eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Konputazio Zientzia eta Adimen Artifiziala Saila

Informatika Fakultatea

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

Ana Zelaia Jauregi

Zuzendariak:
Olatz Arregi Uriarte
Basilio Sierra Araujo

Donostia, 2015

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Konputazio Zientzia eta Adimen Artifiziala Saila

Informatika Fakultatea

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

Ana Zelaia Jauregik Olatz Arregi Uriarteren eta Basilio Sierra Araujoren zuzendaritzapean egindako ikerketa-txostena, Euskal Herriko Unibertsitatean Informatikan Doktore titulua eskuratzeko aurkeztua

Donostia, 2015eko abendua

Aita, ama,
hau zuentzat da,
bihotz bihotzez.

Eskerrak

Luxua izan da niretzat Olatz eta Basi, zuen zuzendaritzapean lan egitea; pribilegio handia, nire akatsak zuzendu dituzuelako, esperimentuetan sortutako trabetatik ateratzen lagundu didazuelako, zuzendu nauzuelako, gidatu nauzuelako.... Baina, batez ere, egindako lana nirekin konpartitu duzuelako, edozein txikikeria komentatzeko prestutasuna erakutsi didazuelako, nire zain egoten jakin duzuelako, nire geldialdiak onartu dituzuelako. Eskerrik asko, Olatz! Eskerrik asko, Basi! Bene-benetan, eskerrik asko!

Eskerrik asko Iñaki, Olatzengana eta Basirengana gerturatzean noranzko egokiari begira kokatu ninduzulako. Eskerrik asko zuri ere, Arantza, eta IXA talde osoari, asko jaso dut eta zuengandik, zuzenean edo zeharka: esperimentuetarako behar izan ditudan corpusak, matrize erraldoien prozesamendua egin ahal izateko txoko bat IXA makinetan... Lasai, Kike, laster utziko dut espazio librea diskoan... Bereziki aipatu nahi ditut Izaskun eta Idoia, euskarazko dokumentuen Testu-Sailkatzea egiteko *Euskaldunon egunkaria*ko artikuluak antolatzen laguntzeagatik, eta Eneko eta Oier, Hitzen Adiera-Desanbiguatze atazan ezaugarri linguistikoekin laguntzeagatik. Txapelketan irabazi zeniguten... Zuek bai artistak!

Itziar, gure aljebrari finena; LSIren oinarri den SVD deskonposaketa ulertzeko zugandik jaso dudan laguntzak konfiantza eman dit. Eskerrik asko!

Atzerantz begiratuz gero, hona iristeko egindako bidean ikerketa-lagun izan ditudanak ikusten ditut. Ana, Jon Ander eta Iraide, zuek zaudete gertuen. Zuekin topo egin arte LSI zer zen ez nekien. Garai hartan hasi ginen lehen probak egiten, eta aplikaziorako arlo egokiak bilatzen. 2002a zen... Pixka bat atzeraxeago begiratuz gero, horra Yosu eta Patxi. Loto jokotik abiatu ginen, eta konbinatoriaren mundu horretan hainbat estalketa-eredu aztertu genituen. Burua zenbakiz beteta genuen... Taula dotoreetan txukundu genituen... 1998a zen... Pixka bat harantzaxeago Elena, Olatz eta Tim ikusten ditut, eta haiekin batera robotak. Fakultatean lanean hasi berria nintzen garai hartan... Bekaria nintzen... 1994a zen... Ah ze garaiak haiek... Gazteak ginen oso... Eskerrik asko, guztioi!

Lan-eguna nirekin konpartitzen duzuen lankideak, goizetako kafe-txokoko lagunak, bazkalorduko solaskideak. Une horietan lanean nagoela ere ahaztu egiten zait. Bulego-zulo horretatik ateratzen nauzue, eta gustura jarduten dugu kontu-kontari: Itziar, Joseba, beste Itziar, Basi, Elena, Ana, beste Elena, Olatz, Iñaki, Patxi, beste Iñaki, Josune, Ander, Yosu, Montse, Aitor...

Lan-girotik aparte, nirekin dagoen jendea datorkit gogora, gertuenekoak bereziki: aita, ama, Belen, Manu, Dabid, Kristina, Ekain, Julen, eta nola ez nire mutilak: Juan Luis, Asier eta Barane. Eskerrik asko ematen didazuen babesagatik. Zutabe sendo bat zarete niretzat!!

Asier, Barane,
Besarkada handi-handi bat zuentzat!
Eta beste bat! Eta beste bat!
Eta beste bat! Eta beste pila-pila bat!
Baina, ze handiak zareten...
Aupa zuek!

Juan Luis, eman eskua eta goazen!
Oraintxe haize freskoa hartzeko beharra ere badut eta...
Goazen!
Elkarrekin abiatu genuen bide honetan gustura noa eta...
Eman eskua!
Zurekin noanean aldapak leunagoak egiten zaizkit eta...
Eman eskua, Juan Luis, eta goazen!

Gaien aurkibidea

SARRERA

I. KAPITULUA

Sarrera

I.1 Motibazioa

Gizakiok modu naturalean komunikatzeko hizkuntza erabiltzen dugu. Konputagailuen eta teknologia berrien garapenari esker, asko handitu dira idatzizko testu-dokumentuak gordetzeko ahalmena eta aukerak. Interneten, esaterako, Wikipedian, blogetan, sare sozialetan eta, oro har, webguneetan makina bat testu dago edonoren eskura. Testuak automatikoki prozesatzea ezinbestekoa gertatzen da bertan dagoen informazio kudeatu eta erabilgarri bihurtu nahi bada.

Horrela sortu zen **Hizkuntzaren Prozesamendua** (NLP, Natural Language Processing) (Jurafsky and Martin, 2009) eta prozesamendurako tresnak garatzeko beharra. Aplikazio-eremuak ugariak dira, batzuk aipatzearren: Informazio-Erauztea (Information Extraction), Informazio-Berreskuratzea (Information Retrieval), Itzulpengintza Automatikoa (Machine Translation), Galdera-Erantzun sistemak (Question Answering) edota Laburpengintza Automatikoa (Automatic Summarization). Ataza horiek oso konplexuak dira, testuen semantika jaso eta diskurtsoaren ulermen sakona eskatzen dutelako. Dokumentuetatik informazio garrantzitsua erauzi nahi bada, ideia horiekin laburpen bat sortzeko, erabiltzaileak egindako galdera bati erantzuteko edota testu zati bat beste hizkuntza batera itzultzeko, makinak hitzen esanahia ulertu beharko du, eta hitzen arteko erlazioak antzeman, esaldiak ulertzeko eta azken batean diskurtso osoaren semantika jasotzeko. Konputagailuak trebeak dira kalkuluak egiten, baina hizkuntzaren ulermena eskatzen duten ataza horiek egiteko gai izan daitezen, kalkuluak egitetik haratago doan adimenaz janztea derrigorrezkoa gertatzen da; horra Adimen

Artifizialaren muinean dagoen erronka!

Maila baxuagoko analisi linguistikotik gertuago dauden beste ataza batzuk ere definitu dira eta baita aurrerapauso handiak lortu ere: Kategoria Gramatikalen Etiketatzea (Part-of-Speech tagging), Lematizazioa, Entitateen Izenen Ezagutzea (Named Entity Recognition), Analizatzaile Sintaktikoak garatzea (Syntax Parsing), Hitzen Adiera-Desanbiguatzea (Word Sense Disambiguation), Korreferentzia-Ebaztea (Coreference Resolution) edo Rol Semantikoak Etiketatzea (Semantic Role Labeling), besteak beste. Hitzen esanahiak antolatzeko ontologiak sortu izan dira, gizakiak hitzen esanahiari eta haien arteko erlazioari buruz duen ezagutza adierazteko, WordNet adibidez. Azken batean, ataza horiek guztiek batera, eta aipatu gabe geratu diren beste askok osatzen dute Hizkuntzaren Prozesamendu Automatikoa. Horietako bakoitzaren ebazpenean emandako urratsek eta lorpenek eragina izango dute ataza nagusietan.

Euskal Herriko Unibertsitateko Informatika Fakultateko IXA taldeak[1] hogeita bost urte baino gehiago daramatza hizkuntzaren tratamendu automatikoa egiten, eta euskarari arreta berezia eskaintzen. Denbora tarte horretan hizkuntzalarien eta informatikarien elkarlanari esker, euskararako sortutako baliabideak eta tresnak ugariak izan dira. Hala nola, Euskararen Datu-Base Lexikala (EDBL), morfeus analizatzaile morfologikoa, hainbat analizatzaile sintaktiko, corpusak, hiztegi elektronikoak, sare semantikoa, etab.

Bestalde, Informazio-Berreskuratze atazari dagokionez, hasiera batean garatu ziren teknikek dokumentuak indexatzeko erabiltzen zuten estrategia dokumentuetan agertzen ziren terminoetan oinarritzen zen. Erabiltzaileak egindako bilaketan erabilitako terminoen arabera, termino horiek zehazki zituzten dokumentuak eta ez besterik itzultzen ziren bilaketaren emaitza moduan. Sinonimia eta polisemia bezalako fenomenoen eraginez, bilatzaileak batzuetan erabiltzailearen interesekoak ez ziren dokumentuak berreskuratzen zituen; beste batzuetan, aldiz, ez zuen lortzen haren interesekoak ziren dokumentuak berreskuratzea.

Arazo horri aurre egiteko helburuarekin sortu zen **Latent Semantic Indexing** (LSI) (Landauer et al., 2013), dokumentuak haien ezkutuko semantikaren arabera indexatuz gero, berreskuratze semantikoa egokiagoa egiteko. Testuen semantika modu automatikoan jasotzeko teknika aproposa zela ikusi zen, hizkuntzaren tratamenduan hiztegiaren aldakortasunak sortutako arazoetarako bereziki trebea zela. LSIk, testuetako hitzen maiztasunen eta agerkidetzen analisian oinarrituz, testuen arteko erlazio semantikoak hobe-

[1] IXA taldea, http://ixa.eus/Ixa

to erakusten dituen adierazpen matematikoak kalkulatzen ditu modu automatikoan. Halako adierazpenak kalkula daitezke, baita, hitzetarako, esaldietarako, paragrafoetarako, etab. Hori dela eta, LSI erlazio semantikoak induzitzeko gaitasuna duen eredu konputagarri bat dela esaten da.

Maila teorikoago batean, kognizioa aztertzen ziharduten zientzialariek hitzen esanahia, hizkuntza eta ezagutza eskuratzea azaltzen duen teoria bat ere badela baieztatu zuten; gizakiok gure bizitzan zehar hizkuntzarekin dugun esperientziatik, hitzen esanahia eta ezagutza nola eskuratzen ditugun azaltzeko baliagarria den teoria konputagarri bat. Teoria horren arabera, hitzek bere horretan ez dute esanahirik. Esanahia lortzen dute beste hitzekiko dituzten erlazioetatik. Konputagailu batean, testuetako hitzen agerpenak aztertuz eta gizakiak sortutako inolako erregela linguistikorik eman gabe, gizakiok hizkuntzarekin dugun esperientzia hori simula daiteke, hitzen, esaldien eta testu zati handiagoen esanahia modu automatikoan induzitzeko.

Ezin esan daiteke LSI hizkuntzaren eta esanahiaren adierazpenerako teoria osatu bat denik, noski, ez baitu hitzen ordenarekin eta esaldia osatzeko moduarekin zerikusi zuzena duten fenomenoak kudeatzeko gaitasunik, ez behintzat tradizionalki aplikatu izan den moduan erabiliz gero. Anaforaren fenomeno linguistikoa, adibidez, testuetako hitzen arteko erlazioa aurkitzean datza, testuetako hitzen agerpena aztertze hutsarekin jaso ezin den erlazioa. Anafora-Ebaztearen atazak eta hark bezala esanahiaren aldaketa ekar dezaketen beste hainbat fenomeno linguistikok trataera berezia eskatzen dute.

LSI hitzen esanahia, edota hizkuntza eta ezagutza adierazteko baliagarria dela baieztatzeko, oso esanguratsuak diren bi esperimentuetan lortutako emaitza onak aipatzen dira. Horietako bat TOEFL (Test of English as a Foreign Language) test estandarrarekin egindakoa da: hitz bat emanik, proposatzen diren beste lau hitzen artean semantikoki antzekoena zein den esan behar da. Gizakiek lortutako emaitzen parekoak lortu zituen LSIk, eta baieztatu zen testuetako hitzei emandako trataera egokia dela, eta hizkuntzaren eta esanahiaren ulermenerako eredu konputagarri egokia izan daitekeela. Oihartzun handia izan zuen beste esperimentu bat hezkuntzaren arlorako garatutako "Summary Street" aplikazioa da, ikasleak laburpengintzan trebatzeko diseinatutakoa: testu bat emanik, ikasleak egindako laburpena modu automatikoan ebaluatzen du tresnak, laburpena artikulu originalarekin konparatuz. Laburpenen ebaluazioan erakutsitako trebezia, oraindik orain, LSIren arrakastaren froga praktiko moduan aipatzen da.

LSIren aplikagarritasuna beste arlo batzuetan egiaztatzeko asmoz, asko izan dira harekin egin diren esperimentuak, hizkuntzaren semantikarekin harreman estuan dauden aplikazio-eremuetan bereziki. Esanguratsuenak hemen aipatzeak merezi du: Testuen Clustering-a, Diskurtsoaren Koherentzia

neurtzea (segidako esaldi eta paragrafoen antzekotasun semantikoa neurtuz), Diskurtsoaren Segmentazioa (paragrafoen arteko antzekotasun txikiak diskurtsoa beste gai baterantz bideratu dela adieraziko du), Laburpengintza Automatikoa (diskurtsoaren segmentaziotik egitura aurkituz eta esaldi esanguratsuenak aukeratuz), Hizkuntzarteko Informazio-Berreskuratzea (Cross-Language Information Retrieval, corpus paraleloekin lan eginez, hizkuntza batean egindako bilaketak erabiltzailearen interesekoak diren beste hizkuntzako dokumentuak berreskuratzeko aukera emanez), edota bi hizkuntzetako hitzen eta dokumentuen arteko antzekotasun semantikoak neurtzeko.

Ikerketa-lan honen muinean dagoen hirugarren zutabea **Ikasketa Automatikoa** (ML, Machine Learning) da (Mitchell, 1997), hau da, emandako datu-multzoetatik konputagailuek ikasteko gaitasuna garatzea helburu duen informatikaren alorra. 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.

Helburu hori lortzeko asmoz eredu asko eta oso izaera desberdinekoak proposatu izan dira, eta portaera ona erakutsi dute aplikazio-eremu askotan: medikuntzan pazienteen diagnostikoak egiteko, eguraldiaren iragarpenak egiteko, iruzurra antzemateko, irudiak sailkatzeko, bioinformatikaren arloan simulazioak egiteko... Problema horiek guztiak, Hizkuntzaren Prozesamendua bezala, Adimen Artifizialaren muinean dauden arazoak dira.

Ikerketa-lan hau abiatzeko motibazio nagusia Latent Semantic Indexing (LSI) eta Ikasketa Automatikoa uztartzeak Hizkuntzaren Prozesamenduaren hainbat atazaren ebazpenean ekar dezaken onura aztertzea izan da.

I.2 Helburuak

LSIk hitzen eta testuen semantika jasotzeko erakutsitako trebezia ikusita, hark kalkulatutako adierazpen matematikoa Ikasketa Automatikoko metodoekin uztartuz, Hizkuntzaren Prozesamenduaren hainbat atazaren ebazpenerako prozedura bat diseinatu dugu, haren portaera eta ekar dezakeen onura aztertzeko.

Hiru dira planteatu diren helburu nagusiak:

- LSI eta Ikasketa Automatikoa konbinatuz, Hizkuntzaren Prozesamendua arlorako metodologia egoki bat proposatzea. Ikasketa Automatikoa aplikatzeko orduan, sailkatzaile sinpleetatik haratago joatea dugu helburu, multi-sailkatzaileen onurak aprobetxatu nahian.

- Metodologia hori, izaera desberdineko hizkuntza oinarri duten corpusetan frogatzea. Aukeratutako hizkuntzak ingelesa eta euskara dira. Lehenengoak gure proposamenaren emaitzak konparatzeko balioko digu, bigarrenak berriz, euskarak, eranskaria izanik, ezaugarri morfologikoak modu egokian metodologian uztartzearen erronka planteatzen du.

- Hizkuntzaz gain, izaera oso ezberdineko Hizkuntzaren Prozesamenduko hiru atazetan probatu nahi izan da metodologia hori.

  Honako hauek dira aukeratutako atazak:

  - Testu-Sailkatzea. LSIren aplikazio-eremu tradizional bat da. Oso izaera desberdineko testuen sailkatzearekin probak egitea eta bi hizkuntzekin esperimentatzea erabaki dugu: euskarazko testuekin eta ingelesezkoekin. Esperimentu bakoitzean, testu-dokumentu guztiak batera hartuz osatu da LSIrako corpusa, informazio orokorra biltzen duen corpusarekin esperimentatzeko asmoz.

  - Hitzen Adiera-Desanbiguatzea. Polisemikoak diren hitz-adierak desanbiguatzeko, hitzen agerpen testuinguruaz baliatzea erabaki dugu. Nazioarte mailako txapelketa batean parte hartzea aukera aparta izan da metodoaren eraginkortasuna neurtzeko. Ataza honetarako ezagutza espezializatua jasoko duten corpus txiki askorekin esperimentatzea erabaki dugu.

  - Korreferentzia-Ebaztea. Entitate berari erreferentzia egiten dioten diskurtsoko agerpenak haien artean erlazionatzea ataza zaila, baina aldi berean garrantzitsua da testuaren ulermena eskatzen duten atazatan. Informazio lexikoa, morfologikoa eta sintaktikoaz gain, beharrezkoa gertatzen da informazio semantikoa eta pragmatikoa erabiltzea korreferentzia ebazteko. Ataza honen ebazpenerako LSI aplikatzea ez da berehalakoa gertatzen. Oraingoan ere euskarazko eta ingelesezko corpusekin lan egitea erabaki dugu.

Egindako lanaren ekarpen nagusiak hiru dira. (1) Erabilitako metodologia, LSI+Ikasketa Automatikoa, (2) LSIren aplikazio-eremu ez hain berehalakoetan haren portaera aztertzea, etorkizunean metodologiak eman lezakeena aztertuz bide berriak urratzeko, (3) metodologia hori, euskarazko corpusak baliatuta, Hizkuntzaren Prozesamenduko hiru atazetan frogatzea.

I.3 Ikerketa-lanaren txostenaren antolaketa

Lanaren txostenak egindako ikerlanaren fruitu diren argitalpen garrantzitsuenak biltzen ditu. Haietan deskribatzen dira egindako lanaren xehetasunak eta lortutako emaitzak. Nazioarte mailako biltzar eta aldizkarietan argitaratuak izan direnez, komunitate zientifikoaren esku daude. Egindako lanetik lortutako emaitzetatik, onetatik eta ez hain onetatik, ikasteko itxaropenez abiatu dugu lan hau; Latent Semantic Indexing eta Ikasketa Automatikoa Hizkuntzalaritza Konputazionalaren mesedegarri izateko problemak nola planteatu behar diren ikasi nahi izan dugulako eta etorkizuneko ikerlanerako bide berriak zabaldu. Gure ekarpen-aletxo hori ildo bertsutik lanean diharduten zientzialarientzat lagungarri bada, zorionekoak gu!

Argitalpenak ingelesez daude, baina ikerketa-txostena euskaraz idatzita dago, euskal komunitate zientifikoari begira idatzia izan delako. Euskaraz egin dugu lan, eta euskararentzat, neurri handi batean. Hizkuntzalaritza Konputazionalaren hiru aplikazio-eremu orokorren ebazpena aitzakia hartuta, orain arte probatu gabeko metodologia berri bat aplikatu dugu, ingelesezko testuekin batzuetan, euskarazkoekin besteetan. Etengabe biak eskutik helduta eraman nahi izan ditugu, izaera desberdineko hizkuntzak izanik ere, handienarekin lortutako aurrerapenak txikienaren mesederako ere izango direlakoan.

Txostena lau 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 eta txostenaren egitura azaltzen dira.

Bigarren atalak OINARRI TEORIKOAK izenburupean bi kapitulu biltzen ditu. II. kapituluan "LSIren oinarri matematikoa" azaltzen da. Testuetako hitzen agerpenen maiztasunak eta agerkidetzak aztertuz, LSIk hitzen eta testu-zatien arteko erlazio semantikoak nola induzitzen dituen ulertu nahi bada, ezinbestekoa gertatzen da haren oinarri teorikoetan pixka bat sakontzea. Kapitulua irakurterraza da Aljebra Linealari buruzko ezagutza minimoa duen irakurlearentzat: bektore-espazioen oinarrizko definizioak biltzen dira, semantikarako izan duten erabilera laburtzen da eta LSIren muinean dauden SVD matrize-deskonposaketa eta dimentsioaren murrizketa azaltzen dira. Azalpen teorikoak modu xumean emanak datoz, eta erreferentzia bibliografiko asko tartekatu dira testuan zehar, informazio sakonagoa nahi duenarentzat. Gainera, ahalegin berezia egin da azalpen teorikoen interpretazio geometrikoak han-hemenka tartekatzeko. Kapitulua luzea da, baina LSI teknika hain ezaguna ez izanik eta haren erabilerak Hizkuntzaren

Prozesamenduan eman dituen fruituak ikusita eta oraindik ere eman ditzakeenak sinetsita, oinarri teorikoetan pixka bat sakontzen duen kapitulua txosten honi eranstea erabaki da. Oraindik orain, teknikaren inplementazio berriak sortzen ari dira, eta foroetan sartu besterik ez dago LSIk bizi-bizi jarraitzen duela eta ikerlariek pil-pilean dauden aplikazio-eremu berrietarako esperimentatzen jarraitzen dutela egiaztatzeko. Zer gutxiago haren oinarri teorikoetara pixka bat gerturatzea baino...

III. kapituluan, "Ikasketa automatikoa. Gainbegiratutako sailkatzea" azaltzen da. Konputagailuek modu egoki eta automatikoan ikastea helburu duen Adimen Artifizialaren adar bat da Ikasketa automatikoa. Oso oinarri teoriko desberdina duten hainbat metodok osatzen dute eta gaur egun oso erabiliak izaten ari dira, batez ere datu-multzo erraldoietatik ezagutza ulergarria eta erabilgarria lortu nahi den esparruetan. Metodo horiei buruz bibliografia zabala existitzen da. Ikerketa-lan honetan erabili diren metodoen aipamen xumea ematen da kapituluan, ebazpen-metodo orokor moduan diseinatu eta inplementatu dugunaren deskribapenean ahalegin berezia eginez.

Hirugarren atala APLIKAZIO EREMUAK biltzen dituena da, eta hiru kapitulutan antolatuta dago. IV. Kapituluan "Testu-Sailkatzea", V. Kapituluan "Hitzen Adiera-Desanbiguatzea" eta VI. Kapituluan "Korreferentzia--Ebaztea". Hiruek egitura bera dute: hasteko ataza deskribatzen da, ondoren problemaren ebazpena nola planteatu den azaltzen da eta atazarako ateratako ondorioak jasotzen dira. 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 VII. kapituluak biltzen ditu ondorio nagusienak eta ikerkuntzarako zabalik geratu diren zenbait bide.

OINARRI TEORIKOAK

II. KAPITULUA

LSIren oinarri matematikoa

LSIren (Latent Semantic Indexing) oinarri matematikoa aztertzerakoan, ezinbestekoa gertatzen da aljebra linealari buruz hitz egitea. Izan ere, LSIk corpus bateko dokumentuen esanahia dokumentuko hitzen esanahian oinarrituz kalkulatzen du. Hitzak eta dokumentuak bektoreen bidez adieraziak izango dira, eta bektore horien arteko eragiketetatik lortuko dira haien arteko antzekotasun semantikorako neurriak.

Kapitulu honetako atalak honela banatuak izan dira. Hasteko, II.1. atalean bektore-espazioen oinarrizko definizioak daude. II.2. atalean bektore-espazioen erabilera aztertuko da semantikarako. II.3. atalean matrizeen deskonposaketa eta dimentsioaren murrizketa azaltzen dira. Amaitzeko, II.4. atalean zenbait ondorio aipatuko ditugu.

Kapitulu honen helburua ez da aljebra lineala sakonki aztertzea, baina bai funtsezkoak gertatzen diren kontzeptuak modu laburrean gogoratzea. Gehiago sakondu nahi duenari honako erreferentzietara jotzea gomendatzen diogu. Aljebra linealari buruzko oinarrizko kontzeptuetarako eta SVD deskonposaketan (Balio Singularretan Deskonposatzea) sakontzeko, ikus (Strang, 2009) eta (C.D.Meyer, 2000), interpretazio geometrikoetarako ikus (Widdows, 2004), eta LSIren oinarri matematikoan sakontzeko ikus (Berry and Browne, 2005), (Berry et al., 1995), (Berry et al., 1999), (Landauer et al., 1998), (Landauer and Dumais, 1997), (Deerwester et al., 1990).

II.1 Aljebra lineala. Bektore-espazioak

II.1.1 Bektore-espazioa. Definizioak

Bektore-espazioa egitura matematiko bat da. Bektore izeneko elementuez eta eskalarrez osatuta dago, eta bi eragiketa definituta daude: bektore arteko batura eta bektore baten eta eskalar baten arteko biderkadura. Eskalarrak, oro har, balio errealak izan ohi dira. Bi eragiketa horiek axioma izeneko zenbait baldintzak bete behar dituzte. Notazioari dagokionez, bektoreak eta matrizeak letra lodiz adieraziak izango dira eskalarretatik bereizteko.

Izan bedi $(\mathbb{R}, +, \cdot)$ gorputza. V bektore multzoa $\mathbb{R}$-ren gaineko **bektore-espazioa** dela esango dugu baldin ondorengo bi eragiketak definituta baditu.

- Bektoreen arteko **batuketa** $(+)$ barne eragiketa. m osagaiko $\mathbf{u}$ eta $\mathbf{v}$ bi bektoreren arteko batura $\mathbf{u}+\mathbf{v}$ moduan adierazten da. Bektoreetako osagaiak batuz lortuko den beste bektoreak ere m osagai izango ditu.

$$\forall \mathbf{u}, \mathbf{v} \in V \quad \Rightarrow \quad \mathbf{u} + \mathbf{v} \in V.$$

- Eskalar batekin **biderketa** $(\cdot_{\mathbb{R}})$ kanpo eragiketa. α eskalarraren eta $\mathbf{u}$ bektorearen arteko biderketaren emaitza $\alpha\mathbf{u}$ bektorea da, $\mathbf{u}$ bektorearen osagai guztiak α eskalarraz biderkatuz lortuko dena.

$$\forall \alpha \in \mathbb{R}\,,\ \forall \mathbf{u} \in V \quad \Rightarrow \quad \alpha\mathbf{u} \in V.$$

V multzoko $\mathbf{u}, \mathbf{v}$ eta $\mathbf{w}$ hiru bektore eta $\alpha, \beta \in \mathbb{R}$ bi eskalar izanik, bektore-espazio bat izateko ondoren aipatzen diren zortzi axiomak bete behar dira. Bektoreen arteko batuketa $(+)$ eragiketari dagokionez,

1. Elkarkorra da. $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V, \quad \mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.
2. Trukakorra da. $\forall \mathbf{u}, \mathbf{v} \in V, \quad \mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.
3. Elementu neutroa du (zero bektorea). $\forall \mathbf{v} \in V,\ \exists \mathbf{0} \in V$ non $\mathbf{v} + \mathbf{0} = \mathbf{v}$.
4. Aurkako elementua du. $\forall \mathbf{v} \in V,\ \exists -\mathbf{v} \in V$ non $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$.

Bektore eta eskalar arteko biderketa $(\cdot_{\mathbb{R}})$ eragiketari dagokionez,

1. Elkarkorra da. $\forall \alpha, \beta \in \mathbb{R}\,, \forall \mathbf{v} \in V,\ \alpha(\beta\mathbf{v}) = (\alpha\beta)\mathbf{v}$.
2. Elementu neutroa du $(1 \in \mathbb{R}\,)$. $\forall \mathbf{v} \in V,\ 1\mathbf{v} = \mathbf{v}$.

3. Banakorra da bektoreen batuketarekiko. $\forall \alpha \in \mathbb{R}\,,\ \forall \mathbf{u}, \mathbf{v} \in V \quad \alpha(\mathbf{u} + \mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$.

4. Banakorra da eskalarren batuketarekiko. $\forall \alpha, \beta \in \mathbb{R}\,,\ \forall \mathbf{v} \in V \quad (\alpha + \beta)\mathbf{v} = \alpha\mathbf{v} + \beta\mathbf{v}$.

Bektore-espazio bat izanik, oinarri bat zer den ulertzea eta bektore-espazioaren dimentsioa definitzea garrantzitsua gertatzen da. Horretarako, honakoak gogoratzea komeni da:

- $\mathbb{R}^m$ bektore-espazioko $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_n$ bektoreak eta $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ eskalarrak izanik, $\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n$ moduko adierazpenari bektoreen arteko **konbinazio lineala** esaten zaio.

- $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^m$ bektoreak **linealki independenteak** dira baldin $\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0}$ konbinazio lineala bete dadin, aukera bakarra eskalar guztiak zero izatea bada, $\alpha_1 = \cdots = \alpha_n = 0$.

- $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^m$ bektoreak **linealki mendekoak** direla esaten da baldin $\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0}$ konbinazio lineala aurki badaiteke, gutxienez α_i eskalar bat zeroren desberdina izanik.

- $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_p\} \subseteq \mathbb{R}^m$ bektore multzoa $\mathbb{R}^m$ espazioaren **multzo sortzailea** dela esaten da baldin $\mathbf{v} \in \mathbb{R}^m$ bektore oro S multzoko bektoreen konbinazio lineal moduan idatz badaiteke, hau da, $\alpha_1, \ldots, \alpha_p \in \mathbb{R}$ eskalarrak existitzen badira, non $\mathbf{v} = \alpha_1\mathbf{v}_1 + \cdots + \alpha_p\mathbf{v}_p$ beteko den.

Horrela, $B = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \subseteq \mathbb{R}^m$ bektore multzoa $\mathbb{R}^m$ bektore-espazioan **oinarria** dela esaten da baldin:

- B multzoko bektoreak linealki independenteak badira, eta

- B multzoa $\mathbb{R}^m$ espazioaren multzo sortzailea bada.

Bektore-espazio bateko oinarri guztiek bektore kopuru bera dute. Kopuru horri bektore-espazioaren **dimentsioa** esaten zaio.

Bektore-espazio baten adibide moduan koordenatuen espazioa aipatzeak merezi du. m zenbaki oso positiboa izanik, $\mathbb{R}$-ko elementuekin osa daitezkeen m-kote guztiek osatzen duten espazioa $\mathbb{R}$-ren gaineko m dimentsioko bektore-espazio bat da. Koordenatu-espazioa esaten zaio, eta $\mathbb{R}^m$ notazioaz

adierazten da. $\mathbb{R}^m$ bektore-espazioko $\mathbf{v}$ bektore bat honela adierazten da:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix},$$

v_i balioak eskalarrak izanik, $i = 1, \ldots, m$. $\mathbf{v}$ bektorea errenkada moduan idazteko $\mathbf{v}^T$ notazioa erabiltzen da.

Bektore-espazio bat izanik, bi bektoreren arteko biderkadura eskalarra definituko dugu segidan. II.1.3. atalean ikusiko dugun bezala, bi bektoreren arteko biderkadura eskalarraren kontzeptua oso erlazionatuta dago distantzia Euklidearrarekin eta kosinu-antzekotasun neurriarekin.

II.1.1 Definizioa. (Biderkadura eskalarra) $\mathbb{R}^m$ bektore-espazioko $\mathbf{u}$ eta $\mathbf{v}$ bi bektoreren arteko biderkadura eskalarra $\mathbf{u}^T \cdot \mathbf{v}$ moduan adierazten da, emaitza eskalar bat da eta honela kalkulatzen da:

$$\mathbf{u}^T \cdot \mathbf{v} = \sum_{i=1}^{m} u_i \cdot v_i.$$

II.1.2 Bektore-espazioaren interpretazio geometrikoa

Bektore-espazioen interpretazio geometrikoa egitean, linealtasunaren kontzeptua agertzen zaigu. Izan ere, bektoreak puntu jakin batetik abiatzen diren gezi zuzenen bidez adierazten dira. $\mathbb{R}^2$ bektore-espazioko bektore baten adierazpen geometrikoa egiteko, koordenatu ardatzak finkatuko ditugu. Koordenatu sistema cartesiarra erabili ohi da.

II.1.1 Adibidea. Izan bitez $\mathbf{u}$ eta $\mathbf{v}$ bi bektore. 19. orriko II.1. Irudian bektoreen interpretazio geometrikoa ikus daiteke $\mathbb{R}^2$ bektore-espazioan.

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \end{pmatrix}.$$

Interpretazio geometrikoari dagokionez, $\mathbf{u}$ eta $\mathbf{v}$ bektoreen arteko biderkadura eskalarra $\mathbf{u}^T \cdot \mathbf{v} = 0$ bada, haien arteko angelua zuzena da eta bektoreak **ortogonalak** (elkarzutak, perpendikularrak) direla esaten da.

II.1.3 Bektoreen arteko distantzia eta antzekotasuna

$\mathbb{R}^m$ espazioko bektoreen arteko distantzia eta antzekotasuna neurtzeko metodo batzuk aipatuko ditugu atal honetan. Kontuan izan behar da, bi bektoreren arteko distantzia txikia bada, haien arteko antzekotasuna handia izango dela.

$\mathbb{R}^2$ koordenatu cartesiarrean adierazitako $\mathbf{u}^T = (u_1, u_2)$ eta $\mathbf{v}^T = (v_1, v_2)$ bi punturen arteko distantzia Euklidearrak bi puntu horien arteko zuzenaren luzera neurtzen du (Pitagoras-en teorema).

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}.$$

Definizio hori orokor daiteke $\mathbb{R}^m$ bektore-espazioko bi bektoreren arteko distantzia kalkulatzeko.

II.1.2 Definizioa. (Distantzia Euklidearra) $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ bi bektoreren (punturen) arteko distantzia Euklidearra honela kalkulatzen da:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}, \quad d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^{m} (u_i - v_i)^2}.$$

II.1.2 Adibidea. Aurreko adibidearekin jarraituz, $\mathbf{u}$ eta $\mathbf{v}$ puntuen arteko distantzia Euklidearra:

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{(4-6)^2 + (3-8)^2} = \sqrt{4+25} = \sqrt{29} = 5,38.$$

Bektore baten luzera neurtu nahi badugu, distantzia euklidearraren definizioa erabil dezakegu. Azken finean, bektore baten luzerak (edo normak) distantzia bat neurtzen du, jatorri puntutik hasita. Hori horrela izanik, ditugun bi bektoreak $\mathbf{v}^T = (v_1, \ldots, v_m)$ eta $\mathbf{0}^T = (0, \ldots, 0)$ dira, eta beren distantzia euklidearra kalkulatuz, zera lortuko dugu:

$$d(\mathbf{v}, \mathbf{0}) = \sqrt{\sum_{i=1}^{m} (0 - v_i)^2} = \sqrt{\sum_{i=1}^{m} v_i^2}.$$

Bektore baten norma edo luzera $\mathbf{v}^T \cdot \mathbf{v}$ biderkadura eskalarraren bidez ere adieraz dezakegu. Horrela definitzen da:

II.1.3 Definizioa. (Bektore baten luzera edo norma)

$\mathbb{R}^m$ espazioko $\mathbf{v}^T = (v_1, \ldots, v_m)$ bektorearen luzera edo norma $||\mathbf{v}||$ notazioaz adierazten da, eta honela definitzen da:

$$||\mathbf{v}|| = \sqrt{\sum_{i=1}^{m} v_i^2} = \sqrt{\mathbf{v}^T \cdot \mathbf{v}}.$$

II.1.3 Adibidea. Adibideko $\mathbf{u}$ eta $\mathbf{v}$ bektoreen norma kalkulatuko dugu.

$$||\mathbf{u}|| = \sqrt{4^2 + 3^2} = 5, \quad ||\mathbf{v}|| = \sqrt{6^2 + 8^2} = 10.$$

Bi bektoreren arteko Distantzia Euklidearraren kalkuluan, bektoreen luzerak eragin handia sortzen du. Izan ere, bi bektoreek adierazten duten puntuen arteko distantzia da kalkulatzen dena. Zenbait kasutan bi bektoreren arteko distantziaren kalkulua haien arteko angeluan oinarrituz neurtzea komeni da, bektoreen luzera kontuan izan gabe. Halakoetan, bektoreak normalizatu egin ohi dira, hau da, unitate bateko luzera duten bektore bihurtzen dira. Horrela, bektoreen arteko antzekotasuna kalkulatzean haien arteko angelua da kontuan hartuko dena, eta ez luzera.

$\mathbf{u}$ bektore bat normalizatzea, bere osagai guztiak $||\mathbf{u}||$ balioaz zatitzea da. Eragiketa horren ondorioz lortuko den bektoreak $\mathbf{u}$ bektorearen noranzko bera izango du, baina unitate bateko luzerara murriztuta.

II.1.4 Adibidea. Adibideko $\mathbf{u}$ eta $\mathbf{v}$ bektoreak normalizatuko ditugu (ikus interpretazio geometrikoa II.1. Irudian).

$$\mathbf{u}' = \begin{pmatrix} \frac{4}{5} \\ \frac{3}{5} \end{pmatrix} = \begin{pmatrix} 0,8 \\ 0,6 \end{pmatrix}, \qquad \mathbf{v}' = \begin{pmatrix} \frac{6}{10} \\ \frac{8}{10} \end{pmatrix} = \begin{pmatrix} 0,6 \\ 0,8 \end{pmatrix}.$$

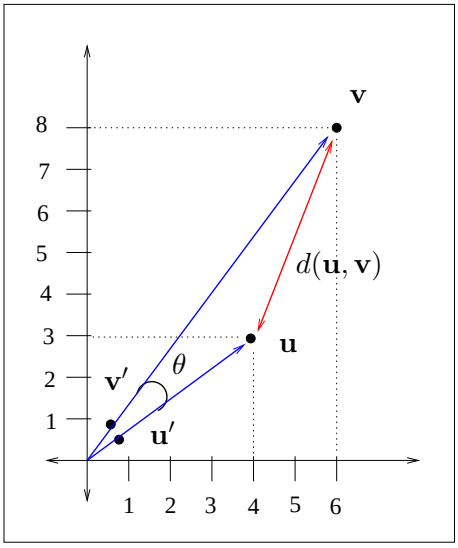
Erraz egiazta daiteke bektoreen luzera 1 dela.

$$||\mathbf{u}'|| = \sqrt{0,8^2 + 0,6^2} = 1, \quad ||\mathbf{v}'|| = \sqrt{0,6^2 + 0,8^2} = 1.$$

Bektoreen arteko antzekotasuna haien angeluan oinarritu nahi den kasuetan oso egokia gertatzen da kosinu-antzekotasuna erabiltzea. Izan ere, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ bi bektore normalizaturen biderkadura eskalarrak haien kosinu-antzekotasuna ematen digu. Kosinu-antzekotasuna honela definitzen da:

II.1.4 Definizioa. (Kosinu-antzekotasuna) Bi bektore $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ izanik, $||\mathbf{u}||$ eta $||\mathbf{v}||$ haien norma eta θ haien arteko angelua, kosinu-antzekotasuna honela kalkulatzen da:

$$\cos(\mathbf{u}, \mathbf{v}) = \cos\theta = \frac{\mathbf{u}^T \cdot \mathbf{v}}{||\mathbf{u}||||\mathbf{v}||} \tag{II.1}$$



II.1 Irudia: Bektoreak, bektoreen arteko distantzia eta antzekotasuna $\mathbb{R}^2$n

II.1.5 Adibidea. Aurreko adibideetako **u** eta **v** bektoreekin jarraituz, haien arteko kosinu-antzekotasuna:

$$\cos(\mathbf{u}, \mathbf{v}) = \cos\theta = \frac{24 + 24}{50} = 0,96.$$

Kosinua oso altua da, 1etik oso gertu dago. Horrek bi bektoreen arteko antzekotasun altua adierazten du.

Bektore-espazioak hainbat problema modelizatzeko erabiliak izan dira. Hizkuntzalaritza Konputazionalaren esparruan, adibidez, LSI aplikazioa sortu zenean, bektore-espazioen erabilera proposatu zen testuen semantika jasotzeko eredu moduan.

Baina ez zen LSI izan bektore-espazioetan oinarritutako lehenengo aplikazioa. Dagoeneko badira ia 50 urte SMART (System for the Mechanical Analysis and Retrieval of Text) Informazio-Berreskuratze sistema proposatu zela (Salton, 1971). Haren oinarri matematikoan bektore-espazioen eredua ageri da. Ikus ditzagun bektore-espazio ereduak testuen semantikarako aplikazio moduan izan zituen hastapenak.

II.2 Bektore-espazio eredua semantikarako

Bektore-espazio ereduari (VSM, Vector Space Model) buruz hitz egitean Salton aipatu beharra dago. Izan ere, Informazio-Berreskuratze (IR, Information Retrieval) ataza finkatzen eta garatzen lan ikaragarria egin zuen. Bera eta bere taldekoak izan ziren bektore-espazio eredua erabiltzea proposatu zutenak (Salton et al., 1975). Horrela sortu zen SMART informazioaren berreskuratzerako sistema. Horri esker, IR atazarako kontzeptu eta teknika asko garatu ziren. Esan daiteke Saltonek eta bere kideek lan handia egin zutela testuetako hitzen agerpenen arteko mendekotasunak aztertzen, eta esperimentu asko egin zituztela agerpenen arteko erlazio horiek informazioaren berreskuratzerako erabiltzeko. Haiek egindako lanak gaur egungo Informazio-Berreskuratze sistemen oinarria finkatu zuten.

Testuen prozesamendu automatikoari buruz Saltonek idatzitako liburuan bektore-espazio ereduaren definizio zabala dator (Salton, 1989). Bertan argi aipatzen da eredua nola erabil daitekeen Informazio-Berreskuratze atazarako. VSM eredua informazioaren berreskuratzerako prozedura bat dela esaten da. Dokumentuak eta kontsultak (query) bektoreen bidez adierazten dira, termino multzo batean oinarritutako bektore-adierazpenean hain zuzen ere. Bektoreko osagaiak zenbaki errealak dira, eta posizio bakoitzean dagoen balioak posizioari dagokion terminoa dokumentuan edo kontsultan ageri deneko maiztasuna adierazten du (edota ageri den edo ez, adibidez). Kontsulten eta dokumentuen arteko antzekotasunak kalkulatzen dira antzekotasun neurriak erabiliz (kosinu-antzekotasuna, adibidez).

Azken finean, VSM eredu matematiko bat da, eta bere erabilera erakargarria gertatzen da. Testuetatik informazio semantikoa ateratzeko oso baliagarria da, eta ez da nekeza gertatzen VSM-ren erabilera. Corpus batetik abiatu behar da, hori bai, baina hortik aurrera nahiko modu automatikoan lortzen da informazio semantikoa. Semantikaren atazarako beste saiakera batzuek egin izan dira, hala nola eskuz kodetutako datu-baseak eraikitzea edota ontologiak sortzea. VSM ereduak ez du halakorik eskatzen, eta emaitza onak ematen ditu testuetako hitz, esaldi eta dokumentuen arteko antzekotasuna neurtzea eskatzen duten atazatan.

VSM eredua proposatu aurretik ere, bektoreak erabili izan dira adierazpen eredu moduan. VSM ereduak ekarri zuen benetako berrikuntza, testuz osatutako corpus bateko hitzen maiztasunak erabiltzea izan zen, haien bidez informazio semantikoa aurkitzeko. Testuetako hitzen maiztasunen analisia Semantika Estatistikoa izenez ezagutzen da. Semantika Estatistikoa hitzen maiztasunen azterketan oinarritzen da hitzen esanahia eta haien arteko erlazioak aztertzeko; gizakiek hitzei emandako erabileratik lortutako egitu-

ra/eredu estatistikoak erabiltzean datza esanahia ulertzeko asmoz. Horrela, testu multzo handietan oinarrituz, eta hitzen agerpenen maiztasunak aztertuz hitz horiek testuaren esanahia zenbateraino jasotzen duten neurtu ahal izango da.

VSM ereduak testuetako hitzen eta testu zatien (dokumentuen) azterketarako balio duela ikusi zen, eta VSM ereduen aplikazio berriak sortu ziren, hala nola Hitzen Adiera Desanbiguatzearen ataza (WSD, Word Sense Disambiguation). (Furnas et al., 1983) ikerlariek Semantika Estatistikoari ekarpen handia egin zioten. Geroztik, makina bat algoritmo proposatu izan da, agerpenen maiztasunak aztertuz semantikaren azterketan sakontzeko, beti ere corpus handietan oinarrituz. Latent Semantic Analysis bera (LSA edo LSI) ikerketa lan horietatik sortutako aplikazio arrakastatsu bat da (Deerwester et al., 1990). VSM ereduaren aplikazioa hasiera batean Informazio-Berreskuratze atazarako proposatu bazen ere, izan zuen arrakastak eraginda Hizkuntzaren Prozesamenduko beste hainbat atazatara zabaltzen hasi zen, eta oso emaitza onekin gainera (Turney and Pantel, 2010).

Atal honetan ikusiko dugu zeintzuk diren VSM eredua testuen semantika jasotzeko erabili nahi denean eman behar diren urratsak. II.2.1. atalean testuen aurreprozesamendu linguistikorako oinarrizko tekniken aipamena egingo dugu. Ondoren, II.2.2. atalean prozesamendu matematikoaz arituko gara: matrizearen sorrera, antzekotasun semantikoen kalkulua eta matrizearen eraldaketa.

II.2.1 Aurreprozesamendu linguistikoa

Testuetako hitzen eta testu zatien bektore bidezko adierazpena kalkulatu aurretik, corpusa osatzen duten testuen prozesamendu linguistikoa egitea oso gomendagarria gertatzen da. Atal honen helburua ez da prozesamendu linguistikorako teknika horien azterketa zabala egitea, baina lan honetan erabilitakoen aipamena egiteak merezi du.

- Tokenizazioa. Tokenizatzea testu bat tokenetan banatzea da. Hizkuntzaren arabera tokenizazio prozesuaren zailtasuna aldatu egiten da. Normalean tokena hitza izan ohi da (zuriune edo puntuazio marken arteko karaktere segida), baina ezin da horretara bakarrik mugatu. Izan ere, testuetan zenbakiak eta karaktere bereziak agertu ohi dira testuarekin batera, eta hitza non hasi eta non bukatzen den bereiztea zaila gerta daiteke. Normalean zenbaki eta hizkiek osatzen dute tokena, eta zuriune eta puntuaketa markak ezabatu egiten dira, baina komeni bada, tokenaren definizio zorrotzagoa egin daiteke.

Lan honetan, ingelesezko edo euskarazko testuez osatutako corpusak erabili ditugu, eta hizkuntza horietarako tokenizazioa nahiko ondo zehaztuta dago. Esperimentuetan LSIren tokenizazioa erabili dugu, eta beraz, puntuaketa markak kendu egin dira, salbuespenak salbuespen.

- "Stop words" zerrendak. Testuetan oso maiztasun altuan ageri diren hitzak eduki semantiko eskasekoak izan ohi dira. Ingelesez "stop words" izenez ezagutzen diren hitz horiek normalean ezabatu egiten dira testuen aurreprozesamendu fasean. Hitz horiekin osatutako zerrenda ezberdinak existitzen dira hainbat hizkuntzatarako. Ingeleserako, adibidez, zerrenda bat baino gehiago existitzen da eta erraz aurki daitezke Interneten. Euskararako ere IXA taldeak sortuak ditu maiztasun altuko euskal hitzen zerrenda batzuk (formen zerrenda eta lemen zerrenda). Hala ere, Informazio-Berreskuratze atazarako LSI erabiltzen den kasuan, zerrenda orokor horietako bat erabiltzea kaltegarri izan daitekeela ikusi da (Zaman et al., 2011).

- Erro-bilaketa. Dokumentuetan erro bereko hitz desberdin asko ager daitezke, nahiz eta guztiak funtsean esanahi berekoak izan (*etxea*, *etxeak*, *etxetik*, *etxearen*...). Hitzen semantika bektoreen bidez adieraztea helburu izanik, zentzuzkoa dirudi aldaera horiek guztiak forma bakarrera laburtzea. Erro-bilaketa hitzen erroa erauztea da, dokumentuan erroak besterik ez uzteko (*etxe*). Erro horrentzat adierazpen bektorial bakarra sortuko da.

  Erro-bilaketarako algoritmo ugari existitzen diren arren, ezagunena Porter-en algoritmoa da (Porter, 1980). Ikerketa lan honetan ingelesez idatzitako testuekin lan egin dugunetan, Porter-en erro-bilatzailea erabili dugu. Euskararako, aldiz, hizkuntza eranskaria izanik, Porter-en algoritmoa ez da egokia gertatzen. Izan ere, euskarak deklinabide sistema aberatsa du, atzizki eta aurrizki ugari ditu, eta lematizazio prozesua beste zenbait hizkuntzatan baino konplexuagoa gertatzen da. IXA taldeak garatutako lematizatzailea da guk euskaraz idatzitako testuekin erabili duguna (Ezeiza et al., 1998).

II.2.2 Prozesamendu matematikoa

Testuen aurreprozesamendu linguistikoa amaitu denean prozesamendu matematikoari ekingo zaio. II.2.2.1. atalean maiztasunen matrizearen sorrera azalduko dugu, eta II.2.2.2. atalean dokumentuen arteko antzekotasun semantikoen kalkulua.

Maiztasunen matrizearekin lan egin beharrean, posiblea da matrize horretako elementuak eraldatzea, eraldaketarekin testuetako semantika modu egokiagoan jasotzeko asmoz. II.2.2.3. atalean matrizearen eraldaketarako oso sarri erabiltzen diren bi metodotan sakonduko dugu: tf-idf ponderazioa eta log-entropy ponderazioa.

II.2.2.1 Maiztasunen matrizea sortzen

Bektore-espazio eredua erabiltzean, *termino-dokumentu* matrize bat eraikiko da. Oro har, *terminoak* dokumentuan ageri diren hitzak dira, baina hitz guztiak ez dira termino moduan aukeratuak izaten. Matrizea eraikitzeko garaian estrategia desberdinak erabil daitezke dokumentuko hitzen artean termino zeintzuk izango diren erabakitzeko. LSIrekin, adibidez, dokumentu kopuru minimo batean agertzen ez diren hitzak ez dira termino izango; kopuru hori finkatzea erabiltzailearen esku geratzen da.

Behin terminoak erauzi direnean, terminoek dokumentuetan duten agerpen maiztasuna aztertzen da. Horri dokumentuak **indexatzea** esaten zaio. Dokumentu guztiak indexatuak izan direnean, n dokumentuz osatutako corpus batetik abiatuz m termino erauzi badira, $\mathbf{M} \in \mathbb{R}^{m \times n}$ matrizea eraikitzen 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.

Corpus eta terminoen hiztegi oso handietarako lortzen den matrizean zero kopuru oso handia agertu ohi da. Halako matrizeak dentsitate gutxikoak (sparse) direla esaten da, eta bektoreen arteko kosinu-antzekotasuna kalkulatzea oso eraginkorra gertatzen da (Baeza-Yates and Ribeiro-Neto, 1999). Horrek kosinu-antzekotasunaren neurriaren erabilera bultzatu izan du.

II.2.1 Adibidea. Zortzi dokumentuz osatutako corpus bat dugu; lehenengo lau dokumentuak Hitzen Adiera Desanbiguazioari (WSD) buruzkoak dira (d_1, d_2, d_3, d_4), eta gainerako lauak Informazio-Berreskuratzeari (IR) buruzkoak (d_5, d_6, d_7, d_8). Corpusetik 10 termino erauzi eta II.2. Irudiko $\mathbf{M} \in \mathbb{R}^{10 \times 8}$ matrizea sortu da. Matrizeko m_{ij} osagaiak maiztasunak dira, hau da, t_i terminoaren agerpen kopurua d_j dokumentuan.

Zortzi dokumentuak Hizkuntzalaritza Konputazionalaren bi esparrukoak dira, hau da, semantikoki ez daude oso urrun. Hala ere, erauzi diren 10 termi-

$$
\mathbf{M} = \begin{array}{c}
\begin{array}{cccccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\
\begin{pmatrix}
8 & 3 & 9 & 4 & 0 & 0 & 0 & 0 \\
10 & 5 & 7 & 6 & 0 & 0 & 0 & 0 \\
3 & 6 & 5 & 2 & 0 & 1 & 0 & 2 \\
6 & 8 & 10 & 8 & 0 & 2 & 1 & 3 \\
2 & 0 & 5 & 0 & 1 & 0 & 1 & 0 \\
1 & 2 & 0 & 0 & 8 & 6 & 0 & 5 \\
0 & 0 & 0 & 0 & 0 & 7 & 0 & 6 \\
0 & 0 & 0 & 0 & 7 & 4 & 8 & 4 \\
3 & 1 & 2 & 0 & 7 & 6 & 0 & 2 \\
5 & 4 & 7 & 6 & 2 & 3 & 2 & 1
\end{pmatrix}
\end{array}
\begin{array}{l}
\\
\rightarrow \; t_1 = \text{adiera} \\
\rightarrow \; t_2 = \text{desanbiguazio} \\
\rightarrow \; t_3 = \text{polisemia} \\
\rightarrow \; t_4 = \text{hitz} \\
\rightarrow \; t_5 = \text{esanahi} \\
\rightarrow \; t_6 = \text{testu} \\
\rightarrow \; t_7 = \text{berreskuratu} \\
\rightarrow \; t_8 = \text{indexatu} \\
\rightarrow \; t_9 = \text{automatiko} \\
\rightarrow \; t_{10} = \text{semantika}
\end{array}
$$

II.2 Irudia: Termino-dokumentu matrize baten adibidea

no horiek aztertuz gero, nabari daiteke termino batzuk WSD atazara gehiago gerturatzen direla (adiera, desanbiguazio); beste termino batzuk, aldiz, IR atazatik gertuago daude (berreskuratu, indexatu). Egia esan, aipatutako lau termino horiek ataza berekoak diren dokumentuetan ageri dira, baina ez beste atazakoak direnetan. Dokumentuen banaketa garbia erakusten dute. Beste termino batzuek modu sakabanatuagoan ageri dira 8 dokumentuetan; semantika, adibidez, dokumentu guztietan ageri da maiztasun desberdinekin.

II.2.2.2 Antzekotasun semantikoak kalkulatzen

Bektore-espazio ereduan, dimentsio bakoitza termino bati dagokio. Bi bektorek termino asko konpartitzen badituzte, antzeko noranzkoan adieraziak izango dira bektore-espazioan; beren arteko angelua txikia izanik, kosinu--antzekotasuna altua izango da.

Dokumentuen arteko antzekotasun semantikoaren neurri moduan kosi-

nu-antzekotasun neurria da LSI aplikazioak erabiltzen duena (ikus 18. orriko II.1. formula). 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.

II.2.2 Adibidea. Aurreko adibideko **M** matrizean, dokumentu bakoitza $\mathbb{R}^{10}$ bektore-espazioko bektore baten bidez adierazten da, guztira 10 termino aukeratuak izan direlako. Bektorearen osagai bakoitza espazioko dimentsio bati dagokio: lehenengo dimentsioa, adibidez, "adiera" terminoari dagokio. $\mathbb{R}^{10}$ bektore-espazioan bektore horien arteko kosinu-antzekotasuna kalkulatuz neurtuko dugu dokumentu horien arteko antzekotasun semantikoa. II.1. Taulan matrizeko dokumentuen arteko kosinu-antzekotasunak erakusten dira.

		WSD				IR			
		d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
WSD	d_1	1							
	d_2	0,84	1						
	d_3	0,93	0,89	1					
	d_4	0,89	0,91	0,92	1				
IR	d_5	0,16	0,15	0,11	0,06	1			
	d_6	0,28	0,34	0,26	0,23	0,97	1		
	d_7	0,14	0,15	0,19	0,19	0,45	0,39	1	
	d_8	0,26	0,43	0,29	0,28	0,90	0,93	0,45	1

II.1 Taula: **M** matrizeko dokumentuen arteko kosinu-antzekotasunak

Ikus daitekeenez, arlo bereko dokumentuen arteko kosinu gehienak nahiko altuak dira (1etik gertu), eta arlo desberdineko dokumentuen artekoak, aldiz, nahiko baxuak (0tik gertukoak), baina ez kasu guztietan. Azter ditzagun, adibidez, d_6-d_7 eta d_2-d_8 dokumentu pareen arteko antzekotasuna.

- d_6-d_7 dokumentu parea. Arlo berekoak izateko kosinu-antzekotasuna

baxua da $(0,39)$. $\mathbf{M}$ matrizean duten bektore-adierazpena aztertuz, ikusten da 3 termino besterik ez dituztela komun (hitz, indexatu eta semantika).

- d_2-d_8 dokumentu parea. Arlo desberdinekoak izateko, kosinu-antzekotasuna altua da $(0,43)$. $\mathbf{M}$ matrizean 5 termino konpartitzen dituzte (polisemia, hitz, testu, automatiko, semantika).

Dokumentuen bektore-adierazpena terminoek baldintzatzen dute. Geometrikoki interpretatuz gero, antzeko noranzkoa izango dute termino asko komun dituzten dokumentuek. Zenbat eta termino gehiago izan komun, orduan eta txikiagoa izango da bektoreen arteko angelua, eta ondorioz, handiagoa kosinu-antzekotasuna.

Aurreko adibideak nahiko argi erakusten du VSM ereduaren planteamendu klasikoaren funtzionamendua. Corpusetik erauzitako terminoek (dimentsioek) eta haien agerpen maiztasunek baldintzatuko dituzte dokumentuen bektore-adierazpenak eta beren arteko erlazioak. Tamalez, eredu honek ez du gaitasunik Hizkuntzalaritza Konputazionalean klasikoa den problema bati aurre egiteko: sinonimia eta polisemia.

- Sinonimiaren arazoa aurkituko dugu bi hitz desberdinek esanahi bera dutenean, adibidez, amama eta amona. 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, arte (zuhaitz mota bat adieraz dezake, edo giza-ekintza bat pintura eta eskultura bezala, besteak beste). Hitzak adierazpen bakarra izango du bektore-espazio ereduan, nahiz eta dokumentu guztietan ez duen esanahi bera izango.

II.3.2. atalean ikusiko dugu bektore-espazioaren dimentsioa murriztuz eta dokumentuen arteko antzekotasuna bertan neurtuz lortuko dela neurri batean sinonimiaren eta polisemiaren arazoari aurre egitea.

Bestalde, bektore-espazio eredua erabiltzen denean, maiztasunen matrizearekin zuzenean lan egitea ez da beti aukerarik aproposena gertatzen. Hurrengo atalean ikusiko dugu maiztasunen matrizeko m_{ij} elementuen eraldaketa egitea oso egokia gertatzen dela testuen semantika harrapatzea helburu duten aplikazioetan.

II.2.2.3 Matrizeko elementuen eraldaketa

Dokumentuen arteko antzekotasun semantikoa kalkulatzeko, maiztasunen termino-dokumentu matrizea bere horretan erabil badaiteke ere, oso gomendagarria gertatzen da maiztasunen matrize hori ponderazioen matrize moduan ezagutu ohi den beste matrize batean eraldatzea. Matrizea eraldatzearen helburua dokumentuetatik erauzitako terminoen garrantzia maila neurtuz, dokumentuen semantika hobeto adieraziko duen beste matrize bat lortzea da.

Izan ere, maiztasunen matrizeko m_{ij} elementuak terminoen garrantzia neurtzeko oso egokiak izan badaitezke ere, ez da beti horrela gertatzen, informazio lokala ematen digutelako (t_i terminoak d_j dokumentua duen agerpen maiztasuna), terminoaren agerpena corpus mailan kontuan hartu gabe. Pentsa, adibidez, euskaraz oso maiztasun altuan ageri diren *eta*, *edo*, *ez*, *hau*... moduko hitzen maiztasunean. Corpuseko dokumentu guztietan nahiko modu homogeneoan sakabanatuta agertu ohi dira eta termino moduan erauziak izanez gero, matrizean m_{ij} balio oso altuak izango dituzte d_j dokumentu guztietarako. Oso arruntak diren hitzak eduki semantiko eskasekoak dira; ez dira oso esanguratsuak izaten, ez dutelako dokumentuen semantikari buruzko informazio handiegirik ematen. Hori dela eta, ez dute dokumentuen artean bereizketarik egiteko gaitasunik eta ohikoa da halakoak kentzea "Stop words" zerrendak erabiliz.

Matrizearen eraldaketa ponderazioen bidez egiten da. Ponderazio lokalak eta ponderazio globalak definitzen dira. t_i terminoak d_j dokumentuan duen garrantzia neurtzeko ponderazio lokalak definitzen dira, eta corpus mailan duen garrantzia neurtzeko ponderazio globalak. Ponderazio horiek kalkulatzeko oinarrizko balio hauek erabiltzen dira normalean:

- $\mathrm{tf}(t_i, d_j)$: "term frequency" edo terminoaren maiztasuna. t_i terminoaren maiztasuna d_j dokumentuan. Jatorrizko $\mathbf{M}$ matrizea maiztasunen matrize bat den kasuetan, bertan ageri diren m_{ij} balioak terminoen maiztasunak dira.
$$\mathrm{tf}(t_i, d_j) = m_{ij}.$$

- $\mathrm{gf}(t_i)$: "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:

$$\mathrm{df}(t_i) = \sum_{j=1}^{n} \min\{m_{ij}, 1\}.$$

Neurri horietan oinarrituz ponderazio lokalak eta globalak definitzeko aukera desberdinak daude. Ohikoenak azalduko ditugu hemen, baina horietaz gain, ponderazio lokalen eta globalen beste aukera asko aurki daitezke bibliografian, eta jatorrizko matrizearen eraldaketarako estrategia desberdinak sortzen dira (Salton and Buckley, 1988, Manning et al., 2008, Zobel and Moffat, 1998). LSIn erabilitako terminoen ponderazioari buruz ere bada bibliografia ugari (Dumais, 1991, Berry and Browne, 2005, Landauer et al., 2013). Hona hemen oso ohikoak diren ponderazio lokal eta ponderazio global batzuk:

- Ponderazio lokala (l_{ij}). Terminoaren garrantzia maila modu lokalean (dokumentu barruan) neurtzen du.
  - tf_{ij} ponderazio lokalak terminoen maiztasunak bere horretan uzten ditu.
    $$\mathrm{tf}_{ij} = \mathrm{tf}(t_i, d_j) = m_{ij}.$$
  - $\log_{ij}$ ponderazio lokalak logaritmo funtzioa aplikatzen die terminoen agerpen maiztasunei, matrizeko maiztasunen arteko diferentziak leuntzeko asmoz. $m_{ij} = 0$ maiztasuna duten terminoen ponderazio lokalarekin arazorik ez izateko, horrela definitu ohi da.
    $$\log_{ij} = \log_2(\mathrm{tf}(t_i, d_j) + 1).$$
- Ponderazio globala (g_i). Terminoaren garrantzia maila modu globalean (corpus barruan) neurtzen du.

  idf_i edo "inverse document frequency" eta $\mathrm{entropy}_i$ ponderazio globalen definizioan sakonduko dugu segidan, IR atazatan duten erabilpen zabalagatik. idf_i eta $\mathrm{entropy}_i$ ponderazio globalek antzeko moduan jokatzen dute, dokumentu kopuru txikiagoan ageri diren terminoei ponderazio global handiagoa esleituz.

Ponderazio lokalak eta globalak kalkulatu ondoren, horrela lortzen dira eraldatutako matrizeko m'_{ij} elementu berriak:

$$m'_{ij} = \mathrm{l}_{ij} \times \mathrm{g}_i.$$

Ondorengo ataletan idf_i eta entropy_i ponderazio globalak aztertuko ditugu, ponderazio lokalekin konbinatuz osatzen dituzten bi ponderazio eredu ezagunetan: tf-idf eta log-entropy.

II.2.2.3.1 tf-idf ponderazio eredua idf ponderazio globala tf ponderazio lokalarekin konbinatuta erabiltzen da askotan, oso ezaguna den tf-idf izeneko ponderazio ereduan. Informazio-Berreskuratze arloan aspaldidanik oso erabilia izaten ari da tf-idf.

Corpus bat izanik, t_i terminoari dagokion idf_i ponderazio globala honela kalkula daiteke:

$$\text{idf}_i = \log_2 \frac{n}{\text{df}(t_i)} = \log_2 n - \log_2 \text{df}(t_i),$$

n izanik corpusean guztira dagoen dokumentu kopurua.

Formulak ponderazio maximoa egokituko die dokumentu bakarrean ageri diren terminoei ($\text{df}(t_i) = 1$ denean, $\text{idf}_i = \log_2 n$), eta ponderazio minimoa dokumentu guztietan ageri diren terminoei ($\text{df}(t_i) = n$ denean, $\text{idf}_i = 0$).

Formula horrek erakusten du idf_i ponderazio globalaren ideia orokorra. Hala ere, bibliografian formula horren aldaera txiki desberdinak aurki daitezke, behekoa da horietako bat, $\text{idf}_i = 0$ ekiditeko asmoz definitua. Hortaz, idf_i ponderazio globalaren definizio bat:

$$\boxed{\text{idf}_i = 1 + \log_2 \frac{n}{\text{df}(t_i)}.}$$

Matrizeko m_{ij} elementuen tf-idf ponderazio bidezko eraldaketa honela egiten da:

$$m'_{ij} = \text{tf}_{ij} \times \text{idf}_i.$$

II.2.3 Adibidea. Demagun zazpi dokumentuz osatutako corpus bat dugula eta lau termino izan direla erauziak. II.3. Irudian, goiko taulan $m_{ij} = \text{tf}(t_i, d_j)$ maiztasunak eta lau terminoen idf_i ponderazio globalak ageri dira. tf-idf ponderazioa aplikatu eta gero, eraldatutako matrizeko m'_{ij} elementuak irudi bereko beheko taulan ikus daitezke.

Ponderazio globalak zenbat eta altuagoak, orduan eta gehiago hazi dira matrizeko elementuak. $\text{idf}_4 > \text{idf}_1 = \text{idf}_3 > \text{idf}_2$. Lau ponderazio globalen artean altuena t_4 terminoarena da, eta elementuen balioak igo arazi egin ditu. t_2 terminoarena aldiz $\text{idf}_2 = 1$ izanik, matrizeko balioak bere horretan mantendu dira. t_1 eta t_3 terminoei ponderazio global bera egokitu zaie, biak agertzen baitira corpuseko 7 dokumentuetatik 3tan.

m_{ij}	d_1	d_2	d_3	d_4	d_5	d_6	d_7	$\text{idf}_i = 1 + \log_2 \frac{n}{df(t_i)}$
t_1	2	0	5	0	0	8	0	$\text{idf}_1 = 1 + \log_2 \frac{7}{3} = 2,22$
t_2	5	3	5	4	5	6	5	$\text{idf}_2 = 1 + \log_2 \frac{7}{7} = 1$
t_3	100	0	105	0	500	0	0	$\text{idf}_3 = 1 + \log_2 \frac{7}{3} = 2,22$
t_4	0	0	0	10	0	0	0	$\text{idf}_4 = 1 + \log_2 \frac{7}{1} = 3,81$

m'_{ij}	d_1	d_2	d_3	d_4	d_5	d_6	d_7
t_1	4,44	0	11,1	0	0	17,76	0
t_2	5	3	5	4	5	6	5
t_3	222	0	233,1	0	1110	0	0
t_4	0	0	0	38,1	0	0	0

II.3 Irudia: Matrizeko elementuen tf-idf eraldaketa

II.2.2.3.2 Log-entropy ponderazioa Entropy ponderazio globala log ponderazio lokalarekin konbinatuta log-entropy izeneko ponderazio ereduan erabiltzen da askotan. LSIrekin egindako hainbat esperimentuetan log-entropy ponderazio eredua erabiliz lortu izan dira emaitzarik onenak (Dumais, 1991).

