# Hitzen Adiera-desanbiguazioa

*Oier Lopez de Lacalle*

IXA taldea, Informatíka Fakultatea, EHU 20018, Donostia
oier.lopezdelacalle@ehu.es

*Eneko Agirre*

IXA taldea, Informatíka Fakultatea, EHU 20018, Donostia
e.agirre@ehu.es

**Laburpena:** Gure hizkuntza anbiguoa da. Hitz batek hainbat interpretazio ditu agertzen den testuinguruaren arabera, eta zein adiera hartzen duen asmatzea ez da lan erraza, nahiz eta guk era naturalean egin. konputazio-metodoak erabiliz hitzen agerpenei adiera egokia ematea hitzaren adiera-desanbiguazioa (HAD) deritzo. HAD automatikoa ezagutzan oinarritzen da, eta ezagutza hori hainbat iturritatik lor dezake: adierez etiketatutako corpus batetik hasita ontologietaraino. Zoritxarrez, baliabide hauen sortze-prozesua garestia da eta ezagutza bereganatzearen arazo bezala ezagutzen da. Eragozpenak gainditua HAD teknologiak heldutasuna lortzen duen unean informazioa atzitzeko dugun modua erabat aldatuko da, Web Semantikoari ateak zabalduz. Lengoaia Naturalaren Prozesamendurako tresnetan ere lagungarri da HAD automatikoa, Itzulpen Automatikoan adibidez, izan ere, bai polisemia eta bai sinonimia arazoak automatikoki gainditzeko balio baitezake. Artikulu honetan hitzen adiera-desanbiguazioaren sarrera orokor bat egiten dugu eta, bereziki, adierez etiketatutako corpusetan oinarritzen diren metodoei erreparatuko diegu. Erakutsiko dugu hitzen adiera-desanbiguazioa ataza zaila dela, hizkuntzaren konplexutasunari aurre egin eta testu hutsetik egitura semantikoa antzeman behar duelako.

**Abstract:** Our language is ambiguous. A word has a different interpretation depending on the context, and it is not an easy task to guess the meaning even if humans do naturally. The automatic process of deciding which sense is being to be used in a particular context is known as Word Sense Disambiguation (WSD). WSD is based on different types of knowledges from sense-labeled corpus to an ontology. Unfortunately, the creation of this type of resources is expensive, which is the main drawback in WSD, and we refer to this as knowledge acquisition bottleneck. Once WSD overcomes the disadvantages of knowledge acquisition, it will be the keystone for information access technology and the Semantic Web. In addition, WSD can be useful for Natural Language Processing tools, such as Machine Translation, which deals automatically with both the polysemy and the synonymy problems. The aim of this paper is to provide a general introduction to Word Sense Disambiguation. Specifically, we will focus on hand tagged corpus-based methods that are based on Machine Learning methods. We will show the difficulty of the WSD task, because it has to deal with the complexity of the language and extract semantic structures from scratch.

## 1. SARRERA

Gure hizkuntza, elkar ulertzeko erabiltzen dugun hizkuntza, anbiguoa da. Hitz batek hainbat interpretazio ditu agertzen den testuinguruaren arabera, *banku* hitzaren ondoko adibideek azaltzen duten bezala:

i: Parkeko *bankuan* eseri nintzen egunkaria irakurtzera.
ii: Dirua *bankuan* gorde dut.

Lehenengo adibidea esertzeko erabiltzen den altzariari buruzkoa da, eta bigarrena banketxeari buruzkoa.

Testuinguru bakoitzean hitz bakoitzak hartzen duen adiera zein den asmatzea ez da lan erraza makinentzat, nahiz eta guk era naturalean egin. Makinak saiatzen dira egituratu gabeko testuak prozesatu, lotutako informazio linguistikoa analizatu, eta, testuinguruaren arabera, hitzen esanahiak interpretatzen. Prozesu automatiko honi *hitzaren adiera-desanbiguazio* (HAD) deritzo, eta konputazio-metodoak erabiliz hitzen agerpenei adiera egokia ematean datza (Agirre & Edmonds, 2006).

Goiko adibideekin jarraituaz, *banku* hitzerako ondoko definizioak topatu ahalko genituzke Euskal WordNet ontologian (Agirre *et al*., 2006b):

*banku 1:* Eserleku luzea, bizkarduna nahiz bizkarrik gabea, hainbat lagun batera esertzeko aukera ematen duena.

*banku 2:* Bezeroen diru-gordailuak onartu eta kreditu-eragiketak egiten dituen enpresa publiko edo pribatua.

Goiko lehen adibidean (i), HAD prozesu automatikoak *banku* hitzaren lehen adiera aukeratu beharko luke *(banku 1)*, eta bigarrenean (ii) *banku 2*. Era berean, HAD sistemak berdin desanbiguatuko lituzke testuinguruko gainontzeko hitzak ere.

HAD ataza Lengoaia Naturalaren Prozesamenduaren (LNP) eta Adimen Artifizialaren (AA) baitan kokatzen da. LNP bezala, HAD ere *AI-complete*[1] arazoa dela esan ohi da. Alegia, problema ebatzi ahal izateko adimen artifizialeko arazo nagusiak ebatzi, eta makinaren ahalmena giza adimenarekin parekatu beharko litzateke (Turing, 1950). Izan ere, gizakiok gure ezagutza eta arrazonamendu-ahalmen guztiak behar ditugu hitzen esanahia zein den atzemateko.

HAD automatikoa ere ezagutzan oinarritzen da, eta ezagutza hori hainbat iturritatik lor dezake: adierez etiketatuko testu multzoak, hiztegi elektronikoak, sare semantikoak, ezagutza-baseak edo ontologiak. Zoritxa-

[1] http://en.wikipedia.org/wiki/Al-complete

rrez, eskuz landutako baliabide horiek sortzeak denbora asko eskatzen du eta oso garestia izan ohi da. Gainera, aplikazio-eremua aldatuz gero (besteak beste, domeinuaren aldaketa, beste hizkuntza bat erabiltzea, edo adiera-hiztegi berria erabiltzea), berregin beharko lirateke baliabideak. Hain zuzen ere, bi horiek dira HAD sistemak aurka dituen oinarrizko eragozpenak, *ezagutza bereganatzearen arazo*[2] bezala ere ezagutzen direnak (Gale *et al*., 1992).

Eragozpenak gainditua HAD teknologiak heldutasuna lortzen duen unean gaur egun informazioa atzitzeko dugun modua erabat aldatuko da. Internet eta informazioaren teknologiek eskura ditugun datuen kopurua era esponentzialean handitzea ekarri dute, eta geroz eta ugariagoak dira dauzkagun dokumentu-bilguneak, web-guneak, zientzia-artikulu, blogak, twitak, etab. Informazio multzo handi hauekin egokiro lan egin ahal izateko, ezinbestekoa bihurtu da prozesu automatiko berrien garapena. Orain arte garatu diren testu-meatzaritza eta informazio-bilaketan erabilitako teknikek baina, hitzen formei baizik ez diete erreparatzen oinarrian; alegia, karaktere kate hutsak baizik ez dituzte kontuan hartzen, eta alde batera uzten dute gizakiok atzematen ditugun esanahiak.

Izan ere, informazio-bilaketan dauden bi arazo nagusi polisemia eta sinonimia dira. Polisemia dela-eta, adibidez, finantza-erakundeei buruzko informazioa nahi dugunean *banku* hitza sakatzen badugu gure aukerako interneteko bilatzailean, eserlekuei buruzko hainbat webgune ere azaldu ahal zaizkigu. Sinonimiari dagokionez, adibide berean, ez genituzke lortuko *banku* hitza esplizituki erabili gabe *banketxe, kutxa, aurrezki-kutxa* edo *finantza-erakunde* bezalako hitzak biltzen dituzten dokumentuak.

HADk bai polisemia eta bai sinonimia arazoak automatikoki gainditzeko balio dezake, eta horrela testu izugarri handi horiei informazioa geruza aberatsago bat gehituta erabiltzaileoi ahalbideratuko litzaiguke esanahiaren bitartez lan egitea, aplikazio aurreratuagoak garatzeko aukera emanez. Geruza aberatsago horren erakusle da, adibidez, Web Semantikoaren proposamena (Berners-Lee *et al*., 2001):

> «Web Semantikoa ... sarearen hedakuntza da; bertan, informazioa zehatz definitutako esanahiaz hornitzen da, eta elkarrekintza hobea ahalbideratuko du ordenagailu eta gizakien artean»

Itzulpen Automatikoan izan dezakeen ekarpena ere jakina da. Adibidez, ingelesetik euskarara itzultzen duen sistema bati *bank* hitza emanik, bere testuinguruaren arabera HAD sistema batek bere adiera desanbiguatuko luke, eta hortaz esanahiaren arabera *banku, itsulapiko, ur-ertz* edo *peralte* itzulpenen artean aukeratzen jakingo luke.

---

[2] Knowledge acquisition bottleneck, ingelesez.

### 1.1. **Aplikazioak**

Arestian aipatutako ahalmen horiek baina, ez dira oraindik aplikazioetara zabaldu. HAD sistemen eraginkortasun maila apala izan daiteke horren arrazoi nagusietako bat, baina kontuan hartu behar da oraindik ere ez dagoela guztiz ebatzia adiera desanbiguazioak ematen duen informazioa aplikazio bakoitzean txertatzeko modu eraginkorrena zein den. Hasiera batean oso era bakunean txertatzen zen informazioa aplikazioetan. Azken urteetan aldiz, metodo konplexuagoak garatzen ari direlarik, emaitza hobeagoak lortzen dira eta etorkizunerako itxaropena zabaltzen ari da.

Azpiatal honetan HAD sistemek eragin positiboa eduki lezaketen aplikazio batzuk zerrendatuko ditugu.

**Informazio Berreskurapena (IB).** Aipatu dugun bezala, gaur egungo bilatzaile arruntek (Google, Bing eta gainontzekoek) ez dute semantikaren erabilera esplizitua egiten dokumentu esanguratsuak ez diren dokumentuak alde batera uzteko; alegia, erabiltzaileak galdetzen duenaren esanahiaren araberako bilaketarik ez dute egiten. HAD beharrezkoa aurreikusten da hitz beraren erabilera ezberdinak bereizi eta antzemateko. Adibidez, modu era honetara, *«depresio»* hitzaz galdetuz gero, zein dokumentu mota itzuli beharko ote lituzke bilatzaileak, ekonomiari buruzkoak, gaixotasunari buruzkoak, ala eguraldiari kasu egiten diotenak? Maila oneko HAD sistema batek bilaketaren doitasuna hobetuko luke interesgarriak ez diren dokumentuak alde batera utziz, edota bilaketaren emaitzak esanahiaren arabera multzokatuz. Hori egiteko gai izanez gero, emaitzen estaldura hobetu egingo litzateke, esanahi berdina duten beste hitzei (sinonimoei) lotutako dokumentuak itzuliz gero.

