hitz solteekin baino esaldi osoekin hobeto funtzionatzen du nabarmen. Sistemari hitz edo esaldiarekin batera testuinguruari buruzko informazioa (hitzak, esaldiak) pasatzea izan daiteke emaitzak hobetzeko modu bat. Euskararen ezaugarri lokalak (ahozko hizkerari dagozkionak, euskalkiak) hobeto tratatzeko asmoz, Aholaben ahots errekonozimenduko sistemarekin probak egingo ditugu baita. Baliteke biak konbinatzeko modua izatea baita.

Bertsolariek, oholtza gainean, kantuan hasi baino lehen doinua adostu ohi dute beren artean. Ahapetik kantatzen diete elkarri holako edo halako doinua. Bertsobotean doinu-errekonozitzailea inplementatzen ari gara *Deep Learning* algoritmoa erabiliz, ahapeka kantatutako doinu hori identifikatu eta dagokion partitura erabili dezan kantuan egiterakoan. Bertsobotarentzat ez ezik, bertso-eskolentzako tresna erabilgarria izan daitekeela uste dugu, doinutegiko aireak identifikatu eta ikasteko.

Amaitzeko, buruz-buru jarri nahi genituzke berriro ere bertsolaria eta Bertsobota. Jendaurreko saio batekin hasi genuena modu berean itxiz. Ikerketa lan honetan egindakoa gizarteratzeko modu egokia litzateke zinez.

Bibliografia

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59.

Agerri, R., Bermudez, J., and Rigau, G. (2014). Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *LREC*, volume 2014, pages 3823–3828.

Agirrezabal, M. (2011). Bertsotarako arbel digitala [digital board for making verses].

Agirrezabal, M., Alegria, I., Arrieta, B., and Hulden, M. (2012). Bad: An assistant tool for making verses in basque. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '12, pages 13–17, Stroudsburg, PA, USA. Association for Computational Linguistics.

Agirrezabal, M., Arrieta, B., Astigarraga, A., and Hulden, M. (2013). Pos-tag based poetry generation with wordnet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 162–166.

Alzua-Sorzabal, A., Fuentetaja, I., and Ariza, M. (2012). Measuring emotional states in touristic events: The case of the emocionómetro. In *Proceedings of Information and communication technologies in tourism*, pages 48–59. Springer.

Amuriza, X. (1981). *Hiztegi errimatua [Rhyme Dictionary]*. Alfabetatze Euskalduntze Koordinakundea.

Amuriza, X. (2016). *Hiztegi errimatua*. Lanku Bertso Zerbitzuak.

Arkin, R. (1998). *Behavior-based robotics*. MIT press.

Armistead, S. and Zulaika, J. (2005). *Voicing the Moment: improvised oral poetry and Basque tradition*. Number 3. University of Nevada Press.

Arvaneh, M., Guan, C., Ang, K., and Quek, C. (2011). Optimizing the channel selection and classification accuracy in EEG-based BCI. *IEEE Transactions on Biomedical Engineering*, 58(6):1865–1873.

Asher, N. and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Astigarraga, A., Arruti, A., Muguerza, J., Santana, R., Martin, J., and Sierra, B. (2016). User adapted motor-imaginary brain-computer interface by means of EEG channel selection based on estimation of distributed algorithms. *Mathematical Problems in Engineering*, 2016.

Astigarraga, A. and Lazkano, E. (2012). *Robot mugikorrak. Oinarriak*. Udako Euskal Unibertsitatea.

Astigarraga, A., Lazkano, E., Ranó, I., Sierra, B., and Zarautz, I. (2003). Sorgin: a software framework for behavior control implementation. In *Proceedings of the International Conference on Control Systems and Computer Science (CSCS)*, volume 1, pages 243–248.

Barachant, A. and Bonnet, S. (2011). Channel selection procedure using riemannian distance for bci applications. In *Proceedings of the 5th International IEEE/EMBS Conference onNeural Engineering (NER)*, pages 348–351.

Barbieri, G., Pachet, F., Roy, P., and Esposti, M. (2012). Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 115–120. IOS Press.

Barzilay, R. and Elhadad, N. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Boden, M. (1990). *The creative mind: Myths and mechanisms*. Psychology Press.