Entropiaren kontzeptua Informazio-Teoriatik dator. Claude Shannon matematikariak ezarri zituen Informazio-Teoriaren oinarriak 1948an argitaratutako lanean eta harrez geroztik Informazio Teoriaren aita moduan ezaguna da (Shannon, 1948). Informazio-teoria probabilitate-teorian eta estatistikan oinarritzen da. Entropiaren kontzeptua, zorizko aldagai baten informazio-kantitatearen definizioan oinarrituz ematen da. Entropiaren kontzeptua ulertzeko, ikus dezagun lehenik informazio kantitatea nola definitzen den.

Definizioz, probabilitatezko gertakari baten informazio-kantitatea gertakari horren probabilitatearen mende dago: zenbat eta txikiagoa izan gertakariaren probabilitatea, orduan eta handiago izango da gertakari horri dagokion informazio-kantitatea. Termino-dokumentu matrizean oinarritutako aplikazioetarako horrela interpreta daiteke: corpus bat izanik, termino bat dokumentu batean agertzeko probabilitatea txikia bada, altua izango da

dokumentuaren semantikari buruz ematen duen informazioa. Probabilitate altuko gertakariek, aldiz, informazio-kantitate txikia emango dute.

Informazio-kantitatearen kontzeptua modu formalean honela defini daiteke. C zorizko aldagaiak $c_1, \ldots, c_n$ balioak $p(c_1), \ldots, p(c_n)$ probabilitatez hartzen baditu, orduan c_i gertakariari dagokion $\mathrm{I}(c_i)$ informazio-kantitatea horrela kalkulatuko da:

$$\mathrm{I}(c_i) = -\log_2 p(c_i).$$

Definizio horren arabera, $p(c_i)$ probabilitatea 1era gerturatzen den neurrian, $\mathrm{I}(c_i)$ balioa zerora gerturatuko da, eta $p(c_i)$ zerora gerturatzen bada, $\mathrm{I}(c_i)$ balioak infiniturantz joko du.

Informazio kantitatearen definizioak entropiaren kontzeptura garamatza. Entropiak ziurgabetasuna adierazten du, kaosa, eta formalki honela defini daiteke. C zorizko aldagaiari dagokion $\mathrm{I}(C)$ informazio-kantitatea zorizko aldagaiak $\mathrm{I}(c_1), \ldots, \mathrm{I}(c_n)$ balioak hartzen baditu $p(c_1), \ldots, p(c_n)$ probabilitatez, orduan $H(C)$ edo Shannon-en entropia horrela kalkulatuko da:

$$\mathrm{H}(C) = \mathrm{E}(\mathrm{I}(C)) = -\sum_{i=1}^{n} p(c_i) \log_2 p(c_i).$$

$\mathrm{E}(\mathrm{I}(C))$ izanik $\mathrm{I}(C)$-ren itxaropen matematikoa. Termino-dokumentu matrize bat izanik, t_i terminoaren agerpena corpus mailan aztertu nahi badugu, ez da gauza bera izango terminoa dokumentu guztietan antzeko maiztasunez (probabilitatez) agertzea edo dokumentu gutxi batzuetan besterik ez agertzea. Entropiaren formula testuinguru horretara horrela egoki daiteke:

$$\mathrm{H}(t_i) = -\sum_{j=1}^{n} p(t_i, d_j) \log_2(p(t_i, d_j)).$$

Entropiaren neurri horrek t_i terminoaren sakabanatze-maila emango digu; zenbat eta sakabanatuagoa terminoaren agerpena corpuseko dokumentuetan, orduan eta altuagoa entropia (ziurgabetasuna).

$\mathrm{entropy}_i$ izeneko ponderazio globala Informazio-Teoriako entropiaren kontzeptuan oinarrituz definitzen da. Hona hemen definizio formala:

$$\boxed{\mathrm{entropy}_i = 1 + \sum_{j=1}^{n} \frac{p(t_i, d_j) \log_2 p(t_i, d_j)}{\log_2 n}, \quad \text{non} \quad p(t_i, d_j) = \frac{\mathrm{tf}(t_i, d_j)}{\mathrm{gf}(t_i)}.}$$

$\mathrm{entropy}_i$ ponderazio globalaren formulan $\mathrm{H}(t_i)$ entropia normalizatuta ageri da. Izan ere, n balio posible dituen t_i aldagaiaren $\mathrm{H}(t_i)$ entropiaren balio maximoa $\log_2 n$ baita. Formulak $\mathrm{entropy}_i$ ponderazio maximoa egokituko

die dokumentu gutxitan ageri diren t_i terminoei eta ponderazio minimoa dokumentu guztietan ageri direnei.

entropy$_i$ ponderazio globala $\log_{ij}$ ponderazio lokalarekin konbinatzen denean, matrizeko m'_{ij} balioak horrela kalkulatuko dira:

$$m'_{ij} = \log_{ij} \times \text{entropy}_i.$$

II.2.4 Adibidea. Aurreko II.2.3 adibideko maiztasunen matrizeko elementuak log-entropy ponderazioa aplikatuz eraldatuko ditugu. II.4. Irudian goiko taulan ageri dira elementuen $\log_{ij}$ ponderazio lokalak eta terminoen entropy$_i$ ponderazio globalak. m_{11} elementuaren eraldaketa lokala eta entropy$_1$ ponderazio globala, adibidez, horrela kalkulatuak izan dira:

$$\log_{11} = \log_2(\text{tf}(t_1, d_1) + 1) = \log_2(2 + 1) = 1,58.$$

$$\text{entropy}_1 = 1 + \frac{\frac{2}{15}\log_2\frac{2}{15} + \frac{5}{15}\log_2\frac{5}{15} + \frac{8}{15}\log_2\frac{8}{15}}{\log_2 7} = 0,5.$$

Oraingoan ere t_2 terminoak du ponderazio global txikiena (entropy$_4$ > entropy$_3$ > entropy$_1$ > entropy$_2$), bere agerpena dokumentuetan oso homogeneoa delako. Horrek matrizeko balioak jaitsi arazi ditu. t_4 terminoak, aldiz, dokumentu bakarrean ageri denez, ponderazio global altua lortu du. Log-entropy ponderazioa aplikatu eta gero, eraldatutako matrizeko m'_{ij} balioak II.4. Irudiko beheko taulan ageri dira.

II.3 Matrizearen deskonposaketa: SVD eta LSI

Atal honetan matrize-deskonposaketarako existitzen diren metodoen arteko bat azalduko dugu: SVD (Singular Value Decomposition edo Balio Singularretan Deskonposatzea) izenekoa. SVD deskonposaketak bektore-espazioaren dimentsioaren murrizketa egiteko aukera ematen du.

Dimentsioa murriztean, terminoen eta dokumentuen arteko erlazio semantikoak hobeto erakutsiko dituen adierazpen bektoriala lor daitekeela ikusiko dugu. Dimentsioaren murrizketaren aplikazio bat besterik ez da LSI, arrakasta handiz erabilia izan dena Informazio-Berreskuratze alorrean bereziki. Ikerketa lan honetan egindako esperimentuetan LSIren dimentsio-murrizketa erabili dugu, bai Testu-Sailkatze atazan eta baita Hizkuntzalaritza Konputazionaleko beste bi atazatan ere.

II.3.1. atalean SVD matrize-deskonposaketa aztertuko dugu. Ondoren, II.3.2. atalean SVD deskonposaketan oinarritutako dimentsio-murrizketa azalduko dugu. Amaitzeko, II.3.3. atalean LSI aplikazioa aurkeztuko dugu.

	$l_{ij} = \log_{ij}$							$g_i = \text{entropy}_i$	
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	$gf(t_i)$	$g_i = \text{entropy}_i$
t_1	1,58	0	2,58	0	0	3,17	0	15	0,5
t_2	2,58	2	2,58	2,32	2,58	2,81	2,58	33	0,01
t_3	6,66	0	6,73	0	8,97	0	0	705	0,59
t_4	0	0	0	3,46	0	0	0	10	1

m'_{ij}	d_1	d_2	d_3	d_4	d_5	d_6	d_7
t_1	0,79	0	1,29	0	0	1,59	0
t_2	0,03	0,02	0,03	0,02	0,03	0,03	0,03
t_3	3,93	0	3,97	0	5,29	0	0
t_4	0	0	0	3,46	0	0	0

II.4 Irudia: $\log_{ij}$ ponderazio lokalak, entropy_i ponderazio globalak eta eraldatutako matrizea

II.3.1 SVD matrize-deskonposaketa

Aljebra linealean, matrize bat beste matrize batzuen biderketa moduan idazteari matrizea deskonposatzea esaten zaio. LSI tresnak matrizearen SVD deskonposaketa kalkulatu eta bektore-espazioaren dimentsioa murrizteko aukera ematen du. Badira bektore-espazioaren dimentsioa murrizteko aukera ematen duten beste matrize-deskonposaketa batzuk, QR faktorizazioa adibidez. Hala ere, SVD deskonposaketak badu abantaila handi bat: matrizeko terminoak eta dokumentuak dimentsio murriztuko espazio berean proiektatzea ahalbidetzen du. Horrek testuen analisi semantikorako aukera zabalak eskaintzen ditu.

Dimentsioaren murrizketari esker problemen soluzioa modu eraginkorragoan aurkitzea posible gertatzen da, dimentsio gutxiagorekin lan egitean kalkulu gutxiago egin beharko direlako eta datuak biltegiratzeko espazio gutxiago beharko delako.

SVD deskonposaketaren xehetasunak aztertu aurretik, matrize-teoriako definizio batzuk laburbilduko ditugu segidan.

- Tamaina bereko $\mathbf{L}, \mathbf{M} \in \mathbb{R}^{m \times n}$ bi matrizeren arteko **batura** (kendura) matrizeetako osagaiak batuz (kenduz) lortuko den $\mathbf{N} \in \mathbb{R}^{m \times n}$ matrize bat da. $\mathbf{L} + \mathbf{M} = \mathbf{N}, \quad l_{ij} + m_{ij} = n_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n.$

- $\mathbf{M} \in \mathbb{R}^{m \times n}$ **matrizearen heina**, linealki independentea den errenkada edo zutabe kopurua da, eta rank($\mathbf{M}$) notazioaz adierazten da. rank($\mathbf{M}$) $\leq \min\{m, n\}$ betetzen da.

- Matrize batean diagonal nagusitik kanpo dauden elementu guztiak zero badira, matrizea **diagonala** dela esaten da. Diagonalean zeroren desberdina den elementu kopurua matrizearen heina da.

- $\mathbb{R}$-ren gaineko $(V, +, \cdot_{\mathbb{R}})$ bektore-espazioa izanik, $\emptyset \neq U \subseteq V$ azpimultzoa V-ren **azpiespazioa** dela esango dugu baldin,

  $$(+)\text{-rekiko itxia bada: } \forall \mathbf{v}_1, \mathbf{v}_2 \in U \quad \Rightarrow \quad \mathbf{v}_1 + \mathbf{v}_2 \in U$$
  $$(\cdot_{\mathbb{R}})\text{-rekiko itxia bada: } \forall c \in \mathbb{R}, \quad \forall \mathbf{v} \in U \quad \Rightarrow \quad c \cdot \mathbf{v} \in U$$

  U multzoa V-ren azpiespazioa bada, orduan bektore-espazioa da.

- $\mathbf{M} \in \mathbb{R}^{m \times n}$ matrizea izanik, $\mathbf{M}$-ren n zutabeek sortzen duten $\mathbb{R}^m$-ko azpiespazioari $\mathbf{M}$-ren **zutabe-espazio** esaten zaio eta R($\mathbf{M}$) notazioaz adierazten da, R($\mathbf{M}$) $= \{\mathbf{Mx} \ : \ \mathbf{x} \in \mathbb{R}^n\}$. Matrizearen zutabeen konbinazio lineal guztien multzoa da, $\mathbf{x}$ bektoreko osagaiak konbinazio linealeko koordenatuak izanik. Modu berean, $\mathbf{M}$-ren m errenkadek sortzen duten $\mathbb{R}^n$-ko azpiespazioari $\mathbf{M}$-ren **errenkada-espazio** esaten zaio, R($\mathbf{M}^T$).

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ matrize karratua ortogonala da $\mathbf{U}^{-1} = \mathbf{U}^T$ betetzen bada, hau da, $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ bada.

- Bektore multzo bat **ortonormala** dela esaten da bektoreak haien artean ortogonalak badira eta bektoreen norma 1 bada.

Ondoren aipatzen den teoremak ziurtatzen du $\mathbf{M} \in \mathbb{R}^{m \times n}$ matrize oro deskonposa daitekeela beste hiru matrizeren biderketa moduan. Deskonposaketa horri Balio Singularretan Deskonposatzea edo SVD esaten zaio.

II.3.1 Teorema. (SVD, Singular Value Decomposition) $\mathbf{M} \in \mathbb{R}^{m\times n}$ matrize ororen SVD deskonposaketa existitzen da,

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \qquad k = \min\{m, n\}, \tag{II.2}$$

non

- $\mathbf{U} = [\mathbf{u_1}, \ldots, \mathbf{u_m}] \in \mathbb{R}^{m\times m}$ matrizeko $\mathbf{u}_i$ zutabe-bektoreak ortonormalak diren elkarren artean. **M**-ren **ezker-bektore singularrak** dira.
- $\Sigma \in \mathbb{R}^{m\times n}$ matrizea diagonala den. Diagonaleko σ_i balioak **M** matrizearen **balio singularrak** dira, eta $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$ betetzen da, $k = \min\{m, n\}$.
- $\mathbf{V} = [\mathbf{v_1}, \ldots, \mathbf{v_n}] \in \mathbb{R}^{n\times n}$ matrizeko $\mathbf{v}_i$ zutabe-bektoreak ortonormalak diren. **M** matrizearen **eskuin-bektore singularrak** dira.

M matrizeak r balio singular baditu zeroren desberdinak, $\text{rank}(\mathbf{M}) = r$ izango da, $r \leq k$. Matrize baten balio singularrak bakarrak dira, baina bektore singularrak ez.

(II.2) formulan ikusten denez, **M** matrizea heina 1 duten r matrizeren batura moduan adieraz daiteke. r balio singularrak eta haien bektore singularrak erabiliz jatorrizko **M** matrizea lortuko da.

U eta **V** matrizeen lehenengo k zutabeak adierazteko honako notazioa erabiliko dugu: $\mathbf{U}_k = [\mathbf{u_1}, \ldots, \mathbf{u_k}] \in \mathbb{R}^{m\times k}$, $\mathbf{V}_k = [\mathbf{v_1}, \ldots, \mathbf{v_k}] \in \mathbb{R}^{n\times k}$.

Matrizearen heina r izanik, **U** matrizearen lehenengo $\{\mathbf{u_1}, \ldots, \mathbf{u_r}\}$ zutabeek **M** matrizearen zutabe-espaziorako oinarri bat osatzen dute, eta ondorioz, bektore-espazio bera sortuko dute, $\text{R}(\mathbf{U}_r) = \text{R}(\mathbf{M})$. Sortutako $\mathbb{R}^m$ bektore-espazio hori **dokumentuen espazioa** dela esaten da. Modu berean, $\text{R}(\mathbf{V}_r) = \text{R}(\mathbf{M}^T)$ betetzen denez, **V** matrizearen lehenengo $\{\mathbf{v_1}, \ldots, \mathbf{v_r}\}$ zutabeek sortutako $\mathbb{R}^n$ bektore-espazioari **terminoen espazioa** deitu ohi zaio. Azken batean, SVD deskonposaketari esker, jatorrizko **M** matrizea diagonalizatzea lortzen da, eta **U**-ko eta **V**-ko zutabeek **M**-ren zutabe-espaziorako eta errenkada-espaziorako oinarri ortonormal bana ematen dizkigute.

Termino-dokumentu matrizearen tamaina izugarri handia izan daiteke. Corpusaren arabera, ez da harritzekoa ehunka edo milaka dokumentu dituen corpus bat prozesatu behar izatea, eta ondorioz, erauzitako termino kopurua ere oso handia izatea. Matrize oso handien deskonposaketa kalkulatzeak zailtasun konputazionalak sor ditzake. Halako matrizeen balio singularrak eta bektore singularrak kalkulatzeko eta SVD deskonposaketa aurkitzeko oso

egokia da Lanczos algoritmoa. Hala ere, biribilketa erroreen pilaketagatik arazoak sor daitezkeenez, algoritmoaren inplementazio desberdinak proposatu izan dira (Golub and Loan, 1996).

II.3.1 Adibidea. Adibide honetan R softwarea erabili dugu SVD deskonposaketa kalkulatzeko. R softwareak LAPACK (Linear Algebra PACKage)[1] liburutegia erabiltzen du $\mathbf{M}$ matrize baten $\mathbf{U}$, Σ eta $\mathbf{V}$ matrizeak kalkulatzeko.

II.5. Irudian ikus daitezke 23. orriko II.2.1. Adibideko $\mathbf{M} \in \mathbb{R}^{10\times 8}$ matrizearen SVD deskonposaketako $\mathbf{U}$, Σ eta $\mathbf{V}$ matrizeak (matrizeetan azpimatrize bana nabarmenduta ageri da, geroago egingo zaien erreferentzian irakurketa errazteko asmoz). Σ matrizearen diagonal nagusian zortzi balio singularrak daude, $\sigma_1 > \sigma_2 > \cdots > \sigma_8 > 0$. Guztiak zeroren desberdinak izanik, $\text{rank}(\mathbf{M}) = 8$ da. $\Sigma \in \mathbb{R}^{10\times 8}$ matrizearen azken bi errenkadak zeroz osatuta daudenez, $\mathbf{U} \in \mathbb{R}^{10\times 10}$ matrizearen azken bi zutabeek ez dute eraginik sortuko $\mathbf{U}\Sigma\mathbf{V}^T$ biderketan. Hori dela eta, II.5. Irudian ez dira $\mathbf{U}$-ren azken bi zutabeak eta Σ-ren azken bi errenkadak erakusten.

Egiazta daiteke hiru matrizeen $\mathbf{U}\Sigma\mathbf{V}^T$ biderkadurak jatorrizko $\mathbf{M}$ matrizea ematen duela, biribilketa-errore txikiak baztertuta. Egiazta daiteke baita, $\mathbf{U}$ eta $\mathbf{V}$ matrizeak ortonormalak direla ($\mathbf{U}\mathbf{U}^T = \mathbf{I}$, $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ eta $||\mathbf{u}_i|| = 1$, $||\mathbf{v}_j|| = 1$).

$\mathbf{U}$ matrizeko zutabeen interpretazioa bereziki interesatzen zaigu. Izan ere, $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ bektoreek dokumentuen espaziorako oinarri bat osatzen dutela ikusi dugu. Ikerketa lan honetan egindako esperimentuetan dokumentu-bektoreekin aritu garenez, dokumentuen espaziorako oinarri bat finkatzeko eta dokumentuak oinarri horrekiko adierazteko beharra sortu zaigu. Ondorengo atalean ikusiko ditugu bibliografian gehien aipatu ohi diren oinarriak dokumentuen espazioan lan egiteko. Hala ere, termino-bektoreak erabiltzeko beharra izanez gero, antzeko interpretazioak egiten dira $\mathbf{V}$ matrizeko zutabeetarako, haiek terminoen espaziorako oinarri bat osatzen baitute.

II.3.1.1 Dokumentuen espaziorako oinarriak eta dokumentuen koordenatuak

Esan dugunez, $\mathbf{M}$ matrizearen heina r izanik, $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ bektoreek dokumentuen espaziorako oinarri bat osatzen dute. SVD deskonposaketa honela idatz dezakegu,

$$\mathbf{M} = \mathbf{U}_r\Sigma_r\mathbf{V}_r^T, \quad \mathbf{M} \in \mathbb{R}^{m\times n},\ \mathbf{U}_r \in \mathbb{R}^{m\times r},\ \mathbf{\Sigma}_r \in \mathbb{R}^{r\times r},\ \mathbf{V}_r \in \mathbb{R}^{n\times r}.$$

[1] http://www.netlib.org/lapack/

$$\mathbf{U} = \begin{pmatrix} -0,40 & 0,21 & -0,05 & -0,45 & -0,25 & -0,08 & 0,24 & -0,59 \\ -0,44 & 0,23 & -0,10 & -0,31 & 0,62 & -0,24 & 0,19 & 0,36 \\ -0,27 & 0,07 & -0,07 & 0,42 & -0,23 & -0,64 & -0,24 & 0,10 \\ -0,54 & 0,15 & 0,08 & 0,54 & -0,11 & 0,24 & 0,13 & -0,21 \\ -0,14 & 0,03 & 0,13 & -0,34 & -0,67 & -0,06 & 0,07 & 0,56 \\ -0,17 & -0,46 & -0,28 & 0,14 & 0,09 & -0,21 & 0,07 & 0,10 \\ -0,15 & -0,59 & -0,28 & 0,02 & -0,10 & 0,22 & 0,48 & 0,07 \\ -0,12 & -0,44 & 0,84 & -0,08 & 0,13 & -0,21 & 0,03 & -0,12 \\ -0,20 & -0,35 & -0,27 & -0,31 & -0,02 & -0,01 & -0,68 & -0,22 \\ -0,39 & 0,02 & 0,17 & 0,02 & 0,07 & 0,57 & -0,36 & 0,28 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 29,77 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 21,42 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 7,55 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 6,38 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 4,09 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 3,19 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 2,19 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,36 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} -0,49 & 0,17 & -0,15 & -0,56 & 0,46 & -0,34 & 0,09 & -0,24 \\ -0,38 & 0,10 & -0,08 & 0,62 & 0,13 & -0,49 & -0,34 & 0,27 \\ -0,58 & 0,23 & 0,08 & -0,14 & -0,75 & 0,12 & 0,01 & 0,03 \\ -0,38 & 0,17 & 0,09 & 0,25 & 0,43 & 0,71 & 0,19 & 0,16 \\ -0,21 & -0,70 & -0,09 & -0,25 & 0,01 & 0,00 & 0,08 & 0,62 \\ -0,21 & -0,48 & -0,18 & 0,06 & 0,01 & 0,27 & -0,57 & -0,54 \\ -0,08 & -0,16 & 0,96 & -0,06 & 0,10 & -0,12 & -0,13 & -0,08 \\ -0,18 & -0,36 & 0,00 & 0,38 & -0,09 & -0,19 & 0,70 & -0,40 \end{pmatrix}$$

II.5 Irudia: Adibideko $\mathbf{M}$ matrizearen SVD deskonposaketa

Matrizeko i. zutabeko $\mathbf{d}_i \in \mathbb{R}^m$ bektoreak corpuseko i. dokumentua adierazten duenez, matrizea $\mathbf{M} = [\mathbf{d}_1 \dots \mathbf{d}_n]$ moduan idatz dezakegu. Hortaz,

$$[\mathbf{d}_1 \dots \mathbf{d}_n] = \mathbf{U}_r \Sigma_r \mathbf{V}_r^T = [\mathbf{u}_1 \dots \mathbf{u}_r] \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_r^T \end{pmatrix}.$$

Aurreko adierazpena $\mathbf{d}_i$ dokumentu-bektorerako interpretatuz gero,

$$\mathbf{d}_i = v_{i1}\sigma_1\mathbf{u}_1 + v_{i2}\sigma_2\mathbf{u}_2 + \cdots + v_{ir}\sigma_r\mathbf{u}_r,$$

ikusten da, $\mathbf{d}_i$ dokumentu-bektorea $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ oinarriarekiko adierazteko koordenatuak $\sigma_1 v_{i1}, \cdots, \sigma_r v_{ir}$ direla, hau da, $\mathbf{V}_r$ matrizeko i. errenkadako osagaiak balio singularrez ponderatuak. Hala ere, SVD deskonposaketan dokumentuen espaziorako beste oinarri bat eta dokumentu-bektoreak adierazteko beste koordenatu batzuk daudela erraz ikus daiteke. Izan ere, $\sigma_1\mathbf{u}_1, \dots, \sigma_r\mathbf{u}_r$ bektore multzoa ere dokumentuen espaziorako oinarri bat da. Ez da ortonormala, baina oinarria da. Kasu honetan, $\mathbf{d}_i$ dokumentu-bektorearen koordenatuak $v_{i1}, \cdots, v_{ir}$ osagaiak dira ($\mathbf{V}_r$ matrizeko i. errenkada). II.6. Irudian SVD deskonposaketa modu desberdinetan idatzita agertzen da. Bigarren adierazpenean koordenatuen kalkulua $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ oinarria erabiltzen den kasurako ikusten da, eta hirugarrenean koordenatuen kalkulua $\{\sigma_1\mathbf{u}_1, \dots, \sigma_r\mathbf{u}_r\}$ oinarrirako. Oinarri batean nahiz bestean lan egitea erabat baliokidea da, dokumentuen arteko angeluak mantentzen direlako, eta beraz haien arteko kosinu-antzekotasuna.

II.3.2 Bektore-espazioaren dimentsioaren murrizketa: espazio semantikoa

$\mathbf{M}$ matrizearen SVD deskonposaketan oinarrituz, bere hurbilketa den $\mathbf{M}_p$ matrize bat kalkula daiteke. Horretarako, matrizearen heina r izanik, $p \leq r$ balio bat finkatu eta Σ matrizean p balio singular handienak mantenduko dira, gainerakoak zero bihurtuz. Lortuko den $\mathbf{M}_p$ matrizearen heina p izango da, eta bere zutabeek sortuko duten zutabe-espazioaren **dimentsioa** p baliora murriztu denez, **dimentsioaren murrizketa** egin dela esaten da.

$\mathbf{M}$ matrizea $\mathbf{M}_p$ matrizeaz hurbiltzean egindako errorea, errore-matrizearen Frobenius normaren bidez neur daiteke, $||\mathbf{M}-\mathbf{M}_p||_F$. Honako definizioan ikus daiteken bezala, Frobenius normaren definizioa bektoreen normaren definizioa matrizeetara orokortzetik dator.

$$(a) \qquad \begin{pmatrix} \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \vdots \\ \mathbf{d}_n^T \end{pmatrix} = \mathbf{V}_r \Sigma_r^T \mathbf{U}_r^T = \mathbf{V}_r \Sigma_r \mathbf{U}_r^T$$

$$(b) \qquad \begin{pmatrix} \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \vdots \\ \mathbf{d}_n^T \end{pmatrix} \mathbf{U}_r = \mathbf{V}_r \Sigma_r \mathbf{U}_r^T \mathbf{U}_r = \mathbf{V}_r \Sigma_r = [\sigma_1 \mathbf{v}_1 \ \sigma_2 \mathbf{v}_2 \ \ldots \ \sigma_r \mathbf{v}_r]$$

$$(c) \qquad \begin{pmatrix} \mathbf{d}_1^T \\ \mathbf{d}_2^T \\ \vdots \\ \mathbf{d}_n^T \end{pmatrix} \mathbf{U}_r \Sigma_r^{-1} = \mathbf{V}_r \Sigma_r \Sigma_r^{-1} = \mathbf{V}_r = [\mathbf{v}_1 \ \mathbf{v}_2 \ \ldots \ \mathbf{v}_r]$$

II.6 Irudia: SVD deskonposaketaren hiru idazkera. (a) $[\mathbf{d}_1 \ldots \mathbf{d}_n] = \mathbf{U}_r \Sigma_r \mathbf{V}_r^T$ berdintzaren bi atalak irauliz. (b) Aurrekoaren bi atalak $\mathbf{U}_r$ matrizeaz biderkatuz, $\mathbf{U}_r^T \mathbf{U}_r = \mathbf{I}$ izanik. (c) Aurrekoaren bi atalak Σ_r^{-1} matrizeaz biderkatuz

II.3.1 Definizioa. (Frobenius norma) $\mathbf{M} \in \mathbb{R}^{m \times n}$ matrizearen Frobenius norma honela kalkulatzen da:

$$||M||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |m_{ij}|^2}.$$

M matrizearen SVD deskonposaketako **U** eta **V** matrizeak ortogonalak

direnez, zera betetzen da:

$$||\mathbf{M}||_F = ||\mathbf{U}\Sigma\mathbf{V}^T||_F = ||\Sigma||_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2},$$

σ_i izanik $\mathbf{M}$ matrizearen balio singularrak, eta $\text{rank}(\mathbf{M}) = r$ matrizearen heina.

$\mathbf{M}$ matrizearen SVD deskonposaketan oinarrituz, bere hurbilketa den $\mathbf{M}_p$ matrizea nola kalkula daitekeen zehazten du II.3.2. Teoremak, eta ziurtatzen du horrela kalkulatutako matrize hurbildua dela heina p edo txikiagoa duten matrizeen artetik errore minimoa dutenetako bat (Eckart and Young, 1936). Gainera, errore horren eta $\mathbf{M}$ matrizearen balio singularren artean dagoen erlazioa finkatzen du.

II.3.2 Teorema. (Eckart, Young) Izan bedi $\mathbf{M} \in \mathbb{R}^{m\times n}$ matrizearen SVD deskonposaketa, $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$. Matrizearen heina r bada, $\mathbf{M}_p$ matrize hurbildu bat honela kalkula daiteke,

$$\mathbf{M}_p = \mathbf{U}_p\Sigma_p\mathbf{V}_p^T = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \qquad p \leq r, \quad r \leq k = \min(m, n),$$

non $\mathbf{U}_p$ eta $\mathbf{V}_p$ matrizeak $\mathbf{U}$ eta $\mathbf{V}$ matrizeen lehenengo p zutabeez osatutako matrizeak diren. Ondokoa betetzen da:

$$\min_{\text{rank}(A)\leq p} ||\mathbf{M} - \mathbf{A}||_F = ||\mathbf{M} - \mathbf{M}_p||_F = \sqrt{\sigma_{p+1}^2 + \cdots + \sigma_r^2}.$$

Zenbat eta handiagoa izan p-ren balioa, orduan eta antzekoagoak izango dira $\mathbf{M}$ eta $\mathbf{M}_p$. Hurbilketarekin egiten den errorea, $\sigma_{p+1}, \sigma_{p+2}, \cdots, \sigma_r$ balio singularren mende dagoela ikusten da, baztertutako balio singularren mende.

II.3.2 Adibidea. Adibideko $\mathbf{M}$ matrizearen hurbilpen bat kalkulatuko dugu. Heina $r = 8$ izanik, $p = 2$ balioa aukeratuko dugu, adibidez. Horrela, $\mathbf{M}_2$ matrize hurbilduaren kalkuluan bi balio singularrik handienak (σ_1, σ_2) dituen Σ_2 azpimatrizea eta haiei dagozkien ezker-bektore eta eskuin-bektore singularrak ($\mathbf{U}_2$, $\mathbf{V}_2$) besterik ez dira kontuan izango (ikus II.5. Irudiko azpimatrizeak).

$$\mathbf{M}_2 = \mathbf{U}_2\Sigma_2\mathbf{V}_2^T = \sum_{i=1}^{2} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T.$$

Kalkuluak eginez, heina 2 duen $\mathbf{M}_2 \in \mathbb{R}^{10\times 8}$ matrizea lortzen da (ikus II.7. Irudia). Frobenius normen kalkuluetatik,

$$||\mathbf{M}-\mathbf{M}_2||_F = 11,38, \quad ||\mathbf{M}||_F = 38,41, \quad \frac{||\mathbf{M}-\mathbf{M}_2||_F}{||\mathbf{M}||_F} = 0,2962,$$

zera ondoriozta dezakegu: $\mathbf{M}$ matrizearen eta $\mathbf{M}_2$ matrizearen artean ia % 30eko aldakuntza dago, hau da, $\mathbf{M}_2$ matrizeak jatorrizkoak jasotzen duen informazioaren % 70 inguru jasotzen du. Dimentsio-murrizketa nahiko handia izanik ere (8tik 2ra), matrizearen edukia neurri handi batean mantendu da. p balioa handituz, jatorrizko matrizera gehiago hurbilduko diren $\mathbf{M}_p$ matrizeak lortzen dira: $||\mathbf{M}-\mathbf{M}_4||_F = 5,6$, $||\mathbf{M}-\mathbf{M}_6||_F = 2,2$.

$$\mathbf{M}_2 = \begin{pmatrix} 6,60 & 4,97 & 7,94 & 5,29 & -0,65 & 0,34 & 0,23 & 0,52 \\ 7,26 & 5,47 & 8,73 & 5,82 & -0,70 & 0,39 & 0,26 & 0,58 \\ 4,19 & 3,20 & 5,01 & 3,31 & 0,64 & 0,97 & 0,40 & 0,91 \\ 8,42 & 6,43 & 10,06 & 6,66 & 1,13 & 1,83 & 0,77 & 1,74 \\ 2,15 & 1,65 & 2,57 & 1,69 & 0,43 & 0,57 & 0,23 & 0,52 \\ 0,80 & 0,94 & 0,67 & 0,25 & 7,96 & 5,79 & 1,98 & 4,46 \\ 0,04 & 0,43 & -0,32 & -0,45 & 9,78 & 7,00 & 2,38 & 5,35 \\ 0,15 & 0,42 & -0,10 & -0,24 & 7,35 & 5,27 & 1,79 & 4,04 \\ 1,64 & 1,51 & 1,73 & 0,99 & 6,50 & 4,85 & 1,68 & 3,77 \\ 5,76 & 4,45 & 6,83 & 4,48 & 2,14 & 2,23 & 0,86 & 1,94 \end{pmatrix}$$

II.7 Irudia: Adibideko $\mathbf{M}$ matrizearen hurbilpena den $\mathbf{M}_2$ matrizea

Dimentsioa p baliora murriztearen ondorioz, jatorrizko $\mathbf{M}$ matrizearen zutabeek sortutako $\mathrm{R}(\mathbf{M}) = \mathrm{R}(\mathbf{U}_r)$ bektore-espaziotik bere hurbilpena den $\mathrm{R}(\mathbf{M}_p) = \mathrm{R}(\mathbf{U}_p)$ espazioan lan egitera pasa gara. Bektore-espazio horri **dimentsio murriztuko espazioa** esaten zaio. p balioa r baino askoz txikiagoa den kasuetan, bektoreak biltegiratzeko behar den espazioa eta haien tratamenduaren kostu konputazionala asko jaistea lortzen da dimentsio murriztuko espazioan lan eginda, nahikoa izango delako p balio singular han-

dienak eta dagozkien bektore singularrak gordetzea eta erabiltzea. Balio singularrak positiboak izateak eta handienetik txikienera ordenatuta agertzeak garrantzia handia du matrize hurbilduaren kalkuluan, dimentsioaren murrizketa egitean balio singularrik handienak eta haien bektore singularrak besterik ez baitira erabiltzen.

Dimentsio murriztuko espazio horri **espazio semantikoa** ere deitu ohi zaio (Lowe, 2001). Intuitiboki esan daiteke dokumentu-bektoreak dimentsio murriztuko espaziora eramatea estuagoa den espazio batean kokatzea dela, nolabait. Estutu behar horretan, antzeko agerkidetzak[2] (co-occurrences) dituzten dokumentuek elkartze aldera jotzen dute. Horrela, hizkuntzaren aldakortasunaren ondorioz itxuraz desberdinak diruditen dokumentuak semantikoki oso antzeko bihurtzen dira eta dimentsio murriztuko espazioko "txoko" berean kokatzen dira, zarata kenduz eta ezkutuko semantika azaleratuz.

Espazio semantikoko "txoko" bakoitza koordenatu ardatzen bidez definitzen da. Ardatz-sistema bakoitza espazioko oinarri bati dagokio, eta dokumentu-bektoreak espazio horretan kokatzeko, oinarri horrekiko koordenatuak kalkulatu beharko dira. Esan dugunez, $\mathrm{R}(\mathbf{U}_p) = \mathrm{R}(\mathbf{M}_p)$, hortaz, $\mathbf{U}$ matrizeko lehenengo $\mathbf{u}_1, \ldots, \mathbf{u}_p$ zutabeek espazio semantikorako oinarri bat osatzen dute (ardatz sistema bat). $\mathbf{u}_i$ ardatz bakoitza dokumentuen semantikarekin erlazionatutako "kontzeptu" bati dagokio, eta σ_i balio singularrak kontzeptuaren indarra neurtzen du, kontzeptu horrek dokumentuen semantikan duen garrantzia. Guztiarekin, espazio semantikoko "txoko" bakoitzak semantikoki antzeko diren dokumentuak bilduko ditu.

Dimentsioa $p < r$ baliora murriztea $\sigma_1, \cdots, \sigma_p$ balio singular handienekin lan egitea denez, corpuseko i. dokumentua honela adierazita geratuko da espazio semantikoan:

$$\mathbf{d}_i' = v_{i1}\sigma_1\mathbf{u}_1 + v_{i2}\sigma_2\mathbf{u}_2 + \cdots + v_{ip}\sigma_p\mathbf{u}_p.$$

Adierazpena espazio semantikoko $\{\sigma_1\mathbf{u}_1, \cdots, \sigma_p\mathbf{u}_p\}$ oinarrirako interpretatuz gero, bektorea $v_{i1}, \ldots, v_{ip}$ koordenatuen bidez ($\mathbf{V}_p$ matrizeko i. errenkadako p osagaien bidez) adierazten dela ikusten da. Bibliografia aztertuz gero, ikusten da autore gehienek $v_{i1}, \cdots, v_{ip}$ koordenatuak erabiliz irudikatzen dituztela dokumentuak espazio semantikoan. Batzuk, ordea, osagai horiek balio singularrez biderkatu behar direla aipatzen dute, $\{\mathbf{u}_1, \ldots, \mathbf{u}_p\}$ oinarrirako koordenatuak, alegia. Pare bat kasu oso adierazgarri aipatzearren, ikus adibidez (Deerwester et al., 1989). Bertan, koordenatuak balio singularrengatik

[2]Hizkuntzalaritzan bi terminoren agerkidetza dagoela esaten da bi termino horiek corpus batean elkarrekin agertzen badira zoriz izango luketen maiztasunaz baino sarriago. Terminoen agerkidetza gertutasun semantiko moduan interpretatzen da.

biderkatu gabe eramaten dira $p = 2$ dimentsioko espaziora sinpletasunean irabazteko, baina berez balio singularrez biderkatu behar direla esaten da. Beste adibide bat (Manning and Schütze, 1999) eta (Manning et al., 2008) liburuetan aurkituko dugu. Irudi bera erabiltzen da bietan, dokumentu batzuen proiekzioa planoan irudikatzeko, baina lehenengoan koordenatuak balio singularrez biderkatu behar direla aipatzen bada ere, bigarrengoan ez da halakorik egiten. Irudian ez da zehazten dimentsio bakoitza zein oinarriri dagokion, eta horrek nahastea sor dezake.

II.3.3 Adibidea. Bektore-espazio eredua azaldu dugunean, 8 dokumentuen kosinuak kalkulatu ditugu (ikus 25. orriko II.1. Taula). Kosinuen kalkulua jatorrizko **M** matrizearen zutabeak erabiliz egina dago hor, eta beraz, bektoreek komun dituzten terminoen maiztasunean oinarritu da kalkulua.

Dimentsio-murrizketaren ondorioz espazio semantikoan lan egitera pasatzean, bektoreen antzekotasuna komun duten "kontzeptuetan" oinarrituko da. Kontzeptu horiek terminoen konbinazio linealetatik eratortzen dira; terminoen agerkidetzek eraginda sortzen dira. Jatorrizko matrizean terminoen agerkidetzak aztertuz, ez da zaila gertatzen bi kontzeptu nagusi horien definizioan terminoek jokatu duten papera zein izan den ikustea (ikus 24. orriko II.2. Irudia). t_1="adiera", t_2="desanbiguazio", t_7="berreskuratu" eta t_8="indexatu" terminoek zortzi dokumentuen bereizketa argia ematen dute; lehenengo biek WSD arloko dokumentuen eduki semantikoa jasotzen dute eta azken biek IR dokumentuena. Haiekin batera agerkidetza argian ikusten ditugu t_3="polisemia" eta t_4="hitz" terminoak (WSD dokumentuetan) eta t_6="testu" eta t_9="automatiko" (IR dokumentuetan). t_5 eta t_{10} terminoek ez dute dokumentuen bereizketan hainbesteko indarrez eragiten eta beraz, kontzeptuen definizioan izango duten eragina txikiagoa da. 2 dimentsioko espazio semantikoan dokumentuak adieraztean, agerkidetza horiek guztiak kontuan hartuak izan dira bi ardatzak definitzerakoan. Ardatzek dokumentuen arteko aldakortasun handiena erakutsiko dute horrela. II.2. Taulan ikus daitezke espazio semantikoan kalkulatutako kosinuak.

Aipatu berri dugun 25. orriko II.1. Taulan, arreta berezia eskaini diegu d_6-d_7 eta d_2-d_8 dokumentu pareei. Ikus dezagun nola aldatu den haien arteko kosinuaren kalkulua.

- d_6-d_7 dokumentu parea. Kosinua $0,39$ izatetik $0,999$ izatera pasa da, hau da, espazio semantikoan neurtuta bi dokumentu horiek semantikoki oso antzeko izatera pasa dira. Komun t_4, t_8 eta t_{10} terminoak besterik ez badituzte ere, t_8 terminoaren agerkidetzei esker d_6 dokumentua d_5, d_7 eta d_8 dokumentuen semantikara asko hurbiltzen dela

		WSD				IR			
		d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
WSD	d_1	1							
	d_2	0,997	1						
	d_3	0,999	0,992	1					
	d_4	0,996	0,987	0,999	1				
IR	d_5	-0,042	0,034	-0,086	-0,129	1			
	d_6	0,078	0,154	0,035	-0,008	0,993	1		
	d_7	0,129	0,205	0,086	0,043	0,985	0,999	1	
	d_8	0,129	0,205	0,086	0,043	0,985	0,997	1	1

II.2 Taula: Dokumentuen arteko kosinu-antzekotasunak espazio semantikoan

ondorioztatu ahal izan da. Izan ere, corpusean t_8 terminoa duten dokumentu askok t_6 eta t_9 terminoen agerkidetza maila altuan erakusten baitute, eta haietako batzuk agerkidetza altua dute $t_7$rekin, etab.

- d_2-d_8 dokumentu parea. Kosinua $0,43$ izatetik $0,205$ izatera pasa da. Kasu honetan, dokumentuek termino asko komun badituzte ere (t_3, t_4, t_6, t_9, t_{10}), terminoen agerkidetzek d_2 dokumentua d_1, d_3 eta d_4 dokumentuetatik semantikoki gertuago dagoela ondorioztatzea eragin du. t_3 eta t_4 terminoen agerkidetzek baldintzatu dute bereizketa, neurri handi batean. Izan ere, t_3 eta t_4 terminoen agerkidetza altuak dituzten dokumentuek t_1 eta t_2 terminoen agerkidetza altuak erakusten dituzte baita. d_8 dokumentua, aldiz, d_5, d_6 eta d_7 dokumentuetara gerturatu da t_6, t_7, t_8 eta t_9 terminoen eta haiekin agerkidetzan agertzen direnen eraginez, bereziki. Horrek guztiak d_2-d_8 dokumentu parea semantikoki urruntzea eragin du.

Espazio semantikoan lan egiteak Hizkuntzaren Prozesamenduan klasikoak diren bi problemei aurre egiten laguntzen du: sinonimiaren eta polisemiaren arazoa.

- Sinonimia. “amama” eta “amona” bi hitz sinonimo izanik, nahiz eta termino desberdinak izan eta matrizean adierazpen desberdina ego-

kitu, semantikoki oso antzeko diren dokumentuetan maiztasun altuz agertuko direnez, “amama” edo “amona” hitzak dituzten dokumentuak espazio semantikoko “txoko” berean multzokatuak agertuko dira, terminoen agerkidetzei esker. Ondorioz, erabiltzaileak kontsulta bat egitean, dela “amama” hitzarekin dela “amona” hitzarekin, bietariko edozein termino duten dokumentuak semantikoki antzeko moduan agertuko zaizkio erabiltzaileari; “amama” terminoarekin egindako bilaketa-kontsulta bati erantzunez “amona” terminoa duen eta “amama” ez duen dokumentu bat berreskuratzea posible gertatzen da.

- Polisemia. Polisemikoa den terminoa duten dokumentuak espazio semantikoan ez dira txoko berean adieraziak izango. Adibidez, “arte” terminoa polisemikoa izanik, semantikoki oso desberdintzat joko dira “arte”, “haritz”, “pago”, “adar” etab. moduko terminoak dituzten dokumentuak “arte”, “pintura”, “eskultura”, etab. dituztenetatik, eta erabiltzailearen interesekoak direnak ekartzeko unean, dokumentu gehiagorekin asmatuko da.

SVD bidezko dimentsio-murrizketa Hizkuntzalaritza Konputazionalerako aplikatzeko asmoz sortu zen LSI, dokumentuen ezkutuko (latent) semantika azaleratzeko gaitasuna duela ikusi baitzen. Ondorengo atalean LSIren sorrera aipatu eta Informazio-Berreskuratze atazarako nola erabil daitekeen azalduko dugu.

II.3.3 Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) 1989an izan zen patentatua (US Patent 4,839,853)(Deerwester et al., 1989). Testuen ezkutuko semantika jasotzeko gaitasuna duen tresna moduan aurkeztu zuten sortzaileek, eta Informazio-Berreskuratze alorrean erabiltzeko aproposa dela baieztatu zuten. Harrez geroztik, ikerlariek hizkuntzaren prozesamendurako ataza desberdinetarako esperimentuak egin izan dituzte, eta emaitza onak ematen dituela ikusi da behin eta berriz. LSIk bere mugak ditu eta corpusa handiegia den kasuetan arazoak sortzen dira (Chen et al., 2001).

Tresna bi izenez ezagutzen da ematen zaion erabileraren arabera: 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.

LSI erabiltzen hasteko, corpus bat behar da. Bertako dokumentuetan eta handik erauzitako terminoetan oinarrituz eraikiko du LSIk termino-dokumentu matrizea eta SVD deskonposaketa kalkulatuz p dimentsioko espazio semantikoa sortuko du. Erabiltzaileak kontsulta bat egitean, kontsultaren $\mathbf{q}$ bektore-adierazpena sortu behar da: q_i osagaiak t_i terminoak kontsultan duen agerpen maiztasuna adieraziko du, $i = 1, \ldots, m$. Ondoren, kontsultaren koordenatuak kalkulatuko dira espazio semantikoan.

$$\mathbf{q}_p = \mathbf{q}^T \mathbf{U}_p \Sigma_p^{-1}.$$

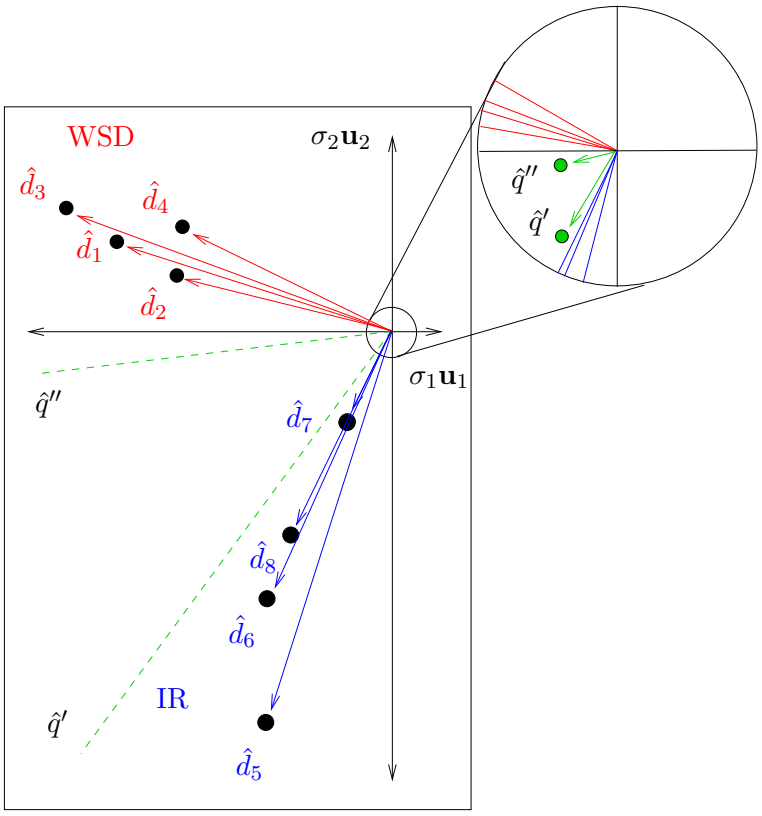
Kontsultaren koordenatu horiek $\{\sigma_1 \mathbf{u}_1, \ldots, \sigma_p \mathbf{u}_p\}$ oinarriari dagozkio. Oinarri horretan lan egin beharrean $\{\mathbf{u}_1, \ldots, \mathbf{u}_p\}$ oinarrian ari bagara, kontsultaren koordenatuak horrela kalkulatuko ditugu:

$$\mathbf{q}_p = \mathbf{q}^T \mathbf{U}_p.$$

Kontsulta eta corpuseko dokumentuak espazio semantikoan koordenatuen bidez adieraziak izan direnean, haien arteko kosinu-antzekotasunak neurtuko dira.

Erabiltzaileak erabaki beharko du zein p dimentsiora murriztu nahi duen jatorrizko bektore-espazioa. p-ren aukeraketa oso garrantzitsua da, LSIk 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.

II.3.4 Adibidea. Ikus dezagun, bada, LSI tresna erabiliz nola egiten den lan dimentsio murriztuko espazioan. Jatorrizko $\mathbf{M}$ matrizea II.2.1. adibidean dugu (24. orrian), SVD deskonposaketa II.3.1. adibidean (37. orrian) eta $\mathbf{M}_2$ matrize hurbildua II.3.2. adibidean (41. orrian). $p = 2$ dimentsioko espazio murriztuan egingo dugu lan. Σ matrizeko lehenengo bi balio singularrak gainerakoak baino dezente handiagoak izateak erabaki hau justifikatzen du ($\sigma_1 = 29,77$, $\sigma_2 = 21,42$, $\sigma_i < 8, i = 3, \ldots, 8$). Esan dugunez, espazio semantikoko ardatz bakoitza dokumentuen semantikarekin erlazionatutako "kontzeptu" bati dagokio, eta balio singularrek kontzeptuen indarra neurtzen dute. Bi balio singular horiek gailendu egiten dira, corpuseko zortzi dokumentuen semantikan "nagusi" diren bi kontzeptu existitzen direla erakutsiz. Indar berezia dutenez, dokumentuen arteko konparaketa semantikoak kontzeptu horietan oinarritzea komeni da. Hortaz, 2 dimentsioko espazio semantikoan dokumentuek duten adierazpenean oinarrituko ditugu konparaketa semantikoak, gainerako dimentsioak (kontzeptuak) baztertuz.



II.8 Irudia: Dokumentuak eta kontsultak 2 dimentsioko espazio semantikoan

II.8. Irudian corpuseko 8 dokumentuak gezien bidez adieraziak agertzen dira. Espazio semantikoaren ardatzak $\{\sigma_1\mathbf{u}_1, \sigma_2\mathbf{u}_2\}$ dira, eta dokumentuen arteko banaketa oso argi ikusten da irudian: lehenengo lau dokumentuak (WSD) espazioko zonalde batean multzokatuak agertzen dira, azken lauetatik (IR) ondo berezituak. Corpuseko dokumentuen koordenatuak honela kalkulatzen dira. d_1 dokumenturako adibidez:

$$\hat{\mathbf{d}}_1 = \mathbf{d}_1^T \mathbf{U}_2 \Sigma_2^{-1} = (-0,49,\ 0,17).$$

Dena den, dokumentuen koordenatuak $\mathbf{V}_2$ matrizeko errenkadetan emanak datozela esan dugu. Corpuseko lehenengo dokumentua $\{\sigma_1\mathbf{u}_1, \sigma_2\mathbf{u}_2\}$ oinarriko bektoreen konbinazio lineal moduan idazteko behar ditugun koordenatuak dira kalkulatu ditugunak. Horrela, $\mathbf{d}_1' = -0,49\sigma_1\mathbf{u}_1 + 0,17\sigma_2\mathbf{u}_2$ bektorea $\mathbf{M}_2$ matrizeko lehenengo zutabea dela egiazta daiteke.

Azter dezagun LSIren portaera erabiltzaile batek egindako kontsulta baten aurrean. Demagun Informazio-Berreskuratzea (IR) modu automatikoan egiteko dauden sistemei buruzko informazioa aurkitu nahi duela erabiltzaileak. Bilaketak egitean, testuen semantikaren tratamendu egokia egiteak duen garrantziaz jakitun, t_3, t_6, t_7, t_9 eta t_{10} terminoak dituen kontsulta egin du, hau da, "testu, berreskuratu, automatiko, semantika, polisemia".

$$\mathbf{q}'^T = (0, 0, 1, 0, 0, 1, 1, 0, 1, 1).$$

M matrizea aztertuz ikusten da kontsultako t_7="berreskuratu" terminoak IR kutsu garbia duela, d_6 eta d_8 dokumentuetan ageri da, eta dokumentu horietan agerkidetzan kontsultako beste termino batzuk, t_6 eta t_9 esaterako. Kontsultako t_3="polisemia" eta t_{10}="semantika" terminoek nahaste pixka bat ekartzen dute, WSD arloko dokumentuetan maiztasun handiagoz agertzen direlako. Kontsulta espazio semantikora eramanez gero, honako koordenatuen bidez adieraziko da:

$$\hat{\mathbf{q}}' = \mathbf{q}'^T \mathbf{U}_2 \Sigma_2^{-1} = (-0,04, -0,06).$$

Kontsultaren koordenatuak irudikatuz gero, koordenatu sistemaren jatorri puntutik oso gertu dagoen puntu bat lortzen da, kontsultan terminoek duten agerpen maiztasuna dokumentuetan dutena baino askoz txikiagoa delako (ikus II.8. Irudia). Kontsulta hobeto ikus dadin, koordenatuek finkatutako puntutik pasatzen den zuzenaren bidez adieraztea erabaki dugu. Azken batean, kosinua bektoreen arteko angeluan oinarritzen denez, antzekotasun semantikoa ez da aldatuko. Irudian ikusten den bezala, $\hat{\mathbf{q}}'$ bektoreak IR arloko dokumentuen aldeko joera erakusten du, eta horiek izango dira erabiltzaileari itzuliko zaizkionak: d_7, d_8, d_6, ... kosinuaren arabera ordenatuta.

Bilaketa pixka bat zailagoa egiten saia gaitezke. Demagun orain erabiltzaileak Hitzen Adiera Desanbiguazioari (WSD) buruzko dokumentuak aurkitu nahi dituela. Desanbiguazioa modu automatikoan egiteko dauden sistemen berri jakin nahi duenez, t_4, t_9 eta t_{10} terminoak dituen kontsulta egitea erabaki du, hau da, "hitz, automatiko, semantika".

$$\mathbf{q}''^T = (0, 0, 0, 1, 0, 0, 0, 0, 1, 1).$$

Kasu honetan esan dezakegu t_4 eta t_{10} terminoek WSD aldera egiteko joera dutela eta t_9 terminoak aldiz IRrako joera, nahiz eta hiru terminoak agertzen diren bai WSDko dokumentuetan eta baita IRkoetan ere. Interpretazio hauek guztiak $\mathbf{M}_2$ matrizea erabiliz ere egin daitezke. Espazio semantikoan kontsulta non kokatua izango den jakiteko, koordenatuak kalkulatu beharko dira:

$$\hat{\mathbf{q}}'' = \mathbf{q}''^T \mathbf{U}_2 \Sigma_2^{-1} = (-0,04, -0,01).$$

Dimentsio-murrizketaren ondorioz, eta terminoen agerkidetzek eraginda, $\hat{\mathbf{q}}''$ kontsulta WSD dokumentuetara gerturatzen dela ikusten da, eta ondorioz erabiltzaileari d_2, d_1, d_3, ... dokumentuak itzuliko zaizkio kontsultaren emaitza moduan.

II.4 Ondorioak

Kapitulu honetan LSIren oinarri matematikoa modu xume eta ulergarrian azaltzen saiatu gara. Irakurleari oinarri matematikoari buruz gehiago jakiteko gogoa piztu bazaio erraz aurkituko du sakontzeko aukera kapituluan zehar emandako erreferentzia bibliografikoetan. Egia esan, SVD deskonposaketak badu bere xarma. Gilbert Strangek Aljebra Linealari buruzko bere liburuan halaxe dio: "I will give you my opinion directly. The SVD is the climax of this linear algebra course". LSI tresnak oinarri matematiko dotorea duenik ezin da ukatu!

LSIren oinarri matematikoak sor lezaken liluraz gain, haren praktikotasuna aipatu behar da. Esan liteke tresnari corpus bat duen fitxategi bat eta hainbat kontsulta dituen beste fitxategi bat ematea nahikoa direla, berak egin beharreko kalkulu guztiak egin eta kontsulten erantzun moduan dokumentuen zerrenda bat erabiltzaileari itzultzeko. Dena den, ikerketa lan honetan LSI ez dugu horrela erabili. Guretzat corpuseko dokumentuen eta erabiltzailearen kontsulten bektoreak lortzeko tresna bat izan da LSI. Behin bektoreak lortuta, Ikasketa Automatikoko sailkatze-ereduak erabiliz hainbat esperimentu egin ditugu.

III. KAPITULUA

Ikasketa automatikoa. Gainbegiratutako sailkatzea

III.1 Sarrera

Ikasketa Automatikoa (Mitchell, 1997) Adimen Artifizialaren adar bat da. Haren helburu nagusia konputagailuek modu egoki eta automatikoan ikastea da. Horretarako, hainbat metodo garatu izan dira eta, horiei esker, konputagailua gai izango da kalkuluak egiteko ahalmenaz haratago doazen trebeziak eskatzen dituzten problemak ebazteko: gaixotasunak diagnostikatzeko, datozen egunetan egingo duen eguraldiaren iragarpena egiteko, etab. Azken batean, metodo horien bidez ezagutza lortu nahi da, egoera berri baten aurrean dagoen profesionalari (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.

Hizkuntzaren Prozesamenduaren esparruan ere, Hizkuntzalaritza Konputazionalean, Ikasketa Automatikoko metodoak oso lagungarriak izaten ari dira azken urteotan. Aplikazio-esparru klasiko bat dokumentuen sailkatze automatikoarena izan da: testu idatziak aldez aurretik emandako hainbat kategoriatan sailkatzearena, alegia. Halakoetan, sekula ikusi gabeko testu berri bat emanik, zein kategoriakoa den erabaki behar du sistemak: kirola, politika, ekonomia etab. Hizkuntzaren Prozesamenduaren beste hainbat esparrutara ere zabaldu da Ikasketa Automatikoko metodoen erabilpena: Informazio-Berreskuratzea, Itzulpen Automatikoa, Hitzen Adiera-Desanbiguatzea, Korreferentzia-Ebaztea, Galdera-Erantzun sistemak etab.

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.

III.2 Gainbegiratutako Sailkatzea

Gainbegiratutako Sailkatzearen (Supervised Learning) barruan kokatzen diren metodoek, oro har, aldez aurretik sailkatuak izan diren kasuetatik abiatuz eredu matematiko bat sortzen dute. Eredu horrekin kasu partikularrek erakusten duten portaera orokortzea lortu nahi izaten da. Arrazonatzeko modu horri *indukzioa* esaten zaio, eta induzitutako eredua gerora etor daitezkeen kasu berriei buruzko iragarpenak egiteko erabiltzen da.

Induzitutako eredu matematikoak arlo jakin bati buruzko ezagutza jaso beharko luke, eta horretarako oso garrantzitsua da aztertu nahi den arloa ondo adieraziko duten instantziez osatutako datu-multzoa izatea. Gainbegiratutako Sailkatzerako tekniken oinarri den datu-multzoaren egitura III.1. Taulan agertzen da. m kasu (instantzia, adibide) dituen datu-multzo bat izanik, instantzia bakoitzari buruzko informazioa hainbat aldagai iragarleren edo ezaugarriren bidez emana dator, X_1, ..., X_n. C klase-aldagaiaren $C = \{y_1, \ldots, y_e\}$ balio posible bakoitzari *etiketa* edo *klasea* esaten zaio. Entrenamendu edo ikasketa fasean, $(\mathbf{x}_i, c_i)$ instantzia etiketatuetatik induzitzen da eredua. c_i bakoitza C multzoko etiketa bat da, $c_i \in C$, eta $\mathbf{x}_i$ instantzia zein klasekoa den adierazten du. Test fasean, eredua erabiliko da $\mathbf{x} = (x_1, \ldots, x_n)$ kasu berria sailkatzeko, hau da, kasu berriari C multzoko etiketa bat esleitzeko; prozesu horri *dedukzioa* esaten zaio.

Gainbegiratutako Sailkatzerako erabili ohi diren oinarrizko sailkatzaile batzuen deskribapen laburra dator ondoren. Gehienak oso ezagunak direnez, ez da haiei buruzko deskribapen sakonik ematen, baina aipatzeak merezi du, guztiak izan baitira erabiliak ikerketa-lan honetan. Batzuetan, Weka paketean inplementatutako sailkatzaileen bertsioak (Hall et al., 2009) edota SNoW paketea (Carlson et al., 1999) erabili baditugu ere, esperimentazio fasean erabili dugun metodologia nagusia guk asmatua eta inplementatua izan da.

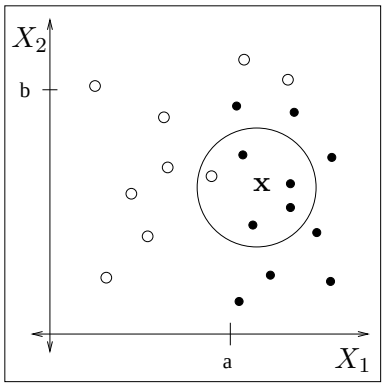
- k-nearest neighbors (k-NN).

  Distantzian oinarritutako sailkatzaile bat da k-NN algoritmoa. Kasu

Datu-multzoa	X_1	...	X_j	...	X_n	C
$(\mathbf{x}_1, c_1)$	x_{11}	...	x_{1j}	...	x_{1n}	c_1
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$(\mathbf{x}_i, c_i)$	x_{i1}	...	x_{ij}	...	x_{in}	c_i
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$(\mathbf{x}_m, c_m)$	x_{m1}	...	x_{mj}	...	x_{mn}	c_m
$\mathbf{x}$	x_1	...	x_j	...	x_n	?

III.1 Taula: Gainbegiratutako sailkatzerako datu-multzoaren egitura.

berri baten C klase-aldagairako iragarpena, entrenamendurako datuen multzoko kasuen artean hurbilen dauden k kasuak aurkituz eta haien klasea aztertuz egiten da (Dasarathy, 1991). Normalean, auzokide diren kasuetan sarrien agertzen den etiketa izaten da kasu berriari esleitzen zaiona.



III.1 Irudia: 5-NN algoritmoaren adierazpen grafikoa

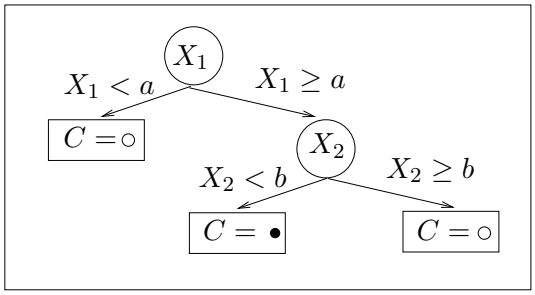
III.1. Irudian X_1 eta X_2 bi aldagai iragarleren bidez deskribatutako instantzien adierazpen grafikoa ikus daiteke. Entrenamendurako datu-multzoko 20 kasuak bi klasetakoak dira: 9 kasuk $\circ$ etiketa dute, eta gainerako 11 kasuek $\bullet$ etiketa. $\mathbf{x}$ kasu berria sailkatzeko, datu-multzoko instantzia guztietara duen distantzia kalkulatu eta 5 auzokide

hurbilenen artean gehiengoa • klasekoa denez, • klasea esleituko zaio kasu berriari.

k-NN algoritmoak ez du eredurik induzitzen; distantziak kalkulatu behar dira kasu berri bat agertzen den aldiro. Distantziak kalkulatzeko metrika desberdinak erabil daitezke. Bestalde, k parametroaren balioa ere finkatu egin behar da.

- Sailkatze-Zuhaitzak (C4.5, REPTree).

  Sailkatze-Zuhaitzak erpinez, ertzez eta hostoez osatutako sailkatze--ereduak dira. Entrenamendurako datu-multzotik abiatuz induzitzen dira, C4.5 eta REPTree moduko algoritmoak erabiliz (Quinlan, 1993), (Frank and Witten, 1998). Zuhaitzeko erpinak aldagai iragarleei dagozkie, eta hostoetan klasearen balioak kokatzen dira. Entrenamendu fasean, aldagai iragarleek klasearen iragarpenerako ematen duten informazioa neurtzen da (informazio-irabazia edo irabazi-ratioa, adibidez), eta balio altueneko aldagaia kokatzen da aldiro erpinean. Horrek datu-multzoko kasuen partizio bat sortzen du, adar bakoitzari dagokion datuen azpimultzoarekin prozesua errepikatuko delarik. Adarretik datorren datuen azpimultzoko instantzien etiketaren arabera erabakitzen da hostoan kokatuko den etiketa.



III.2 Irudia: Sailkatze-Zuhaitz bat III.1. Irudiko adibiderako

III.2. Irudian III.1. Irudiko datu-multzotik abiatuz induzitutako Sailkatze-Zuhaitz baten adibidea agertzen da. Test fasean, kasu berriaren aldagai iragarleen balioen arabera adarrak zeharkatuko dira, hosto batera iritsi arte. Hostoan agertzen den etiketa esleituko zaio kasu berriari.

Sailkatze-Zuhaitzen indukzioan, entrenamendurako datuetara gehiegi egokitzearen arazoa (overfitting) oso ohikoa izaten da. Gehiegi ego-

kitzea gertatzen dela esaten da ikasketa fasean induzitutako eredua entrenamenduko datu-multzoa sailkatzeko bereziki ona denean, baina datu horietara gehiegi egokitu izanagatik orokortze-ahalmen eskasekoa bada. Arazo horrek test fasean zuhaitzak hain emaitza onak ez ematea eragiten du. Hori ekiditeko hainbat inausketa estrategia definitu izan dira.

- Naive Bayes (NB).

  Probabilitatean oinarritutako sailkatzaile bat da Naive Bayes, abiapuntu moduan Bayes-en teorema hartuta sortutako sailkatzaileen familia bateko sailkatzailerik sinpleena (Duda and Hart, 1973). Bayes-en teorema erabiliz $\mathbf{x} = (x_1, \ldots, x_n)$ kasu berriak y_j klasekoa izateko duen $p(y_j|\mathbf{x})$ probabilitatea horrela kalkulatzen da:

$$p(y_j|\mathbf{x}) = \frac{p(y_j)p(\mathbf{x}|y_j)}{\sum_{j=1}^{e} p(y_j)p(\mathbf{x}|y_j)}$$

  $C = \{y_1, \ldots, y_e\}$ izanik klase-aldagairako dauden e etiketa posibleak, estimatu beharreko probabilitate kopurua izugarri handia izan daiteke datu-multzoaren tamainaren, aldagai iragarleen kopuruaren eta aldagaiek har ditzaketen balio posibleen arabera. Hori dela eta, klasea emanik aldagai iragarleak haien artean independenteak direla suposatzen da. Hipotesi horri esker, sinplea eta eraginkorra den Naive Bayes sailkatzailea lortzen da:

$$p(y_j|x_1, \ldots, x_n) \propto p(y_j) \prod_{i=1}^{n} p(x_i|y_j)$$

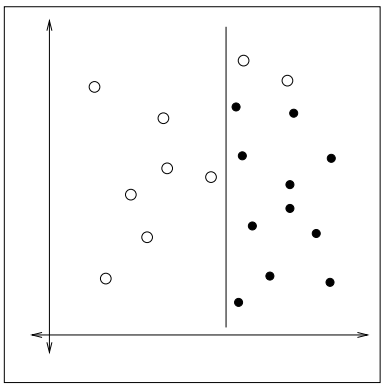
  y_j etiketa guztien artetik balio handiena lortzen duena izango da kasu berriari esleituko zaiona. Aurreko adibidearekin jarraituz,

$$p(\bullet|x_1, x_2) \propto p(\bullet)p(x_1|\bullet)p(x_2|\bullet) \quad \text{eta} \quad p(\circ|x_1, x_2) \propto p(\circ)p(x_1|\circ)p(x_2|\circ)$$

  balioen artetik handiena duen $\bullet$ edo $\circ$ etiketa esleituko zaio $\mathbf{x} = (x_1, x_2)$ kasuari. Aldagai iragarleen arteko independentziaren hipotesia oso eztabaidatua izan den arren, Naive Bayes sailkatzaileak oso emaitza onak ematen ditu.