Hasierako ikerketek iradoki zuten HAD sistemaren asmatze-tasak gutxienez %90ekoa izan behar duela IB sistema batean laguntzeko, baina azken ikerketek HAD egiteak hizkuntza anitzeko IBn eta dokumentu sailkapenean lagungarri dela erakutsi dute (Vossen *et al*., 2006; Bloehdorn & Hotho, 2004; Clough & Stevenson, 2004; Otegi *et al*., 2008).

**Informazio Erauzketa (IE)**. Testuaren analisi egoki bat egiteko beharrezkoa da adieren arteko bereizketa egitea. Horrela posible izango dugu medikuntzako domeinuan sendagaien artean bereizketa egitea, ala beharrezkoa ikusiz gero, biologia arloko dokumentuetan geneen eta proteinen artean bereizketa egitea. Bestalde, Web Semantikora jauzi eginez gero, beharrezkotzat jotzen da dokumentuak semantikaren arabera aberastuak egotea (hau da, erreferentziazko ontologia batera lotuta egotea); horrela, agenteen arteko elkarrekintza aurreratua ahalbideratuko dugu, hau da, karaktere kate hutsetik kontzeptuen eremura salto egitea. Horren adierazpide dira, Kyoto proiektuan egiten ari diren ahaleginak (Agirre *et al*., 2010).

**Itzulpen Automatikoa (IA).** Testuinguruaren arabera hitzak adiera bat edo beste hartzen duenez, arestian esan dugun bezala, HAD beharrezkoa

ikusten da lexikoaren hautapen zuzena egiteko. Azken urteotako lanek agerian utzi dute HAD sistemen erabilgarritasuna IA sistemetan (Carpuat & Wu, 2005; Chan *et al*., 2007).

**Lexikografia.** HAD sistemen eta lexikografoen arteko hartu-emana onuragarria izan daiteke. HAD egiten duen sistema batek esanahiaren araberako hitzen multzoak egin ditzake, estatistikoki ezberdinak diren adiera berrien testuinguruak identifikatuz. Honek, adibidez, hiztegi hobeak egiteko aukera emango luke. Lexikografoek, adierak hobeto desberdintzeko sistema bat eskuartean izanik, hiztegi hobeak sortuko lituzkete, eta alderantziz, hiztegi landuagoak egonda HAD hobea egin ahalko genuke. Horren adibide da *Sketch-engine* deritzon konputazio-lexigrafirako tresna (Killgariff & Tugwell, 2004; Kilgarriff *et al.,* 2010).

## 1.2. **HADn historia laburtua**

Hitzaren adiera-desanbiguazioa LNPko lehen atazetako bat izan zen. 40. hamarkadaren bukaera aldera itzulpen automatikoko oinarrizko ataza bezala sortu zuten (Weaver, 1949). Laster ikusi zen testuinguruaren garrantzia adiera zuzena aukeratzeko orduan. Neurri estatistikoek eta ezagutza-iturriek lagundu zezaketela ikusita, neurri hauek garatzeari ekin zioten orduko ikerlariek. 60. hamarkadarako HAD sistemaren zailtasuna agerian geratu zen, konputazio-teknikak erabiliz aurrera egitea ez baitzuten lortu. Hori horrela, 70. hamarkadan adimen artifizialeko hasierako metodoak erabiltzen hasi ziren munduaren ezagutza HAD sistemetan baliatze aldera. Teknikak ugaritu egin ziren, baina hiztegi elektronikoen falta zela-eta, zaila zen emaitzak orokortzea eta ondorio orokorrak ateratzea ezinezko suertatu zitzaien garai hartako ikertzaileei. Aldiz, 80. hamarkadan hiztegi elektronikoak ugaritu egin ziren —eta horiekin ezagutza erauzteko teknikak ere bai—; ondorioz, emaitzen orokorpena ahalbideratu zen. Azkenik, 90. hamarkadan metodo estatistikoak gailendu ziren. Garai honetan hasten dira HAD sistemaren ebaluazio-esparru orokorrak garatzen, hots, adiera-desanbiguazioko lehen lehiaketak sortu ziren sistemen ebaluazio bateratuak egiteko.

## 1.3. **HADn definizioa**

Hitzaren adiera-desanbiguazioan, testuinguru jakin bateko adiera zehazten da konputazio prozesu baten bidez. HAD sistema sailkapen arazo bezala formalizatu ohi da, bertan hitz bakoitzerako adierak aukeratu beharreko *klaseak* azaltzen direlarik. Hori dela-eta, *sailkapen automatikorako metodoak* erabiltzen dira hitzari adiera bat edo gehiago esleitzeko. HAD sistemak *testuingurutik* erauzitako ebidentzien eta *kanpoko ezagutza* iturrietako ezagutzaren arteko konbinazioan oinarritzen dira desanbiguazioa egiteko.

LNPko beste ataza asko ere sailkapen arazo bezala formalizatu izan dira: hitzen gramatika-kategoria zehaztea (adibidez, testuinguru *jakin* batean jakin izen edo adjektibo bezala erabili den zehaztea), entitate izendunen sailkapena (adibidez, testuinguru jakin batean *Azpeitia* abizen bezala erabili dela zehaztea), edota dokumentuen sailkapena (adibidez, dokumentu jakin bat kirol ala ekonomia domeinukoa den zehaztea). Ataza horien eta HADn arteko ezberdintasun nagusia sailkapenerako klaseetan dago: Aurreko atazak aurrez definitutako sailkapen-zerrenda mugatu batean oinarritzen diren bitartean, HADn desanbiguatzera goazen hitzaren araberakoa da klase-zerrenda, hau da, hitzaren adieren araberakoa. Era honetara, HAD *N* sailkapen arazo bezala ebatzi ohi da, *N* hiztegiko hitzsarrera kopurua delarik.

Jarraian HAD atazan parte hartzen duten lau elementu nagusiei erreparatuko diegu: Hitzen adieren aukeraketa, kanpoko ezagutza-iturriak, testuinguruaren irudikapena eta metodo automatikoaren aukeraketa. Guztiak azalduko ditugu baina azken bi elementuei buruz sakonago arituko gara artikuluan zehar.

## 2. **ADIERAREN DEFINIZIOA**

Hitzen adierak testuinguruaren arabera aldatzen dira, hots, testuinguruaren arabera hitzak adiera bat edo beste izango du. Lexikografoek, hitzen adierak aztertzen dituztenean, zailtasunak dituzte hitzak azpiadieretan banatzeko. Askotan sumatzen dituzte adieren definizio zehaztugabeak edota bata bestearekin teilakatzen diren adierak. Hitzaren adiera berri bat zehazterako orduan, erabilera ohikoenetatik beste erabilera bat antzematen dutenean sortzen dute adiera berria. Lan zaila izanik, zenbait ikerketa lanek hainbat teoria aurkeztu dituzte hitzen adieren ereduak aurkezteko, baina teoria hauek hasierako ikerkuntza-fasean daude oraindik.

*Polisemiak* adiera anitza esan nahi du, eta testuingururik ez dagoenean hitzaren berezko ezaugarria da «anbiguotasuna». Ondorioz, anbiguotasuna testuaren ezaugarria ere bada, beti ere prozesamenduaren ikuspegitik. Idaztean ala hitz egitean zehaztasun faltak anbiguotasuna dakar eta, polisemiak anbiguetate potentzialaren berri ematen duen bitartean, testuinguruak anbiguetate horren aurka egiten du; alegia, testuinguruko hitzen elkarrekintzaren bidez adierak zehazten dira eta esanahia argitzen da.

Hitzaren adierak elkarren arteko erlaziorik ez badute, hitz homografo deritze. Ondorioz, erraz bereizgarriak diren adierak dira HADn ikuspegitik. Bereizketa xeheago batera jotzean adierek erlazio hertsiak erakusten dituzte beren egituretan. Fenomeno honi polisemia deritzo orokorrean, eta adiera hauen erabileren hedaduraz *metafora* antzeko fenomenoak aurkitzen ditugu. Beraz, adieren bereizketa mailaren arabera, adiera multzo batekin

ala beste batekin lan egingo dugu. Aurrerago ikusiko dugunez, adieren banaketa fin batek zailtasuna gehituko dio atazari.

## 3. **EZAGUTZA-ITURRIAK**

Ezagutza, oinarrizkoa den osagaia da HAD egiteko unean. Ezagutza-iturriek beharrezko informazioaz hornitzen gaituzte testuko hitzak adierekin uztartzeko. Ezagutza-iturri mota asko daude: testu-corpus etiketatuak, testu-corpus ezetiketatuak, hiztegiak, tesauroak, ontologiak, e.a. Hemen ezagutza-iturri erabilienen deskribapen motz bat egingo dugu. Deskribapen luzeago baterako jo (Agirre & Edmonds, 2006) lanera.

— Baliabide egituratuak:

- *Tesauroak*: Hauetan hitzen arteko erlazioen berri eskuratzen dugu. Oinarrian sinonimia eta antonimia erlazioak gordetzen dituzte hitzek, baina beste erlazio batzuk gorde ditzakete.
- *Hiztegiak:* Hitzen definizioekin batera, adierak ematen dizkigute, eta gaur egun oinarrizko baliabide ditugu. Hiztegiaren arabera adiera xeheagoak ala zabalagoak (definizio orokorragoak) izango ditugu.
- *Ontologiak*: Mundu errealaren gaineko kontzeptualizazioak dira, eta bertan hitzekin izendatzen ditugun kontzeptuak modu hierarkikoan antolatuta daude (normalean egitura taxonomikoa jarraitzen dute) eta aukera ematen dute mundu errealari buruzko inferentziak egiteko. Esparru horretan, WordNet da LNPan erabiliena den ontologia, baina badira beste batzuk. WordNet hiperonimia, hiponimia edo sinonimia bezalako erlazio semantikoak biltzen dituen sare semantikoa da. Hasieran ingeleserako sortu bazen ere, hainbat hizkuntzatarako sortu da, beraien artean euskararako ere (Agirre *et al*., 2006b).

— Hauen artean garrantzitsuenak testu-corpusak dira, hauetatik hitzak desanbiguatuko dituzten ereduak erauziko dira-eta. Orokorrean bi mota daude:

- *Corpus gordinak:* Testu gordinak dira, inongo hizkuntz informazio espliziturik gabekoak. Corpus mota hauek nahiko ugariak eta heterogeneoak dira. Ingeleserako ezagunenak *British National Corpus* (BNC) eta *Wall Street Journal* (WSJ) corpusak dira. Gurean, euskarako hainbat corpus sortuak badaude; LNPan erabilienak *Euskaldunon Egunkaria* corpusa eta Elhuyarren *Zientzia eta Teknika Corpusa* (ZTC) dira. Corpus hauetatik hitzen askotariko maiztasun-kontaketak lortzen dira; hauek oso erabilgarriak dira ikasketa automatikoan.