Breazeal, C. (2004). *Designing sociable robots*. Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge MA, USA.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23.

Brooks, R. (1991). Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159.

Busemann, S. and Horacek, H. (1997). Generating air quality reports from environmental data. In *Proceedings of the DFKI Workshop on Natural Language Generation*, pages 15–21.

Čapek, K. (2001). *R.U.R (Rossum's Universal Robots). 1920.* Trans. Paul Selver and Nigel Playfair. Dover Pub. Inc., Mineola, NY.

Coch, J., David, R., and Magnoler, J. (1995). Quality tests for a mail generation system. In *IA 95. Journées internationales*, pages 435–443.

Colton, S., Goodwin, J., and Veale, T. (2012). Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, pages 95–102, Dublin, Ireland.

Dahl, T. S. and Boulos, M. (2013). Robots in health and social care: A complementary technology to home care and telehealthcare? *Robotics*, 3(1):1–21.

Das, A. and Gambäck, B. (2014). Poetic machine: Computational creativity for automatic poetry generation in bengali. In *Proceedings of the 5th International Conference on Computational Creativity (ICCC)*.

Dasarathy, B. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Díaz-Agudo, B., Gervás, P., and González-Calero, P. (2002). Poetry generation in colibri. *Advances in Case-Based Reasoning*, pages 157–159.

Dorronsoro, J. (1995). *Bertso doinutegia [Verse melodies repository]*. Euskal Herriko Bertsolari Elkartea.

Egaña, A. (2007). The process of creating improvised bertsos. *Oral Tradition*, 22(2):117–142.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Fasola, J. and Matarič, M. (2012). Using socially assistive human–robot interaction to motivate physical exercise for older adults. *Proceedings of the IEEE*, 100(8):2512–2526.

Fernández, A. and Mauricio, J. (2016). *TheatreBot: Studying emotion projection and emotion enrichment system for autonomous theatrical robot*. PhD thesis, Politecnico di Milano.

Fernandez, J. and Bonarini, A. (2014). Theatrebot: A software architecture for a theatrical robot. In *Towards Autonomous Robotic Systems (TAROS)*, pages 446–457. Springer, Birminghan, UK.

Firth, J. (1957). A synopsis of linguistic theory, 1930-1955.

Forlizzi, J. and DiSalvo, C. (2006). Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 258–265.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1983). Human factors and behavioral science: Statistical semantics: Analysis of the potential performance of key-word information systems. *The Bell System Technical Journal*, 62(6):1753–1806.

García-Peñalvo, F., Rees, A., Hughes, J., Jormanainen, I., Toivonen, T., and Vermeersch, J. (2016). A survey of resources for introducing coding into schools. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 19–26. ACM.

Garzia, J. (2000). *Gaur egungo bertsolarien baliabide poetiko-erretorikoak:(marko teorikoa eta aplikazio didaktikoa)*. PhD thesis, Euskal herriko Unibertsitatea.

Garzia, J. (2007). History of improvised bertsolaritza: A proposal. *Oral Tradition*, 22(2):77–115.

Garzia, J., Sarasua, J., and Egaña, A. (2001). *Bat bateko bertsolaritza: Gakoak eta azterbideak*. Bertsozale Elkartea.

Gervás, P. (2000). Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100.

Gervás, P. (2001). An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*, 14(3):181–188.

Gervás, P. (2013). Computational modelling of poetry generation. In *Artificial Intelligence and Poetry Symposium, AISB Convention*.

Gervás, P. (2015). Deconstructing computer poets: Making selected processes available as services. *Computational Intelligence*, pages 3–31.

Gervás, P., Díaz-Agudo, B., Peinado, F., and Hervás, R. (2005). Story plot generation based on cbr. *Knowledge-Based Systems*, 18(4):235–242.

Giarratano, J. C. and Riley, G. (1998). *Expert systems*. PWS Publishing Co.

Goldberg, E., Driedger, N., and Kittredge, R. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Goodrich, M. and Schultz, A. (2007). Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275.

Gratch, J. and Marsella, S. (2005). Lessons from emotion psychology for the design of lifelike characters. *Applied Artificial Intelligence*, 19(3-4):215–233.

Guger, C., Allison, B., and Müller-Putz, G. (2015). Brain-computer interface research: A state-of-the-art summary 4. In *Brain-Computer Interface Research*, pages 1–8. Springer.