- Support Vector Machines (SVM).

Bi klase desberdineko kasuez osatutako entrenamendurako datu-multzo batetik abiatuz, SVM algoritmoaren bidez klase arteko bereizketa (marjina) maximizatzen duen hiperplanoa kalkulatzen da (Boser et al., 1992). Entrenamenduko kasuak modu optimoan banangarriak diren espazio batean adierazten dira. Entrenamenduko kasuak adierazi diren espazio berean adieraziko dira test kasuak ere, eta hiperplanoaren zein aldetan proiektatzen denaren arabera, kasuari etiketa bat ala bestea egokituko zaio. Bertsiorik sinpleenean, hiperplano hori funtzio lineal bat izango da.



III.3 Irudia: SVM lineal baten adibidea

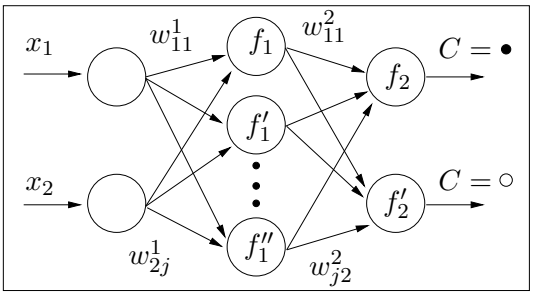
III.3. Irudian adibiderako SVM lineal bat agertzen da. Lineala izanik ezinezkoa gertatzen da errorerik gabeko bereizketa egitea adibide horretan. Azken hamarkadetan sailkatze-problemetarako oso erabiliak izaten ari dira SVMak, oso emaitza onak ematen dituztelako hainbat atazatan. Ikerketa-lan honetan Weka paketeko SMO inplementazioa erabili dugu (Platt, 1999).

- Neurona-Sare Artifizialak (Multilayer Perceptron, MLP).

  Neurona-Sare biologikoen funtzionamendua simulatuz sortutako eredu konputazionalak dira Neurona-Sare Artifizialak. Multilayer Perceptron edo MLP sailkatze-problemetan asko erabiltzen den Neurona-Sare Artifizial bat da (Rumelhart et al., 1988). Oro har, hainbat geruzatan eta elkarri konektatuta dauden erpinez (neuronez) osatutako sistemak dira eta erpinen artean informazio-trukea gertatzen da. Erpinen arteko konexioa zenbakizko ponderazioen bidez adierazten da, eta ponderazio horiek eguneratuz sistemak ikasteko gaitasuna lortzen du. Erpinek

sarrera moduan jasotzen dituzten balioak funtzioen bidez eraldatzen dira, eta irteerako konexioetatik hurrengo geruzan dauden erpinetara pasatzen dira. Horrela, geruzaz geruza erpinak aktibatuz eta balio berriak kalkulatuz doaz. Irteera-geruzan aktibatzen den erpinaren arabera erabakitzen da kasu berriari esleituko zaion klasea.

MLP algoritmoa Perceptron izenez ezagutzen den sailkatzaile linealaren aldakuntza moduan sortu zen. Hura ez bezala, MLP algoritmoa gai da linealki banangarriak ez diren datu-multzoak ondo sailkatzeko, erpinei egokitutako funtzio horietako batzuk linealak ez direlako.



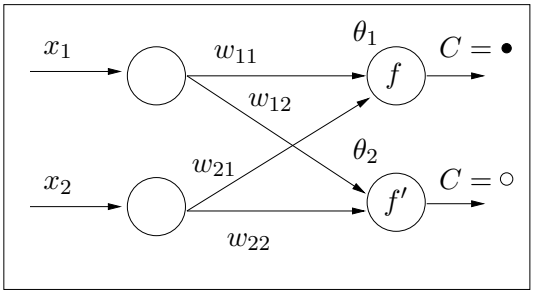
III.4 Irudia: MLP baten adibidea

III.4. Irudian adibiderako MLP bat agertzen da. Neurona-Sare Artifizialeko sarrera-erpinek $\mathbf{x}$ kasu berria deskribatzen duten x_1 eta x_2 balioak jaso eta erpinetako funtzioen arabera w^k_{ij} ponderazioak eguneratuko dira. Azken geruzara iristean, $\bullet$ klaseari edo $\circ$ klaseari dagokion irteera-erpina aktibatuko da, eta horren arabera etiketa bat ala bestea esleituko zaio kasu berriari.

- Winnow algoritmoa.

  Winnow algoritmoa Perceptron algoritmoaren oso antzekoa den beste sailkatzaile lineal bat da. Sarrerako erpinetan $\mathbf{x}$ kasu berria deskribatzen duten $X_1, \ldots, X_n$ ezaugarrietarako balioak jasotzen dira. Erpin horiek $C = \{y_1, \ldots, y_e\}$ etiketetarako dauden irteera-erpinetara konektatuta daude. Sarrerako erpinen eta irteerakoen arteko loturak w_{it} ponderazioen bidez emanak datoz, $i = 1, \ldots, n$, $t = 1, \ldots, e$. Ponderazio horiek eguneratuz doaz, baina Perceptron algoritmoan ez bezala, Winnow algoritmoan ponderazioen eguneraketa biderketa-erregela batean oinarritzen da. Horrek aukera ematen dio Winnow algoritmoari

esanguratsu ez diren aldagai iragarleak kontuan ez hartzeko (desaktibatzeko).



III.5 Irudia: Winnow baten adibidea

Algoritmoak hainbat parametro erabiltzen ditu: t irteera-erpin bakoitzak θ_t parametro bat du esleituta. Atalase bat da eta horren arabera erabakiko da instantzia bat irteera-erpin horri dagokion klasekoa den ala ez. Gainera, eguneratuz doazen $\alpha_t > 1$ igoerarako parametroa eta $0 < \beta_t < 1$ jaitsierarako parametroak daude. Instantzia baten iragarpena okerra izan den bakoitzean, irteerako t erpinera iristen diren eta aktibo dauden ezaugarriei dagozkien w_{it} ponderazioak eguneratzeko erabiltzen dira. Test fasean, ponderazio horiek $\mathbf{x}$ kasu berriaren iragarpena egiteko erabiltzen dira. III.5. Irudian ikus daiteke adibiderako Winnow algoritmoaren adierazpen grafikoa.

Winnow algoritmoa SNoW (Sparse Network of Winnows) ikasketa-arkitekturaren barne dago (Carlson et al., 1999) eta Hizkuntzaren Prozesamenduko hainbat atazatan emaitza onak eman ditu.

- Voting Feature Intervals (VFI)

  Aldagai iragarleen edo ezaugarrien balio tarteen kalkuluan oinarritzen den sailkatze-algoritmoa da VFI (Demiröz and Güvenir, 1997). Tarte horien kalkulua modu independentean egiten da, entrenamendurako datu-multzoko instantziek ezaugarrietan dituzten balioak erabiliz. Klase bakoitzerako eta ezaugarri bakoitzerako balio erreal bat kalkulatzen da eta sailkatze-prozesuan ezaugarri guztiek hartzen dute parte. Bozketa bidez balio altuena lortzen duen klasea da kasu berriari egokitzen zaiona.

Ikusi dugunez, sailkatze-problema baten aurrean erabil daitezkeen oinarrizko sailkatzaileak asko dira, eta oso izaera desberdinekoak gainera. Sailka-

tze-problema eta arlo guztietarako onena den sailkatzailerik ez da existitzen. Hori dela eta, sailkatzaile desberdinen arteko konbinaketa egitea proposatu izan da, konbinaketari esker sailkatzaile batek egindako akatsak beste sailkatzaile batzuek egindako iragarpen egokiekin zuzentzea posible izango delako, kasu batzuetan.

III.3 Sailkatzaileen konbinaketa: multi-sailkatzaileak

Problema jakin baten aurrean erabaki bat hartu behar denean, erabaki hori garrantzia handikoa denean bereziki, aditu multzo bat elkartu ohi da guztien iritzia kontuan hartzeko. Ideia horretan oinarrituz sortu dira multi-sailkatzaileak, zenbait sailkatzaile entrenatu eta haien artean onena aukeratu ordez haien iragarpenak konbinatzeko (Ho et al., 1994). Oinarrizko sailkatzaileekin lortutako emaitzak hobetzen dira, gehienetan, multi-sailkatzaileekin. Ikerketa-lan honetan erabili diren multi-sailkatzaileen deskribapen laburra dator ondoren.

- Naive Bayes Tree (NB-Tree)

  Sailkatzaile hibrido bat da, haren zatiak beste sailkatzaileez osatuta daudelako. Entrenamendu fasean induzitzen den eredua hostoetan Naive Bayes sailkatzaileak dituen Sailkatze-Zuhaitza da (Kohavi, 1996). Hortaz, zati batzuetan Sailkatze-Zuhaitza da eta besteetan Naive Bayes. Test fasean, kasu berriaren sailkatzea zuhaitzeko adarrak zeharkatzearekin hasten da. Hosto batera iristean, hostoari dagokion entrenamendurako datuen azpimultzoan oinarrituz egiten dira probabilitateen estimazioak; Naive Bayes sailkatzailea modu lokalean aplikatuz, alegia.

- Bagging (Bootstrap AGGregatING)

  Entrenamendurako datuen multzotik hainbat Bootstrap lagin desberdin sortuz posible gertatzen da hainbat sailkatzaile desberdin induzitzea, guztiak izaera berekoak izanagatik ere haien artean desberdinak izango direnak. Ideia horretan oinarritzen dira Bagging bidez sortutako multi-sailkatzaileak (Breiman, 1996). Oinarrizko sailkatzaileak modu independentean induzitzen dira laginetatik, haien arteko elkarrekintzarik gabe.

  Bagging moduan konbinatutako oinarrizko sailkatzailearen arabera multi-sailkatzaile desberdinak sor daitezke. Ikerketa-lan honetan k-NN sailkatzaileez osatutako multi-sailkatzailea erabili dugu. Hartu

behar izan ditugun erabakiak eta inplementatu ditugun estrategiak, funtsean, honakoak dira:

- Lagin kopurua ($S_1, \ldots, S_s$ sailkatzaile kopurua). Erabaki behar da entrenamendurako datu-multzotik zenbat lagin sortuko diren. S_i sailkatzaile bakoitza lagin horietako batean oinarrituko da klase-iragarpena egiteko, $i = 1, \ldots, s$. Gure kasuan, sailkatzailea k-NN izanik, k auzokide hurbilenak kalkulatzeko erabiliko dira laginak.

- Laginketa mota eta tamaina. Bagging multi-sailkatzaileetan Bootstrap metodoaren bidez sortzen dira laginak, hau da entrenamenduko datu-multzoak duen adina instantzia zoriz eta birjarpenarekin aukeratuz. Hori horrela izanik, instantzia batzuk behin baino gehiagotan ager daitezke lagin batean eta beste batzuk, aldiz, ez dira agertuko. Horrela sortutako laginetan ez da bermatzen jatorrizko datu-multzoko klase-banaketa mantenduko denik.

  Guk sortutako multi-sailkatzaileak azpilaginak sortzen ditu, hau da, laginak jatorrizko datu-multzoa baino txikiagoak dira (Random subsampling). azpilaginen tamaina (datu-multzo originalaren proportzioa) aukeratu beharreko parametro bat izan da. Instantzia-kopuru txikiko klaseetarako gutxienez 2 instantzia egongo direla bermatu dugu. Bestalde, azpilaginak estratifikatuak dira, hau da, jatorrizko datu-multzoko klase-banaketa mantenduko dela bermatu dugu.

- Antzekotasun neurria. Kosinu antzekotasuna erabili dugu k-NN algoritmoaren bidez k auzokide hurbilenak aurkitzeko. Izan ere, multi-sailkatzailea erabili dugu LSI aplikatuz lortutako bektoreak sailkatzeko, eta LSIk antzekotasun semantikoak neurtzeko erabiltzen duen neurria kosinua izanik, neurri bera erabiltzea erabaki dugu.

- Entrenamendu fasea. k-NN sailkatzaileak ez du berez entrenamendu faserik behar, ez delako ereduaren indukziorik gertatzen; alde horretatik, nahiko sailkatzaile berezia da. Hala ere, algoritmoaren aldaera desberdinak proposatu izan dira, eta guk inplementatu dugun bertsioak badu ikasketa-fase bat. Entrenamendurako datu-multzoan oinarrituz, konfiantza-balio (confidence value) batzuk kalkulatzen ditugu S_i sailkatzaile eta $C = \{y_1, \ldots, y_e\}$ klase guztietarako, $i = 1, \ldots, s$. Balio horiek entrenamendu fa-

sean sailkatzaile bakoitzak klase bakoitzaren iragarpenean erakutsi duen trebetasuna adierazten dute.

- Test fasea. Instantzia berri bat iristean, S_i sailkatzaile bakoitzak dagokion laginean oinarrituz kasu berriaren auzokide hurbilenak kalkulatu, eta haien klasea aztertuz a_i eta dagokion cv_{a_i} konfiantza-balioa itzuliko ditu, $a_i \in C = \{y_1, \ldots, y_e\}$ klase bat izanik. Konfiantza-balioaren arabera, sailkatzaileak iragarpena indar handiagoz ala txikiagoz egiten duela esan daiteke.

- Bozketa. $S_1, \ldots, S_s$ sailkatzaileek egindako iragarpenak konbinatzeko *Bozketa Bayestarra* erabili dugu (Dietterich, 1998). $a_1, \ldots, a_s$ klaseak eta $\mathrm{cv}_{a_1}, \ldots, \mathrm{cv}_{a_s}$ konfiantza-balioak izanik, klase bera proposatu duten sailkatzaileen konfiantza-balioak batu egiten dira, eta batura altuena lortzen duen y_j etiketa da multi-sailkatzaileak kasu berriari esleitzen diona, $j = 1, \ldots, e$.

- Random Forests (RF).

  Bagging multi-sailkatzaileetan egiten den bezala, Random Forests multi-sailkatzailea eraikitzeko ere Bootstrap laginak sortzen dira, eta haietan oinarrituz hainbat Sailkatze-Zuhaitz induzitzen dira. Hala ere, zuhaitzen indukziorako ez dira aldagai iragarle guztiak kontuan izaten; aldiro zoriz aukeratutako aldagai iragarleen azpimultzo bat erabiltzen da. Zuhaitzak bere osotasunean garatzen dira, inolako inausketa estrategiarik aplikatu gabe. Test fasean, zuhaitzen iragarpenak konbinatzeko gehiengoaren aldeko bozketa aplikatzen da (Breiman, 2001).

- Oinarrizko sailkatzaile desberdinez osatutako multi-sailkatzaileak.

  Entrenamendurako datuen multzotik abiatuz izaera desberdineko hainbat sailkatzaile induzitu eta guztien iragarpenak konbinatzea multi-sailkatzaile bat sortzeko beste modu bat da. Oinarrizko sailkatzaileen iragarpenak konbinatzeko estrategia desberdinak erabil daitezke: gehiengoaren aldeko bozketa, probabilitateen batez bestekoa, probabilitateen biderkadura (Kuncheva, 2004).

Gainbegiratutako sailkatzerako eredu bat induzitu ondoren, ereduaren ontasun-maila neurtzea garrantzitsua gertatzen da, horren arabera erabaki ahal izango baita ebatzi nahi den sailkatze-problemarako baliozkoa den.

III.4 Sailkatzearen ebaluazioa

Sailkatzaile bat ebaluatzeko orduan, sailkatzailea bitarra den edo klase anitzekoa den berezitu behar da. Izan ere, sailkatze-problema batzuetan C klase-aldagaiak bi balio posible besterik ez dituen bitartean, beste zenbaitetan balio gehiago har ditzake. Horren arabera, sailkatzailea modu desberdinean ebaluatuko da.

Ikerketa-lan honetan sailkatze-problema desberdinak ebatzi ditugu; horietako batzuk sailkatze-problema bitar moduan emanak etorri zaizkigu (Anafora Pronominala-Ebaztea eta Korreferentzia-Ebaztea), beste batzuk aldiz, klase anitzeko sailkatze-problemak izan dira (Testu-Sailkatzean 17, 135 eta 45 klaseko hiru sailkatze-problema ebatzi ditugu eta Hitzen Adiera-Desanbiguatzean 13 klasera artekoak).

III.4.1 Sailkatze bitarraren ebaluazioa (binary)

Sailkatze-problema bat bitarra dela esaten da C klase-aldagaia bitarra denean. Ohikoa den notazioari jarraituz, $C = 1$ notazioaz adierazten da klase positiboa eta $C = 0$ notazioaz klase negatiboa. Klase hori *erreala* dela esaten da, datuen multzoko kasuek C klase-aldagaian duten balioa adierazten duelako. Sailkatzaile bitar batek datu multzoko kasu horiei buruzko iragarpena egitean, klase positiboa $C_M = 1$ edo negatiboa $C_M = 0$ iragarriko du. Azpi-indizeko M horrek sailkatze-ereduari egiten dio erreferentzia (ingelesezko Model).

Klase erreala eta iragarritakoa izanik, datu multzoko instantzietarako egindako iragarpenak zuzenak izan diren egiazta daiteke, eta horren arabera *kontingentzia-taula* osatu (ikus III.2. Taula). Diagonal nagusian klase positiborako (TP, True Positive) eta negatiborako (TN, True Negative) egindako iragarpen zuzen kopuruak agertzen dira. Diagonal nagusitik kanpo oker egindako iragarpen kopuruak, klase positibokoak (FP, False Positive) eta negatibokoak (FN, False Negative).

Kontingentzia-taulako lau balio horietan oinarrituz, hainbat ebaluazio-neurri definitzen dira. Hona hemen oso erabiliak diren horietako batzuk:

- Asmatze-tasa (Accuracy): Testeatuak izan diren kasuen artean, iragarpen zuzena jaso duten kasuen proportzioa da. Ez da bereizten asmatutako iragarpen horiek klase positibokoak ala negatibokoak diren.

$$\text{asmatze-tasa} = \frac{TP + TN}{TP + FP + FN + TN}$$

		Klase erreala	
		$C = 1$	$C = 0$
Iragarritako klasea	$C_M = 1$	TP	FP
	$C_M = 0$	FN	TN

III.2 Taula: Sailkatze bitarrerako kontingentzia-taula.

- Doitasuna (Precision): Testeatuak izan eta klase positiboko iragarri diren kasuen artean zein proportziorekin asmatu den neurtzen du.

$$\text{doitasuna} = \frac{TP}{TP + FP}$$

- Estaldura (Recall): Testeatuak izan eta klase positiboko diren kasuen artean zein proportziorekin asmatu den neurtzen du.

$$\text{estaldura} = \frac{TP}{TP + FN}$$

Doitasunaren eta estalduraren arteko oreka aurkitzea ez da erraza izaten. Izan ere, doitasun altuko sailkatzaileak estaldura baxukoak izatea edota doitasun baxukoak estaldura altuko izatea nahiko arrunta gertatzen da. Hori dela eta, bi neurriak konbinatzen dituen beste neurri bat erabili ohi da sailkatzaileak ebaluatzeko: F_1 neurria.

- F_1 neurria (F_1-score edo F_1-measure): Doitasunaren eta estalduraren batez besteko harmonikoa da.

$$F_1 = \frac{2 \cdot \text{doitasuna} \cdot \text{estaldura}}{\text{doitasuna} + \text{estaldura}}$$

Aipatutako ebaluazio-neurri horiek kalkulatzeko, datu-multzoko instantziak testeatu (sailkatu) egin behar dira. Horretarako, kontuan izan behar da datuen multzoa antolatzeko eta sailkatze-eredua ebaluatzeko metodo desberdinak daudela. Ikerketa-lan honetan erabili diren ebaluazio-metodoak honakoak dira:

1. Baztertze bidezko balidazioa (Hold-out validation).

Datuen multzotik azpimultzo bat baztertzen da, sailkatze-ereduaren indukzioa instantzia horiek kontuan hartu gabe egiteko. Horrela, datuen multzoa bi azpimultzo disjuntutan banatuta geratuko da: *entrenamendurako azpimultzoa*, ereduaren indukziorako erabiliko dena (datu-multzoko instantzien %66, adibidez) eta *testerako azpimultzoa*, induzitutako ereduaren ebaluazioa egiteko erabiliko dena (gainerako instantziak, %33). Ebaluazioaren emaitza egindako entrenamendu-test banaketaren mendekoa da.

2. k geruzako balidazio gurutzatua (k-fold cross-validation)

   Datuen multzoa beren artean disjuntu eta tamainaz antzeko diren k geruzetan banatzen da eta geruzetako instantziak zoriz aukeratzen dira. Entrenamendu-test faseak k aldiz burutu behar dira, aldiro geruza bat testeatuko delako gainerako geruzak erabiliz induzitutako sailkatze-ereduarekin. Baztertze bidezko balidazioan ez bezala, k geruzako balidazio gurutzatuaren bidez datu-multzoko kasu guztiak testeatzea lortzen da.

Sailkatze-problema guztiak ez dira bitarrak. Hori dela eta, sailkatzaile bitarren ebaluaziorako neurriak egokitu egin behar dira klase anitzeko sailkatzaileen ebaluazioa egiteko.

III.4.2 Klase anitzeko sailkatzearen ebaluazioa (Multi-class)

Sailkatze-problema bat klase anitza dela esaten da C klase-aldagairako e etiketa edo klase existitzen badira, $C = \{y_1, \ldots, y_e\}$, kopuru hori bi baino handiagoa izanik, $|C| = e > 2$. Klase anitzeko sailkatze-problemetan, instantzia berri bat sailkatzea instantziari etiketa horietakoren bat esleitzea da.

Klase anitzeko sailkatzean, datu-multzoko C klase-aldagaiaren y_j etiketa bakoitzari erreparatuz gero, y_j klasekoak diren eta ez diren instantziak daudela esan daiteke ($C = y_j$, $C = \bar{y_j}$). Sailkatze-prozesuan test kasu bati esleitutako etiketaren arabera y_j klasekoa dela edo ez dela iragartzen dela interpreta daiteke ($C_M = y_j$, $C_M = \bar{y_j}$). Hori dela eta, sailkatze bitarrerako definitutako kontingentzia-taula osa daiteke y_j klaserako, $j = 1, \ldots, e$.

III.3. Taulan y_j klaserako kontingentzia-taula ikusten da. Sailkatze-problemak duen klase kopurua adina kontingentzia-taula osa daitezke horrela eta horietatik abiatuz, y_j klase bakoitzerako doitasuna$_j$ eta estaldura$_j$ kalkulatu.

$$\text{doitasuna}_j = \frac{TP_j}{TP_j + FP_j}, \quad \text{estaldura}_j = \frac{TP_j}{TP_j + FN_j}$$

		Klase erreala	
		$C = y_j$	$C = \bar{y_j}$
Iragarritako klasea	$C_M = y_j$	TP_j	FP_j
	$C_M = \bar{y_j}$	FN_j	TN_j

III.3 Taula: y_j klaserako kontingentzia-taula.

Doitasun eta estaldura horiekin y_j klaseetarako F_1^j neurriak kalkulatzen dira. Klaseen zailtasunaren arabera oso balio desberdinak ager daitezke; doitasun eta estaldura altuko klaseek F_1^j neurri altua izango dute, sailkatze-prozesua klase horietan oso ona izan denaren adierazgarri. Balio baxua lortu duten klaseetan sailkatze-prozesuaren eraginkortasuna baxua dela interpretatuko da. F_1^j horien batez bestekoa kalkulatuz, klase guztiak kontuan hartzen dituen ebaluazio-neurri bat lortzen da: makro-F_1 neurria. Batez bestekoa modu honetara kalkulatzean, y_j klase guztiei garrantzia bera ematea lortzen da.

$$\text{makro-}F_1 = \frac{\sum_{j=1}^{e} F_1^j}{e}$$

Doitasuna eta estaldura klase bakoitzerako osatutako kontingentzia-taulatik kalkulatu beharrean, balio guztiak elkartu eta kontingentzia-taula bakar bat osa daiteke (ikus III.4. Taula). Klase guztietarako lortutako balioen batura agertzen da taulan, kontaketa orokor bat, alegia. Taulako gelaxka bakoitzak biltzen ditu mota bereko iragarpen zuzen eta oker kopuruak, instantziak zein y_j klasekoak diren berezitu gabe. Horrela, testeatutako instantzia guztiei garrantzia bera ematea lortzen da.

Kontaketa orokor hori erabiliz kalkulatutako doitasunari eta estaldurari mikro-doitasuna eta mikro-estaldura esaten zaie.

$$\text{mikro-doitasuna} = \frac{\sum_{j=1}^{e} TP_j}{\sum_{j=1}^{e} (TP_j + FP_j)}, \quad \text{mikro-estaldura} = \frac{\sum_{j=1}^{e} TP_j}{\sum_{j=1}^{e} (TP_j + FN_j)}$$

Mikro-doitasun eta mikro-estaldura horiekin mikro-F_1 neurria kalkulatzen

		Klase erreala	
	$j = 1, \ldots, e$	$C = y_j$	$C = \bar{y_j}$
Iragarritako klasea	$C_M = y_j$	$\sum_{j=1}^{e} TP_j$	$\sum_{j=1}^{e} FP_j$
	$C_M = \bar{y_j}$	$\sum_{j=1}^{e} FN_j$	$\sum_{j=1}^{e} TN_j$

III.4 Taula: Klase anitzeko sailkatzerako kontingentzia-taula. Matrizeak $C = \{y_1, \ldots, y_e\}$ klase guztietarako balioak biltzen ditu

da. Neurri horrek ez die y_j klase guztiei garrantzia bera ematen, test instantzia positibo kopuruarekiko proportzionala den garrantzia baizik.

Laburbilduz, mikro-F_1 neurriarekin instantzia positibo askoko klaseak (handiak) saritzen diren bitartean, makro-F_1 neurriarekin hobeto ikus daiteke sailkatzaile-prozesua eraginkorra izan ote den instantzia positibo kopuru txikiko klaseetan ere.

III.5 Datu-multzo etiketa anitza (Multi-label)

Datu-multzo bat *etiketa anitza* dela esaten da bertako instantziek klase-etiketa bat baino gehiago badute. Kapitulu honen hasieran definitutako datu-multzoan, instantziek $C = \{y_1, \ldots, y_e\}$ klase-etiketen arteko bat besterik ez dute esleituta (ikus 53. orriko III.1. Taula). Datu-multzo etiketa anitzetan, ordea, $\mathbf{x_i}$ instantzia bakoitzak etiketen multzoko $Y_i \subseteq C$ azpimultzo bat du esleituta. III.5. Taulako datu-multzoan 4 instantzia daude eta etiketen multzoa $C = \{y_1, y_2, y_3, y_4, y_5\}$ da. Sailkatze-problema etiketa anitza izanik, $\mathbf{x}$ kasu berria sailkatzea kasuari etiketen azpimultzo bat esleitzea da.

Etiketa kopuruari dagokionez, datu-multzo etiketa anitz guztiak ez dira berdinak. Datu multzo batean batez bestean instantzia bakoitzak duen etiketa kopuruari *etiketa-kardinalitatea* esaten zaio. III.5. Taulako datu-multzoaren etiketa-kardinalitatea, adibidez, 2koa da.

Aplikazio-eremu askotan sortzen da datu-multzo etiketa anitzekin lan egiteko beharra. Hori dela eta, azken urteotan tresna bereziak sortu dira haiekin lan egiteko. Ezagunenak Mulan (Tsoumakas et al., 2011) eta Meka[1] dira, biak Weka softwarea oinarri hartuz garatuak izan direnak. Gainera,

[1] http://meka.sourceforge.net/

Datu-multzoa	X_1	X_2	X_3	X_4	X_5	$Y \subseteq C$
$(\mathbf{x_1}, Y_1)$	8	4	4	3	5	$\{y_1, y_3\}$
$(\mathbf{x_2}, Y_2)$	3	2	5	4	6	$\{y_2\}$
$(\mathbf{x_3}, Y_3)$	7	7	6	5	4	$\{y_1, y_5\}$
$(\mathbf{x_4}, Y_4)$	4	8	7	5	8	$\{y_2, y_4, y_5\}$
$\mathbf{x}$	5	6	4	3	8	$\{?\}$

III.5 Taula: Sailkatze-problema etiketa anitz baterako adibidea.

hainbat datu-multzo etiketa anitz formatu egokian prestatu eta komunitate zientifikoaren eskura jarriak izan dira, bai Mulan-erako[2] eta baita Meka-rako[3] ere.

Etiketa anitzeko sailkatzea bi modutara egin daiteke: sailkatze-problema eraldatuz edo sailkatze-algoritmoa egokituz (Tsoumakas et al., 2010). Lehenengoaren arabera, etiketa anitzeko problema eraldatu eta etiketa bakar bihurtzen da, ondoren sailkatze-algoritmo tradizionalen bidez ebazteko. Bigarrenaren arabera, sailkatze-algoritmoa da egokitzen dena, entrenamendu fasean instantziek esleituta dituzten etiketa guztiak kontuan har daitezen eta test fasean kasu berrien sailkatzea etiketa anitza izan dadin.

Ikerketa-lan honetan ebatzi diren etiketa anitzeko bi sailkatze-problemetan sailkatze-algoritmoa egokitzearen estrategia erabili da. 59. orrian deskribatutako multi-sailkatzailea izan da egokitu duguna, Bagging moduan konbinatutako hainbat k-NN sailkatzailez osatutakoa. Hona hemen egindako egokitzapenen xehetasunak:

- Entrenamendu fasea. Entrenamendurako datu-multzoan instantziek etiketa bat baino gehiago izan dezaketenez, guztiak kontuan hartuak izan dira konfiantza-balioak kalkulatzerakoan. Klaseetarako kontingentzia-taulak eraikiz kalkulatu dira konfiantza-balioak.

- Test fasea. Instantzia berri bat iristean, S_i sailkatzaile bakoitzak dagokion laginean oinarrituz kasu berriaren auzokide hurbilenak kalkulatu eta haien klaseak aztertuz, a_i klasea eta dagokion cv_{a_i} konfiantza-balioa itzuliko ditu.

[2] http://mulan.sourceforge.net/datasets-mlc.html
[3] http://sourceforge.net/projects/meka/files/Datasets/

- Bozketa. $S_1, \ldots, S_s$ sailkatzaileek egindako iragarpenak Bozketa Bayestarra erabiliz konbinatu ditugu, baina estrategia berri bat inplementatu behar izan dugu, sailkatu beharreko instantziari etiketa bakar bat esleitu ordez etiketen azpimultzo bat esleitzeko. Erabili ditugun bi datu-multzo etiketa anitzen kardinalitatea 1,2koa denez, test kasu batzuei $\{y'\}$ etiketa bakarra eta beste batzuei $\{y', y''\}$ bi etiketa esleitzea erabaki dugu, $y' \in C$ izanik bozketan konfiantza-balioen batura total maximoa ($\mathrm{cv}_{y'}^{tot}$) lortu duen etiketa eta $y'' \in C$ hurrengo handiena ($\mathrm{cv}_{y''}^{tot}$) lortu duena. Bigarren etiketa esleitzeko erabilitako irizpidea (III.1) ekuaziokoa da:

$$\mathrm{cv}_{y''}^{tot} > \mathrm{cv}_{y'}^{tot} \times \lambda, \qquad \lambda = 0.1, 0.2, \ldots, 0.9, 1 \tag{III.1}$$

  Irizpidearen arabera, haien arteko diferentzia nahiko txikia denean, test kasuari $\{y', y''\}$ bi etiketak esleituko zaizkio, eta kontrako kasuan $\{y'\}$ etiketa bakarra. λ parametroaren arabera kontrolatu dugu bi konfiantza-balio horien arteko diferentzia. $\lambda = 1$ denean, (III.1) ekuazioko baldintza ez da inoiz beteko, eta ondorioz, ez da sekula bigarren etiketarik proposatuko test kasuarentzat. λ parametroaren optimizazioa egin dugu, balioa txikituz bi konfiantza-balioen arteko diferentzia desberdinetarako probak eginez. Parametroaren balio optimoarekin test-multzoko instantzietarako 1,2ko etiketa-kardinalitatea lortu dugu, gutxi gora behera.

III.5.1 Etiketa anitzeko sailkatzearen ebaluazioa

Sailkatze-problema etiketa anitzetan, test kasu baten sailkatzea zuzena izan den erabakitzea ez da berehalakoa gertatzen. Izan ere, kasuari etiketa multzo bat badagokio eta sailkatzaileak horietako batzuekin asmatu badu, ezin esan daiteke erabat asmatu duenik. Hori dela eta, instantzien ebaluaziorako, eta oro har, sailkatze etiketa anitzaren ebaluaziorako neurri berriak proposatu izan dira azken urteotan. Neurri horietako batzuek *instantzien ebaluazioan* oinarritzen dira; instantzien sailkatzea ze neurritan izan den zuzena neurtzen da. Beste ebaluazio-neurri batzuk, ordea, *etiketen ebaluazioan* oinarritzen dira; sailkatze-problemarako dauden $C = \{y_1, \ldots, y_e\}$ etiketen ebaluazioa egitera bideratuta dauden neurriak dira (Tsoumakas et al., 2010).

Ikerketa-lan honetan ebatzitako etiketa anitzeko sailkatze-problemen ebaluazioa etiketen ebaluazioan oinarritu dugu: mikro-F_1 eta makro-F_1 neurriak erabili ditugu, klase anitzeko sailkatzearen ebaluaziorako erabiltzen diren berberak. Neurri horiek erabiltzea ezinbestekoa izan da guretzat, emaitzak

beste ikerlariek lortutakoekin konparagarri egin nahi izan baititugu (Sebastiani, 2002). Instantzien iragarpenean egindako akatsek isla dute etiketen ebaluazioan oinarritutako F_1 neurri horien kalkuluan, noski. Izan ere, instantzia bati dagokion etiketa ez iragartzeak etiketa horren estalduraren jaitsiera eragingo du; instantzia bati ez dagokion etiketa baten iragartzeak, aldiz, etiketa horren doitasuna jaitsiko du.

III.6 Aldagaien aukeraketa

Datu-multzoan instantzien informazioa hainbat ezaugarri edo aldagai iragarleren bidez emana dator. Aldagai iragarle guztiak erabiliz sailkatzailea induzitu ordez, askotan egokiagoa gertatzen da aldagaien azpimultzo bat aukeratu eta horretan oinarritzea entrenamendu fasea. Izan ere, gerta daiteke aldagai horietako asko esanguratsuak ez izatea edota erredundanteak izatea C klase-aldagaiaren iragarpenean. Hori dela eta, aldagaien azpimultzo egokiaren aukeraketa egin behar izaten da. Azpimultzo egokia aukeratuz gero, azpimultzo horretan oinarrituz induzitutako sailkatzaileak jatorrizko datu-multzoak dituen aldagai guztietan oinarrituz induzitutakoak baino hobe adieraziko du instantzien bidez emana datorren problema, eta horri esker, orokortzeko gaitasun handiagoa erakutsiko du, test fasean sailkatzailearen portaera hobea izango delarik (Guyon, 2003).

Ikerketa-lan honetan aldagaien aukeraketa batez ere hirugarren atazan egin dugu: euskarazko Anafora Pronominala Ebaztearen atazan. Horretarako, ezaugarrien garrantzia neurtu dugu, eta azpimultzo desberdinetarako probak egin ditugu, aldagaien aukeraketak azken emaitzan duen eragina aztertzeko.

Aldagaien aukeraketa dimentsioaren murrizketa egiteko modu bat da, baina ez da bakarra. Izan ere, aldagaien multzoan azpimultzo bat aukeratu ordez, aldagaien eraldaketa egiten duten teknikak existitzen dira. Dimentsioaren murrizketa egitea da, azken batean, lortu nahi izaten dena. Ikerketa-lan honetan Balio Singularretan Deskonposatzearen teknika erabili dugu. Egia esan, dimentsioaren murrizketaren azterketa ikerketaren helburu nagusietako bat izan da. Duen garrantziagatik eta Latent Semantic Indexing tresnarekin duen harreman zuzenagatik, LSIren oinarri matematikoari buruzko kapituluan arreta berezia eskaini diogu.

APLIKAZIO EREMUAK

IV. KAPITULUA

Testu-Sailkatzea

Azken hamarkadetan izugarri hazi da formatu digitalean eskuragarri dagoen testuen kopurua eta dokumentu horien edukia modu automatikoan kudeatzeko beharra sortu da. Albiste agentzietan, adibidez, testuak automatikoki sailkatzeko beharra aspaldi sumatu zen. Horrek Testu-Sailkatze automatikoa modako jarduera bihurtu zuen eta informazioaren kudeaketan ari ziren ikerlarien arreta erabat erakarri zuen.

Testuei aldez aurretik definitutako kategoria semantikoak esleitzeari *Testu-Sailkatzea* (*Text Categorization*, *Text Classification*) deritzo (Sebastiani, 2002). Hasiera batean jarduera modu automatikoan burutzen zuten sistemak eskuz diseinatutako erregeletan oinarritzen baziren ere, 1990eko hamarkadan Ikasketa Automatikoan oinarritutako sistemak sortzen hasi ziren. Horiek lortu zuten arrakasta ikusita, etengabean sortu dira aplikazio-eremu desberdinetarako eta behar berriei aurre egiteko sistema berriak.

Tamalez, testuen esanahia modu automatikoan kudeatzea zaila gertatzen da, are gehiago sinonimia eta polisemia bezalako fenomenoekin. Testuinguru horretan sortu zen Latent Semantic Indexing (LSI), dokumentuen ezkutuko (latent) semantika jasotzeko bereziki abila den teknika, eta oso portaera ona erakutsi zuen Testu-Berreskuratzean (*Text Retrieval*). Esperimentu haiek erakutsi zuten, sinonimoak diren hitzak antzeko testuinguruetan agertzen direnez, agerkidetza horien kudeaketatik LSIk lortzen duela semantikoki antzeko diren dokumentuak eta hitzak adierazteko antzeko adierazpen matematikoak kalkulatzea. Agerkidetzen tratamendu horrek Testu-Berreskuratzea asko hobetzen du, bilaketan erabilitako hitza zehazki ez duten baina semantikoki antzeko diren testuak berreskuratzea posiblea gertatzen delako, sinonimiak eta polisemiak tradizionalki sortutako zailtasunak neurri handi

batean gaindituz (Deerwester et al., 1990).

LSI tresnak testuen semantika jasotzeko erakusten duen trebetasunagatik, testuak sailkatzeko oso egokia gertatzen da eta horregatik aukeratu da ikerketa-lan honetan Testu-Sailkatzearen ataza lehenengo aplikazio-eremu. Euskaraz eta ingelesez idatzitako dokumentuekin egin dugu lan. Egia esan, euskarazko dokumentuen sailkatzea, bere horretan, abiapuntu natural eta interesgarria izan da. Gainera, euskarak baditu beste hizkuntza batzuek ez dituzten hainbat ezaugarri, Testu-Sailkatzea zailagoa egiten dutenak. Izan ere, euskara eranskaria da eta ezaugarri morfosintaktikoak kontuan hartu behar izan dira. Ingelesezko corpus estandar batekin ere egin dira esperimentuak, emaitzak beste autore batzuek lortutakoekin konparagarri egiteko.

Kapitulu honetan aurkezten diren esperimentuetan erabilitako corpusak oso izaera desberdinekoak dira. Hasteko, oso domeinu desberdinetako testuez osatuta daude: kazetaritza arlokoak, gai ekonomikoei buruzkoak eta txosten klinikoak. Corpus horien informazio zehatzagoa hurrengo ataletan ageri da. Bestalde, tamaina aldetik oso desberdinak dira, eta horrek sailkatzearen eraginkortasunean erabateko eragina izan dezake. Sailkatze-problema motari dagokionez, landutako hiru domeinuetan ebatzi diren sailkatze-problemak klase anitzekoak dira eta horietako bi, gainera, etiketa-anitzak.

Hain izaera desberdineko corpusekin lan egin izanak, aukera zabala eskaini digu Ikasketa Automatikoko teknikak aplikatzeko eta dimentsioaren murrizketarekin esperimentatzeko. Kapitulu honetan Testu-Sailkatze atazari egindako ekarpena aurkezten da. Hasteko, aplikazio-eremu izan diren hiru domeinuak aurkezten dira: IV.1. atalean kazetaritza arloko Testu-Sailkatzea, IV.2. atalean gai ekonomikoei buruzko dokumentuena eta IV.3. atalean dokumentu klinikoena. IV.4. atalean egindako esperimentuetatik ondorioztatutakoak aipatzen dira, eta argitalpenak IV.5. atalean bildu dira.

IV.1 Kazetaritza arloko Testu-Sailkatzea

Kazetaritza arloko Testu-Sailkatzea egiteko, *Euskaldunon egunkaria* egunkariko artikuluez osatutako corpusa erabili da lan honetan. Guztira 6064 dokumentuz osatutako corpusa da; entrenamendurako 4548 dokumentu erabili ditugu eta testerako gainerako 1516 dokumentuak. Klase anitzeko datu-multzoa da, artikuluak 17 kategoriakoak direlako, kazetaritza arloko dokumentuak sailkatzeko International Press Telecommunications Council (IPTC)[1] elkarteak sortutako 17 IPTC kode nagusietakoak: Kultura, Justizia, Honda-

[1] http://www.iptc.org/ http://xml.coverpages.org/NITF30-subject-codes.html

mendiak, Ekonomia, Hezkuntza, Ingurugiroa, Osasuna, Giza Interesa, Lana, Bizimodua, Politika, Erlijioa, Zientzia, Gizartea, Kirola, Liskarrak eta Eguraldia. Corpuseko dokumentu bakoitzak klase bakarra du esleituta; dokumentu-multzoa ez da etiketa-anitza.

kros donostia kros jokatu izan gaur zubi zalditoki oso ireki behin europar izan faborito erredakzio Donostia Europa **atleta** izan faborito nagusi gaur goiz zubi zalditoki jokatu izan Donostia 44. nazioarteko kros irabazi behin Kenya eta Etiopia korrikalari bigarren maila izan izan eta euskal herri Espainia eta **atleta** arte erabaki izan garaipen antolatzaile ez ukan izar handi ekarri eta nahi izan edun maila paretsu ukan **atleta** ekarri lasterketa lehiatsu izan edin euskal telebista zuzenean eskaini edun kros 11:30 aurre gizonezko maila Jonathan Brown Martin Fiz Julio Rey Fermin Cacho eta Fabian Roncero osatu edun faborito zerrenda zubi zalditoki basa beterik egon izan kontu izan Jonathan Brown Elgoibar bigarren izan izan galestar izan faborito nagusi gaurko Donostia nazioarte kros Martin bi Julio Rey Fermin Cacho eta Fabian Roncero antolatzaile kontratatu edun azken izar bestalde ezusteko eman saiatu izan Reyes Estévez eta Teodoro Cuñado ere oso kontu hartu korrikalari izan bestalde Kenya eta Etiopia **atleta** faborito ez izan ere Kipkurui Misoi Philip Mosima eta Julius Chelule animatzaile izan izan Fita Bayisa eta Chala Kelile etiopiar esperientzia han gizon izan lehen eta bigarren geratu izan 1992 ekitaldi eta ez izan egun pasatu etorri guzti berrogeita bost **atleta** inguru izan izan gizonezko lasterketa

IV.1 Irudia: *Euskaldunon egunkaria*ko artikuluez osatutako corpuseko dokumentu lematizatu bat

Corpus horrekin egindako esperimentuetan Ikasketa Automatikoko lau sailkatzaile erabili dira (Naive Bayes, Winnow, Support Vector Machines eta k-Nearest Neighbors), eta arreta berezia eskaini zaio dimentsioaren murrizketarako dauden hainbat teknikari. Gainera, euskara hizkuntza eranskaria izanik, testuak lematizatzearen eta ez lematizatzearen arteko aldea neurtu nahi izan da, eta horretarako corpusaren hiru formatu desberdinekin egin dira probak: hitzez osatutako corpusa (W), lemez osatutakoa (L) eta izenez osatutakoa (N). IV.1. Irudikoa lematizatua izan den dokumentu bat da, *Kirola* klasekoa. Urdinez agertzen den "atleta" lema, adibidez, jatorrizko dokumentuan hainbat forma desberdinetan agertzen da, "atletak" esaterako. Forma guztiak lema bihurtuz, funtsean esanahi bera duten hitzak bakar batean biltzen dira. Horrela, testuaren semantika hobeto jasoko duen adierazpena lor daiteke.

Euskarazko corpus horrekin egindako sailkatze-esperimentuen helburua Ikasketa Automatikoko sailkatzaileen eraginkortasuna neurtzea eta dimentsioaren murrizketaren eragina aztertzea izan da. Funtsean bi esperimentu bideratu dira.

- Lehenengo esperimentuan terminoen aukeraketarako lau estrategia desberdin erabili dira (guztiak aukeratu, gutxienez 2 dokumentutan agertzen direnak, gutxienez 3tan agertzen direnak eta gutxienez 4tan agertzen direnak): (W) corpusean 73728, 33294, 22821 eta 17776 hitz izan dira aukeratuak, (L) corpusean 34729, 15175, 10750 eta 8542 lema eta (N) corpusean 10381, 7301, 5913 eta 5050 izen. Dokumentuen sailkatzearekin hainbat proba egin dira 12 dimentsio horietan.

- Bigarren esperimentuan LSI/SVD bidezko dimentsio-murrizketa probatu dugu. Entrenamendurako 4548 dokumentuak aztertuz, dokumentuetan gutxienez 2 aldiz agertzen diren hitzak (34288), lemak (14648) edo izenak (7209) termino izateko aukeratuak izan dira, eta haiekin sortu dira hiru termino-dokumentu maiztasun matrizeak: 34288×4548 (W), 14648×4548 (L) eta 7209×4548 (N). Matrizeei Log-entropy eraldaketa aplikatu zaie SVD deskonposaketa kalkulatu aurretik. Dimentsio-murrizketa eginez, 100, 200, 300, 400, 500 eta 1000 dimentsiorekin egin dira probak.

Matrizeak sortu eta SVD deskonposaketa egitearen kostu konputazionala, kalkulatu nahi den balio singular kopuruaren araberakoa da. 34288×4548 matrizean 500 balio singular, eta dagozkien bektore singularrak kalkulatzeko, adibidez, ordu eta erdi inguru behar izan da[2]. Test-bektoreen proiekzio koordenatuen kalkuluak ere antzeko denbora behar izan du. Gaur egungo prozesadoreekin esperimentu hauek azkarrago egiteko aukera egongo litzateke.

Sailkatzaileek erakutsi duten portaerari dagokionez, lehenengo esperimentuan Support Vector Machines sailkatzailea izan da emaitza onenak eman dituena (%84.56ko mikro-F_1 neurria), eta LSI/SVD dimentsioak erabili direnean k-NN sailkatzailea (%87.33ko mikro-F_1 neurria); bi kasuetan lematizatutako corpusean lortu dira aipatutako emaitzak. Esperimentuen gainerako xehetasunak artikuluan agertzen dira.

Lortutako emaitzak "Human Language Technologies as a Challenge for Computer Science and Linguistics"(L&TC-05) biltzarrean aurkeztuak izan dira 2005. urtean Poznań hirian (Polonia). Gainera, artikulua "Archives of Control Sciences"aldizkarian[3] argitaratua izateko aukeratua izan da.

- Analyzing the Effect of Dimensionality Reduction in Document Categorization for Basque. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Proceedings of the 2th Language and Technology Conference

[2]IXAko sisx02: Sun Enterprise 250 − 2 × UltraSPARC II 400 MHz

[3]http://acs.polsl.pl/

(L&TC-05), 72-75, (2005), Poznań, Polonia. (Zelaia et al., 2005a). Selected and also published in: Archives of Control Sciences, Vol. 15(4), 703-710, (2005). (Zelaia et al., 2005b)

Testu-Sailkatze automatikoak teknologian oinarritutako hezkuntzarako tresnen garapenean, eta bereziki haien domeinu-moduluaren garapenean duen garrantziaz ohartuta, egindako lana eta emaitzak "International Conference on Computer Supported Education"(CSEDU-2009) biltzarrean aurkeztuak izan dira 2009. urtean Lisboan (Portugal).

- Exploring Basque Document Categorization for Educational Purposes using LSI. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Ana Arruarte, Arantza Díaz de Illarraza, Jon Ander Elorriaga, Basilio Sierra. Proceedings of the first International Conference on Computer Supported Education (CSEDU-2009), Vol. 1, 5-10, (2009), Lisboa, Portugal. (Zelaia et al., 2009a)

IV.2 Gai ekonomikoei buruzko Testu-Sailkatzea

Reuters-21578 corpusa[4] gai ekonomikoei buruzko artikuluez osatuta dago eta estandar bihurtu da Testu-Sailkatze sistemen ebaluazioan (Debole and Sebastiani, 2005), (Lewis, 2004). Testuen kategorizazioan esperimentuak egin izan dituzten autoreek corpusaren hainbat banaketa desberdin erabili izan dituzte. Lan honetan ModApte banaketa erabili da, entrenamendurako 9603 dokumentu eta testerako 3299 dituena.

Dokumentuak 135 kategoriatan sailkatuta daudenez, klase anitzeko datu-multzoa da. Egia esan, dokumentuen banaketa kategoria horietan nahiko xelebrea da: kategoria batzuetarako badaude dokumentuak entrenamendurako datu-multzoan baina ez testekoan. Beste batzuetan, aldiz, kontrakoa gertatzen da. Halaber, badira ez entrenamendurako datu-multzoan ez testekoan dokumenturik ez duten kategoriak. Hori dela eta, sailkatze-sistemen ebaluaziorako kategorien hiru azpimultzo erabiltzen dira: Top-10, R(90) eta R(115). Top-10 izenez ezagutzen dena, entrenamendurako datu-multzoan dokumentu kopuru altuena duten honako 10 kategoriek osatzen dute: Earnings, Acquisitions, Money-fx, Grain, Crude, Trade, Interest, Ship, Wheat eta Corn. R(90) azpimultzoan bai entrenamenduan eta bai testean gutxienez dokumentu bat baduten 90 kategoriak, eta R(115) azpimultzoan entrenamenduan gutxienez dokumentu bat baduten 115 kategoriak daude.

[4] http://www.daviddlewis.com/resources/testcollections/reuters21578

Datu-multzoa klase anitzekoa izateaz gain etiketa anitza ere bada, kategoria bat baino gehiagokoak diren dokumentuak daudelako. Datu-multzoaren etiketa-kardinalitatea, hau da, batez beste dokumentuek esleituta duten etiketa kopurua, 1,2koa da. IV.2. Irudiko dokumentuak, adibidez, bi etiketa ditu esleituta, etiketatzaileek bi kategorietakoa dela erabaki baitute; Wheat eta Grain, biak Top-10 azpimultzokoak.

```
BONUS WHEAT FLOUR FOR NORTH YEMEN  -- USDA

The Commodity Credit Corporation, CCC, has accepted an export bonus offer to cover
the sale of 37,000 long tons of wheat flour to North Yemen, the U.S. Agriculture
Department said. The wheat four is for shipment March-May and the bonus awarded
was 119.05 dlrs per tonnes and will be paid in the form of commodities from the
CCC inventory. The bonus was awarded to the Pillsbury Company. The wheat flour
purchases complete the Export Enhancement Program initiative announced in April,
1986, it said.
```

IV.2 Irudia: Reuters-21578 corpuseko dokumentu bat

Datu-multzo etiketa anitzei buruzko III.5. atalean aipatu den bezala, azken urteotan mota horretako sailkatze-problemekin lan egiteko tresnak garatu dira. Mulanek eta Mekak, adibidez, egokitutako algoritmoen inplementazioak eskaintzeaz gain, hainbat datu-multzo etiketa anitzekin lan egiteko aukera ematen dute. Reuters-21578 corpusaren bertsio berri bat den RCV1 (Reuters Corpus Volume 1) corpusa horien artean dago (Lewis et al., 2004). Ikerketa lan hau egin zenean tresna horiek oraindik garatu gabe zeuden. Hemen aurkezten diren emaitzak lan honetarako diseinatutako eta inplementatutako sailkatze-sistemarekin lortutakoak dira, eta lana egindako garaian gainerako autoreek corpusaren banaketa berarekin argitaratutako emaitzekin konparatzen dira.

Euskarazko testuez osatutako corpusarekin egin bezala, Reuters-21578 corpusarekin egindako esperimentuetan arreta berezia eskaini zaio dimentsioaren murrizketari. Alde batetik, jatorrizko testuekin eta Porter-en erro--bilatzailea aplikatu ondoren lortutakoekin lan egitearen arteko aldea neurtu nahi izan da, eta horretarako corpusaren bi formatu desberdinekin egin dira probak: jatorrizko corpusa (BoW, Bag of Words) eta hitzen erroez osatutakoa (BoS, Bag of Stems). Entrenamendurako 9603 dokumentuak aztertuz, dokumentuetan gutxienez 2 aldiz agertzen diren hitzak (15591) edo erroak (11114) termino izateko aukeratuak izan dira, eta haiekin sortu dira bi termino-dokumentu maiztasun matrizeak: 15591×9603 (BoW) eta 11114×9603

(BoS). Matrizeei Log-entropy eraldaketa aplikatu zaie.

Matrizeak tamaina aldetik euskarazko corpusarekin sortutakoen parekoak direla esan daiteke, eta oraingoan ere matrizeen SVD deskonposaketa kalkulatzeko antzeko denbora behar izan da. Kasu honetan, 15591×9603 matrizean 500 balio singular eta haien bektore singularrak kalkulatzeko ordubete inguru behar izan da, aurreko domeinurako erabilitako makina bera erabiliz. Test corpuseko dokumentuen proiekzio-bektoreak kalkulatzeko ere ordubete inguru behar izan da.

LSI/SVD bidezko dimentsio-murrizketa egitean, normalean 100 eta 300 dimentsio artean erabiltzea gomendatzen da. Lan honetan aurkezten diren esperimentuetan 1000 balio singular eta bektore singular kalkulatu dira, eta egiaztatu ahal izan da emaitzarik onenak tradizionalki gomendatutako balioen inguruan lortu direla.

Euskarazko corpusarekin LSI/SVD dimentsioetan adierazitako dokumentuak sailkatzean portaera onena erakutsi duen sailkatzailea k-NN izan denez, hainbat k-NN sailkatzaile Bagging bidez sortutako multi-sailkatzaile batean konbinatzea erabaki da. Hasiera batean egindako esperimentuetan 30 sailkatzailez osatutako multi-sailkatzailea erabili da, eta bi parametroren optimizaziorako probak egin dira. Alde batetik, entrenamendurako datu-multzotik sortutako 30 azpilaginen tamaina optimizatu da. Bestetik, datu-multzoa etiketa anitza izanik, sailkatzaileek egindako iragarpenen bozketa unean bigarren etiketa esleitzeko erabilitako irizpidea optimizatu da. Bi parametro horiek finkatuta, multi-sailkatzailearen portaera neurtu da BoW eta BoS corpusetan eta LSI/SVD dimentsio desberdinetarako. Esperimentuaren gainerako xehetasunak artikuluan agertzen dira.

Lehen fase honetan lortutako emaitzak "11th Conference of the European Chapter of the Association for Computational Linguistics" (EACL-06) biltzarrean aurkeztuak[5] izan dira 2006. urtean Trenton (Italia).

- A Multiclassifier based Document Categorization System: profiting from the Singular Value Decomposition Dimensionality Reduction Technique. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, 25-32, (2006), Trento, Italia. (Zelaia et al., 2006).

Bigarren fase batean, ahalegin berezia egin da parametroen optimizazioa hobetzeko. Horrela, aurreko fasean optimizatutako bi parametroez gain, hirugarren parametro bat ere optimizatu da: multi-sailkatzailea eraikitzeko

[5]http://www.aclweb.org/anthology/W/W06/W06-2604.pdf

sortutako azpilagin kopurua, hau da, sailkatzaile kopurua. Hiru parametro horiek finkatuta aurreko faseko emaitzak hobetzea lortu da. Horretaz gain, multi-sailkatzailearen eragina neurtu nahi izan da eta horretarako k-NN bakar bat erabiliz esperimentua errepikatu da; multi-sailkatzailea erabiltzeak hobekuntza nabarmena sortu duela egiaztatu da. Top-10 kategorietan gainerako autoreek lortutako emaitza hobetzea lortu da (%94.10eko mikro-F_1 neurria), eta R(90) kategorietan haien pareko emaitzak lortu dira (%88.26ko mikro-F_1 neurria). Esperimentu osoaren deskribapena eta azken emaitzak jasotzen dituen artikulua "Applied Soft Computing" aldizkarian argitaratua izan da.

- A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Applied Soft Computing Vol. 11, 4981-4990, (2011). (Zelaia et al., 2011). Inpaktu-faktorea: 2.612 (Q1)

IV.3 Dokumentu Klinikoen Sailkatzea

Testu-Sailkatzean azken urteotan garrantzia handia hartzen ari den aplikazio-eremu bat mediku-txostenen sailkatzearena da. Gaur egungo ospitaleetan, mediku-txosten ugari sortzen da egunero formatu elektronikoan: behaketa proben ondorioz sortutakoak, gaixoarekin izandako kontsulten ondoren sortuak, etab. Txosten horietan informazio gehiena testu huts moduan agertzen denez, hizkuntzaren prozesamendu automatikorako teknikak oso lagungarriak gertatzen dira testuetan dagoen informazioa erabili ahal izateko.

Hori dela eta, mediku-txostenen sailkatze automatikoa Hizkuntzalaritza Konputazionalaren aplikazio-eremu garrantzitsu bihurtu da. Arlo espezifiko horren inguruko ikerketa bultzatzeko eta sistema automatikoen garapena sustatzeko antolatutako txapelketa batean parte hartu dugu[6], eta aukera ezin hobea izan da Reuters-21578 corpusarekin egindako esperimentuetarako eraikitako multi-sailkatzailearen portaera beste domeinu bateko testuez osatutako corpus batean neurtzeko. Txapelketa honako izenburupean antolatu da: *CMC'07: Computational Medicine Center's 2007 Medical Natural Language Processing International Challenge: classifying clinical free text using natural language processing.*

Ebatzi beharreko sailkatze-problemaren azalpena, corpusaren zehaztapenak eta emaitzen ebaluazioaren inguruko xehetasunak (Pestian et al., 2007) artikuluan ematen dira. Corpuseko dokumentuak Cincinnati Children's Hos-

[6] http://osdir.com/ml/science.linguistics.corpora/2007-01/msg00153.html

```
<doc id="97726428" type="RADIOLOGY_REPORT">
  <codes>
    <code origin="CMC_MAJORITY" type="ICD-9-CM">486</code>
    <code origin="CMC_MAJORITY" type="ICD-9-CM">518.0</code>
    <code origin="CMC_MAJORITY" type="ICD-9-CM">786.07</code>
    <code origin="COMPANY3" type="ICD-9-CM">786.07</code>
    <code origin="COMPANY1" type="ICD-9-CM">486</code>
    <code origin="COMPANY1" type="ICD-9-CM">518.0</code>
    <code origin="COMPANY1" type="ICD-9-CM">786.07</code>
    <code origin="COMPANY2" type="ICD-9-CM">486</code>
    <code origin="COMPANY2" type="ICD-9-CM">518.0</code>
  </codes>
  <texts>
    <text origin="CCHMC_RADIOLOGY" type="CLINICAL_HISTORY">
    This is a 7-month - old male with wheezing.</text>
    <text origin="CCHMC_RADIOLOGY" type="IMPRESSION">
    Borderline hyperinflation with left lower lobe atelectasis versus pneumonia.
    Clinical correlation would be helpful. Unless there is clinical information
    supporting pneumonia such as fever and cough, I favor atelectasis.</text>
  </texts>
</doc>
```

IV.3 Irudia: 2007ChallengeTrainData.xml corpuseko dokumentu bat

pital Medical Center's (CCHMC) ospitaleko Erradiologia saileko txosten klinikoak dira: etiketatutako 978 dokumentu entrenamendurako eta etiketatu gabeko 976 testerako.

Dokumentuak gaixotasunak adierazten dituzten 45 kategoriatan sailkatuta daudenez, klase anitzeko sailkatze-problema da. Klaseak ICD-9-CM kodeak dira, nazioarte mailan gaixotasunak sailkatzeko definitutakoak. Horien artean daude, adibidez, 486 (*Pneumonia, organism unspecified*), 518.0 (*Pulmonary collapse*) eta 786.07 (*Wheezing*). ICD-9-CM kodeei buruzko informazioa kode medikuetarako webgunean[7] aurki daiteke.

Datu-multzoa klase anitzekoa izateaz gain etiketa anitza ere bada. Dokumentuak eskuz etiketatuak izan dira hiru etiketatzaileren bidez, eta gehiengoaren aldeko bozketa aplikatuz esleitu zaizkie ICD-9-CM kodeak. IV.3. Irudiko dokumentuan ikus daitezke hiru etiketatzaileek ("companyx") proposatutako kodeak eta gehiengoaren aldeko bozketaz ("cmc_majority") dokumentuak esleituta dituenak: urdinez dauden 486, 518.0 eta 786.07. Kasu honetan, dokumentuak hiru etiketa ditu esleituta, nahiz eta datu-multzoak 1,2ko etiketa-kardinalitatea duen. Reuters-21578 corpusaren kardinalitate bera du, hortaz, eta hura bezala, dokumentu-multzo hau ere Mulan eta Me-

[7]ICD-9-CM, (International Classification of Diseases) http://www.icd9data.com/

```
<?xml version='1.0' standalone='yes'?>
<docs>
  <doc id="97634811" type="RADIOLOGY_REPORT">
    <codes></codes>
    <texts>
      <text origin="CCHMC_RADIOLOGY" type="CLINICAL_HISTORY">
      Seventeen year old with cough.</text>
      <text origin="CCHMC_RADIOLOGY" type="IMPRESSION">
      Normal.</text>
    </texts>
  </doc>
```

IV.4 Irudia: 2007ChallengeTestDataNoCodes.xml corpuseko dokumentu bat

ka tresnekin lan egiteko egokitua izan da eta erraz eskura daiteke tresnen webguneetatik; *Medical* izenez agertzen den datu-multzoa da.

Test dokumentu baten adibidea IV.4. Irudian ikus daiteke. Ikusten den bezala, testua bi ataletan banatuta dago. "Clinical_history" atalean erradiografia egin aurretiko informazioa dago eta "Impression" atalean ondorenekoa. Morez agertzen diren "Seventeen year old with cough." eta "Normal." testuak dira. Hain motzak izanik, bi ataletako testu zatiei aparteko trataera eman ordez, biak dokumentu bakar batean elkartzea erabaki da. Gainera, dokumentu kopuruari erreparatuz, corpusa oso txikia dela esan dezakegu, aurreko bi domeinuetarako izan ditugunak baino txikiagoa. Entrenamendurako 978 dokumentuetan gutxienez 2 aldiz agertzen diren hitzak (872) termino izateko aukeratuak izan dira, eta haiekin sortu da 872×978 termino-dokumentu maiztasun matrizea. Oraingoan ere matrizearen SVD deskonposaketa makina berean egin da, aldez aurretik Log-entropy eraldaketa aplikatuz. 300 balio singular eta haien bektore singularrak kalkulatzeko 3 minutu besterik ez dira behar izan.

Hain tamaina txikiko matrizea izanik, proba desberdinak egin dira, eta matrizearen heina, hau da balio singular kopuru totala 781ekoa dela ikusi da. Ez da balio singular kopuru total maximoa lortu, diagonal nagusian zeroak daudelako. Horrek bektoreen artean mendekotasun lineala dagoela erakusten du, corpusaren tamaina txikiak eta bertan aukeratutako termino kopuru eskasak eragindakoa.

LSI/SVD bidezko dimentsio-murrizketarako probak 100, 150 eta 200 balioetarako egin dira, eta Reuters-21578 corpusarekin lehen fasean erabilitako sailkatze-metodoa aplikatu da: 30 k-NN sailkatzailez osatutako multi-sailkatzailea. Test dokumentuak 150 dimentsio erabiliz sailkatu eta emaitza bidali

ondoren lortu dugun ebaluazioa IV.5. Irudian ikus daiteke. Informazio horren arabera, %66ko mikro-F_1 neurria lortu dugu.

```
Results Validation and Submission

FILE CONFORMS TO RELAX-NG SCHEMA: Passed
ALL CODES HAVE SAME ORIGIN: Passed
DOC IDS MATCH TRAINING/TESTING DATASET: Passed

Main ranking measure: 66%
Cost sensitive accuracy: 67%
```

IV.5 Irudia: CMC'07 txapelketara 150 dimentsio erabiliz bidalitako emaitzari dagokion informazioa.

IV.4 Ondorioak

Testu-Sailkatze automatikorekin egindako esperimentuetatik ateratako ondorioak horrela labur daitezke:

- Euskarazko dokumentuen sailkatze automatikoan emaitza onenak corpus lematizatuan lortu dira; lematizatzeak dokumentuen sailkatzea hobetu du. Ingelesezko corpusean erro-bilatzailea erabiltzeak sortutako onura ez da hain nabarmena izan. Hizkuntzen ezaugarriak kontuan hartuta, euskara eranskaria da eta ingelesa ez, ondorioa zentzuzkoa da.

- Euskarazko dokumentuen sailkatze automatikoan, dokumentu bakarrean agertzen diren hitzak ezabatzea terminoen aukeraketarako estrategia ona izan da. Dimentsioa murriztu da eta kasu gehienetan emaitzak hobetu dira.

- Corpusen tamainari dagokionez, *Euskaldunon egunkaria* corpusa eta Reuters-21578 oso egokiak izan dira, bai dokumentu kopuruari dagokionez eta baita dokumentuen tamainari dagokionez ere, Testu-Sailkatzea Ikasketa Automatikoko sailkatzaileen bidez egiteko. LSI/SVD bidezko dimentsio-murrizketa egiteko ere tamaina aproposeko corpusak izan dira. Dokumentu klinikoen sailkatzean lortutako emaitzak ez

dira onak izan. Aurreko bi domeinuekin konparatuz, diferentzia nagusia corpusaren eta dokumentuen tamaina da. Halako kasuetan termino guztiak aukeratzearen estrategia egokiagoa izan daiteke.

- LSI/SVD bidezko dimentsio-murrizketa aplikatuz lortu dira emaitzarik onenak, bai kazetaritza arloko testuekin eta baita gai ekonomikoei buruzkoekin ere. Gainera, normalean gomendatzen den dimentsio kopururako lortu dira emaitza onenak: 100-300 artean.

- Reuters-21578 corpusarekin Top-10 kategorietarako lortutako emaitzak gainerako autoreek argitaratutakoak baino hobeak dira. Oro har, dokumentu kopuru txikiko kategorietan lortutako emaitzak baxuagoak dira, eta emaitza orokorra jaitsi arazten dute.

- Bagging bidez sortutako multi-sailkatzaileak oinarrizko sailkatzaile bakarrarekin lortutako emaitzak hobetzen ditu.

- Sailkatze-problema etiketa anitzerako diseinatutako estrategiak ondo funtzionatu du.

