Euskal Herriko Unibertsitatea / Universidad del País Vasco


Lengoaia eta Sistema Informatikoak Saila

# Azaleko sintaxiaren tratamendua ikasketa automatikoko tekniken bidez: euskarako kateen eta perpausen identifikazioa eta bere erabilera koma-zuzentzaile batean

**Bertol Arrieta Cortajarenak**

Informatikan Doktore titulua eskuratzeko aurkezturiko

Tesia

Donostia, 2010eko maiatza.

Euskal Herriko Unibertsitatea / Universidad del País Vasco


Lengoaia eta Sistema Informatikoak Saila

# Azaleko sintaxiaren tratamendua ikasketa automatikoko tekniken bidez: euskarako kateen eta perpausen identifikazioa eta bere erabilera koma-zuzentzaile batean

Bertol Arrieta Cortajarenak Iñaki Alegriaren eta Arantza Díaz de Ilarrazaren zuzendaritzapean egindako tesiaren txostena, Euskal Herriko Unibertsitatean Informatikan Doktore titulua eskuratzeko aurkeztua.

Donostia, 2010eko maiatza.

*Lan honetarako, Eusko Jaurlaritzako Hezkuntza, Unibertsitate eta Ikerketa sailak emandako doktoretza aurreko beka bat izan nuen, ikerkuntza-proiektu jakin bati lotutakoa, zazpi hilabetez, 2001. urtean.*

*The SGI/IZO-SGIker UPV/EHU (supported by the Development and Innovation - Fondo Social Europeo, MCyT and Basque Government) is gratefully acknowledged for generous allocation of computational resources.*

*La coma, esa puerta giratoria del pensamiento.*

Julio Cortázar

*I was working on the proof of one of my poems all the morning, and took out a comma. In the afternoon, I put it back again.*

Oscar Wilde

*Today, I learned, the comma, this a comma (,) a period, with a tail, Miss Kinnian, says its important, because, it makes writing, better, she said, somebody, could lose, a lot of money, if a comma, isnt, in the, right place, I dont have, any money, and I dont see, how a comma, keeps you, from losing it.*

*But she says, everybody, uses commas, so Ill use, Them too.*

Flowers for Algernoon (Daniel Keyes)

*Puntuazio markak euli-kakak bezalakoak dira: txikiak, beltz-beltzak, ezdeusak. Ez diegu garrantzirik ematen. Baina punta-puntako idazle, akademiko, euskaltzain asko dabil bazterretan puntuazio marken azterketa gainditu ezinik: puntuazio markena dugu ikasgairik zailetan zailena, puntuazio marka ondo erabiliek morfosintaxiaren ezagutza sakona islatzen dute.*

Berria (2009/10/16). Anjel Lertxundi

*Eliri (eta bidean datorrenari)*

# Eskerrik asko!

Tesi hau egin ahal izateko, jende askoren laguntza izan dut, eta hauei guztiei eskerrak eman nahi nizkieke:

- IXA taldeko kide guztiei, lan hau aurrera eramateko eskaini didazuen laguntza guztiarengatik.
- Zuzendariei —Arantzari eta Iñakiri—, berez astuna dena hain astuna izan ez zedin egindako ahaleginarengatik.
- A Lluís Màrquez i sobre tot a Xavi Carreras, per respondre amb molta paciència a totes les meves preguntes.
- Nereari, analizatzaile morfosintaktiko komagabea eta antzeko gauza arraroak sortzen eta ebaluatzen laguntzeagatik.
- Taldeko hizkuntzalari guztiei, eta, batez ere, Larraitzi, Izaskuni eta Maxuxi nire zalantza guztiak argitzen laguntzeagatik, eta makinari, ikas dezan, bazka emateagatik. Ikasiko ez du, bada, ondo, modu horretan!
- Edurne Aldasorori, hau, hori eta bestea etiketatzen hartutako lanarengatik.
- Koldori, Olatz Arregiri eta Nereari, txosten hau irakurtzen eta zuzentzen hartutako lanarengatik.
- Oierri, *perl*ekin emandako laguntzarengatik, eta hainbat burutazioren eta arazoren aurrean bere ikuspegi gardena erakusteagatik.
- Kepari eta Xabier Artolari, nire klaseak arintzeko egindako ahaleginarengatik.

- Kortako *abelkide* ohiei, aldapatsua izan den azken urtean nire umore aldaketak jasateagatik eta emandako laguntzarengatik.

- Kikeri, Estherri eta Amaiari; beti laguntzen eta beti laguntzeko prest.

- *Hirekin* azpitaldeko guztiei; hau zuena ere bada.

- Aingeruri, azken hilabete hauetan nire agoniak agoantatzeagatik.

- *Latex*arekin —lanean— lagundu didazuenei (Aitor Soroa, Eli, Itziar, Gorka, Oier...).

- *Gym* taldetxoari, tesiak jarritako kartolak pixka bat apartatzen laguntzeagatik.

- Arantxa Otegiri, Oscar Cermenori eta Iara Jimenezi hainbat aurrerapauso ematen laguntzeagatik.

- Juan Garziari, Igone Zabalari, Juan Carlos Odriozolari eta Joxerra Etxeberriari; puntuazioari buruzko euren iritziak azaltzeko patxada hartzeagatik.

- Txema Mercerori eta Edu Ogandori, tesi honen azken partean superkonputagailuarekin emandako superlaguntzarengatik.

- Lagunei, garagardo baten itzala eskaintzeagatik, behar nuen guztietan. Eta asko berotu du eguzkiak...

- Senide guztiei, aurrera jarraitzeko bultzada txiki horiengatik guztiengatik.

- Gurasoei, Ierari eta Liberi, emandako animo guztiengatik, eta beti alboan sentitu zaituztedalako.

- Eliri: bide luze honetan, egunero-egunero, eman dizkidazun animoengatik; azken txanpa luzean, nire lana arinago egitearren, bioi zegozkigun lan asko zure gain hartzeagatik; nire umore txarreko momentuetan horrenbeste pazientzia izateagatik; denbora oparitzeagatik; erraztasunak emateagatik; behar zintudan guztietan nirekin egoteagatik. Eskerrik asko.

Eskerrik asko denoi!

# Laburtzapenak

*Euskaraz:*

**EDBL:** Euskararen Datu-Base Lexikala
**EPEC:** Euskararen Prozesamendurako Erreferentzia-Corpusa
**EusWN:** Euskal WordNet
**HP:** Hizkuntzaren Prozesamendua

*Ingelesez:*

**ACL:** The Association for Computational Linguistics
**BNC:** British National Corpus
**CFG:** Context Free Grammar edo testuingururik gabeko gramatikak
**CG:** Constraint Grammar edo Murriztapen Gramatika
**CoNLL:** Conference on Natural Language Learning
**CRF:** Conditional Random Fields
**FR-P:** Filtering and Ranking with Perceptrons
**FST:** Finite State Transducers
**HMM:** Hidden Markov Model edo Markoven eredu ezkutuak
**LSA:** Latent Semantic Analysis edo ezkutuko semantikaren analisia
**MBL:** Memory Based Learning edo memorietan oinarritutako ikasketa
**ML:** Machine Learning edo ikasketa automatikoa
**NAACL:** The North American Association for Computational Linguistics
**SVM:** Support Vector Machines
**TBL:** Transformation Based Learning edo transformazioan oinarritutako ikasketa
**WN:** WordNet
**WSD:** Word Sense Disambiguation edo hitzen adiera-desanbiguazioa

# Gaien aurkibidea

**Eskerrik asko!** **ix**

**Laburtzapenak** **xi**

**Aurkibidea** **xiii**

**Irudien zerrenda** **xix**

**Taulen zerrenda** **xxi**

**I Tesi-lanaren aurkezpen orokorra** **1**

I.1 Gaiaren motibazioa eta kokapena . . . . . 1

I.1.1 *Nork jan du aita?* . . . . . 1

I.1.2 Gaiaren kokapena . . . . . 3

I.1.2.1 Hizkuntzaren Prozesamenduko bi hurbilpen nagusiak . . . . . 3

I.1.2.2 Testuingurua . . . . . 4

I.2 Helburuak . . . . . 6

I.3 Tesi-txostenaren eskema . . . . . 8

I.4 Tesiarekin lotutako argitalpenak . . . . . 8

I.4.1 Tesiari hertsiki lotutakoak . . . . . 8

I.4.2 HParen alorrean egindako gainontzekoak . . . . . 10

**II Ikasketa automatikoko teknikak azaleko sintaxiaren tratamenduan eta erroreen detekzioan** **13**

II.1 Ikasketa automatikoa HPan . . . . . 14

II.1.1 Sarrera . . . . . 14

II.1.2 Ikasketa automatikoaren funtsa . . . . . 16

II.1.3 Ikasketa automatikoaren arazoak HPan . . . . . 17

II.1.4 Ikasketa automatikoko tekniken sailkapena . . . . 20
II.1.5 Oinarrizko ikasketa-algoritmoak . . . . . 21
II.1.5.1 *Naive Bayes* . . . . . 21
II.1.5.2 Erabaki-zuhaitzak . . . . . 22
II.1.5.3 *Pertzeptroi*ak . . . . . 24
II.1.5.4 *Support Vector Machines* edo *sostengu-bektoreen makinak* . . . . . 25
II.2 Sintaxiaren tratamendu automatikoa . . . . . 27
II.2.1 Sintaxiaren tratamendu automatikoaren joera nagusiak . . . . . 28
II.2.2 Sintaxiaren tratamendu automatikoa ikasketa automatikoko tekniken bidez . . . . . 32
II.2.3 Sintaxiaren tratamendu automatikoa IXA taldean 34
II.2.3.1 Analisi-katea . . . . . 34
II.2.3.2 EPEC: euskararen prozesamendurako erreferentzia-corpusa . . . . . 37
II.3 Erroreen detekzio automatikoa . . . . . 40
II.3.1 Erroreak eta desbideratzeak . . . . . 40
II.3.2 Erroreen detekziorako teknikak . . . . . 42
II.3.2.1 Hizkuntza-ezagutzan oinarritutako teknikak . 43
II.3.2.2 Corpusetan oinarritutako teknikak . . . . . 44
II.3.3 XUXENg: euskarako gramatika-zuzentzaile automatikoa . . . . . 53
II.3.3.1 Euskarako corpus erroreduna . . . . . 55
II.3.3.2 Euskarako erroreen sailkapena . . . . . 57
II.3.3.3 Euskarako erroreen datu-baseak . . . . . 60
II.3.3.4 Euskarako erroreen detekzioa ikasketa automatikoa erabiliz . . . . . 61
II.4 Ondorioak . . . . . 62