Guizzo, E. (2008). Three engineers, hundreds of robots, one warehouse. *IEEE spectrum*, 45(7):26–34.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Hanson, D. (2015). Hanson Robotics. http://www.hansonrobotics.com.

Harnad, S. (2006). The annotation game: On turing (1950) on computing, machinery, and intelligence. In *The Turing test sourcebook: philosophical and methodological issues in the quest for the thinking computer*. Kluwer.

Hayashi, Y., Murata, M., Fan, L., and Tokuhisa, M. (2013). Japanese sentence order estimation using supervised machine learning with rich linguistic clues. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 1–12.

Hernandez, J. (2011). Bertsolaritza, definizio berri baterantz? *Emakunde aldizkaria*.

Hobbs, J. (1979). Coherence and coreference. *Cognitive science*, 3(1):67–90.

Hoffman, G., Kubat, R., and Breazeal, C. (2008). A hybrid control system for puppeteering a live robotic stage actor. In *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 354–359. IEEE.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196.

Imbir, K., Jarymowicz, M., Spustek, T., Kuś, R., and Żygierewicz, J. (2015). Origin of emotion effects on erp correlates of emotional word processing: the emotion duality approach. *PLOS ONE*, 10(5):1–19.

inuktun (2017). Versatrax. http://www.inuktun.com/crawler-vehicles/.

Ishiguro, H. (2015). ATR. http://www.geminoid.jp/en/index.html.

Jauregi, O. (2013). Euskal testuetako silaba egituren maiztasuna diakronikoki. *Anuario del Seminario de Filología Vasca "Julio de Urquijo"*, 37(1):393–410.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142.

Jones, J. L. (2004). *Robot programming*. McGraw Hill.

Jordanous, A. (2015). Four pppperspectives on computational creativity. In *AISB 2015 Symposium on Computational Creativity*, pages 16–22.

Ju, Z., Yang, C., and Ma, H. (2014). Kinematics modeling and experimental verification of baxter robot. In *33rd Chinese Control Conference (CCC)*, pages 8518–8523. IEEE.

Jurafsky, D. and Martin, J. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall.

Kee, C., Chetty, R., Khoo, B., and Ponnambalam, S. (2012). Genetic algorithm and bayesian linear discriminant analysis based channel selection method for p300 bci. In *Trends in Intelligent Robotics, Automation, and Manufacturing*, pages 226–235. Springer.

Kirke, A. and Miranda, E. (2013). Emotional and multi-agent systems in computer-aided writing and poetry. In *Proceedings of the Artificial Intelligence and Poetry Symposium*.

Kleinginna, P. and Kleinginna, A. (1981). A categorized list of emotion definitions, with suggestions for a consensual definition. *Motivation and emotion*, 5(4):345–379.

Knight, H. (2011). Eight lessons learned about non-verbal interactions through robot theater. *Social Robotics*, pages 42–51.

Kortenkamp, D., Simmons, R., and Brugali, D. (2016). Robotic systems architectures and programming. In *Springer Handbook of Robotics*, pages 283–306. Springer.

Kotsiantis, S. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands. IOS Press.

Laborde, D. (2005). *La Mémoire et l'instant. Les improvisations chantées du bertsulari basque*. Elkar argitaletxea, Arsenal Plaza, Baiona.

Lakoff, G. and Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. Basic books.

Lal, T., Schroder, M., Hinterberger, T., Weston, J., Bogdan, M., Birmauer, N., and Scholkopf, B. (2004). Support vector channel selection in BCI. *IEEE Transactions on biomedical Engineering*, 51(6):1003–1010.

Lamb, C., Brown, D., and Clarke, C. (2016). A taxonomy of generative poetry techniques. In *Proceedings of the Bridges Finland Conference*, pages 195–202.

Landauer, T., McNamara, D., Dennis, S., and Kintsch, W. (2013). *Handbook of latent semantic analysis*. Psychology Press.

Larrañaga, J. (2013). *Jolas sakona: txapelketaren prozesu errituala eta bertsolariaren arrazoi sortzailea XXI mendeko agoran*. PhD thesis, Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU).