IV.5 Argiltapenak

Testu-Sailkatzea

Kazetaritza arloko Testu-Sailkatzeari buruzko artikuluak:

- Analyzing the Effect of Dimensionality Reduction in Document Categorization for Basque. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Proceedings of the 2th Language and Technology Conference (L&TC-05), 72-75, (2005), Poznań, Polonia. (Zelaia et al., 2005a). Selected and also published in: Archives of Control Sciences, Vol. 15(4), 703-710, (2005). (Zelaia et al., 2005b)
- Exploring Basque Document Categorization for Educational Purposes using LSI. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Ana Arruarte, Arantza Díaz de Illarraza, Jon Ander Elorriaga, Basilio Sierra. Proceedings of the first international conference on computer supported education (CSEDU-2009), Vol. 1, 5-10, (2009). (Zelaia et al., 2009a)

Gai ekonomikoei buruzko Testu-Sailkatzeari buruzko artikuluak:

- A Multiclassifier based Document Categorization System: profiting from the Singular Value Decomposition Dimensionality Reduction Technique. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, 25-32, (2006), Trento, Italia. (Zelaia et al., 2006).
- A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. Ana Zelaia, Iñaki Alegria, Olatz Arregi, Basilio Sierra. Applied Soft Computing Vol. 11, 4981-4990, (2011). (Zelaia et al., 2011). Inpaktu-faktorea: 2.612 (Q1)

Archives of Control Sciences
Volume 15(LI), 2005
No. 4, *pages 565–572*

Analyzing the effect of dimensionality reduction in document categorization for Basque

ANA ZELAIA, IÑAKI ALEGRIA, OLATZ ARREGI and BASILIO SIERRA

This paper analyzes the incidence that dimensionality reduction techniques have in the process of text categorization of documents written in Basque. Classification techniques such as Naïve Bayes, Winnow, SVMs and *k*-NN have been selected. The Singular Value Decomposition (SVD) dimensionality reduction technique together with lemmatization and noun selection have been used in our experiments. The results obtained show that the approach which combines SVD and *k*-NN for a lemmatized corpus gives the best accuracy rates of all with a remarkable difference.

Key words: text categorization, singular value decomposition (SVD), supervised classification

1. Introduction

Since the early 90s, automated categorization of texts into predefined categories has increased interest because the amount of available documents in digital form are growing fast. Most researchers propose approaches based on machine learning techniques [16], where automatically built classifiers learn from a set of previously classified documents.

The work we are presenting here analyzes the categorization of documents written in Basque. Several experiments have been made to classify documents written in extended languages such as English. But, the reality of lesser-used languages, as is the case of Basque, is different. In practice, one of the main problems we encounter is that only a short amount of manually classified documents is available. This fact restricts the capacity of the classifiers and may, consequently, produce poorer results. In addition to that problem, we must take into account that Basque is an agglutinative and highly inflected language whose declension system has numerous cases [1]. This morphosyntactic feature makes the categorization task more difficult, because semantic information

The Authors are with University of the Basque Country, UPV-EHU, Computer Science Faculty, 649 postakutxa, 20.080 Donostia, Gipuzkoa, Euskal-Herria, Spain, e-mails: {ccpzejaa, acpalloi, acparuro, ccpsiarb}@si.ehu.es

This work is funded by the University of the Basque Country (UPV00141.226-T-14816/2002), the Basque Government (UE02/B11), and Gipuzkoa Council in a European Union Program.

Received 13.10.2005.

is not really contained in word-forms but in their corresponding lemma. Therefore, it seems interesting to preprocess the corpus lemmatizing it and so, at the same time the dimension of the information to treat is reduced, an improvement in the efficiency of the system can be produced. In this paper we analyze the effect that dimensionality reduction techniques such as lemmatization, noun selection and in particular SVD (Singular Value Decomposition) have in the process of text categorization of Basque documents. Latent Semantic Indexing (LSI) implementation has been used to calculate the SVD of the matrix constructed for the training corpus. We have selected some of the most popular classification algorithms and two different experiments have been performed. In the first experiment, the classification techniques are used without applying SVD. In the second one, the same classification techniques are used but previously, the SVD technique has been applied to reduce the dimension. We use three different corpora in both experiments: words, lemmas and nouns. Obtained results show that the SVD dimensionality reduction technique combined with the *k*-NN classification algorithm gives the best results. Moreover, we find that they are obtained for the lemmatized corpus.

This paper is structured as follows. First, we reference previous work on algorithms we use for document categorization, and examine the foundations of LSI. Afterwards, the experimental setup is introduced, where both training and test corpora are described and lemmatization, noun selection and document frequency based feature selection processes are introduced. In the next section, experimental results are shown, compared and discussed. Finally, conclusions and future work are presented.

2. Related work

Text categorization consists in assigning predefined categories to text documents [16]. Simple but effective, the bag-of-words text document representation is one of the most frequently used. In this kind of text representation, the number of attributes in the corpus is usually considerable, and this can be problematic in inductive classification. Therefore, it is usually convenient to apply techniques that reduce the dimension of the representation. This reduction can be carried out in different ways: eliminating irrelevant features (terms), substituting some words by others that represent them (lemmas, synonyms, hyperonyms, etc.), applying SVD technique, etc. In our two experiments we use classification algorithms which have reported good results for text categorization in other languages; in this way, we use Naïve Bayes [14], Winnow [5], SVMs [13] and *k*-NN [6]. Next, we briefly describe the foundations of LSI, which uses SVD for dimensionality reduction.

2.1. SVD using Latent Semantic Indexing (LSI)

LSI[1] was first introduced in 1988 originally developed in the context of Information Retrieval [7,9]. It takes as input a collection of texts composed of *n* documents and

[1] http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

m terms and represents it as an $m \times n$ term-document matrix. The elements m_{ij} of the term-document matrix are the occurrences of term i in a particular document j. This way, we obtain matrix M, with documents represented by a vector in an m-dimensional space [2]. The SVD technique compresses vectors representing documents into vectors of a lower-dimensional space. It consists in factoring matrix $M \in \mathbb{R}^{m \times n}$ into the product of three matrices, $M = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, where $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix of singular values $\sigma_1 \geqslant \cdots \geqslant \sigma_k \geqslant 0$ being $k = \min\{m,n\}$, and U and V are orthogonal matrices of singular vectors. Once matrix M has been factored, it can be approximated by a lower rank M_p which is calculated using the p largest singular triplets of M. This operation is called dimensionality reduction, and the p-dimensional space to which document vectors are projected is called the reduced space. When using the reduced space generated by M_p instead of the one generated by M, most of the important underlying structure that associates terms with documents is captured and consequently, noise is reduced. This results in a representation where similar documents have similar vectors. For text categorization purposes, LSI represents each document to be categorized by a p-dimensional vector. Afterwards, the similarity among it and all the documents in the training set (reduced space) is calculated using the cosine similarity measure. LSI has been successfully used in the categorization of written documents [10,3,8].

3. Experimental setup

The aim of this section is to describe the document collection used in our experiments and to give an account of the lemmatization, noun selection and document frequency based feature selection technique we have applied.

3.1. Document collection

We are interested in the categorization of documents written in Basque. Among all the electronic documents available in Basque, we have selected newspaper texts, because there are standardized categories for this domain, and we have access to a sufficient amount of documents manually categorized. The documents used in this experiment correspond to the Basque newspaper *Euskaldunon Egunkaria* corresponding to the articles published during two months of 1999. They are a total of 6.064 documents categorized to the 17 standard first level IPTC categories[2]. Each of the documents has a unique category associated to it. It must be noted that all categories do not have the same number of documents, as can be seen in Table 1. Document categorization is achieved in two steps: during the *training* step an inductive generalization of the set of documents is obtained, and during the *test* step the effectiveness of the system is measured. Therefore, the 6,064 documents have been split into two different sets of documents: 4,548 documents for training (75 %) and 1,516 documents for testing (25 %). This proportion stands in each one of the 17 categories, as can be observed in Table 1.

[2] http://www.iptc.org

Category	Training	Test
1. Culture	600	202
2. Law & Justice	129	42
3. Disasters	75	26
4. Economy	234	78
5. Education	82	27
6. Environmental Issues	69	22
7. Health	35	12
8. Human interests	36	11
9. Labour	132	43
10. Lifestyle	40	13
11. Politics	1.184	393
12. Religion	25	8
13. Science	35	12
14. Social Issues	464	156
15. Sport	1.283	429
16. Conflicts	100	33
17. Weather	25	9
TOTAL	4.548	1.516

Table 1. Number of documents distributed by categories.

3.2. Feature selection. Lemmatization

As we have mentioned in the introduction, Basque is an agglutinative and highly inflected language. In order to face the difficulties derived from these morphosyntactic features, we have applied two types of feature selection. On the one hand, stopword lists have been used to eliminate non-relevant words, i.e. the most frequent words and words that appear less than a threshold in the training corpus. On the other hand, we use linguistic methods such as lemmatization and noun selection to reduce the number of features.

The studies of the effects that stemming algorithms produce in text categorization are controversial for languages with a low level of inflection such as English, but recent experiments show that lemmatization helps in the process of categorizing documents written in an inflected language using LSI [15]. Therefore, we expect that lemmatization, and noun selection in particular, should allow us to maintain the same semantic information, reducing the number of attributes to be processed.

We have used the Basque lemmatizer designed by the IXA[3] group [11], which obtains for each word in the document, its corresponding lemma, as well as its part-of-speech tag. This system reduces the different number of features from each category by more than 50%. While the number of different word-forms in the whole document collection is 92,373, there are 38,654 different lemmas, among which 14,213 are nouns. So, we have created three different corpora: bag-of-words (W), bag-of-lemmas (L) and bag-of-nouns (N).

[3] http://ixa.si.ehu.es

4. Experimental results

In this section we show the results obtained in the two experiments. In both of them we use the general-purpose classifier named SNoW [4] for Naïve Bayes and Winnow algorithms and Weka [17] for SVMs. In order to evaluate the results, we have concentrated in effectiveness issues, rather than on efficiency ones, and calculate the accuracy rate for each categorization method.

4.1. Experiment before applying SVD

In this experiment, elimination of irrelevant words, lemmas and nouns has been performed based on the word frequency in documents. Terms that appear in more than 1, 2 or 3 documents (>1 doc, >2 doc, etc.) are kept and a constant high threshold has been applied in order to discard functional terms. The resulting number of attributes in the training corpora are shown at the top part of Table 2. The accuracy rates using the test-corpus for each classification technique are shown in the rest part of the table. The best results obtained for each technique and corpus appear printed in boldface.

		all	> 1 doc	> 2 doc	> 3 doc
Numb. of Attrib.	W	73728	33294	22821	17776
	L	34729	15175	10750	8542
	N	10381	7301	5913	5050
Naïve Bayes	W	**80.09**%	78.89%	78.10%	77.77%
	L	**81.53**%	81.07%	80.74%	80.28%
	N	79.49%	**79.62**%	79.35%	79.62%
Winnow	W	80.09%	**81.13**%	80.47%	79.49%
	L	80.15%	**80.47**%	78.10%	77.77%
	N	**79.35**%	78.83%	76.78%	76.45%
SVMs	W	81.53%	82.72%	83.18%	**83.71**%
	L	84.10%	**84.56**%	83.58%	83.11%
	N	81.40%	**82.58**%	81.60%	81.99%
k-NN	W	37.80%	**54.75**%	38.32%	40.96%
	L	50.66%	40.11%	58.91%	**59.17**%
	N	61.08%	69.53%	70.84%	**72.16**%

Table 2. Accuracy rates before applying SVD

As shown in Table 2, the best result has been obtained by using SVMs after removing words that appear in only 1 document (>1 doc) and using the lemmatized corpus (84.56 %). We want to emphasize that, taking into account the morphosyntactic features of Basque and the reduced corpora used, the accuracy rates obtained with this method are high for all the three corpora. In fact, they are as good as some results reported for other similar corpora and language features [15]. Results obtained using Naïve Bayes and Winnow are also very good. Both have been obtained using SNoW, and we argue that the processing it performs is very adequate for text categorization tasks. Both work better with more attributes, in general. Moreover, we can see that lemmatization and noun selection help Naïve Bayes in general, but this is not the case for Winnow.

However, results show that *k*-NN algorithm is not suitable for text categorization using raw data, even though noun selection gives acceptable accuracy rates (72.16 % the best). Results in the table was obtained for different *k* values (*k*=1,..., 10), and using the Euclidean distance. Finally, we want to state that most of the best accuracy rates have been obtained by eliminating words that only appear in one document (> 1 doc case).

4.2. Experiment after applying SVD

In this second experiment, LSI has been used to create the three reduced spaces for the training document collections. The sizes of the training matrices created are 34288×4548 (W), 14648×4548 (L) and 7209×4548 (N). The selection of terms is made automatically by LSI. Different number of dimensions have been experimented $(p = 100, 200, 300, 400, 500, 1000)$. The weighting scheme used has been logarithm for local weighting and entropy for global one.

When using *k*-NN, different experiments for different number of neighbours ($k = 1, \cdots, 10$) have been made and the following criteria has been followed: regarding the categories of the *k* closest (with the highest cosine), the most frequent one was selected. In case the result is a tie, the category with the highest mean is chosen.

		LSI dim.	Accuracy
SVD+SVMs	W	1000	75.00%
	L	500	**81.46**%
	N	500	80.34%
SVD+*k*-NN	W	300	84.89%
	L	400	**87.33**%
	N	200	85.36%

Table 3. Accuracy rates for SVMs and *k*-NN after SVD

	100	200	300	400	500
W	82.98%	84.30%	**84.89**%	84.76%	84.63%
L	85.95%	86.61%	86.81%	**87.33**%	87.07%
N	84.37%	**85.36**%	84.83%	85.03%	84.76%

Table 4. SVD + *k*-NN accuracy rates.

The best results in this experiment have been obtained by using *k*-NN In Table 3 the best result for each corpus is shown, and it can be observed that, using *k*-NN they are all superior to the best results obtained in the previous experiment for each of the corpus and classification method. The highest accuracy rate has been obtained for the lemmatized corpus, which significantly improves and increases up to 87.33 %. This confirms our hypothesis that lemmatization helps improving results in agglutinative languages such as Basque. Selecting nouns also gives better results than word-forms, but they do not give the best ones. However, when SVMs are used after applying SVD, results become poorer. This is because SVMs are good when the number of features is high, and consequently, the dimensionality reduction does not benefit to them.

We have also used Naïve Bayes and Winnow to categorize the documents after applying SVD, but we do not include the results obtained in Table 3 because they are quite

worse that the ones obtained before applying SVD. The reason may be that the way SNoW treats data makes it adequate to work with raw texts instead of with the reduced dimensional vectors obtained after the SVD.

Finally, given that the best results have been obtained by combining SVD and *k*-NN, we consider interesting to show all the accuracy rates obtained for different dimensions and number of neighbous. In Table 4 the results for the best *k* are shown: *k*=10 (W) and *k*=3 (L)(N). We want to remark that when the lemmatized corpus is used, the results for every dimensionality experimented increase the best result met before applying SVD (84.56 % in Table 2).

5. Conclusions and future work

In this paper we have shown the foundations and results of an experiment conducted to validate different methods for categorizing documents written in Basque. In our opinion, the most important conclusion is that combining SVD for dimensionality reduction and the *k*-NN algorithm yields to an important improvement in the categorization accuracy rate. We would like to emphasize that when lemmatization is used, results increase up to 87.33%. For future work, we intend to test other combinations of methods constructing a multi-classifier [18] and trying to perform a more sophisticated feature selection technique [19,12] over the features given by the SVD. Finally, we intend to confirm the good results of combining LSI and *k*-NN algorithm for other languages and corpora (Reuters-21578), and to outperform the results with the techniques proposed as future work.

References

[1] I. ALEGRIA, X. ARTOLA, K. SARASOLA and M. URKIA: Automatic morphological analysis of basque. *Literary & Linguistic Computing*, **11** (1996).

[2] M.W. BERRY and M. BROWNE: Understanding Search Engines: Mathematical Modeling and Text Retrieval. Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia, 1999.

[3] M.W. BERRY, S.T. DUMAIS and G.W. O'BRIEN: Using linear algebra for intelligent information retrieval. *SIAM Review*, **37**(4), (1995), 573-595.

[4] A.J. Carlson, C.M. Cumby, J.L. Rosen, and D. Roth. Snow. *UIUC Tech report UIUC-DCS-R-99-210*, 1999. University of Illinois.

[5] I. Dagan, Y. Karov, and D. Roth. Mistake-driven learning in text categorization. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 55–63, 1997.

[6] B.V. Dasarathy. Nearest neighbor (nn) norms: Nn pattern recognition classification techniques. *IEEE Computer Society Press*, 1991.

[7] 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*, 41:391–407, 1990.

[8] R. Dolin, J. Pierre, M. Butler, and R. Avedon. Practical evaluation of ir within automated classification systems. *Proceedings of the International Conference on Information and Knowledge Management CIKM*, pages 322–329, November 1999.

[9] S. Dumais. Latent semantic analysis. *ARIST (Annual Review of Information Science Technology)*, 38:189–230, 2004.

[10] S.T. Dumais. Using lsi for information filtering: Trec-3 experiments. In D. Harman, editor, *Third Text REtrieval Conference (TREC3)*, pages 219–230, 1995.

[11] N. Ezeiza, I. Aduriz, I. Alegria, J.M. Arriola, and R. Urizar. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. *COLING-ACL'98*, 1998.

[12] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. Feature subset selection by bayesian network-based optimization. *Artificial Intelligence*, 123:157–184, 2000.

[13] T. Joachims. Transductive inference for text classification using support vector machines. *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209, 1999.

[14] M. Minsky. Steps toward artificial intelligence. In *Proceedings of the Institute of Radio Engineers*, volume 49, pages 8–30, 1961.

[15] P. Nakov, E. Valchanova, and G. Angelova. Towards deeper understanding of the lsa performance. In *Proc. of the Int. Conference RANLP-03 "Recent Advances in Natural Language Processing"*, pages 311–318, Bulgaria, 2003.

[16] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

[17] I.H. Witten and E. Frank. Data mining. practical machine learning tools and techniques with java implementations. *Morgan Kaufmann Publishers*, 1999.

[18] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[19] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In Morgan Kaufmann, editor, *Proceedings of the Fourteenth International Conference on Machine Learning, ICML'97*, pages 412–420, 1997.

EXPLORING BASQUE DOCUMENT CATEGORIZATION FOR EDUCATIONAL PURPOSES USING LSI

A. Zelaia, I. Alegria, O. Arregi, A. Arruarte, A. Díaz de Ilarraza, J.A. Elorriaga and B. Sierra
University of the Basque Country, UPV-EHU, Spain

Keywords: Document Categorization, Latent Semantic Indexing (LSI), Computer Supported Learning Systems (CSLSs), Domain Module.

Abstract: In the process of preparing learning material for Computer Supported Learning Systems (CSLSs), one of the first steps involves finding documents relevant to the topics and to the students. This requires documents to be categorized according to some criteria. In this paper we analyze the behaviour of classification techniques such as Naïve Bayes, Winnow, SVMs and *k*-NN, together with lemmatization and noun selection, in the categorization of documents written in Basque. In a second experiment, we study the effect of applying the Singular Value Decomposition (SVD) dimensionality reduction technique before using the mentioned classification techniques. The results obtained show that the approach which combines SVD and *k*-NN for a lemmatized corpus gives the best categorization of all with a remarkable difference. The final aim pursued in this project is to facilitate the semiautomatic construction of the domain module of a CSLS.

1 INTRODUCTION

In the Information Age, learning occurs in contexts where information and knowledge are constantly changing. Finding documents relevant to topics and to the users involves one of the first steps in the process of preparing learning material (Vereoustre and McLean, 2003). Learning is opportunistic. It occurs in dynamic environments where new information, processes and people are appearing and disappearing. Current electronic document search engines do not provide a reasonable answer to most people's opportunistic learning needs. Following the guidelines set in (Aleven et al., 2003), the paper here presented tries to establish synergies between research occurring in the fields of Artificial Intelligence in Education and Electronic Document Technologies.

In Computer Supported Learning Systems (CSLSs) one of the main components is the *domain module*, where the subject to be learnt is modelled. The final aim of our project is to facilitate the construction of the domain module in a semi-automatic way. The process of creating it implies first the identification of learning material, i.e. the selection of the appropriate documents. This requires documents to be categorized according to some educational criteria. Most researchers propose approaches based on machine learning techniques, where automatically built classifiers learn from a set of previously classified documents. In our experiments, we use four classification techniques which have reported good results for categorizing documents: Naïve Bayes, Winnow, SVMs and *k*-NN.

Several experiments have been made to classify documents written in extended languages such as English. But, the reality of lesser-used languages, as it is the case of Basque, is different. In practice, one of the main problems we encounter is that only a short amount of manually classified documents is available. This fact restricts the capacity of the classifier and may, consequently, produce poorer results. In addition, we find that for educational use, even for extended languages such as English, there is no educational collection of documents (Nakayama and Shimizu, 2003). Taking this fact into account we have decided to separate the experimental design into two steps. In the first step, presented in this paper, we analyze the behaviour of the classification algorithms using documents which correspond to a Basque newspaper and which are categorized according to a recognized standard classification. In a future second phase, we will analyze the behaviour of the classification techniques using an educational corpus for Basque, which will have to be previously constructed.

Zelaia A., Alegria I., Arregi O., Arruarte A., Díaz de Ilarraza A., Elorriaga J. and Sierra B. (2009).
EXPLORING BASQUE DOCUMENT CATEGORIZATION FOR EDUCATIONAL PURPOSES USING LSI.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 5-9
DOI: 10.5220/0001834300050009

There is also another reason because of which we have to make a special effort in our classification task; the morphosintactical features of Basque. In fact, we must take into account that Basque is an agglutinative language whose declension system has numerous cases (Alegria et al., 1996). This makes the categorization task even more difficult, because semantic information is not really contained in word-forms but in their corresponding lemma. Thus, the categorization of documents written in Basque turns out to be challenging. In our experiments we analyze the effect of preprocessing the corpus in order to reduce the dimension of the information to treat. In this way, we analyze dimensionality reduction techniques such as lemmatization, noun selection and Singular Value Decomposition (SVD).

In this work we perform two experiments. In the first one, we apply the classification techniques to three different corpora. In the second experiment, we apply the SVD dimensionality reduction technique by means of Latent Semantic Indexing[1] (LSI) implementation, before applying the classification techniques to the same corpora.

This paper is structured as follows. In Section 2 the research context is presented. In Section 3, we reference the classification algorithms used in our experiments, and examine the applications of LSI in text categorization problems and for educational purposes. In Section 4 the experimental setup is introduced, where both training and test corpora are described and lemmatization and noun selection processes are introduced. In Section 5, experimental results are shown, compared and discussed. Finally, Section 6 contains some conclusions and comments on future work.

2 RESEARCH CONTEXT

This work is part of a project that aims to acquire semiautomatically the domain for CSLSs. Concretely, the system that is being developed takes an electronic document as the base for building the Domain Module (Larrañaga et al., 2003). This module is enriched with additional documents and other didactic material. The process is divided into three different phases: Domain Module structure acquisition, generation of didactic material and domain enrichment and maintenance. In the Domain Module structure acquisition phase, first the document table of contents is analyzed obtaining the main topics of the domain and the relations among them. These topics and their relations constitute the first version of the domain ontology. Once the initial process has been finished, the whole document is analyzed in order to look for new topics and relations. The generation of didactic material is an ontology-driven analysis which splits out the whole document into Learning Objects (homepage, 2001) categorizing them according to some pedagogical purpose.

Finally, in order to enrich the Domain Module with more didactic material and to maintain it up to date, new documents are analyzed and incorporated to the domain module. The work presented in this paper will help in this last phase. Document classification will allow to connect the new documents to the concepts of the domain.

3 CLASSIFICATION TECHNIQUES AND LATENT SEMANTIC INDEXING (LSI)

Text categorization consists in assigning predefined categories to text documents (Sebastiani, 2005). When the bag-of-words text document representation is used, the number of attributes in the corpus is usually considerable, and this can be problematic in inductive classification. Therefore, it is usually convenient to apply techniques that reduce the dimension of the representation. This reduction can be carried out in different ways: eliminating irrelevant features (terms), substituting some words by others that represent them (lemmas, etc.), applying SVD technique, etc.

The SVD technique compresses vectors representing documents into vectors of a lower-dimensional space (Berry and Browne, 1999). This operation is called dimensionality reduction, and the space to which document vectors are projected is called the reduced space. When using the reduced space, most of the important underlying structure that associates terms with documents is captured and consequently, noise is reduced.

In our experiments we use LSI (Deerwester et al., 1990) (Dumais, 2004) to calculate the SVD and the cosine similarity measure among the document to be categorized and all the documents in the reduced space (training set). LSI has been successfully used in the categorization of documents written in english (Dolin et al., 1999) (Dumais, 1995). It has also been used for a variety of educational applications, such as the representation of knowledge in CSLSs (Zampa and Lemaire, 2002), tutoring dialog (Graesser et al.,

[1] http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

2001) and automatic essay grading (Miller, 2003).

We use classification algorithms which have reported good results for text categorization in other languages; in this way, we use Naïve Bayes (Minsky, 1961), Winnow (Dagan et al., 1997), SVMs (Joachims, 1999) and *k*-NN (Dasarathy, 1991).

4 EXPERIMENTAL SETUP

The purpose of this section is to describe the document collection used in our experiments and to give an account of the lemmatization, noun selection and feature selection techniques we have applied.

4.1 Document Collection

As we have pointed out in the introduction, we are interested in the categorization of documents written in Basque with educational purposes. The ideal would be to have available an educational collection of documents categorized according to some standard labelling, but there is neither such educational corpus nor a standard educational classification. Among all the electronic documents available in Basque, we have selected newspaper texts, because there are standardized categories for this domain, and we have access to a sufficient amount of documents manually categorized. This will allow us to analyze the behaviour of the selected classification techniques when applied to Basque documents.

The documents used in this experiment correspond to the *Euskaldunon Egunkaria* newspaper, corresponding to the articles published during two months of 1999. They are a total of 6.064 documents categorized to the 17 standard first level IPTC categories[2]. Each of the documents has a unique category associated to it. It must be noted that all categories do not have the same number of documents, as can be seen in Table 1.

Document categorization is achieved in two steps: during the *training* step an inductive generalization of the set of documents is obtained, and during the *test* step the effectiveness of the system is measured. Therefore, the 6,064 documents have been split into two different sets of documents: 4,548 documents for training (75 %) and 1,516 documents for testing (25 %). This proportion stands in each one of the 17 categories, as can be observed in Table 1.

[2]http://www.iptc.org

Table 1: Number of documents distributed by categories.

Category	Training	Test
1. Culture	600	202
2. Justice	129	42
3. Disasters	75	26
4. Economy	234	78
5. Education	82	27
6. Environmental Issues	69	22
7. Health	35	12
8. Human interests	36	11
9. Labour	132	43
10. Lifestyle	40	13
11. Politics	1.184	393
12. Religion	25	8
13. Science	35	12
14. Social Issues	464	156
15. Sport	1.283	429
16. Conflicts	100	33
17. Weather	25	9
TOTAL	4.548	1.516

4.2 Feature Selection. Lemmatization

Basque is an agglutinative and highly inflected language. In order to face the difficulties derived from these morphosintactical features, we have applied two types of feature selection techniques. On the one hand, stopword lists have been used to eliminate non-relevant words, i.e. the most and least frequent words in the training corpus. On the other hand, we use linguistic methods such as lemmatization and noun selection to reduce the number of features. Indeed, recent experiments show that lemmatization helps in the process of categorizing documents written in an inflected language using LSI (Nakov et al., 2003). Therefore, we expect that lemmatization, and noun selection in particular, should allow us to maintain the same semantic information, reducing the number of attributes to be processed.

We have used the Basque lemmatizer designed by the IXA natural language processing group (Ezeiza et al., 1998), which obtains for each word in the document, its corresponding lemma, as well as its part-of-speech tag. This system reduces the different number of features from each category by more than 50%.

5 EXPERIMENTAL RESULTS

In this section we show the results obtained in the two experiments. In both of them we use the general-purpose classifier named SNoW (Carlson et al., 1999) for Naïve Bayes and Winnow algorithms and Weka

Table 2: Accuracy rates before applying SVD.

		all	>1	>2	>3
Naïve Bayes	Words	**80.09**	78.89	78.10	77.77
	Lemmas	**81.53**	81.07	80.74	80.28
	Nouns	79.49	**79.62**	79.35	79.62
Winnow	Words	80.09	**81.13**	80.47	79.49
	Lemmas	80.15	**80.47**	78.10	77.77
	Nouns	**79.35**	78.83	76.78	76.45
SVMs	Words	81.53	82.72	83.18	**83.71**
	Lemmas	84.10	**84.56**	83.58	83.11
	Nouns	81.40	**82.58**	81.60	81.99
k-NN	Words	37.80	**54.75**	38.32	40.96
	Lemmas	50.66	40.11	58.91	**59.17**
	Nouns	61.08	69.53	70.84	**72.16**

(Witten and Frank, 2005) for SVMs. We apply the classification algorithms to three different corpora: a corpus of text documents (words), a second one of lemmatized documents and a third one in which only nouns appearing in documents have been kept.

5.1 Experiment before Applying SVD

In this experiment, elimination of irrelevant words, lemmas and nouns has been performed based on the word frequency in documents; terms that appear in more than 1, 2 or 3 documents (>1, >2, etc.) are kept. The accuracy rates using the test-corpus for each classification technique are shown in Table 2. The best results obtained for each technique and corpus appear printed in boldface.

As shown in Table 2, the best result has been obtained by using SVMs after removing words that appear in only 1 document (>1) and using the lemmatized corpus (84.56 %). We want to emphasize that, taking into account the morphosintactical features of Basque and the reduced corpora used, the accuracy rates obtained with this method are high for all the three corpora. In fact, they are as good as some results reported for other similar corpora and language features (Nakov et al., 2003).

Results obtained using Naïve Bayes and Winnow are also very good. Both have been obtained using SNoW, and we argue that the processing it performs is very adequate for text categorization tasks. Both work better with more attributes, in general. Moreover, we can see that lemmatization and noun selection help Naïve Bayes in general, but this is not the case for Winnow.

However, results show that *k*-NN algorithm is not suitable for text categorization using raw data, even though noun selection gives acceptable accuracy rates (72.16 % the best). The accuracy rates in the table have been obtained for different k values (k=1,..., 10), and using the Euclidean distance.

Table 3: Accuracy rates for SVMs and *k*-NN after SVD.

		LSI dim.	Accuracy
SVD+SVMs	Words	1000	75.00%
	Lemmas	500	**81.46**%
	Nouns	500	80.34%
SVD+*k*-NN	Words	300	84.89%
	Lemmas	400	**87.33**%
	Nouns	200	85.36%

5.2 Experiment after Applying SVD

In this second experiment, LSI has been used to create the three reduced spaces for the training document collections. Different number of dimensions have been experimented $(100, 200, 300, 400, 500, 1000)$. The weighting scheme used has been logarithm for local weighting and entropy for global one.

When using *k*-NN, different experiments for different number of neighbours ($k = 1, \cdots, 10$) have been made and the following criteria has been followed: regarding the categories of the k closest (with the highest cosine), the most frequent one was selected. In case the result is a tie, the category with the highest mean is chosen.

The best results in this experiment have been obtained by using *k*-NN. In Table 3 the best result for each corpus is shown, and it can be observed that, the highest accuracy rate has been obtained for the lemmatized corpus, which significantly improves and increases up to 87.33 %. This confirms our hypothesis that lemmatization helps improving results in agglutinative languages such as Basque. Selecting nouns also gives better results than word-forms, but they do not give the best ones.

However, when SVMs are used after applying SVD, results become poorer. This is because SVMs are good enough when the number of features is high, and consequently, the dimensionality reduction does not benefit to them.

We have also used Naïve Bayes and Winnow to categorize the documents after applying SVD, but we do not include the results in Table 3 because they are fairly worse that the ones obtained before applying SVD. The reason may be that the way SNoW treats data makes it adequate to work with raw texts instead of with the reduced dimensional vectors obtained after the SVD.

Finally, given that the best results have been obtained by combining SVD and *k*-NN, we consider interesting to show all the accuracy rates obtained for different dimensions and number of neighbours. In Table 4 the results for the best k are shown: k=10 (Words) and k=3 (Lemmas and Nouns).

Table 4: SVD + k-NN accuracy rates for Words, Lemmas and Nouns.

	100	200	300	400	500
W.	82.98	84.30	**84.89**	84.76	84.63
L.	85.95	86.61	86.81	**87.33**	87.07
N.	84.37	**85.36**	84.83	85.03	84.76

6 CONCLUSIONS AND FUTURE WORK

Along this paper, we have analyzed the categorization of documents written in Basque with the purpose of facilitating the construction of the domain module in a CSLS. This work constitutes an important step in the process of semi-automatically acquiring the domain module of CSLSs. The two experiments performed in this study show that advances in the field of Electronic Document Technologies can find interesting applications in the field of Artificial Intelligence in Education. Results demonstrate that the k-NN classification algorithm combined with the SVD dimensionality reduction technique gives very good results even for a lesser-used and highly inflected language such as Basque. We would like to emphasize that when lemmatization is used, results increase up to 87.33%.

In our experiments we have confirmed that categorization results are also good when documents are written in Basque. This will permit us to face the Basque document categorization problem for an educational environment in a more established way. It will be a great advance in the process of constructing the domain module for CSLSs in a semi-automatic way. However, the lack of a Basque educational collection of documents makes this first step of acquisition of learning material be harder. Our future work will be conducted to construct such a corpus (Ghani et al., 2001) and repeat the experiments in order to confirm the good results.

Regarding the domain acquisition task, we are currently working in the automatic extraction of the main topics and the pedagogical relations among them represented, explicitly or implicitly, in the table of contents of a document. A set of heuristics that infer such relations and the part-of-speech information have been already defined (Larrañaga et al., 2004) (Larrañaga et al., 2008).

ACKNOWLEDGEMENTS

This work is supported by the MEC (TIN2006-14968-C02-01) and by the University of the Basque Country (UE06/19).

REFERENCES

Alegria, I., Artola, X., Sarasola, K., and Urkia, M. (1996). Automatic morphological analysis of basque. *Literary & Linguistic Computing*, 11.

Aleven, V., Hoppe, U., Kay, J., Mizoguchi, R., Pain, H., Verdejo, F., and Yacef, K., editors (2003). *Technologies for Electronic Documents for Supporting Learning*.

Berry, M. and Browne, M. (1999). *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia.

Carlson, A., Cumby, C., Rosen, J., and Roth, D. (1999). Snow. *UIUC Tech report UIUC-DCS-R-99-210*. University of Illinois.

Dagan, I., Karov, Y., and Roth, D. (1997). Mistake-driven learning in text categorization. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 55–63.

Dasarathy, B. (1991). Nearest neighbor (nn) norms: Nn pattern recognition 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*, 41:391–407.

Dolin, R., Pierre, J., Butler, M., and Avedon, R. (1999). Practical evaluation of ir within automated classification systems. *Proceedings of the International Conference on Information and Knowledge Management CIKM*, pages 322–329.

Dumais, S. (1995). Using lsi for information filtering: Trec-3 experiments. In Harman, D., editor, *Third Text REtrieval Conference (TREC3)*, pages 219–230.

Dumais, S. (2004). Latent semantic analysis. *ARIST (Annual Review of Information Science Technology)*, 38:189–230.

Ezeiza, N., Aduriz, I., Alegria, I., Arriola, J., and Urizar, R. (1998). Combining stochastic and rule-based methods for disambiguation in agglutinative languages. *COLING-ACL'98*.

Ghani, R., Jones, R., and Mladenic, D. (2001). Using the web to create minority language corpora. In *International Conference on Infroamtion and Knowledge Management (CIKM 2001)*.

Graesser, A., Person, N., Harter, D., and Group, T. T. R. (2001). Teaching tactics and dialog in autotutor. *International Journal of Artificial Intelligence in Education*, 12(3):257–279.

homepage, L. L. O. M. W. G. (2001). IEEE P1484.12. http://ltsc.ieee.org/wg12/.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of*

ICML-99, 16th International Conference on Machine Learning, pages 200–209.

Larrañaga, M., Elorriaga, J., and Arruarte, A. (2008). A heuristic nlp based approach for getting didactic resources from electronic documents. In *Proceedings of the 3th Europeanl Conference on Technology-Enhanced Learning, Springer, LNCS 5192*, pages 197–202.

Larrañaga, M., Rueda, U., Elorriaga, J., and Arruarte, A. (2003). Index analysis: A means to acquire the domain module structure. In *X CAEPIA - V TTIA*, volume II, pages 339–342.

Larrañaga, M., Rueda, U., Elorriaga, J., and Arruarte, A. (2004). Acquisition of the domain structure from document indexes using heuristic reasoning. In Lester, J., Vicari, R., and Paraguacu, F., editors, *Intelligent Tutoring Systems, LNCS 3220*, pages 175–186.

Miller, T. (2003). Essay assessment with latent semantic analysis. *Journal of Educational Computing Research*, 28.

Minsky, M. (1961). Steps toward artificial intelligence. In *Proceedings of the Institute of Radio Engineers*, volume 49, pages 8–30.

Nakayama, M. and Shimizu, Y. (2003). Subject categorization for web educational resources using mlp. In *European Symposium on Artificial Neural Networks, ESANN'2003*, pages 9–14.

Nakov, P., Valchanova, E., and Angelova, G. (2003). Towards deeper understanding of the lsa performance. In *Proc. of the Int. Conference RANLP-03 "Recent Advances in Natural Language Processing"*, pages 311–318, Bulgaria.

Sebastiani, F. (2005). Text categorization. *Text Mining and its Applications*, pages 109–129.

Vereoustre, A. and McLean, A. (2003). Reusing educational material for teaching and learning: Current approaches and directions. In Aleven, V., Hoppe, U., Kay, J., Mizoguchi, R., Pain, H., Verdejo, F., and Yacef, K., editors, *Supplementary Proceedings of AIED2003*, pages 621–630.

Witten, I. and Frank, E. (2005). Data mining. practical machine learning tools and techniques. *Morgan Kaufmann Publishers*.

Zampa, V. and Lemaire, B. (2002). Latent semantic analysis for user modeling. *Journal of Intelligent Information Systems. Special Issue on Education Applications.*, 18(1):15–30,.

A Multiclassifier based Document Categorization System: profiting from the Singular Value Decomposition Dimensionality Reduction Technique

Ana Zelaia | **Iñaki Alegria** | **Olatz Arregi** | **Basilio Sierra**
UPV-EHU | UPV-EHU | UPV-EHU | UPV-EHU
Basque Country | Basque Country | Basque Country | Basque Country
ccpzejaa@si.ehu.es | acpalloi@si.ehu.es | acparuro@si.ehu.es | ccpsiarb@si.ehu.es

Abstract

In this paper we present a multiclassifier approach for multilabel document classification problems, where a set of k-NN classifiers is used to predict the category of text documents based on different training subsampling databases. These databases are obtained from the original training database by random subsampling. In order to combine the predictions generated by the multiclassifier, Bayesian voting is applied. Through all the classification process, a reduced dimension vector representation obtained by Singular Value Decomposition (SVD) is used for training and testing documents. The good results of our experiments give an indication of the potentiality of the proposed approach.

1 Introduction

Document Categorization, the assignment of natural language texts to one or more predefined categories based on their content, is an important component in many information organization and management tasks. Researchers have concentrated their efforts in finding the appropriate way to represent documents, index them and construct classifiers to assign the correct categories to each document. Both, document representation and classification method are crucial steps in the categorization process.

In this paper we concentrate on both issues. On the one hand, we use Latent Semantic Indexing (LSI) (Deerwester et al., 1990), which is a variant of the vector space model (VSM) (Salton and McGill, 1983), in order to obtain the vector representation of documents. This technique compresses vectors representing documents into vectors of a lower-dimensional space. LSI, which is based on Singular Value Decomposition (SVD) of matrices, has showed to have the ability to extract the relations among words and documents by means of their context of use, and has been successfully applied to Information Retrieval tasks.

On the other hand, we construct a multiclassifier (Ho et al., 1994) which uses different training databases. These databases are obtained from the original training set by random subsampling. We implement this approach by bagging, and use the k-NN classification algorithm to make the category predictions for testing documents. Finally, we combine all predictions made for a given document by Bayesian voting.

The experiment we present has been evaluated for Reuters-21578 standard document collection. Reuters-21578 is a multilabel document collection, which means that categories are not mutually exclusive because the same document may be relevant to more than one category. Being aware of the results published in the most recent literature, and having obtained good results in our experiments, we consider the categorization method presented in this paper an interesting contribution for text categorization tasks.

The remainder of this paper is organized as follows: Section 2, discusses related work on document categorization for Reuters-21578 collection. In Section 3, we present our approach to deal with the multilabel text categorization task. In Section 4 the experimental setup is introduced, and details about the Reuters database, the preprocessing applied and some parameter setting are provided. In Section 5, experimental results are presented and discussed. Finally, Section 6 contains some conclusions and comments on future work.

2 Related Work

As previously mentioned in the introduction, text categorization consists in assigning predefined categories to text documents. In the past two decades, document categorization has received much attention and a considerable number of machine learning based approaches have been proposed. A good tutorial on the state-of-the-art of document categorization techniques can be found in (Sebastiani, 2002).

In the document categorization task we can find two cases; (1) the multilabel case, which means that categories are not mutually exclusive, because the same document may be relevant to more than one category (1 to m category labels may be assigned to the same document, being m the total number of predefined categories), and (2) the single-label case, where exactly one category is assigned to each document. While most machine learning systems are designated to handle multiclass data[1], much less common are systems that can handle multilabel data.

For experimentation purposes, there are standard document collections available in the public domain that can be used for document categorization. The most widely used is Reuters-21578 collection, which is a multiclass (135 categories) and multilabel (the mean number of categories assigned to a document is 1.2) dataset. Many experiments have been carried out for the Reuters collection. However, they have been performed in different experimental conditions. This makes results difficult to compare among them. In fact, effectiveness results can only be compared between studies that use the same training and testing sets. In order to lead researchers to use the same training/testing divisions, the Reuters documents have been specifically tagged, and researchers are encouraged to use one of those divisions. In our experiment we use the "ModApte" split (Lewis, 2004).

In this section, we analize the category subsets, evaluation measures and results obtained in the past and in the recent years for Reuters-21578 ModApte split.

2.1 Category subsets

Concerning the evaluation of the classification system, we restrict our attention to the TOPICS group of categories that labels Reuters dataset, which contains 135 categories. However, many categories appear in no document and consequently, and because inductive based learning classifiers learn from training examples, these categories are not usually considered at evaluation time. The most widely used subsets are the following:

- Top-10: It is the set of the 10 categories which have the highest number of documents in the training set.
- R(90): It is the set of 90 categories which have at least one document in the training set and one in the testing set.
- R(115): It is the set of 115 categories which have at least one document in the training set.

In order to analyze the relative hardness of the three category subsets, a very recent paper has been published by Debole and Sebastiani (Debole and Sebastiani, 2005) where a systematic, comparative experimental study has been carried out.

The results of the classification system we propose are evaluated according to these three category subsets.

2.2 Evaluation measures

The evaluation of a text categorization system is usually done experimentally, by measuring the effectiveness, i.e. average correctness of the categorization. In binary text categorization, two known statistics are widely used to measure this effectiveness: precision and recall. Precision (Prec) is the percentage of documents correctly classified into a given category, and recall (Rec) is the percentage of documents belonging to a given category that are indeed classified into it.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines precision and recall. Various such measures have been proposed. The breakeven point, the value at which precision equals recall, has been frequently used during the past decade. However, it has been recently criticized by its proposer ((Sebastiani, 2002) footnote 19). Nowadays, the F_1 score is more frequently used. The F_1 score combines recall and precision with an equal weight in the following way:

$$F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

[1] Categorization problems where there are more than two possible categories.

Since precision and recall are defined only for binary classification tasks, for multiclass problems results need to be averaged to get a single performance value. This will be done using *microaveraging* and *macroaveraging*. In microaveraging, which is calculated by globally summing over all individual cases, categories count proportionally to the number of their positive testing examples. In macroaveraging, which is calculated by averaging over the results of the different categories, all categories count the same. See (Debole and Sebastiani, 2005; Yang, 1999) for more detailed explanation of the evaluation measures mentioned above.

2.3 Comparative Results

Sebastiani (Sebastiani, 2002) presents a table where lists results of experiments for various training/testing divisions of Reuters. Although we are aware that the results listed are microaveraged breakeven point measures, and consequently, are not directly comparable to the ones we present in this paper, F_1, we want to remark some of them. In Table 1 we summarize the best results reported for the ModApte split listed by Sebastiani.

Results reported by	R(90)	Top-10
(Joachims, 1998)	86.4	
(Dumais et al., 1998)	87.0	92.0
(Weiss et.al., 1999)	87.8	

Table 1: Microaveraged breakeven point results reported by Sebastiani for the Reuters-21578 ModApte split.

In Table 2 we include some more recent results, evaluated according to the microaveraged F_1 score. For R(115) there is also a good result, $F_1 = 87.2$, obtained by (Zhang and Oles, 2001)[2].

3 Proposed Approach

In this paper we propose a multiclassifier based document categorization system. Documents in the training and testing sets are represented in a reduced dimensional vector space. Different training databases are generated from the original training dataset in order to construct the multiclassifier. We use the k-NN classification algorithm, which according to each training database makes a prediction for testing documents. Finally, a Bayesian voting scheme is used in order to definitively assign category labels to testing documents.

[2]Actually, this result is obtained for 118 categories which correspond to the 115 mentioned before and three more categories which have testing documents but no training document assigned.

Results reported by	R(90)	Top-10
(Gao et al., 2003)	88.42	93.07
(Kim et al., 2005)	87.11	92.21
(Gliozzo and Strapparava, 2005)		92.80

Table 2: F_1 results reported for the Reuters-21578 ModApte split.

In the rest of this section we make a brief review of the SVD dimensionality reduction technique, the k-NN algorithm and the combination of classifiers used.

3.1 The SVD Dimensionality Reduction Technique

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization tasks. The newer method of Latent Semantic Indexing (LSI) [3] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional space created from the input training dataset. It is based on the assumption that there is some underlying latent semantic structure in the term-document matrix that is corrupted by the wide variety of words used in documents. This is referred to as the problem of polysemy and synonymy. The basic idea is that if two document vectors represent two very similar topics, many words will co-occur on them, and they will have very close semantic structures after dimension reduction.

The SVD technique used by LSI consists in factoring term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values in non-increasing order, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively). Matrix M can be approximated by a lower rank M_p which is calculated by using the p largest singular values of M. This operation is called dimensionality reduction, and the p-dimensional

[3]http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

space to which document vectors are projected is called the reduced space. Choosing the right dimension p is required for successful application of the LSI/SVD technique. However, since there is no theoretical optimum value for it, potentially expensive experimentation may be required to determine it (Berry and Browne, 1999).

For document categorization purposes (Dumais, 2004), the testing document q is also projected to the p-dimensional space, $q_p = q^T U_p \Sigma_p^{-1}$, and the cosine is usually calculated to measure the semantic similarity between training and testing document vectors.

In Figure 1 we can see an ilustration of the document vector projection. Documents in the training collection are represented by using the term-document matrix M, and each one of the documents is represented by a vector in the $\mathbb{R}^m$ vector space like in the traditional vector space model (VSM) scheme. Afterwards, the dimension p is selected, and by applying SVD vectors are projected to the reduced space. Documents in the testing collection will also be projected to the same reduced space.

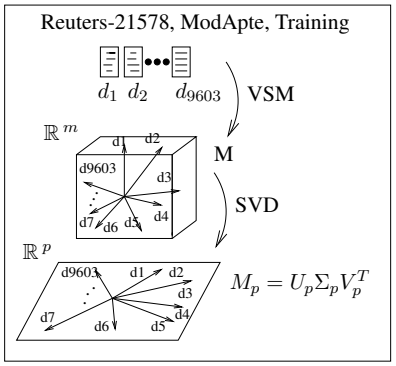


Figure 1: Vectors in the VSM are projected to the reduced space by using SVD.

3.2 The k nearest neighbor classification algorithm (k-NN)

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing document, the k-NN classifier ranks its nearest neighbors among the training documents, and uses the categories of the k top-ranking neighbors to predict the categories of the testing document (Dasarathy, 1991). In this paper, the training and testing documents are represented as reduced dimensional vectors in the lower dimensional space, and in order to find the nearest neighbors of a given document, we calculate the cosine similarity measure.

In Figure 2 an ilustration of this phase can be seen, where some training documents and a testing document q are projected in the $\mathbb{R}^p$ reduced space. The nearest to the q_p testing document are considered to be the vectors which have the smallest angle with q_p. According to the category labels of the nearest documents, a category label prediction, c, will be made for testing document q.

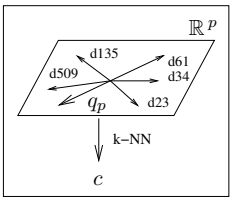


Figure 2: The k-NN classifier is applied to q_p testing document and c category label is predicted.

We have decided to use the k-NN classifier because it has been found that on the Reuters-21578 database it performs best among the conventional methods (Joachims, 1998; Yang, 1999) and because we have obtained good results in our previous work on text categorization for documents written in Basque, a highly inflected language (Zelaia et al., 2005). Besides, the k-NN classification algorithm can be easily adapted to multilabel categorization problems such as Reuters.

3.3 Combination of classifiers

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). Two widely used techniques to implement this approach are *bagging* (Breiman, 1996), that uses more than one model of the same paradigm; and *boosting* (Freund and Schapire, 1999), in which a different weight is given to different training examples looking for a better accuracy.

In our experiment we have decided to construct a multiclassifier via bagging. In bagging, a set of training databases TD_i is generated by selecting n training examples drawn randomly with replacement from the original training database TD of n examples. When a set of n_1 training examples,

$n_1 < n$, is chosen from the original training collection, the bagging is said to be applied by random subsampling. This is the approach used in our work. The n_1 parameter has been selected via tuning. In Section 4.3 the selection will be explained in a more extended way.

According to the random subsampling, given a testing document q, the classifier will make a label prediction c^i based on each one of the training databases TD_i. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}^i_{c_j}$ is calculated for each training database TD_i and category c_j to be predicted. These confidence values have been calculated based on the original training collection. Confidence values are summed by category. The category c_j that gets the highest value is finally proposed as a prediction for the testing document.

In Figure 3 an ilustration of the whole experiment can be seen. First, vectors in the VSM are projected to the reduced space by using SVD. Next, random subsampling is applied to the training database TD to obtain different training databases TD_i. Afterwards the k-NN classifier is applied for each TD_i to make category label predictions. Finally, Bayesian voting is used to combine predictions, and c_j, and in some cases c_k as well, will be the final category label prediction of the categorization system for testing document q. In Section 4.3 the cases when a second category label prediction c_k is given are explained.

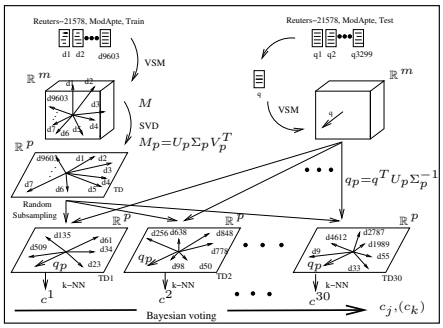


Figure 3: Proposed approach for multilabel document categorization tasks.

4 Experimental Setup

The aim of this section is to describe the document collection used in our experiment and to give an account of the preprocessing techniques and parameter settings we have applied.

When machine learning and other approaches are applied to text categorization problems, a common technique has been to decompose the multiclass problem into multiple, independent binary classification problems. In this paper, we adopt a different approach. We will be primarily interested in a classifier which produces a ranking of possible labels for a given document, with the hope that the appropriate labels will appear at the top of the ranking.

4.1 Document Collection

As previously mentioned, the experiment reported in this paper has been carried out for the Reuters-21578 dataset[4] compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. We use one of the most widely used training/testing divisions, the "ModApte" split, in which 75 % of the documents (9,603 documents) are selected for training and the remaining 25 % (3299 documents) to test the accuracy of the classifier.

Document distribution over categories in both the training and the testing sets is very unbalanced: the 10 most frequent categories, top-10, account 75% of the training documents; the rest is distributed among the other 108 categories.

According to the number of labels assigned to each document, many of them (19% in training and 8.48% in testing) are not assigned to any category, and some of them are assigned to 12. We have decided to keep the unlabeled documents in both the training and testing collections, as it is suggested in (Lewis, 2004)[5].

4.2 Preprocessing

The original format of the text documents is in SGML. We perform some preprocessing to filter out the unused parts of a document. We preserved only the title and the body text, punctuation and numbers have been removed and all letters have been converted to lowercase. We have

[4] http://daviddlewis.com/resources/testcollections

[5] In the "ModApte" Split section it is suggested as follows: "If you are using a learning algorithm that requires each training document to have at least TOPICS category, you can screen out the training documents with no TOPICS categories. Please do NOT screen out any of the 3,299 documents - that will make your results incomparable with other studies."

used the tools provided in the web[6] in order to extract text and categories from each document. We have stemmed the training and testing documents by using the Porter stemmer (Porter, 1980)[7]. By using it, case and flection information are removed from words. Consequently, the same experiment has been carried out for the two forms of the document collection: word-forms and Porter stems.

According to the dimension reduction, we have created the matrices for the two mentioned document collection forms. The sizes of the training matrices created are 15591×9603 for word-forms and 11114×9603 for Porter stems. Different number of dimensions have been experimented $(p = 100, 300, 500, 700)$.

4.3 Parameter setting

We have designed our experiment in order to optimize the microaveraged F_1 score. Based on previous experiments (Zelaia et al., 2005), we have set parameter k for the k-NN algorithm to $k = 3$. This way, the k-NN classifier will give a category label prediction based on the categories of the 3 nearest ones.

On the other hand, we also needed to decide the number of training databases TD_i to create. It has to be taken into account that a high number of training databases implies an increasing computational cost for the final classification system. We decided to create 30 training databases. However, this is a parameter that has not been optimized.

There are two other parameters which have been tuned: the size of each training database and the threshold for multilabeling. We now briefly give some cues about the tuning performed.

4.3.1 The size of the training databases

As we have previously mentioned, documents have been randomly selected from the original training database in order to construct the 30 training databases TD_i used in our classification system. There are $n = 9,603$ documents in the original Reuters training collection. We had to decide the number of documents to select in order to construct each TD_i. The number of documents selected from each category preserves the proportion of documents in the original one. We have experimented to select different numbers $n_1 < n$ of documents, according to the following formula:

$$n_1 = \sum_{i=1}^{115} 2 + \frac{t_i}{j}, \qquad j = 10, 20, \ldots, 70,$$

where t_i is the total number of training documents in category i. In Figure 4 it can be seen the variation of the n_1 parameter depending on the value of parameter j. We have experimented different j values, and evaluated the results. Based on the results obtained we decided to select $j = 60$, which means that each one of the 30 training databases will have $n_1 = 298$ documents. As we can see, the final classification system will be using training databases which are quite smaller that the original one. This gives a lower computational cost, and makes the classification system faster.

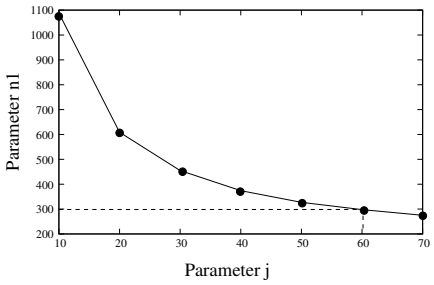


Figure 4: Random subsampling rate.

4.3.2 Threshold for multilabeling

The k-NN algorithm predicts a unique category label for each testing document, based on the ranked list of categories obtained for each training database TD_i[8]. As previously mentioned, we use Bayesian voting to combine the predictions.

The Reuters-21578 is a multilabel database, and therefore, we had to decide in which cases to assign a second category label to a testing document. Given that c_j is the category with the highest value in Bayesian voting and c_k the next one, the second c_k category label will be assigned when the following relation is true:

$$\text{cv}_{c_k} > \text{cv}_{c_j} \times r, \qquad r = 0.1, 0.2, \ldots, 0.9, 1$$

In Figure 5 we can see the mean number of categories assigned to a document for different values

[6] http://www.lins.fju.edu.tw/∼tseng/Collections/Reuters-21578.html

[7] http://tartarus.org/martin/PorterStemmer/

[8] It has to be noted that unlabeled documents have been preserved, and thus, our classification system treats unlabeled documents as documents of a new category

of r. Results obtained were evaluated and based on them we decided to select $r = 0.4$, which corresponds to a ratio of 1.05 categories.

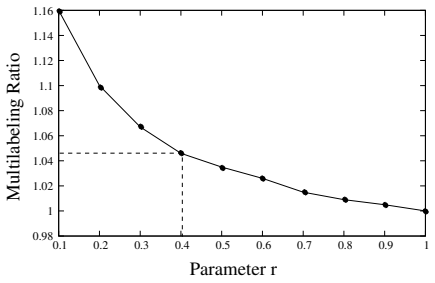


Figure 5: Threshold for multilabeling.

Corpus	Dimension reduction 100	300	500	700
Words(10)	93.06	93.17	93.44	92.00
Porter(10)	**93.57**	93.20	93.50	92.57
Words(90)	84.90	86.71	87.09	86.18
Porter(90)	85.34	86.64	**87.27**	86.30
Words(115)	84.66	86.44	86.73	85.84
Porter(115)	85.13	86.47	**87.01**	86.00

Table 3: Microaveraged F_1 scores for Reuters-21578 ModApte split.

5 Experimental Results

In Table 3 microaveraged F_1 scores obtained in our experiment are shown. As it could be expected, a simple stemming process increases slightly results, and it can be observed that the best result for the three category subsets has been obtained for the stemmed corpus, even though gain is low (less than 0.6).

The evaluation for the Top-10 category subset gives the best results, reaching up to 93.57%. In fact, this is the expected behavior, as the number of categories to be evaluated is small and the number of documents in each category is high. For this subset the best result has been obtained for 100 dimensions, although the variation is low among results for 100, 300 and 500 dimensions. When using higher dimensions results become poorer.

According to the R(90) and R(115) subsets, the best results are 87.27% and 87.01% respectively. Given that the difficulty of these subsets is quite similar, their behavior is also analogous. As we can see in the table, most of the best results for these subsets have been obtained by reducing the dimension of the space to 500.

6 Conclusions and Future Work

In this paper we present an approach for multilabel document categorization problems which consists in a multiclassifier system based on the k-NN algorithm. The documents are represented in a reduced dimensional space calculated by SVD. We want to emphasize that, due to the multilabel character of the database used, we have adapted the classification system in order for it to be multilabel too. The learning of the system has been unique (9603 training documents) and the category label predictions made by the classifier have been evaluated on the testing set according to the three category sets: top-10, R(90) and R(115). The microaveraged F_1 scores we obtain are among the best reported for the Reuters-21578.

As future work, we want to experiment with generating more than 30 training databases, and in a preliminary phase select the best among them. The predictions made using the selected training databases will be combined to obtain the final predictions.

When there is a low number of documents available for a given category, the power of LSI gets limited to create a space that reflects interesting properties of the data. As future work we want to include background text in the training collection and use an expanded term-document matrix that includes, besides the 9603 training documents, some other relevant texts. This may increase results, specially for the categories with less documents (Zelikovitz and Hirsh, 2001).

In order to see the consistency of our classifier, we also plan to repeat the experiment for the RCV1 (Lewis et al., 2004), a new benchmark collection for text categorization tasks which consists of 800,000 manually categorized newswire stories recently made available by Reuters.

7 Acknowledgements

This research was supported by the University of the Basque Country (UPV00141.226-T-15948/2004) and Gipuzkoa Council in a European

Union Program.

References

Berry, M.W. and Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia, (1999)

Breiman, L.: Bagging Predictors. Machine Learning, **24**(2), 123–140, (1996)

Cristianini, N., Shawe-Taylor, J. and Lodhi, H.: Latent Semantic Kernels. Proceedings of ICML'01, 18th International Conference on Machine Learning, 66–73, Morgan Kaufmann Publishers, (2001)

Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques. IEEE Computer Society Press, (1991)

Debole, F. and Sebastiani, F.: An Analysis of the Relative Hardness of Reuters-21578 Subsets. Journal of the American Society for Information Science and Technology, **56**(6),584–596, (2005)

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, **41**, 391–407, (1990)

Dietterich, T.G.: Machine-Learning Research: Four Current Directions. The AI Magazine, **18**(4), 97–136, (1998)

Dumais, S.T., Platt, J., Heckerman, D. and Sahami, M.: Inductive Learning Algorithms and Representations for Text Categorization. Proceedings of CIKM'98: 7th International Conference on Information and Knowledge Management, ACM Press, 148–155 (1998)

Dumais, S.: Latent Semantic Analysis. ARIST, Annual Review of Information Science Technology, **38**, 189–230, (2004)

Freund, Y. and Schapire, R.E.: A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, **14**(5), 771-780, (1999)

Gao, S., Wu, W., Lee, C.H. and Chua, T.S.: A Maximal Figure-of-Merit Learning Approach to Text Categorization. Proceedings of SIGIR'03: 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 174–181, ACM Press, (2003)

Gliozzo, A. and Strapparava, C.: Domain Kernels for Text Categorization. Proceedings of CoNLL'05: 9th Conference on Computational Natural Language Learning, 56–63, (2005)

Ho, T.K., Hull, J.J. and Srihari, S.N.: Decision Combination in Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, **16**(1), 66–75, (1994)

Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of ECML'98: 10th European Conference on Machine Learning, Springer 1398, 137–142, (1998)

Kim, H., Howland, P. and Park, H.: Dimension Reduction in Text Classification with Support Vector Machines. Journal of Machine Learning Research, **6**, 37–53, MIT Press, (2005)

Lewis, D.D.: Reuters-21578 Text Categorization Test Collection, Distribution 1.0. http://daviddlewis.com/resources/testcollections README file (v 1.3), (2004)

Lewis, D.D., Yang, Y., Rose, T.G. and Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, **5**, 361–397, (2004)

Porter, M.F.: An Algorithm for Suffix Stripping. Program, **14**(3), 130–137, (1980)

Salton, G. and McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York, (1983)

Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, **34**(1), 1–47, (2002)

Weiss, S.M., Apte, C., Damerau, F.J., Johnson, D.E., Oles, F.J., Goetz, T. and Hampp, T.: Maximizing Text-Mining Performance. IEEE Intelligent Systems, **14**(4),63–69, (1999)

Yang, Y. An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval. Kluwer Academic Publishers, **1**,(1/2), 69–90, (1999)

Zelaia, A., Alegria, I., Arregi, O. and Sierra, B.: Analyzing the Effect of Dimensionality Reduction in Document Categorization for Basque. Proceedings of L&TC'05: 2nd Language & Technology Conference, 72–75, (2005)

Zelikovitz, S. and Hirsh, H.: Using LSI for Text Classification in the Presence of Background Text. Proceedings of CIKM'01: 10th ACM International Conference on Information and Knowledge Management, ACM Press, 113–118, (2001)

Zhang, T. and Oles, F.J.: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval, **4**(1): 5–31, Kluwer Academic Publishers, (2001)

Applied Soft Computing 11 (2011) 4981–4990

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension

A. Zelaia*, I. Alegria, O. Arregi, B. Sierra

University of the Basque Country, UPV-EHU, Computer Science Faculty, 649 postakutxa, 20.080 Donostia, Gipuzkoa, Euskal-Herria, Spain

ARTICLE INFO

Article history:
Received 16 December 2009
Received in revised form
20 December 2010
Accepted 12 June 2011
Available online 2 July 2011

Keywords:
Document categorization
Vector space models
Multiclassifiers
Distance based classifiers

ABSTRACT

This article presents a multiclassifier approach for multiclass/multilabel document categorization problems. For the categorization process, we use a reduced vector representation obtained by SVD for training and testing documents, and a set of *k*-NN classifiers to predict the category of test documents; each *k*-NN classifier uses a reduced database subsampled from the original training database. To perform multilabeling classifications, a new approach based on Bayesian weighted voting is also presented. The good results obtained in the experiments give an indication of the potential of the proposed approach.

1. Introduction

Document categorization, the assignment of natural language texts, according to their content, to one or more predefined categories is an important component in many information organization and management tasks. Researchers have concentrated their efforts on finding the appropriate way to represent documents, index them and construct classifiers to assign each document to the correct categories. Both, document representation and classification method are crucial steps in the categorization process, and they are the object of this paper.

With respect to document representation, in order to obtain the vector representation of documents latent semantic indexing (LSI) [6], a variant of the vector space model, is used. This technique compresses vectors representing documents into vectors of a lower-dimensional space. LSI, which is based on singular value decomposition (SVD) of matrices [1], has the ability to extract the relations among words and documents by means of their context of use, and has been successfully applied to Information Retrieval tasks.

Once the representation of the documents is determined, a multiclassifier [14] is used to perform the categorization process. We use different training databases obtained from the original one by random subsampling, and a category prediction is given for each of them. Finally, to make the category predictions of testing documents, we use a model inspired in bagging [2] which uses *k*-NN classifiers [4].

Document representation and categorization do not solve the problem of multilabeling; the fact that one document can effectively belong to more than one of the categories considered. The most widely used technique for multilabeling in the literature is based on a binary selection for each category, where each document is tested as belonging or not to each category. In this paper we propose a new approach to multilabeling based on Bayesian voting.

The experiment presented in this article has been evaluated for Reuters-21578 standard document collection.[1] Keeping in mind the results published in the most recent literature, and having obtained promising results in our experiments, we consider the new categorization method presented in this article an interesting contribution for text categorization tasks.

The remainder of this article is organized as follows: Section 2 discusses related work on document categorization for Reuters-21578 collection. Section 3 presents our approach to the multiclass/multilabel text categorization. In Section 4 the experimental setup is introduced, and details are provided about the Reuters database, the preprocessing applied and the parameters to tune. The parameter tuning process is explained in detail in Section 5, and the experimental results are presented and

* Corresponding author.
E-mail addresses: ccpzejaa@si.ehu.es, ana.zelaia@ehu.es (A. Zelaia), acpalloi@si.ehu.es (I. Alegria), acparuro@si.ehu.es (O. Arregi), ccpsiarb@si.ehu.es (B. Sierra).

[1] http://daviddlewis.com/resources/testcollections.

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.06.002

discussed in Section 6. Finally, Section 7 contains some conclusions and comments on future work.

2. Related work

Text categorization consists in assigning predefined categories to text documents. In the past two decades, document categorization has received much attention and a considerable number of machine learning based approaches have been proposed. A good tutorial on the state-of-the-art of document categorization techniques can be found in [26].

In the document categorization task, different types of problems can be found,

- single-label vs. multilabel document categorization problems. In single-label document categorization tasks exactly one category is assigned to each document. In the multilabel case, categories are not mutually exclusive because the same document may be relevant to more than one category (1 to m category labels may be assigned to the same document, being m the total number of predefined categories).
- Binary classification problems vs. multiclass classification problems. In binary classification only two categories are involved. Multiclass problems arise when a document can be categorized under more than 2 categories.

Most of the classification systems which handle multilabel data in a multiclass problem decompose the multiclass problem into multiple, independent binary classification problems [16]. In this article we present a classifier which handles multilabel data in a multiclass problem; first, it produces a ranking of possible labels for a given document, expecting that the appropriate labels will appear at the top of the ranking. Then, it selects the number of labels to assign to a document (one or two). See also [20] and [36].

In order to reduce the feature vector representation, many authors use the SVD technique in text categorization problems [32] and [21].

For experimentation purposes, there are standard document collections available in the public domain that can be used for document categorization. The most widely used is Reuters-21578 collection, which is a multiclass (135 categories) and multilabel (the mean number of categories assigned to a document is 1.2) dataset. Many experiments have been carried out for the Reuters collection. However, they have not been performed under the same experimental conditions. So, it is difficult to establish comparisons among them. In order to overcome this problem and to lead researchers to use the same training/testing divisions, the Reuters documents have been specifically tagged, and researchers are encouraged to use one of these divisions. In our experiment we used the "ModApte" split [19].

In this section, the category subsets, evaluation measures and results obtained in the past and in recent years for Reuters-21578, ModApte split are analyzed.