**III Azaleko sintaxiaren tratamendua euskaraz: kateen eta perpausen identifikazioa 65**
III.1 Sarrera . . . . . 67
III.2 Testuingurua . . . . . 70
III.2.1 *Hitz multzo*ak: definizio formal bat kateen eta perpausen deskribapenerako . . . . . 70
III.2.2 Kateak . . . . . 73
III.2.2.1 Kateak: hurbilpen linguistikoa . . . . . 73

III.2.2.2 Euskarako kateen identifikazioa hurbilpen linguistikoa erabiliz . . . 74
III.2.2.3 Kateen identifikazioa ikasketa automatikoa erabiliz . . . 78
III.2.3 Perpausak . . . 80
III.2.3.1 Perpausak: hurbilpen linguistikoa . . . 81
III.2.3.2 Euskarako perpausen identifikazioa hurbilpen linguistikoa erabiliz . . . 83
III.2.3.3 Perpausen identifikazioa ikasketa automatikoa erabiliz . . . 88
III.3 Iragazketa eta sailkapena, *pertzeptroiekin* . . . 92
III.3.1 *FR-Perceptron* algoritmoa: iragazketa eta sailkapena, *pertzeptroi*ekin . . . 93
III.3.2 *Iragazketa eta sailkapena* algoritmoaren adibide bat . . . 95
III.3.3 *FR-Perceptron* algoritmoan egindako egokitzapenak . . . 97
III.4 Esperimentuen prestaketa . . . 97
III.4.1 Corpusa . . . 98
III.4.1.1 *CoNLL formatua* . . . 101
III.4.2 Ebaluaziorako neurriak . . . 103
III.4.3 *Oinarrizko neurri*ak . . . 105
III.5 Kateen identifikazio automatikoa . . . 107
III.5.1 Kateen identifikazioa ikasketa automatikoa erabiliz 108
III.5.1.1 Epoch-zenbakiaren eragina . . . 108
III.5.1.2 Lehen probak, oinarrizko ezaugarriekin . . . 109
III.5.1.3 Ezaugarri linguistikoak gehituz . . . 110
III.5.2 Kateen identifikazioa, erregelak eta ikasketa automatikoa konbinatuz . . . 113
III.5.3 Ikasketa automatikoko algoritmoa baloratuz . . . 115
III.5.4 Ikasketa-corpusa handituz . . . 116
III.5.5 Emaitza idealak, eskuz desanbiguatutako informazio linguistikoa erabiliz . . . 116
III.5.6 Azken emaitza, test-corpusean . . . 118
III.6 Perpausen identifikazio automatikoa . . . 118
III.6.1 Perpausen identifikazioa ikasketa automatikoa erabiliz . . . 119
III.6.1.1 Lehen probak, oinarrizko ezaugarriekin . . . 119

III.6.1.2 Informazio linguistikoa gehituz . . . . . 120
III.6.2 Perpausen identifikazioa, erregelak eta ikasketa automatikoa konbinatuz . . . . . 121
III.6.3 Ikasketa-corpusa handituz . . . . . 122
III.6.4 Emaitza idealak, eskuz desanbiguatutako informazio linguistikoa erabiliz . . . . . 123
III.6.5 Azken emaitza, test-corpusean . . . . . 124
III.7 Ondorioak . . . . . 124

**IV Euskarako estilo- eta puntuazio-zuzentzailerantz: komaren zuzenketa automatikoa 131**
IV.1 Sarrera . . . . . 132
IV.2 Puntuazioaren garrantzia HPan . . . . . 135
IV.3 Estilo-zuzentzailearen lehen hurbilpena . . . . . 141
IV.4 Komaren erabilera: azterketa linguistikoa . . . . . 143
IV.4.1 Komaren erabileraren konparaketa: euskara eta ingelesa . . . . . 148
IV.5 Komen zuzenketa erregeletan oinarrituta . . . . . 153
IV.6 Komen zuzenketa ikasketa automatikoan oinarrituta . . . 155
IV.6.1 Esperimentuen prestaketa . . . . . 156
IV.6.1.1 Corpusaren aukeraketa . . . . . 156
IV.6.1.2 Ebaluazioa . . . . . 157
IV.6.1.3 Ikasi beharreko kontzeptua . . . . . 158
IV.6.1.4 *Oinarrizko neurriak* . . . . . 159
IV.6.1.5 Ikasketa-algoritmoak . . . . . 160
IV.6.1.6 Atributuak edo ezaugarri linguistikoak . . . . 161
IV.6.1.7 Leihoa . . . . . 164
IV.6.1.8 Jatorrizko komen eragina saihesten . . . . . 164
IV.6.2 Egindako saioak . . . . . 168
IV.6.2.1 Leihoaren aukeraketa . . . . . 168
IV.6.2.2 Ikasketa-algoritmo egokienaren aukeraketa . . 168
IV.6.2.3 Adibideen aukeraketa . . . . . 170
IV.6.2.4 Corpus motaren eragina . . . . . 171
IV.6.2.5 Ingeleseko corpusarekin komak ikasten . . . . 173
IV.6.2.6 Atributu berrien gehikuntza . . . . . 175
IV.6.2.7 Ikasketa-algoritmoa finkatuz . . . . . 175
IV.6.2.8 Corpusaren tamainaren eragina . . . . . 176

IV.6.2.9 Kateen eta perpausen identifikatzaileen informazioa koma-zuzentzailea hobetzeko . . . . . 178
IV.6.3 Komen zuzenketa, erregelak eta ikasketa automatikoa konbinatuz . . . . . 181
IV.6.4 Jatorrizko komen eragina saihesten . . . . . 182
IV.6.4.1 Corpus komagabea eta desanbiguatzaile *komaduna* erabiliz . . . . . 183
IV.6.4.2 Komarik gabeko analizatzailearen erabilpena . 184
IV.6.4.3 Adibideen azterketa . . . . . 186
IV.6.5 Ebaluazio kualitatiboa . . . . . 188
IV.6.6 Erroreen analisia . . . . . 191
IV.7 Ondorioak . . . . . 195

**V Ondorioak eta etorkizuneko lanak 203**
V.1 Ekarpenak . . . . . 203
V.1.1 Euskarako kateen eta perpausen identifikazio automatikoa . . . . . 203
V.1.2 Euskarako koma-zuzentzailea . . . . . 205
V.1.3 Bestelakoak . . . . . 208
V.2 Ondorioak . . . . . 208
V.3 Etorkizuneko lanak . . . . . 213

**Bibliografia 219**

**Glosategia 247**

**Eranskinak 258**

**A Komak zuzentzeko CG erregelak 259**

**B Komen zuzentzailea lortzeko urratsak 269**
B.1 Komaren ikasketa, corpus eta analizatzaile *komadun*ekin . 269
B.2 Komaren ikasketa, corpus *komagabe*a eta analizatzaile *komadun*arekin . . . . . 271
B.3 Komaren ikasketa, corpus eta analizatzaile *komagabe*ekin . 271

# Irudien zerrenda


II.1 Komak ikasteko erabaki-zuhaitz sinple baten adibidea. . . . . 23
II.2 SVM algoritmoaren adibide bat . . . . . 26
II.3 Geruza anitzeko euskarako sintaxi-analizatzailea. . . . . 35
II.4 Eskuz desanbiguatutako euskarako corpusa: adibide bat . . . . . 38
II.5 Dependentzietan oinarritutako etiketatzea: adibide bat . . . . . 39
II.6 Gobernatzaileen eta buruen arteko dependentzia-erlazioak . . . . 39

III.1 *Iragazketa eta sailkapena* algoritmoaren adibide bat . . . . . 95
III.2 Osagaien etiketatzean erroreak. . . . . 99
III.3 Osagaien etiketatzean errorea: zuhaitza. . . . . 99
III.4 *Epoch-zenbakiaren* eragina *FR-Perceptron* algoritmoan . . . . . 108

IV.1 Komen teoria formalizatzeko erabilitako metodologia. . . . . 144
IV.2 Komak zuzentzeko CG erregelen adibide bat . . . . . 154
IV.3 Arff formatuaren adibide bat, leihorik gabe. . . . . 165
IV.4 Corpusaren tamainaren eragina: *Euskaldunon Egunkaria* . . . . 177
IV.5 Corpusaren tamainaren eragina: *ZT corpusa* . . . . . 178
IV.6 Esaldi zuzenen proportzioa, esaldi-luzeraren arabera . . . . . 192

B.1 Komaren ikasketarako —*Eustagger komadun*a erabiliz— eman beharreko urratsen eskema-irudia. . . . . 270
B.2 Komaren ikasketarako —corpus *komagabe*a eta *Eustagger komadun*a erabiliz— eman beharreko urratsen eskema-irudia. . . . . 272
B.3 Komaren ikasketarako —*Eustagger komagabea* erabiliz— eman beharreko urratsen eskema-irudia. . . . . 274

# Taulen zerrenda


I.1 Kapitulu bakoitzarekin lotutako argitalpenak. . . . . 10

II.1 Oinarrizko ikasketa-eskemen sailkapena . . . . . 27
II.2 Euskarako corpus erroredunaren osaera. . . . . 57

III.1 Erregeletan oinarritutako kate-identifikatzailearen emaitzak . . . 77
III.2 Ingeleseko kate-identifikatzaile onenak, *CoNLL 2000*ko baldintzetan . . . . . 79
III.3 CG erregeletan oinarritutako euskarako mugatzailea . . . . . 86
III.4 CG erregeletan oinarritutako euskarako mugatzailea, perpausen identifikatzaile bihurtuta . . . . . 88
III.5 Ingeleseko perpaus-identifikatzaile onenak, *CoNLL 2001*eko baldintzetan . . . . . 90
III.6 Kateen eta perpausen identifikaziorako erabilitako corpusaren neurria. . . . . 100
III.7 Estatistikak kalkulatzeko kontingentzia-taula. . . . . 103
III.8 Kateen identifikazioko *oinarrizko neurrien* konparaketa, euskarako eta ingeleseko corpusen artekoa. . . . . 106
III.9 Perpausen identifikazioko *oinarrizko neurri*en konparaketa, euskarako eta ingeleseko corpusen artekoa. . . . . 106
III.10 Kate-identifikatzailea oinarrizko ezaugarriekin . . . . . 109
III.11 Kate-identifikatzailea: ezaugarri berriak erantsiz . . . . . 111
III.12 Kate-identifikatzailea: deklinabidearen garrantzia . . . . . 112
III.13 Kate-identifikatzailea: erregelen informazioa erantsiz . . . . . 113
III.14 Kate-identifikatzailea: C4.5 vs SVM . . . . . 115
III.15 Kate-identifikatzailea: corpusaren tamainaren arabera . . . . . 116
III.16 Kate-identifikatzailea: eskuzko etiketatzearekin . . . . . 117
III.17 *Eustagger*-en emaitzak, desanbiguazio mailaren arabera. . . . . 117
III.18 Kate-identifikatzailea: azken emaitzak test-corpusean . . . . . 118
III.19 Perpaus-identifikatzailea: oinarrizko ezaugarriak . . . . . 120