- *Corpus etiketatuak:* Gure kasuan hitzak dagokien adieraz etiketatutako testu-corpusak dira. Ikasketa automatikorako guztiz beharrezko den baliabideak dira; izan ere, corpus hauetatik hitz-adieren sailkatzaileak sortzen dituzte. Esan beharrik ez dago, halako baliabideek denbora eta kostu handia eskatzen dutela. Ondorioz, oso eskasak dira corpus etiketatuak. Corpus etiketatuen artean, SemCor (Miller *et al.,* 1993) ingelesarako WordNeteko adierez etiketatutako corpusa da gehien erabiltzen dena. Euskararako ere sortu da horren parekoa (Agirre *et al.,* 2006a). Corpus hauen aukera onetik dator gehien erabiltzen den ontologia WordNet izatea.

## 4. TESTUINGURUAREN GARRANTZIA: IKASKETARAKO EZAUGARRIAK

Desanbiguatzera goazen hitzaren testuinguruak emango digu HAD prozesuan aurrera egiteko beharrezkoa den informazioa. Testuingurua, hutsean hartuta, egiturarik gabeko hitzen segida da, eta hortik informazioa erauzi ahal izateko hainbat analisi ezartzen zaizkio testu hutsari. Informazio egituratuari *ikasketarako ezaugarriak* deritzo. Beste hitz batzuetan esanda, ezaugarriek testuingurutik beharrezkoa den informazioa antzematen dute aztergai duten hitza desanbiguatzeko. Konputazio-mugak direla-eta lortzen diren ezaugarriek testuinguruaren informazio zati bat erauzten dute; alegia, ezaugarriek testuinguruaren orokortze bat kodetzen dute. Ezaugarriek kodetzen duten ezagutzari buruz gehiago jakiteko jo bedi (Agirre & Edmonds, 2006) liburura.

Normalean, aurreprozesu konplexua eramaten da aurrera ezaugarri multzoa lortzeko. Horrela sarrerako testuari (e.g., esaldia, paragrafoa edota dokumentu osoari) ondoko analisi katea ezartzen zaio:

— *Tokenizazioa:* Normalizaziorako pausoa da. Testua *tokenetan* banatzen da, normalean hitz bakoitzaren hasiera eta bukaera adierazita.
— *Gramatika-kategorien zehaztapena:* Testuko hitz bakoitzari dagokion gramatika-kategoria zehazten da pauso honetan. Adibidez: «Basoa/IZE perretxikoz/IZE bete/ADI zen/ADL./PUNT», non IZE, ADI eta ADL izena, aditza eta aditz laguntzailea diren hurrenez hurren.
— *Lematizazioa:* hitzak oinarrizko formara murrizten dira pauso urrats honetan, eta horrela, *basoa* → *baso* bihurtzen da eta, *perretxikoz* → *perretxiko, bete* → *bete, zen* → *izan*.
— *Chunking* edo sintagmetan banatzea: Testua banatu egiten da elkarren artean erlazio sintaktikoak dituzten zatitan. Aurreko adibidearekin jarraituz: $[\text{Basoa}]_{IS}$ $[\text{perretxikoz}]_{IS}$ $[\text{bete zen}]_{AM}$ non IS izen-sintagma eta AM aditz-multzoa diren.
— *Parsing* edo egitura sintaktikoaren analisia: Testuko izen-sintagmen edo hitzen arteko erlazioak erauzten dira azken pauso honetan.

Horrela, lortzen dugu sarrerako testutik hainbat mailatako hizkuntz informazio erauztea. Aurreprozesuan hitzaren testuingurutik (izan esaldia ala dokumentu osoa) egituratze ezberdineko informazioa erauzi eta hau bektore moduan adierazten da. Hitzaren adiera bereizteko ikasketarako ezaugarriak dira euskarri garrantzitsuenak. Gaiko hainbat lanek erakutsi digute mota askotariko ezaugarriak erabili behar direla HADren etekin ona bermatu ahal izateko. Gai honen inguruan sakondu nahi izanez gero, jo bedi (Agirre & Martínez, 2001) lanera.

Orokorrean ikasketa-ezaugarriak 4 multzotan bana ditzakegu:

— **Ezaugarri lokalak:** desanbiguatzera goazen hitzaren inguruan dauden bigramek (jarraian datozen bi hitzetako multzoek) eta trigramek (hiru hitzetako hitz multzoek) osatzen dute. Ezaugarri hauek lemaz, hitz-formaz edo beraien kategoriaz osatuta daude. Beste ezaugarri lokalak testuinguruko aurreko/ondorengo lema/hitz-formek osatzen dituzte.
— **Ezaugarri topikalak:** testuingurua osatzen duten lema guztiak (izen, adjektibo, aditz, aditzondo) hartzen dira eta agerpen-hurrenkera kontuan hartu gabe *hitzen poltsa* deritzon ezaugarri multzoa sortzen da. Beste teknikak batzuek, adibidez, (Pedersen, 2001) lanean deskribatutakoak, lortzen dute testuinguruko bigrama esanguratsuenak erauztea, hauek ezaugarri bezala erabiltzeko.
— **Ezaugarri sintaktikoak:** patroi heuristikoak eta desanbiguatzera goazen hitzaren inguruko kategoriaren bidez definitutako *adierazpen erregularrak*[3] erabilita erauzten den informazio sintaktikoa. Ondorengo erlazioak erauzten dira normalean, gehiago izan badaitezke ere: osagarri zuzena, subjektua, modifikatzailea, preposizioa eta anai-arreba sintaktikoak.
— **Ezaugarri semantikoak:**Informazio semantikoa adierazten da, eta era honetara, aurretik lortutako beste hitzen adierak erabil daitezke. Hala ere, normalean testuaren domeinua ala domeinuak zehazten dira.

**1. taula:** Gramatika-kategoriez osatutako ezaugarri-bektore sinplea baso hitzaren (a) eta (b) adibideetatik erauzia

| esaldia | W-2 | W-1 | W+1 | W+2 | adiera |
|---|---|---|---|---|---|
| (a) | adjektiboa | aditza | - | - | OIHAN |
| (b) | - | - | adjektiboa | izena | EDALONTZI |

[3] Adierazpen erregularrekin testuetatik interesatzen zaigun informazio egituratua lortzen dugu hainbat patroi definituz.

Goiko sailkapenean oinarritutako ondoko ezaugarriak erauziko genituzke beheko bi esaldietatik. Demagun *baso* hitza dela desanbiguatu beharrekoa:

(a) Zuhaitz handiak ditu basoak.
(b) Baso bete ur edan dut.

4. taulak (a) eta (b) esaldietako baso hitzaren ikasketa-ezaugarriak erakusten ditu. 4 hitzeko leihoan (±2 hitz desanbiguatzera goazen hitzetik abiatuta) azaltzen diren gramatika-kategoriak bektore eran ipintzen dira. Ezaugarri hauek taulako eskumako adierak ikasteko erabiltzen dira. Noski, ezaugarri-konbinazio konplexuagoa eskatzen du atazak ikasketa aurrera eraman ahal izateko. Horrela, arestian esandako beste ezaugarri lokal, topikal, sintaktiko eta semantiko batzuez betetzen dira ezaugarri-bektoreak.

Mota honetako ezaugarri lauak (hau da, ezaugarri-bektore lauak) ikasketa gainbegiraturako dira egokiak. Aldiz, egituratutako adierazpideak metodo gainbegiratugabeetan eta, bereziki, ezagutza-baseetan oinarritzen diren ikasketa metodoetan dira erabilgarri. Jarraian ikasketarako metodoen hurbilpen esanguratsuenak aurkeztuko dizkizuegu. Guztiaren ikuspegi orokorra erakutsiko dizuegu baina ikasketa gainbegiratuko hurbilpenetan sakonduko dugu gehien bat.

## 5. **HAD EGITEKO METODOAK**

Hainbat modu daude HAD egiteko teknikak sailkatzeko. Metodoak erabiltzen duen ezagutza motaren arabera sailkatzea da ohikoena: alde batetik ezagutzan oinarritutakoak (ezagutza aberatsekoak edota hiztegietan oinarritutakoak bezala ere ezagutzen direnak), eta corpusean oinarritutakoak (edo ezagutza urrikoak) bestetik. Lehenengoek baliabide lexikal egituratuak erabiltzen dituzte eta haien erabilera esplizituan oinarritzen dira. Corpusean oinarritutakoek aldiz, ez dute inongo ezagutzarik erabiltzen eta ikasketarako eman zaien corpus etiketatuan bakarrik oinarritzen dira, adiera-desanbiguazioa egiteko. Informazio-iturri biak konbinatzen dituzten metodoak ere badaude. Hauei metodo hibridoak deritze, eta gero eta gehiago erabiltzen ari dira egun.

Corpusean oinarritutakoak erabiltzen den ikasketa automatiko motaren arabera ere sailka daitezke:

— HAD gainbegiratua: Aurretik eskuz desanbiguatu eta etiketatutako testuetan oinarritzen dira hitz adierak ikasteko. Aurreko atalean adierazi bezala, corpus batetik erauzten dira ikasketa-ezaugarriak, eta ondoren eredua induzitzen da hitz bakoitzerako. Induzitutako ereduak dira sailkatzailearen oinarria.

— HAD gainbegiratugabea: Eskuz etiketatu ez den corpusetan oinarritzen diren metodoak dira. Aurrez erabakitako adiera multzoetan oinarritzen ez direnez, hitzen erabilerak multzokatzen dituztela esaten dugu. Multzokatze metodo ezagunenak erabiltzen dira HAD gainbegiratugabea egiteko.

Azkenik, HAD teknika *tokenean* edo *hitz motan* oinarritu daiteke. Tokenean oinarritzen dena hitzaren agerpenaren testuinguruaren araberako desanbiguazioa burutuko du. Bestetik hitz motetan oinarrituz gero, hitzaren agerpen guztiei adiera bera esleituko zaie. Izan ere, hipotesi modura onartuko dugu testu jakin batean agertzen den hitzak adiera bera izango duela beti.

Eskarmentua dugu aipatutako HAD mota horietan guztietan (Agirre & Soroa, 2009), baina lan honetan, HAD egiteko metodoak aztertuko ditugu batez ere.

## 6. **HAD GAINBEGIRATUA**

Esan bezala gainbegiratutako HAD metodoek ikasketa automatikoa erabiltzen dute eskuz etiketatuko datu multzotik sailkatzailea induzitzeko ala ikasteko. Normalean, sailkatzaileak hitz bakarrari erreparatzen dio eta honen agerpenak desanbiguatzen saiatzen da, dagokion adierarekin sailkatuz. Hortaz, entrenamendurako datu multzoak itu-hitzen (desanbiguatu beharreko hitzen) adibideak, hiztegi bati dagokien adierez etiketatuak, gordetzen ditu.

Modu formalagoan esanda, gainbegiratutako sailkapena, $S$ entrenamendurako datu multzoa emanik, ezezaguna den $f$ funtzioaren $h$ hurbilketa induzitzea da. Induzitutako funtzioak $X$ sarrerako espazio diskretua $Y = \{1,..., K\}$ irteerako espazio ez-ordenatura mapatuko du.