2.1. Category subsets

Concerning the evaluation of the classification system, the TOPICS group of categories that labels Reuters dataset contains 135 categories. However, since many of the categories do not appear in any of the documents, and given that inductive based learning classifiers learn from training examples, these categories are not usually considered at evaluation time. The most widely used subsets are the following:

- Top-10: It is the set of the 10 categories which have the highest number of documents in the training set.
- R(90): It is the set of 90 categories which have at least one document in the training set and one in the testing set.
- R(115): It is the set of 115 categories which have at least one document in the training set.

In order to analyze the relative hardness of the three category subsets, a very recent article has been published by Debole and Sebastiani [5] where a systematic comparative experimental study has been carried out.

The results of the classification system proposed in this article are evaluated according to these three category subsets; once all the test documents have been classified, the evaluation measure is calculated for Top-10, R(90) and R(115).

2.2. Evaluation measures

The evaluation of a text categorization system is usually done experimentally by measuring its effectiveness, i.e. average correctness of the categorization. In binary text categorization, two known statistics are widely used to measure this effectiveness: precision and recall. Precision (Prec_i) is the percentage of documents correctly classified into a given category c_i, and recall (Rec_i) is the percentage of documents belonging to a given category c_i that are indeed classified into it.

$$\mathrm{Prec}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \qquad \mathrm{Rec}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}$$

where TP_i are true positives—documents correctly deemed to belong to c_i; FP_i are false positives—documents incorrectly deemed to belong to c_i; and FN_i are false negatives—documents incorrectly deemed not to belong to c_i.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by a measure which combines precision and recall. Various such measures have been proposed along the years. The breakeven point (BEP), the value at which precision equals recall, has been frequently used during the past decade. However, it has been recently criticized by its proposer ([26], footnote 19). Nowadays, the F_1 score is more frequently used. The F_1 score combines recall and precision with an equal weight. Given that Prec_i and Rec_i have been calculated for a given category c_i, the F_1 score for category i is calculated as follows:

$$F_1^i = \frac{2 \cdot \mathrm{Prec}_i \cdot \mathrm{Rec}_i}{\mathrm{Prec}_i + \mathrm{Rec}_i}$$

Since precision and recall are defined only for binary classification tasks, for multiclass problems results need to be averaged to get a single performance value. This is done by calculating the *microaverage* and *macroaverage* of results. In microaveraging, which is calculated by globally summing over all individual cases, categories count proportionally to the number of their positive testing examples. In macroaveraging, which is calculated by averaging over the results of the different categories, all categories count the same. Being $|C|$ the total number of categories in the multiclass problem, microaveraging (F_1^{μ}) and macroaveraging (F_1^{M}) are calculated as follows:

$$F_1^{\mu} = \frac{2\sum_{i=1}^{|C|}\mathrm{TP}_i}{2\sum_{i=1}^{|C|}\mathrm{TP}_i + \sum_{i=1}^{|C|}\mathrm{FP}_i + \sum_{i=1}^{|C|}\mathrm{FN}_i} \qquad F_1^{\mathrm{M}} = \frac{\sum_{i=1}^{|C|}F_1^i}{|C|}$$

Table 1
Some results reported for the Reuters-21578, ModApte split.

Type	Results reported by	Measure	R(90)	Top-10
SVM	Joachims [16]	BEP	86.4	–
SVM	Dumais et al. [9]	BEP	87.0	92.0
Commitee	Weiss et al. [28]	BEP	87.8	–
MFoM	Gao et al. [11]	F_1^μ	88.42	93.07
SVM	Kim et al. [17]	F_1^μ	87.11	92.21
SVM	Gliozzo and Strapparava [13]	F_1^μ	–	92.80
Combination	Debole and Sebastiani [5]	F_1^μ	78.7	85.20

See [5,30] for a more detailed explanation of the evaluation measures mentioned above. Results presented in this article are microaveraged (F_1^μ) and macroaveraged (F_1^M) F_1 scores.

2.3. Comparative results

Sebastiani [26] presents a table which lists results of experiments for various training/testing divisions of Reuters. Although the results listed by Sebastiani are microaveraged breakeven point (BEP) measures, and consequently, are not directly comparable to the ones presented in this article, we want to point out some of them.

In Table 1 some of the best results reported for the Reuters-21578, ModApte split are summarized. In the first part of the table, the three best results reported in [26] have been extracted. Two of them have been obtained by using support vector machines and the third one by using a commitee of multiple decision trees. As we have said earlier, they are microaveraged BEP measures. In the second part of the table, more recent microaveraged F_1 scores are included. MFoM learning approach has been used in [11,12], SVMs in [17] and domain kernel inside a SVM in [13]. Results reported by [5] give the average effectiveness of any combination of a learning method, a term selection function, a reduction factor and a term weighting policy.

Results for each one of the 10 most frequent categories can also be found in the literature. To facilitate the comparison of results, some of them are shown in Section 6 together with the ones obtained in our experiment.

3. Proposed approach

In this article we propose a multiclassifier based document categorization system which classifies documents represented in a reduced dimensional vector space. Different training databases are generated from the original training dataset in order to construct the multiclassifier. The *k*-NN classification algorithm is used which, according to each training database, makes a prediction for the testing documents. Finally, a Bayesian voting scheme is used to definitively assign category labels to the testing documents.

In the rest of this section, we provide details of our classification system proposal, particularly the way we construct the multiclassifier and how we obtain and combine the category label predictions. We also explain why and how we perform the dimensionality reduction to the vectors which represent documents.

3.1. The SVD dimensionality reduction technique

The classical vector space model (VSM) has been successfully employed to represent documents in text categorization tasks. The

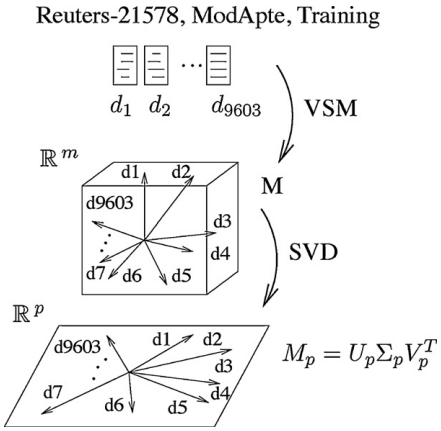


Fig. 1. Vectors in the VSM are projected to the reduced space by using SVD.

newer method of latent semantic indexing (LSI)[2][6] is a variant of the VSM [25] in which documents are represented in a lower dimensional space by applying the singular value decomposition (SVD) technique. LSI is based on the assumption that there is an underlying latent semantic structure in the term-document matrix that is corrupted by the wide variety of words used in documents. This is referred to as the problem of polysemy and synonymy. The basic idea is that if two document vectors represent two very similar topics, many words will co-occur on them, and they will have very close semantic structures after dimension reduction.

The SVD technique consists in factoring the term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values in non-increasing order, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively). Matrix M can be approximated by a lower rank M_p which is calculated by using the p largest singular values of M. This operation is called dimensionality reduction, and the p-dimensional space to which document vectors are projected is called the reduced space. The right dimension p must be chosen for successful application of the LSI/SVD technique. However, since there is no theoretical optimum value for p, potentially expensive experimentation may be required to determine it. A very good overview about the SVD technique and the way it is used in information retrieval systems can be found in [1].

For document categorization purposes [8], the testing document q is also projected to the p-dimensional space, $q_p = q^T U_p \Sigma_p^{-1}$, and the cosine is usually calculated to measure the semantic similarity between training and testing document vectors. The use of this reduced dimensional vector representation facilitates conceptual indexing, so that related documents which may not share common terms are still represented by nearby vectors in a p-dimensional vector space.

In Fig. 1 an illustration of the document vector projection can be seen. Documents in the training collection are represented by using the term-document matrix M, and each one of the documents is represented by a vector in the $\mathbb{R}^m$ vector space like in the traditional vector space model (VSM) scheme. Then, the dimension p is selected, and by applying SVD vectors are projected to the reduced

[2] http://lsi.research.telcordia.com, http://www.cs.utk.edu/ lsi.

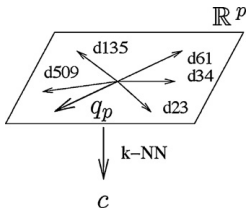


Fig. 2. The k-NN classifier is applied to q_p testing document and c category label is predicted.

space $\mathbb{R}^p$. Documents in the testing collection will also be projected to the same reduced space.

3.2. The k nearest neighbor classification algorithm (k-NN)

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing document, the k-NN classifier ranks its nearest neighbors among the training documents, and uses the categories of the k top-ranking neighbors to predict the categories of the testing document [4]. In the approach presented in this article, the training and testing documents are represented as reduced dimensional vectors in the lower dimensional space, and in order to find the nearest neighbors of a given document, the cosine similarity measure is calculated.

In Fig. 2 an illustration of this phase can be seen, where some training documents and a testing document q_p are projected in the reduced space $\mathbb{R}^p$. The nearest to the q_p testing document are considered to be the vectors which have the smallest angle with respect to q_p, and thus the highest cosine. According to the category labels of the nearest documents, a category label prediction, c, will be made for testing document q_p. Given the reduced size of the training database used, and to look for a variability in category labels, we set k to 1. This implies that the k-NN classifier will give a category label prediction based on the categories of the nearest one.

We decided to use the k-NN classifier because it performs best among the conventional methods [16,30,27,31] on the Reuters-21578 database and because we obtained good results in our previous work on text categorization for documents written in Basque [33]. Besides, the k-NN classification algorithm can be easily adapted to multiclass/multilabel categorization problems such as Reuters.

3.3. The induction and combination of multiple classifiers

The combination of multiple classifiers consists in applying different classifiers to the same classification task and in combining their outcome appropriately. By doing so, a better performance than that of any of the individual components is sought [14]. There are different ways to combine classifiers which improve accuracy over single classifiers. To decide which classifiers to use and how to combine the different outcomes becomes extremely relevant. Concerning the classifiers choice, several approaches have been studied, among them: bagging [2], which uses more than one model of the same paradigm in order to reduce errors; boosting [10], in which a different weight is given to different training documents; random forests [3], an improvement over bagging; bi-layer classifiers [29], where different models from different paradigms are combined in a parallel mode to obtain individual decisions to be used as predictor variables for a new classifier which makes the final decision. There are other combination approaches in serial or semi-parallel architectures [22]. A good review about classifier combination methods can be found in [18].

Methods for voting classification algorithms have been shown to be very successful in improving the accuracy of single classifiers. Typically, three patterns are used: unanimity, simple majority and plurality. As a multiclass problem is to be dealt with, plurality seems to be the most appropriate method. Within the different approaches present in the literature (Weighted Linear Combination, Dynamic Classifier Selection, Naive Bayesian voting, etc.) [26], and due to the characteristics of the categorization task, a Bayesian Weighted voting system has been used in this paper [15].

In our experiment we decided to construct a multiclassifier via bagging. In bagging, a set of training databases is generated by selecting n training documents randomly with replacement from the original training database TD of n documents. When a set of $n_1 < n$ training documents is chosen from the original training collection, the bagging is said to be applied by random subsampling [2]. This is the approach used in our work and the n_1 parameter has been selected via tuning. In Section 4.3 the selection will be explained in a more extended way.

Given a testing document q, each one of the classifiers will make a label prediction based on each one of the training databases. Regarding the combination of the different outcomes, it has to be pointed out that single voting scheme obtains worse results than Bayesian voting in the experiments carried out. In Bayesian voting [7], a confidence value $cv^i_{c_j}$ is calculated for each training database and category c_j to be predicted. These confidence values have been calculated based on the training collection. Confidence values are added by category; the category c_j that gets the highest value is finally proposed as a prediction for the testing document.

In Fig. 3 an illustration of the whole experiment can be seen. First, vectors in the VSM are projected to the reduced space by using SVD. Next, random subsampling is applied to the training database TD to obtain different training databases. Then the k-NN classifier is applied to each one of the training databases $TD_1, \ldots, TD_L$ to make category label predictions. Finally, Bayesian voting is used to combine predictions. c' will be the final category label prediction of the categorization system for testing document q. In some cases, a second category label c'' will also be assigned to the testing document. The conditions required to give this second category label prediction are explained in Section 4.3.

4. Experimental setup

In this section we describe the document collection used in our experiment and give an account of the preprocessing techniques applied and the parameters tuned.

4.1. Document collection

As previously mentioned, the experiment reported in this article was carried out for the Reuters-21578 dataset[3] compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. One of the most widely used training/testing divisions is used, the "ModApte" split, in which 75% of the documents (9603 documents) are selected for training and the remaining 25% (3299 documents) to test the accuracy of the classifier.

Document distribution over categories in both the training and the testing sets is very unbalanced: the 10 most frequent categories, Top-10, account for 75% of the training documents; the rest is distributed among the other 108 categories.[4]

[3] http://daviddlewis.com/resources/testcollections.

[4] It has to be noted that unlabeled documents have been preserved, and thus, our classification system treats unlabeled documents as documents of a new category.

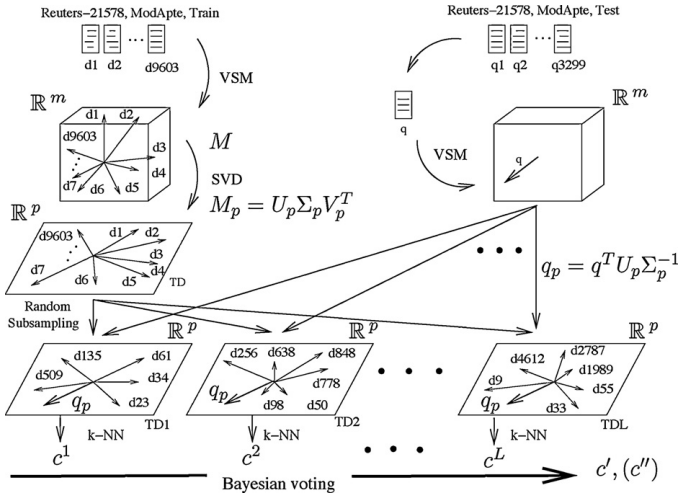


Fig. 3. Proposed approach for multiclass/multilabel document categorization tasks.

According to the number of labels assigned to each document, many of them (19% in training and 8.48% in testing) are not assigned to any category, and some of them are assigned to 12. We decided to keep the unlabeled documents in both the training and testing collections, as it is suggested in [19].[5]

4.2. Preprocessing

The original format of the text documents is in SGML. A preprocessing was performed to filter out the unused parts of a document. Only the title and the body text were preserved, punctuation and numbers were removed and all letters were converted to lowercase. The tools provided in the web[6] were used to extract text and categories from each document. Moreover, the training and testing documents were stemmed by using the Porter stemmer [23].[7] By doing so, case and flection information were removed from words. The experiment was carried out for the two forms of the document collection: the Bag-of-Words (BoW) and the Bag-of-Stems (BoS).

For the dimension reduction, it has to be noted that after preprocessing was applied, the training document collection was represented by 15,591 features, and so, the size of the training matrix created was $15,591 \times 9603$ for the BoW corpus. After applying the Porter stemmer, the number of features was reduced to 11,114, and a matrix of $11,114 \times 9603$ was obtained for the BoS corpus. By applying the SVD, the number of features in both corpora was reduced significantly. Experiments have been performed for dimensions $p = 100, \ldots, 1000$, although in this article we only publish results obtained for $p = 100, 300, 500$, because results obtained for higher dimensions were less significant.

Thus, and as a consequence of having two forms of the document collection (BoW and BoS) and three different dimensions ($p = 100$, 300, 500), we have six different representations of documents: BoW-100, BoW-300, BoW-500, BoS-100, BoS-300 and BoS-500. The experiment was performed and results evaluated for each one of the six different representations. In the illustration of the experiment in Fig. 3, each one of the six representations corresponds to the original training database (TD) to which random subsampling is applied.

4.3. Parameters

In the experimental approach proposed in this article, there were some decisions that needed to be made. We had to determine

(1) how many documents should be selected from the TD to create each one of the training databases: parameter n_1;
(2) which were the cases when a second category label should be assigned to a testing document after Bayesian voting was applied: parameter λ;
(3) which was the appropriate number of training databases that should be created: parameter L.

Therefore, a parameter tuning phase was carried out in order to fix the three parameters. This parameter tuning phase was not carried out based on the Reuters original training/testing document collections. Instead, a training subcollection (75%, 7242 docs.) and a validation subcollection (25%, 2361 docs.) were created randomly from the original training document collection of 9603 documents. This subdivision preserved the proportion of documents by category in the original training document collection. For categories with a very low number of documents (less than 4), at least one document in the training subcollection was kept.

In the following subsections, the three parameters are briefly introduced and in the next section the tuning process is explained in more detail.

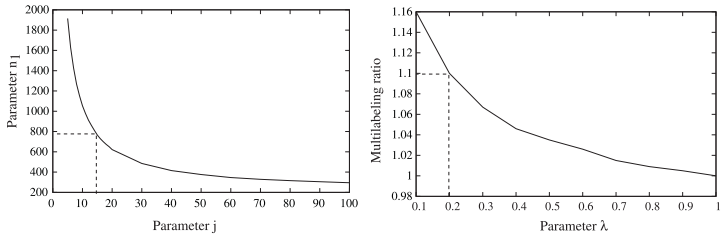
4.3.1. The size of each of the training databases: parameter n_1

As it was mentioned earlier, the multiclassifier is implemented by random subsampling, where a set of $n_1 < n$ training documents is chosen from the original training collection of n documents

[5] In the "ModApte" Split section it is suggested as follows: "If you are using a learning algorithm that requires each training document to have at least TOPICS category, you can screen out the training documents with no TOPICS categories. Please do NOT screen out any of the 3299 documents—that will make your results incomparable with other studies."

[6] http://www.lins.fju.edu.tw/ tseng/Collections/Reuters-21578.html.

[7] http://tartarus.org/martin/PorterStemmer/.



(a) Parameter $n1$: the size of each training database

(b) Parameter λ: the threshold for multilabeling

Fig. 4. Tuning of parameters n_1 and λ.

at random ($n = 7242$ during the tuning phase, $n = 9603$ during the experimental phase). Consequently, the size of each training database will vary depending on the value of n_1. The selection of different numbers of documents was experimented, according to the following equation:

$$n_1 = \sum_{i=1}^{115}(2 + \lfloor \frac{t_i}{j} \rfloor), \qquad j = 5, \ldots, 100 \tag{1}$$

where t_i is the total number of training documents in category c_i. Note that values for t_i vary depending on the training document collection referred to, i.e. the original or the subcollection created for the tuning phase.

By dividing t_i by j, the number of documents selected from each category preserves the proportion of documents per category in the original one. However, it has to be taken into account that some of the categories have a very low number of documents assigned to them. By adding 2, at least 2 documents will be selected from each category. In Fig. 4(a) the variation of the parameter n_1 depending on the value of j is outlined.

4.3.2. The threshold for multilabeling: parameter λ

Being Reuters-21578 a multilabel database, we decided to construct a classifier that, in some cases, assigns a second category label to a testing document. The multilabeling ratio we define is based on confidence values which are calculated in the following way: by using the training data, a missclassification matrix is constructed for each of the classifiers, where value in row m column n represents the number of documents that, belonging to class n have been classified as being of class m. The confidence value cv_{c_m} for category c_m is the percentage of documents correctly classified into a given category c_m among those classified as belonging to this category c_m. These confidence values are used as a weight value in Bayesian voting. Given that c' is the category with the highest confidence value in Bayesian voting and c'' the next one, the second category label c'' is assigned when the following relation is true:

$$cv_{c''} > cv_{c'} \times \lambda, \qquad \lambda = 0.1, 0.2, \ldots, 0.9, 1 \tag{2}$$

By applying Eq. (2), and depending on the value of parameter λ, the difference between the confidence values calculated for categories c' and c'' is measured. The lowest multilabeling ratio is obtained when $\lambda = 1$, in which case the classifier becomes single-label because the relation in the equation will never be hold. By reducing the value of parameter λ, different thresholds for the multilabeling ratio are experimented. In Fig. 4(b) the variation of the multilabeling ratio depending on the value of parameter λ is outlined.

4.3.3. The number of classifiers: parameter L

The classification approach presented in this article is based on the construction of a multiclassifier which uses different training databases to make category label predictions. The number of classifiers to construct is a parameter that needs to be tuned. Given that it is computationally too expensive to tune the three parameters at the same time, we decided to tune parameter L after the rest of parameters were tuned and set to their optimal values. So, based on our previous work [34], we decided to create 30 training databases and to tune parameters n_1 and λ previously introduced. Once n_1 and λ were set to their optimal values, parameter L was tuned by creating different numbers of training databases, ranging L from 10 to 300.

5. Parameter tuning

5.1. Tuning the parameter n_1: the size of each training database

In order to decide the optimal value for parameter n_1, the classification experiment was carried out varying j from 5 to 100 according to Eq. (1). Results obtained by using the multiclassifier system composed by 30 k-NN single classifiers appear graphically represented in Fig. 5. In fact, graphics are restricted to the range of parameter j where best results were obtained: $j = 5, \ldots, 20$.

A first glance at the graphics leads us to pay attention to Fig. 5(c) and (d) where the highest results for Top-10, R(90) and R(115) are obtained. Actually, the best ones for R(90) are obtained for the BoS-300 validation subcollection (an average microaveraged F_1 score of 87.57%), even though they are just slightly better than the ones obtained for the BoW-300 subcollection (87.42%); they both correspond to $j = 15$ (see discontinuous lines drawn in the graphics). According to Eq. (1), this implies that each of the training databases will be created by selecting $n_1 = 766$ documents in the tuning phase (see discontinuous line in Fig. 4(a)). It has to be noted that, being j the first parameter to be tuned, results depicted in Fig. 5 correspond to the average of the results obtained for $\lambda = 0.1, \ldots, 1$.

5.2. Tuning the parameter λ: the threshold for multilabeling

The tuning of parameter n_1 in the previous subsection was made based on the average of microaveraged F_1 scores obtained for $\lambda = 0.1, \ldots, 1$ and led us to set j to 15. In Table 2 results calculated for the six forms of the document subcollections are shown explicitly for $j = 15$. It can be seen that in most cases the results obtained by using 300 dimensions are superior than the ones obtained by using 100 and 500 dimensions.

However, it is not clear whether the stemming process improves results; by observing the average of results at the bottom of the table, the best ones are obtained for the stemmed documents

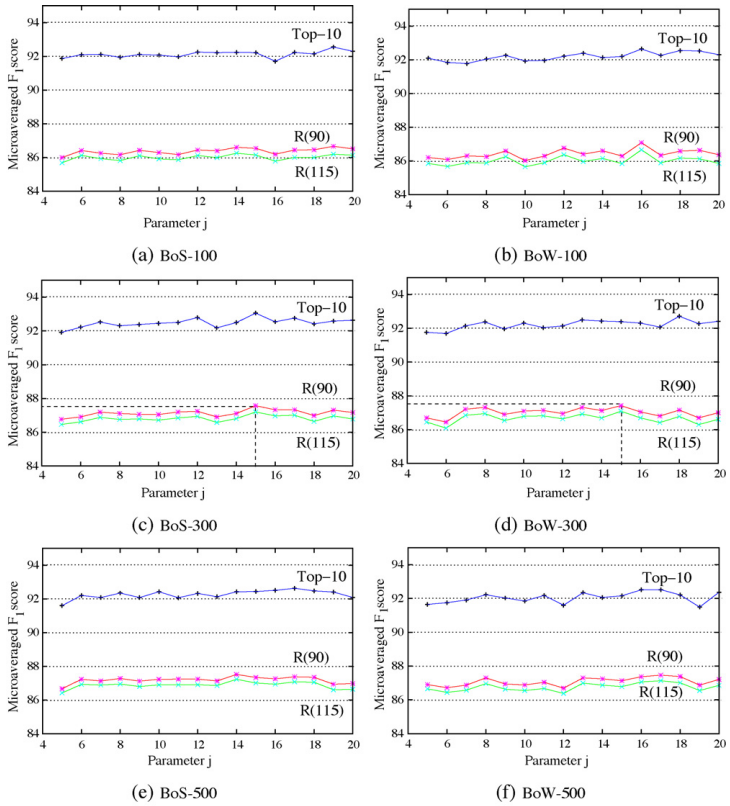


Fig. 5. Average microaveraged F_1 scores measured for the validation subcollection of documents; tuning parameter j.

(BoS-300, 87.57%), but they do not differ much from the ones obtained for the BoW-300 corpus (87.42%) (see also Fig. 5(c) and (d)). The best microaveraged F_1 result in Table 2 without calculating the average (88.96%) is obtained for the BoW-300 corpus.

In any case, the optimal results set parameter λ to 0.2, which according to Eq. (2), gives a multilabeling ratio of 1.1 categories per document in the validation subcollection (see Fig. 4(b)).

Given that the best results were obtained by using 300 dimensions, on the remaining of the tuning phase and during the experimental phase, only the BoW-300 and BoS-300 corpora were used.

Table 2
Microaveraged F_1 scores for $j = 15$ evaluated for the R(90) category subset by using the validation subcollection of documents; tuning parameter λ.

λ	BoS-100	BoS-300	BoS-500	BoW-100	BoW-300	BoW-500
0.1	87.28	88.42	87.90	86.85	88.46	87.89
0.2	87.68	**88.83**	88.54	87.30	**88.96**	88.65
0.3	87.37	88.73	88.55	87.03	88.42	88.42
0.4	86.87	88.40	88.06	86.74	87.97	87.73
0.5	86.60	87.93	87.70	86.34	87.63	87.24
0.6	86.32	87.48	86.93	86.12	87.19	86.86
0.7	86.07	86.98	86.75	85.87	86.77	86.35
0.8	86.00	86.49	86.50	85.68	86.43	86.28
0.9	85.68	86.37	86.32	85.53	86.34	86.06
1	85.57	86.08	86.14	85.40	86.04	85.80
Avg	86.54	**87.57**	87.34	86.29	**87.42**	87.13

5.3. Tuning the parameter L: the number of classifiers

Finally, and being aware that parameters n_1 and λ were tuned by creating 30 training databases ($L = 30$), we proceeded to optimize the number of classifiers to create for the final multiclassifier system, i.e. the number of individual k-NN algorithms to be used by the multiclassifier in order to combine opinions by Bayesian voting. The creation of different numbers of training databases, $L = 10, \ldots,$ 300 was experimented, and results were evaluated for $j = 15$ and $\lambda = 0.2$.

Fig. 6 shows results obtained for both the BoS-300 and the BoW-300 corpora. Graphics seem to suggest that a minimum number of classifiers (around 100) is needed for the multiclassifier system to give promising results. For a higher number of classifiers, the behavior of the system seems to stabilize. The best results for the R(90) category subset sets parameter L to 120 for the BoS-300 corpus (89.86%) and L to 190 for the BoW-300 corpus (89.52%). Once again, final results obtained for BoS-300 and BoW-300 are very similar. That is why it was decided to perform the final experiment for both forms by creating 120 and 190 classifiers, respectively.

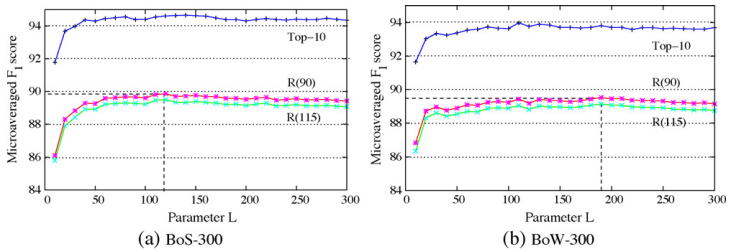


Fig. 6. Microaveraged F_1 scores for $j = 15$ and $\lambda = 0.2$: tuning parameter L.

Table 3
F_1 scores for Reuters-21578, ModApte split obtained for BoS (Bag-of-Stems) and BoW (Bag-of-Words) by using 300 dimensions in the reduced vector space representation.

Our results	Microaveraged scores			Macroaveraged scores		
	Top-10	R(90)	R(115)	Top-10	R(90)	R(115)
BoS-300	94.07	**88.26**	**88.26**	84.41	**52.86**	**41.58**
BoW-300	**94.10**	88.00	87.90	**85.30**	51.04	40.10
Single-BoS-300	83.18	75.59	75.52	59.51	33.23	26.20
Single-BoW-300	82.78	75.26	75.22	59.13	33.92	26.74

Table 4
Best results found in the literature. Results in [5] show the mean of the scores obtained by using different text classifiers.

	Microaveraged scores		
Results reported by	Top-10	R(90)	R(115)
Gao et al. [11]	93.07	88.42	–
Kim et al. [17]	92.21	87.11	–
Gliozzo and Strapparava [13]	92.80	–	–
Yang and Liu [31]	–	85.67	–
Schapire and Singer [27]	–	85.30	–
Debole and Sebastiani [5]	85.20	78.70	78.40

6. Experimental results

The final experiment was conducted with the optimal values for parameters set in the previous section: $j = 15$, $\lambda = 0.2$ and parameter $L = 120$ for the BoS-300 and $L = 190$ for the BoW-300. Results published in this section were calculated by evaluating results obtained for the original Reuters-21578 training-testing document collections. This implies a variation on the final size of each training database to $n_1 = 961$ (see Eq. (1)).

Table 3 shows microaveraged and macroaveraged F_1 scores obtained for the three category subsets. The first thing we want to emphasize is that, as far as we know, the microaveraged evaluation for the Top-10 category subset we achieve is the best one reported so far in the literature: 94.10% microaveraged F_1 score for BoW-300 and 94.07% for BoS-300. Moreover, it has to be noted that these results were obtained by using a pure ModApte split, i.e. without eliminating unlabeled documents. In addition, it is important to make clear that the evaluation was made after all documents in the testing collection were classified.

Results obtained for the R(90) category subset are among the best found in the literature (see Tables 3 and 4 to compare). They reach up to 88.26% microaveraged F_1 score, although they do not outperform results published in [11]. However, it should be noted that in the aforementioned work unlabeled documents were removed from training and testing document collections, and that the classification process was simplyfied by using only R(90) categories.

Results obtained for the R(115) category subset are analogous to the ones obtained for the R(90) subset as it could be expected, since the difficulty of these subsets is similar.

Regarding the macroaveraged performance achieved by our classification system, it can be said that even though the aim was not to optimize macroaveraged results, the system presented in this article behaves positively. Unfortunately, most of the researchers do not report macroaveraged results and consequently it is not easy to establish comparisons. In [11] a macroaveraged F_1 score of 87.78% for the Top-10 subset and 55.57% for the R(90) is reported. They are higher than the ones presented in this article, but once again, it has to be taken into account that the ModApte split is not used in the same way, and therefore, results are not directly comparable.

Analyzing results obtained for BoS-300 and BoW-300, it can be observed that the stemming process slightly improves results in most of the cases (R(90) and R(115)). In our previous work [33] we

Table 5
Results for Reuters-21578, ModApte split, evaluated for the Top-10 category subset, reported by: (a) [28], (b) [35], (c) [11], (d) [17], BoS-300: our F_1 results for BoS-300, BoW-300: our F_1 results for BoW-300.

Category	Train	Test	(a)	(b)	(c)	(d)	BoS-300	BoW-300
Earnings	2877	1087	97.78	98.4	97.9	98.25	**99.45**	**99.45**
Acquisitions	1650	719	95.69	95.4	96.8	95.57	**98.47**	97.86
Money-fx	538	179	76.44	76.0	82.6	75.78	89.58	**89.84**
Grain	433	149	**93.41**	90.3	90.6	92.88	88.37	87.21
Crude	389	189	88.63	84.9	89.7	88.11	**89.87**	89.65
Trade	369	118	75.41	76.3	80.7	75.32	89.54	**90.76**
Interest	347	131	72.95	75.7	79.2	77.99	83.06	**85.83**
Ship	197	89	80.96	83.6	**87.8**	84.09	75.86	73.61
Wheat	212	71	**89.59**	88.5	87.0	84.14	68.53	71.53
Corn	182	56	**89.43**	88.1	89.1	87.27	61.36	67.31
Macroaveraged scores			86.03	85.72	**88.14**	85.94	84.41	85.30
Microaveraged scores					93.07	92.21	94.07	**94.10**

verified that gain is higher when the stemming process is applied to a highly inflected language.

Results obtained by a single k-NN classifier ($L = 1$, $\lambda = 1$) are shown in Table 3, both for the stemmed (single-BoS-300) and not stemmed (single-BoW-300) corpus, in order to see to what extent the combination of multiple classifiers used in the experiment increases results. Certainly, the use of the multiclassifier contributes to improve results considerably; from an increase of more than 10 points for the microaveraged F_1 scores evaluated for the Top-10 by using the BoS-300 corpus (from 83.18% to 94.07%) to an increase of more than 26 points for the macroaveraged Top-10 BoW-300 (from 59.13% to 85.30%).

In Table 5 the F_1 scores for each one of the 10 most frequent categories are presented. Columns labeled as "Train" and "Test" show the number of documents assigned to each category in the Reuters-21578, ModApte split. The following four columns, labeled as (a)–(d), show F_1 scores reported in the literature. The last two columns, BoS-300 and BoW-300, present F_1 scores obtained by applying the approach proposed in this article.

Results obtained for each of the 10 categories are, in general, very good. Values marked in bold (best results for each category) show that, compared to the results published in the references mentioned in the table, our system obtains the best in 6 out of 10 of the categories. When these results are microaveraged, they are still better than the ones reported by some of the researchers. However, when macroaveraged, results do not improve. This may be because our classification system might not be suited for smaller categories i.e., "Wheat" and "Corn".

7. Conclusions and future work

In this article we present an approach for multiclass/multilabel document categorization problems which consists in a multiclassifier system based on the k-NN algorithm. The classifier was evaluated for the Reuters-21578, ModApte split testing collection, which is a multiclass and multilabel document collection. The microaveraged F_1 scores obtained are among the best reported in the literature, and the macroaveraged performance achieved by our classification system shows a positive behaviour.

Results obtained show that the construction of a multiclassifier, together with the use of Bayesian voting to combine category label predictions, plays an important role in the improvement of results.

A great methodological effort was put into the experimental phase. There were some parameters that needed to be set, but it was not possible to test all the possibilities because of computational load. To compensate, we decided to perform a tuning phase in a sound way by setting parameter n_1, λ and L, in that order, to their optimal values.

We also want to emphasize that we used the SVD dimensionality reduction technique in order to reduce the vector representation of documents. By doing so, documents that originally were represented by 15,000 features in the Bag-of-Words form and by 11,000 in the Bag-of-Lemmas simplify their representation to 300 features, consequently saving space and time.

As future work, we consider adapting the system in order to change the multilabeling ratio. In fact, our system assigns one or two labels to each testing document, but changing parameter λ it should be possible to assign different numbers of labels to documents. Thus, the system could be easily adapted to classify documents in collections with a higher multilabeling ratio.

We also intend to repeat the experiments for the RCV1 Reuters corpus[8] which consists of 800,000 manually categorized documents recently made available.

[8] http://www.daviddlewis.com/resources/testcollections/rcv1/.

Acknowledgements

This work was supported in part by KNOW2 project (TIN2009-14715-C04-01), and by the Basque Country Government under the Research Team Grant.

References

[1] M. Berry, M. Browne, Understanding Search Engines: Mathematical Modeling and Text Retrieval , SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2005, ISBN: 0-89871-581-4.
[2] L. Breiman, Bagging predictors , Machine Learning 24 (2) (1996) 123–140.
[3] L. Breiman, Random Forests , Machine Learning 45 (1) (2001) 5–32.
[4] B. Dasarathy, Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques , IEEE Computer Society Press, 1991.
[5] F. Debole, F. Sebastiani, An analysis of the relative hardness of Reuters-21578 subsets , Journal of the American Society for Information Science and Technology 56 (6) (2005) 584–596.
[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis , Journal of the American Society for Information Science 41 (1990) 391–407.
[7] T. Dietterich, Machine learning research: Four current directions , The AI Magazine 18 (4) (1998) 97–136.
[8] S. Dumais, Latent semantic analysis. , in: ARIST (Annual Review of Information Science Technology), vol. 38, 2004, pp. 189–230.
[9] S. Dumais, J. Platt, D. Heckerman, M. Sahami, Inductive learning algorithms and representations for text categorization , in: Proceedings of CIKM'98: 7th International Conference on Information and Knowledge Management, ACM Press, 1998, pp. 148–155.
[10] Y. Freund, R. Schapire, A short introduction to boosting , Journal of Japanese Society for Artificial Intelligence 14 (5) (1999) 771–780.
[11] S. Gao, W. Wu, C. Lee, T. Chua, A maximal figure-of-merit learning approach to text categorization , in: Proceedings of SIGIR'03: 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 174–181.
[12] S. Gao, W. Wu, C. Lee, A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization , in: ICML'04: Proceedings of the Twenty-first International Conference on Machine Learning, 2004, pp. 329–336.
[13] A. Gliozzo, C. Strapparava, Domain kernels for text categorization , in: Proceedings of CoNLL'05: 9th Conference on Computational Natural Language Learning, 2005, pp. 56–63.
[14] T. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems , IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1) (1994) 66–75.
[15] J.A. Hoeting, Methodology for Bayesian Model Averaging: An Update , in: Proceedings—Manuscripts of Invited Paper Presentations, International Biometric Conference, 2002, pp. 231–240.
[16] T. Joachims, Text categorization with support vector machines: Learning with many relevant features , in: Proceedings of ECML'98: 10th European Conference on Machine Learning, 1998, pp. 137–142.
[17] H. Kim, P. Howland, H. Park, Dimension reduction in text classification with support vector machines , Journal of Machine Learning Research 6 (2005) 37–53.
[18] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms , Wiley, 2004.
[19] D. Lewis, 2004. Reuters-21578 text categorization test collection. distribution 1.0. readme file (v 1.3). http://daviddlewis.com/resources/testcollections.
[20] T. Li, S. Zhu, M. Ogihara, Efficient multi-way text categorization via generalized discriminant analysis , in: Proceedings of CIKM'03: Twelfth International Conference on Information and Knowledge Management, 2003, pp. 317–324, http://doi.acm.org/10.1145/956863.956924.
[21] C.H. Li, S.C. Park, Combination of modified BPNN algorithms and an efficient feature selection method for text categorization , Information Processing and Management 45 (2009) 329–340.
[22] J.M. Martinez-Otzeta, B. Sierra, E. Lazkano, A. Astigarraga, Classifier hierarchy learning by means of genetic algorithms , Pattern Recognition Letters 27 (16) (2006).
[23] M. Porter, An algorithm for suffix stripping , Program 14 (3) (1980) 130–137.
[25] G. Salton, M. McGill, Introduction to Modern Information Retrieval , McGraw-Hill, 1983.
[26] F. Sebastiani, Machine learning in automated text categorization , ACM Computing Surveys 34 (1) (2002) 1–47.
[27] R.E. Schapire, Y. Singer, BoosTexter: a boosting-based system for text categorization , Machine Learning 39 (2/3) (2000) 135–168.
[28] S. Weiss, C. Apte, F. Damerau, D. Johnson, F. Oles, T. Goetz, T. Hampp, Maximizing text-mining performance , IEEE Intelligent Systems 14 (4) (1999) 63–69.
[29] D.H. Wolpert, Stacked Generalization , Neural Networks 5 (1992) 241–259.
[30] Y. Yang, An evaluation of statistical approaches to text categorization , Journal of Information Retrieval 1 (1/2) (1999) 69–90.
[31] Y. Yang, X. Liu, A re-examination of text categorization methods , in: 22nd Annual International SIGIR, 1999, pp. 42–49.
[32] B. Yu, Z. Xu, C. Li, Latent semantic analysis for text categorization using neural network , Knowledge-Based Systems 21 (2008) 900–904.

[33] A. Zelaia, I. Alegria, O. Arregi, B. Sierra, Analyzing the effect of dimensionality reduction in document categorization for basque , in: Proceedings of L&TC'05: 2nd Language & Technology Conference, 2005, pp. 72–75.
[34] A. Zelaia, I. Alegria, O. Arregi, B. Sierra, A multiclassifier based document categorization system: profiting from the singular value decomposition dimensionality reduction technique , in: Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, 2006, pp. 25–32.
[35] T. Zhang, F. Oles, Text categorization based on regularized linear classification methods , Information Retrieval 4 (1) (2001) 5–31.
[36] S. Zhu, X. Ji, W. Xu, Y. Gong, Multi-labelled classification using maximum entropy method , in: SIGIR'05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 274–281.

V. KAPITULUA

Hitzen Adiera-Desanbiguatzea

Hitz batek esanahi bat baino gehiago baditu, polisemikoa dela esaten da. Hitz polisemikoak sarri agertu ohi dira testuetan. Euskarazko *gorri* hitzak, adibidez, hainbat esanahi edo adiera ditu esleituta. *Arrosa gorria da* esaldian *gorri* hitzak kolore bat adierazten du. *Negu gorria* edo *gose gorria* moduko esamoldeetan aldiz, *gorri* hitzak larritasuna adierazten du; *larru gorritan* egotea biluzik egotea da, eta *Joan Batista Elizanburu gorria zen* esaldian *gorri* izatea ezkertiar izatea dela ulertzen da. Adiera desberdin asko, beraz, hitz bakarrarentzat.

Gizakiok trebeak izan ohi gara testuinguru jakin batean agertzen den hitz polisemiko bati zein adiera dagokion bereizten. Hizkuntzalaritza Konputazionalaren ikuspuntutik, aldiz, hitz polisemikoen adierak bereiztea ez da erraza gertatzen, eta ondorioz arazoak sortzen dira hizkuntzaren prozesamenduaren ataza orokorragoetan. Itzulpen automatikorako sistemetan, esaterako, beharrezkoa gertatzen da hitz-adierak ondo bereiztea itzulpena zuzen egiteko. Modu berean, informazioa erauzteko sistemek haren interesekoa ez den informazioa ekarriko diote erabiltzaileari, adiera oker interpretatuz gero.

Hitz-adierak automatikoki desanbiguatzeko metodo desberdinak erabili izan dira (Agirre and Edmonds, 2007). Proposatu diren algoritmoen eta tekniken artean, (Schütze, 1998) lana aipatu nahi dugu, hemen aurkezten dugun ikerketa-lanarekin duen gertutasunagatik eta izan duen oihartzunagatik. Egileak SVD bidezko dimentsio-murrizketarekin esperimentatzen du eta gainbegiratu gabeko sailkatze-teknikak aplikatzen ditu adierak automatikoki induzitzeko. Egindako esperimentuetan, SVD bidezko dimentsio-murrizketarekin emaitzak hobetzen direla egiaztatzen da. Gerora argitaratu den

(Gliozzo et al., 2005) artikuluan, hitzen adierak desanbiguatzeko sistema bat hobetzeko LSI/LSA erabiltzea proposatzen da.

Hitz polisemikoen adierak hiztegietan definitzen dira. Horrela, *Hitzen Adiera-Desanbiguatzea* (WSD, Word Sense Disambiguation) hitzaren agerpen bakoitza hiztegiko zein adierari dagokion ebaztea da. Hitz polisemikoen adiera-definizioak ez dira hiztegi guztietan berdin agertzen. Izan ere, adiera batzuk haien artean oso desberdinak badira ere, beste zenbaiten arteko aldea ez da hain handia izaten. Hori horrela izanik, Hitzen Adiera-Desanbiguatzea adiera-azpimultzo desberdinetarako egin daiteke. Adieren arteko bereizketa ahalik eta handiena egiten denean *ale xeheko desanbiguatzea* (*fine-grained*) egiten dela esaten da; oso antzeko esanahia duten adierak multzo bakar batean bilduta hartzen direnean, aldiz, *ale larriko desanbiguatzea* (*coarse--grained*) egiten da.

Bestalde, corpusa antolatzeko moduaren arabera, Hitzen Adiera-Desanbiguatzearen ataza bi modutara proposa daiteke. Aldez aurretik aukeratutako hitzen lagin bat desanbiguatu behar denean, ataza *lagin lexiko* (*lexical sample*) motakoa dela esaten da. Testu batean agertzen diren hitz guztiak desanbiguatu behar direnean, aldiz, ataza *hitz guztiak* (*all words*) motakoa dela esaten da. *Hitz guztiak* motako desanbiguazioa praktikan egin behar izaten denetik gertuago badago ere, zailagoa gertatzen da corpus etiketatuak sortzea.

Hitz-adierak desanbiguatzeko sistema automatikoen garapena eta ebaluazioa helburu hartuta sortu zen *Senseval* nazioarteko elkartea[1] duela 15 urte baino gehiago eta handia izan da orduz geroztik egin den ahalegina polisemiaren fenomenoa ulertzeko eta desanbiguatze-sistema automatikoak hobetzeko. Hainbat txapelketa eta biltzar antolatu izan dira urte horietan zehar, horien artean SemEval-2007.

Ikerketa-lan honen bigarren aplikazio-eremua Hitzen Adiera-Desanbiguatzea da. SemEval-2007 txapelketan parte hartu dugu eta aukera paregabea izan da gure Bagging multi-sailkatzailea beste aplikazio-eremu honetarako egokitzeko eta LSI/SVD bidezko dimentsio-murrizketaren eraginkortasuna neurtzeko.

Kapitulua horrela antolatuta dago. V.1. atalean SemEval-2007 biltzarreko Hitzen Adiera-Desanbiguatze ataza aurkezten da. Horrez gain, ikerketa--lan honetan ataza ebazteko erabilitako ezaugarri linguistikoak aipatzen dira eta ikasketa automatikorako corpusen antolaketa aurkezten da. V.2. atalean egindako esperimentuetatik lortutako ondorioak biltzen dira. V.3. atalean argitaratutako artikuluak jaso dira.

[1] http://www.senseval.org/

V.1 SemEval-2007. WSD ataza eta corpusak

Ebaluazio semantikoetarako nazioarteko biltzar bat da SemEval-2007[2]. Biltzarrerako proposatutako atazen artean guk 17. atazan[3] hartu dugu parte: *English Lexical Sample, English SRL and English All-Words Tasks* ataza-ko *Coarse-grained English Lexical Sample WSD* azpiatazan. Xehetasunak eta parte-hartzaileen emaitza ofizialak (Pradhan et al., 2007b) artikuluan daude.

```
<instance id="27:0@30@brown/cn/cn01@brown@en@on" docsrc="brown">
<answer instance="27:0@30@brown/cn/cn01@brown@en@on" senseid="1" wn="1,5"
wn-version="2.1"/><context>
They were dirty, their clothes were torn, and the girl was so exhausted that
she fell when she was still twenty feet from the front door. She lay there,
making no effort to get back on her feet. The boy came on to the porch and
sat down, his gaze on Morgan as if half expecting him to shoot and not really
<head> caring </head>. Morgan hesitated, thinking that if this was a trick,
it was a good one. He didn't think it was possible for this couple to be
pretending.
</context>
</instance>
<instance id="6:0@43@brown/cn/cn29@brown@en@on" docsrc="brown">
<answer instance="6:0@43@brown/cn/cn29@brown@en@on" senseid="2" wn="2,4"
wn-version="2.1"/><context>
The boy yanked her back hard, tugging her off her feet, and gathered her into
the crook of his arm . ``Now stay with me, Kitty'', he snapped irritably.
``I vowed to take <head> care </head> of you -- and that's what I'm gonna do.
We don't know this guy''. ``Oooo, square bit'', Feathertop screwed his
face up.
</context>
</instance>
<instance id="3:0@18@brown/cr/cr05@brown@en@on" docsrc="brown">
<answer instance="3:0@18@brown/cr/cr05@brown@en@on" senseid="3" wn="3"
wn-version="2.1"/><context>
Another case involves a newspaper reporter who tripped up a politician.
`` Mr. Jones, you may recall that we printed last week your denial of having
retracted the contradiction of your original statement. Now would you
<head> care </head> to have us say that you were misquoted in regard to it''?
Questions like this, framed in verbal fog, are perhaps the only kind that
have ever stumped an experienced politician. They recall Byron's classic
comment: ``I wish he would explain his explanation''.
</context></instance>
```

V.1 Irudia: care.v lemarako entrenamendu corpuseko hiru instantzia

[2]http://nlp.cs.swarthmore.edu/semeval/index.shtml
[3]http://nlp.cs.swarthmore.edu/semeval/tasks/task17/description.shtml

Lagin lexiko motako Hitzen Adiera-Desanbiguatze ataza honetan 100 lema desanbiguatu behar dira. Lemak 35 izen eta 65 aditz dira eta horietako bakoitzerako entrenamendu corpusa eta test corpusa ematen dira[4]. V.1. Irudian lema horietako baten entrenamendu corpuseko hiru instantzia ikus daitezke. Instantzia bakoitzerako identifikatzailea beltzez ageri da, lemaren agerpena morez eta dagokion adiera urdinez.

Esan bezala, hitz polisemikoen adierak hiztegietan definitzen dira. Hiztegi horiei *adiera-inbentario* (*sense-inventory*) esaten zaie eta bertan ematen da adiera bakoitzaren kodea, definizioa eta haren erabileraren hainbat adibide. V.1. Irudiko adierak, adibidez, WordNet[5] adiera-inbentariokoak dira. V.2. Irudian ikus daitezke care.v lemak adiera-inbentarioan definituta dituen hiru WordNet adierak.

SemEval-2007ko Hitzen Adiera-Desanbiguatze ataza hau ebaztea test corpuseko instantzia bakoitzari inbentarioko adiera bat esleitzea da. care.v lemaren test corpuseko instantzia bat V.3. Irudian ikus daiteke.

Atazako 100 lemen entrenamendu corpusak oso desberdinak dira. Polisemia maila aldetik, lema batzuen corpuseko instantzia guztiak adiera berekoak diren bitartean, beste batzuetan 13 adiera desberdineko instantziak daude. Tamaina aldetik ere diferentziak agerikoak dira: entrenamendu corpusik txikiena duen lemarako 19 instantzia besterik ez dauden bitartean, handienerako 2536 instantzia daude. Diferentzia horiek direla eta, hitz-adierak desanbiguatzea zailagoa gertatzen da lema batzuetan besteetan baino; parte-hartzaileen sistemek lema desberdinetarako lortutako emaitzek argi islatzen dute hori (Pradhan et al., 2007b).

V.1.1 Ikasketa automatikorako corpusak. Ezaugarri linguistikoak

Ikerketa-lan honetan, SemEval-2007 atazako lemen desanbiguatzea modu independentean ebatzi da. Hasteko, entrenamendu eta test corpusetako instantziak ezaugarri linguistikoen bidez adierazi dira. Hiru motakoak dira instantziak deskribatzeko erabili diren ezaugarriak.

- Kolokazio lokalak. Desanbiguatu beharreko hitzaren inguruan dauden hitzek, lemek eta haien kategoria gramatikalek (PoS, Part of Speech) emandako informazioa jasotzen duten ezaugarriak dira. Ingurune lokala markatzeko, desanbiguatu beharreko hitzarekin osatutako bigramak eta trigramak erabiltzen dira. Horrez gain, desanbiguatu beharreko hitzaren inguruan dauden kategoria gramatikal zehatz batzuetako (ize-

[4] http://nlp.cs.swarthmore.edu/semeval/tasks/task17/data.shtml
[5] http://wordnet.princeton.edu/

```
<inventory lemma="care-v"><commentary></commentary>
<sense group="1" n="1" name="consider as important enough for
   attention, concern or liking" type=""><commentary></commentary>
   <examples> I really care about my work. She's never cared very much about
   her appearance. I really don't care whether we go out or not. I don't care
   how much it costs, just buy it. Your parents are only doing this because
   they care about you. People who care enough, should do something about it.
   I have the impression that my wife no longer cares for me. Go ahead. I
   couldn't care less. I don't care a hang what happens to me. </examples>
   <mappings>
   <wn version="2.1">1,5</wn>
   <wn lemma="care_a_hang" version="2.1">1</wn>
   <wn lemma="care_for" version="2.1">1,2</wn>
   <omega/><pb/></mappings><SENSE_META clarity=""/></sense>
<sense group="1" n="2" name="tend or supervise somebody, provide protection
   and assistance" type=""><commentary></commentary>
   <examples> She cared for her grandmother for nine years and currently cares
   for her husband. The article tells you how to care for various type of pets.
   The nurse was caring for the wounded. They cared for the estate for a time
   but sold it when it became too much for them to handle. </examples>
   <mappings>
   <wn version="2.1">2,4</wn>
   <wn lemma="care_for" version="2.1">3</wn>
   <omega/><pb/></mappings><SENSE_META clarity=""/></sense>
<sense group="1" n="3" name="want, desire or perfer to do something" type="">
   <commentary> </commentary>
   <examples> Do you care for a drink? Would you care to join us for dinner?
   Care to hand over the salt? </examples>
   <mappings>
   <wn version="2.1">3</wn>
   <omega/><pb/></mappings><SENSE_META clarity=""/></sense>
   <WORD_META authors="hwangd" sample_score="-"/></inventory>
```

V.2 Irudia: care.v lemaren adiera-inbentarioa: definizioak eta adibideak

```
 <instance id="35:0@27@brown/cf/cf13@brown@en@on" docsrc="brown"><context>
This may be true whether the farm is owned or rented. If the farm is rented,
the rent must be paid. If it is owned, taxes must be paid, and if the place
is not free of mortgage, there will be interest and payments on the principal
to take <head> care </head> of. Advantages: A farm provides a wholesome and
healthful environment for children.
</context></instance>
```

V.3 Irudia: care.v lemaren test corpuseko instantzia bat

nak, adjektiboak, aditzak etab.) lemak eta hitzak ere kontuan hartzen dira.

- Mendekotasun sintaktikoak. Kasu honetan erabili direnak objektua, subjektua, izen modifikatzailea, preposizioa eta haurridea.

- Bag-of-words ezaugarriak. 4 tamainako leihoan eta testuinguruan (instantzian) agertzen diren lemak dira.

V.3. Irudian desanbiguatu beharreko hitza morez agertzen den *care* da. Aurrekoarekin eta ondorengoarekin bi bigrama (take care, care of) eta hiru trigrama (to take care, take care of, care of .) osatzen dira. Urdinez agertzen dira testuinguruan agertzen diren lemak: advantage, be, child, environment, farm, free, healthful, interest, mortgage, not, own, pay, payment, place, principal, provide, rent, take, tax, true, wholesome. Lema horiekin batera, desanbiguatu beharreko agerpenaren mendekotasun sintaktikoak eta haren gertukoen kategoria gramatikala ere jasotzen dute ezaugarriek. Testu-Sailkatze atazan dokumentuak adierazteko terminoen agerpen maiztasunak erabili diren bezala, hitz-adierak desanbiguatzeko hitzaren testuinguruko terminoen agerpen maiztasunek eragingo dute batez ere. Hori horrela izanik, LSI aplikatzea zentzuzkoa eta naturala gertatzen da.

Entrenamendurako 100 datu-multzoak klase anitzekoak dira, salbuespenak salbuespen, eta etiketa bakarrekoak; lemaren arabera klase edo adiera kopurua 13 artekoa izan badaiteke ere, instantziek beti dute adiera bakarra esleituta. LSIren bidez eraiki diren termino-dokumentu matrizeak ezaugarri-instantzia maiztasun matrizeak dira, desanbiguatu beharreko hitzaren testuingurua ezaugarri linguistikoen bidez deskribatua izan delako. Tamaina aldetik entrenamendu corpus batzuk oso txikiak direnez, matrizea ahalik eta termino kopuru handienarekin osatzea erabaki da. Hori dela eta, termino moduan aukeratuak izan dira corpuseko hitz (ezaugarri) guztiak, nahiz eta behin besterik ez agertu corpus osoan. V.1. Taulan instantzia kopuru oso txikia duten lema batzuen matrizeetatik ateratako informazioa ikus daiteke: adiera kopurua, termino kopurua, instantzia kopurua eta SVD dimentsioa (balio singular kopurua). Lema guztien artean instantzia kopuru txikienekoa 19 instantzia besterik ez dituen grant.v da. Haren matrizea 1167×19 tamainakoa da eta SVD deskonposaketan 19 balio singular lortu dira. Tamaina horretako matrizearentzat balio kopuru maximoa bada ere, oso txikia da.

Matrizeak sortu eta SVD deskonposaketa egitearen kostu konputazionala[6] oso txikia izan da lema guztietarako. Instantzia gehien duen share.n le-

[6]IXAko sisx05: Sun SPARC Enterprise M3000 SPARC64 VII+ Quad Core @ 2,75 Ghz

Lema	Adiera kopurua	Termino kopurua	Instantzia kopurua	Balio singular kopurua
care.v	3	2718	69	69
contribute.v	2	2106	35	35
fix.v	5	1771	32	32
grant.v	2	1167	19	19

V.1 Taula: Instantzia kopuru txikiko lau lemari buruzko informazioa.

marako sortu da matrizerik handiena, 28157×2536 termino-dokumentu matrizea eta 300 balio singularretarako deskonposaketa kalkulatzeko ordu laurdena besterik ez da behar izan. Matrizerik txikiena 1167×19 tamainakoa da, eta 19 balio singularretarako deskonposaketa kalkulatzeko segundoak besterik ez dira behar izan. Deskonposaketa egin aurretik log-entropy eraldaketa aplikatu zaie matrizeei.

V.1.2 Esperimentuak eta argitalpenak

Tradizionalki, LSI testu-dokumentuak sailkatzeko erabili izan da eta tamaina handiko matrizeekin egin izan dira esperimentuak. Dimentsio murriztua modu esperimentalean finkatzen den arren, normalean 100 eta 300 arteko balioren bat erabiltzea gomendatzen da. Desanbiguatze-ataza honetan tamainaz hain desberdinak diren termino-dokumentu matrizeak izanik, dimentsio-murrizketa entrenamendu corpuseko instantzia kopuruarekiko proportzionalki egitea erabaki da. Esperimentuak proportzio desberdinetarako egin dira, horien artean dimentsio-murrizketarik ez egitea ere probatu delarik.

LSI/SVD dimentsioetan adierazitako instantziak sailkatzeko, k-NN sailkatzaileez osatutako Bagging multi-sailkatzailea erabili da eta hainbat parametro optimizatu dira, lema bakoitzerako modu independentean: entrenamendurako datu-multzotik sortutako azpi-laginen tamaina, dimentsio-murrizketa, k-NN sailkatzaile kopurua eta k-NN sailkatzailerako auzokide hurbilenen kopurua.

Txapelketako 13 parte-hartzaileen emaitza ofizialak (Pradhan et al., 2007b) artikuluan aurki daitezke. Ikerketa-lan honetan aurkezten den desanbiguatze-sistemak 9. postua lortu du, batez bestekotik gorako emaitzarekin (%79.9)

eta ACL-SIGLEX[7] taldeak 2007an Pragan antolatutako ebaluazio semantikoen (SemEval-2007) biltzarrean aurkeztua[8] izan da.

- UBC-ZAS: A k-NN based Multiclassifier System to perform WSD in a Reduced Dimensional Vector Space. Ana Zelaia, Olatz Arregi and Basilio Sierra. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 358-361 orriak, (2007), Praga, Txekiar Errepublika. (Zelaia et al., 2007)[9].

Pragatik bueltan, sailkatze-sistemaren hainbat parametroren optimizazioa modu sakonagoan egin eta emaitzak hobetzea lortu da (%85.65). Azken emaitzak ACL-SIGSEM[10] taldeak 2009an Tilburgen antolatutako (IWCS-8 '09) biltzarrean[11] aurkeztuak izan dira.

- A Multiclassifier Based Approach for Word Sense Disambiguation Using Singular Value Decomposition. Ana Zelaia, Olatz Arregi and Basilio Sierra. Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8 2009), 248-259 orriak, (2009), Tilburg, Herbehereak. (Zelaia et al., 2009b).

V.2 Ondorioak

Ateratako ondorioak horrela labur daitezke:

- Sistemak portaera ona erakutsi du Hitzen Adiera-Desanbiguatze automatikoan. Multi-sailkatzailea lema bakoitzaren desanbiguaziorako egokitu da eta txapelketan bataz-bestekotik gorako emaitza lortu da.

- Lema batzuetarako entrenamendu corpusa oso txikia denez, ahalik eta termino kopuru handiena aukeratzearen estrategia erabili da. Termino kopurua txikiegia izateak arriskuan jar dezake test instantzia berrien bektore bidezko adierazpena, instantzia berriak entrenamendu fasean aukeratutako terminorik ez izatekotan, dokumentua bera adierazteko modurik ez bait legoke. Hitz guztiak termino izanik, arrisku hori txikitzen da eta, oro har, estrategia onuragarri gertatzen da instantzia kopuru txikiko corpusekin lan egiteko.

[7] ACL-SIGLEX: ACL Special Interest Group on the Lexicon
[8] http://nlp.cs.swarthmore.edu/semeval/program.php
[9] http://aclweb.org/anthology/S/S07/S07-1078.pdf
[10] ACL-SIGSEM: ACL Special Interest Group on Computational Semantics
[11] http://iwcs.uvt.nl/iwcs8/

Tamaina handiagoko corpusetan ere estrategia bera aplikatu da. Oro har, estrategia ez da egokia tamaina oso handiko corpusetarako, alferrikako zarata sor daitekeelako; matrizea dentsitate gutxiagokoa (sparse) izango da eta hitzen agerkidetzen aldetik ez du onura handirik eragingo. Dena den, ikerketa-lan honetan ebatzi diren corpusak ez dira tamainaz handiegiak, eta guztietarako estrategia bera erabiltzeak emaitza onak eman ditu.

- Eraiki diren 100 termino-dokumentu matrizeetan, termino kopurua dokumentu kopurua baino askoz handiagoa da. SVD deskonposaketa egitean, dokumentu kopuruak mugatu du beti matrizearen balio singular kopurua. Kopuru hori oso txikia izanik ere, beti lortu da balio singular kopuru maximoa. Adierazpen bektorialaren interpretazioari dagokionez, entrenamendurako erabilitako instantzien artean mendekotasun linealik ez dagoela esan nahi du horrek. Zer esanik ez, instantziak deskribatzeko bag-of-words moduko ezaugarriak erabili izanak eta terminoen aukeraketarako erabilitako estrategiak eragin zuzena izan du matrizeen egituran.

- LSI/SVD bidezko dimentsio-murrizketa aplikatuz lortu dira emaitzarik onenak lema guztietarako. Hasiera batean instantzia kopururik txikieneko lemetan dimentsioa ez murriztea hobea izango zela pentsa bazitekeen ere, emaitzek erakutsi dute kasu horietan ere murriztea egokia dela.

V.3 Argitalpenak

Hitzen Adiera-Desanbiguatzea

- UBC-ZAS: A k-NN based Multiclassifier System to perform WSD in a Reduced Dimensional Vector Space. Ana Zelaia, Olatz Arregi and Basilio Sierra. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 358-361, (2007), Praga, Txekiar Errepublika. (Zelaia et al., 2007)[12].

- A Multiclassifier Based Approach for Word Sense Disambiguation Using Singular Value Decomposition. Ana Zelaia, Olatz Arregi and Basilio Sierra. Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8 2009), 248-259, (2009), Tilburg, Herbehereak. (Zelaia et al., 2009b).

[12]http://aclweb.org/anthology/S/S07/S07-1078.pdf

UBC-ZAS: A k-NN based Multiclassifier System to perform WSD in a Reduced Dimensional Vector Space

Ana Zelaia, Olatz Arregi and Basilio Sierra
Computer Science Faculty
University of the Basque Country
ana.zelaia@ehu.es

Abstract

In this article a multiclassifier approach for word sense disambiguation (WSD) problems is presented, where a set of k-NN classifiers is used to predict the category (sense) of each word. In order to combine the predictions generated by the multiclassifier, Bayesian voting is applied. Through all the classification process, a reduced dimensional vector representation obtained by Singular Value Decomposition (SVD) is used. Each word is considered an independent classification problem, and so different parameter setting, selected after a tuning phase, is applied to each word. The approach has been applied to the lexical sample WSD subtask of SemEval 2007 (task 17).

1 Introduction

Word Sense Disambiguation (WSD) is an important component in many information organization and management tasks. Both, word representation and classification method are crucial steps in the word sense disambiguation process. In this article both issues are considered. On the one hand, Latent Semantic Indexing (LSI) (Deerwester et al., 1990), which is a variant of the vector space model (VSM) (Salton and McGill, 1983), is used in order to obtain the vector representation of the corresponding word. This technique compresses vectors representing word related contexts into vectors of a lower-dimensional space. LSI, which is based on Singular Value Decomposition (SVD) (Berry and Browne, 1999) of matrices, has shown to have the ability to extract the relations among features representing words by means of their context of use, and has been successfully applied to Information Retrieval tasks.

On the other hand, a multiclassifier (Ho et al., 1994) which uses different training databases is constructed. These databases are obtained from the original training set by random subsampling. The implementation of this approach is made by a model inspired in bagging (Breiman, 1996), and the k-NN classification algorithm (Dasarathy, 1991) is used to make the sense predictions for testing words.

Our group (UBC-ZAS) has participated in the lexical sample subtask of SemEval-2007 for task 17, which consists on 100 different words for which a training and testing database have been provided.

The aim of this article is to give a brief description of our approach to deal with the WSD task and to show the results obtained. In Section 2, our approach is presented. In Section 3, the experimental setup is introduced. The experimental results are presented and discussed in Section 4, and finally, Section 5 contains some conclusions and comments on future work.

2 Proposed Approach

In this article a multiclassifier based WSD system which classifies word senses represented in a reduced dimensional vector space is proposed.

In Figure 1 an illustration of the experiment performed for each one of the 100 words can be seen. First, vectors in the VSM are projected to the reduced space by using SVD. Next, random subsampling is applied to the training database TD to obtain

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 358–361,
Prague, June 2007.

different training databases TD_i. Afterwards the k-NN classifier is applied for each TD_i to make sense label predictions. Finally, Bayesian voting scheme is used to combine predictions, and c_j will be the final sense label prediction for testing word q.

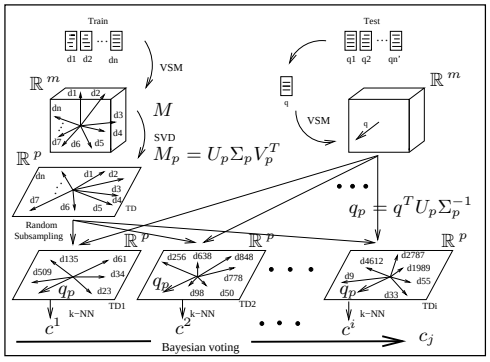


Figure 1: Proposed approach for WSD task

In the rest of this section, the preprocessing applied, the SVD dimensionality reduction technique, the k-NN algorithm and the combination of classifiers used are briefly reviewed.

2.1 Preprocessing

In order to obtain the vector representation for each of the word contexts (documents, cases) given by the organizers of the SemEval-2007 task, we used the features extracted by the UBC-ALM participating group (Agirre and Lopez de Lacalle, 2007). These features are local collocations (bigrams and trigrams formed with the words around the target), syntactic dependencies (object, subject, noun-modifier, preposition, and sibling) and Bag-of-words features (basically lemmas of the content words in the whole context, and in a $\pm$4-word window).

2.2 The SVD Dimensionality Reduction Technique

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization tasks. The newer method of Latent Semantic Indexing (LSI) [1] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional space created from the input training dataset. The SVD technique used by LSI consists in factoring term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively).

For classification purposes (Dumais, 2004), the training and testing documents are projected to the reduced dimensional space, $q_p = q^T U_p \Sigma_p^{-1}$, by using p singular values and the cosine is usually calculated to measure the similarity between training and testing document vectors.

2.3 The k-NN classification algorithm

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training word senses, and uses the sense of the k top-ranking neighbors to predict the corresponding to the word which is being analyzed (Dasarathy, 1991).

2.4 Combination of classifiers

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training databases TD_i is generated by selecting n training cases drawn randomly with replacement from the original training database TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In fact, this is the approach used in our work and the n_1 parameter has been selected via tuning.

According to the random subsampling, given a testing case q, the classifier will make a label prediction c^i based on each one of the training databases TD_i. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}_{c_j}^i$ is calculated for each training database TD_i and sense c_j to be predicted. These confidence values have been calculated based on the training collection. Confidence values are summed

[1] http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

by sense; the sense c_j that gets the highest value is finally proposed as a prediction for the testing examples.