III.20 Perpaus-identifikatzailea: ezaugarriak erantsiz . . . . . 120
III.21 Perpaus-identifikatzailea: erregelen informazioa erantsiz . . . . . 121
III.22 Perpaus-identifikatzailea: corpusaren tamainaren eragina . . . . 122
III.23 Perpaus-identifikatzailea: eskuzko etiketatzearekin . . . . . 124
III.24 Perpaus-identifikatzailea: azken emaitzak test-corpusean . . . . 124
III.25 Euskarako kate-identifikatzaileen konparazioa . . . . . 125
III.26 Euskarako perpaus-identifikatzaileen konparazioa . . . . . 125
III.27 Euskarako kate-identifikatzailearen emaitzen laburpena . . . . . 126
III.28 Euskarako perpaus-identifikatzaileen emaitzen laburpena . . . . 127
III.29 Euskarako eta ingeleseko kate-identifikatzaileen konparazioa . . 127
III.30 Euskarako eta ingeleseko perpaus-identifikatzaileen konparazioa 128

IV.1 Koma jartzeko arauen konparazioa: euskara vs ingelesa. . . . . 152
IV.2 Komen identifikazioaren emaitzak, CG formalismoa baliatuz. . . 154
IV.3 Komak ikasteko eta ebaluatzeko erabilitako *Euskaldunon Egunkaria*ko corpusaren banaketa. . . . . 158
IV.4 *Baseline-neurriak* edo *oinarrizko neurriak*. . . . . 160
IV.5 Leihoaren aukeraketa . . . . . 169
IV.6 Ikasketa-algoritmoaren aukeraketa . . . . . 169
IV.7 Adibideen aukeraketa . . . . . 171
IV.8 Corpus motaren eragina . . . . . 172
IV.9 Euskararen eta ingelesaren arteko konparazioa . . . . . 174
IV.10 300 atributu gehituz . . . . . 175
IV.11 Ikasketa-algoritmoa finkatzea . . . . . 176
IV.12 Komaren eragina, *FR-Perceptron* bidezko euskarako kateen identifikatzailean. . . . . 179
IV.13 Komaren eragina, *FR-Perceptron* bidezko euskarako perpausen identifikatzailean. . . . . 179
IV.14 Kate- eta perpaus-identifikatzaile berriak gehituz . . . . . 180
IV.15 CG erregelen informazioa gehituz . . . . . 181
IV.16 Corpus komagabea erabiliz . . . . . 183
IV.17 Corpus komagabea eta *Eustagger* komagabea erabiliz . . . . . 185
IV.18 Hizkuntzalarien etiketatzearen emaitzak . . . . . 189
IV.19 Hizkuntzalarien arteko adostasuna. . . . . 189
IV.20 Tokenka egindako ebaluazio kualitatiboa . . . . . 191
IV.21 Euskarako koma-zuzentzailearen emaitzen laburpena . . . . . 197
IV.22 Ebaluazio kualitatiboaren laburpena . . . . . 199

V.1 Koma-zuzentzailearen azken emaitzak test-corpusean eta ebaluazio kualitatiboa . . . . . 206
V.2 Euskarako eta ingeleseko kate-identifikatzaileen *FR-Perceptron* bidezko emaitzak . . . . . 209
V.3 Euskarako eta ingeleseko perpaus-identifikatzaileen *FR-Perceptron* bidezko emaitzak . . . . . 210

# I. KAPITULUA

# Tesi-lanaren aurkezpen orokorra

## I.1 Gaiaren motibazioa eta kokapena

### I.1.1 *Nork jan du aita?*

Gizakiak, garuneko hainbat mekanismoren bidez, hizkuntza —bai idatzia, bai ahozkoa— ulertzeko gaitasuna dauka. Horrek desberdintzen omen gaitu nagusiki animaliengandik, eta baita makinengandik ere. Baina guk hain erraz (apenas esfortzurik gabe) egiten dugun prozesu hori, aldamenekoarekin ahoz edo idatziz komunikatzeko prozesu hori, dirudiena baino askoz konplexuagoa da; hainbesteraino, ezen hizkuntza —bere osoan— ulertzeko gai den makina bat sortzea ezinezkoa baita gaur egun. Hizkuntzaren anbiguotasun handia da, nagusiki, horren erruduna. Hitz bakar batek, adibidez, hiruzpalau adiera izan ditzake; perpaus baten esanahia ulertzeko, berriz, perpauseko hitz guztien esanahi egokia bereganatzeaz gain, hitzen arteko loturak —sintaxiak adierazten dizkigunak— ere ulertu beharko lituzke makinak.

Hizkuntzaren ulermen osoaren utopiaz gaindi, hizkuntzari lotutako eginkizun espezifikoak eta mugatuak ebaztea posible da, ordea, neurri handiagoan edo txikiagoan. Hizkuntzaren Prozesamendua[1] (aurrerantzean, HP) da eginkizun konkretu horiek lantzen dituen alorra. Hala, eremu mugatuko aplikazioak konbinatuz, ahalmen handiagoko sistemak lor litezke gero.

Itzulpen automatikoa izan zen, 1950. urte aldera, helburu handinahie-

[1] Natural Language Processing (NLP)

giak ezartzearen ondorioak sufritu zituen lehen arloetariko bat. Izan ere, garai hartan egindako hainbat saiakerek porrot egin zuten, eta itzulpen automatikoa uste bezain erraza ez zela ikusi zen. Hala, eginkizun konplexu horietan aritu beharrean, HPko zenbait ataza *sinpleagoetara* bideratu ziren komunitatearen ahaleginak: esaterako, informazioaren eskuratzea, laburpenen sorkuntza edo zuzentzaile gramatikalen garapena. Ezin esan, ordea, hauek ere helburu errazak direnik.

Tesi-lan honi dagokion zuzentzaile gramatikalaren kasuan, adibidez, esaldiaren zuhaitz sintaktiko osoa eraiki beharko litzateke automatikoki, esaldi horretan dauden zenbait gramatika-errore —komunztadura-akatsak, kasu— detektatu ahal izateko. Txosten honetan azalduko dugun moduan, ordea, batzuetan nahikoa da azaleko sintaxiak ematen digun informazioa; zuzendu nahi den gramatika-errorearen arabera, informazio linguistiko gehiago edo gutxiago behar da. Errore mota bat ala bestea izan, hortaz, estrategia desberdinak definitzen dira detekziorako.

Gramatika-zuzentzailearen baitan kokatu dugun estilo- eta puntuazio-zuzentzailea —koma-zuzentzailea, batik bat— landuko dugu tesi-lan honetan, gramatika-zuzentzailea osatzeko helburuarekin, batetik, eta makinak hizkuntzaren ezagutza osoagoa izan dezan, bestetik. Izan ere, erroreen zuzentzaile bat ez da soilik akatsa egin duena ohartarazteko tresna bat, baita makinari hizkuntzaren ulermena errazteko gailu bat ere. Azken gerezia plateretik desagertu dela ikusita, semeak aitari galdetutakoa idatziz jartzerakoan, "*nork jan du aita?*" idazten duenari, ez dela zuzen ari esan beharko zaio, bada? Gerezia nork jan duen jakin nahi badu, "*nork jan du, aita?*" idatzi beharko duela ulertarazi beharko zaio, bada? Eta zer ulertuko luke makinak, zuzenketarik egin izan ez balitz, koma hori jarri izan ez bagenu?

Koma da puntuazio-marketan tratatzen konplikatuena. Izan ere, komak zuzentzeko ezagutza-linguistiko konplexua behar da, baina ez beste zenbait gramatika-errore detektatzeko behar adina. Egia da batzuetan informazio semantikoa ere beharko litzatekeela koma zuzenak berreskuratzeko ("*nork jan du aita*" adibidean, esaterako), baina kasu gehienetan, azaleko analisi sintaktikoak ematen digun informazioa nahikoa da, zuhaitz sintaktiko osoaren beharrik gabe, IV. kapituluan ikusiko dugun moduan.

Hori dela eta, tesi-lan honetan azaleko sintaxiaren tratamendu automatikoa ere landuko dugu. Izan ere, koma-zuzentzailerako baliagarria izateaz gain, HPko beste zenbait arlotarako —itzulpen automatikorako, ahotsaren ezagutzarako...—, eta baita sintaxiaren sakoneko analisia bideratzeko ere, oso erabilgarria da.

1980. hamarkadatik aurrera —1990. hamarkadan, batez ere— itzulpen automatikoaren helburu orokorrak indarra hartu zuen berriz ere (Mitkov, 2003), estatistikan eta ikasketa automatikoan oinarritutako teknikek izan zuten gorakadaren ondorioz (garai haietako baliabide informatikoek eta hainbat corpusen sorrerak bultzatuta). Gaur egun, itzulpen automatiko guztiz zuzena pentsaezina bada ere, aurrerapen anitz egin da, eta ez bakarrik itzulpen automatikoaren alorrean. HPko beste zenbait arlotan ere, estatistikan eta ikasketa automatikoan oinarritutako teknikak baliatuz, aurrerapauso handiak eman dira.

Tesi-lan honetan, hain zuzen, ikasketa automatikoa erabiliko dugu gehienbat, eta teknika honen bidez jorratuko ditugu batez ere, bai komaren zuzenketa, eta baita horretarako beharrezkoak diren azaleko sintaxiko tresnak ere: kateen eta perpausen identifikatzaileak.