Demagun entrenamendurako datu-multzoak $m$ adibide dituela, $S = \{(x1, y1) ,..., (x^m, y^m)\}$, non $(x, y)$ bikote bakoitzean $x \in X$ eta $y = f(x)$ diren. $x = (x_1,..., x_n)$ bektore bat da eta ikasketa-ezaugarri deritzen bere osagaiek, balio diskretuak edota balio errealak gordetzen dituzte. Ikasketa-bektoreko osagaiek entrenamendurako adibidearen informazio eta propietate esanguratsuak deskribatzen dituzte. Entrenamendurako adibideen irteerako $Y$ balioei *klasea* deritze. Horrela, entrenamenduko adibide bakoitza ezaugarri-bektoreen bidez eta dagokien klase-etiketaren bidez deskribatzen da. Gure kasurako entrenamendurako adibideak itu-hitzaren agerpenak dira (ezaugarri-bektoreez adierazita) eta klaseak, adibide horiei esleitutako adierak.

Màrquez eta beste autore batzuek (Agirre & Edmonds, 2006) liburuan gainbegiratutako HAD tekniken sailkapen orokor bat ematen dute. Ondokoa da egiten duten sailkapena:

— *Metodo probabilistikoak:* Adieraren eta ikasketa-ezaugarrien baldintzapeko probabilitatea adierazten duten parametro probabilistikoen estimazioan oinarritzen dira. *Naïve Baye*s eta *Maximum Entropy* dira hauetan metodo ezagunenak[4].
— *Adibideen antzekotasunean oinarritzen direnak:* Antzekotasun neurri baten arabera erabakitzen dute hitzaren adiera. Antzekotasuna askotarikoa izan daiteke, baina normalean ezaugarriek definitutako espazioan emaniko distantziak (adibidez, distantzia euklidearra) hartzen dira kontuan. Kategoria honetan Bektore Espazioaren Eredua eta *k-Nearest Neighbours* dira metodo arrakastatsuenak.
— *Diskriminazio erregeletan oinarritzen direnak:* Erregela multzo bat betetzen duten adiera edo adierak esleitzen dituzte metodo hauek. Hauetan Erabaki Zuhaitzak eta Erabaki Zerrendak dira ezagunenak.
— *Sailkatzaile linealak eta kerneletan oinarritutakoak:* Adiera desberdineko adibideak banatzen dituen hiperplanoa kalkulatzen dute. Ezagunenak Sostengu Bektoreen Makinak (*Support Vector Machine* izenez ezagunagoak), Pertzeptroia eta Winnow dira.
— *Konbinazio-erregeletan oinarritzen direnak:* AdaBoost bezala, sailkatzaile bakun eta doitasun handirik gabekoen konbinazio lineal bat egiten dutenak.

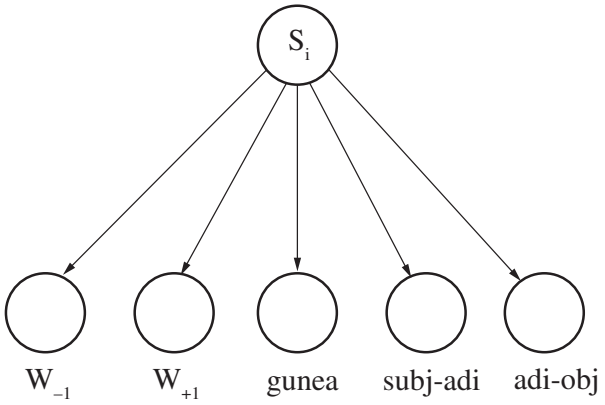
Orokorrean, gorago esan bezala, itu-hitz baten agerpena deskribatzeko ezaugarriak erauziko ditugu eta ondoren ikasketa automatikoko metodoak pisu (probabilitate) bat emango digu adiera bakoitzeko. Azkenik horietan balio handiena duen adiera aukeratuko dugu.

Eraginkortasunari dagokionean, corpusean oinarritzen diren metodo gainbegiratuak dira emaitza hoberenak lortzen dituztenak. Beraien doitasun maila faktore askoren menpe badago ere, adierez etiketatutako adibideen kopurua da faktore garrantzitsuena (Yarowsky & Florian, 2002). Ikus bedi 7. atala adibide gehiago lortu nahi izanez gero.

Jarraian artearen egoera definitzen duten eta erabilienak diren ikasketa automatikoko metodoak banan-banan azalduko ditugu.

[4] Sare Bayesiarrak kategoria honetan sartuko lirateke. Dena den, ezaugarri multzoaren konplexutasuna dela-eta, ez dago teknika hau erabiltzen duen lan esanguratsurik. Esan daiteke Naïve Bayes dela Sare Bayesiar bakunena.

## 6.1. **Naïve Bayes**




**1. irudia.** Naive Bayes sare bayesiar eredua

Hauxe da Sailkatzaile estokastiko[5] bakunena, baina baita ikasketa automatikoaren arloan eta LNPn arrakasta erdietsi duena. Bayes sailkatzailearen inplementazio jakin honek, izan ere, ezaugarri asko eraginkortasunez konbinatzeko gaitasuna du. Gainera, ondo dabil sarrerako datuen multzoa oso handia denean.

Probabilitatearen banakako banaketan oinarritzen da. Adibide baten klasea asmatzeko, behatutako adibidearen probabilitatea maximizatzen duena aukeratzen da. Horretarako, Bayesen teorematik eratorritako formula bakun bat erabiltzen da; bertan ezaugarri guztiei dagozkien balioak emanik $(f_1, f_2, \ldots, f_j, \ldots f_n)$ adiera klase probableena $(\hat{S})$ aukeratzen da:

$$\hat{S} = \arg\max_i P(S_i|f_1, \quad, f_m) = \arg\max_i \frac{P(f_1, \quad f_m \mid S_i)P(S_i)}{P(f_1, \quad f_m)} \sim P(S_i)\ \Pi_{j=1}^{m}\ P(f_j \mid S_i) \tag{1}$$

Naïve Bayes sailkatzailea hipotesi batetik abiatzen da: atazaren deskribapenerako erabilitako atributu edo ezaugarri bakoitza beste edozein ezaugarri bezain garrantzitsua da; alegia, adiera klasearen baldintzapean independenteak direla atributu guztiak. Hipotesi hau, ordea, ez da betetzen askotan. Hala eta guztiz ere, baldintza hori betetzen dela suposatzeak dakarren sinplifikazioak eredu dotore eta eraginkorrak eskaini ohi ditu (Ng, 1997; Escudero *et al*., 2000). Eskuratutako ezagutza ezin da era ulergarrian azaldu eta horregatik,

[5] Atazan definitutako atributuen dependentzia probabilistikoak deskribatzen dituzte eredu estokastikoek, eta grafoen bidez irudikatzen dira normalean. Grafoko adabegi bakoitzak zorizko aldagai bat adierazten du eta probabilitate-banaketa bat du esleituta (ikus 1. irudia)

ikasketa azpisinbolikoko algoritmoa dela esaten da. Internet sarean badaude Naïve Bayes algoritmoaren erabileraren adibide argigarriak[6].

Azken erabakian ekuazioko izendatzaileak garrantzirik ez duenez, eragiketak arintzeko kendu egiten da. $P(S_i)$ eta $P(f_j / S_i)$ probabilitateak entrenamendu corpusetik maiztasun erlatiboen bitartez estimatzen dira. Ikasketa-corpusean inoiz gertatu ez diren kasuak gerta daitezke test-corpusean. Batez ere, ikasketa-corpusa txikia den kasuetan gertatzen da hau. Halakoetan, probabilitateekin gabiltzanez, teknika desberdinak daude inoiz gertatzen ez diren kasuei zero ez den probabilitate txiki-txiki bat emateko. *Leuntzea* edo *smoothing* deitzen zaio prozesu honi.

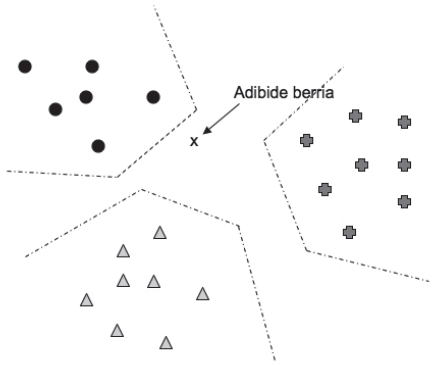
1. irudiak NBen sare bayesiarretan adierazitako eredu sinplea erakusten digu; alegia, irudiak ikasketa-ezaugarrien ($w_{-1}$, $w_{+1}$, *gunea, subj-adi, adi-obj)* arteko baldintzapeko independentzia *(S* klasearen baldintzapean) erakusten du. Demagun *«Basoa perretxikoz bete zen»* esaldian *baso* hitza desanbiguatu nahi dugula, eta ondorengo ezaugarriak erauzi ditugula: $\{w_{-1} = —$, $w_{+1} = perretxikoz$, $gunea = basoa$, $subj\text{-}adi = izan$, $adi\text{-}obj = —\}$; azken bi ezaugarriek adierazten dute esaldiko subjektuaren aditza eta objektua zein diren. Demagun ondoko probabilitateak estimatzen ditugula entrenamenduko datuetatik: $P(w_{-1} = — \mid baso_{\text{OIHAN}}) = 0{,}35$ , $P(w_{+1} = perretxikoz \setminus baso_{\text{OIHAN}}) = 0{,}76$, $P(gunea = basoa \setminus baso_{\text{OIHAN}}) = 0{,}66$, $P(subj\text{-}adi = izan \setminus baso_{\text{OIHAN}}) = 0{,}44$, $P(adi\text{-}obj = — \mid baso_{\text{OIHAN}}) = 0{,}6$. $\text{Baso}_{\text{OIHAN}}$ adieraren *a priori* probabilitatea ere kalkulatuko dugu: $P(baso_{\text{OIHAN}}) = 0{,}36$. Hortaz, adieraren azken pisua honakoa litzateke:

$$pisua(baso_{\text{OIHAN}}) = 0,36 \cdot 0,35 \cdot 0,76 \cdot 0,66 \cdot 0,44 \cdot 0,6 = 0,016$$

## 6.2. *k* Nearest Neighbours

*k* Nearest Neighbours (*k*-NN) da adibideetan oinarritzen diren algoritmoen artean ezagunena. Algoritmoak adibide berriaren adiera-sailkapena entrenamenduko *k* adibide antzekoenak (*nearest neighbours*) bilatuz egiten du. *k* adibide hauetatik gehien erabiltzen den adiera aukeratzen da (ikus 2. irudia). Kasu errazenean, entrenamendurik ez da egiten eta ikasketarako adibide guztiak memorian gordetzen dira. Ereduaren orokortzea adibide berri bat sailkatzera goazenean egiten da. Horregatik esaten zaio kasu batzuetan ikasketa alferra (*lazy learning*).

[6] http://www.inf.u-szeged.hu/ormandi/teaching/mi2/ webgunean Naïve Bayes atala aukeratuta, edota http://www.statsoft.com/textbook/stnaiveb.html webgunean.




**2. irudia.** *k*-NN algoritmoaren irudikapen grafikoa. Adibide berriak entrenamendu datu multzotik antzekoen *k* adibideak hautatu eta horien artean aukeratuko du adiera.