3 Experimental Setup

In the approach proposed in this article there are some decisions that need to be taken, because it is not clear (1) how many examples should be selected from the TD of each word in order to create each one of the TD_i; (2) which is the appropriate dimension to be used in order to represent word related contexts (cases) for each word database; (3) which is the appropriate number of TD_i that should be created (number of classifiers to be used) and (4) which is the appropriate number of neighbors to be considered by the k-NN algorithm.

Therefore, a parameter tuning phase was carried out in order to fix the parameters. We decided to adjust them for each word independently.

In the following, the parameters are introduced and the tuning process carried out is explained. For two of the parameters (the number of classifiers and the number of neigbors for k-NN), the tuning phase was performed based on our previous experiments on document categorization tasks.

3.1 The size of each TD_i

As it was mentioned, the multiclassifier is implemented by random subsampling, where a set of $n_1 < n$ vectors is chosen from the original training collection of n examples for a given word (n is a different value for each one of the 100 words). Consequently, the size of each TD_i will vary depending on the value of n_1. The selection of different numbers of cases was experimented for each word in two different ways:

a) according to the following equation:

$$n_1 = \sum_{i=1}^{s}(2 + \lfloor \frac{t_i}{j} \rfloor), \qquad j = 1, \ldots, 10$$

where t_i is the total number of training cases in the sense c_i and s is the total number of senses for the given word. By dividing parameter t_i by j, the number of cases selected from each sense preserves the proportion of cases per sense in the original one. However, it has to be taken into account that some of the senses have a very low number of cases assigned to them. By summing 2, at least 2 cases will be selected from each sense. In order to decide the optimal value for parameter j, the classification experiment was carried out varying j from 1 to 10 for each word.

b) selecting a fixed number of cases for each of the senses which appeared for the word in the training database. Again, in the tuning phase, different numbers of cases (from 1 to 10) have been used for each of the 100 words in order to select a value for each of the words.

We optimized the size of each TD_i for each word by selecting the number of cases sometimes by procedure a) and sometimes by b).

3.2 The dimension of the reduced Vector Space Model

Taking into account the wide differences among the training case numbers for different words, we decided to project vectors representing them to different reduced dimensional spaces. The selection of those dimensions is based on the number of training cases available for each word, and limited to 500; the used dimensions vary from 19 (for the word *grant*) to 481 (for the word *part*).

3.3 The number of classifiers (TD_i)

Based on previous experiments carried out for document categorization (Zelaia et al., 2006), we decided to create 30 classifiers for some words and 50 for others, i.e. 30 or 50 individual k-NN algorithms will be used by the multiclassifier in order to combine opinions by Bayesian voting.

3.4 Number of neigbors for k-NN

Based on our previous experiments, we decided to use $k = 1$ and $k = 5$, and to select the best for each of the words. The cosine similarity measure is used in order to find the nearest or the 5 nearests.

4 Experimental Results

The experiment was conducted by considering the optimal values for parameters tuned by using the training case set.

Results published in this section were calculated by the SemEval-2007 organizers. Table 1 shows accuracy rates obtained by the 13 participants in the SemEval-2007, 17 task, lexical sample WSD subtask.

System	Accuracy	System	Accuracy
1.	0.887	8.	0.803
2.	0.869	**9.**	**0.799**
3.	0.864	10.	0.796
4.	0.857	11.	0.743
5.	0.851	12.	0.538
6.	0.851	13.	0.521
7.	0.838		

Table 1: Accuracy rates obtained by the 13 participants. SemEval-2007, 17 task (Lexical Sample)

The result obtained by our system is 0.799 (the 9th among 13 participants), 1 point over the mean accuracy (0.786).

5 Conclusions and Future Work

Results obtained show that the construction of a multiclassifier, together with the use of Bayesian voting to combine label predictions, plays an important role in the improvement of results. We also want to remark that we used the SVD dimensionality reduction technique in order to reduce the vector representation of cases.

The approach presented in this paper was already used in a document categorization task. However, we never used it for WSD task. Therefore, in order to adapt the method to the new task, we fixed some parameters based on our previous experiments (30-50 classifiers, $k = 1, 5$ for the k-NN algorithm) and tuned some other parameters by experimenting quite a high number of TD_i sizes and using different dimensions for each word. However, we noticed that the application of our approach to a different task is not straightforward. Greater effort will have to be made in order to tune the different parameters to this specific task of WSD.

One of the main difficulties we found was the difference in the number of training cases, comparing with the high number usually available in other tasks like text categorization.

As future work, we can think of applying a new preprocessing approach in order to extract better features from the training database which could help the SVD technique improving the accuracy after a dimensionality reduction is applied. The use of Wordnet may help.

6 Acknowledgements

This research was supported by the University of the Basque Country by the project “ANHITZ 2006: Language Technologies for Multilingual Interaction in Intelligent Environments”, IE 06-185

We wish to thank to the UBC-ALM group for helping us extracting learning features.

References

E. Agirre and O. Lopez de Lacalle. 2007. Ubc-alm: Combining k-nn with svd for wsd. submited for publication to SemEval-2007.

M.W. Berry and M. Browne. 1999. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia.

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

B.V. Dasarathy. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

T.G. Dietterich. 1998. Machine learning research: Four current directions. *The AI Magazine*, 18(4):97–136.

S. Dumais. 2004. Latent semantic analysis. In *ARIST (Annual Review of Information Science Technology)*, volume 38, pages 189–230.

T.K. Ho, J.J. Hull, and S.N. Srihari. 1994. Decision combination in multiple classifi er systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75.

G. Salton and M. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

A. Zelaia, I. Alegria, O. Arregi, and B. Sierra. 2006. A multiclassifi er based document categorization system: profi ting from the singular value decomposition dimensionality reduction technique. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*, pages 25–32.

A Multiclassifier based Approach for Word Sense Disambiguation using Singular Value Decomposition

Ana Zelaia, Olatz Arregi and Basilio Sierra
Computer Science Faculty
University of the Basque Country
ana.zelaia@ehu.es

Abstract

In this paper a multiclassifier based approach is presented for a word sense disambiguation (WSD) problem. A vector representation is used for training and testing cases and the Singular Value Decomposition (SVD) technique is applied to reduce the dimension of the representation. The approach we present consists in creating a set of k-NN classifiers and combining the predictions generated in order to give a final word sense prediction for each case to be classified. The combination is done by applying a Bayesian voting scheme. The approach has been applied to a database of 100 words made available by the lexical sample WSD subtask of SemEval-2007 (task 17) organizers. Each of the words was considered an independent classification problem. A methodological parameter tuning phase was applied in order to optimize parameter setting for each word. Results achieved are among the best and make the approach encouraging to apply to other WSD tasks.

1 Introduction

Word sense disambiguation (WSD) is the problem of determining which sense of a word is used when a word appears in a particular context. In fact, WSD is an important component in many information organization tasks, and fundamentally consists in a classification problem: given some word-contexts corresponding to some possible senses, the WSD system has to classify an occurrence of the word into one of its possible senses.

Proceedings of the 8th International Conference on Computational Semantics, pages 248–259,
Tilburg, January 2009.

In the approach presented in this paper, a vector representation is used for training and testing word cases and the Singular Value Decomposition of matrices is applied in order to reduce the dimension of the representation. In particular, Latent Semantic Indexing (LSI) [2] is used to make the dimension reduction. This technique compresses vectors representing word related contexts into vectors of a lower-dimensional space and has shown to have the ability to extract the relations among features representing words by means of their context of use.

We present a multiclassifier [8] based approach which uses different training databases. These databases are obtained from the original training dataset by random subsampling. The implementation of this approach is made by a model inspired in bagging [3], and the k-NN classification algorithm [4] is used to make sense predictions for testing words.

For experimentation, a previous tuning phase was performed to training data in order to automatically set some system parameters to their optimal values. Four are the parameters to be optimized, and the combination of all of them gives the possibility to perform the complete disambiguation process by 1440 different ways for each of the 100 words to be disambiguated. The tuning phase has been performed in a sound manner with the aim to improve our previous work [10]. Although the computational payload is high, it is a systematic way to fix the optimal values for parameters.

The aim of this article is to give a brief description of our approach to deal with the WSD task and to show the results achieved. In Section 2, our approach is presented. In Section 3, the experimental setup is introduced. The experimental results are presented and discussed in Section 4, and finally, Section 5 contains some conclusions and future work.

2 Proposed Approach

In this section, our approach is presented and the techniques used are briefly reviewed. First the dataset used in our experiments is described and previous results are presented. Next, the data preparation is explained in more detail. A short introduction to the SVD theory and to the k-NN classification algorithm is given afterwards. Finally, the multiclassifier construction is shown.

2.1 Dataset and previous results

The dataset we use in the experiments was obtained from the 4th International Workshop on Semantic Evaluations (SemEval-2007) web page[1], task 17, subtask 1: Coarse-grained English Lexical Sample WSD. This task consists of lexical sample style training and testing data for 100 lemmas (35 nouns and 65 verbs) of different degree of polysemy (ranging from 1 to 13) and number of instances annotated (ranging from 19 instances in training for the word *grant* to 2536 instances at *share*).

The average inter-annotator agreement for these lemmas is over 90%. In [9] task organizers describe the results achieved by the participating systems. They define a baseline for the task based on giving the most frequent sense in training (F-score: 78.0%). The best system performance (89.1%) was closely approaching the inter-annotator agreement but still below it.

2.2 Data Preparation

Once we downloaded the training and testing datasets, some features were extracted and vector representations were constructed for each training and testing case. The features were extracted by [1] and are local collocations (bigrams and trigrams formed with lemmas, word-forms or PoS tags around the target), syntactic dependencies (using relations like object, subject, noun modifier, preposition and sibling) and Bag-of-words features. This way, the original training and testing databases were converted to feature databases.

2.3 The SVD technique using LSI

The SVD technique consists in factoring term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively). Being k the number of singular values in matrix Σ and selecting the p highest singular values $p < k$, a vector representation for the training and testing cases can be calculated in the reduced dimensional vector space $\mathbb{R}^p$.

In our experiments we construct one feature-case matrix for each of the 100 words using the corresponding feature training dataset. Each of the columns in this matrix gives a vector representation to each of the training cases. As the number of training cases varies among different words, the number of columns present in the matrices is different; consequently, the

[1] http://nlp.cs.swarthmore.edu/semeval/tasks/task17/data.shtml

number of singular values changes as well. Taking this in consideration, we calculate the SVD of each matrix and obtain the reduced vector representations for training and testing cases for different p values. In order to calculate the SVD of the matrices, we use Latent Semantic Indexing (LSI) [2] [5], which has been successfully used for classification purposes [7],

2.4 The k-NN classification algorithm

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training cases [4].

In the approach presented in this article, the training and testing cases for each word are represented by vectors in each reduced dimensional vector space. The nearest to a testing case are considered to be the vectors which have the smallest angle with respect to it, and thus the highest cosine. That is why the cosine is usually calculated to measure the similarity between vectors. The word senses associated with the k top-ranking neighbors are used to make a prediction for the testing case. Parameter k was optimized for each word during tuning phase.

2.5 The multiclassifier construction

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components [8]. A widely used technique to implement this approach is *bagging* [3], where a set of training databases TD_i is generated by selecting n training cases drawn randomly with replacement from the original training database TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling.

In our work, we construct a multiclassifier by applying random subsampling for each word. As the number n of training cases is different for each word, we optimize via tuning the parameter n_1 for each multiclassifier constructed. This way, we work with training databases TD_i of different sizes. Moreover, the number of training databases TD_i to create for each multiclassifier, is also optimized via tuning.

Once the multiclassifiers are constructed, and given a testing case q for a word, the corresponding multiclassifier will make a word-sense label prediction c^i based on each one of the training databases TD_i. In order to calculate these confidence values, word-sense predictions are made for training cases

[2] http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

and the accuracies obtained give the confidence values which indicate the accuracy level that may be expected when a prediction is made for a testing case based on each training database TD_i and word-sense c_j to be predicted. The way we combine such predictions is by applying Bayesian voting [6], where a confidence value $\mathrm{cv}_{c_j}^i$ is calculated for each training database TD_i and word-sense c_j to be predicted. In testing phase, confidence values obtained for the testing cases are summed by sense; the sense c_j that gets the highest value is finally proposed as a prediction for the testing case q. This process is repeated for every testing case.

In Fig. 1 an illustration of the experiment performed for each one of the 100 words can be seen. First, vectors in the original Vector Space are projected to the reduced space using SVD; next, random subsampling is applied to the training database TD to obtain different training databases TD_i; afterwards, the k-NN classifier is applied for each TD_i to make sense label predictions; finally, Bayesian voting scheme is used to combine predictions, and c will be the final sense label prediction for testing case q.

3 Experimental Setup. The tuning phase

The experiments were carried out in two phases. First, a parameter tuning phase was performed in order to set the following parameters to their optimal values:

- The dimension p of the reduced dimensional vector space $\mathbb{R}^p$ to which word-case vectors are projected for each word.
- The number of classifiers, training databases TD_i, to create for each word.
- The number k of nearest neighbors to be considered by the k-NN classifier for each word.
- The number n_1 of cases to select from the TD of each word in order to create each one of the TD_i, that is, the size of each TD_i.

All the four parameters were adjusted independently for each word, because of the different characteristics of words with respect to the number of training and testing cases present in the dataset and the number of word-senses associated to each of them.

Validation and testing data subsets used in the tuning phase were extracted form the original training database TD for each word. Both subsets

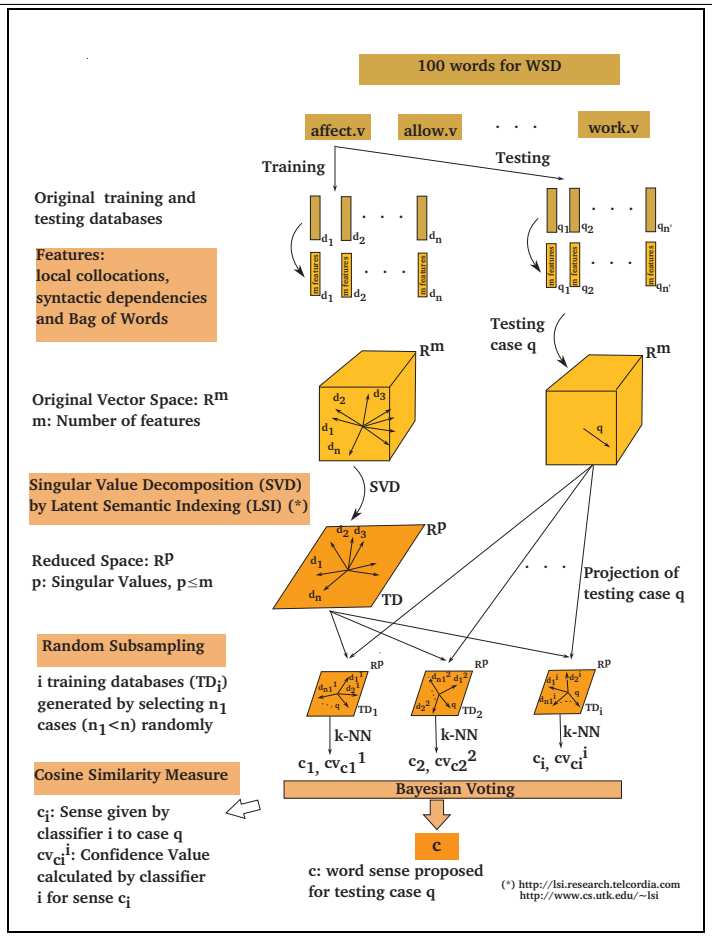


Figure 1: Proposed multiclassifier approach for WSD task

were constructed by random selection of cases, where 75% of the cases were selected for the validation subset and the rest for the tuning purposed made testing subset.

In the following the optimization of parameters is explained. Parameters were optimized in the same order as presented in this subsection, that is, the dimension reduction first, the number of classifiers second, the number k of nearest neighbors third and the size of each TD_i last. When the first parameter was being optimized, all possibilities for the other three parameters

were taken into account, and the optimization of the parameter was made based on the average of the 10% best results. Once a parameter was fixed, the same method was applied in order to optimize the rest of the parameters. This optimization method implies that the experiment was performed for all the combinations of the four parameters. This implies a high computational cost during the tuning phase. For testing phase, the experiments are performed using the optimal values for parameters.

3.1 The dimension p of $\mathbb{R}^p$

This is the first parameter we tuned. As it was previously mentioned in Section 2.3, the dimension p of the reduced dimensional vector space $\mathbb{R}^p$ to which training and testing cases are projected varies for different words. The reason for that is the difference in the number of cases present in the dataset for each word. For words with a high number of cases, the dimension was previously reduced to 500 (see [2]). Then, for every word we experimented by keeping the number of dimensions in a proportion. This proportion is given by parameter λ. We analyze four proportions by setting parameter λ to: $\lambda = 0$ keep all dimensions, $\lambda = 1$ keep 2/3 of the dimensions, $\lambda = 2$ keep half of the dimensions and $\lambda = 3$ keep a third of the dimensions. We calculated four different values for p. Training and testing cases were represented in the four $\mathbb{R}^p$ spaces and word-sense label predictions calculated for all of them. All the possibilities were tried for the rest of the parameters (detailed in the following subsections). For each value of λ, we selected the 10% best results from the 1440 we have, calculated the average of them and set parameter λ to its optimal value for each word. The optimization of λ gives a final optimal value for parameter p for each word.

3.2 The number of classifiers, TD_i

The number of classifiers, or TD_i to create for each word is also a parameter that needs to be tuned. This is because the number of cases present for each word is quite variable, and this fact may have some influence in the number of TD_i to construct. In our work, we experimented with 6 different values for parameter $i = 3, 5, 10, 20, 30, 40$. We performed the disambiguation process for each of them by considering the results for the optimal value of parameter λ, already optimized, and all the possible values for the rest of the parameters for each word. We then selected the best 10% average results achieved for each value of i, calculated the average, and based on these average results set the optimal value for parameter i for each word.

3.3 The number k of nearest neighbors for k-NN

At this stage of the tuning phase, and having already optimized the dimensionality reduction and the number of classifiers to create for each word, we take both optimal values and experiment with all possible values for the rest of the parameters. We calculate the average for six different values of k, $k = 3, 5, 7, 9, 11, 13$. We set the optimal value of k for each word based on the maximum average obtained.

3.4 The size of training databases TD_i: parameter n_1

As it was mentioned in Section 2.5, the parameter n_1 will be optimized for each word in order to create training databases TD_i of different sizes. The selection of different values for n_1 was experimented for each word according to the following equation:

$$n_1 = \sum_{i=1}^{s} (2 + \lfloor \frac{t_i}{j} \rfloor), \qquad j = 1, \ldots, 10$$

where t_i is the total number of training cases in the sense c_i and s is the total number of senses for the given word. By dividing t_i by j, the number of training-cases selected from each word-sense preserves the proportion of cases per sense in the original one. However, it has to be taken into account that some of the word-senses have a very low number of training-cases assigned to them. By summing 2, at least 2 training-cases will be selected from each word-sense. In order to decide the optimal value for j, the classification experiment was carried out varying j from 1 to 10 for each word. Given that parameters p, i and k are already set to their optimal values for each word, we calculate results for the 10 possible values of j, and set it to its optimal value.

4 Experimental Results

The experiment was conducted by considering the optimal values for parameters tuned. Original training and testing datasets were used for the final experiment, and results achieved were compared to the ones made available by task organizers [9].

Our system achieved an F-score of 85.65%, which compared to the baseline defined (78.0%) is a very good result, although still below the best published by task organizers (89.1%).

In [9] the performance of the top-8 systems on individual verbs and nouns is shown; 73 of the 100 lemmas are included in a table in two separated groups. Lemmas that have perfect or almost perfect accuracies have been removed. In TABLE 1 the average results achieved by our system for the two groups of lemmas are compared to the ones published in the cited paper. We can observe that our system performs better than the average of the top-8 systems disambiguating nouns, but slightly worse for verbs. In the overall, our system is very near to the average performance of the top-8 systems.

	Top-8	Our system
Verbs	70.44	67.78
Nouns	79.86	82.96
Overall	74.32	74.02

Table 1: Average performance compared to the top-8 in [9]

We want to remark that our system uses only the official training and testing data, without including background knowledge of any type. Some of the top-8 systems used background knowledge in order to assist in resolving ambiguities.

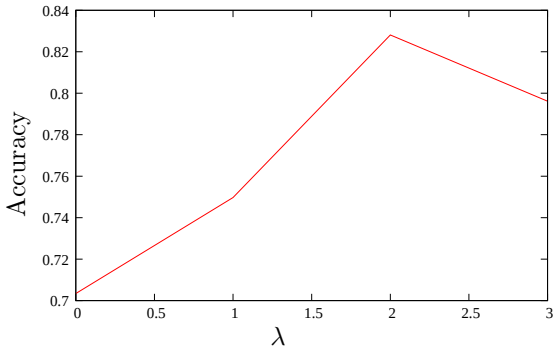


Figure 2: Average accuracy related to parameter $\lambda = 0, 1, 2, 3$

An analysis of the parameter optimization performed in the tuning phase lead us to observe that there is a relation between the dimensionality reduction level applied by SVD and the accuracy achieved for a word disambiguation (see Fig. 2). Words with more than 500 cases in the training dataset were not depicted in the figure because an additional dimension reduction was applied to them (see section 3.1). The graphic in Fig. 2 suggests that

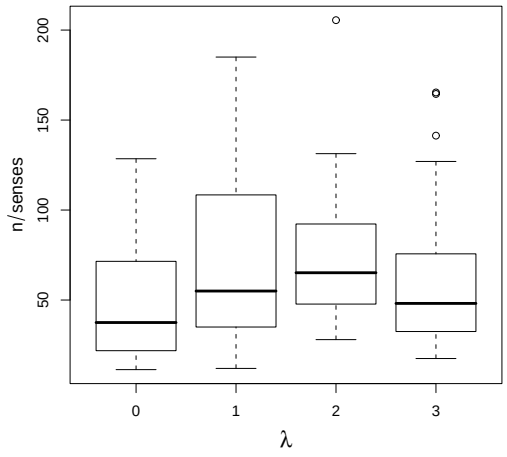


Figure 3: Complexity related to parameter $\lambda = 0, 1, 2, 3$

a dimensionality reduction of half of the features, $\lambda = 2$, is appropriate for words where a high level of accuracy is reached.

In order to analyze the adequacy of the parameter tuning performed, we created a new variable dividing the case number n of the training database by the number of senses for each word. This calculus is meant to represent the complexity of each word. In Fig. 3 the interquartile relationships found among the parameter λ and the complexity of the words is presented. For each value of λ the segments represent the minimum and the maximum value of the complexity, while the bold line shows the median and the rectangular area represents the density of the second and third quartiles. As it can be seen, the evolution of the median value, as well as the minimum values, are similar to the observed in the accuracies. This allows to say that the λ value was properly selected by the automatic selection used, and also that higher values of λ would not ensure better solutions for the most complex words.

5 Conclusions and Future Work

The good results achieved by our system show that the construction of multiclassifiers, together with the use of Bayesian voting to combine word-

sense label predictions, plays an important role in disambiguation tasks. The use of the SVD technique in order to reduce the vector representation of cases has been proved to behave appropriately.

We also want to remark that, our disambiguation system has been adapted to the task of disambiguating each one of the 100 words by applying a methodological parameter tuning directed to find the optimal values for each word. This makes possible to have a unique disambiguation system applicable to words with very different characteristics.

Moreover, in our experiments we used only the training data supplied for sense disambiguation in test set, with no inclusion of background knowledge at all, while most of the top-8 systems participating in the task do use some kind of background knowledge. As future work, we intend to make use of such knowledge and hope that results will increase. We also intend to apply this approach to other disambiguation tasks.

References

[1] E. Agirre and O. Lopez de Lacalle. Ubc-alm: Combining k-nn with svd for wsd. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval-2007*, pages 342–345, 2007.

[2] M. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia, 1999.

[3] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[4] B. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press, 1991.

[5] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[6] T. Dietterich. Machine learning research: Four current directions. *The AI Magazine*, 18(4):97–136, 1998.

[7] S. Dumais. Latent semantic analysis. In *ARIST (Annual Review of Information Science Technology)*, volume 38, pages 189–230, 2004.

[8] T. Ho, J. Hull, and S. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[9] S. Pradhan, E. Loper, D. Dligach, and M. Palmer. Semeval-2007 task 17: English lexical sample, srl and all words. In A. for Computational Linguistics, editor, *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval-2007*, pages 87–92, 2007.

[10] A. Zelaia, O. Arregi, and B. Sierra. Ubc-zas: A k-nn based multiclassifier system to perform wsd in a reduced dimensional vector space. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval-2007*, pages 358–361, 2007.

VI. KAPITULUA

Korreferentzia-Ebaztea

Komunikaziorako hizkuntza erabiltzen denean, ahozkoa nahiz idatzizkoa izan, diskurtsoa osatzen duten esaldien artean nolabaiteko lotura izaten da. Izan ere, esaldi batek hurrengo esaldiarekin duen loturari esker, diskurtsoak kohesioa izatea lortzen da. Diskurtsoaren ulermena egokia izan dadin, garrantzitsua gertatzen da lotura horiek antzematea.

Korreferentzia izeneko fenomeno linguistikoa gertatzen da, diskurtsoko zenbait elementuk objektu edo gertakari berari aipamen egiten diotenean, hau da, mundu errealeko erreferente bera dutenean. *Korreferentzia-Ebazte* (Coreference Resolution) izenez ezagutzen den ataza korreferentzian dauden elementuak aurkitzean datza; erreferente berari aipamen egiten dioten diskurtso zatiak aurkitu eta haien arteko loturak sortu (Ng, 2010).

Testu zati batean korreferente diren *aipamen* (mention) guztiek *korreferentzia-katea* osatzen dute. VI.1. Irudian korreferentzia-kate baten adibidea ikus daiteke. Letra lodiz agertzen diren aipamen guztiek Zidane futbol jokalariari egiten diote erreferentzia. Testua ondo ulertzeko, beharrezkoa gertatzen da erreferente bera duten aipamen horiek guztiak aurkitzea eta guztien arteko lotura egitea.

Korreferentzia fenomeno linguistikoarekin oso erlazionatuta dagoen beste fenomeno bat anaforarena da. Esaldi batean espresio bat *anafora* dela esaten da testuinguruan dagoeneko agertu den beste espresio bati erreferentzia egiten badio, eta bere interpretazioa posible izan dadin derrigorrezkoa gertatzen bada erreferentzia zein espresiori egiten dion ebaztea. VI.2. Irudiko adibideko **hark** izenordaina anafora da, dagoeneko agertu den beste espresio bat seinalatzen ari delako: **Keizo Obuchi** izen-sintagma. Lotura hori argitu ezean, anafora interpretatzea ez da posible. *Anafora-Ebaztea* (Anaphora

Munduko futbolaririk onena dela egiaztatzeko aukera dauka **Zinedine Zidane frantziarrak** gaur, Eurokopako finalean, erronka handia daukalako aurrean: Italiako harresia haustea. **Zidanek** badaki hori, baina **jokalari handia** da **bera** eta erantzukizuna hartuko du. Lehengo asteazkenean ere halaxe hartu zuen, Portugalen kontrako finalerdian luzapenean penaltia bota zuenean. Hor egongo da gakoa, Zidaneren partiduan. Dino Zoff hautatzaile italiarra saiatuko da **hura** gelditzen, eta ohiko taktika defentsiboa zelairatuko du.

VI.1 Irudia: Korreferentzia-kate baten adibidea

Resolution) anaforak erreferentzia zein beste espresiori egiten dion argitzea da, espresio horri *aurrekari* deitzen zaiolarik.

Anafora edozein kategoria gramatikaleko unitate lexiko edo sintagma izan daiteke (izena, adjektiboa, izenordaina, aditza, etab.), eta horren arabera anafora mota desberdinak bereizten dira. Elementu anaforikoa izenordaina edo izenordain funtzioa betetzen duen determinatzailea denean *Anafora Pronominal* izenez ezagutzen da.

Apirilean, LDPko buruek **Keizo Obuchi** ordezkatzeko Yoshiro Mori hautatu zuten, **hark** tronbosia izan ondoren.

VI.2 Irudia: Anafora Pronominal baten adibidea

Diskurtsoan anaforak erabiltzea, hau da, testuan lehenago (edo ondoren, kataforen kasuan) agertzen den espresio bat erakustea, esaldien arteko kohesioa lortzeko erabiltzen den baliabide sintaktiko bat da. Kohesioa indartzeaz gain, testua errepikakorra ez gertatzea ere lortu nahi izaten da. Anafora dagokion aurrekariaz ordezkatzean bere interpretazioa ebatzita geratzen da. VI.2. Irudiko anafora ebatziz gero, esaldia horrela geratuko da: "Apirilean, LDPko buruek Keizo Obuchi ordezkatzeko Yoshiro Mori hautatu zuten, Keizo Obuchik tronbosia izan ondoren". Esan beharra dago, **hark** anafora pronominala eta **Keizo Obuchi** aurrekaria erlazio anaforikoan agertzeaz gain, korreferenteak ere badirela, mundu errealeko erreferente bera dutelako biek.

Korreferentziak ebaztea Hizkuntzaren Prozesamenduaren ataza garrantzitsu bilakatu da azken urteotan. Izan ere, korreferentzia ebazteak zuzenean eragiten baitu diskurtsoaren ulermen sakona eskatzen duten beste hain-

bat atazetan, hala nola, Itzulpen Automatikoan edo Informazioa-erauzketan (Recasens and Vila, 2010). Hori dela eta, azken hogei urteotan arreta berezia eskaini zaio korreferentzia modu automatikoan ebazteari. Ahalegin horretan, hainbat txapelketa eta biltzar antolatu izan dira. MUC-6 (Grishman and Sundheim, 1996) eta MUC-7 (Hirschman and Chinchor, 1998) lehenengotakoak izan ziren. Gerora ospatutako CoNLL-2011 (Pradhan et al., 2011) eta CoNLL-2012 (Pradhan et al., 2012) txapelketek Korreferentzia-ebaztearen inguruko ikerketa sustatu dute eta hainbat hizkuntzatarako sistema berriak proposatu dira.

Ataza konplexua da oso, eta ez Hizkuntzalaritza Konputazionalaren ikuspuntutik bakarrik. Corpusetan korreferentzia-kateak eskuz markatze lanetan jarduten diren adituen artean ere desadostasunak izan ohi dira, atazaren zailtasuna agerian geratzen delarik (Hovy et al., 2013). Mitkov-ek berak ere halaxe dio: *even within the narrow domain of NP coreference it is not always easy to decide which NPs should be marked as coreferencial. This is indicative of how complex anaphora and coreference are. As a consequence, the annotation process is often considered to be far from reliable in that inter-annotator agreement may be disappointingly low* (Mitkov, 2002).

Korreferentzia ebazteko sistema automatikoek normalean bi urratsetan banatzen dute ataza: lehen urrats batean korreferentzian dauden aipamenak aurkitu behar dira testuan zehar, bigarren urrats batean aipamen bakoitza zein erreferenteri dagokion erabakitzeko. Sistema horien ebaluaziorako erabili izan diren neurriak atazaren bi urratsak batera ebaluatzeko diseinatu izan badira ere, bada urrats bakoitza bere aldetik ebaluatu beharko litzatekeela aldarrikatzen duenik ere, oso izaera desberdineko bi azpiataza direla argudiatuz (Recasens and Hovy, 2011). Kapitulu honetan aurkezten den sistemak bigarren urratsari erantzuten dio. Aipamenak korreferentzia-kateen bidez etiketatuta dituen corpus batetik abiatuz, haien artean korreferente badirenak eta ez direnak bereizten lagunduko duen sistema automatiko bat garatzea izan da egindako lanaren helburua. Hemendik aurrera, Korreferentzia-Ebazteari buruz hitz egiten dugunean, urrats horri egin nahi diogu erreferentzia. Anafora Pronominala Ebaztearen atazarako euskarazko corpus etiketatu bat erabili da eta Korreferentzia-Ebaztearen atazarako ingelesezko corpus estandar bat.

Kapitulua horrela antolatuta dago. VI.1. atalean Anafora Pronominala ebazteko egindako lana aurkezten da eta VI.2. atalean Korreferentzia ebazteko egindako lana. VI.3. atalean egindako lanetik ateratako ondorioak aipatzen dira. Amaitzeko, VI.4. atalean jaso dira Anafora Pronominala ebazteko eta Korreferentzia ebazteko egindako lanarekin argitaratutako artikuluak.

VI.1 Anafora Pronominala. Euskarazko corpusa

Anafora Pronominala ebaztearen atazak arreta berezia jaso izan du Hizkuntzalaritza Konputazionalaren arloan. Izan ere, korreferentzia ebazteko sistemak garatzean, anafora pronominala ebaztea atazaren lehen urrats moduan planteatu izan da. Dagoeneko badira urte batzuk euskararako korreferentzia ebazteko tresna bat garatzeko asmoari ekin zitzaiola. Egindako lana euskarazko Anafora Pronominalaren analisitik eta corpusaren markaketatik abiatu zen (Aduriz et al., 2005), (Aduriz et al., 2006), (Aduriz et al., 2007). Ondoren egindakoak izan dira, besteak beste, euskarazko corpusaren anotaziorako korreferentziaren analisia (Aduriz et al., 2008), euskarazko korreferentziaren etiketatze automatikoa (Goenaga et al., 2012), aipamenen detekzioa euskarazko korreferentzia ebazteko sistema automatiko baten lehen urrats moduan (Soraluze et al., 2012) eta korreferentzia ebazteko sistema baten egokitzapena euskararako (Soraluze et al., 2015). Korreferentzia-Ebaztearen atazarako corpus handiago bat etiketatzea ere lortu da azken urteotan.

Atal honetan aurkezten den lanean erabili den corpusa anafora pronominalaz etiketatu den euskarazko lehenengo corpusa da, lan hau egiteari ekin genion garaian hura besterik ez zegoelako. Eus3LB corpusaren zati bat da, guztira 50000 hitz inguru dituena (Palomar et al., 2004). *Euskaldunon egunkaria* egunkariko 2001. urteko artikuluez osatutako corpusa da. Testuak gai desberdinei buruzkoak dira: kirola, politika, ekonomia, etab. Etiketatze-prozesuari dagokionez, anafora pronominalak eskuz identifikatuak izan dira, eta haien aurrekariekin duten lotura eskuz markatua izan da. Markatutako anafora pronominalak **hau**, **hori** eta **hura** determinatzaile erakusleak eta beraien pluraleko formak, **hauek**, **horiek** eta **haiek** dira. Guztira, 349 anafora pronominal eta dagozkien aurrekariak besterik ez daude markatuta. Esperimentuak corpus etiketatu handiak erabiliz egitea garrantzitsua da, baina corpusak eskuz etiketatzea lan nekeza eta garestia gertatzen da (Poesio et al., 2008). Besterik ezean, euskarazko anafora pronominala ebazteko lana corpus hori erabiliz egin da.

Corpuseko testuak analizatzaile morfologiko eta sintagma zatikatzaileen (phrase chunker) bidez analizatuak izan dira eta hitzen informazio morfosintaktikoa lortu da. Tamalez, hitz-adierak ez daude desanbiguatuta. Hitz polisemikoetarako analisi bat baino gehiago ematen da eta halako kasuetan, hitzaren lehenengo adierari dagokion analisia da ontzat eman dena, nahiz eta jakin hori ez dela kasu guztietan analisi egokia. Ondorioz, hainbat instantziaren definiziorako erabilitako informazio linguistikoa ez da guztiz zuzena.

VI.1.1 Ikasketa Automatikorako corpusa. Ezaugarri linguistikoak

Testuez osatutako corpus etiketatutik instantzia positiboz eta negatiboz osatutako corpus bat sortu da. Ikasketa Automatikorako prestatutako corpus horretan, instantzia bakoitza anafora batek eta bere aurrekari izan daitekeen izen-sintagma batek (*aurrekarigai* hemendik aurrera) osatzen dute. Instantziak (Soon et al., 2001) artikuluan proposatutako ereduari jarraituz sortu dira, hau da, 349 anafora pronominalak haien aurrekariekin bikotea osatuz instantzia positiboak sortu dira. Instantzia negatiboak sortzeko, anafora pronominal bakoitzaren eta dagokion aurrekariaren artean dagoen izen-sintagma bakoitza anaforarekin elkartuz sortu dira anafora-aurrekarigai pareak, guztira 619. Hortaz, esperimentuetan erabilitako corpusak guztira 968 instantzia ditu; instantzia positibo bakoitzeko bi instantzia negatibo, gutxi gorabehera.

Instantzia bakoitza honako 16 ezaugarri linguistikoen bidez deskribatua izan da:

- Aurrekarigaiari buruzkoak: hitza, lema, kategoria sintaktikoa, deklinabide kasua, numeroa, gradua, izen-sintagmaren mota, funtzio sintaktikoa eta entitate-mota.
- Anafora pronominalari buruzko ezaugarriak: deklinabide kasua, funtzio sintaktikoa, izen-sintagmaren mota eta numeroa.
- Anaforaren eta aurrekarigaiaren arteko loturari buruzkoak: distantzia (izen-sintagma kopurua), esaldi berean dauden ala ez eta numero bera duten ala ez.

Ikusten den bezala, generoa ez da erabilitako ezaugarri linguistikoen artean ageri. Beste hizkuntza batzuetan generoak emandako informazioa anafora pronominala ebazteko garrantzitsua bada ere, euskarazko *hau*, *hori*, *hura*, *hauek*, *horiek* eta *haiek* izenordain eta determinatzaile erakusleek ez dute genero bereizketarik egiten. Informazio morfologikoa erabiltzea, aldiz, ezinbestekoa gertatzen da, euskara hizkuntza eranskaria den heinean.

VI.1.2 Esperimentuak eta argitalpenak

Ikasketa Automatikorako corpusa sortu den moduagatik, Anafora Pronominala ebaztea sailkatze-problema bitar moduan emana dator. Instantzia bat emanik, klase positibokoa ala negatibokoa den erabaki behar da, hau da, aurrekarigai den izen-sintagma anaforaren aurrekaria den ala ez. Atal honetan azaltzen diren esperimentuetan Weka softwarea erabili da.

Lehenengo urrats moduan, ezaugarri linguistikoen garrantzia neurtu da, hau da, instantzia bat zein klasekoa den erabakitzeko unean ezaugarri bakoitzak emandako informazioa zenbaterainoko garrantzitsua den. Esperimentu honetan lortutako emaitzak "Sociedad Española para el Procesamiento del Lenguaje Natural" (SEPLN) elkarteak 2010ean Valentzian ospatutako "Evaluación de Tecnologías de Lenguaje Humano para Lenguas de la Península Ibérica", (IBEREVAL-2010) biltzarrean aurkeztu dira.

- Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque. Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga, Basilio Sierra, Ana Zelaia. SEPLN, 45:291-294, (2010). (Arregi et al., 2010a).

Bigarren esperimentu batean, anafora-aurrekarigai instantziak sailkatzeari ekin zaio. Hainbat proba desberdin egin dira, ezaugarri multzo desberdinekin eta sailkatzaile desberdinekin jokatuz. Esperimentu honetan lortutako emaitzak 2010ean Argentinan ospatutako "Ibero-American Conference on Artificial Intelligence" (IBERAMIA-2010) biltzarrean aurkeztu dira.

- A First Machine Learning Approach to Pronominal Anaphora Resolution in Basque. Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga, Basilio Sierra, Ana Zelaia. Lecture Notes in Artificial Intelligence, LNAI 6433, pp. 234-243, (2010). (Arregi et al., 2010b).

Oinarrizko sailkatzaileekin lortutako emaitzak hobetzeko asmoz, sailkatzaileak konbinatzea erabaki da. Multi-sailkatzaileekin egindako esperimentuan hasierako 16 ezaugarriekin egin da lan eta lortutako emaitzak 2010ean Brasilen ospatutako "15th Iberoamerican Congress on Pattern Recognition" (CIARP 2010) biltzarrean aurkeztu dira.

- A Combination of Classiffiers for the Pronominal Anaphora Resolution in Basque. Ana Zelaia, Basilio Sierra, Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga. Lecture Notes in Computer Science, LNCS 6419, pp. 253-260, (2010). (Zelaia et al., 2010).

Weka paketea erabiliz lortutako emaitza horiek oinarri hartuta, hurrengo helburua LSIrekin esperimentatzea izan da, instantziak LSI/SVD dimentsioetan adieraziz eta Testu-Sailkatze atazan eta Hitzen Adiera-Desanbiguatze atazan emaitza onak eman dituen Bagging multi-sailkatzaile bidezko sailkatze-metodologia bera aplikatuz emaitzak hobetzen saiatzeko.

Esperimentu desberdinak egin dira. Alde batetik, ezaugarrien azpimultzo desberdinekin egin dira probak: 16 ezaugarriak erabiliz eta aurreko esperimentuetan emaitza onenak eman dituzten 14 ezaugarriak besterik ez erabiliz (*hitza* eta *lema* kenduta). Bestalde, terminoak aukeratzeko estrategia desberdinekin esperimentatu da. Termino moduan corpuseko hitz guztiak aukeratu direnean 1000 termino inguru aukeratuak izan dira; corpusean gutxienez 2 aldiz agertzen direnak aukeratu direnean, aldiz, 100 baino gutxiago. Terminoen aukeraketarako bi estrategiekin matrizeek balio singular kopuru oso altua dute, maximotik gertu. Dimentsio-murrizketarako balio desberdinekin eta sailkatzaile kopuru desberdinekin ere egin dira probak.

Bost geruzako balidazio gurutzatua erabili da sailkatzailearen ebaluaziorako metodo moduan. Horretarako, corpuseko 968 instantziak bost geruza disjuntutan banatu dira. Instantzien %36a klase positibokoa izanik, proportzio bera mantenduz antolatu dira instantziak bost geruzetan: 194 instantzia inguruko geruzak izanik 775 instantzia inguruko entrenamendu corpusak sortu dira.

Bost entrenamendu corpus horietarako termino-dokumentu matrize bana sortu da LSI erabiliz eta log-entropy ponderazioa aplikatu da. Anafora-aurrekarigai pareak (instantziak) zutabeetan eta termino moduan aukeratuak izan diren ezaugarrietarako balioak errenkadetan. Testu-Sailkatze atazan eta Hitzen Adiera-Desanbiguatze atazan dokumentuak batez ere bag-of-words adierazpena erabiliz adierazi badira ere (terminoen agerpen maiztasunen bidez, alegia), orain matrizeak bitarrak dira, eta oso dentsitate gutxikoak (sparse). Tamalez, lortu diren emaitzak ez dira onak izan. Asmatze-tasa %75 ingurukoa eta F_1 neurria %55 ingurukoa.

Emaitzak kaskarrak dira eta corpuseko instantzia positibo kopuru txikia eta terminoen agerkidetza eskasa erabakigarriak izan direla uste dugu. Hala ere, ez da ahaztu behar atazaren zailtasun maila altua dela. Azter dezagun berriro ere kapitulu honen sarrerako VI.2. Irudiko adibidea, 148. orrialdean. Esan dugunez, **hark** anafora pronominalaren aurrekaria **Keizo Obuchi** da. Gizakiontzat ez da hain zaila aurrekaria **Yoshiro Mori** ez dela ondorioztatzea, tronbosia zer den eta ze ondorio eragin ditzaken ulertzen badugu: ardura-karguan ordezkatua izan beharra eta kargua hartuko duena hautatu behar izatea, kasu honetan. Hori guztia oinarrizko ezagutza (background knowledge) da, ezaugarri linguistikoen bidez emandako informazioarekin batera ezinbestekoa gertatzen dena, lotura egokiak egin eta sistemak anafora pronominala zuzen ebatz dezan.

VI.2 Korreferentzia-Ebaztea. Ingelesezko corpusa

Ikerketa-lan honetan, Korreferentzia-Ebaztearen atazarako OntoNotes corpusa erabili da. Anotazio linguistikoak dituen corpus bat da, gaur egun oso erabilia izaten ari dena. Corpus etiketatu hori sortu aurretik baziren beste batzuk, "Message Understanding Conference" biltzarretan erabilitako MUC corpusa (Grishman and Sundheim, 1996), (Hirschman and Chinchor, 1998) edota "Automatic Content Extraction" programaren barruan sortutako ACE corpusa (Doddington et al., 2004), adibidez. Azken urteotan, hizkuntzalaritza Konputazionalaren arloko artikulu zientifikoez osatutako "ACL Anthology" corpusa ere sortu da korreferentziari buruzko anotazioekin (Schäfer et al., 2012). Duela urte batzuk, etiketatzaileen arteko adostasun maila altuagoa izango zuen corpus handiago baten beharra sumatu zen, eta aurreko esperientzietatik ikasitakoarekin eta informazio linguistiko aberatsagoa etiketatzeko helburuarekin Ontonotes corpusa sortu zen.

Ontonotes corpusean informazio sintaktikoa, semantikoa eta diskurtsoari buruzkoa dago etiketatua, hainbat geruza bateragarritan (Pradhan et al., 2007a). Geruza horietako batean dago korreferentzia anaforiko orokorra; entitateak eta gertakariak (Pradhan et al., 2007c). Etiketatze-prozesua bi etiketatzaile desberdinek eskuz egindako anotazioetan oinarritzen da. Nabarmentzekoa da antolatzaileen arteko adostasuna altua dela, %90etik gorakoa (Hovy et al., 2006).

Corpusa hiru hizkuntza desberdinetako testuek osatzen dute (ingelesa, txinera eta arabiera) eta genero desberdinetakoak dira: broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) eta web data (WB). Atal honetan aurkezten diren esperimentuetan OntoNotes v4.0 bertsioaren ingeleseko zatia erabili da, 2011. urtean "Modeling Unrestricted Coreference in OntoNotes" izenburupean ospatutako "Computational Natural Language Learning" (CoNLL-2011) atazan erabili zena[1]. Corpusari buruzko informazio zabala LDC katalogoetako dokumentazioan[2] aurki daiteke, bai etiketatze orokorrari[3] buruz eta baita korreferentziaren etiketatzeari[4] buruz ere.

[1] CoNLL-2011 Shared Task, http://conll.cemantix.org/2011/introduction.html
[2] Linguistic Data Consortium (LDC), https://www.ldc.upenn.edu/
[3] https://catalog.ldc.upenn.edu/docs/LDC2011T03/OntoNotes-Release-4.0.pdf
[4] https://catalog.ldc.upenn.edu/docs/LDC2011T03/coreference/english-coref.pdf

VI.2.1 Ikasketa Automatikorako corpusa. Ezaugarri linguistikoak

CoNLL-2011ko webgunetik[5] jaitsitako fitxategi-eskeletoetan Ontonotes corpuseko hitzak txertatuz lortu dira entrenamendurako, parametroen optimizaziorako eta testerako corpus etiketatu ofizialak (training, development, testing). Ikasketa Automatikorako corpusak sortzeko, korreferentziazko etiketa duten aipamen guztiekin aipamen-pareak (mention-pair) osatu dira, baina ez Anafora Pronominalaren atazan anafora-aurrekarigai pareak osatzeko erabilitako estrategia berarekin, RelaxCor sistemaren garatzaileek proposatutakoarekin baizik. Gainera, sistema horrek Korreferentzia-Ebaztearen atazarako erabiltzen dituen ezaugarri linguistikoak hartu dira kontuan.

RelaxCor[6] Ikasketa Automatikoan oinarritzen den korreferentzia ebazteko sistema bat da eta hainbat txapelketetan hartu du parte, tartean CoNLL--2011 atazan ere, bigarren postua lortuz (Sapena et al., 2011). Sistema horretan ezaugarri linguistikoen bidez ematen da korreferentziazko aipamenei buruzko informazioa (Sapena et al., 2013). Guztira 127 ezaugarri linguistiko bitar definitzen dira, eta ematen duten informazioa honela labur daiteke: distantziari eta posizioari buruzkoak (aipamenen arteko distantzia, esaldi kopuruan neurtua, sintagma kopuruan neurtua, esaldiko lehenengo aipamena den, etab.), ezaugarri lexikalak (kontuan hartzen dituzte aipamenen arteko kate-parekatze edo string matching-ak, izenordain izan eta biak kate--parekatzen diren, etab.), ezaugarri morfologikoak (aipamenek numero bera duten, genero bera duten, hirugarren pertsona den, etab.), ezaugarri sintaktikoak (aipamen bat bestearen barruan dagoen, izen-sintagma mugatua den, izen-sintagma mugagabea den, etab.) eta ezaugarri semantikoak (klase semantiko berekoak diren, lehen aipamena pertsona, lekua edo erakundea den, etab.).

Esan bezala, atal honetan aurkezten den lanean, RelaxCor sisteman aipamen-pareak osatzeko erabili den estrategia bera aplikatzea eta ezaugarri linguistiko berberekin lan egitea erabaki da. Horrela, korreferentziazko etiketa duten aipamen guztiekin aipamen-pareak osatu dira fitxategi guztietan: etiketa bereko aipamen-pareekin instantzia positiboak eta etiketa desberdinekoekin negatiboak.

Egia esan, Ikasketa Automatikorako corpusak prestatzea korapilatsua izan da. Alde batetik, fitxategietako aipamen guztiak gainerako denekin konbinatzeko estrategia erabiltzeagatik, instantzia negatibo gehiegi sortu dira, batez ere aipamen kopuru altuko fitxategietan. Instantzia positiboen eta negatiboen arteko oreka aurkitzearren, distantzia bat definitu eta horren

[5] http://conll.cemantix.org/2011/data.html
[6] http://nlp.lsi.upc.edu/relaxcor/

arabera hainbat instantzia negatibo ezabatu dira, training, development eta testing instantzia-fitxategietan. Bestalde, sortu diren instantzia negatibo asko positiboekin kontraesanean zeuden, hau da, ezaugarri linguistiko guztietarako balio berberak izanik kontrako klasekoak ziren. Instantzia positiboek jatorrizko corpusean etiketatutako korreferentzia errealak adierazten dituzte, Ontonotes corpusak duen fidagarritasun mailarekin. Negatiboak aldiz, Ikasketa Automatikorako instantzia negatiboak sortzeko egindako aipamenen parekatzetik datoz. Hori dela eta, training, development eta testing instantzia-fitxategietan kontraesanean dauden instantzia negatibo guztiak ezabatzea erabaki da. RelaxCor sistemaren garatzaileek ere aipatzen dute gehiegizko instantzia negatiboen eta positiboekin kontraesanean dauden instantzia negatiboen sorrera eta antzeko estrategiak erabili dituzte aurreprozesaketa moduan.

Instantzia-fitxategiak bilduz bost generoetarako corpusak sortu direnean, aurreprozesaketa sakonagoa egiteko beharra sumatu da. Izan ere, corpus handiegiak sortu dira eta instantzia gehiegi izateak arazo konputazionalak eman ditu LSIrekin sortutako matrizeen SVD deskonposaketa kalkulatzeko; 4 milioitik gora instantzia BC generoan, adibidez. Arazoa konpontzeko, corpusak txikitzea erabaki da, corpuseko instantzia positiboen eta negatiboen proportzioa mantenduz eta instantzien aukeraketa ausaz eginez. Horrez gain, corpusetan instantzia asko errepikatuta agertzen direla egiaztatu da. BN generoan, adibidez, entrenamendu corpuseko instantzien %46,5a errepikatuta zeuden. Errepikatutako instantziak corpusetatik ezabatu dira, test corpusetako instantzia positibo errepikatuak izan ezik. Guztira, VI.1. Taulan laburbiltzen diren training, development and testing tamainetako corpusak erabili dira bost generoetarako egindako esperimentuetan.

Bost entrenamendu corpus horietarako termino-dokumentu matrize bana sortu da LSI erabiliz eta log-entropy ponderazioa aplikatu da. Aipamen-pareak (instantziak) zutabeetan eta ezaugarrietarako balioak errenkadetan. Terminoak aukeratzeko estrategia moduan corpuseko hitz guztiak aukeratzearena erabili da, eta oraingoan ere, Anafora Pronominala Ebaztearen atazan bezala, matrize bitarrak eta oso dentsitate gutxikoak (sparse) eraiki dira. VI.2. Taulan ikus daiteke matrizeen tamaina eta aurkitutako balio singular kopurua. Kasu guztietan, zutabe kopurua errenkada kopurua baino askoz handiagoa da.

Matrizeen balio singular kopurua oso txikia da, eta neurri batean espero zitekeen. Izan ere, terminoak aukeratzeko estrategiagatik ezaugarri bakoitzerako bi balioek matrizean errenkada bat izan dezakete esleituta. Matrizeetan ez daude 254 errenkada, ezaugarri batzuentzat ez delako agerpenik existitzen bi balioetarako. BC generoan, adibidez, 27 ezaugarri bitar dau-

	BC	BN	MZ	NW	WB
Training (+)	20206	44515	25103	31034	24501
Training (-)	26623	55921	23568	50687	26948
Development (+)	4056	5920	3873	4776	3531
Development (-)	5831	8609	4864	7615	5732
Testing (+)	29363	10771	3918	15857	17146
Testing (-)	16591	12480	3209	15759	5505

VI.1 Taula: Corpusen tamaina. Instantzia positibo (korreferente direnak) eta negatibo (ez direnak). Generoak: broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) eta web data (WB).

	BC	BN	MZ	NW	WB
Terminoak (ezaugarrien balioak)	227	230	227	229	230
Dokumentuak (aipamen-pareak)	46829	100436	48671	81721	51449
Balio singularrak	83	86	85	86	87

VI.2 Taula: Terminoak, dokumentuak eta balio singularrak bost generoetan.

de corpuseko instantzia guztietan balio bera hartzen dutenak. Horrek 1ekoz osatutako 27 errenkada esan nahi du. Termino horiek ez dute dokumentuen arteko erlazio semantikoei buruz informaziorik ematen. Gainerako 200 errenkadak binaka elkarren osagarriak dira, ezaugarriak bitarrak direlako. Hortaz, matrizearen heina gehienez 100ekoa izango da. SVD deskonposaketan 83 balio singular besterik ez badira aurkitu, oraindik ere matrizean mendekotasun lineala dagoela esan nahi du. Matrizea hein urrikoa da (rank deficient matrix), informazio aldetik eduki eskasekoa. Berdinak diren zutabeak ez daude aurreprozesaketan ezabatuak izan direlako, baina geratu direnen artean mendekotasun lineala dago. Ezaugarri linguistikoen artean ere mendekotasun lineala dagoela ikusten da.

Hasiera batean, balio singular kopurua txikia izatea ez da arazo bat. SVD deskonposaketan halako mendekotasun linealak detektatzen dira, eta dimentsio esanguratsuenak kalkulatzen dira. Benetan garrantzitsua dena terminoen agerkidetza dokumentuen artean aberatsa izatea da, horri esker lortuko baita matrize originaleko zutabeen konparaketatik aurkituko ez liratekeen antzekotasun semantikoak azaleratzea. Agerkidetzen jokoa zenbat eta aberatsagoa, orduan eta hobeak izango dira lortutako dimentsioak.

Matrizeak sortu eta SVD deskonposaketa egitearen kostu konputazionala[7] oso txikia izan da genero guztietarako; 20 minututik behera kasu guztietan.

VI.2.2 Esperimentuak eta argitalpenak

Ikasketa Automatikorako corpusak sortu diren moduagatik, Korreferentzia-Ebaztearen problema hau sailkatze-problema bitarra da. Instantzia bat emanik, klase positibokoa ala negatibokoa den erabaki behar da, hau da, aipamen-parea osatzen duten bi aipamenak korreferente diren ala ez. LSI/SVD dimentsioetan adierazitako instantziak sailkatzeko, k-NN sailkatzaileez osatutako Bagging multi-sailkatzailea erabili da. Oraingoan optimizatu diren parametroak dimentsio-murrizketa eta sailkatzaile kopurua dira, eta lortutako emaitzak beste sailkatzaile batzuekin lortutakoak baino hobeak dira genero batzuetan: %66,10−%76,20 arteko asmatze-tasak eta %67,80−%84,10 arteko F_1 neurriak.

Egindako esperimentuen hasierako fase batean lortutako emaitzak 2015ean Ameriketako Estatu Batuetan ospatutako "North American Chapter of the Association for Computational Linguistics-Human Language Technologies"

[7] IXAko sisx05: Sun SPARC Enterprise M3000 SPARC64 VII+ Quad Core @ 2,75 Ghz

(NAACL-HLT 2015) biltzarreko "Vector Space Modeling for Natural Language Processing" (VSM-NLP) tailerrean aurkeztu dira.

- A Multi-classifier Approach to support Coreference Resolution in a Vector Space Model. Ana Zelaia, Olatz Arregi, Basilio Sierra. Proceedings of NAACL-HLT, Workshop on Vector Space Modeling for Natural Language Processing, 17-24, Denver, Colorado, Ameriketako Estatu Batuak (2015). (Zelaia et al., 2015b).

Esperimentu osoaren deskribapena biltzen duen artikulua "Engineering Applications of Artificial Intelligence" aldizkarian argitaratua izan da.

- Combining Singular Value Decomposition and a Multi-Classifier: a New Approach to Support Coreference Resolution. Ana Zelaia, Olatz Arregi, Basilio Sierra. Engineering Applications of Artificial Intelligence, Vol. 46, Part A, Pages 279-286 (2015). (Zelaia et al., 2015a). Inpaktu-faktorea: 2.207 (Q1)

VI.3 Ondorioak

Anafora Pronominala ebazteko eta korreferentzia ebazteko egindako esperimentuetatik ateratako ondorioak horrela labur daitezke:

- Euskarazko corpusean Anafora Pronominala ebazteko egindako lanean emaitza onak lortu dira Ikasketa automatikoko sailkatze-sistema sinpleekin. Tamalez, ikerketa-lan honetan aurkezten den metodologiak ez ditu eman espero zitezkeen emaitzak. Horren arrazoi nagusia corpuseko instantzia positibo kopuru txikia eta terminoen agerkidetza eskasa direla uste dugu. Nahikoa agerkidetza egon ezean, LSI ez da gai adierazpen matematiko egokia kalkulatzeko.

- Sistemak portaera ona erakutsi du Ontonotes corpusean agerpen-pareak sailkatzerakoan. Genero batzuetan, LSI/SVD bidez kalkulatutako dimentsioak erabiliz lortu dira emaitzarik onenak, beste sailkatzaile batzuekin lortutako emaitzak hobetuz. SVD bidez instantzietarako kalkulatutako adierazpenak egokiak direla baiezta daiteke. Dimentsio-murrizketa aplikatu denean emaitzak ez dira hobetu.

- Ebatzi diren bi problemetan, instantziak (anafora-aurrekarigai pareak eta aipamen-pareak) ezaugarri linguistikoen bidez deskribatu dira eta

ondorioz, sortutako matrizeak bitarrak dira eta oso dentsitate gutxikoak. Egia esan, LSI teknika aproposa da dentsitate gutxiko matrizeekin lan egiteko baina maiztasun-matrizeetan aplikatu izan da tradizionalki.

Nahiz eta jatorrizko matrizea bitarra izan, SVD bidezko murrizketarekin lortzen diren adierazpenak dentsoak dira. Horrela, terminorik konpartitzen ez duten dokumentuen arteko loturak aurki daitezke. Alde horretatik, jatorrizko matrizearekin lan egitea baino egokiagoa da SVD aplikatu ondorenekoarekin lan egitea, ezinbestekoak diren terminoen agerkidetzak badaude, noski.

Matrize bitarrekin lan egitean, mendekotasun linealak errazago emango dira. Hori gertatu da lan honetan Korreferentzia-Ebazte atazan. Ondorioz, matrizeak hein urrikoak dira eta informazio aldetik eduki eskasekoak. Horren guztiaren ondorio izan daiteke dimentsio-murrizketa aplikatzean emaitzak ez hobetu izana.

- Matrizeen informazio eskasia ikusita, ahalik eta termino kopuru handiena aukeratzearen estrategia erabili da bi atazetan. Corpus osoan behin bakarrik agertzen diren ezaugarriak termino moduan aukeratzeak ez du laguntzen instantzien arteko erlazio semantikoak aurkitzen, baina lagun dezake test instantzien adierazpena ematen. Terminoak aukeratzeko estrategia hau gomendagarria gertatzen da termino kopurua txikiegia den kasuetan.

VI.4 Argitalpenak

Anafora Pronominala Ebaztea. Euskarazko corpusa

- Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque. Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga, Basilio Sierra, Ana Zelaia. SEPLN, 45:291-294, (2010). (Arregi et al., 2010a).

- A First Machine Learning Approach to Pronominal Anaphora Resolution in Basque. Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga, Basilio Sierra, Ana Zelaia. Lecture Notes in Artificial Intelligence, LNAI 6433, pp. 234-243, (2010). (Arregi et al., 2010b).

- A Combination of Classiffiers for the Pronominal Anaphora Resolution in Basque. Ana Zelaia, Basilio Sierra, Olatz Arregi, Klara Ceberio, Arantza Díaz de Ilarraza, Iakes Goenaga. Lecture Notes in Computer Science, LNCS 6419, pp. 253-260, (2010). (Zelaia et al., 2010).

Korreferentzia-Ebaztea. Ingelesezko corpusa

- A Multi-classifier Approach to support Coreference Resolution in a Vector Space Model. Ana Zelaia, Olatz Arregi, Basilio Sierra. Proceedings of NAACL-HLT, Workshop on Vector Space Modeling for Natural Language Processing, 17-24, Denver, Colorado, Ameriketako Estatu Batuak (2015). (Zelaia et al., 2015b).

- Combining Singular Value Decomposition and a Multi-Classifier: a New Approach to Support Coreference Resolution. Ana Zelaia, Olatz Arregi, Basilio Sierra. Engineering Applications of Artificial Intelligence, Engineering Applications of Artificial Intelligence, Vol. 46, Part A, Pages 279-286 (2015). (Zelaia et al., 2015a). Inpaktu-faktorea: 2.207 (Q1)

Procesamiento del Lenguaje Natural, Revista nº 45, septiembre 2010, pp 291-294 recibido 21-06-10 revisado 05-07-10 aceptado 06-07-10

Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque*

Determinación de características en una aproximación basada en el aprendizaje automático para la resolución de anáforas pronominales en euskara

O.Arregi, K.Ceberio, A.Díaz de Illarraza, I.Goenaga, B.Sierra, A.Zelaia
University of the Basque Country
Manuel Lardizabal pasealekua 1, 20018 Donostia-San Sebastián
olatz.arregi@ehu.es, jipdisaa@si.ehu.es, b.sierra@ehu.es, ana.zelaia@ehu.es

Resumen: En este trabajo presentamos una primera aproximación basada en el aprendizaje automático para resolver la anáfora pronominal en euskara. Asímismo, determinamos las características más relevantes para esta tarea.
Palabras clave: Resolución de anáfora, aprendizaje automático

Abstract: In this paper we present the preliminaries for a machine learning approach to resolve the pronominal anaphora in Basque language. In this work we determine the appropriate features to be used in this task.
Keywords: Anaphora resolution, machine learning

1. Introduction

Pronominal anaphora resolution is related to the task of identifying noun phrases that refer to the same entity mentioned in a document.

Anaphora resolution is crucial in real-world natural language processing applications e.g. machine translation or information extraction. Although it has been a wide-open research field in the area since 1970, the work presented in this article is the first dealing with the subject for Basque, especially in the task of determining anaphoric relationship using a machine learning approach.

The first problem to carry out is the lack of a big annotated corpus in Basque. Mitkov in [5] highlights the importance of an annotated corpus for research purposes: *The annotation of corpora is an indispensable, albeit time-consuming, preliminary to anaphora resolution (and to most NLP tasks or applications), since the data they provide are critical to the development, optimization and evaluation of new approaches.*

Recently, an annotated corpus has been published in Basque with pronominal anaphora tags [2] and thanks to that, this work could be managed.

Although the literature about anaphora resolution with machine learning approaches is very large, we will concentrate on those references directly linked to the work done here. In [10] they apply a noun phrase (NP) coreference system based on decision trees to MUC6 and MUC7 data sets. It is usually used as a baseline in the coreference resolution literature.

The state of the art of other languages varies considerably. In [8] they propose a rule-based system for anaphora resolution in Czech. In [11] the author uses a system based on a loglinear statistical model to resolve noun phrase coreference in German texts. On the other hand, [6] and [7] present an approach to Persian pronoun resolution based on machine learning techniques. They developed a corpus with 2,006 labeled pronouns.

2. Selection of Features

Basque is not an Indo-European language and differs considerably in grammar from languages spoken in other regions around. It is an agglutinative language, in which grammatical relations between components within a clause are represented by suffixes. This is a distinguishing characteristic since morphological information of words is richer than in the surrounding languages. Given that Basque is a head final language at the syntactic level, the morphological information of the phrase, which is considered to be the head,

* This work was supported by KNOW2 (TIN2009-14715-C04-01) and Berbatek (IE09-262) projects.

ISSN 1135-5948

O.Arregi, K.Ceberio, A.Díaz de Illarraza, I.Goenaga, B.Sierra, A.Zelaia

is in the attached suffix. That is why morphosyntactic analysis is essential.

In this work we specifically focus on the pronominal anaphora; concretely, the demonstrative determiners when they behave as pronouns. In Basque there are not different forms for third person pronouns and demonstrative determiners are used as third person pronominals. There are three degrees of demonstratives that are closely related to the distance of the referent: hau (this/he/she/it), hori (that/he/she/it), hura (that/he/she/it). As we will see in the example of Section 2.2 demostratives in Basque do not allow to infer whether the referent is a person (he, she) or it is an impersonal one (it).

Moreover, there is no gender distinction in the Basque morphological system; the gender is not a valid feature to detect the antecedent of a pronominal anaphora.

2.1. Determination of Feature Vectors

In order to use a machine learning method, a suitable annotated corpus is needed. We use part of the Eus3LB Corpus[1] which contains approximately 50.000 words from journalistic texts previously parsed. It contains 349 annotated pronominal anaphora.

In this work, we first focus on features obtainable with our linguistic processing system proposed in [1]. We can not use some of the common features used by most systems due to linguistic differences (i.e. gender). Nevertheless, we use some specific features that linguistic researchers consider important for this task.

The features are grouped in three categories: features of the anaphoric pronoun, features of the antecedent candidate, and features that describe the relationship between both.

- Features of the anaphoric pronoun

f_1 - *dec_ana*: Declension case of anaphor.
f_2 - *sf_ana*: Syntactic function of anaphor.
f_3 - *phrase_ana*: Whether the anaphor has the phrase tag or not.
f_4 - *num_ana*: Number of anaphor.

- Features of the antecedent candidate

f_5 - *word*: Word of antecedent.
f_6 - *lemma*: Lemma of antecedent.
f_7 - *cat_np*: Syntactic category of NP.
f_8 - *dec_np*: Declension case of NP.
f_9 - *num_np*: Number of NP.
f_{10} - *degree*: Degree of the NP that contains a comparative.
f_{11} - *np*: Whether the noun phrase is a simple NP or a composed NP.
f_{12} - *sf_np*: Syntactic function of NP.
f_{13} - *enti_np*: Type of entity.

- Relational features

f_{14} - *dist*: The distance between the anaphor and the antecedent candidate in terms of number of Noun Phrases.
f_{15} - *same_sent*: If the anaphor and the antecedent candidate are in the same sentence.
f_{16} - *same_num*: Besides to singular and plural numbers, there is another one in Basque: the indefinite. Thus, this feature has more than two possible values.

In summary we would like to remark that we include morphosyntactic information in our pronoun features such as the syntactic function it accomplishes, the kind of phrase it is, and its number. We also include the pronoun declension case. We use the same features for the antecedent candidate and we add the syntactic category and the degree of the noun phrase that contains a comparative. We also include information about name entities indicating the type (person, location and organization). The word and lemma of the noun phrase are also taken into account. The set of relational features includes three features: the distance between the anaphor and the antecedent candidate, a Boolean feature that shows whether they are in the same sentence or not, and the number agreement between them.