## I.1.2 Gaiaren kokapena

Atal honetan, HPan lantzen diren bi hurbilpen nagusiak aztertuko ditugu lehenik (I.1.2.1 atala). Ondoren, gure lana zer testuingurutan kokatzen den deskribatuko dugu (I.1.2.2 atalean); hau da, gure lanak orain arte IXA taldean[2] egindakoekin duen lotura azalduko dugu, eta I.1.2.1 atalean deskribatutako zein hurbilpen erabiliko den zehaztuko dugu.

### I.1.2.1 Hizkuntzaren Prozesamenduko bi hurbilpen nagusiak

HPan bi hurbilpen nagusi jorratzen dira, oro har (Oronoz, 2009):

1. Hizkuntza-ezagutzan oinarritutako teknikak edo teknika sinbolikoak: aditu batek formalismoren baten bidez definitutako erregela gramatikalak darabiltzate hauek, oro har.

2. Corpusetan oinarritutako teknikak edo teknika enpirikoak: ikasketa automatiko bidezko metodoak edo teknika estatistikoak baliatzen dituzte hauek, eskuarki.

Hizkuntza-ezagutzan oinarritutako teknikek hizkuntzari buruzko ezagutza erregeletan edo bestelako adierazpide formaletan kodetzen dute modu

[2]IXA taldea (http://ixa.si.ehu.es): Hizkuntzaren Prozesamenduan eginiko ikerketa-lana helburu duen Euskal Herriko Unibertsitateko taldea. Euskararen ikerketa aplikatua da taldearen xede nagusia.

esplizituan (Dale, 2000). Lan hau, oro har, tratatzen den fenomenoan aditu direnek —hizkuntzalariek— egiten dute, eskuz. Teknika sinboliko hauek oso erabiliak izan dira azken 40 urteetan, baina zenbait arazo ematen dituzte. Hasteko, tratatzen den hizkuntzaren mende daude; alegia, lantzen den hizkuntzaren arabera, aldatu egin behar dira erregelak. Gainera, oso zaila da hizkuntzaren fenomeno bakoitza, bere osotasunean, erregela formal batzuen bidez adieraztea; hizkuntzaren ezagutza handia behar da.

Corpusetan oinarritutako teknikek —estatistika eta ikasketa automatikoa baliatzeagatik ezagunak, batez ere—, hizkuntzaren ezagutzan oinarritutakoen aldean, hazkunde izugarria izan dute azken 20 urteotan (Oronoz, 2009), corpusen eta beharrezko baliabide informatikoen sorrerak bultzatuta, eta hizkuntza-ezagutzan oinarritutako tekniken aipatutako arazoak zirela medio. Teknika hauen desabantaila handiena zera da, ikasi nahi den kontzeptu horri buruzko informazio esplizitua duten corpus handiak behar izaten direla; alegia, eskuzko etiketatzean, lan handia egin behar izaten da kasu gehienetan[3]. Corpus horietan oinarrituta, ordea, testuetatik automatikoki erauzten da ezagutza, teknika estatistikoak baliatuz. Eskuartean darabilgun alorrean, corpusa testuzkoa izango da, eta, kasu gehienetan, nolabait etiketatua egongo da. Teknika enpirikoetan, corpusean agertzen diren elementuetatik —eta beren maiztasunetatik— erauzten da ataza desberdinak ebazteko beharrezkoa den hizkuntza-ezagutza. Gertaera linguistikoen deskribapen zabala izateko, ordea, corpusak tamaina handia behar izaten du, eta duen helburuaren arabera, maila batean edo bestean —morfologia mailan, sintaxi mailan, hitzen adiera mailan...— etiketatu behar izaten da.

#### I.1.2.2 Testuingurua

IXA taldeak hogei urte baino gehiago daramatza HPan lanean. Arlo zabal horren barruan, euskararen ikerketa aplikatua izan da taldearen xede nagusia, eta helburu horrekin, orain arte, morfologia, sintaxia eta semantika landu dira, batez ere.

Hizkuntzalarien eta informatikarien elkarlanari esker, euskarako baliabide eta tresna ugari sortu dira. Hala nola, Euskararen Datu-Base Lexikala (EDBL) (Aldezabal *et al.*, 2001), MORFEUS analizatzaile morfologikoa

[3]Aurrerago ikusiko dugun moduan (ikus II.1.4 eta II.3.2.2 atalak), badira eskuzko etiketatzerik behar ez duten —edo etiketatze-lan txikiagoarekin moldatzen diren— teknika batzuk, corpusetan oinarritutakoen artean: teknika ez-gainbegiratuak edo erdi-gainbegiratuak, kasu.

(Aduriz *et al.*, 1998), MORFEUSen oinarritutako Xuxen euskarako zuzentzaile ortografikoa (Agirre *et al.*, 1992; Alegria *et al.*, 2008b), zenbait analizatzaile sintaktiko (Aduriz *et al.*, 2006a), corpusak (Aduriz *et al.*, 2006b; Agirre *et al.*, 2006), hiztegi elektronikoak (Arregi *et al.*, 2003; Díaz de Ilarraza *et al.*, 2007), sare semantikoa (Pociello, 2008) eta beste hainbat.

Baina orain arte, IXA taldean, eta batez ere sintaxiaren eta erroreen tratamendu automatikoaren alorrean, landu diren tresna eta baliabide gehienak lehen hurbilpenari jarraiki garatu dira; alegia, hizkuntzaren ezagutzan oinarritzen diren teknikak baliatuz.

Sintaxiaren alorrean, adibidez, (Aduriz, 2000) eta (Arriola, 2000) lanei segida eman zaio (Aranzabe, 2008) tesi-lanean, eta desanbiguazio morfosintaktikorako euskararen murriztapen-gramatika, kateen detekzioa egiten duen gramatika eta dependentzien gramatika finkatuta geratu dira. Hiru tesi-lan hauetan landutako teknikak hizkuntza-ezagutzan oinarritutakoak izan dira. Corpusetan oinarritutako teknikekin ere, egin da lanik, ordea. Ezeizak (2002) euskarako etiketatzaile morfosintaktikoa burutzeko desanbiguatzaile estokastiko bat eraiki zuen, esate baterako. Gainera, dependentzietan oinarritutako euskarako analizatzaile sintaktiko estatistiko bat (Bengoetxea eta Gojenola, 2007) ere bidean da, dagoeneko.

Euskarako erroreen tratamenduan, berriz, (Maritxalar, 1999), (Oronoz, 2009) eta (Uria, 2009) tesi-lanak aipatu beharrean gaude. Lehenengoan, bigarren hizkuntzako ikasleen hizkuntza-ezagutza eskuratzeko sistema bat diseinatzeaz gain, ikasleari, idazteko prozesuan, laguntza ematen dion aplikazioa ere sortu zen. Hala, erroreen tratamenduan eta hizkuntzen ikaskuntzan aurrera jotzeko lehen urratsa izan zen tesi-lan hau. Beste bi lanetan, euskaraz egin ohi diren zenbait erroreren azterketa eta detekzioa landu zen. Oronozek (2009), esaterako, komunztadura-erroreak, datetan egindako erroreak edo postposizio-lokuzioetan egindakoak detektatzeko lana egin zuen; Uriak (2009), berriz, determinatzaile-erroreak detektatzeko gramatika bat osatu zuen, besteak beste. Errore mota hauek guztiak automatikoki detektatzeko erabilitako teknikak, halaber, hizkuntza-ezagutzan oinarritutakoen artean kokatzen dira; hau da, hizkuntzari buruzko ezagutza erregeletan edo antzeko adierazpideetan adierazia dator modu esplizituan.

Sintaxiaren eta erroreen tratamenduaren alorretatik kanpo, semantikaren eta itzulpen automatikoaren arloetan, esaterako, gehiago landu dira corpusetan oinarritutako teknikak IXA taldean. Hala, teknika hauek baliatu zituzten, besteak beste, Martinezek (2004) eta Lopez De Lacallek (2009) hi-

tzen adieren desanbiguazioa[4] lantzeko, eta baita Labakak (2010) ere, itzulpen automatikoaren alorrean.

Tesi-lan hau, beraz, IXA taldean egindako beste horien testuinguruan ulertu beharra dago, nahitaez. Hain zuzen, corpusetan oinarritutako tekniken artean erabiliena —ikasketa automatikoa— baliatuko dugu, aipatutako bi alorretan aurrerapenak egiteko: azaleko sintaxian eta erroreen tratamendu automatikoan. Gainera, hizkuntza-ezagutzan oinarritutako hurbilpena erabiliz IXA taldean lehendik egindako lanak kontuan hartu eta corpusetan oinarrituekin uztartuko ditugu, emaitzak hobetuko direlakoan.

Hala, azaleko sintaxiaren analisi automatikoan, kateen eta perpausen identifikazioa landuko dugu (ikus III. kapitulua). Kate- eta perpaus-identifikatzaile onak izatea hainbat ikerketa-lerroren garapenean oso baliagarria bada ere, tesi-lan honetan, erroreen tratamendurako baliatuko ditugu; koma-zuzentzaile bat garatzeko, hain zuzen. Izan ere, kateak eta perpausak ondo identifikatuta izatea oso lagungarria da koma-zuzentzaile bat sortzeko, IV. kapituluan ikusiko dugun moduan.

Laburbilduz, koma-zuzentzaile baten garapenean datza tesi-lan hau, eta baita horretarako beharrezkoak diren azaleko analisi sintaktikoko tresnen inplementazioan ere: euskarako kateen eta perpausen identifikatzaileak. IXA taldean orain arte alor hauetan apenas erabili den hurbilpen enpirikoa edo corpusetan oinarritutakoa baliatuko dugu, nagusiki, horretarako.

## I.2 Helburuak

Tesi-lan honen helburu nagusia, beraz, ikasketa automatikoko teknikak baliatuz euskararen prozesamendua lantzea da. Helburu orokor horren baitan, bi xede nagusi hauek landuko ditugu, batik bat:

1. Euskarako kate- eta perpaus-identifikatzaileen sorkuntza.

   IXA taldean azken urte hauetan azaleko sintaxiaren tratamendu automatikoan egindako lanari jarraipena emanez, eta azken hamabost urteetan corpusetan oinarritutako teknikek lortutako emaitza onek eta euskarako corpus egokien sorrerak bultzatuta, ikasketa automatikoko teknikak baliatuko ditugu, hizkuntza-ezagutzan oinarrituekin uztartuz, euskarako kateen eta perpausen identifikatzaileak sortzeko. Zeregin

[4] Word sense disambiguation (WSD).

hauetarako egokiena den ikasketa automatikoko algoritmoa aukeratuko dugu, eta ataza bakoitzerako baliagarriak diren ezaugarri linguistikoak identifikatzen saiatuko gara.