Bi adibideen arteko antzekotasun edo distantzia formulan datza algoritmoaren ezaugarri garrantzitsuetako bat. Hainbat distantzia kalkulatzeko era aurkitu ahal dira *k*-NNri buruzko literaturan, baina esan genezake hiru direla erabilienak: Hamming distantzia, oinarrian bi ezaugarri-bektoreek partekatutako ezaugarriak kontatzen ditu, kosinuaren formula, bi bektorek sortzen duten angeluaren araberako distantzia oinarritzen dena, eta azkenik distantzia euklidearra.

Azkenik, adiera etiketatzeko *k* antzekoenen bozak batzen dira, eta boto gehien dituen adiera aukeratuko dugu. Bozketa hainbat eratan egin daiteke. Kasu orokorrenean antzekotasunaren araberako sailkapenaren arabera bozak nolabaiteko garrantzia hartuko du. Bigarren ekuazioak adierazten duen moduan formaliza dezakegu *k*-NN algoritmoa, non $C_i$.garren antzekoenaren adiera den.

$$\arg\max_{S_j} = \sum_{i=1}^{k} \begin{cases} \frac{1}{i} & \text{baldin } C_i = S_j \\ 0 & \text{bestela} \end{cases} \qquad (2)$$

Autore batzuek diote *k*-NN algoritmoa dela HADrako aukera onena (Ng, 1997). Beste batzuek aldiz, (Daelemans *et al.*, 1999) uste dute memorian oinarritzen diren metodoak hobeak izaten direla LNP hurbilpenetan, ez dutelako inongo orokortzerik egiten eta, ondorioz, ez dutelako salbuespen gisa ezer baztertzen.

### 6.3. **Erabaki-zerrendak**

Erabaki-zerrendak edo *Decision List* direlakoak ordenatutako erregela multzoak dira (Yarowsky, 1994) eta hitzaren adiera zehazteko ezartzen dira erregela hauek, ondoko forma hartuta:

*[ezaugarri-balioa, adiera iragarpen, pisua]*

Entrenamendu-corpusean kalkulatuz pisuak esleitzen zaizkie ezaugarriei. Egiantz handieneko estimatzailearen[7] bidez *(log-likehood)* kalkulatutako pisuek ezaugarri-balioak adiera bat izateko probabilitatea adierazten dute. Erregela hauek pisuaren arabera ordenatzen dira. Beraz, adibide berri bat desanbiguatzera goazenean parekatutako ezaugarrien artean pisu gehien duenaren adiera aukeratzen dugu. $S_k$ adieraren pisua $f$ ezaugarria ematen denean ondorengo formula erabilita kalkulatzen da:

$$\arg\max_k w(s_k, f_i) = \log\left(\frac{P(s_k \mid f_i)}{\sum_{j\neq k} P(s_k \mid f_i)}\right) \quad (3)$$

Ekuazio hauetan ere, probabilitateekin lanean ari garelako, *smoothing* teknikak erabiltzen dira 0 ematen duten probabilitateak eragozteko. Konponbide sinple bat, 0,1 ordeztuz konpontzen da ezaugarriaren maiztasuna zero denean izendatzailean. Arestian esan bezala, badaude modu sofistikatuagoak.

**2. taula.** Erabaki-zerrenda baten adibidea

| Ezaugarri-balioa | Adiera pred. | Pisua |
|---|---|---|
| *ekosistema*... basoa | $\text{baso}_{oihan}$ | 4,88 |
| baso *bete ur.* | $\text{baso}_{edalontzi}$ | 3,35 |
| *edan/ADI*... baso | $\text{baso}_{edalontzi}$ | 2,77 |
| *zuhaitzetako*... baso | $\text{baso}_{oihan}$ | 1,47 |
| … | … | … |

Bigarren taulak erabaki-zerrenden adibide sinplifikatu bat erakusten digu. Lehenengo erregelak *baso* itu-hitza *ekosistema* hitza espero du testuinguruko ezkerraldean OIHAN adieraz etiketatzeko. Bigarrenak aldiz, *baso bete ur* edalontziaren adiera esleituko lioke testu zatiari.

### 6.4. **Sostengu-Bektoreen Makina**

Ingelesez *Support Vector Machine* (SVM) bezala ezagutzen den algoritmoa, eredu linealetan oinarritzen da. Oinarrian marjina handieneko hi-

[7] Corpusean gertatzen diren ezaugarrien banaketa kontutan hartuz kalkulatzen diren probabilitateak dira. Bere forma bakunenean ezaugarri batek corpusarekiko duen maiztasun erlatiboarekin pareka daiteke.

perplanoa deitzen zaion eredu lineal berezi bat kalkulatzen du. Dena den, eredu linealek dituzten arazoak konpontzeko abilezia du. Izan ere, linealak ez diren datu multzoetarako konponbide bat ematen du. Hainbat atazatan erabiltzen da eredu lineal hau.

Har dezagun, adibidez, bi klaseko datu multzo bat, zeina linealki banagarria den; beste era batera esanda, instantzien espazioan bada hiperplano bat —zuzen bat, alegia—, zeinak instantzia guztiak sailkatzen dituen zuzenaren alde batera eta bestera. (Witten & Frank, 2005) liburuko adibide batean oinarritutako 3. irudian, hobeto uler dezakegu azaltzen ari garena. Kontuan izan behar da marraz beteak dauden zirkuluak klase batekoak direla; hutsak, beste klasekoak.

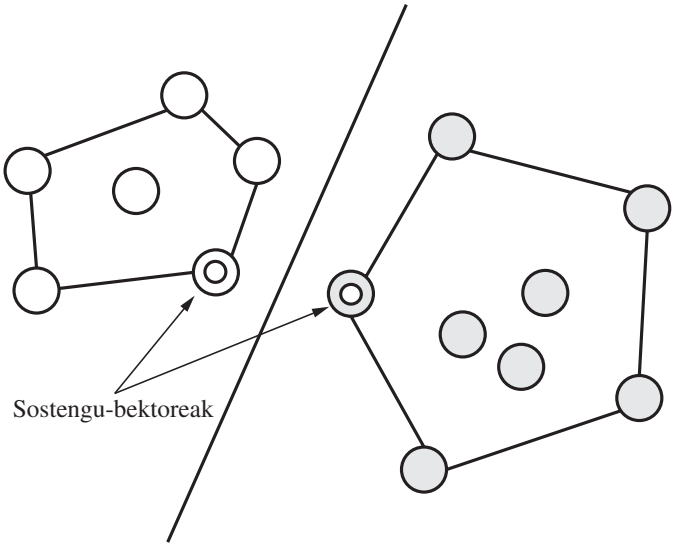
Marjina handieneko hiperplanoa da bi klaseen artean banaketa handiena ematen diguna; hots, hiperplanoko alde bateko eta besteko ikasketa-adibideak elkarrengandik urrutien jartzen dituena (Witten & Frank, 2005). Hiperplanotik gertuen dauden ikasketa-adibideei *support vector* deritze (sostengu-bektorea). Gutxienez, sostengu-bektore bana dago klase bakoitzeko. Marjina handieneko hiperplanoa eskuratu ondoren eta bi klaseetako sostengu-bektoreak lorturik, gainerako ikasketa-adibideak guztiak baztergarriak lirateke.

Horrela bada, sailkatzaile linealeko hiperplanoa bi elementu nagusiez eratua dago: (1) $w$ ezaugarri bakoitzari garrantzi jakin bat esleitzen dien bektorea, eta (2) $b$ alborapen-koefizientea, jatorriko puntuaren eta hiperplanoaren arteko distantzia zehazten duena. Bi osagaiak batuz definituko dugu sailkatzaile bitarra:

$$h(x) = \begin{cases} +1 & \text{if}(w \cdot x) + b \geq 0 \\ -1 & \text{bestela} \end{cases} \tag{4}$$

SVM algoritmoa sailkatzaile bitarra denez, HAD atazarako erabilgarria izateko klase anitzeko sailkatzaile bihurtu behar da. Era errazenean, adiera bakoitzeko sailkatzaile bitar bat eratzen da; alegia, $S_i$ adiera *versus* gainontzeko adierak egin eta konfiantza emaitza handiena duen adiera aukeratuko dugu hitza desanbiguatzeko.

Esan dugun moduan, ikasketarako datuak linealki banagarriak ez diren kasuetarako ere orokortu daiteke eskema hau, kernel deituriko funtzioen bitartez. Sarrerako ezaugarrien espazioa dimentsio handiagoko espazio bilaka daiteke funtzio ez linealen bat ezartzen badugu.




**3. irudia.** Sostengu-bektoreen makinen irudia. Hiperplanoak klaseak bitan banatzen ditu eta sostengu bektoreen distantzi berdinera kokatzen da

Praktikan, goi muga antzeko bat markatzen duen parametro bat kalkulatzea da gakoa, eta horretarako, esperimentuak egitea beste aukerarik ez dago.

### 6.5. **Konbinazioetan oinarritutako HAD sistema**

Batzuetan hainbat sailkatzaile erabiltzen dira emaitza orokorrak hobetzeko asmoz. Konbinazio modu hauek izaera ezberdinak dituzten sailkatzaileak bateratzen dituzte, hots, ezaugarri ezberdinak dituzten sailkatzaileak konbinatzen dira. Zehatzago esanda, ikuspegi ezberdin eta bereziak irudikatzen dituzten ezaugarri multzoetan ikasitako sailkatzaileen konbinazioak egiten dira. Kasu horietan sailkatzaile bakoitzaren erantzunak ahalik eta beregainenak izatea bilatzen da. Konbinazio arruntenak hauek dira: Ezaugarri lexikaletan oinarritutakoak, gramatika-ezaugarrietan oinarritutakoak eta ezaugarri semantikoetan oinarritutakoak. Azken urteotan sailkatzaileen konbinazioa gero eta gehiago erabiltzen da, bukaerako sailkatzailea hobetu eta trinkotu egiten duelako.

Hainbat modu daude sailkatzaile bakunak konbinatzeko. Hemen lau konbinazio ezaguni erreparatuko diegu: *Gehiengoaren bozketa, probabilitate-nahastearen* araberako konbinazioa, *sailkapenean oinarritutakoa*, eta *AdaBoost* konbinazioa. Dena den, beste konbinazio batzuk ere aztertu dira literaturan; horrela, *gehienezko entropiakoa* eta *bozketa pisatua* oso erabiliak dira.

Jarraian emango ditugun azalpenetarako, sailkatzaile bakunak honela izendatuko ditugu: $C_1$, $C_2$,…, $C_m$.

### 6.5.1. *Gehiengoaren bozketa*

Desanbiguatzera goazen hitza emanik (*w*), konbinazioko osagai bakoitzak *w* hitzaren adieratako bati bozka bat emango dio. Bozka gehien jasotzen duen *S* adiera aukeratuko da:

$$\hat{S} = \arg\max_{S_i \in adierak(w)} |\{j \ : \ bozka(C_j) = S_i\}|,$$

non *bozka* funtzioak sailkatzaileak aukeraturiko adiera itzultzen duen. Berdinketa balego ausaz aukeratuko litzateke adiera irabazlea.