Lasarte, G., Alvarez, A., Ugalde, A., Lekue, P., and Martinez, J. (2015). Emakume bertsolariak: ahanzturatik diskurtso propiora. Technical report, Emakunde- Emakumearen Euskal Institutua.

Lazkano, E. (2004). *Pautas para el desarrollo incremental de una arquitectura de control basada en el comportamiento para la navegación de robots en entornos semi-estructurados*. PhD thesis, UPV/EHU,.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lee, D., Chuang, H., and Seamons, K. (1997). Document ranking and the vector-space model. *Software, IEEE*, 14(2):67–75.

Lejeune, D. (2012). *The radical use of chance in 20th century art*, volume 366. Rodopi.

Lekuona, J. (1982). *Ahozko euskal literatura*. Erein.

Lekuona, M. (1974). *Bertsolaritza*. Kardaberaz bilduma.

Lemaignan, S., Gharbi, M., Mainprice, J., Herrb, M., and Alami, R. (2012). Roboscopie: A theatre performance for a human and a robot [video abstract]. In *Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 427–427.

Leshin, L., Mahaffy, P., Webster, C., Cabane, M., Coll, P., Conrad, P., Archer, P., Atreya, S., Brunner, A., and Buch, A. (2013). Volatile, isotope, and organic analysis of martian fines with the mars curiosity rover. *Science*, 341:1238937:1–9.

Levy, R. (2001). A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proccedings of the ICCBR-01 Workshop on Creative Systems*. Citeseer.

Li, B. (2016). Robot. http://www.blancali.com/en/event/99/robot.

Libbrecht, M. W. and Noble, W. S. (2015). Machine learning in genetics and genomics. *Nature Reviews. Genetics*, 16(6):321.

Lifchitz, A., Jhean-Larose, S., and Denhière, G. (2009). Effect of tuned parameters on an lsa multiple choice questions answering model. *Behavior research methods*, 41(4):1201–1209.

Lin, C., Tseng, C., Teng, W., Lee, W., Kuo, C., Gu, H., Chung, K., and Fahn, C. (2009). The realization of robot theater: Humanoid robots and theatric performance. In *Proceedings of the IEEE International Conference on Advanced Robotics (ICAR)*, pages 1–6, Munich, Germany.

Lizarazu, I. (2014). Lehen urratsak bertsotarako gaien azterketa automatikoan: gaiaren erregistroaren predikzioa eta gaiaren eta bertsoaren arteko erlazio semantikoaren azterketa. Master's thesis, UPV/EHU.

Lujanbio, M. (2015). Poesia (zubi bat) bertsolaritza. http://www.berria.eus/paperekoa/1944/030/001/2015-07-19/ezin_kabiturik.htm.

Lutz, T. (1960). *Über ein Programm zur Erzeugung stochastisch-logistischer Texte*. na.

Machado, P. and Cardoso, A. (2002). All the truth about NEvAr. *Applied Intelligence*, 16(2):101–118.

Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. (2015). Dopelearning: A computational approach to rap lyrics generation. *arXiv preprint arXiv:1505.04771*.

Manning, C. and Schütze, H. (1999). *Foundations of statistical natural language processing*, volume 999. MIT Press.

Manurung, H. (2003). *An evolutionary algorithm approach to poetry generation*. PhD thesis, School of informatics, University of Edinburgh.

Manurung, R., Ritchie, G., and Thompson, H. (2012). Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):43–64.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. (2011). Big data: The next frontier for innovation, competition, and productivity.

Marz, N. and Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co.

Mataric̀, M. (1997). Behaviour-based control: Examples from navigation, learning, and group behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336.

Matarić, M. (2007). *The robotics primer*. Mit Press.

McGregor, S., Purver, M., and Wiggins, G. (2016). Process based evaluation of computer generated poetry. In *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation*, pages 51–60.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Misztal, J. and Indurkhya, B. (2014). Poetry generation system with an emotional personality. In *Proceedingqs of the 5th International Conference on Computational Creativity*.

Mitchell, T. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

Mitxelena, K. (1988). *Historia de la literatura vasca*. Erein.

Nakano, M., Hasegawa, Y., Funakoshi, K., Takeuchi, J., Torii, T., Nakadai, K., Kanda, N., Komatani, K., Okuno, H., and Tsujino, H. (2011). A multi-expert model for dialogue and behavior control of conversational robots and agents. *Knowledge-Based Systems*, 24(2):248–256.