2. Euskarako koma-zuzentzaile automatikoaren garapena.

   Hizkuntzaren ezagutzan oinarritutako teknikak erabiliz lehendabizi, baina corpusetan oinarritutakoak baliatuz batez ere, komen identifikazio automatikoa landuko dugu. Ikasketa automatikoko teknikak erabili ahal izateko, iturri desberdinetatik —egunkariak, aldizkariak, zenbait liburu...— lortutako corpusek komak zuzen jarrita dituztela suposatuko dugu. Etiketatze-lana aurreztuko dugu modu horretan. Gainera, euskarako kate- eta perpaus-identifikatzaileek emandako informazioa baliatuko dugu, komen zuzentzailea hobetzeko asmoz. Halaber, informazio linguistikoa lortzeko dauzkagun tresnekiko mendekotasuna aztertuko dugu. Azkenik, ebaluazio kualitatiboarekin osatuko dugu ebaluazio automatikoa.

Bi helburu nagusi hauek betetzeko, beste lan batzuk ere jorratuko ditugu, ordea.

Batetik, HPan ikasketa automatikoko tekniken erabilerari buruzko azterketa egingo dugu. Ikasketa automatikoko teknikak gauza askotarako erabil daitezke, ez HPrako soilik. Azken arlo honetako erabilpena interesatzen zaigu, ordea. Tesi-lan honetan, beraz, HPan egiten den ikasketa automatikoaren erabilera aztertuko dugu, eta ikasketa-algoritmo interesgarrienak deskribatuko ditugu. Azken urteetan IXA taldean sintaxiaren eta erroreen analisiaren alorretan egindako lana osatuko dugu, corpusetan oinarritutako teknikekin. Hala, analizatzaile sintaktiko eta gramatika-zuzentzaile sendoagoak lortzen lagundu nahi dugu.

Bestetik, erroreen detekziorako oinarrizko tresnen eta baliabideen garapena ere landuko dugu. Euskaraz egiten diren erroreen azterketa burutu eta erroreen tratamendu automatikorako beharrezko diren oinarrizko tresnak sortuko ditugu: hala nola, erroreak etiketatuta dituzten corpusak edo euskarako erroreen sailkapena.

Azkenik, koma-zuzentzaile automatiko bat sortzeko helburuarekin, komaren azterketa teorikoa egingo dugu. Sintaxian eta puntuazioan adituak diren zenbait hizkuntzalariren gogoetak bilduko ditugu, euskaraz egiten den komaren erabilera formalizatzeko. Gainera, ingelesez egiten den komaren erabilera ere aztertuko dugu, eta baita euskarakoarekin konparatu ere.

## I.3 Tesi-txostenaren eskema

Lehen kapitulu honen ostean, ikasketa automatikoko zenbait teknika aztertuko ditugu II. kapituluan, eta euskarako azaleko sintaxiaren tratamendu automatikoan eta erroreen detekzio automatikoan teknika hauek baliatuz zer egin daitekeen azalduko dugu. Erroreen detekzioa jorratzeko sortutako oinarrizko tresnak eta baliabideak ere aurkeztuko ditugu.

III. kapituluan, berriz, euskararen azaleko sintaxiaren tratamendu automatikoan —kateen eta perpausen identifikazioan, batik bat— egin ditugun ikerketak aurkeztuko ditugu.

IV. kapituluan, euskararen estilo- eta puntuazio-zuzentzaile bat lortzeko helburuarekin egindako lana azalduko dugu. Koma-zuzentzailea garatzeko egindako lana da, hain zuzen, kapitulu honen ardatza.

Bukatzeko, tesi-lan hau garatzean atera ditugun ondorioak eta egindako hausnarketak plazaratuko ditugu V. kapituluan. Gainera, lan honetan abiatutako ildoei jarraiki, etorkizunean zein bide har daitezkeen aztertuko dugu.

Kapitulu hauez gain, bi eranskinek osatzen dute tesi-txosten hau: testu batean komak zuzentzeko egindako erregelak azaltzen dituena, bata; eta komen zuzentzailea lortzeko emandako urratsak laburbiltzen dituena, bestea.

## I.4 Tesiarekin lotutako argitalpenak

Sarrera-kapitulu honi bukaera emateko, tesi-lan honi lotuta burututako argitalpenen zerrenda aurkeztuko dugu jarraian. Batetik, tesiari hertsiki lotutako argitalpenak zerrendatu ditugu; bestetik, tesiarekin lotura estua izan ez arren, arlo honetan egindako ikerketei dagozkienak. I.1 taulan, argitalpen bakoitza zein kapituluri lotzen zaion zehaztu dugu[5].

### I.4.1 Tesiari hertsiki lotutakoak

- Arrieta B., Alegria I., Díaz de Ilarraza A., Aranzabe M., Aldezabal I. Using a Clause Identifier to improve a Comma Checker for Basque: testing the agreement with human judges. *ICETAL: Proceedings of the*

[5]Hauek guztiak web orri honetan daude atzigarri:
`http://ixa.si.ehu.es/Ixa/Argitalpenak/kidearen_argitalpenak?kidea=1000808992`
(2010-04-25ean atzitua).

*7th international conference on Natural Language Processing.* Reykjavik. Iceland. 2010 (argitaratzeke).

- Alegria I., Arrieta B., Carreras X., Díaz de Ilarraza A., Uria L. Chunk and Clause Identification for Basque by Filtering and Ranking with Perceptrons. *Revista del procesamiento del lenguaje natural, n 41, pags: 5-12;* 2008.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. Basque error corpora: a framework to classify and store it. *In the Proceedings of the 4th Corpus Linguistic Conference.* Birmingham. UK. 2007.

- Alegria I., Arrieta B., Díaz de Ilarraza A., Izagirre E., Maritxalar M. Using Machine Learning Techniques to Build a Comma Checker for Basque. *Proceedings of Coling-ACL.* Sydney. Australia. 2006.

- Aduriz I., Arrieta B., Arriola J.M., Díaz de Ilarraza A., Izagirre E., Ondarra A. Muga Gramatikaren Optimizazioa. *EHU/LSI/TR 26-2005.* Donostia. Euskal Herria. 2005

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L. Propuesta de una clasificación general y dinámica para la definición de errores. *Revista de Psicodidáctica, EHU. Vol 10, N 2, p. 47-60.* 2005.

- Ansa O., Arregi X., Arrieta B., Díaz de Ilarraza A., Ezeiza N., Fernandez I., Garmendia A., Gojenola K., Laskurain B., Martínez E., Oronoz M., Otegi A., Sarasola K., Uria L. Integrating NLP Tools for Basque in Text Editors. *Workshop on International Proofing Tools and Language Technologies.* University of Patras. Greece. 2004.

- Aldezabal I., Aranzabe M., Arrieta B., Maritxalar M., Oronoz M. Toward a punctuation checker for Basque. *ATALA workshop. Le role de la typographie et de la ponctuation dans le traitement automatique des langues.* Paris. France. 2003.

- Arrieta B., Díaz de Ilarraza A., Gojenola K., Maritxalar M., Oronoz M. A database system for storing second language learner corpora. *Learner corpora workshop. Corpus linguistics 2003. Volume 16, Part 1. p.: 33-41;* Lancaster, UK. 2003.

- Aduriz I., Aldezabal I., Aranzabe M., Arrieta B., Arriola J., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K., Urizar R. The design of a digital resource to store the knowledge of linguistic errors. *DRH2002 (Digital Resources for the Humanities), pp 76-78.* Edinburgh. Scotland. 2002.

| ***Kapitulua*** | ***Argitalpenak*** |
|---|---|
| II | Aldabe *et al.* (2007)<br>Aldabe *et al.* (2005b)<br>Ansa *et al.* (2004)<br>Arrieta *et al.* (2003)<br>Aduriz *et al.* (2002) |
| III | Alegria *et al.* (2008a)<br>Aduriz *et al.* (2006c) |
| IV | Arrieta *et al.* (2010)<br>Alegria *et al.* (2006)<br>Aldezabal *et al.* (2003b) |

Taula I.1: Kapitulu bakoitzarekin lotutako argitalpenak.

### I.4.2 HParen alorrean egindako gainontzekoak

- Uria L. Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M. Determiner errors in Basque: Analysis and Automatic Detection. *Proceedings de XXV Conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN), Revista n 43, pp. 41-48*, 2009.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. Hizkuntzaren Tratamendu Automatikoa Euskararen Irakaskuntzan. *BAT Soziolinguistika aldizkaria, 2008 (I), 66 zk., 61-69 or.*, 2008.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L., Amoros L. Learner and Error Corpora Based Computational Systems. *In Corpora and ICT in Language Studies: PALC 2005, J. Walinski, K. Kredens, S. Gozdz-Roszkowski (eds.), Peter Lang. Vol. 13*, 2007.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., Uria L. The Use of NLP tools for Basque in a multiple user CALL environment and its feedback. *TAL and ALAO workshop. TALN 2006. In Proceedings of the 13th Conference Sur Le Traitement Automatique des Langues Naturelles. Volume 2. p.: 815-824.* Belgium. 2006.

- Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., Uria L., Leire Amoros IRAKAZI: a web-based system to assess the learning process of Basque language learners. *EuroCALL.* Cracovia. Polonia. 2005.

- Aduriz I., Aldezabal I., Aranzabe M., Arrieta B., Arriola J., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K. Construcción de un corpus etiquetado sintácticamente para el euskera. *Revista de la Asociación Española para el Procesamiento del Lenguaje Natural. N 29, pgs 5-11.* Valladolid. España. 2002.

- Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernández G., Lersundi M. EDBL: a General Lexical Basis for the Automatic Processing of Basque *IRCS Workshop on linguistic databases.* Philadelphia (USA). 2001.

- Alegria I., Arregi X., Arrieta B. Bertsolaritzarako errima-aurkitzaile informatikoa *Elhuyar Zientzia eta Teknika. Zenb. 162; or. 20-25*, 2001.

- Arrieta B., Alegria I., Arregi X. An assistant tool for Verse-Making in Basque based on Two-Level Morphology. *Literary and Linguistic Computing. Vol. 16, No. 1; pag 29-43; Oxford University press*, 2001.