2.2. Generation of Training Instances

The method we use to create training instances is similar to the one explained in [10]. Positive instances are created for each annotated anaphor and its antecedent. Negative instances are created by pairing each annotated anaphor with each of its preceding noun phrases that are between the anaphor and the antecedent. When the antecedent candidate is composed, we use the information of the last word of the noun phrase to create the features due to the fact that in Basque this word is the one that contains the morphosyntactic information.

[1] Eus3LB is part of the 3LB project [9]

In order to clarify the results of our system, we introduce the following example:

Ben Amor *ere ez da Mundiala amaitu arte etorriko Irunera*, **honek** *ere Tunisiarekin parte hartuko baitu Mundialean.*

(**Ben Amor** *is not coming to Irun before the world championship is finished, since* **he** *will play with Tunisia in the World Championship*).

The word *honek* (he) in bold is the anaphor and *Ben Amor* its antecedent. The noun phrases between them are *Mundiala* and *Irunera*. The next table shows the generation of training instances from the example.

Antecedent	Anaphor	Positive
Ben Amor	honek (he/it)	1
Mundiala	honek (he/it)	0
Irunera	honek (he/it)	0

Generating the training instances in that way, we obtained a corpus with 968 instances; 349 of them are positive, and the rest, 619, negatives.

3. Evaluation

In order to evaluate the performance of our system, we use the above mentioned corpus. Due to the size of the corpus, a 10 fold cross-validation is performed. It is worth to say that we are trying to increase the size of the corpus.

We consider different machine learning paradigms from Weka toolkit [3] in order to find the best system for the task. The classifiers used are: SVM (polynomial kernel), Multilayer Perceptron, Naïve Bayes (NB), k-NN ($k = 1$), Random Forest (RF), NB-Tree and Voting Feature Intervals (VFI). We tried some other traditional methods like rules or simple decision trees, but they do not report good results for our corpus.

Table 1. shows the results obtained with these classifiers.

	Precision	Recall	F-measure
VFI	0,653	0,673	0,663
Perceptron	0,692	0,682	**0,687**
RF	0,666	0,702	0,683
SVM	0,803	0,539	0,645
NB-tree	0,771	0,559	0,648
NB	0,737	0,587	0,654
k-NN	0,652	0,616	0,633

Cuadro 1: Results of different algorithms

The best result is obtained by using the Multilayer Perceptron algorithm, F-measure 68.7 %. In general, precision obtained is higher than recall. The best precision is obtained with SVM (80.3 %), followed by NB-tree (77.1 %). In both cases, the recall is similar, 53.9 % and 55.9 %.

These results are not directly comparable with those obtained for other languages such as English, but we think they are a good baseline for Basque language. We must emphasize that only the pronominal anaphora is treated here, so actual comparisons are difficult.

4. Contribution of Features Used

To better understand which of the features used are more efficient, we evaluate the weight of attributes by different measurements: Information Gain, Relief algorithm, Symmetrical Uncertainty, Chi Squared statistic, and Gain Ratio. The order of features derive from each of the measurements is quite similar in all cases except for the Relief algorithm [4]. Although the first four features are the same in all cases (with slight order variations), the Relief algorithm shows a different order beyond the fifth feature, giving more weight to *word* or *lemma* features than to others relating to anaphor.

Fig. 1. shows the weights of these features taking into account all the measurements used.

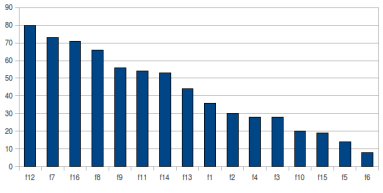


Figura 1: The average weight of features

As expected, the features *word* and *lemma* do not contribute much to the classification process, and we can say that, in general, features relating to the anaphor are not very important for this task, while relational features like *same_num* (agreement in number) or *dist* (distance) appeared to be important. Moreover, all measurements show that features corresponding to the noun phrase are meaningful for this task, as indicated by other authors ([8], [10]).

5. Conclusions and Future Work

This is the first study carried out on resolution of pronominal anaphora in Basque using a machine learning approach. It has been a useful start in defining criteria for anaphora resolution. The results obtained from this work will be helpful for the development of a better anaphora resolution tool for Basque.

We consider seven machine learning algorithms for our first approach in order to decide which kind of method can be the best for this task. The best results are obtained with two classifiers (Random Forest and VFI) which are not the most used for this task in other languages. This may be due to the chosen feature set, the noise of the corpus, and the Basque language characteristics. Traditional methods like SVM, give us a good precision but an F-measure four points below the best system. Anyway, the corpus used in this work is quite small, so we think that the results we obtain can be improved with a larger corpus.

The combination of classifiers has been intensively studied with the aim of improving the accuracy of individual components. We intend to apply a multiclassifier based approach to this task and combine the predictions generated applying a Bayesian voting scheme.

We plan to expand our approach to other types of anaphoric relations with the aim of generating a system to determine the coreference chains for a document.

Finally, the interest of a modular tool to develop coreference applications is unquestionable. Every day more people research in the area of the NLP for Basque and a tool of this kind can be very helpful.

Referencias

[1] Aduriz, I., Aranzabe, M. J., Arriola, J.M., Díaz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.: A Cascaded Syntactic Analyser for Basque. CICLing 2004. Seoul, Korea (2004)

[2] Aduriz, I., Aranzabe, M. J., Arriola, J.M., Atutxa, A., Díaz de Ilarraza, A., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., and Urizar, R.: Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing. Language and Computers, Corpus Linguistics Around the World. Edited by Andrew Wilson, Dawn Archer, Paul Rayson, pp. 1 – 15(15). Rodopi, Netherlands (2006)

[3] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1 (2009)

[4] Kira, K., Rendell, L. A.: A Practical Approach to Feature Selection. Ninth International Workshop on Machine Learning, pp. 249 – 256, (1992)

[5] Mitkov, R.: Anaphora resolution. London: Longman, (2002)

[6] Moosavi, N. S., and Ghassem-Sani, G.: Using Machine Learning Approaches for Persian Pronoun Resolution. Workshop on Corpus-Based Approaches to Coreference Resolution in Romance Languages. CBA-08, (2008)

[7] Moosavi, N. S., and Ghassem-Sani, G.: A Ranking Approach to Persian Pronoun Resolution. Advances in Computational Linguistics. Research in Computing Science 41, pp. 169 – 180, (2009)

[8] Nguy and Zabokrtský.: Rule-based Approach to Pronominal Anaphora Resolution Method Using the Prague Dependency Treebank 2.0 Data. . Proceedings of DAARC 2007 (6th Discourse Anaphora and Anaphor Resolution Colloquium), (2007)

[9] Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M. J., Ageno, A., Martí, M.A. and Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico-semánticos para el catalán, euskera y español. XX. Congreso SEPLN, Barcelona, (2004)

[10] Soon, W. M., Ng, H. T., and Lim, D. C. Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics, 27(4):521 – 544, (2001)

[11] Versley, Y.: A Constraint-based Approach to Noum Phrase Coreference Resolution in German Newspaper Text. In Konferenz zur Verarbeitung Natürlicher Sprache KONVENS, (2006)

A First Machine Learning Approach to Pronominal Anaphora Resolution in Basque

O. Arregi, K. Ceberio, A. Díaz de Illarraza,
I. Goenaga, B. Sierra, and A. Zelaia

University of the Basque Country
olatz.arregi@ehu.es

Abstract. In this paper we present the first machine learning approach to resolve the pronominal anaphora in Basque language. In this work we consider different classifiers in order to find the system that fits best to the characteristics of the language under examination. We do not restrict our study to the classifiers typically used for this task, we have considered others, such as Random Forest or VFI, in order to make a general comparison. We determine the feature vector obtained with our linguistic processing system and we analyze the contribution of different subsets of features, as well as the weight of each feature used in the task.

1 Introduction

Pronominal anaphora resolution is related to the task of identifying noun phrases that refer to the same entity mentioned in a document.

According to [7]: *anaphora, in discourse, is a device for making an abbreviated reference (containing fewer bits of disambiguating information, rather than being lexically or phonetically shorter) to some entity (or entities).*

Anaphora resolution is crucial in real-world natural language processing applications e.g. machine translation or information extraction. Although it has been a wide-open research field in the area since 1970, the work presented in this article is the first dealing with the subject for Basque, especially in the task of determining anaphoric relationship using a machine learning approach.

The first problem to carry out is the lack of a big annotated corpus in Basque. Mitkov in [12] highlights the importance of an annotated corpus for research purposes: *The annotation of corpora is an indispensable, albeit time-consuming, preliminary to anaphora resolution (and to most NLP tasks or applications), since the data they provide are critical to the development, optimization and evaluation of new approaches.*

Recently, an annotated corpus has been published in Basque with pronominal anaphora tags [2] and thanks to that, this work could be managed.

2 Related Work

Although the literature about anaphora resolution with machine learning approaches is very large, we will concentrate on those references directly linked to

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 234–243, 2010.

the work done here. In [20] they apply a noun phrase (NP) coreference system based on decision trees to MUC6 and MUC7 data sets ([15], [16]). It is usually used as a baseline in the coreference resolution literature.

Kernel functions to learn the resolution classifier are applied in [23]. They use structured syntactic knowledge to tackle pronoun resolution, and the results obtained for the ACE dataset show an improvement for all the different domains.

In [22] the authors propose kernel-based methods to resolve three coreference resolution subtasks (binding constraint detection, expletive identification and aliasing). They conclude that using kernel methods is a promising research direction to achieve state of the art coreference resolution results.

A rich syntactic and semantic processing is poposed in [5]. It outperforms all unsupervised systems and most supervised ones.

The state of the art of other languages varies considerably. In [18] they propose a rule-based system for anaphora resolution in Czech. They use the Treebank data, which contains more than 45,000 coreference links in almost 50,000 manually annotated Czech sentences. In [21] the author uses a system based on a loglinear statistical model to resolve noun phrase coreference in German texts. On the other hand, [13] and [14] present an approach to Persian pronoun resolution based on machine learning techniques. They developed a corpus with 2,006 labeled pronouns.

A similar work was carried out for Turkish [24]. They apply a decision tree and a rule-based algorithm to an annotated Turkish text.

3 Selection of Features

3.1 Main Characteristics of Pronominal Anaphora in Basque

Basque is not an Indo-European language and differs considerably in grammar from languages spoken in other regions around. It is an agglutinative language, in which grammatical relations between components within a clause are represented by suffixes. This is a distinguishing characteristic since morphological information of words is richer than in the surrounding languages. Given that Basque is a head final language at the syntactic level, the morphological information of the phrase (number, case, etc.), which is considered to be the head, is in the attached suffix. That is why morphosyntactic analysis is essential.

In this work we specifically focus on the pronominal anaphora; concretely, the demonstrative determiners when they behave as pronouns. In Basque there are not different forms for third person pronouns and demonstrative determiners are used as third person pronominals [11]. There are three degrees of demonstratives that are closely related to the distance of the referent: hau (this/he/she/it), hori (that/he/she/it), hura (that/he/she/it). As we will see in the example of Section 3.3 demostratives in Basque do not allow to infer whether the referent is a person (he, she) or it is an impersonal one (it).

Moreover, demostrative determiners do not have any gender in Basque. Hence, the gender is not a valid feature to detect the antecedent of a pronominal anaphora because there is no gender distinction in the Basque morphological system.

3.2 Determination of Feature Vectors

In order to use a machine learning method, a suitable annotated corpus is needed. We use part of the Eus3LB Corpus[1] which contains approximately 50.000 words from journalistic texts previously parsed. It contains 349 annotated pronominal anaphora.

In this work, we first focus on features obtainable with our linguistic processing system proposed in [1]. We can not use some of the common features used by most systems ([20], [17], [23]) due to linguistic differences. For example the gender, as we previously said. Nevertheless, we use some specific features that linguistic researchers consider important for this task.

The features used are grouped in three categories: features of the anaphoric pronoun, features of the antecedent candidate, and features that describe the relationship between both.

- Features of the anaphoric pronoun
  - f_1 - *dec_ana*: The declension case of the anaphor.
  - f_2 - *sf_ana*: The syntactic function of the anaphor.
  - f_3 - *phrase_ana*: Whether the anaphor has the phrase tag or not.
  - f_4 - *num_ana*: The number of the anaphor.
- Features of the antecedent candidate
  - f_5 - *word*: The word of the antecedent candidate.
  - f_6 - *lemma*: The lemma of the antecedent candidate.
  - f_7 - *cat_np*: The syntactic category of the NP.
  - f_8 - *dec_np*: The declension case of the NP.
  - f_9 - *num_np*: The number of the NP.
  - f_{10} - *degree*: The degree of the NP that contains a comparative.
  - f_{11} - *np*: Whether the noun phrase is a simple NP or a composed NP.
  - f_{12} - *sf_np*: The syntactic function of the NP.
  - f_{13} - *enti_np*: The type of entity (PER, LOC, ORG).
- Relational features
  - f_{14} - *dist*: The distance between the anaphor and the antecedent candidate. Its possible values are from 1 to 15, the maximum distance shown in the corpus from an anaphor to its antecedent. The distance is measured in terms of number of Noun Phrases.
  - f_{15} - *same_sent*: If the anaphor and the antecedent candidate are in the same sentence the value is 0, otherwise the value is 1.
  - f_{16} - *same_num*: Its possible values are 0, 1, 2, and 3. If the anaphor and the antecedent candidate agree in number the value is 3, otherwise the value is 0. When the number of the noun phrase is unknown the value is 1. If the noun phrase is an entity, its number is indefinite and the anaphor is singular, then the value is 2. This last case is needed in Basque because person entities do not have singular or plural tags, but indefinite tag.

[1] Eus3LB is part of the 3LB project [19].

In summary we would like to remark that we include morphosyntactic information in our pronoun features such as the syntactic function it accomplishes, the kind of phrase it is, and its number. We also include the pronoun declension case. We use the same features for the antecedent candidate and we add the syntactic category and the degree of the noun phrase that contains a comparative. We also include information about name entities indicating the type (person, location and organization). The word and lemma of the noun phrase are also taken into account. The set of relational features includes three features: the distance between the anaphor and the antecedent candidate, a Boolean feature that shows whether they are in the same sentence or not, and the number agreement between them.

3.3 Generation of Training Instances

The method we use to create training instances is similar to the one explained in [20]. Positive instances are created for each annotated anaphor and its antecedent. Negative instances are created by pairing each annotated anaphor with each of its preceding noun phrases that are between the anaphor and the antecedent. When the antecedent candidate is composed, we use the information of the last word of the noun phrase to create the features due to the fact that in Basque this word is the one that contains the morphosyntactic information.

In order to clarify the results of our system, we introduce the following example: **Ben Amor** *ere ez da Mundiala amaitu arte etorriko Irunera*, **honek** *ere Tunisiarekin parte hartuko baitu Mundialean.*

(**Ben Amor** *is not coming to Irun before the world championship is finished, since* **he** *will play with Tunisia in the World Championship*).

The word *honek* (he) in bold is the anaphor and *Ben Amor* its antecedent. The noun phrases between them are *Mundiala* and *Irunera*. The next table shows the generation of training instances from the sentence of the example.

Antecedent Candidate	Anaphor	Positive
Ben Amor	honek (he/it)	1
Mundiala	honek (he/it)	0
Irunera	honek (he/it)	0

Generating the training instances in that way, we obtained a corpus with 968 instances; 349 of them are positive, and the rest, 619, negatives.

4 Evaluation

In order to evaluate the performance of our system, we use the above mentioned corpus, with 349 positive and 619 negatives instances. Due to the size of the corpus, a 10 fold cross-validation is performed. It is worth to say that we are trying to increase the size of the corpus.

4.1 Learning Algorithms

We consider different machine learning paradigms from Weka toolkit [6] in order to find the best system for the task. The classifiers used are: SVM, Multilayer Perceptron, NB, k-NN, Random Forest (RF), NB-Tree and Voting Feature Intervals (VFI). We tried some other traditional methods like rules or simple decision trees, but they do not report good results for our corpus.

The SVM learner was evaluated by a polynomial kernel of degree 1. The k-NN classifier, $k = 1$, uses the Euclidean distance as distance function in order to find neigbours. Multilayer Perceptron is a neural network that uses backpropagation to learn the weights among the connections, whereas that NB is a simple probabilistic classifier based on applying Bayes' theorem, and NB-Tree generates a decision tree with naive Bayes classifiers at the leaves. Random forest and VFI are traditionally less used algorithms; however, they produce the best results for our corpus. Random forest is a combination of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [3]. VFI constructs feature intervals for each feature. An interval represents a set of values for a given feature, where the same subset of class values is observed. Two neighbouring intervals contain different sets of classes [4].

4.2 Overall Results

Table 1. shows the results obtained with these classifiers.

Table 1. Results of different algorithms

	Precision	Recall	F-measure
VFI	0.653	0.673	0.663
Perceptron	0.692	0.682	**0.687**
RF	0.666	0.702	0.683
SVM	0.803	0.539	0.645
NB-tree	0.771	0.559	0.648
NB	0.737	0.587	0.654
k-nn	0.652	0.616	0.633

The best result is obtained by using the Multilayer Perceptron algorithm, F-measure 68.7%.

In general, precision obtained is higher than recall. The best precision is obtained with SVM (80.3%), followed by NB-tree (77.1%). In both cases, the recall is similar, 53.9% and 55.9%.

These results are not directly comparable with those obtained for other languages such as English, but we think that they are a good baseline for Basque language. We must emphasize that only the pronominal anaphora is treated here, so actual comparisons are difficult.

5 Contribution of Features Used

Our next step is to determine the attributes to be used in the learning process. When there is a large number of attributes, even some relevant attributes may be redundant in the presence of others. Relevant attributes may contain useful information directly applicable to the given task by itself, or the information may be (partially) hidden among a subset of attributes [10].

To better understand which of the features used are more efficient, we evaluate the weight of attributes by different measurements: Information Gain, Relief algorithm, Symmetrical Uncertainty, Chi Squared statistic, and Gain Ratio. The order of features derive from each of the measurements is quite similar in all cases except for the Relief algorithm [8]. Although the first four features are the same in all cases (with slight order variations), the Relief algorithm shows a different order beyond the fifth feature, giving more weight to *word* or *lemma* features than to others relating to anaphor.

Fig. 1. shows the weight of these features taking into account all the measurements used.

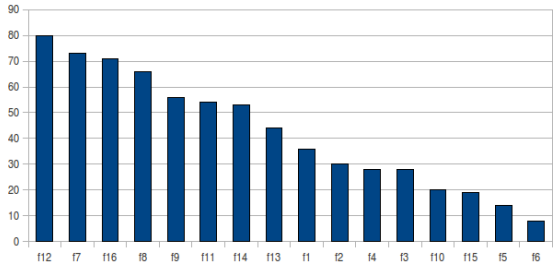


Fig. 1. The average weight of features

As expected, the features *word* and *lemma* do not contribute much to the classification process, and we can say that, in general, features relating to the anaphor are not very important for this task, while relational features like *same_num* (agreement in number) or *dist* (distance) appeared to be important. Moreover, all measurements show that features corresponding to the noun phrase are meaningful for this task, as indicated by other authors.

If we test the algorithms presented in Section 4.1, taking into account the new order of features, and considering smaller subsets of features, the results are similar to the originals. In general, decreasing the number of features gives lower results. The best result (70%) is obtained with 14 features: the original set without the features *word* and *lemma*.

Table 2. shows the best F-measure results obtained with the classifiers mentioned above, taking into account different feature subsets. Only five methods are shown here, due to the fact that results obtained with SVM and NB-tree are not meaningful. SVM method does not improve the first result (64.5%) and NB-tree provides similar results to the ones obtained by simple NB.

Table 2. Results of five algorithms with different number of features

Number of features	VFI	Perceptron	RF	NB	k-nn
16	0.663	0.678	0.683	0.654	0.633
15	0.669	0.669	0.678	0.656	0.648
14	0.671	**0.692** all - $\{f_1, f_2\}$	**0.7** all - $\{f_5, f_6\}$	0.665	0.655
13	0.670	0.678	0.679	0.663	0.666
12	0.669	0.671	0.677	0.665	0.662
11	0.672	0.670	0.690	0.666	**0.674**
10	0.675	0.679	0.674	**0.669**	0.656
9	0.674	0.687	0.679	0.666	0.665
8	0.674	0.672	0.682	0.661	0.661
7	0.677	0.668	0.661	0.655	0.644
6	**0.684**	0.652	0.664	0.650	0.640
5	0.673	0.645	0.652	0.640	0.625
4	0.655	0.619	0.628	0.632	0.600
3	0.646	0.639	0.661	0.619	0.616
2	0.629	0.635	0.626	0.607	0.617

Although the two best results were obtained with 14 features, 69.2% (perceptron) and 70% (RF), the set of attributes selected in both cases is different, since in the first case the best selection of features is produced by the relief algorithm (all features except *sf_ana* and *dec_ana*), and in the second case features were chosen following the order established by the Gain Ratio measurement (all features except *word* and *lemma*). For the rest of the algorithms, the best results are obtained by using a smaller set of attributes (from 6 to 11); nevertheless these results are lower than those mentioned above. For all the algorithms we obtained a higher value than the original F-measure. Table 3. shows these values.

Table 3. Results obtained with different subsets of features

	original F-measure	best F-measure	Number of features
VFI	0.663	0.684	6
Perceptron	0.687	0.692	14
RF	0.683	**0.700**	14
NB	0.654	0.669	10
k-nn	0.633	0.670	11

For the k-NN method the measurement which offers the best results is, in most cases, the Relief algorithm. This result was expected as this algorithm evaluates the weight of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different class. So, given an instance, Relief algorithm searches for its two nearest neighbours,

and the k-NN algorithm is based on the same idea. The selection of the nearest neighbours is crucial in Relief. The purpose is to find the nearest neighbours with respect to important attributes [9].

5.1 The Contribution of Single Attributes

If we use a single attribute each time for the classification process, we can determine that the best attribute is *sf_np*, that is, the syntactic function of the noun phrase, with an F-measure equal to 0.480 but a precision of 0.905.

Table 4. shows the results obtained for this test applying Random Forest algorithm. Unsurprisingly many of the attributes result in zero. It should be noted that as in other works [20], selected attributes provide high values for precision, although the recall is very low. The first four attributes of the table, which are the same as those selected by the measurements introduced at the beginning of this section, provide a precision above 65%, reaching to 90% in the case of the first attribute (*sf_np*). In contrast, the F-measure values are lower than 50%.

Table 4. Results obtained using just one attribute at a time

	Precision	Recall	F-measure
sf_np	0.905	0.327	0.480
cat_np	0.659	0.309	0.421
same_num	0.811	0.123	0.214
dec_np	0.837	0.249	0.384
lemma	0.421	0.381	0.400
word	0.378	0.347	0.362
dist	0.364	0.011	0.022
Rest of attributes	0.000	0.000	0.000

6 Conclusions and Future Work

This is the first study carried out on resolution of pronominal anaphora in Basque using a machine learning approach. It has been a useful start in defining criteria for anaphora resolution. The results obtained from this work will be helpful for the development of a better anaphora resolution tool for Basque.

We consider seven machine learning algorithms for our first approach in order to decide which kind of method can be the best for this task. The best results are obtained with two classifiers (Random Forest and VFI) which are not the most used for this task in other languages. This may be due to the chosen feature set, the noise of the corpus, and the Basque language characteristics. Traditional methods like SVM, give us a good precision but an F-measure four points below the best system. Anyway, the corpus used in this work is quite small, so we think that the results we obtain can be improved with a larger corpus.

We also analyzed the contribution of features used in order to decide which of them are important and which are not. With a good combination of features we obtain an F-measure of 70%, which is the best result obtained in this work.

There are several interesting directions for further research and development based on this work. The introduction of other knowledge sources to generate new features and the use of composite features can be a way to improve the system.

The combination of classifiers has been intensively studied with the aim of improving the accuracy of individual components. We intend to apply a multiclassifier based approach to this task and combine the predictions generated applying a Bayesian voting scheme.

We plan to expand our approach to other types of anaphoric relations with the aim of generating a system to determine the coreference chains for a document.

Finally, the interest of a modular tool to develop coreference applications is unquestionable. Every day more people research in the area of the NLP for Basque and a tool of this kind can be very helpful.

Acknowledgments

This work was supported in part by KNOW2 (TIN2009-14715-C04-01) and Berbatek (IE09-262) projects.

References

1. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Daz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.: A Cascaded Syntactic Analyser for Basque. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 124–134. Springer, Heidelberg (2004)
2. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Atutxa, A., Daz de Ilarraza, A., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., Urizar, R.: Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing. In: Wilson, A., Archer, D., Rayson, P. (eds.) Language and Computers, Corpus Linguistics Around the World, Rodopi, Netherlands, pp. 1–15 (2006)
3. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
4. Demiroz, G., Guvenir, A.: Classification by voting feature intervals. In: 9th European Conference on Machine Learning, pp. 85–92 (1997)
5. Haghighi, A., Klein, D.: Simple Coreference Resolution with Rich Syntactic and Semantic Features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 1152–1161 (2009)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
7. Hirst, G.: Anaphora in Natural Language Understanding. Springer, Berlin (1981)
8. Kira, K., Rendell, L.A.: A Practical Approach to Feature Selection. In: Ninth International Workshop on Machine Learning, pp. 249–256 (1992)
9. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning, pp. 171–182 (1994)
10. Kononenko, I., Hong, S.J.: Attribute Selection for Modeling. Future Generation Computer Systems 13, 181–195 (1997)

11. Laka, I.: A Brief Grammar of Euskara, the Basque Language. Euskarako errektoreordetza, EHU (2000), http://www.ehu.es/grammar
12. Mitkov, R.: Anaphora resolution. Longman, London (2002)
13. Moosavi, N.S., Ghassem-Sani, G.: Using Machine Learning Approaches for Persian Pronoun Resolution. In: Workshop on Corpus-Based Approaches to Coreference Resolution in Romance Languages. CBA 2008 (2008)
14. Moosavi, N.S., Ghassem-Sani, G.: A Ranking Approach to Persian Pronoun Resolution. Advances in Computational Linguistics. Research in Computing Science 41, 169–180 (2009)
15. MUC-6.: Proceedings of the Sixth Message Understanding Conference (MUC-6). Morgan Kaufmann, San Francisco, CA (1995)
16. MUC-7.: Proceedings of the Seventh Message Understanding Conference (MUC-7). Morgan Kaufmann, San Francisco, CA (1998)
17. Ng, V., Cardie, C.: Improving Machine Learning Approach to Coreference Resolution. In: Proceedings of the ACL, pp. 104–111 (2002)
18. Nguy, Zabokrtský: Rule-based Approach to Pronominal Anaphora Resolution Method Using the Prague Dependency Treebank 2.0 Data. In: Proceedings of DAARC 2007 (6th Discourse Anaphora and Anaphor Resolution Colloquium) (2007)
19. Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M.J., Ageno, A., Mart, M.A., Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico-semánticos para el catalán, euskera y español. XX. Congreso SEPLN, Barcelona (2004)
20. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4), 521–544 (2001)
21. Versley, Y.: A Constraint-based Approach to Noum Phrase Coreference Resolution in German Newspaper Text. In: Konferenz zur Verarbeitung Natrlicher Sprache KONVENS (2006)
22. Versley, Y., Moschitti, A., Poesio, M., Yang, X.: Coreference System based on Kernels Methods. In: Proceedings of the 22nd International Coreference on Computational Linguistics (Coling 2008), Manchester, pp. 961–968 (2008)
23. Yang, X., Su, J., Tan, C.L.: Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. In: Proc. COLING/ACL 2006, Sydney, pp. 41–48 (2006)
24. Yldrm, S., Klaslan, Y., Yldz, T.: Pronoun Resolution in Turkish Using Decision Tree and Rule-Based Learning Algorithms. In: Human Language Technology. Challenges of the Information Society. LNCS. Springer, Heidelberg (2009)

A Combination of Classifiers for the Pronominal Anaphora Resolution in Basque

Ana Zelaia Jauregi, Basilio Sierra, Olatz Arregi Uriarte, Klara Ceberio, Arantza Díaz de Illarraza, and Iakes Goenaga

University of the Basque Country
ana.zelaia@ehu.es

Abstract. In this paper we present a machine learning approach to resolve the pronominal anaphora in Basque language. We consider different classifiers in order to find the system that fits best to the characteristics of the language under examination. We apply the combination of classifiers which improves results obtained with single classifiers. The main contribution of the paper is the use of bagging having as base classifier a non-soft one for the anaphora resolution in Basque.

1 Introduction

Pronominal anaphora resolution is related to the task of identifying noun phrases that refer to the same entity mentioned in a document.

According to [5], *anaphora, in discourse, is a device for making an abbreviated reference (containing fewer bits of disambiguating information, rather than being lexically or phonetically shorter) to some entity (or entities).*

Anaphora resolution is crucial in real-world natural language processing applications e.g. machine translation or information extraction. Although it has been a wide-open research field in the area since 1970, the work presented in this article is the first dealing with the subject for Basque, especially in the task of determining anaphoric relationship using a machine learning approach.

Recently, an annotated corpus has been published in Basque with pronominal anaphora tags [2] and thanks to that, this work could be managed.

Although the literature about anaphora resolution with machine learning approaches is very large, we will concentrate on those references directly linked to the work done here. In [10] they apply a noun phrase (NP) coreference system based on decision trees to MUC6 and MUC7 data sets. It is usually used as a baseline in the coreference resolution literature. Combination methods have been recently applied to coreference resolution problems. In [11] the authors use bagging and boosting techniques in order to improve single classifiers results.

The state of the art of other languages varies considerably. In [8] they propose a rule-based system for anaphora resolution in Czech. They use the Treebank data, which contains more than 45,000 coreference links in almost 50,000 manually annotated Czech sentences. In [12] the author uses a system based on a loglinear statistical model to resolve noun phrase coreference in German texts.

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 253–260, 2010.

On the other hand, [6] and [7] present an approach to Persian pronoun resolution based on machine learning techniques. They developed a corpus with 2,006 labeled pronouns.

The paper we present describes a baseline framework for Basque pronominal anaphora resolution using a machine learning approach. In Section 2 some general characteristics of Basque pronominal anaphora are explained. Section 3 shows the results obtained for different machine learning methods. The combination of classifiers is presented in Section 4, and finally, in Section 5, we present some conclusions and point out future work lines.

2 Pronominal Anaphora Resolution in Basque

2.1 Main Characteristics of Pronominal Anaphora in Basque

Basque is not an Indo-European language and differs considerably in grammar from languages spoken in other regions around. It is an agglutinative language, in which grammatical relations between components within a clause are represented by suffixes. This is a distinguishing characteristic since morphological information of words is richer than in the surrounding languages. Given that Basque is a head final language at the syntactic level, the morphological information of the phrase (number, case, etc.), which is considered to be the head, is in the attached suffix. That is why morphosyntactic analysis is essential.

In this work we specifically focus on the pronominal anaphora; concretely, the demonstrative determiners when they behave as pronouns. In Basque there are not different forms for third person pronouns and demonstrative determiners are used as third person pronominals. There are three degrees of demonstratives that are closely related to the distance of the referent: hau (this/he/she/it), hori (that/he/she/it), hura (that/he/she/it). As we will see in the example of Section 2.3 demostratives in Basque do not allow to infer whether the referent is a person (he, she) or it is an impersonal one (it).

Moreover, demostrative determiners do not have any gender in Basque. Hence, the gender is not a valid feature to detect the antecedent of a pronominal anaphora because there is no gender distinction in the Basque morphological system.

2.2 Determination of Feature Vectors

In order to use a machine learning method, a suitable annotated corpus is needed. We use part of the Eus3LB Corpus[1] which contains approximately 50.000 words from journalistic texts previously parsed. It contains 349 annotated pronominal anaphora.

In this work, we first focus on features obtainable with the linguistic processing system proposed in [1]. We can not use some of the common features used by

[1] Eus3LB is part of the 3LB project [9].

most systems due to linguistic differences. For example the gender, as we previously said. Nevertheless, we use some specific features that linguistic researchers consider important for this task.

The features used are grouped in three categories: features of the anaphoric pronoun, features of the antecedent candidate, and features that describe the relationship between both.

- Features of the anaphoric pronoun
  - f_1 - *dec_ana*: The declension case of the anaphor.
  - f_2 - *sf_ana*: The syntactic function of the anaphor.
  - f_3 - *phrase_ana*: Whether the anaphor has the phrase tag or not.
  - f_4 - *num_ana*: The number of the anaphor.
- Features of the antecedent candidate
  - f_5 - *word*: The word of the antecedent candidate.
  - f_6 - *lemma*: The lemma of the antecedent candidate.
  - f_7 - *cat_np*: The syntactic category of the NP.
  - f_8 - *dec_np*: The declension case of the NP.
  - f_9 - *num_np*: The number of the NP.
  - f_{10} - *degree*: The degree of the NP that contains a comparative.
  - f_{11} - *np*: Whether the noun phrase is a simple NP or a composed NP.
  - f_{12} - *sf_np*: The syntactic function of the NP.
  - f_{13} - *enti_np*: The type of entity (PER, LOC, ORG).
- Relational features
  - f_{14} - *dist*: The distance between the anaphor and the antecedent candidate in terms of number of Noun Phrases.
  - f_{15} - *same_sent*: If the anaphor and the antecedent candidate are in the same sentence.
  - f_{16} - *same_num*: Besides to singular and plural numbers, there is another one in Basque: the indefinite. Thus, this feature has more than two possible values.

In summary we would like to remark that we include morphosyntactic information in our pronoun features such as the syntactic function it accomplishes, the kind of phrase it is, and its number. We also include the pronoun declension case. We use the same features for the antecedent candidate and we add the syntactic category and the degree of the noun phrase that contains a comparative. We also include information about name entities indicating the type (person, location and organization). The word and lemma of the noun phrase are also taken into account. The set of relational features includes three features: the distance between the anaphor and the antecedent candidate, a Boolean feature that shows whether they are in the same sentence or not, and the number agreement between them.

2.3 Generation of Training Instances

The method we use to create training instances is similar to the one explained in [10]. Positive instances are created for each annotated anaphor and its antecedent. Negative instances are created by pairing each annotated anaphor with

each of its preceding noun phrases that are between the anaphor and the antecedent. When the antecedent candidate is composed, we use the information of the last word of the noun phrase to create the features due to the fact that in Basque this word is the one that contains the morphosyntactic information.

In order to clarify the results of our system, we introduce the following example: **Ben Amor** *ere ez da Mundiala amaitu arte etorriko Irunera*, **honek** *ere Tunisiarekin parte hartuko baitu Mundialean.*

(**Ben Amor** *is not coming to Irun before the world championship is finished, since* **he** *will play with Tunisia in the World Championship*).

The word *honek* (he) in bold is the anaphor and *Ben Amor* its antecedent. The noun phrases between them are *Mundiala* and *Irunera*. The next table shows the generation of training instances from the sentence of the example.

Antecedent Candidate	Anaphor	Positive
Ben Amor	honek (he/it)	1
Mundiala	honek (he/it)	0
Irunera	honek (he/it)	0

Generating the training instances in that way, we obtained a corpus with 968 instances; 349 of them are positive, and the rest, 619, negatives.

3 Experimental Setup

In order to evaluate the performance of our system, we use the above mentioned corpus. Due to the size of the corpus, a 10 fold cross-validation is performed. It is worth to say that we are trying to increase the size of the corpus.

3.1 Learning Algorithms

We consider different machine learning paradigms from Weka toolkit [4] in order to find the best system for the task. On one hand, we use some typical classifiers like SVM, Multilayer Perceptron, Naïve Bayes, k-NN, and simple decision trees like C4.5 and REPTree. On the other hand, we use classifiers not so frequently used such as Random Forest (RF), NB-Tree and Voting Feature Intervals (VFI).

The SVM learner was evaluated by a polynomial kernel of degree 1. The k-NN classifier, $k = 1$, uses the Euclidean distance as distance function in order to find neigbours. Multilayer Perceptron is a neural network that uses backpropagation to learn the weights among the connections, whereas NB is a simple probabilistic classifier based on applying Bayes' theorem, and NB-Tree generates a decision tree with Naïve Bayes classifiers at the leaves. C4.5 and REPTree are well known decision tree classifiers. Random Forest and VFI are traditionally less used algorithms; however, they produce good results for our corpus. Random forest is a combination of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. VFI constructs feature intervals for each feature. An interval represents a set of values for a given feature, where the same subset of class values is observed. Two neighbouring intervals contain different sets of classes.

3.2 Results for Single Classifiers

The results obtained with these classifiers are shown in Table 1. The best result is obtained by using the Multilayer Perceptron algorithm, an F-measure of 68.7%.

Table 1. Results for different algorithms

	Precision	Recall	F-measure
VFI	0.653	0.673	0.663
Perceptron	0.692	0.682	**0.687**
RF	0.666	**0.702**	0.683
SVM	**0.803**	0.539	0.645
NB-tree	0.771	0.559	0.648
NB	0.737	0.587	0.654
k-NN	0.652	0.616	0.633
C4.5	0.736	0.438	0.549
REPTree	0.715	0.524	0.605

In general, precision obtained is higher than recall. The best precision is obtained with SVM (80.3%), followed by NB-tree (77.1%). Althought C4.5 and REPTree are traditionally used for this task, they do not report good results for our corpus, as it can be observed in the table.

These results are not directly comparable with those obtained for other languages such as English, but we think that they are a good baseline for Basque language. We must emphasize that only the pronominal anaphora is treated here, so actual comparisons are difficult.

4 Experimental Results

In this section the experimental results obtained are shown. It is worth to mention that one of the main contributions of this paper is concerned with the selection of single classifiers in order to perform the combination.

4.1 Combination of Classifiers

Classifier combination is very used in the Machine Learning community. The main idea is to combine some paradigms from the supervised classification trying to improve the individual accuracies of the component classifiers.

According to the architecture used to combine different single classifiers, there are three possible configurations: cascaded, parallel and hierarchical. In this paper we use two parallel combinations of classifiers. One of the ways to combine the classifiers in parallel consists of using several base classifiers, applying them to the database, and then combining their predictions using a vote process. But even with a unique base classifier, it is still possible to build an ensemble, applying it to different training sets in order to generate several different models. A way to get several training sets from a given dataset is bootstrap sampling, which is used in bagging [3].

4.2 Results Obtained

We tried both vote and bagging combination approaches based on the results obtained in the previous section for the single classifiers. We selected five single classifiers, which belong to different paradigms, and which obtain good results for our corpus: Multilayer Perceptron, Random Forest, VFI, NB and k-NN. We performed the experiments in the following way:

- We make a votation with those five classifiers. Three different voting criteria were used: Majority, average of probabilities and product of probabilities.
- We apply the bagging multiclassifier with those five single classifiers, using different number of classifiers: 10, 15, 20, 30 and 40.

Results obtained by applying the vote combination schema are shown in Table 2. As it can be seen a slight increase in results is obtained with the majority voting achieving an F-measure of 69.2%.

Table 2. Results for different voting criteria

Classifier voting criteria	F-measure
Majority voting	0.692
Vote: average of probabilities	0.684
Vote: product of probabilities	0.636

The bagging multiclassifier is supposed to obtain better results when "soft" base classifiers are used. Classification trees are a typical example of soft classifier. That is why, for comparison reasons, we applied a bagging combination of C4.5 and REPTree trees. In Table 3 just the best results obtained from the bagging process for each classifier are shown. Although it is not recommended, we applied bagging to the selected classifiers, some of which are not considered to be "soft". As it can be seen, results obtained using classification trees are worse than those obtained with the selected classifiers. However, they are the single classifiers which obtain the highest benefit from the combination.

The best result is obtained by the multilayer perceptron classifier as the base one, obtaining an F-measure of 70.3%.

Table 3. Results for the bagging multiclassifier

	Single	Bagging
C4.5	0.549	0.654
REPTree	0.605	0.657
VFI	0.663	0.664
Perceptron	0.687	**0.703**
RF	0.683	0.702
NB	0.654	0.654
k-NN	0.633	0.634

5 Conclusions and Future Work

This paper presents a study carried out on resolution of pronominal anaphora in Basque using a machine learning multiclassifier. The results obtained from this work will be helpful for the development of a better anaphora resolution tool for Basque.

We considered nine machine learning algorithms as single classifiers in order to decide which of them select to combine in a parallel manner. Two different classifier combination approaches were used: vote and bagging. The main contribution of the paper is the use of bagging having as base classifier a non-soft one for the anaphora resolution in Basque.

There are several interesting directions for further research and development based on this work. The introduction of other knowledge sources to generate new features and the use of composite features can be a way to improve the system.

We plan to expand our approach to other types of anaphoric relations with the aim of generating a system to determine the coreference chains for a document.

Finally, the interest of a modular tool to develop coreference applications is unquestionable. Every day more people research in the area of the NLP for Basque and a tool of this kind can be very helpful.

Acknowledgments

This work was supported in part by KNOW2 (TIN2009-14715-C04-01) and Berbatek (IE09-262) projects.

References

1. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Daz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.: A cascaded syntactic analyser for basque. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 124–134. Springer, Heidelberg (2004)
2. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Atutxa, A., Daz de Ilarraza, A., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., Urizar, R.: Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing. In: Wilson, A., Archer, D., Rayson, P. (eds.) Language and Computers, Corpus Linguistics Around the World, Rodopi, Netherlands, pp. 1–15 (2006)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
5. Hirst, G.: Anaphora in Natural Language Understanding. Springer, Berlin (1981)
6. Moosavi, N.S., Ghassem-Sani, G.: Using Machine Learning Approaches for Persian Pronoun Resolution. In: Workshop on Corpus-Based Approaches to Conference Resolution in Romance Languages, CBA 2008 (2008)
7. Moosavi, N.S., Ghassem-Sani, G.: A Ranking Approach to Persian Pronoun Resolution. Advances in Computational Linguistics. Research in Computing Science 41, 169–180 (2009)

8. Nguy, G.L., Zabokrtský, Z.: Rule-based Approach to Pronominal Anaphora Resolution Method Using the Prague Dependency Treebank 2.0 Data. In: Proceedings of DAARC 2007, 6th Discourse Anaphora and Anaphor Resolution Colloquium (2007)
9. Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M.J., Ageno, A., Mart, M.A., Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico-semánticos para el catalán, euskera y español. In: XX. Congreso SEPLN, Barcelona (2004)
10. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4), 521–544 (2001)
11. Vemulapalli, S., Luo, X., Pitrelli, J.F., Zitouni, I.: Using Bagging and Boosting Techniques for Improving Coreference Resolution. Informatica 34, 111–118 (2010)
12. Versley, Y.: A Constraint-based Approach to Noum Phrase Coreference Resolution in German Newspaper Text. In: Konferenz zur Verarbeitung Natrlicher Sprache KONVENS (2006)

A Multi-classifier Approach to support Coreference Resolution in a Vector Space Model

Ana Zelaia
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
ana.zelaia@ehu.eus

Olatz Arregi
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
olatz.arregi@ehu.eus

Basilio Sierra
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
b.sierra@ehu.eus

Abstract

In this paper a different machine learning approach is presented to deal with the coreference resolution task. This approach consists of a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space. The vector representation for mention-pairs is generated using a rich set of linguistic features. The SVD technique is used to generate the reduced dimensional vector space.

The approach is applied to the OntoNotes v4.0 Release Corpus for the column-format files used in CONLL-2011 coreference resolution shared task. The results obtained show that the reduced dimensional representation obtained by SVD is very adequate to appropriately classify mention-pair vectors. Moreover, we can state that the multi-classifier plays an important role in improving the results.

1 Introduction

Coreference resolution deals with the problem of finding all expressions that refer to the same entity in a text (Mitkov, 2002). It is an important subtask in Natural Language Processing that require natural language understanding, and hence, it is considered to be difficult.

A coreference resolution system has to automatically identify the mentions of entities in text and link the corefering mentions (the ones that refer to the same entity) to form coreference chains. Systems are expected to perform both, mention detection and coreference resolution.

Preliminary researches proposed heuristic approaches to the task, but thanks to the annotated coreference corpora made available in the last years and the progress achieved in statistical NLP methods, machine learning approaches to the coreference resolution task are being proposed. (Ng, 2010) presents an interesting survey of the progress in coreference resolution.

In this paper we present a different machine learning approach to deal with the coreference resolution task. Given a corpus with annotated mentions, the multi-classifier system we present classifies mention-pairs in a reduced dimensional vector space. We use the typical mention-pair model, where each pair of mentions is represented by a rich set of linguistic features; positive instances correspond to mention-pairs that corefer. Coreference resolution is tackled as a binary classification problem (Soon et al., 2001) in this paper; the subsequent linking of mentions into coreference chains is not considered. In fact, the aim of our experiment is to measure to what extent working with feature vectors in a reduced dimensional vector space and applying a multi-classifier system helps to determine the coreference of mention-pairs. To the best of our knowledge, there are no approaches to the coreference resolution task which make use of multi-classifier systems to classify mention-pairs in a reduced dimensional vector space.

This paper gives a brief description of our approach to deal with the problem of identifying whether two mentions corefer and shows the results obtained. Section 2 presents related work. In Section 3 our approach is presented. Section 4 presents the case study, where details about the dataset used in the experiments and the preprocessing applied are

Proceedings of NAACL-HLT 2015, pages 17–24,
Denver, Colorado, May 31 – June 5, 2015.

given. In Section 5 the experimental setup is briefly introduced. The experimental results are presented and discussed in Section 6, and finally, Section 7 contains some conclusions and comments on future work.

2 Related Work

Much attention has been paid to the problem of coreference resolution in the past two decades. Conferences specifically focusing coreference resolution have been organized since 1995. The sixth and seventh Message Understanding Conferences (MUC-6, 1995; MUC-7, 1998) included a specific task on coreference resolution. The Automatic Context Extraction (ACE) Program focused on identifying certain types of relations between a predefined set of entities (Doddington et al., 2004) while the Anaphora Resolution Exercise (ARE) involved anaphora resolution and NP coreference resolution (Orăsan et al., 2008).

More recently, SemEval-2010 Task 1 was dedicated to coreference resolution in multiple languages. One year later, in the CoNLL-2011 shared task (Pradhan et al., 2011), participants had to model unrestricted coreference in the English-language OntoNotes corpora and CoNLL-2012 Shared Task (Pradhan et al., 2012) involved predicting coreference in three languages: English, Chinese and Arabic.

Recent work on coreference resolution has been largely dominated by machine learning approaches. In the SemEval-2010 task on Coreference Resolution in Multiple Languages (Recasens et al., 2010), most of the systems were based on these techniques (Broscheit et al., 2010; Uryupina, 2010; Kobdani et al., 2010). The same occurs at CoNLL-2011, where (Chang et al., 2011; Björkelund et al., 2011; dos Santos et al., 2011) were based on machine learning techniques. The advantage of these approaches is that there are many open-source platforms for machine learning and machine learning based coreference systems such as BART (Versley et al., 2008), the Illinois Coreference Package (Bengtson et al., 2008) or the Stanford CoreNLP (Manning et al., 2014), among others.

Nevertheless, rule-based systems have also been applied successfully (Lappin et al., 1994; Mitkov, 1998; Lee et al., 2013). The authors of this last system propose a coreference resolution system that is an incremental extension of the multi-pass sieve system proposed by (Raghunathan et al., 2010). This system is shifting from the supervised learning setting to an unsupervised setting, and obtained the best result in the CoNLL-2011 Shared Task.

Some very interesting uses of vector space models for the coreference resolution task can be found in the literature. (Nilsson et al., 2009) investigate the effect of using vector space models as an approximation of the kind of lexico-semantic and commonsense knowledge needed for coreference resolution for Swedish texts. They also work with reduced dimensional vector spaces and obtain encouraging results. In an attempt to increase the performance of a coreference resolution engine, (Bryl et al., 2010) make use of structured semantic knowledge available in the web. One of the strategies they adopt is to apply the SVD to Wikipedia articles and classify mentions in a reduced dimensional vector space.

3 Proposed Approach

The approach we present consists of a multi-classifier system which classifies mention-pairs in a reduced dimensional vector space. This multi-classifier is composed of several k-NN classifiers. A set of linguistic features is used to generate the vector representations for the mention-pairs. The training dataset is used to create a reduced dimensional vector space using the SVD technique. Mention-pairs in the training, development and test sets are represented using the same linguistic features and projected onto the reduced dimensional space.

The classification process is performed in the reduced dimensional space. To create the multi-classifier, we apply random subsampling and obtain training datasets $TD_1, \ldots, TD_i$ for the reduced dimensional space. Given a testing case q, the k-NN classifier makes a label prediction c^i based on each one of the training datasets TD_i, and predictions $c^1, \ldots c^i$ are combined to obtain the final prediction c_j using a Bayesian voting scheme. It is a binary classification system where the final prediction c_j may be positive (mentions tested corefer) or negative (mentions do not corefer). Figure 1 shows an illustration of the fundamental steps of the experi-

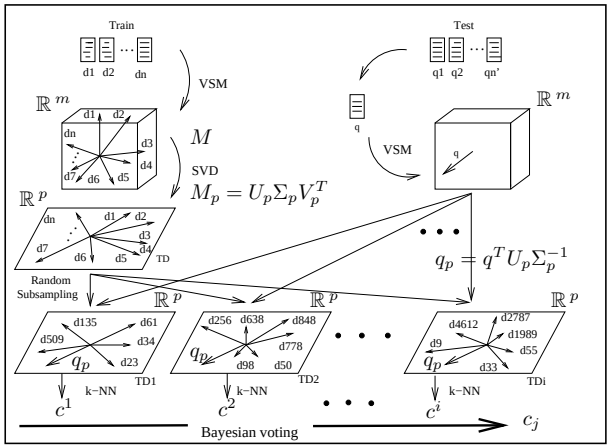


Figure 1: Fundamental steps of the proposed approach. $\mathbb{R}^m$ is the original vector space, $\mathbb{R}^p$ is the reduced dimensional space where vectors are projected. The multi-classifier is composed of several k-NN classifiers. c_j is the final classification label for testing case q.

ment.

In the rest of this section, details about the SVD dimensionality reduction technique, the k-NN classification algorithm, the combination of classifiers and the evaluation measures used are briefly reviewed.

3.1 The SVD Dimensionality Reduction

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization and Information Retrieval tasks. Latent Semantic Indexing (LSI) [1] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional vector space created from a training dataset. To create such a lower dimensional vector space, LSI generates a term-document matrix M and computes its SVD matrix decomposition, $M = U\Sigma V^T$. As a result, r singular values are obtained, and terms and documents are mapped to the r-dimensional vector space. By reducing the r to p, a reduced dimensional space is created, the p-dimensional space onto which vectors are projected. This reduced dimensional space is used for classification purposes, and the cosine similarity is usually used to measure the similarity between vectors (Berry et al., 1995).

[1] http://lsi.research.telcordia.com,http://www.cs.utk.edu/∼lsi

It has been proved that computing the similarity of vectors in the reduced dimensional space gives better results than working in the original space. In fact, LSI is said to be able to capture the latent relationships among words in documents thanks to the word co-occurrence analysis performed by the SVD technique, and therefore, cluster semantically terms and documents. This powerful technique is being used to better capture the semantics of texts in applications such as Information Retrieval (Berry et al., 2005). LSI is referred to as Latent Semantic Analysis (LSA) when it is used as a model of the acquisition, induction and representation of language and the focus is on the analysis of texts (Dumais, 2004).

For the sake of the coreference resolution task, each document corresponds to a mention-pair, and words in each document are the linguistic feature values for the associated mention-pair. Section 4.2 gives details about the linguistic features used to represent each mention-pair. Matrix M is constructed for the selected feature values (terms) and all mention-pairs considered (documents). The SVD decomposition is computed and the p-dimensional reduced space is created. We use U as the reduced dimensional representation, and compute the coordinates to project mention-pair vectors onto the reduced space and compare them.

3.2 The k-NN classification algorithm

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training cases, and uses the class of the k top-ranking neighbors to do the prediction for the testing case being analyzed (Dasarathy, 1991).

In our experiments, parameter k is set to 3. Given a testing mention-pair vector, the 3-NN classifier is used to find the three nearest neighbor mention-pair vectors in the reduced dimensional vector space. The cosine is used to measure vector similarity and find the nearests.

We also consider the k-NN classifier provided with the Weka package (Hall et al., 2009; Aha et al., 1991). We use it to obtain a honest comparison for the results.

3.3 Multi-classifier systems

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training datasets TD_i is generated by selecting n training cases drawn randomly with replacement from the original training dataset TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In fact, this is the approach used in our work and the n_1 parameter is set to be 60% of the total number of training cases n. The proportion of positive and negative cases in the training dataset TD is preserved in the different TD_i datasets generated.

According to the random subsampling, given a testing case q, the classifier makes a label prediction c^i based on each one of the training datasets TD_i. Label predictions c^i may be either positive or negative. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}^i_{c_j}$ is calculated for each training dataset TD_i and label to be predicted. These confidence values are calculated based on the training collection. Confidence values are summed by label; the label c_j that gets the highest value is finally proposed as a prediction for the testing case q.

3.4 Evaluation measures

The approach presented in this paper is a binary classification system where the final prediction c_j may be positive (mentions tested corefer) or negative (mentions do not corefer). There are many metrics that can be used to measure the performance of a classifier. In binary classification problems precision and recall are very widely used. Precision (Prec) is the number of correct positive results divided by the number of all positive results, and recall (Rec) is the number of correct positive results divided by the number of positive results that should have been returned.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines them. The F_1-score can be interpreted as a weighted average of precision and recall; it reaches its best value at 1 and worst score at 0.

$$F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

Accuracy is also used as a statistical measure of performance in binary classification tasks. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases tested.

4 Case study

This section briefly reviews the dataset used in the experiments and the preprocessing applied.

4.1 Dataset

The OntoNotes v4.0 Release Corpus is used in the experiments[2]. It provides a large-scale multi-genre corpus with multiple layers of annotation (syntactic, semantic and discourse information) which also include coreference tags. A nice description of the coreference annotation in OntoNotes can be found in (Pradhan et al., 2007a) and (Pradhan et al., 2007b).

Although OntoNotes is a multi-lingual resource for English, Chinese and Arabic, for the scope of this paper, we just look at the English portion. We

[2]Downloaded from Linguistic Data Consortium (LDC) Catalog No.: LDC2011T03, https://catalog.ldc.upenn.edu/LDC2011T03. For more information, see OntoNotesRelease4.0.pdf and coreference/english-coref.pdf files in LDC directory

use English texts for five different genres or types of sources: broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) and web data (WB).

The English language portion of the OntoNotes v4.0 Release Corpus was used in the CONLL-2011 coreference resolution Shared task[3]. The task was to automatically identify mentions of entities and events in text and to link the corefering mentions together to form mention chains (Pradhan et al., 2011; Pradhan et al., 2012). Since OntoNotes coreference data spans multiple genre, the task organizers created a test set spanning all the genres. The training, development and test files were downloaded from the CONLL-2011 website, and the *_conll files were generated from each corresponding *_skel files using the scripts made available by the organizers.

The *_conll files contain information in a tabular structure where the last column contains coreference chain information. Two types of *_conll files may be generated, depending on how the annotation was generated; *gold_conll files were hand-annotated and adjudicated quality, whereas annotations in *auto_conll files were produced using a combination of automatic tools. *gold_conll files are used in the experiments presented in this paper.

4.2 Preprocessing

In order to obtain the vector representation for each pair of mentions, we used the features defined by (Sapena et al., 2011). The 127 binary features they define are related to distance, position, lexical information, morphological information, syntactic dependencies and semantic features. The authors developed a coreference resolution system called RelaxCor[4] and participated in the CoNLL-2011 shared task obtaining very good results. It is an open source software available for anyone who wishes to use it.

RelaxCor is a constraint-based hypergraph partitioning approach to coreference resolution, solved by relaxation labeling. It generates feature vectors for all mention-pairs in the *_conll files as part of the system and uses them to solve the task. We decided to use the perl scripts distributed by the authors and generate the positive and negative feature vectors for all *_conll files. These feature vectors consist of binary values for the 127 binary features and a label: a positive label (+) indicates that the feature vector corresponds to a corefering mention-pair, whereas a negative label (-) indicates that the two mentions do not corefer.

Note that each mention in a file is combined with all the rest of mentions in the same file to form mention-pairs and consequently, a very large amount of negative examples is generated, specially for large files. We decided to reduce the amount of negative examples, in a similar manner as (Sapena et al., 2011) and therefore, negative examples with more than five feature values different from any positive example in each file were eliminated. In order to obtain the training, development and test corpora for the 5 genres, we brought together the examples generated from files of the same split and genre. We removed contradictions (negative examples with identical feature values as a positive example) and examples that appeared more than once in the same corpus. We noticed that the size of the corpora was too large for some of the genres; the broadcast conversations (BC) genre training corpus for instance had more than 4 million examples. We decided to reduce all corpora to a reasonable size to compute the SVD.

	BC	BN	MZ	NW	WB
Train (+)	20206	44515	25103	31034	24501
Train (-)	26623	55921	23568	50687	26948
Dev (+)	4056	5920	3873	4776	3531
Dev (-)	5831	8609	4864	7615	5732
Test (+)	29363	10771	3918	15857	17146
Test (-)	16591	12480	3209	15759	5505

Table 1: Size of corpora used in the experiments.

Table 1. gives detailed information about the number of positive and negative mention-pairs in the training, development and test corpora used in the experiments. A matrix is constructed for each of the training corpus. Feature values that appear at least once in the corpus are selected as terms. Even though theoretically we could have a maximum number of 254 different terms in each training corpus (127 $\times$ 2, because the 127 features are binary), the real value is between 227 and 230. The

[3] http://conll.cemantix.org/2011/introduction.html
[4] http://nlp.lsi.upc.edu/relaxcor/

sizes of the matrices created are given by the number of terms and documents (sum of (+) and (-) examples in the training corpus) and can be seen in Table 2.

	BC	BN	MZ	NW	WB
Terms	227	230	227	229	230
Docs	46829	100436	48671	81721	51449

Table 2: Size of term-document matrices M.

5 Experimental Setup

To optimize the behaviour of the multi-classifier system, the number of TD_i training datasets is adjusted in a parameter tuning phase. This optimization process is performed in an independent way for each of the genres because the five genres correspond to texts coming from different sources and may have very different characteristics (Uryupina et al., 2012). Therefore, we treat them as five different classification problems.

The five development corpora are used to adjust parameter i (the amount of TD_i training datasets). We experimented with the following values for i: 5, 10, 20, 30, 40, 50, 60, 70, 80. Table 3 shows the optimal values obtained for each genre. This means that testing cases for the BC genre, for instance, are classified by a multi-classifier formed by 60 k-NN classifiers, after having generated 60 TD_i training datasets from the original TD.

	BC	BN	MZ	NW	WB
Optimal i	60	30	50	20	40
Singular Values	83	86	85	86	87

Table 3: Optimal values for the number of TD_i datasets. Number of singular values computed by SVD

Two different dimensional representations are experimented for mention-pair vectors. On the one hand, we consider mention-pair vectors represented in the original 127 dimensions. On the other hand, the SVD-computed dimensional vector representation is being experimented. Table 3 shows the number of singular values (dimensions) computed by SVD for each of the genres.

6 Experimental Results

Three experiments were carried out in the test phase using the optimal values for parameter i and the two different representations for mention-pair vectors. Table 4 shows the results obtained for each of the experiments: accuracy values in a first row (Acc.) and F_1-scores in a second (F_1).

In a first experiment (Exp.1), the Weka 3-NN classifier is applied to classify testing cases represented in the original 127 dimensional space. The same 3-NN classifier is applied in a second experiment (Exp.2), but training and testing cases are represented using the dimensions computed by SVD (see Singular Values in Table 3). In a last experiment (Exp.3), our approach is applied and a multi-classifier system classifies testing vectors in the same SVD-dimensional vector space as in the previous experiment. The multi-classifier is generated according to the optimal values for parameter i in each genre.

Exp.	BC	BN	MZ	NW	WB	Mean
1 Acc.	**0.719**	0.704	**0.706**	0.707	0.669	0.701
F_1	**0.762**	0.686	**0.731**	0.679	0.744	0.720
2 Acc.	0.672	0.725	0.662	0.725	**0.783**	0.713
F_1	0.742	0.71	0.717	0.715	**0.85**	**0.747**
3 Acc.	0.669	**0.755**	0.661	**0.742**	0.776	**0.721**
F_1	0.739	**0.728**	0.707	**0.716**	0.841	0.746

Table 4: Accuracy and F_1-score for the test corpora. Exp.1: 3-NN and 127 dimensions. Exp.2: 3-NN and SVD dimensions. Exp.3: multi-classifier and SVD dimensions. Last column: mean values

The results shown in bold in the first part of Table 4 are the best for each genre. Note that the two performance measures computed (accuracy and F_1-score) are very correlated in the five cases. Taking into account that the proportion of positive and negative examples varies from genre to genre, this correlation gives consistency to the interpretation of the results obtained.

The best results for BC and MZ genres are obtained in the first experiment, applying the 3-NN classifier to the 127 dimensional vectors (Exp.1, F_1-scores: 0.762 and 0.731, respectively). For the rest of the genres, the best results are obtained for the SVD-dimensional vectors. An F_1-score of 0.85 is

obtained for the WB genre in the second experiment (Exp.2). The approach proposed in this paper (Exp.3) achieves the best results for two out of the five genre, with an F_1-score of 0.728 for BN and 0.716 for NW.

The last column in Table 4 shows the mean accuracy and F_1-scores obtained in each experiment, taking into account the five genres as a whole (the best are shown in bold). The best mean F_1-score is obtained in Experiment 2, where vectors are classified in the SVD-dimensional vector space. In fact, this result is very closely followed by the one obtained in Experiment 3 with our approach, (mean F_1-scores: 0.747 and 0.746, respectively). The best mean accuracy is obtained when our approach is applied (mean accuracy: 0.721). This good results seem to suggest that the dimensions computed by the SVD technique are very appropriate to represent mention-pairs and classify them. Moreover, the use of the multi-classifier system gets to achieve even better results, outperforming the ones obtained by the other classification systems.

7 Conclusions and Future Work

In this paper a different machine learning approach to deal with the coreference resolution task is presented: a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space created by applying the SVD technique. The results obtained for the OntoNotes corpus are very good, outperforming the ones obtained by other classification systems for some genres. Moreover, when mean results per experiment are considered, the SVD generated dimensional representation always achieves the best results, which seems to suggest that it is a very robust and suitable representation for coreference mention-pairs.

As future work, we plan to experiment with some other kind of multi-classifer systems and basic classifiers such as SVM. It is important to note that the approach may be applied to corpora in other languages as well.

Acknowledgments

We gratefully acknowledge Emili Sapena, who helped us solve some file format problems. This work was supported by the University of the Basque Country, UPV/EHU, ikerketaren arloko errektoreordetza / Vicerrectorado de Investigación.

References

David W. Aha, Dennis Kibler, and Marc K. Albert. 1991. *Machine Learning*, volume 6(1).

Eric Bengtson and Dan Roth. 2008. *Understanding the value of features for coreference resolution*. Proceedings of the EMNLP '08: 294–303.

Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. 1995. *Using Linear Algebra for Intelligent Information Retrieval*, volume 37(4):573–595. SIAM.

Michael W. Berry and Murray Browne. 2005. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM.

Anders Bjrkelund and Pierre Nugues. 2011. *Exploring lexicalized features for coreference resolution*. Proceedings of the CONLL'11 Shared Task, 45–50.

Leo Breiman. 1996. *Bagging Predictors*. Machine Learning, volume 24(2):123–140.

Samuel Broscheit, Massimo Poesio, Simone Paolo Ponzetto, Kepa Joseba Rodriguez, Lorenza Romano, Olga Uryupina, Yannick Versley, and Roberto Zanoli. 2010. *BART: A multilingual anaphora resolution system*. Proceedings of the SemEval-2010, pages 104–107.

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. *Using Background Knowledge to Support Coreference Resolution*. IOS Press, volume 215:759–764.

Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. 2011. *Inference protocols for coreference resolution*. Proceedings of the CoNLL'11 Shared Task, 40–44.

Belur V. Dasarathy. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press.

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.

Thomas G. Dietterich. 1998. *Machine Learning Research: Four Current Directions*. The AI Magazine, volume 18(4):97–136.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. *The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation*. Proceedings of the LREC-2004, 837–840.

Susan T. Dumais. 2004. *Latent Semantic Analysis*. ARIST (Annual Review of Information Science Technology), volume 38:189–230.

Mark Hall, Eibe Franke, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, volume 11(1):10–18.

Tin K. Ho, Jonathan J. Hull, and Sargur N. Srihari. 1994. *Decision Combination in Multiple Classifier Systems.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 16(1):66–75.

Hamidreza Kobdani and Hinrich Schütze. 2010. *Sucre: A modular system for coreference resolution*. Proceedings of the SemEval-2010, pp. 92–95.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. *Deterministic coreference resolution based on entity-centric, precision-ranked rules*. Computational Linguistics, 39(4):885–916.

Shalom Lappin and Herbert J. Leass. 1994. *An algorithm for pronominal anaphora resolution*. Computational linguistics, 20(4):535–561.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. *The Stanford CoreNLP Natural Language Processing Toolkit*. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 55–60.

Ruslan Mitkov. 1998. *Robust pronoun resolution with limited knowledge*. Proceedings of the COLING'98, volume 2: 869–875.

Ruslan Mitkov. 2002. *Anaphora Resolution*. Pearson Education.

MUC-6. 1995. *Coreference task definition*. Proceedings of the MUC, 335–344.

MUC-7. 1998. *Coreference task definition*. Proceedings of the MUC.

Vincent Ng. 2010. *Supervised Noun Phrase Coreference Research: The First Fifteen Years.* Proceedings of the ACL'10, 1396–1411.

Kristina Nilsson and Hans Hjelm. 2009. *Using Semantic Features Derived from Word-Space Models for Swedish Coreference Resolution*. Proceedings of the NoDaLiDa'09, volume 4:134–141.

Constantin Orăsan, Dan Cristea, Ruslan Mitkov, and António Branco. 2008. *Anaphora Resolution Exercise: an Overview*. Proceedings of the LREC'08.

Sameer Pradhan, Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel 2007a. *Ontonotes: a Unified Relational Semantic Representation*. International Journal of Semantic Computing, volume 1(4):405–419.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007b. *Unrestricted Coreference: Identifying Entities and Events in OntoNotes.* Proceedings of the ICSC, pp. 446–453.

Sameer Pradhan, Martha Palmer, Lance Ramshaw, Ralph Weischedel, Mitchell Marcus, and Nianwen Xue. 2011. *CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes*. Proceedings of the CONLL'11 Shared Task, 1–27.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. *CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes*. Proceedings of the CONLL'12 Shared Task, 1–40. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. *A multi-pass sieve for coreference resolution*. Proceedings of the EMNLP'10, pp. 492–501.

Marta Recasens, Lluís Márquez, Emili Sapena, M. Antónia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. *SemEval-2010 Task 1: Coreference Resolution in Multiple Language*. Proceedings of the SemEval-2010, pp. 1–8.

C. N. dos Santos and D. L. Carvalho. 2011. *Rule and tree ensembles for unrestricted coreference resolution*. Proceedings of the CONLL'11 Shared Task, pp. 51–55.

Emili Sapena, Lluís Padró, and Jordi Turmo. 2011. *RelaxCor Participation in CoNLL Shared Task on Coreference Resolution*. Proceedings of the CONLL'11 Shared Task, pp. 35–39.

Wee M. Soon, Hwee Ng, and Daniel C. Y. Lim. 2001. *A Machine Learning Approach to Coreference Resolution of Noun Phrases*. Association for Computational Linguistics, volume 27(4): 521–544.

Olga Uryupina. 2010. *Corry: A system for coreference resolution*. Proceedings of the SemEval-2010, 100–103.

Olga Uryupina, and Massimo Poesio. 2012. *Domain-specific vs. Uniform Modeling for Coreference Resolution*. Proceedings of the LREC-2012: 187–191.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. *Bart: a modular toolkit for coreference resolution*. Proceedings of the HLT-Demonstrations'08, pp. 9–12.