### 6.5.2. *Probabilitate-nahastea*

Demagun sailkatzaile bakunak ($C_j$) adiera bakoitzaren konfiantza maila itzultzen duela pisu bat emanez. Horrela bada, pisuak normalizatuz adiera bakoitzaren probabilitatea lortuko dugu. Formalago esanda, $c_j$ sailkatzailea eta adiera bakoitzari ematen dion pisuak $\{pisua(C_j, S_i)\}_{i=1}^{|adierak(w)|}$ emanik, adiera bakoitzaren probabilitatea $P_{c_j}(S_i) = \frac{\{pisua(C_j, S_i)\}_{i=1}^{|adierak(w)|}}{\max_k \{pisua(C_j, S_k)\}}$ lor dezakegu. Probabilitatea maximizatzen duen adiera aukeratuko dugu erantzun bezala:

$$\hat{S} = \arg\max_{S_i \in adierak(w)} \sum_{j=1}^{m} P_{c_j}(S_i)$$

### 6.5.3. *Sailkapenean oinarritutako konbinazioa*

Demagun sailkatzaile bakunek hitz adieren sailkapen bat itzultzen dutela desanbiguatzera goazen hitz bakoitzeko (*w*). Sailkapenean oinarritutako konbinazioak, sailkapena maximizatzen duen adiera *($\hat{S}$)* aukeratuko du, sailkatzaile guztiak kontuan hartuz ($C_1$,$C_2$, …, $C_m$):

$$\hat{S} = \arg\max_{S_i \in adierak(w)} \sum_{j=1}^{m} - Rank_{c_j}(S_i),$$

non $Rank_{cj}(S_i)$ adieraren sailkapena itzultzen duen funtzioa den (1 konfiantza handieneko adierarentzat, 2 bigarrenarentzat...).

### 6.5.4. *AdaBoost*

*AdaBoost* (*adaptative boosting*) (Freund & Schapire, 1999) lanean aurkeztutako konbinazio eredu orokor bat da. Sailkatzaile ahulen konbina-

zio lineal batetik abiatuz sailkatzaile trinko bat lortzea helburu duen metodo bat da. Modu iteratiboan, sailkatzaile bakoitzaren pisua doitzen du, gaizki sailkatutako adibideak hurrengo iterazioan asmatu ahal izateko. Algoritmoak *m* iterazio egiten ditu, bat sailkatzaile bakoitzeko. Iterazio bakoitzean, gaizki sailkatutako adibideen pisuak handitzen ditu eta, horrela, hurrengo sailkatzailea adibide horietan oinarritzen da lana ondo egiteko. Iterazio bakoitzeko emaitza modura ($j \in \{1,\ldots, m\}$) $\alpha$ pisua esleitzen zaio sailkatzaileari. $\alpha_j$ harremanetan dago sailkatzaileak eginiko errore-tasarekin. Laburbilduz, sailkatzaileak ondoko moduan konbinatzen dira:

$$H(x) = sign\left(\sum_{j=1}^{m} \alpha_j C_j(x)\right),$$

non $x$ ikasketa adibidearen ezaugarri-bektorea den, $C_1,\ldots ,C_m$ sailkatzaile bakunak diren, $\alpha_j$-k sailkatzailearen garrantzia adierazten duen, eta $H$ bukaerako sailkatzaile trinkoa den.

## 7. **EBALUAZIO METODOLOGIA**

Orokorrean bi modu nagusi daude HAD sistema baten kalitatea neurtzeko. Batetik, *in vitro* ebaluazioak sistema bere horretan ebaluatzen du, hots, sistema beregaintzat hartuz, neurtzen da zenbaterainoko emaitza onak lortzen diren. Bestetik, HAD sistema aplikazio handiago baten barruan integratzen da eta honek zer laguntza dakarren neurtzen da, sistema handiaren eraginkortasuna neurtuz. Azken honi *in vivo* ebaluazioa deitzen zaio eta ebaluazio errealagoa da. Hala ere, ebaluazio mota honek dakartzan zailtasunak direla-eta (sistema konplexuak izaten ohi dira eta gehienetan integrazio modua ez da zuzenekoa), lehenengo motako ebaluazioa egin izan da egundaino.

Lan honetan *in vitro* neurrien berri emango dugu, horietan egin baita lan gehien. Lehenbizi erabiltzen diren bi ataza motak aipatuko ditugu, ondoren ebaluazio-neurriak, eta azkenik SemEval deritzon ebaluazio-kanpainan atera izan diren emaitzak.

### 7.1. **Ebaluazio-ataza motak**

Bi HAD ebaluazio-ataza mota definitu izan dira:

— *Hitzen Lagina*[8]*:* Ataza mota honetan hitz jakin batzuk aztertu nahi dira, eta hainbat esaldi ala testuinguru ematen dira hitz bakoitzeko.

[8] Lexical sample

Hortaz, desanbiguatu behar den hitz bakoitzeko eskuz etiketatutako adibide asko izan ohi dira, bai *ikasketa* egiteko, baita *ebaluazioa* egiteko ere. Testuinguru horietan azaltzen diren gainontzeko hitzak ez dira zertan desanbiguatu behar. Ikasketarako adibide asko daudenez, ataza mota honetan metodo gainbegiratuak izan ohi dira hoberenak (ikus 6. atala).

— *Hitz Guztiak*[9]*:* Hainbat testu aukeratzen dira, eta testu horietako izen, aditz, adberbio eta adjektibo guztiak desanbiguatu behar dira ataza mota honetan. Ikasketarako adibiderik ez da eskaintzen, eta hortaz gainbegiratutako sistemek arazoak izan ohi dituzte halako atazetan[10].

## 7.2. **Ebaluazio neurriak**

HAD sistema baten kalitatea neurtzeko neurri hauek erabili ohi dira: doitasuna (*precision*), estaldura (*recall*), saiakera (*coverage*) eta F-neurria (*F-score*). Hartutako erabakien zuzentasuna neurtzen du doitasunak; estaldurak, berriz, zuzenak direnetatik asmatzen direnen portzentaia ematen du; saiakerak aldiz, zenbat erabaki hartu dituen test-corpuseko adibideetatik; eta F-neurria doitasunaren eta estalduraren arteko batez besteko harmonikoa da. HAD sisteman eskuz etiketatuko osagaiak dira zuzentzat jotzen direnak. Hala, desanbiguatzaile automatiko batek osagai bat zuzen etiketatu duela esango dugu, baldin eskuz etiketatutako osagaiaren adiera berdina bada.

Demagun $N$ test-adibide ditugula, horietatik desanbiguatzaileak $C$ adibide sailkatzen dituela zuzen, eta $A$ erabaki hartzeko gaitasuna erakutsi duela. Ondorioz, matematikoki era modu honetara defini ditzakegu lau neurriok:

— Doitasuna: $P = \frac{C}{A}$

— Estaldura: $R = \frac{C}{N}$

— Saiakera: $S = \frac{A}{N}$

— F-neurria: $F_1 = \frac{2 \cdot P \cdot R}{P+R} = \frac{2C}{A+N}$

F-neurriak doitasunaren eta estalduraren arteko balantzea egiten du bien arteko batez besteko harmonikoa kalkulatuz. Neurria oso erabilgarria da %100eko saiakera ez duten sistemetan. Gainera, batez besteko aritmetikoak ez bezala, doitasun ala estaldura txikiko sistemak zigortzen ditu. Ohar bedi $P = \%100$ eta zerotik hurbil dagoen estaldura duen sistema batek %50eko etekina jasoko lukeela batez besteko aritmetikoa ($\frac{P+R}{2}$) eginez gero eta, ondorioz, ez lukeela errealitatea ondo islatuko.

---

[9] *All-words*

[10] Ingeleserako SemCor izeneko corpusa dagoen arren, tamaina mugatua du, eta hitz batzuk 2 edo 3 aldiz bakarrik agertu ahal dira.

## 8. **LEHIAKETAK**

HAD sistemak beraien artean konparatzea oso zaila da, hiztegi, adiera kopuru eta definizio desberdinak darabiltzatenean, edota desanbiguazio prozesurako hautatutako hitz desberdinak, testu desberdinak, eta, ondorioz, testuinguru desberdinak erabiltzen direnean. Aldaera hauek direla-medio, oso zaila izan ohi zen ondorio orokorrak ateratzea eta, are gehiago, hainbat sistemen arteko erkaketak egitea. Esparru orokor bat falta zelarik ikerketa lanek ebaluazio propioa baldintza jakinetan egiten zuten, baina zaila zen orduko lanetatik ondorio argirik ateratzea.

Horri guztiari aurre egiteko, 1998tik aurrera eta hiru urtez behin Senseval[11] (orain SemEval) lehiaketak antolatzen dira. Lehiaketa hauen helburu nagusia sistema eta tekniken arteko konparazio objektiboa egitea da. Hainbat ataza sortzen dituzte, hainbat hizkuntzatarako HAD atazatik harago, baina denak semantikaren prozesatze automatikoa helburu izanda. Ondorioz, ikerketak emaniko hainbat teknika baldintza berdinetan ebaluatzen dira, haien arteko konparazio zuzena ahalbideratuta.

Gaur egun dira Senseval ebaluazioak erreferentzia bihurtu eta etorkizunerako bide eta helburu berriak zehazteko balio dute. Bestela esanda, artearen egoera zehazteko balio dute lehiaketa hauek. Horrez gain, balio erantsia dute, gauza baitira periodikoki esperimentuak eginez datu multzo berriak eskaintzeko. Izan ere, datu hauek ikerketa zuzendua egitea ahalbideratzen dute.

**3. taula.** Euskarazko Hitz Lagina atazan sistemek lortutako emaitza ofizialak, estalduraren arabera taxutuak

| Sailkapena | Sistema | Sailkatzaile mota | Doitasuna | Estaldura | Saiakera |
|---|---|---|---|---|---|
| 1 | basque-swat-hk-bo | Adaboost | 71.1 | 70.4 | 99.04 |
| 2 | BCU-Basque-svm | SVM | 69.9 | 69.9 | 100.00 |
| 3 | BCU-Basque-Comb | Konbinazioa | 69.5 | 69.5 | 100.00 |
| 4 | swat-hk-basque | Konbinazioa | 67.0 | 67.0 | 100.00 |
| 5 | IRST-Kernels-bas | SVM | 65.5 | 65.5 | 100.00 |
| 6 | swat-basque | Konbinazioa | 64.6 | 64.6 | 100.00 |
| 7 | Duluth-BLSS | Konbinazioa | 60.8 | 60.8 | 100.00 |
| 8 | UMD-SST1 | SVM | 65.6 | 58.7 | 89.42 |
| 9 | Hitzaren Adiera Usuena | - | 55.8 | 55.8 | 100.00 |

[11] http://senseval.org

Orain arte 1998 eta 2007 bitartean 4 Senseval/SemEval lehiaketa antolatu dira. Hainbat hizkuntza jorratu izan dira, euskara barne; ingelesezkoa atazak izan dira parte-hartzaile gehien lortu dituztenak, baliabide aberatsagoak zeudelako. Bestalde, espero zen bezala, erakutsi izan da adiera desanbiguazio teknikek jokaera bera azaltzen dutela hizkuntza ezberdinetan, eta teknika mota berdinak direla arrakastatsuak hizkuntza ezberdinetan. Lan honetan euskararako antolatu ziren bi lehiaketetatik azkena eta ingelesera ko antolatu den azkena azalduko ditugu.