# II. KAPITULUA

## Ikasketa automatikoko teknikak azaleko sintaxiaren tratamenduan eta erroreen detekzioan

I. kapituluan aipatu dugun moduan, ikasketa automatikoko teknikak aztertu eta landu ditugu tesi-lan honetan, bi helburu nagusirekin:

- Batetik, euskarako sintaxiaren tratamendu automatikoa hobetzea; azaleko sintaxiaren tratamendu automatikoa, zehatzago esanda. Gai honen baitan, euskarako kateen eta perpausen identifikazioan aurrerapausoak ematen saiatu gara, eta ataza hauek dituzten ezaugarri morfosintaktikoak kontuan hartuta metodo desberdinak probatu ditugu horretarako.

- Bestetik, euskarako zenbait errore detektatzea; batik bat, estilo- eta puntuazio-kontuak landu ditugu, eta, horien artean, komaren kasu konkretua aztertu dugu, batez ere.

Bi helburu horiek lortzeko bidea, ordea, ez dugu hutsetik abiatu: HPan gai horien inguruan eginda zeuden hainbat ikerketa-lan baliatu ditugu, eta euskararen prozesamenduan jadanik garatuta zeuden tresnak erabili eta irekita zeuden ikerketa-lerroei segida eman diegu, gainera.

Kapitulu honetan, beraz, tesi-lan hau egiterakoan aintzat hartu ditugun tekniken eta tresnen berri emango dugu, eta, ondoren, erroreen analisirako eraikitako oinarrizko tresnak azalduko ditugu.

Hiru atal nagusitan banatu dugu kapitulua. Hasteko, ikasketa automatikoko teknika desberdinak aztertuko ditugu. Teknika hauek HPan azken urteetan hartu duten garrantzia azalduko dugu, eta tesi-lan honetan erabili diren ikasketa automatikoko algoritmoak deskribatuko ditugu. Bestalde, sintaxiaren tratamendu automatikoari, orain arte, nola egin zaion aurre aztertuko dugu, eta IXA taldean alor honetan gaur egunera arte egindako lanak deskribatuko ditugu, arreta berezia jarriz azaleko sintaxiaren tratamenduari. Kapitulu honekin bukatzeko, erroreen detekzio automatikoa aztertuko dugu, eta euskarako erroreen detekziorako egin ditugun oinarrizko tresnak azalduko ditugu. Oro har, ikasketa automatikoak, bai erroreen detekzioan, eta baita azaleko sintaxiaren tratamenduan ere, nola lagun dezakeen aztertuko dugu kapitulu honetan.

## II.1 Ikasketa automatikoa HPan

### II.1.1 Sarrera

Abney-ren (2008) arabera, hitzen kategoriaren desanbiguazio estatistikoari buruzko bi artikuluk azpimarratu zuten ikasketa automatikoaren garrantzia HPan: Church-ena (1988) eta DeRose-na (1988). Hortik aurrera —eta, batez ere, 90. hamarkadan— metodo enpirikoen eta estatistikoen suspertze handi bat izan zen, hainbat faktore zirela medio. Hauek aipatzen ditu Màrquez-ek (2002):

- Informazio linguistikoaren tratamendu sendoan oinarritutako aplikazioak ugaldu ziren.
- Baliabide linguistiko orokorren beharra sortu zen (estaldura handikoak izango zirenak, eramangarriak, moldagarriak...).
- Baliabide eta aplikazio hauek eraikitzeko aukera ematen zuten bitarteko informatikoak sortu ziren: corpus handiak, makina hobeak, algoritmo eraginkorrak...

Hala, HPan ordura arte eskuzkoa zen lana automatizatzen lagundu zuten teknika estatistikoek; esaterako, gramatiken sorkuntza edota ezagutza-baseena. Bestalde, HPko hainbat atazatarako tresna eta aplikazio sendoak sortu

ziren: desanbiguazio lexikorako edo morfosintaktikorako, egiturazko desanbiguaziorako (esaldiaren baitako erlazioen ebazpena), azaleko analisi sintaktikorako, informazioaren eskuratzerako, laburpenen sorkuntza automatikorako, itzulpen automatikorako...

Ezagutza linguistikoa eskuratzeko teknikak, hasiera batean, estatistikan oinarritu ziren, baina urteen poderioz, ikasketa automatikoko (ML[1]) teknikak nagusitu dira. Dena dela, estatistikan eta ikasketa automatikoan oinarritutakoak bereiztea ez da batere erraza. Izan ere, ikasketa automatikoko teknikak, oro har, estatistikaz baliatzen dira. Bi tekniken arteko muga lausoa dela esan daiteke.

Estatistikaz lagundutako teknikak eta ikasketa automatikokoak —biak ala biak— tratatzen dituzten problemak desanbiguazioko atazak izaten dira normalean. Beraz, egin beharrekoa zera izaten da, laburbilduz: interpretazio anitzen artean, testuinguru horretarako zuzena dena aukeratzea (Màrquez, 2002). Kontuan hartu behar da desanbiguazio-arazoak daudela HPko maila guzti-guztietan (maila lexikoan, sintaktikoan, semantikoan eta pragmatikoan).

Ikasketa automatikoan algoritmo anitz aplikatu izan dira: batetik, ikasketa sinbolikoko algoritmo klasikoak[2] (erabaki-zuhaitzak, erregelen indukzioa, adibideetan oinarritutako ikasketa...); bestetik, metodo azpisinbolikoak (neurona-sareak, algoritmo genetikoak, *support vector machines*...); hauetaz gain, ikasketa estokastikoa ere erabili izan da, eta ikasketa ez-gainbegiratua ere bai (*clustering*[3] delakoa, esaterako).

Ikasketa automatikoaren funtsa ataza jakin batzuk betetzeko ezagutza eskuratzean datza, betiere ezagutza hori modu —gehiago edo gutxiago— inplizituan adierazia dakarten datu batzuk baliatuz; hau da, adibide batzuk baliatzen dira kontzeptu bat ikasteko, domeinu bereko beste adibide batzuetan kontzeptu berari buruzko iragarpena egiteko (Màrquez, 2002).

[1] *Machine Learning*.

[2] Ez nahastu hizkuntzaren ezagutzan oinarritutako hurbilpenekin; *teknika sinbolikoak* deitzen zaie horiei ere. *Ikasketa sinbolikoko algoritmoak*, ordea, ikasketa automatikoaren familiakoak izanik, bigarren hurbilpenean kokatzen dira: corpusean oinarritutako tekniken artean, hain zuzen. *Sinbolikoak* direla esaten da, hizkuntza-ezagutzan oinarritutako hurbilpenen moduan, eskuratzen den ezagutza gizaki batek ulertzeko modukoa delako (aurrerago ikusiko dugun moduan).

[3] *Clustering* teknika: ikasketa ez-gainbegiratua egiteko teknika bat da. Ikasketa-corpuseko instantziak sailkatu gabe ditugunean —euren klasea ezagutzen ez dugunean, alegia—, instantzia horiek euren arteko zenbait antzekotasunen arabera bil daitezke; horietako multzo bakoitza klase bat edo *cluster* bat dela esaten da.

### II.1.2 Ikasketa automatikoaren funtsa

Corpusetan oinarritutako hurbilpenaren baitan kokaturiko teknika honen eragozpen handiena, arestian aipatu dugun moduan, ikasi nahi den atazarako egoki etiketatutako corpusa biltzean datza; alegia, ikasi nahi den kontzeptuari buruzko informazioa emango digun ikasketa-corpus bat behar da. Sarrerako datu hauek —ikasketa-corpusa, alegia— modu konkretu batean antolatu behar dira, baina laburbilduz, ikasi nahi den kontzeptuari eta kontzeptu horrekin erlazionatuta dauden zenbait ezaugarriri buruzko informazioa adibide edo instantzia multzo baten bidez adieraztean datza corpusaren antolamendu hau.

Ikasketa automatikoaren baitan, ikasketa-prozesuaren emaitza da kontzeptua; alegia, ikasi nahi dugun horixe bera. Kontzeptu hori ikasteko, ordea, informazioa eman behar zaio sistemari, eta informazio hori instantzia multzo batek adierazten du. Instantzia bakoitza ikasi nahi den kontzeptuari buruzko adibide independentea izan ohi da. Badira adibide soltetan ezin bana daitezkeen problemak; baina HParen baitan, oro har, adibide edo instantzia multzo batekin adieraz daiteke informazioa (Witten eta Frank, 2005).

Instantzia beraren zenbait ezaugarri neurtzen dituzten atributuen balioz espresatzen da instantzia edo adibide bakoitza. Ikasi nahi dugun kontzeptuari buruzko informazio esanguratsua duten ezaugarriak hartu behar dira kontuan. Hau da, ezaugarri bakoitzeko —atributu bakoitzeko—, balio bat izango du instantzia orok.

Hala —ikasketa automatikoko software bakoitzak bere formatu propioa baldin badauka ere—, guztiek izango dute bi dimentsiotako matrize moduko bat, non lerro bakoitza adibide edo instantzia bat izango den eta zutabe bakoitzak adibide bakoitzari dagozkion atributuen balioak gordeko dituen. Azken zutabea izan ohi da kontzeptuari dagokion balioa gordetzen duena (emaitza-atributua deituko diogu honi, hemendik aurrera). Kontzeptuaren balio posibleen arabera antola litezke adibide guztiak. Honela, adibideak klase desberdinekoak direla esaten da, ikasi nahi den kontzeptuaren balioaren arabera.

II.1.1 adibidean, CoNLL batzarreko 2000. urteko ataza partekatuan erabili zuten ikasketa-corpusaren adibide bat ikus daiteke. Lerro bakoitza instantzia bat da, adibide bat; zutabe bakoitzak, berriz, ezaugarri bati buruzko balioa gordetzen du, adierazten duen ezaugarriari dagokion balioa, hain zuzen. Lehen zutabea hitzari dagokio; bigarren zutabea, kategoriari; eta hirugarrenak kateei buruzko informazioa gordetzen du. Azken zutabe

hau (emaitza-atributua) da, hain zuzen, ikasi beharreko kontzeptuari buruzko informazioa daukana. Lehen bi zutabeek (hitza eta kategoria atributuei dagozkienek) kontzeptu hau ikasteko baliagarriak diren ezaugarrien balioak gordetzen dituzte. Datu hauek, beraz, kateen ikasketarako prestatuta daude. III.4.1.1 atalean, xehetasun gehiagorekin arituko gara formatu honetaz.