Nooraei, B. (2012). *A real-time architecture for conversational agents*. PhD thesis, Worcester Polytechnic Institute.

Olasagasti, E. (2004). Eszenaratzea eta dramaturgia egungo bertso agerkarietan. Technical report, Euskal Herriko Bertsozale Elkartea.

Oliveira, H. (2012). Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1.

Oliveira, H. (2015). Automatic generation of poetry inspired by twitter trends. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 13–27. Springer.

Oliveira, H., Hervás, R., Dıaz, A., and Gervás, P. (2014). Adapting a generic platform for poetry generation to produce spanish poems. In *5th International Conference on Computational Creativity, ICCC*.

Osinalde, M. (2013). Agur-bertsoetako egitura diskurtsiboaren xerk. Master's thesis, UPV/EHU.

Oulipo (1988). *Atlas de littérature potentielle*, volume 109. Editions Gallimard.

Paris, C., Swartout, W., and Mann, W. (2013). *Natural language generation in artificial intelligence and computational linguistics*, volume 119. Springer Science & Business Media.

Paya Ruiz, X. (2013). *Towards a taxonomy for Basque oral poetry Bertsolaritza*. PhD thesis, University of Birmingham.

Polanyi, L., Van Den Berg, M., and Ahn, D. (2003). Discourse structure and sentential information structure. an initial proposal. *Journal of Logic, Language and Information*, 12(3):337–350.

Queneau, R. (1961). 100.000. 000.000. 000 de poèmes. *Gallimard Series. Schoenhof's Foreign Books, Incorporated*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, Kobe, Japan.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Raibert, M., Blankespoor, K., Nelson, G., Playter, R., and Team, T. (2008). Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th world congress*, volume 17, pages 10822–10825. Seoul, Korea.

Ramakrishnan, A., Kuppan, S., and Devi, S. (2009). Automatic generation of tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 40–46. Association for Computational Linguistics.

Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

Ramos-Soto, A., Bugarín, A., and Barro, S. (2016). On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets and Systems*, 285:31–51.

Rashel, F. and Manurung, R. (2014). Pemuisi: a constraint satisfaction-based generator of topical indonesian poetry. In *Proceedings of the 5th International Conference on Computational Creativity (ICCC)*, pages 82–90.

Rodríguez, I., Aguado, A., Parra, O., Lazkano, E., and Sierra, B. (2017). Nao robot as rehabilitation assistant in a kinect controlled system. In *Converging Clinical and Engineering Research on Neurorehabilitation II*, pages 419–423. Springer International Publishing.

RSAIT research group (2012a). Bertsobot - galtxagorri eta maialen oinak emanda. https://www.youtube.com/watch?v=OpQBVmkzRWg.

RSAIT research group (2012b). Bertsobot - tartalo eta egaña puntuka. https://www.youtube.com/watch?v=x8w4YuNY-Z0.

RSAIT research group (2013). NAO-bachelor-students. https://www.youtube.com/watch?v=KpappW06oDs.

RSAIT research group (2014). Bertsobot: Emakumeen egunar. https://www.youtube.com/watch?v=7akuD-lnZk4.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

San Vicente, I., Saralegi, X., and Agerri, R. (2015). EliXa: A modular and flexible ABSA platform. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pages 748–752.

Sarasua, J. (2004). Ideia batzuk bapateko kantuaren etorkizun-ikuspegirako. Technical report, Euskal Herriko Bertsozale Elkartea.

Schreiber, G., Stemmer, A., and Bischoff, R. (2010). The fast research interface for the kuka lightweight robot. In *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, pages 15–21. Citeseer.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Sheridan, T. B. (2016). Human–robot interaction status and challenges. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 58(4):525–532.

Sung, G. T. and Gill, I. S. (2001). Robotic laparoscopic surgery: a comparison of the da vinci and zeus systems. *Urology*, 58(6):893–898.

Susperregi, L., Fernandez, I., Fernandez, A., Fernandez, S., Maurtua, I., and Lopez de Vallejo, I. (2012). Interacting with a robot: a guide robot understanding natural language instructions. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 185–192. Springer.