### 8.1. **Senseval-3: euskararako atala**

2004ko Senseval edizioa Bartzelonan antolatu zen, ACL[12] (*Association of Computational Linguistics*) kongresuaren barne. 14 ataza antolatu ziren, eta 55 taldek hartu zuten parte, 160 sistema ezberdin zituztela. Atazen artean arestian azaldutako ohiko HAD atazak aurkeztu ziren, alegia, *Hitzen Lagina* eta *Hitz Guztiak* motako atazak.

Beste hizkuntzen artean euskararako ataza antolatu zen (Agirre, Aldabe, Lersundi, Martínez, Pociello, & Uria, 2004). Adierak Euskal WordNet delakotik hartu ziren, eta testuak Egunkariako corpusetik. Hitzen Lagina izeneko ataza da, eta hortaz 40 izen, aditz eta adjektibo hautatu ziren, eta horietako bakoitzerako 100 adibide etiketatu ziren, bi heren entrenatzeko eta gainontzeko herena ebaluaziorako zirelarik. Antolatzaileek automatikoki analizatu zituzten adibideak, parte hartzaile guztiek lema, kategoria eta deklinabide atzizkiak erabil zitzaten. Bost taldek hartu zuten parte, bostak sistema gainbegiratuak zituztelarik, eta bostek erabili zituztelarik emandako analisiak. 3. taulak erakusten ditu emaitzak, hitzaren adiera usuena barne. Denek lortu zituzten hitzaren adiera usuena baino emaitza hobeagoak, eta emaitza hoberenak Adaboost eta SVM teknikak erabiliaz lortu ziren.

### 8.2. **SemEval-1: ingelesearako ataza**

Senseval azken edizioan izena SemEval izatera aldatzea erabaki zen. 2007ko SemEval edizioa Pragan antolatu zen, hau ere ACL kongresuaren barne. Guztira 18 ataza gauzatu ziren, l00etik gorako parte-hartzaile talde batek 125 sistema aurkeztu zituzten.

Gure atazari dagokionez, (17. ataza) Wall Street Journal eta Brown Corpusetatik ateratako testuak etiketatuz hiru azpi ataza antolatu zituzten: i) WordNetko adierez etiketatutako *Hitz Guztiak* ataza, ii) *Ontoneseko*[13] adiera-definizio orokorragoez etiketatuko Hitzen Lagina ataza eta iii) Rol Semantikoen Etiketatzeko buruzko ataza. Guk geuk, lehenengo bi azpitazei erreparatuko diegu.

---

[12] http://www.aclweb.org/

[13] WordNeteko adierak multzokatuz sortutako adiera definizio berriak dira.

**4. taula.** *Hitzen Lagina* atazako emaitza ofizialak

| Sailkapena | Sistema | Sailkatzaile mota | F-neurria |
|---|---|---|---|
| 1 | NUS-ML | SVM | 88.7±1.2 |
| 2 | UBC-ALM | kNN | 86.9±1.2 |
| 3 | I2R | gainbegiratua | 86.4±1.2 |
| 4 | USP-IBM-2 | SVM | 85.7±1.2 |
| 5 | USP-IBM-1 | ILP | 85.1±1.2 |
| 5 | KU | erdi-gainbegiratua | 85.1±1.2 |
| 6 | OE | NB, SVM | 83.8±1.2 |
| 7 | VUTBR | NB | 80.3±1.2 |
| 8 | UBC-ZAS | kNN | 79.9±1.2 |
| 9 | ITC-irst | SVM | 79.6±1.2 |
| 10 | *Hitzaren Adiera Usuena* | - | 78.0±1.2 |
| 11 | USYD | SVM | 74.3±1.2 |
| 12 | UMND1 | gainbegiratu gabea | 53.8±1.2 |
| 13 | Tor | gainbegiratu gabea | 52.1±1.2 |

*Hitzen Lagina* atazari dagokionez, 100 lema aukeratu zituzten (65 aditz eta 35 izen), polisemia maila eta agerpen kopuruaren arabera aukeratuta. Parte hartu zuten sistemen emaitzak 4. taulan adierazita daude; bertan, lerro bakoitzean F-neurriaren arabera ordenatuta sistema bakoitzaren asmatze-tasa azaltzen da. Sistema guztien saiakera %100 izan zenez, doitasun- eta estaldura-neurriek balio bera dute, F-neurriak bezalaxe. F-neurriari %95ko konfidantza-tartea gehitzen zaio. Tarte honek bi sistemen arteko diferentzia estatistikoki esanguratsua ote den esaten digu.

Taulak erakusten duenez, 12 sistema ebaluatu ziren *Hitzen Lagina* atazarako (gogoratu bedi *hitzaren adiera usuena* dela oinarrizko sistema). Hitz-adieren definizio orokorragoak izateak, beste era batean esanda, errazago bereizgarriak direnak, sistemek asmatze-tasa handiak lortzea ahalbideratu zuen, agerian utziz adieren definizioak azken emaitzetan duten eragina.

Sistemen sailkapenari dagokionez, taulako goi aldean ageri diren sistema gehienak kerneletan oinarritutako SVM eta *k*-NN sailkatzaileak dira (ikus bedi 6.4. eta 6.2. atalak hurrenez hurren). Emaitzek erakusten dute SVM sailkatzaileak direla onenak, ikasketarako nahikoa adibide daudenean. Ezaugarriei dagokienez, sistema gehienek 4. atalean azaldutako ezaugarri lokal, topikal, eta sintaktikoetan oinarritzen dira. Horrez gain, irabazleek (Cai *et al*., 2007) *Latent Dirichlet Allocation* (LDA) metodo probabilistikoa erabiltzen dute, entrenamenduko ezaugarriei garrantzia-

ren araberako pisua emateko. Bigarrenekoek (Agirre & Lopez de Lacalle, 2007) aldiz *k*-NN sailkatzaileen konbinazioa egitea proposatzen dute. Lau ezaugarri multzotan entrenatutako *k*-NNak konbinatzen dituzte: ezaugarri lokalak, topikalak, sintaktikoak eta balio singularretan (*Singular Value Decomposition*) deskonposatzetik lortutako ezaugarriak. Aipatzekoa da sistema gehienek garaitzen dutela adiera usuena erabiliz etiketatzen duen sistema bakuna. Gainbegiratua den sailkatzaile bakarrak eta gainbegiratua ez diren bik ez dute muga hori gainditzen.

**5. taula.** *Hitz Guztiak* atazan sistemek lortutako emaitza ofizialak

| Sailkapena | Sistema | Sailkatzaile mota | F-neurria |
|---|---|---|---|
| 1 | PNNL | MaxEnt | 59.1±4.5 |
| 2 | NUS-PT | SVM | 58.7±4.5 |
| 3 | UNT-Yahoo | Memory-based | 58.3±4.5 |
| 4 | NUS-ML | naive Bayes | 57.6±4.5 |
| 5 | UBC-ALM | kNN | 54.4±4.5 |
| 6 | UBC-UMB-2 | kNN | 54.0±4.5 |
| 7 | PU-BCD | Exponential Model | 53.9±4.5 |
| 8 | RACAI | gainbegiratu gabea | 52.7±4.5 |
| 9 | *Hitzaren Adiera Usuena* | - | 51.4±4.5 |
| 10 | UPV-WSD | gainbegiratu gabea | 46.9±4.5 |
| 11 | JU-SKNSB | gainbegiratu gabea | 40.2±4.5 |
| 12 | UBC-UMB-1 | gainbegiratu gabea | 39.9±4.5 |
| 14 | tkb-uo | gainbegiratu gabea | 32.5±4.5 |
| 15 | PUTOP | gainbegiratu gabea | 13.2±4.5 |

*Hitz Guztiak* atazari dagokionez, Wall Street Journal corpusetik 3.500 hitzeko testu zatian 465 lema etiketatu ziren test-corpusa sortzeko. 465 agerpenak WordNeteko adierez etiketatu ziren, eta adiera xeheak izateak zailtasuna gehitzen zion atazari. *Hitzen Lagina* atazan ez bezala, ataza honetan ez da ikasketarako corpusik ematen, eta emaitzek agerian uzten dutenez, ataza errealistagoa izanik, askoz ere zailagoa da.

Azpiataza honetan 14 sistema aurkeztu ziren, 5. taulak erakutsi bezala. Orokorrean emaitzak aurreko azpiatazarekin alderatuz oso murritzak dira. Bi faktorek eragiten dute nagusiki emaitza apalak izatea: batetik adieren definizio finek zaila egiten dute adierak bereiztea eta, bestetik, entrenamendurako adibide gutxi izanda (hitz bakoitzeko adibide gutxi batzuk besterik ezin dira SemCorretik atera), zaila izaten da ondo egingo duen sailkatzailerik induzitzea.

Taulari begiratuz gero, entropia maximoan oinarritzen den sailkatzaile probabilistikoa da emaitza hoberena lortzen duena. Ondoren, SVM, NB, *k*-NN motako sailkatzaileak dira emaitza hoberenak lortzen dituztenak. Edozein modutan ere, sailkatzaile gainbegiratuek, gainbegiratu gabeek baino F-neurri altuagoa lortzen dute, entrenatzeko adibide faltaren eragozpena izanda ere.

Ataza honetan zaila da adiera usuenaz etiketatzearen muga hobetzea, eta gainbegiratu guztiek hobetzen duten arren, gainbegiratu gabea den sistema bakarrak (RACAI) hobetzen du muga hori. Honek agerian uzten du HAD atazak zailtasuna duela mundu errealean aplikatu ahal izateko.

Bi kontutan laburbil daiteke bi taula hauetatik ondoriozta daitekeena:

— HAD egiteko faktore garrantzitsuak bi dira: 1) Adieraren definizio burutsu batek emaitza onak lortzeko bidea erakusten du; alegia, %80-%90 bitarteko doitasuna lortzea posible dela erakutsi dute *Hitzen Lagina* atazako ematzek. Eta 2) Sailkatzaile gainbegiratuetan lan eginez gero, beharrezkoa da ahalik eta ikasketarako adibide gehien edukitzea emaitza ona bermatzeko.
— Emaitzek agerian utzi dute atazaren zailtasuna: Horixe da egoera erreala simulatzen duen *Hitz Guztiak* atazan lortutako emaitza apaletik atera daitekeen ondorio nagusia.

## 9. **ONDORIOAK**

Artikulu honetan hitzen adiera-desanbiguazioaren sarrera orokor bat egin dugu. Erakutsi dugu hitzen adiera-desanbiguazioa ataza zaila dela, hizkuntzaren konplexutasunari aurre egin eta testu hutsetik egitura semantikoa antzeman behar duelako. Hainbat ikasketa modu azaldu ditugu, baina batez ere, ikasketa automatiko gainbegiratuko zenbait sailkatzaile aztertu dira artikuluan. Tradizio handiko alorra denez (1950ean jar dezakegu HAD atazaren hasiera), gaiari buruzko literatura zabala aurki dezakegu, (Ide & Véronis, 1998), (Navigli, 2009), eta (Agirre & Edmonds, 2006) izanik atazaren ikuspegi osatu bat ematen duten lanak.