**Adibidea II.1.1**

CoNLL 2000ko eginkizun partekaturako corpusaren formatuaren adibide bat.

| | | |
|---|---|---|
| *The* | *DT* | *B-NP* |
| *deregulation* | *NN* | *I-NP* |
| *of* | *IN* | *B-PP* |
| *railroads* | *NNS* | *B-NP* |
| *and* | *CC* | *O* |
| *trucking* | *NN* | *B-NP* |
| *companies* | *NNS* | *I-NP* |
| *that* | *WDT* | *B-NP* |
| *began* | *VBD* | *B-VP* |
| *in* | *IN* | *B-PP* |
| *1980* | *CD* | *B-NP* |
| *enabled* | *VBD* | *B-VP* |
| *shippers* | *NNS* | *B-NP* |
| *to* | *TO* | *B-VP* |
| *bargain* | *VB* | *I-VP* |
| *for* | *IN* | *B-PP* |
| *transportation* | *NN* | *B-NP* |
| *.* | *.* | *O* |

### II.1.3 Ikasketa automatikoaren arazoak HPan

Ikasketa automatikoko algoritmo guzti-guztiek zenbait arazori aurre egin behar izaten diete, bai ikasketa automatikoaren berezko arazoei, eta baita HPko problemen izaera dela-eta sortzen direnei. Jarraian, garrantzitsuenak zerrendatuko ditugu:

- Atributu asko:

  Atributu asko erabiltzen dira HPan, oro har (Màrquez, 2002). Honek problemaren ebazpena zailtzen du; izan ere, atributu-konbinazio egokiena aurkitzea (*feature selection* edo *atributuen aukeraketa* delakoa) konplexuago bilakatzen du.

- Garrantzirik gabeko atributu asko:

  Atributu asko izan arren, horietako zenbaitek ez du azkenean sailkatzailean batere eragiten (Màrquez, 2002). Horiek identifikatu egin behar dira, ordea.

- Erroreak:

  Ikasketa-corpusean erroreak egotea ohikoa izaten da, eta honek, jakina, sailkatzailearen doitasunean eragiten du. *Zarata* (*noise*) dagoela esaten da, halakoetan (Màrquez, 2002).

- Atributuen izaera:

  Zenbakizkoak diren atributuak —eta diskretuak ez diren guztiak— problema bat dira ikasketa automatikoko zenbait algoritmorentzat. Izan ere, ikasketa automatikoko algoritmo batzuek ez dituzte onartzen mota honetako atributuak, eta beste batzuek —onartu arren— eraginkortasuna galtzen dute (erabaki-zuhaitzen inplementazio gehienak, esaterako, asko moteltzen dira gisa honetako atributuekin). Halakoetan, normalean, *diskretizazio* prozesu bat ematen da; alegia, zenbakizko balio bakoitzari, balio multzo jakin bateko balio konkretu bat esleitzen zaio. Metodo desberdinak daude horretarako (Witten eta Frank, 2005).

- *Overfitting* edo *gehiegizko egokitzea*:

  Ikasketa-corpusari ongi eta test-corpusari ez hain ongi moldatzen zaion eredu bat dugunean gertatzen da arazo hau; iragarpenerako baliagarriak ez diren ikasketa-corpuseko adibideei garrantzi handiegia emateagatik. Iraganean gertatu ziren kasuak, baina etorkizunean emango ez direnak adierazten dituzte adibide hauek (Abney, 2008).

- *Missing values* edo *falta diren balioak*:

  Ikasketa-corpus guztietan, praktikan, balioak falta izan ohi dira. Hauek, normalean, balio-tartetik kanpoko balio batez adierazten dira: 0 zenbakia izango da balio hori, esaterako, balio positiboak soilik har ditzakeen ezaugarri batean; -1, segur aski, zenbaki naturalak soilik har ditzakeen batean; enumeratuen bidez adierazitako atributu edo ezaugarrientzat, berriz, zuriunea edo marratxoa erabiltzen da, eta abar. *Falta diren balio* desberdinak ere egon daitezke ezaugarriren batean, mota desberdineko *falta diren balioak* adierazi nahi badira; esaterako, balio ezezagunak, garrantzirik gabeak edo gorde gabeak desberdintzeko.

Ikasketa automatikoko eskema gehienetan, ez zaie garrantzi handirik ematen *falta diren balio*ei: ezezagunak dira, eta kito. Baina arrazoi on bat egon daiteke balioa falta izatearen atzean. Hauen azterketa egitea komeni izaten da, behar bezala kodetzeko, mota desberdinetako *falta diren balio*ak era desberdinean adieraziz. Izan ere, ez da gauza bera falta den balio bat falta izatea ezezaguna delako, edo mediku batek, adibidez, atributu horri balioa ematen dion proba ez egitea erabaki duelako (ez dela esanguratsua uste duelako). Aztergai den problemaren domeinuko adituak kontsultatzea komeni da kasu hauek ebazteko (Witten eta Frank, 2005).

- Estimazio-arazoak: *datuen sakabanaketa* eta *leuntze*aren beharra.

  - *Datuen sakabanaketa* edo *corpus sparseness*: ondorio estatistikoak ateratzeko, corpusean datuak sakabanatuegi daudenean gertatzen da. Beste modu batean esanda, normalean, gertaera batzuk corpusean oso gutxitan edo inoiz ez dira agertzen, eta, beraz, gertaera horiei buruz ezin da ondorio garbirik atera (Màrquez, 2002).

  - *Leuntze*a edo *smoothing*-aren beharra: Ikasketa-corpusean inoiz gertatu ez diren kasuak izan daitezke test-corpusean. Ikasketa-corpusa txikia den kasuetan ematen da arazo hau, batez ere. Halakoetan, probabilitateekin gabiltzanez, teknika desberdinak daude inoiz gertatzen ez diren kasuei zero ez den probabilitate txiki-txiki bat emateko. *Leuntze*a (*smoothing*) deitzen zaio prozesu honi.

- *Ikasketa inkrementala*ren edo *on-line* ikasketaren beharra:

  Problema batzuetan beharrezkoa da ikasketa automatikoko eskema batek instantzia bati esleitu dion etiketa kontuan hartzea hurrengo instantzian; alegia, batzuetan, instantzia baten kontzeptu bat ikasteko, komenigarria izaten da emaitza-atributuan bere aurreko instantziek duten balioa jakitea. Horretarako, *ikasketa inkrementala* erabiltzen da. Hala, instantzia bati emaitza-atributuan esleitzen zaion balioa hurrengo instantziari jakinarazten zaio, horretarako aurreikusitako atributuari balio hori emanez.

### II.1.4 Ikasketa automatikoko tekniken sailkapena

Ikasketa automatikoko teknikak modu askotara sailka daitezke, nahiz eta multzoen arteko mugak ez izan beti garbi-garbiak (sailkapenak egiterakoan maiz gertatzen den moduan, bestalde):

> *"Dependiendo del tipo de conocimiento a adquirir, podemos hablar de conocimiento (y aprendizaje) simbólico o subsimbólico. Desde el punto de vista de la forma del aprendizaje, se puede hablar de aprendizaje supervisado o aprendizaje no supervisado. Desde el punto de vista de las técnicas empleadas, podemos hablar de sistemas basados en técnicas estadísticas (o modelos estocásticos) y sistemas basados en razonamiento inductivo."*
>
> (Màrquez, 2002)

Indukziozko arrazoibidean eta estatistikan oinarritutako sistemen arteko bereizketa lausoa da. Izan ere, gure alorrean erabiltzen diren ikasketako metodo guztiek oinarri estatistikoak dituzte: ikasketako adibideen bidez halako orokortze induktibo bat egiten saiatzen dira, adibide berriei buruzko inferentziak egiteko. Gainontzeko bereizketak ulertzeko, azalpen labur bat emango dugu:

- Sinbolikoak vs azpisinbolikoak:

  Eskuratzen den ezagutza modu esplizituan adieraz daitekeenean, ikasketa sinbolikoa terminoa erabiltzen da; esaterako, erregela bidez edo sailkapen-zuhaitzen bidez adieraz daitekeenean. Beste modu batean esanda: eskuratutako ezagutza gizakiarentzat ulergarria den modu batean gordetzen da, zeina hizkuntza-ezagutzan oinarritutako sistemetan garatzen diren adierazpideen antzekoa izango den. Interpretagarria den ezagutza horrek, beraz, fenomeno linguistiko horietan esku hartzeko aukera ematen dio hizkuntzalari adituari, sistemaren portaera hobetzen saiatzeko. Gainera, hizkuntza-ezagutzan oinarritutako sistemetan, errazago integra daiteke ikasketa sinboliko bidez lortutako ezagutza hori (Mooney, 2003). Metodo azpisinboliko batez eskuratutako ezagutza, aitzitik, gizakiok zuzenean interpretatu ezin dugun modu inplizituan adierazia dator.

- Gainbegiratuak vs ez-gainbegiratuak:

  Eredu gainbegiratuan, klaseak finkatuta daude, eta ikasketa-adibide bakoitza zein klaseri dagokion jakin badaki ikasketa-algoritmoak. Helburua, beraz, orokortzea da, gerora adibide berriak (ikusi gabeak) sailkatzeko.

Batzuetan, ikasi behar dena inplizituki etiketatuta dator, baina, normalean, eskuz etiketatu behar da. Testuinguruaren araberako zuzenketa ortografikoan, esaterako, testuinguruaren arabera zuzenak ala okerrak izan daitezkeen hitzak testu zuzenetan zuzen idatzita daudela suposatzen da, eta ikasketa-adibide gisa erabiltzen dira (eskuzko etiketatzearen beharrik gabe). Horixe da komarekin egin duguna, hain zuzen: zenbait testu konkretuko komak zuzen jarrita daudela suposatuz, testu horiekin ikasketa-prozesua burutu. Gehienetan, ordea, ezin da halakorik egin, eta eskuz etiketatu behar izaten da ikasi nahi den kontzeptua (kateen eta perpausen kasuan, adibidez).

Eredu ez-gainbegiratuan, berriz, klaseak ez dira ezagutzen *a priori*, eta ikasketa-algoritmoak gai izan behar du klase horiek zein diren ebazteko, antzeko adibideak multzotan (*cluster* deiturikoetan) bilduz.