Engineering Applications of Artificial Intelligence 46 (2015) 279–286

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Combining Singular Value Decomposition and a multi-classifier: A new approach to support coreference resolution

Ana Zelaia *, Olatz Arregi, Basilio Sierra

Faculty of Informatics, University of the Basque Country, UPV/EHU, Manuel Lardizabal Pasealekua 1, 20018 Donostia-San Sebastián, Basque Country, Spain

ARTICLE INFO

Article history:
Received 20 May 2015
Received in revised form
11 September 2015
Accepted 16 September 2015
Available online 23 October 2015

Keywords:
Coreference resolution
Machine learning
Multi-classifier
Singular Value Decomposition
Latent semantic indexing

ABSTRACT

In this paper a new machine learning approach is presented to deal with the coreference resolution task. This approach consists of a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space. The vector representation for mention-pairs is generated using a rich set of linguistic features. The (Singular Value Decomposition) SVD technique is used to generate the reduced dimensional vector space. The approach is applied to the OntoNotes v4.0 Release Corpus for the column-format files used in CONLL-2011 coreference resolution shared task. The results obtained show that the reduced dimensional representation obtained by SVD is very adequate to appropriately classify mention-pair vectors. Moreover, it can be stated that the multi-classifier plays an important role in improving the results.

1. Introduction

Coreference resolution deals with the problem of finding all expressions that refer to the same entity in a text (Mitkov, 2002). It is an important subtask in Natural Language Processing (NLP) tasks that require natural language understanding, and hence, it is considered to be difficult.

A coreference resolution system has to automatically identify the mentions of entities in text and link the corefering mentions (the ones that refer to the same entity) to form coreference chains. Systems are expected to perform both, mention detection and coreference resolution.

Preliminary researches proposed heuristic approaches to the task, but thanks to the annotated coreference corpora made available in the last years and the progress achieved in statistical NLP methods, machine learning approaches to the coreference resolution task are being proposed. In Ng (2010) the authors present an interesting survey of the progress in coreference resolution.

In this paper a new machine learning approach is presented to deal with the coreference resolution task. Given a corpus with annotated mentions, the multi-classifier system presented classifies mention-pairs in a reduced dimensional vector space. The typical mention-pair model is used, where each pair of mentions is represented by a rich set of linguistic features; positive instances correspond to mention-pairs that corefer. In this paper, coreference resolution is tackled as a binary classification problem (Soon et al., 2001); the subsequent linking of mentions into coreference chains is not considered. In fact, the aim of the experiment performed is to measure to what extent working with feature vectors in a reduced dimensional vector space and applying a multi-classifier system helps to determine the coreference of mention-pairs. To the best of our knowledge, there are no approaches to the coreference resolution task which make use of multi-classifier systems to classify mention-pairs in a reduced dimensional vector space.

This paper gives a description of a new approach to deal with the problem of identifying whether two mentions corefer and shows the results obtained. Section 2 presents related work. In Section 3 the new approach is presented. Section 4 presents the case study, where details about the dataset used in the experiments and the preprocessing applied are given. In Section 5 the experimental setup is presented. The experimental results are shown and discussed in Section 6, and finally, Section 7 contains some conclusions and comments on future work.

2. Related work

Much attention has been paid to the problem of coreference resolution in the past two decades. Conferences specifically focusing coreference resolution have been organized since 1995. The sixth

* Corresponding author.
E-mail addresses: ana.zelaia@ehu.eus (A. Zelaia), olatz.arregi@ehu.eus (O. Arregi), b.sierra@ehu.eus (B. Sierra).

http://dx.doi.org/10.1016/j.engappai.2015.09.007
0952-1976/

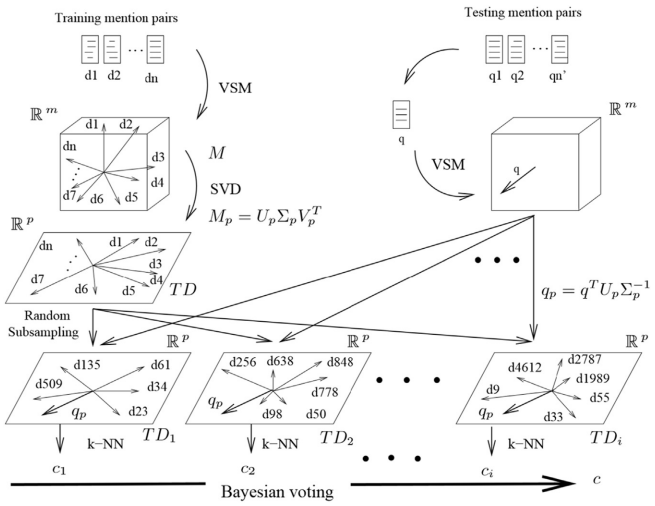


Fig. 1. Fundamental steps of the proposed approach. Original representation for mention-pairs: $\mathbb{R}^m$. SVD-dimensional vector representation computed by LSI: $\mathbb{R}^p$. Multi-classifier constructed based on training datasets $TD_1, \ldots, TD_i$ and several k-NN classifiers. Testing mention-pair q is projected to the SVD-dimensional vector space. Label predictions are combined to compute the final c: $(+)$ mentions corefer, $(-)$ they do not corefer.

and seventh Message Understanding Conferences included a specific task on coreference resolution (MUC6, 1995; Hirschman and Chinchor, 1998). The Automatic Context Extraction (ACE) Program focused on identifying certain types of relations between a predefined set of entities (Doddington et al., 2004) while the Anaphora Resolution Exercise (ARE) involved anaphora resolution and Noun Phrase coreference resolution (Orasan et al., 2008).

More recently, SemEval-2010 Task 1 was dedicated to coreference resolution in multiple languages. One year later, in the CoNLL-2011 shared task (Pradhan et al., 2011), participants had to model unrestricted coreference in the English-language OntoNotes corpora and CoNLL-2012 Shared Task (Pradhan et al., 2012) involved predicting coreference in three languages: English, Chinese and Arabic.

Recent work on coreference resolution has been largely dominated by machine learning approaches. In the SemEval-2010 task on Coreference Resolution in Multiple Languages (Recasens et al., 2010), most of the systems were based on these techniques (Broscheit et al., 2010; Uryupina, 2010; Kobdani and Schütze, 2010). The same occurred at CoNLL-2011, where Chang et al. (2011), Björkelund and Nugues (2011), and Nogueira dos Santos and Lopes Carvalho (2011) were based on machine learning techniques. There are many open-source platforms and machine learning based coreference systems such as BART (Versley et al., 2008) and the Illinois Coreference Package (Bengtson and Roth, 2008), among others.

Nevertheless, rule-based systems have also been applied successfully (Lappin and Leass, 1994; Mitkov, 1998; Lee et al., 2013). The authors of Lee et al. (2013) propose a coreference resolution system that is an incremental extension of the multi-pass sieve system proposed in Raghunathan et al. (2010). This system is shifting from the supervised learning setting to an unsupervised setting, and obtained the best result in the CoNLL-2011 Shared Task. It is integrated in the Stanford CoreNLP toolkit (Manning et al., 2014).

Some very interesting uses of vector space models for the coreference resolution task can be found in the literature. In Nilsson and Hjelm (2009) the authors investigate the effect of using vector space models as an approximation of the kind of lexico-semantic and common-sense knowledge needed for coreference resolution for Swedish texts. They also work with reduced dimensional vector spaces and obtain encouraging results. In an attempt to increase the performance of a coreference resolution engine, structured semantic knowledge available in the web is used in Bryl et al. (2010). One of the strategies they adopt is to apply the SVD to Wikipedia articles and classify mentions in a reduced dimensional vector space.

3. Proposed approach

The approach presented in this paper consists of a multi-classifier system which classifies mention-pairs in a reduced dimensional vector space. This multi-classifier is composed of several k-Nearest Neighbors (k-NN) classifiers. A set of linguistic features is used to generate the vector representations for the mention-pairs. The training dataset is used to create a reduced dimensional vector space using the SVD technique. Mention-pairs in the training, development and testing sets are represented using the same linguistic features and projected onto the reduced dimensional space.

The classification process is performed in the reduced dimensional space. To create the multi-classifier, random subsampling is applied and $TD_1, \ldots, TD_i$ training datasets are obtained for the reduced dimensional space. Given a testing case q, the k-NN classifiers make label predictions $c_1, \ldots, c_i$ based on the training datasets $TD_1, \ldots, TD_i$. These predictions are combined to obtain the final prediction c using a Bayesian voting scheme and based on the confidence values computed. It is a binary classification system where the final prediction c may be positive (mentions tested corefer) or negative (mentions do not corefer). Fig. 1 shows the fundamental steps of the experiment.

In the rest of this section, details about the SVD dimensionality reduction technique, the k-NN classification algorithm, the combination of classifiers and the evaluation measures used are briefly reviewed.

3.1. The SVD dimensionality reduction technique

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization and Information Retrieval tasks. Latent Semantic Indexing (LSI)[1] is a variant of the VSM in which documents are represented in a lower dimensional vector space created from a training dataset (Deerwester et al., 1990). To create such a lower dimensional vector space, LSI generates a term-document matrix M and computes its Singular Value Decomposition (SVD) matrix decomposition, $M = U\Sigma V^T$. As a result, r singular values are obtained, and terms and documents are mapped to the r-dimensional vector space. By reducing the r to p, a reduced dimensional space is created, the p-dimensional space onto which vectors are projected. This reduced dimensional space is used for classification purposes, and the cosine similarity is usually used to measure the similarity between vectors (Berry et al., 1995).

It has been proved that computing the similarity of vectors in the reduced dimensional space gives better results than working in the original space. In fact, LSI is said to be able to capture the latent relationships among words in documents thanks to the word co-occurrence analysis performed by the SVD technique, and therefore, cluster semantically terms and documents. This powerful technique is being used to better capture the semantics of texts in applications such as Information Retrieval (Berry and Browne, 2005). LSI is referred to as Latent Semantic Analysis (LSA) when it is used as a model of the acquisition, induction and representation of language and the focus is on the analysis of texts (Dumais, 2004).

For the sake of the coreference resolution task, each document corresponds to a mention-pair, and words in each document are the linguistic feature values for the associated mention-pair. Matrix M is constructed for the selected feature values (terms) and all mention-pairs considered (documents) in the training dataset. The SVD decomposition is computed and the p-dimensional reduced space is created. In the approach presented U is used as the reduced dimensional representation, and the coordinates are computed to project mention-pair vectors onto the reduced space and compare them.

3.2. The k-NN classification algorithm

The k-Nearest Neighbors algorithm (k-NN) is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training cases, and uses the class of the k top-ranking neighbors to do the prediction for the testing case being analyzed (Dasarathy, 1991; Aha et al., 1991).

Parameter k is set to 3 in the approach presented, based on our previous experiments (Zelaia et al., 2005). Given a testing mention-pair vector q, the 3-NN classifier is used to find the three nearest neighbor mention-pair vectors in the reduced dimensional vector space. The cosine is used to measure vector similarity and find the nearest.

In this paper, the k-NN classifier provided with the Weka package (Hall et al., 2009) is also used. Results obtained with it are considered a baseline and make it possible to provide a honest comparison to the ones obtained with the proposed approach.

3.3. Multi-classifier systems

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training datasets TD_i is generated by selecting n training cases drawn randomly with replacement from the original training dataset TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In the approach presented in this paper, parameter n_1 is set to be 60% of the total number of training cases n, based on some previous experiments carried out for this task. The proportion of positive and negative cases in the training dataset TD is preserved in the different TD_i datasets generated.

Given a testing case q, the multi-classifier makes label predictions $c_1, \ldots, c_i$ based on each one of the training datasets $TD_1, \ldots, TD_i$. These label predictions may be either positive $(+)$ or negative $(-)$. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value is calculated for each training dataset TD_j, $j = 1, \ldots, i$ and label to be predicted $(c = +, c = -)$: $cv^j_{(+)}$, $cv^j_{(-)}$. These confidence values are calculated based on the training collection. Confidence values are summed by label; the label c that gets the highest value is finally proposed as a prediction for the testing case q.

3.4. Evaluation measures

The approach presented in this paper is a binary classification system where the final prediction c may be positive (mentions tested corefer) or negative (mentions do not corefer). There are many metrics that can be used to measure the performance of a classifier. In binary classification problems precision and recall are very widely used. Precision (Prec) is the number of correct positive results divided by the number of all positive results, and recall (Rec) is the number of correct positive results divided by the number of positive results that should have been returned.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines them. The F_1-score can be interpreted as a weighted average of precision and recall:

$$F_1 = \frac{2 \cdot \mathrm{Prec} \cdot \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}$$

Accuracy is also used as a statistical measure of performance in binary classification tasks. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases tested.

4. Case study

This section briefly reviews the dataset used in the experiments and the preprocessing applied.

4.1. Dataset

The OntoNotes v4.0 Release Corpus is used in the experiments.[2] It provides a large-scale multi-genre corpus with multiple layers of annotation (syntactic, semantic and discourse information) which also include coreference tags. A nice description of the coreference annotation in OntoNotes can be found in Pradhan et al. (2007a) and Pradhan et al. (2007b).

Although OntoNotes is a multi-lingual resource for English, Chinese and Arabic, for the scope of this paper, just the English texts for five different genres or types of sources are used:

[1] http://lsi.research.telcordia.com,http://www.cs.utk.edu/~lsi.

[2] Downloaded from Linguistic Data Consortium (LDC) Catalog No.: LDC2011T03, https://catalog.ldc.upenn.edu/LDC2011T03. For more information, see OntoNotesRelease4.0.pdf and coreference/englishcoref.pdf files in LDC directory.

```
#begin document (bn/abc/00/abc_0011); part 000
bn/abc/00/abc_0011 0  0 President NNP  (TOP(S(NP* - - - -   * (ARG1* (ARG0*   (0
bn/abc/00/abc_0011 0  1 Clinton   NNP  *)         - - - -   (PERSON) *) *)     0)
bn/abc/00/abc_0011 0  2 is        VBZ  (VP*       be 01 1 - * (V*)     *     -
bn/abc/00/abc_0011 0  3 on        IN   (PP*       - - - -   * (ARG2*   *     -
bn/abc/00/abc_0011 0  4 his       PRP (NP(NP*    - - - -   * *        *      (0)
bn/abc/00/abc_0011 0  5 way       NN    *)        - - 4 -   * *        *     -
bn/abc/00/abc_0011 0  6 to        IN   (PP*       - - - -   * *        *     -
bn/abc/00/abc_0011 0  7 Egypt     NNP  (NP*))))   - - - -   (GPE)  *)  *     -
bn/abc/00/abc_0011 0  8 to        TO   (S(VP*     - - - -   * (ARGM-PRP*  *  -
bn/abc/00/abc_0011 0  9 attend    VB   (VP*       attend 01 1 - * * (V*)     -
bn/abc/00/abc_0011 0 10 tomorrow  NN   (NP(NP*    - - - -   (DATE) * (ARG1*   (1
bn/abc/00/abc_0011 0 11 's        POS   *)        - - - -   * *        *      1)
bn/abc/00/abc_0011 0 12 emergency NN    *         - - 1 -   * *        *     -
bn/abc/00/abc_0011 0 13 summit    NN    *)))))    - - 3 -   * *)       *)    -
bn/abc/00/abc_0011 0 14 .         .     *))       - - - -   * *        *     -
bn/abc/00/abc_0011 0  0 Tensions  NNS   (TOP(S(S(NP(NP*)- - - - * (ARG1* * * -
bn/abc/00/abc_0011 0  1 in        IN    (PP*      - - - -   * * *      *     -
bn/abc/00/abc_0011 0  2 the       DT    (NP*      - - - -  (LOC* * *   *     -
bn/abc/00/abc_0011 0  3 Middle    NNP    *        - - - -   *    * *   *     -
bn/abc/00/abc_0011 0  4 East      NNP    *)))     - - - -   *) *)  *   *     -
bn/abc/00/abc_0011 0  5 remain    VBP   (VP*      remain 01 1 - * (V*) * *   -
bn/abc/00/abc_0011 0  6 very      RB    (ADJP*    - - - -   * (ARG3* * *     -
bn/abc/00/abc_0011 0  7 high      JJ     *)       - - - -   * *)     * *     -
bn/abc/00/abc_0011 0  8 after     IN    (PP*      - - - -   * (ARGM-TMP* * * -
bn/abc/00/abc_0011 0  9 two       CD    (NP(NP*   - - - -   (DATE* * * *     -
bn/abc/00/abc_0011 0 10 weeks     NNS    *)       - - - -   *) *   * *       -
bn/abc/00/abc_0011 0 11 of        IN    (PP*      - - - -   *  *   * *       -
bn/abc/00/abc_0011 0 12 violence  NN    (NP*))))))- - - -   *  *)  * *       -
bn/abc/00/abc_0011 0 13 and       CC     *        - - - -   *  *   * *       -
bn/abc/00/abc_0011 0 14 the       DT    (S(NP(NP* - - - -   *  * (ARG1* *    -
bn/abc/00/abc_0011 0 15 immediate JJ     *        - - - -   *  *   *    *    -
bn/abc/00/abc_0011 0 16 goal      NN     *)       - - 1 -   *  *   *    *    -
bn/abc/00/abc_0011 0 17 tomorrow  NN    (NP*))    - - - -  (DATE) * *)  *     (1)
bn/abc/00/abc_0011 0 18 is        VBZ   (VP*      be 01 1 - *  * (V*)   *    -
bn/abc/00/abc_0011 0 19 to        TO    (S(VP*    - - - -   *  * (ARG2* *    -
bn/abc/00/abc_0011 0 20 stop      VB    (VP*      stop 01 2 - * * *  (V*)    -
bn/abc/00/abc_0011 0 21 the       DT    (NP*      - - - -   * * * (ARG1*     -
bn/abc/00/abc_0011 0 22 killing   NN     *))))))  - - 1 -   * * *)     *)    -
bn/abc/00/abc_0011 0 23 .          .     *))      - - - -   * * *      *     -
#end document
```

Fig. 2. An example of *_conll file. There are four coreference mentions: m_1 = *President Clinton*, m_2 = *his*, m_3 = *tomorrow's*, m_4 = *tomorrow* and two coreference chains: {*President Clinton*, *his*} and {*tomorrow's*, *tomorrow*}.

broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) and web data (WB).

The English language portion of the OntoNotes v4.0 Release Corpus was used in the CONLL-2011 coreference resolution Shared task.[3] The task is to automatically identify mentions of entities and events in text and to link the corefering mentions together to form mention chains (Pradhan et al., 2011, 2012). Since OntoNotes coreference data spans multiple genre, the task organizers created a testing set spanning all the genres. The training, development and testing files are downloaded from the CONLL-2011 website. In this work, hand-annotated gold files are used for the experiments. The *_conll files contain information in a tabular structure where the last column contains coreference chain information. The example of Fig. 2 shows a *_conll file with four coreference mentions annotated: Mentions m_1 = *President Clinton* and m_2 = *his* are coreferent and therefore have the same label (0) in the last column, and mentions m_3 = *tomorrow's* and m_4 = *tomorrow*, which are also coreferent, have label (1). These four mentions form two coreference chains: {*President Clinton*, *his*} and {*tomorrow's*, *tomorrow*}.

4.2. Preprocessing

In order to obtain the vector representation for each pair of mentions, the features defined in Sapena et al. (2011) and Sapena et al. (2013) are used. The authors of the cited papers developed a coreference resolution system called RelaxCor[4] and participated in

[3] http://conll.cemantix.org/2011/introduction.html.

[4] http://nlp.lsi.upc.edu/relaxcor/.

the CoNLL-2011 shared task obtaining very good results. It is an open source software available for anyone who wishes to use it. Results computed for these original feature vectors are used as a baseline for the proposed approach.

The 127 binary features used contain morphosyntactic and lexicosemantic information. These features are related to the distance between the two mentions (in the same sentence, in consecutive sentences, is the first mention, etc.), lexical information (string matching of mentions, both are pronouns and their strings match, etc.), morphological information (the number of both mentions matches, the gender of both mentions matches, etc.), syntactic dependencies (one mention is included in the other, etc.) and semantic information (the same semantic role, one mention is an alias of the other, etc.). Using the coreference information given in the *_conll files, mention-pairs are generated, their corresponding feature vector is created and a label is assigned to each of them: a positive label (+) indicates that the two mentions corefer, whereas a negative label (−) indicates that they do not corefer. According to the example of Fig. 2, there are two positive mention-pairs: m_1–m_2 and m_3–m_4. There are four more possible mention-pairs, all of which are negative: m_1–m_3, m_1–m_4, m_2–m_3 and m_2–m_4.

Note that each mention in a file is combined with all the rest of mentions in the same file to form mention-pairs. Pairing the four mentions in Fig. 2, for example, six mention-pairs can be generated, only two of which are positive. Consequently, a very large amount of negative instances is generated, specially for large files. In order to reduce the amount of negative instances in a similar manner as in Sapena et al. (2011), negative instances with more than five feature values different from any positive instance in each file are eliminated. Bringing together the instances generated from files of the same split and genre, the training, development and testing corpora for the 5 genres are created. Contradictions (negative instances with identical feature values as a positive instance) and instances that appear more than once in the same corpus are removed. Since the size of the corpora generated was too large for some of the genres, a stratified random sampling strategy is applied to reduce the size of all corpora; the broadcast conversations (BC) genre training corpus, for example, had more than 4 million instances before the size reduction strategy was applied. Table 1. gives detailed information about the number of positive and negative mention-pairs in the training, development and testing corpora used in the experiments.

Applying Latent Semantic Indexing (LSI) a term-document matrix is constructed for each of the five training corpora. Documents represent mention-pairs and each of them consists of 127 words (linguistic feature values) out of the 254 possible ones. The ones selected by LSI are assigned a row in the term-document matrix. Feature values found in each corpus are indexed and counted in order to compute a table of documents and words. Only feature values that appear above an established frequency threshold in the training corpora are selected as terms. A matrix that reflects whether each term appears in each document is created. Note that this matrix is binary.

Table 2 shows the number of terms (selected feature values) and documents (positive and negative mention-pairs) found in the training corpora for each genre. The third row in the table shows the number of singular values (dimensions) computed by SVD for each of the genres. These values are quite similar for the five genres, ranging from 83 to 87. By means of these SVD-dimensions the SVD-dimensional vector representation for the documents (mention-pairs) is obtained.

5. Experimental setup

To optimize the behavior of the proposed approach, the five development corpora are used to adjust two parameters in a parameter tuning phase. The two parameters optimized are the dimension of the vector space and the number of classifiers for the multi-classifier system.

- *The dimension of the vector space*: the reduction of the SVD-dimension is analyzed to see if results improve by means of a reduced dimensional representation for mention-pairs. The following values are experimented: 5, 10, 15, 20, 25, 30, 40.
- *The number of classifiers*: To optimize the behavior of the multi-classifier system, the number of training datasets is adjusted. The following values are experimented: 5, 10, 20, 30, 40, 50, 60, 70, 80.

The five genres correspond to texts coming from different sources and may have very different characteristics (Uryupina and Poesio, 2012). That is why they are treated as five different classification problems and therefore, the parameter optimization process is performed in an independent way for each of the genres.

Tables 3–7 show the results for the different values of the two parameters using the development corpora. Rows in the tables correspond to values for the reduced dimension, and columns correspond to the number of classifiers. Two evaluation measures are computed for each parameter-pair; results in the first row are accuracy rates; the ones in the second, F_1-scores. The highest value in each row is shown in bold, and the highest F_1-score in each table is shown in a box.

The optimal values for parameters are determined by the highest F_1-score in each table (the values in a box) and are summarized in Table 8. According to these optimal values, testing mention-pair vectors for the BC genre, for example, are projected onto the 30-dimensional vector space and classified by a multi-classifier formed by 60 *k*-NN classifiers. This implies that 60 training datasets (TD_i) have to be sampled in the reduced 30 dimensional space and each one is used to obtain a classification for a given testing mention-pair using the *k*-NN classifier.

Table 1
Size of corpora used in the experiments.

	BC	BN	MZ	NW	WB
Training (+)	20,206	44,515	25,103	31,034	24,501
Training (−)	26,623	55,921	23,568	50,687	26,948
Development (+)	4,056	5,920	3,873	4,776	3,531
Development (−)	5,831	8,609	4,864	7,615	5,732
Testing (+)	29,363	10,771	3,918	15,857	17,146
Testing (−)	16,591	12,480	3,209	15,759	5,505

Table 2
Terms, documents and singular values (SVD-dimensions) for the five training corpora.

	BC	BN	MZ	NW	WB
Terms (selected feature values)	227	230	227	229	230
Documents (mention-pairs)	46,829	100,436	48,671	81,721	51,449
Singular Values (SVD-dimensions)	83	86	85	86	87

6. Experimental results

In order to evaluate the impact of LSI in this task, some experiments are carried out in the testing phase.

- *Baseline*: To compute a baseline for the proposed approach, the classification of testing mention-pairs represented by the

Table 3
Parameter tuning for the BC genre. Accuracy and F_1-score.

BC genre		Number of TD_i training datasets							
Dimension		5	10	20	30	40	50	60	70
10	Acc.	67.40	**67.99**	67.68	67.86	67.69	67.80	67.96	67.87
	F_1	57.99	57.75	57.88	**58.42**	58.18	58.39	58.37	58.41
15	Acc.	65.26	**66.61**	66.19	66.43	66.22	66.30	66.38	66.39
	F_1	58.91	58.79	59.39	59.43	59.44	59.66	59.64	**59.73**
20	Acc.	65.51	66.67	**66.77**	66.50	66.75	66.07	66.02	66.20
	F_1	59.97	60.23	60.67	60.33	**60.75**	60.30	60.18	60.22
25	Acc.	64.86	65.85	66.06	66.15	66.17	66.16	**66.22**	66.15
	F_1	60.15	60.12	60.40	60.63	60.85	60.78	60.84	**60.89**
30	Acc.	64.68	66.25	65.93	66.13	65.85	65.98	**66.44**	66.18
	F_1	60.60	61.10	61.49	61.49	61.50	61.66	**61.85**	61.71
40	Acc.	65.75	64.50	65.52	**66.11**	65.69	65.82	65.65	65.70
	F_1	61.31	61.06	61.16	61.40	61.17	**61.47**	61.17	61.24

Table 4
Parameter tuning for the BN genre. Accuracy and F_1-score.

BN genre		Number of TD_i training datasets					
Dimension		5	10	20	30	40	50
10	Acc.	70.13	70.90	**71.42**	71.30	70.96	71.12
	F_1	65.42	65.25	66.08	**66.32**	66.07	66.05
15	Acc.	68.08	**69.75**	69.59	69.65	69.50	69.44
	F_1	64.01	64.32	64.61	**64.88**	64.66	64.82
20	Acc.	70.55	**72.15**	71.85	71.84	71.69	71.82
	F_1	66.14	66.63	66.85	67.01	66.96	**67.17**
25	Acc.	70.70	71.91	71.84	**72.16**	71.98	71.98
	F_1	65.97	66.33	66.68	**67.36**	67.09	67.21
30	Acc.	69.99	**71.98**	71.93	71.91	71.80	71.96
	F_1	65.40	66.25	66.69	66.92	66.75	**66.97**
40	Acc.	70.58	72.13	72.09	71.81	**72.20**	71.99
	F_1	65.56	66.21	66.51	66.33	**66.95**	66.68

Table 5
Parameter tuning for the MZ genre. Accuracy and F_1-score.

MZ genre		Number of TD_i training datasets						
Dimension		5	10	20	30	40	50	60
10	Acc.	63.84	63.29	63.33	64.40	64.11	64.36	**64.53**
	F_1	65.59	65.96	65.66	66.34	66.11	**66.36**	66.30
15	Acc.	65.29	65.45	65.92	66.25	**66.26**	66.17	66.20
	F_1	66.52	67.13	67.31	**67.43**	67.41	67.30	67.24
20	Acc.	64.71	65.27	66.05	65.93	65.93	**66.41**	66.00
	F_1	66.57	67.34	67.87	67.73	67.63	**68.10**	67.66
25	Acc.	64.94	65.10	65.03	65.99	65.68	65.86	**66.06**
	F_1	66.86	67.53	67.21	67.78	67.56	67.57	**67.81**
30	Acc.	65.19	64.47	64.96	65.07	65.35	**65.51**	65.51
	F_1	67.02	67.11	67.31	67.31	67.30	**67.53**	67.48
40	Acc.	65.22	64.89	64.65	65.25	**65.46**	65.37	65.21
	F_1	67.34	67.42	66.98	**67.52**	**67.52**	67.36	67.36

original 127 binary features is considered. They are also used by RelaxCor, the existing most similar method to the proposed approach. Mention-pairs are classified using a single 3-NN classifier.

- *Single classification*: In a second experiment, some very widely used standard classification algorithms such as Naive Bayes (NB), classification trees (C4.5), Support Vector Machines (SVM) and *k*-nearest neighbors are used to classify mention-pairs represented in the SVD-dimensional vector space created by LSI (see the SVD-dimensions used for the five genres in Table 2).
- *Proposed approach*: In a third experiment the proposed approach is used. First, a multi-classifier system composed of several 3-NN classifiers classifies testing mention-pairs in the same SVD-dimensional vector space as in the previous experiment (MultiCl.$_{opt}$ + SVD). This multi-classifier is generated according to the optimal number of classifiers for each genre (see Table 8). Finally, the same multi-classifier is applied for the optimal SVD-dimensions per genre (MultiCl.$_{opt}$ + SVD$_{opt}$) (see optimal number of classifiers and optimal SVD-dimensions in Table 8).

Table 6
Parameter tuning for the NW genre. Accuracy and F_1-score.

NW genre		Number of TD_i training datasets				
Dimension		5	10	20	30	40
10	Acc.	77.64	77.94	**78.39**	78.05	78.33
	F_1	69.58	69.17	69.85	69.62	**69.99**
15	Acc.	77.12	78.15	**78.44**	78.34	78.21
	F_1	69.21	69.62	**70.19**	**70.19**	70.18
20	Acc.	76.95	78.37	78.15	78.29	**78.38**
	F_1	69.27	70.01	69.87	70.20	**70.41**
25	Acc.	77.15	78.41	**78.73**	78.54	78.52
	F_1	69.48	70.02	**70.85**	70.73	70.69
30	Acc.	76.41	77.99	77.75	**78.16**	78.07
	F_1	68.44	69.09	69.24	**69.91**	69.82
40	Acc.	77.33	**78.58**	78.35	78.42	78.52
	F_1	69.80	70.26	70.29	70.56	**70.60**

Table 7
Parameter tuning for the WB genre. Accuracy and F_1-score.

WB genre		Number of TD_i training datasets					
Dimension		5	10	20	30	40	50
5	Acc.	66.53	67.73	67.77	**67.90**	67.64	67.46
	F_1	62.02	62.48	62.41	**62.64**	62.47	62.32
10	Acc.	67.51	**67.91**	67.62	67.75	67.79	67.47
	F_1	65.22	65.05	65.03	65.45	**65.49**	65.06
15	Acc.	65.31	**67.61**	67.13	66.96	67.11	67.30
	F_1	62.87	64.14	64.11	64.05	64.36	**64.50**
20	Acc.	66.03	**67.39**	67.24	66.91	66.92	67.12
	F_1	63.18	63.73	**64.27**	64.00	64.03	**64.27**
25	Acc.	65.21	66.65	**66.69**	66.51	66.63	66.37
	F_1	62.70	63.21	63.71	63.57	**63.89**	63.66
30	Acc.	64.92	**66.55**	65.88	66.04	66.14	65.86
	F_1	62.33	62.80	62.88	63.04	**63.25**	63.09
40	Acc.	65.25	**66.54**	66.30	66.06	66.13	66.32
	F_1	62.94	63.45	63.43	63.45	63.78	**63.92**

Table 8
Optimal dimension and number of classifiers.

Optimal parameters	BC	BN	MZ	NW	WB
Optimal dimension	30	25	20	25	10
Optimal number of classifiers	60	30	50	20	40

Table 9 shows the results obtained in each of the experiments. The results shown in bold in the columns that correspond to the five genres are the best accuracy and F_1-score for each genre. Note

Table 9
Testing results for the five genres. Last column: mean accuracy and F_1-scores.

Experiment		BC	BN	MZ	NW	WB	Mean
Baseline	Acc.	**71.90**	70.40	**70.60**	70.70	66.90	70.10
(RelaxCor)	F_1	**76.20**	68.60	**73.10**	67.90	74.40	72.00
Single classification	Acc.	42.21	61.90	61.95	66.76	41.85	54.93
(NB + SVD)	F_1	34.90	35.90	58.30	57.00	43.00	45.82
Single classification	Acc.	64.86	69.76	65.32	70.82	69.67	68.09
(C4.5 + SVD)	F_1	72.70	64.70	71.10	69.70	78.10	71.26
Single classification	Acc.	63.62	51.98	68.75	68.62	70.92	64.78
(SVM + SVD)	F_1	68.60	3.80	71.10	59.90	79.70	56.62
Single classification	Acc.	67.20	72.50	66.20	72.50	**78.30**	71.30
(3-NN + SVD)	F_1	74.20	71.00	71.70	71.50	**85.00**	**74.70**
Proposed approach	Acc.	66.90	**75.50**	66.10	**74.20**	77.60	**72.10**
MultiCl$_{opt}$+ SVD	F_1	73.90	**72.80**	70.70	**71.60**	84.10	74.60
Proposed approach	Acc.	66.30	74.30	68.50	71.20	76.20	71.30
MultiCl$_{opt}$+SVD$_{opt}$	F_1	72.40	71.40	71.50	67.80	83.10	73.20

that the two performance measures computed are very correlated in the five cases. Taking into account that the proportion of positive and negative instances varies from genre to genre, this correlation gives consistency to the interpretation of the results obtained.

The best results for BC and MZ genres are obtained by the Baseline, applying a single 3-NN classifier to original RelaxCor vectors (F_1-scores: 76.2 and 73.1, respectively). For the rest of the genres, the best results are obtained when the SVD-dimensional representation is used for mention-pairs. An F_1-score of 85 is obtained for the WB genre with a single 3-NN classifier (3-NN + SVD). The proposed approach achieves the best results for two out of the five genres, with an F_1-score of 72.8 for BN and 71.6 for NW, when the SVD-dimensional vectors are classified by the optimized multi-classifier (MultiCl$_{opt}$+ SVD). Surprisingly, when the optimized dimensions are used, results do not improve (MultiCl$_{opt}$+SVD$_{opt}$).

The last column in Table 9 shows the mean accuracy and F_1-scores obtained in each experiment, taking into account the five genres as a whole (the best are shown in bold). The best mean F_1-score is obtained when mention-pairs are classified in the SVD-dimensional vector space by a single 3-NN classifier. In fact, this result is very closely followed by the one obtained in the proposed approach with the multi-classifier, (74.7 and 74.6, respectively). The best mean accuracy is obtained by the proposed approach (72.1). This good results seem to suggest that the dimensions computed by the SVD technique are very appropriate to represent mention-pairs and classify them. Moreover, the use of the multi-classifier system gets to achieve even better results for some of the genres, outperforming the ones obtained by the other classification systems.

7. Conclusions and future work

In this paper a different machine learning approach to deal with the coreference resolution task is presented: a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space created by applying the SVD technique. The approach is tested for OntoNotes, the corpus used in the most recent international challenges such as CONLL-2011 and CONLL-2012, devoted to evaluate coreference resolution systems.

A parameter tuning phase is performed to adjust the dimension of the vector space and the number of classifiers. This optimization process is carried out in an independent way for each genre. Results show different behaviors for the five genres and, therefore, make it difficult to find a general solution and treat the five genres as a unique classification problem.

Three experiments are carried out. In a first experiment, the most similar method to the proposed approach is considered, and results are computed using the original feature vectors and a single 3-NN classifier to set a baseline. A second experiment is performed to measure to what extent working with feature vectors in a reduced dimensional vector space helps to determine coreference resolution of mention pairs. Four single classifiers are applied, being 3-NN the one that obtains the best results. In fact, it outperforms baseline results for three out of the five genres (BN, NW, WB) and is the best for WB genre. In a final experiment, the proposed approach is applied and very promising results are obtained. As a matter of fact, the best results are obtained using it for BN and NW genres.

When mean results per experiment are considered, the SVD-dimensional representation always achieves the best results. This is a very significant fact, because it seems to suggest that the SVD-dimensional representation computed by LSI is a very robust and suitable representation for coreference mention-pairs. The use of such a representation, compared to existing approaches that do not make use of it, may benefit the performance of systems that solve the complete task of mention detection and coreference resolution and, consequently, have an important impact in more general Natural Language Processing tasks that require natural language understanding.

As future work, we plan to experiment with OntoNotes v5.0 Release, the new version available. We also intend to experiment with some other kind of multi-classifer systems. It is important to note that the approach may be applied to corpora in other languages as well.

Acknowledgments

This work was supported by the University of the Basque Country, UPV/EHU, Ikerketaren arloko Errektoreordetza/Vicerrectorado de Investigación.

References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach. Learn. 6 (1), 37–66.
Bengtson, E., Roth, D., 2008. Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, Association for Computational Linguistics, pp. 294–303.
Berry, M.W., Browne, M., 2005. Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM, Philadelphia, USA.
Berry, M.W., Dumais, S.T., O'Brien, G.W., 1995. Using linear algebra for intelligent information retrieval. SIAM Rev. 37 (4), 573–595.
Björkelund, A., Nugues, P., 2011. Exploring lexicalized features for coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 45–50.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.
Broscheit, S., Poesio, M., Ponzetto, S.P., Rodriguez, K.J., Romano, L., Uryupina, O., Versley, Y., Zanoli, R., 2010. Bart: a multilingual anaphora resolution system. In: Proceedings of the SemEval-2010, pp. 104–107.
Bryl, V., Giuliano, C., Serafini, L., Tymoshenko, K., 2010. Using background knowledge to support coreference resolution. In: ECAI, Frontiers in Artificial Intelligence and Applications, vol. 215. IOS Press, Amsterdam, The Netherlands, pp. 759–764.
Chang, K.-W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., Roth, D., 2011. Inference protocols for coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 40–44.
Dasarathy, B., 1991. Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA.
Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R., 1990. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41 (6), 391–407.

Dietterich, T., 1998. Machine learning research: four current directions. AI Mag. 18 (4), 97–136.
Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R., 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004), European Language Resources Association (ELRA), Lisbon, Portugal.
Dumais, S., 2004. Latent semantic analysis. In: ARIST (Annual Review of Information Science Technology), vol. 38. pp. 189–230.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. SIGKDD Explor. Newsl. 11 (1), 10–18.
Hirschman, L., Chinchor, N., 1998. Coreference task definition. In: Proceedings of the Seventh Message Understanding Conference, MUC-7.
Ho, T., Hull, J., Srihari, S., 1994. Decision combination in multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. 16 (1), 66–75.
Kobdani, H., Schütze, H., 2010. Sucre: a modular system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp. 92–95.
Lappin, S., Leass, H.J., 1994. An algorithm for pronominal anaphora resolution. Comput. Linguist. 20 (4), 535–561.
Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D., 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. Comput. Linguist. 39 (4), 885–916.
Manning, C., Bauer, J., Surdeanu, M., Finkel, J., Bethard, S., McClosky, D., 2014. The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60.
Mitkov, R., 1998. Robust pronoun resolution with limited knowledge. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol. 2, pp. 869–875.
Mitkov, R., 2002. Anaphora Resolution, Studies in Language and Linguistics. Longman, Great Britain.
MUC6, 1995. Coreference task definition. In: Proceedings of the 6th Conference on Message Understanding, MUC-6, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 335–344.
Ng, V., 2010. Supervised noun phrase coreference research: the first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1396–1411.
Nilsson, K., Hjelm, H., 2009. Using semantic features derived from word-space models for swedish coreference resolution. In: Proceedings of the 17th Nordic Conference of Computational Linguistics NODALIDA 2009, vol. 4, Northern European Association for Language Technology (NEALT), Stockholm, Sweden, pp. 134–141.
Nogueira dos Santos, C., Lopes Carvalho, D., 2011. Rule and tree ensembles for unrestricted coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, Portland, Oregon, USA, pp. 51–55.
Orasan, C., Cristea, D., Mitkov, R., Branco, A., 2008. Anaphora resolution exercise: an overview. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco.
Pradhan, S.S., Hovy, E.H., Marcus, M.P., Palmer, M., Ramshaw, L.A., Weischedel, R.M., 2007a. Ontonotes: a unified relational semantic representation. Int. J. Semant. Comput. 1 (4), 405–419.
Pradhan, S.S., Ramshaw, L., Weischedel, R.M., MacBride, J., Micciulla, L., 2007b. Unrestricted coreference: identifying entities and events in ontonotes. In: ICSC, IEEE Computer Society, pp. 446–453.
Pradhan, S., Palmer, M., Ramshaw, L., Weischedel, R., Marcus, M., Xue, N., 2011. Conll-2011 shared task: modeling unrestricted coreference in ontonotes. Proceedings of the Fifteenth Conference on Computational Natural Language Learning.
Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y., 2012. CoNLL-2012 shared task: modeling multilingual unrestricted coreference in OntoNotes. In: Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012).
Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C., 2010. A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 492–501.
Recasens, M., Màrquez, L., Sapena, E., Martí, M.A., Taulé, M., Hoste, V., Poesio, M., Versley, Y., 2010. Semeval-2010 task 1: coreference resolution in multiple languages. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–8.
Sapena, E., Padró, L., Turmo, J., 2011. Relaxcor participation in conll shared task on coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CONLL, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 35–39.
Sapena, E., Padró, L., Turmo, J., 2013. A constraint-based hypergraph partitioning approach to coreference resolution. Comput. Linguist. 39 (4), 847–884.
Soon, W.M., Ng, H.T., Lim, D.C.Y., 2001. A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. 27 (4), 521–544.
Uryupina, O., 2010. Corry: a system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp. 100–103.
Uryupina, O., Poesio, M., 2012. Domain-specific vs. uniform modeling for coreference resolution. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), European Language Resources Association (ELRA), Istanbul, Turkey, pp. 187–191.
Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A., 2008. Bart: a modular toolkit for coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, HLT, Stroudsburg, PA, USA, pp. 9–12.
Zelaia, A., Alegria, I., Arregi, O., Sierra, B., 2005. Analyzing the effect of dimensionality reduction in document categorization for basque. Arch. Control Sci. 15 (4), 703–710.

ONDORIOAK ETA ETORKIZUNERAKO LANA

VII. KAPITULUA

Ondorioak eta etorkizunerako lana

Ikerlan honetan LSI tresnaren eta SVD bidezko dimentsio-murrizketaren bidez kalkulatutako adierazpen matematikoan oinarritzen den sailkatze-sistema bat aurkeztu da. Sistema horrek Hizkuntzaren Prozesamenduaren hainbat atazaren ebazpenean ekar dezaken onura aztertu da. Azken kapitulu honetan, egindako lanaren analisitik ateratako ondorio nagusiak, egindako ekarpenak eta sortu diren etorkizunerako ikerlerroak zerrendatzen dira.

VII.1 Ekarpen nagusiak eta ondorioak

Aurkeztutako metodologiari buruz, hiru dira azpimarratzeko puntuak: (1) instantzietarako erabilitako adierazpena, (2) diseinatutako eta inplementatutako Bagging bidezko multi-sailkatzailea eta (3) mota desberdineko sailkatze-problemak ebatzi izana, horretarako beharrezkoak izan diren egokitzapenak inplementatuz: sailkatze-problema bitarrak eta klase anitzekoak, etiketa bakarrekoak eta etiketa anitzak.

Bestalde, metodologia Hizkuntzaren Prozesamenduaren hiru ataza desberdinetan probatu da, haren portaera aztertzeko asmoz. Esperimentuetan lortutako emaitzetatik metodologiaren abileziak eta ahultasunak gertutik aztertu ditugu. Ataza bakoitzari dagokion kapituluaren amaieran bildu dira lortu ditugun ondorioen xehetasunak. Dena dela, hona ekarri ditugu berriz atera ditugun ondorio nagusiak:

- Testu-Sailkatzea. (1) Aurkeztutako metodologia oso aproposa izan da atazarako. Bagging bidez sortutako multi-sailkatzaileak oinarrizko sailkatzaile bakarrarekin lortutako emaitzak hobetzen ditu. (2) LSI/SVD

bidezko dimentsio-murrizketa egiteko corpusak egokiak izan dira, eta erabili den adierazpena aproposa. Euskarazko corpusarekin eta lematizazioa aplikatuz emaitza onak lortu dira. Ingelesezko Reuters-21578 corpusarekin lortutako emaitzak ere oso onak izan dira. Esan daiteke lematizatzeak alde batetik, eta dimentsioa murrizteak bestetik, dokumentuen sailkatzea hobetzen duela. (3) Ataza LSIren aplikazio-eremu naturala izanik (bag-of-words), espero zitezkeen emaitzak lortu dira.

- Hitzen Adiera-Desanbiguatzea. (1) Multi-sailkatzailea lema bakoitzaren desanbiguaziorako egokituta lortu dira emaitza egokiak, ondorioz, esan daiteke aurkeztutako metodologiak portaera ona erakutsi duela atazarako. Gainera, SemEval-2007 txapelketan gainerako parte-hartzaileek lortutako emaitzen parekoak lortu dira. (2) Ataza LSIren aplikazio-eremu naturaletik gertu dagoela esan daiteke, desanbiguatu beharreko hitzen testuinguruko beste hitzak kontuan hartu direlako (bag-of-words). (3) Hala ere, ezaugarri linguistikoek emandako informazioa ezinbestekoa dela ondorioztatu da. (4) Bukatzeko, LSI/SVD bidezko dimentsio-murrizketak ondorio zuzena eta positiboa duela emaitzetan ikusi da.

- Korreferentzia-Ebaztea. Euskarazko corpus batean Anafora Pronominalaren ebazpenarekin eta OntoNotes corpusean Korreferentzia-Ebaztearekin esperimentatu da. (1) Anafora Pronominala ebazteko egindako lanean emaitza onak lortu dira Ikasketa Automatikoko sailkatze-sistema sinpleekin, tamalez, ikerketa-lan honetan aurkezten den metodologiak ez ditu eman espero zitezkeen emaitzak. (2) Sailkatze-problema planteatzeko anafora-aurrekarigai eta aipamen-pareak antolatzeko eredua erabili da. Instantziak adierazteko ezaugarri linguistikoak besterik ez dira erabili eta lortu diren matrizeak bitarrak eta eduki eskasekoak izan dira. Ondorioz, uste dugu erabilitako ezaugarriek ez dutela behar bezala adierazten ataza honetarako behar den informazioa. Hala ere, metodologiak emaitza onak eman ditu OntoNotes corpusean, nahiz eta emaitza horiek ez hobetu dimentsio-murrizketa aplikatu denean.

  Lortutako emaitzek, ataza honen ebazpenaren inguruan hausnarketa sakona egitera bultzatu gaituzte. Korreferentzia-Ebazte atazan aipamen-pareen ereduan soilik oinarritzen diren sistemak ezbaian daude gaur egun. Atazaren izaerak berak erakusten du testuinguruari buruzko informazioa ematen duten ezaugarri linguistikoak erabiltzeaz gain, sistemari oinarrizko ezagutza (background knowledge) ematea ezinbestekoa dela. Hemen aurkeztu dugun metodologian sumatu dugun

informazio eskasia bera sumatu dute oso izaera desberdineko metodologiak erabiliz atazaren ebaztearekin lanean ari diren beste hainbat ikerlarik ere (Recasens, 2010). Oinarrizko ezagutza hori Wikipediatik lortzeko saiakerak egin dira, baina ez dira lortu esperotako emaitzak (Sapena et al., 2013).

Ikerketa-lan honetan esperimentazioan hartu ditugun erabaki batzuen inguruan zalantzak sortu izan zaizkigu. Beste batzuetan, berriz, lortutako emaitzen interpretazioa adituren batekin eztabaidatzeko gogoa piztu zaigu. Michael W. Berry ikertzaileak (berry@cs.utk.edu) lan handia egin du LSIren oinarri matematikoaren arloan, eta harekin partekatu ditugu gure zalantza batzuk argibide bila.

VII.1.1 Solasaldia Michael W. Berry ikertzailearekin

2015eko uztailean postaz harekin harremanetan jarri ginen, eta haren jarrera ezin hobea izan zen gure zalantzak argitzeko. Atal honetan jaso nahi izan ditugu harekin izandako solasaldiaren pasarte batzuk.

1.- Different strategies to select terms I've used different strategies to make LSI select more or less terms. Document frequency (minimum 1 occurrence per document), global frequency (minimum 1 occurrence per corpus), for instance. Does this make sense? A term with just one occurrence in the whole corpus means a row in the matrix, full of 0 values, and just a 1. I see some of my colleagues use this strategy to force the selection of more terms in the matrix. However, this will make the matrix be sparse, and no co-occurrences for LSI to work properly, isn't it? A minimum of 2 occurrences in the whole corpus seems a more appropriate criteria... Nevertheless, I guess that having this kind of rows in the matrix does not affect the final result, but I may be wrong...

MB: Yes, the word frequency is definitely going to control the sparsing of the term-document matrix. Allowing singletons (words occurring only once) will usually not be that helpful for building semantic relationships with other words. This more of an art form than a science, if the user expects their query term to be in the dictionary regardless of how rare it is, it is sometimes necessary to add such rare words back. For some applications, a control vocabulary (e.g., based on an ontology) will dictate how words will be acceptable as their might be external semantic networks to exploit for their meaning (e.g., WordNet).

(Ana) Yes. I agree.

2.- LSI with sparse matrices. Can we say that SVD is a technique that

helps to deal with data sparseness? After having applied log-entropy and SVD to the matrix, we obtained the worst results for the more sparse matrices in our experiments (zero values in 98% of the positions in the original matrix, 84% of zero values in another experiment, 56% in another...). In fact, co-occurrences are the clue for LSI to work properly... Highly sparse matrices don't make co-occurrences to happen very easily... No magic...

MB: Yes, "noise"is extremely helpful in the sense of LSI in that you do not want too sparse of a matrix. What the SVD essentially does in lower ranks space is to fully connect a very sparse and disconnected graph associated with the original term-document associations. You may need to experiment with how restrictive of a dictionary you can tolerate to get accurate query matching.

(Ana) Ok.

3.- Interpretation of the number of singular values. I've got a 227x46820 matrix with only 83 singular values. Rows correspond to 127 binary features (127x2=254, 227 of which were selected to be terms), and therefore, I can understand that at most 127 features can be linearly independent among them. Don't you find 83 a very low amount of singular values for this matrix? Does this mean that the information in the matrix is poor? In some other experiments, the number of singular values is equal to the minimum between the number of rows and columns, normally... Again, it is a sparse matrix...

MB: It is possible you have a rank-deficient matrix to start with and the sparsity could have played a role here again. Typically spaces of dimension (numbers of singular triplets) are on the order of 300 to 500 for many LSI applications. You might just remove those documents that had no terms and recompute the truncated SVD on them to see what the s-value distribution looks like.

(Ana) I think the problem is not related to the 46829 documents. Since I ask LSI to select terms according to the df>0, gf>0 strategy (document frequency>0, global frequency>0), I understand that, in fact, I am forcing LSI to select EVERY word in the corpus to be a term, since just one appearance in the whole corpus is enough to be a term. Therefore, all documents contain terms. Isn't it?

MB: Yes, but you may have a lot of duplicate columns which would explain a lower column rank (number of linearly indep. columns).

(Ana) I guess that the problem is in the rows... In the terms... I guess that there is linear dependency among the 227 terms selected (list of terms attached). Terms are feature values for 127 binary features. This experiment corresponds to the Coreference resolution task (article attached). Instances

(documents) are mention-pairs. Terms are feature values. It is a binary classification problem: we need to decide whether mentions corefer or they don't.

MB: Perhaps you need to aggregate some of those features and have more than 2 values (1 or 0) associated with them.

(Ana) If only 83 dimensions are found in the matrix, then there is linear dependency among features... Should we conclude that features (terms) do not represent appropriately the necessary information for mention-pairs (documents)? It is hard to arrive at this conclusion, because we use the same features as the Relaxcor system, second in the ConLL-2011 shared task (articles attached).

MB: I agree. You don't want to have so many highly correlated features. Removing some of them should help.

4.- I'm sure that understanding the underlying mathematical theory of SVD and sparse matrices may help a lot in interpreting all these information extracted from my experimental results. Most of the problems detailed previously correspond to the experiments performed for the Coreference Resolution, both using a very small corpus (Basque) and a very large one (Ontonotes). Both matrices are very sparse. It would be a valuable help to receive any feedback from you, so that we could improve the application of this technique to these and some other applications of Computational Linguistics.

MB: If you want to send the documents (say as a large text file with a blank line separately each document) I can parse it and see what type of matrices I am getting and share them back with you to evaluate.

(Ana) Thanks a lot! Well, that's more than what I expected!! However, I wouldn't like to bother you... Just in case you want to see the documents I'm using to create the matrix, I copied some files to my dropbox and shared the folder with you. You can accept it and look at the folder contents, or simply forget it. It's ok, anyway.

In any case, thank you very much. It's been a long time since I started to work with LSI, and your articles and book about "Understanding search engines"have been fundamental for my research. It's been a pleasure for me to share my doubts with you and a honor to receive feedback from you.

I wish you all the best,

MB: Ana, attached are two output files generated from your documents. I used the GenSim python library for this (see http://radimrehurek.com/gensim/). One file has the dictionary terms parsed (singletons eliminated) and the other file shows the term vectors for the first 10 dimensions of a 100-factor LSI model I generated.

(Ana) Thanks a lot, Mike!

VII.2 Etorkizunerako lana

Ikergaian aurrera egiteko bidea zabala da. Lanaren bi arlo nagusitan sakontzeko aukera ikusi dugu, batetik, erabilitako metodologia bera izan daiteke aztergai, eta bestetik, Hizkuntzaren Prozesamenduaren hainbat atal har ditzakegu aitzakia lanean jarduteko. Honako hauek dira aukera interesgarri eta bideragarrienak etorkizuneko lanari heltzeko:

- Hizkuntzaren Prozesamenduko atazekin lanean segituko dugu. Dagoeneko aztertu ditugun hiru atazetatik azkena da, Korreferentzia-Ebaztea, aurrera egiteko aukeratu dugun lehena. Egun, badago korreferentziaz etiketatutako euskarazko corpus bat, baina lanari ekin aurretik ondo aztertu beharko da problema nola planteatzea komeni den, eta LSIk nola lagun dezaken sistemari behar duen oinarrizko ezagutza ahal den neurrian emateko.

- Gainera, orain arteko jardunean, Korreferentzia-Ebazte ataza ez dugu bere osotasunean ebatzi. Bi dira jarraitu beharreko urratsak, aipamenak detektatu lehendabizi, eta aipamen bakoitza zein erreferenteri dagokion erabaki ondoren. Gure sistemak bigarren honetan hartzen du parte, baina, aztertzear dago, zer nolako eragina eduki dezakeen sistema oso batean integratuz gero. Hau da ba, epe motzean, egin nahi dugun froga. Esperimentu honetatik lortuko den emaitza, erregeletan oinarritutako sistema batekin lortutako emaitzarekin konparatzeko aukera izango dugu.

- Bestalde, ataza berriekin ere saiatu nahi dugu. Horietako batzuk LSIren aplikaziorako oso egokiak izan daitezkeela aurreikusten dugu: Informazio-Erauztea, Galdera-Erantzun sistemak edota Laburpengintza Automatikoa. Beste batzuk ordea, zailagoak izan daitezkeela uste dugu. Hauen artean, Rol Semantikoak Etiketatzea (Semantic Role Labeling) dugu. Epe ertain batean, ataza honekin hasteko asmoa dugu. Dagoeneko, bada lan bat egina arlo honetan euskarazko corpus batekin, eta aurreko kasuan bezala, lortutako emaitzak zerekin konparatu izango genuke. Ezaugarri linguistikoekin lan egitea eskatzen du atazak, eta ondorioz, ezaugarrien aurreprozesaketan ahalegin berezia egin beharko da, haien arteko mendekotasunak ekiditeko, besteak beste.

- Euskararen prozesamendu automatikoarekin segitzeaz gain, ez dugu ingelesa albo batera uzteko asmorik. Euskararako egingo diren lanen emaitzak aztertuta, horiek ingelesera eramateko bideak jorratuko dira,

eta ahal den neurrian, esperimentuak hizkuntza honetan ere egingo dira.

- Ikasketa Automatikoaren ildotik ere etorkizunean sailkatze-eredu berriekin lan egitea aurreikusten da; izan ere, erabilitako multi-sailkatzailea Bagging familikoa da, eta dagoeneko bi geruzez osatutako sailkatzaile anitzekin nahiz sailkatzaile hierarkikoekin lan egiten hasiak gara.

Hizkuntzalaritza Konputazionala asko aurreratzen ari da azken urteotan. (Hirschberg and Manning, 2015) artikuluan aipatzen denez, Ikasketa Automatikoan azken urteotan ematen ari diren aurrerapenak eta deep-learning-ak zeresan handia emango dute etorkizunean, eta bide batez, ate berriak irekiko dizkigute lanean jarraitzeko.

Bibliografia

Aduriz, I., Ceberio, K., and Díaz de Ilarraza, A. (2005). Euskarazko anafora pronominala: ikuspuntu konputazionala eta corpus baten garapena. *Gogoa: Euskal Herriko Unibersitateko hizkuntza, ezagutza, komunikazio eta ekintzari buruzko aldizkaria*, 5(1):91–116.

Aduriz, I., Ceberio, K., and Díaz de Ilarraza, A. (2006). Pronominal anaphora in basque: annotation of a real corpus. *SEPLN*, 37:99–104.

Aduriz, I., Ceberio, K., and Díaz de Ilarraza, A. (2007). Pronominal anaphora in basque: Annotation issues for later computational treatment. In *6th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC2007*, pages 1–7, Lagos (Portugal).

Aduriz, I., Ceberio, K., Díaz de Ilarraza, A., and García, I. (2008). Análisis de la correferencia para su anotación en un corpus en euskera. In *Actas del VIII Congreso de Lingüística General*, pages 496–512, Madrid.

Agirre, E. and Edmonds, P. (2007). *Word Sense Disambiguation: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition.

Arregi, O., Ceberio, K., de Ilarraza, A. D., Goenaga, I., Sierra, B., and Zelaia, A. (2010a). Determination of features for a machine learning approach to pronominal anaphora resolution in basque. *SEPLN*, 45:291–294.

Arregi, O., Ceberio, K., de Ilarraza, A. D., Goenaga, I., Sierra, B., and Zelaia, A. (2010b). *A first machine learning approach to pronominal anaphora resolution in Basque.*, volume 6433/2010, pages 234–243.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York.

Berry, M. and Browne, M. (2005). *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM, Philadelphia.

Berry, M., Drmac, Z., and Jessup, E. (1999). Matrices, vector spaces and information retrieval. *SIAM Review*, 41(2):335–362.

Berry, M. W., Dumais, S. T., and O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595.

Boser, B. E., Guyon, I. M., and Vapnik, V.Ñ. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Carlson, A., Cumby, C., Rosen, J., and Roth, D. (1999). The snow learning architecture. Number UIUCDCS-R-99-2101.

C.D.Meyer (2000). *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics.

Chen, C., Stofel, N., Post, M., Basu, C., Bassu, D., and Behrens, C. (2001). Telcordia lsi engine: Implementation and scalability issues. In Aberer, K. and Liu, L., editors, *RIDE-DM*, pages 51–58. IEEE Computer Society.

Dasarathy, B. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press.

Debole, F. and Sebastiani, F. (2005). An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and Technology*, 56(6):584–596.

Deerwester, S., Dumais, S., Furnas, G., Harshman, R., Landauer, T., Lochbaum, K., and Streeter, L. (1989). Computer information retrieval using latent semantic structure. US Patent 4,839,853.

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.

Demiröz, G. and Güvenir, H. A. (1997). Classification by voting feature intervals. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 85–92, London, UK, UK. Springer-Verlag.

Dietterich, T. G. (1998). Machine-learning research: Four current directions. *The AI Magazine*, 18(4):97–136.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program. tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal. European Language Resources Association (ELRA). ACL Anthology Identifier: L04-1011.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk.

Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Ezeiza, N., Aduriz, I., Alegria, I., Arriola, J., and Urizar, R. (1998). Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *COLING-ACL'98, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 380–384, Montreal (Canada). Association for Computational Linguistics.

Frank, E. and Witten, I. H. (1998). Reduced-error pruning with significance tests. page 98.

Furnas, G., Landauer, T., Gomez, L., and Dumais, S. (1983). Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell System Technical Journal*, 62(6):1753–1806.

Gliozzo, A., Giuliano, C., and Strapparava, C. (2005). Domain kernels for word sense disambiguation. In *In Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL-05*, pages 403–410.

Goenaga, I., Arregi, O., Ceberio, K., de Ilarraza, A. D., and Jimeno, A. (2012). Automatic coreference annotation in basque. In *11th International Workshop on Treebanks and Linguistic Theories*.

Golub, G. and Loan, C. V. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guyon, I. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. In *SIGKDD Explorations*.

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Hirschman, L. and Chinchor, N. (1998). Muc-7 coreference task definition. In *Proceedings of the Seventh Message Understanding Conference*.

Ho, T., Hull, J., and Srihari, S. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: The 90 In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hovy, E., Mitamura, T., Verdejo, F., Araki, J., and Philpot, A. (2013). *Workshop on Events: Definition, Detection, Coreference, and Representation*, chapter Events are Not Simple: Identity, Non-Identity, and Quasi-Identity, pages 21–28. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207. AAAI Press.

Kuncheva, L. (2004). Combining pattern classifiers methods and algorithms. john wiley&sons. *Inc. Publication, Hoboken*.

Landauer, T. and Dumais, S. (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

Landauer, T., Laham, D., and Foltz, P. (1998). Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Landauer, T., McNamara, D., Dennis, S., and Kintsch, W. (2013). *Handbook of Latent Semantic Analysis*. Taylor & Francis.

Lewis, D. (2004). Reuters-21578 text categorization test collection. distribution 1.0. readme file (v 1.3). http://daviddlewis.com/resources/testcollections.

Lewis, D., Yang, Y., Rose, T., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Lowe, W. (2001). Towards a theory of semantic space. In *Proceedings of the 23rd Conference of the Cognitive Science Society*, pages 576–581.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Mitkov, R. (2002). *Anaphora Resolution*. Studies in Language and Linguistics. Longman.

Ng, V. (2010). Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411.

Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M., Ageno, A., and Martí, M.A.Ñavarro, B. (2004). 3lb: Construcción de una base de datos de árboles sintáctico-semánticos para el catalán, euskera y español. *SEPLN*, 33:81–88.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, BioNLP '07, pages 97–104, Stroudsburg, PA, USA. Association for Computational Linguistics.

Platt, J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA.

Poesio, M., Kruschwitz, U., and Chamberlain, J. (2008). Anawiki: Creating anaphorically annotated resources through web cooperation. In *Proceedings of LREC'08*.

Porter, M. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 40(3):211–218.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

Pradhan, S., Palmer, M., Ramshaw, L., Weischedel, R., Marcus, M., and Xue, N. (2011). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. Proceedings of the Fifteenth Conference on Computational Natural Language Learning.

Pradhan, S. S., Hovy, E. H., Marcus, M. P., Palmer, M., Ramshaw, L. A., and Weischedel, R. M. (2007a). Ontonotes: a unified relational semantic representation. *Int. J. Semantic Computing*, 1(4):405–419.

Pradhan, S. S., Loper, E., Dligach, D., and Palmer, M. (2007b). Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 87–92, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pradhan, S. S., Ramshaw, L., Weischedel, R. M., MacBride, J., and Micciulla, L. (2007c). Unrestricted coreference: Identifying entities and events in ontonotes. In *ICSC*, pages 446–453. IEEE Computer Society.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Recasens, M. (2010). *Coreference: Theory, Annotation, Resolution and Evaluation*. Tesis doctoral. Universidad de Barcelona.

Recasens, M. and Hovy, E. (2011). Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Recasens, M. and Vila, M. (2010). On paraphrase and coreference. *Computational Linguistics*, 36(4):639–647.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Internal Representations by Error Propagation, pages 673–695. MIT Press, Cambridge, MA, USA.

Salton, G. (1971). *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall.

Salton, G. (1989). *Automatic Text Processing. The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, volume 24, pages 513–523.

Salton, G., Wong, A., and Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sapena, E., Padró, L., and Turmo, J. (2011). Relaxcor participation in conll shared task on coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 35–39, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sapena, E., Padró, L., and Turmo, J. (2013). A constraint-based hypergraph partitioning approach to coreference resolution. *Comput. Linguist.*, 39(4):847–884.

Schäfer, U., Spurk, C., and Steffen, J. (2012). A fully coreference-annotated corpus of scholarly papers from the ACL anthology. In *Proceedings of COLING 2012: Posters*, pages 1059–1070, Mumbai, India. The COLING 2012 Organizing Committee.

Schütze, H. (1998). Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544.

Soraluze, A., Arregi, O., Arregi, X., Ceberio, K., and Díaz de Ilarraza, A. (2012). Mention detection: First steps in the development of a basque coreference resolution system. In *Proceedings of KONVENS Conference on Natural Language Processing*, pages 128–136, Vienna.

Soraluze, A., Arregi, O., Arregi, X., and Díaz de Ilarraza, A. (2015). Coreference resolution for morphologically rich languages. adaptation of the stanford system to basque. *SEPLN*, 55:23–30.

Strang, G. (2009). *Introduction to Linear Algebra, Fourth Edition*. Wellesley-Cambridge Press.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining multi-label data. In *In Data Mining and Knowledge Discovery Handbook*, pages 667–685.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

Turney, P. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188.

Widdows, D. (2004). *Geometry and Meaning*. Center for the Study of Language and Inf.

Zaman, A., Matsakis, P., and Brown, C. (2011). Evaluation of stop word lists in text retrieval using latent semantic indexing. In *ICDIM, IEEE*, pages 133–136.

Zelaia, A., Alegria, I., Arregi, O., Arruarte, A., Díaz de Illarraza, A., Elorriaga, J. A., and Sierra, B. (2009a). Exploring basque document categorization for educational purposes using lsi. In *Proceedings of the first international conference on computer supported education, volume 1*, CSEDU-2009, pages 5–10, Setubal, Portugal. INSTICC Press.

Zelaia, A., Alegria, I., Arregi, O., and Sierra, B. (2005a). Analyzing the effect of dimensionality reduction in document categorization for basque. In *Proceedings of the 2nd Language & Technology Conference (L&TC'05)*, pages 72–75.

Zelaia, A., Alegria, I., Arregi, O., and Sierra, B. (2005b). Analyzing the effect of dimensionality reduction in document categorization for basque. *Archives of Control Sciences*, 15(4):703–710.

Zelaia, A., Alegria, I., Arregi, O., and Sierra, B. (2006). A multiclassifier based document categorization system: profiting from the singular value decomposition dimensionality reduction technique. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*, pages 25–32.

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

Zelaia, A., Arregi, O., and Sierra, B. (2007). Ubc-zas: A k-nn based multiclassifier system to perform wsd in a reduced dimensional vector space. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 358–361, Prague, Czech Republic. Association for Computational Linguistics.

Zelaia, A., Arregi, O., and Sierra, B. (2009b). A multiclassifier based approach for word sense disambiguation using singular value decomposition. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 248–259, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zelaia, A., Arregi, O., and Sierra, B. (2015a). Combining singular value decomposition and a multi-classifier: A new approach to support coreference resolution. *Engineering Applications of Artificial Intelligence*, 46, Part A:279–286.

Zelaia, A., Arregi, O., and Sierra, B. (2015b). A multi-classifier approach to support coreference resolution in a vector space model. In *Proceedings of NAACL-HLT, Workshop on Vector Space Modeling for Natural Language Processing*, pages 17–24, Denver, Colorado. Association for Computational Linguistics.

Zelaia, A., Sierra, B., Arregi, O., Ceberio, K., de Ilarraza, A. D., and Goenaga, I. (2010). *A combination of clasiffiers for the pronominal anaphora resolution in Basque.*, pages 253–260. Number 6419/2010.

Zobel, J. and Moffat, A. (1998). Exploring the similarity space. *SIGIR Forum*, 32(1):18–34.