Artikuluan zehar esan dugun moduan, HADn zailtasuna adiera-fintasunean datza. Zenbait lanek erakutsi dute %95eko zuzentasuna lor dezakegula adiera orokorrekin lan eginez gero, adibidez hitz homografoen desanbiguazioa eginez gero. Baina adiera xeheagoak egin ahala, ataza geroz eta zailagoa bilakatzen da; eta adiera finegiak badira, etiketatzaileen arteko adostasuna ere esanguratsuki jaisten da.

Emaitzei erreparatuz gero, erraz esan dezakegu sistema gainbegiratuak direla HAD egiteko teknika egokienak. Hala ere, datu multzo etiketatuak

behar dituzte, eta sorkuntza oso garestia da; beraz, badakigu ez dela HAD egiteko estrategia egokiena. Izan ere, hizkuntza-aldaketak, domeinu-aldaketak edota hiztegi-aldaketak etiketatze-prozesua berriz egitera behartuko gintuzkete. Bestetik, badaude ezagutza-baseetan oinarritzen diren teknikak. Horrela lortzen diren emaitzak gainbegiratuenenak bezain onak ez diren arren, datu etiketatuen menpe ez daude eta WordNeten bertsio finduak eta berezituak gero eta ugariagoak direlarik, hauxe bide da datozen urteotako ikerlerro nagusia.

Azken urteotan ikerketak HAD atazaren erabilgarritasunean lan egiten hasi dira, hau da, *in vivo* ebaluazioak geroz eta ugariagoak dira. Azken lan interesgarri batzuek, aztertu dute besteak beste informazio berreskurapenean edo itzulpen automakikoan informazio semantikoa lagungarri ote den eta nola lagun dezakeen. Dena den, *in vitro* ebaluazioak beharrezkotzat jotzen dira oraindik, algoritmo berrien portaera hobeto ulertzen laguntzen dutelako.

*Hitz Guztiak* atazaren balioa garbia da baina, garatu egin beharko litzateke estaldura txikiko baxuko, baina doitasun handiko desanbiguatzailea zenbait aplikazioetan integratzeko. Adibidez, web semantikoaren ikuspegia abian jar daiteke, bai domeinu itxi baterako eta bai domeinuarekiko interesgarriak diren hitzen desanbiguatzaile espezializatuak sortuz gero. Era honetara webean hartueman semantiko eta automatiko partziala egiteko aukera sortuko genuke. HAD aplikazioen ikerketak bide hori urratu beharko lukeela uste dugu.

Euskarari dagokionez, IXA taldean ahalegin berezia egin da alor honetan. Euskal WordNet (Agirre et al., 2006b)[14] ontologia eta Euskal Semcor adierez etiketatutako corpusak dira adiera-desanbiguaziorako sortu diren baliabide garrantzitsuenak (Agirre *et al*., 2006a)[15]. Ebaluaziorako Senseval-2 (Agirre *et al*., 2001) eta Senseval-3 (Agirre et al., 2004) lehiaketetan euskararako ataza antolatu zen, eta horri esker euskararako HAD sistemak beste hizkuntzetako sistemekin alderatu daitezke. Kontuan hartu behar da bai desanbiguaziorako ontologia eta bai entrenamendurako corpusen sorkuntza oso lan garesti eta saiatuak direla, eta urteak daramatzagula horietan lanean. Bestalde, HAD teknika asko garatu dira taldean, eta horien lekuko dira HAD sistema gainbegiratu baten demoa[16], eta ezagutzan oinarritutako HAD egiteko UKB softwareak (Agirre & Soroa, 2009)[17].

[14] http://http://ixa2.si.ehu.es/mcr/wei.htm
[15] http://http://ixa2.si.ehu.es/mcr/wei.htm
[16] http://ixa3.si.ehu.es/wsd-demo/
[17] http://ixa2.si.ehu.es/ukb/

## REFERENCES

AGIRRE, E., ALDABE, I., LERSUNDI, M., MARTÍNEZ, D., POCIELLO, E., & URIA, L. (2004). «The Basque lexical-sample task». In *Proceedings ofthe 3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL)*, pp. 1-4.

AGIRRE, E., ALDEZABAL, L, ETXEBERRIA, J., IRUSKIETA, M., IZAGIRRE, E., MENDIZABAL, K., & POCIELLO, E. (2006a). «A methodology for the joint development of the Basque WordNet and Semcor». In *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC).*

AGIRRE, E., ALDEZABAL, I., & POCIELLO, E. (2006b).« Euskararako ezagutzabase lexiko-semantikoaren eredu-hautaketa eta garapena: EuskalWordNet». *GOGOA*, 1(6), 237-266.

AGIRRE, E., & EDMONDS, P. (Eds.). (2006). *Word Sense Disambiguation. Algorithms and Applicaüon, Vol. 33 of Text, Speech and Language Technology.* Springer, P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

AGIRRE, E., GARCÍA, E., LERSUNDI, M., MARTÍNEZ, D., & POCIELLO, E. (2001). «The Basque task: did systems perform in the upperbound?». In *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL'2001/EACL'2001*, pp. 9-12.

AGIRRE, E., & LÓPEZ DE LACALLE, O. (2007). «UBC-ALM: Combining k-NN with SVD for WSD». In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 342-345, Prague, Czech Republic. Association for Computational Linguistics.

AGIRRE, E., & MARTÍNEZ, D. (2001). «Knowledge sources for Word Sense Disambiguation». In Matousek, V., Mautner, P., Moucek, R., & Tauser, K. (Eds.), *Proceedings of the Fourth International Conference TSD,* Published in the Springer Verlag Lecture Notes in Computer Science series. Copyright Springer-Verlag.

AGIRRE, E., RIGAU, G., SOROA, A., VOSSEN, P., & BOSMA, W. (2010). «A full Knowledge Cycle for Semantic Interoperability». In *Proceedings ofthe 5th Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation, in conjunction with the Second International Conference on Global Interoperability for Language Resources (ICGL 10).*

AGIRRE, E., & SOROA, A. (2009). «Personalizing pagerank for word sense disambiguation». In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, Athens, Greece.

BERNERS-LEE, T., HENDLER, J., & LASSILA, O. (2001). «The semantic web». *Scientific American*, 284(5), 34-43.

BLOEHDORN, S., & HOTHO, A. (2004). «Text classification by boosting weak learners based on terms and concepts». In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pp. 331-334.

CAI, J. F., LEE, W. S., & TEH, Y. W. (2007). «NUS-ML:Improving Word Sense Disambiguation Using Topic Features». In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 249-252, Prague, Czech Republic. Association for Computational Linguistics.

CARPUAT, M., & WU, D. (2005). «Evaluating the Word Sense Disambiguation Performance of Statistical Machine Translation». In *Second International Joint Conference on Natural Language Processing (IJCNLP-2005).*

CHAN, Y. S., NG, H. T., & CHIANG, D. (2007). «Word sense disambiguation improves statistical machine translation». In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 33-40, Prague, Czech Republic. Association for Computational Linguistics.

CLOUGH, P., & STEVENSON, M. (2004). «Cross-language information retrieval using eurowordnet and word sense disambiguation». In *Advances in Information Retrieval, 26th European Conference on IR Research (ECIR),* pp. 327-337, Sunderland, UK.

DAELEMANS, W., VAN DEN BOSCH, A., & ZAVREL, J. (1999). «Forgeting exceptions is harmful in language learning». *Machine Learning*, 34, 11-41.

ESCUDERO, C, MÁRQUEZ, L., & RIGAU, G. (2000). «An empirical study of the domain depen-dence of supervised word sense didanbiguation systems». In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Laxge Corpora, EMNLP/VLC,* Hong Kong.

FREUND, Y., & SCHAPIRE, R. (1999). «A short introduction to boosting». *Journal of Japanese Social Artificial Intelligence* 14, 14i 771-780.

GALE, W. A., CHURCH, K., & YAROWSKY, D. (1992). «A method for disambiguating word senses in a corpus». *Comput. Human.*, 26, 415-439.

IDE, N., & VÉRONIS, J. (1998). «Word Sense Disambiguation: The state of the art». *Computational Linguistics*, 24(1), 1-40.

KILGARRIFF, A., REDDY, S., POMIKÁLEK, J., & AVINESH, P. V. S. (2010). «A corpus factory for many languages». In *Proceedings of the 7th International Conference on Language Resources and Evaluations (LREC).*

KILLGARIFF, A., & TUGWELL, D. (2004). Sketching words. In Corréard, M.-H. (Ed.), *Lexicography and Natural Language Processing: A Festschrift in Honour of B. T. S. Atkins,* chap. Sketching words, pp. 125-137. EURALEX.

MILLER, G., LEACOCK, C, TENGI, R., & R.BUNKER (1993). «A Semantic Concordance». In *Proceedings of the ARPA Human Language Technology Workshop*. Distributed as Human Language Technology by San Mateo, CA: Morgan Kaufmann Publishers., pp. 303-308, Princeton, NJ.

NAVIGLI, R. (2009). «Word sense disambiguation: A survey». *ACM Computing Surveys,* 42(2).

NG, H. T. (1997). «Exemplar-Based Word Sense Disambiguation: Some Recent Improvements». In Cardie, C, & Weischedel, R. (Eds.), *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 208-213. Association for Computational Linguistics, Somerset, New Jersey.

OTEGI, A., AGIRRE, E., & RIGAU, G. (2008). «IXA at CLEF 2008 Robust-WSD Task: using Word Sense Disambiguation for (Cross Lingual) Information Retrieval». In *Working Notes of the Cross-Lingual Evaluation Forum.*

PEDERSEN, T. (2001). «A Decisión Tree of Bigrams is an Accurate Predictor of Word Sense». In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, Pittsburgh, PA.

TURING, A. M. (1950). «Computing machinery and intelligence». *Mind*, 46, 443-460.
VOSSEN, P., RIGAU, G., ALEGRÍA, L, AGIRRE, E., FARWELL, D., & FUENTES, M. (2006). «Meaningful results for information retrieval in the MEANING project». In *Proceedings of the 3rd Global Wordnet Conference*, Jeju Island, Korea.
WEAVER, W. (1949). *Translation,* pp. 15-23. John Wiley & Sons.
WITTEN, L, & FRANK, E. (2005). *Data Mining. Practical Machine Learning Tools and Techniques with Java Inplementations*. Morgan Kaufmann Publishers.
YAROWSKY, D. (1994). «Decision Lists for lexical ambiguity resolution: Application to accent restoration in Spanish and Frenen». In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 88-95.
YAROWSKY, D., & FLORIAN, R. (2002). «Evaluating sense disambiguation across diverse parameter spaces». *Natural Language Engineering,* 8(4), 293-310.