Toivanen, J., Gross, O., and Toivonen, H. (2014). The officer is taller than you, who race yourself! using document specific word associations in poetry generation. In *Proceedings of the 5th International Conference on Computational Creativity*, pages 355–359.

Toivanen, J., Järvisalo, M., and Toivonen, H. (2013). Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity*, pages 160–167.

Toivanen, J., Toivonen, H., Valitutti, A., and Gross, O. (2012). Corpus-based generation of content and form in poetry. In *Proceedings of the Third International Conference on Computational Creativity*, pages 175–179.

Tomašic, P., Znidaršic, M., and Papa, G. (2014). Implementation of a slogan generator. In *Proceedings of 5th International Conference on Computational Creativity, Ljubljana, Slovenia*, volume 301, pages 340–343.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.

Turney, P. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Valitutti, A. (2009). *Computational Production of Affect-Based Verbal Humorous Expressions*. PhD thesis, University of Trento.

Veale, T. (2013). Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the International Conference on Computational Creativity*, pages 152–159.

Veale, T. and Hao, Y. (2008). A fluid knowledge representation for understanding and generating creative metaphors. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 945–952. Association for Computational Linguistics.

Wada, K., Ikeda, Y., Inoue, K., and Uehara, R. (2010). Development and preliminary evaluation of a caregiver's manual for robot therapy using the therapeutic seal robot paro. In *Proceedings of the 19th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 533–538.

Westerink, J., Ouwerkerk, M., de Vries, G., de Waele, S., van den Eerenbeemd, J., and van Boven, M. (2009). Emotion measurement platform for daily life situations. In *Proceedings of 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII)*, pages 1–6. IEEE.

White, M. and Caldwell, T. (1998). A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation (INLG)*, pages 238–247.

White, T. (2012). *Hadoop: The definitive guide*. O'Reilly Media, Inc.

Wiggins, G., Pearce, M., and Müllensiefen, D. (2009). Computational modelling of music cognition and musical creativity. pages 383–420.

Witten, I., Frank, E., Hall, M., and Pal, C. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wittgenstein, L. (1953). *Philosophical investigations. Philosophische Untersuchungen*. Blackwell.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52.

Wong, M., Chun, A., Li, Q., Chen, S., and Xu, A. (2008). Automatic haiku generation using vsm. In *WSEAS International Conference. Mathematics and Computers in Science and Engineering*, volume 7, pages 318–323, Hangzhou, China. World Scientific and Engineering Academy and Society.

Wu, D. and Addanki, K. (2015). Learning to rap battle with bilingual recursive neural networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2524–2530.

Xingjian, S., Chen, Z., Wang, H., Yeung, D., Wong, W., and Woo, W. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810.

Yamauchi, B. (2004). Packbot: A versatile platform for military robotics. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics.

Yan, R., Jiang, H., Lapata, M., Lin, S., Lv, X., and Li, X. (2013). i, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2197–2203.

Yang, J., Singh, H., Hines, E., Schlaghecken, F., Iliescu, D., Leeson, M., and Stocks, N. (2012). Channel selection and classification of electroencephalogram signals: an artificial neural network and genetic algorithm-based approach. *Artificial intelligence in medicine*, 55(2):117–126.

Zelaia, A. (2016). *Latent semantic indexing eta ikasketa automatikoa hizkuntzaren prozesamenduaren arloan: testu-sailkatzea, hitzen adiera-desanbiguatzea eta korreferentzia-ebaztea SVD bidezko dimentsio murrizketa eta multi-sailkatzailea konbinatuz.* PhD thesis, Universidad del País Vasco/Euskal Herriko Unibertsitatea.

Zelaia, A., Baragaña, I., and Yurramendi, Y. (2011). Lsaren oinarri matematikoa. *EKAIA Euskal Herriko Unibertsitateko Zientzi eta Teknologi Aldizkaria*, (17).

Zipitria, I., Arruarte, A., and Elorriaga, J. (2006). Observing lemmatization effect in lsa coherence and comprehension grading of learner summaries. In *International Conference on Intelligent Tutoring Systems*, pages 595–603. Springer.

Zipitria, I., Sierra, B., Arruarte, A., and Elorriaga, J. (2012). Cohesion grading decisions in a summary evaluation environment: A machine learning approach. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society (CogSci)*, pages 2615–2620.