Hurrengo atalean, tesi-lan honetan baliatu ditugun ikasketa-algoritmoak azalduko ditugu. Guztiak dira gainbegiratuak; izan ere, bai kateen identifikazioan, bai perpausen identifikazioan, eta baita komen identifikazioan ere, hasi aurretik dakigu emaitza-atributuaren klase posibleak zein diren.

### II.1.5 Oinarrizko ikasketa-algoritmoak

Atal honetan, tesi-lanean erabilitako oinarrizko ikasketa-algoritmoak azalduko ditugu: *Naive Bayes*, erabaki-zuhaitzak, *pertzeptroi*ak eta *support vector machines* edo *sostengu-bektoreen makinak*(*SVM*). *Naive Bayes* algoritmoa erabili dugu, ikasketa-algoritmo sinpleenetako bat delako; erabaki-zuhaitzak, berriz, adibide askorekin portaera ona erakusten dutelako; *pertzeptroi*ak erabili ditugu, kateen eta perpausen identifikazioan *pertzeptroi*ak erabiliz sortutako algoritmo batek (Carreras, 2005) literaturako emaitzarik onentsuenak lortu zituelako; *support vector machines* algoritmoa erabili dugu, atributu askorekin portaera ona erakutsi izan duelako, eta gaur egun HPan puri-purian dagoen ikasketa-algoritmoa delako.

#### II.1.5.1 *Naive Bayes*

Sailkatzaile estokastiko[4] sinpleena da, baina ikasketa automatikoaren arloan eta HPan arrakastaz erabili izan dena. Bayes sailkatzailearen instantzia kon-

[4]Atazan inplikatutako atributuen dependentzia probabilistikoak deskribatzen dituzte eredu estokastikoek, graforen baten bidez normalean. Grafoko adabegi bakoitzak zorizko

kretu honek, izan ere, ezaugarri asko eraginkortasunez konbinatzeko gaitasuna du (Mitchell, 1997). Gainera, ondo dabil sarrerako datuen multzoa oso handia denean (StatSoft, 2007).

Probabilitatearen banakako banaketan oinarritzen da. Adibide baten klasea asmatzeko, behatutako adibidearen probabilitatea maximizatzen duena aukeratzen da. Horretarako, Bayes-en teorematik eratorritako formula sinple bat erabiltzen da, non atributu guztiei dagozkien balioak emanik $(a_1, a_2, \ldots a_n)$ emaitza-atributuaren klase probableena ($V_{nb}$) aukeratzen baita:

$$V_{nb} = argmax_{v_j \in V} P(v_j) \prod P(a_i|v_j)$$

*Naive Bayes* sailkatzaileak hipotesi batekin jokatzen du: atazaren deskribapenerako erabilitako atributu edo ezaugarri bakoitza beste edozein bezain garrantzitsua dela; alegia, independenteak direla atributu guztiak. Hipotesi hau, ordea, ez da betetzen askotan, baina, hala eta guztiz ere, baldintza hori betetzen dela suposatzeak dakarren sinplifikazioak eredu dotore eta eraginkorrak eman ohi ditu (Manning eta Schutze, 2003).

Eskuratutako ezagutza ezin denez modu ulergarri batean adierazi, ikasketa azpisinbolikoko algoritmoa dela esaten da.

*Naive Bayes* algoritmoaren erabileraren adibide argigarriak daude Internet sarean[5].

HPan, besteak beste, eginkizun hauetarako erabili izan da: testuinguruaren araberako zuzenketa ortografikoa, morfosintaxiaren etiketatzea, preposizio-sintagmen desanbiguazioa, desanbiguazio semantikoa eta dokumentuen sailkapena (Màrquez, 2002).

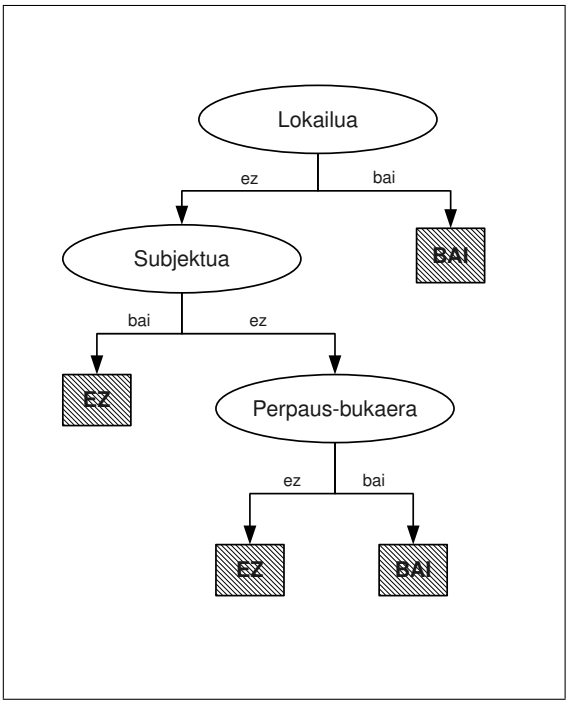
#### II.1.5.2 Erabaki-zuhaitzak

Ikasketa automatikoko eskema klasiko hau *zatitu eta irabazi* teknikan oinarritzen da (Witten eta Frank, 2005), eta grafikoki adierazita, zuhaitz baten itxura hartzen du; hortik datorkio izena. Erabaki-zuhaitzak sortzeko prozesua modu errekurtsiboan azal daiteke. Lehendabizi, atributu edo ezaugarri

aldagai bat adierazten du eta probabilitate-banaketa bat du esleituta. Banakako banaketa hauen bidez, behatutako adibide guztien baterako banaketa kalkula daiteke (Màrquez, 2002).

[5] http://www.inf.u-szeged.hu/∼ormandi/teaching/mi2/ webgunean *naive bayes* atala aukeratuta, edota http://www.statsoft.com/textbook/stnaiveb.html webgunean (StatSoft, 2007).

bat aukeratu behar da erro-adabegian kokatzeko, eta bere balio posible bakoitzeko adar bat egiten da. Gero, prozesua errepika daiteke errekurtsiboki, adar bakoitzerako, baina adar bakoitzeko baldintzak bete dituzten adibideekin soilik. Adabegi-ume bakoitzeko adibide guztiek sailkapen bera dutenean amaitzen da prozesua, kasu horretan ezingo baita adabegia gehiagotan banatu; adabegi hori, beraz, hostoa izango da. Erabaki beharreko gauza bakarra, eskema honetan, zera da: une bakoitzean aukeratu beharreko atributua. Unean uneko atributuaren aukeraketak, ordea, berebiziko garrantzia dauka, behin atributu hori erabili eta gero ez baita gerora hartuko diren erabakietan atributu bera berriz erabiltzen. Bestalde, geroz eta atributu gehiago izan, orduan eta denbora gehiago beharko du ikasketa-algoritmoak.




Irudia II.1: Komak ikasteko erabaki-zuhaitz sinple baten adibidea.

II.1 irudian hitz bakoitzaren ondoren koma jarri behar den (BAI) ala ez (EZ) erabakitzeko zuhaitz sinple bat ikus daiteke, zeina hiru atributuren mende soilik baitago (*Lokailua, Subjektua, Perpaus-bukaera*). Hiru atributu horien informazioaren arabera, erabaki-zuhaitz honek edozein instantzia berri sailkatu ahal izango luke (eredu sinplifikatua da, azalpena errazteko

adibide gisa jarritakoa). Bertan ikus dezakegun moduan, algoritmo jalea (*greedy*) denez, oso garrantzitsuak dira hasierako aukeraketak; erro-adabegiko atributuak berak zuhaitzaren alde batera edo bestera eramango zaitu, eta erabaki hori txarra izan bada, ez du izango konponbiderik.

Eskema honen abantailetako bat, ordea, zera da: ikasketa sinbolikoko algoritmo bat izanik, eskuratutako ezagutza adierazpide ulergarri batean jar daitekeela; alegia, zuhaitz formako hierarkia bat osatzen dute erabaki-zuhaitzek, eta hierarkia horretatik erregelak erauz daitezke. Hala, gaian aditua denak erregela horiek interpreta litzake. Desabantailen artean, zenbakizko atributuak erabiltzeko dituen arazoak aipa daitezke. Izan ere, zenbakizko atributuak erabiltzeko moldaketaren bat egin beharra dago eskema hau erabiltzekotan; zenbaki bakoitza atributu bitar bihurtzea da moldaketarik ohikoenetakoa (*diskretizatzea*, alegia). Askoz era naturalagoan egiten du lan zenbakizkoak ez diren atributuekin.

HPko ia maila guztietan erabili izan dira erabaki-zuhaitzak. Emaitza onak lortu izan dira, esaterako, ahotsaren ezagutzan, morfosintaxiaren etiketatzean, desanbiguazio semantikoan, analisi sintaktikoan, laburpenen sorkuntzan, entitateen ezagutzan, dokumentuen sailkapenean eta itzulpen automatikoan (Màrquez, 2002; Alegria *et al.*, 2004).

#### II.1.5.3 *Pertzeptroi*ak

Aukeratutako ezaugarrientzat edo atributuentzat pisu multzo bat ikasten du sailkatzaile lineal sinple honek. Pisu horiek atributu bakoitzaren garrantzia adierazten dute. Sailkatzaile bitarra izan ohi da hau; alegia, emaitza-atributuak bi balio soilik har ditzakeenean erabiltzen da; hau da, problemak 0 edo 1 klaseko emaitzak darabiltzanean.

Sailkapena egiteko, atributu multzoaren konbinazio lineal bat egiten da (normalean sailkatzeko dagoen adibidearen atributuen pisuen batura haztatu bat), eta klase positiboa esleitzen zaio, baldin eta emaitzak muga bat gainditzen badu; bestela, klase negatiboa (Màrquez, 2002). Zehatzago esanda, sailkatzeko dagoen adibide bakoitzeko, atributu edo ezaugarri bakoitzari dagozkion sarrerako datuak izanik ($x_1, x_2 ... x_n$), hauxe da *pertzeptroi*ak lortutako emaitza (non $w_i$ bakoitzak $x_i$ bakoitzaren pisua zehazten baitu):

$$O(x_1, x_2 ... x_n) = \begin{cases} 1 & baldin\ w_o + w_1 x_1 + w_2 x_2 + ... + w_n x_n > 0 \\ -1 & bestela \end{cases}$$

Emaitza onak lortzen ditu ikasketa multzoan erroreak daudenean eta, batez ere, ikasi nahi den kontzeptua atributu gutxi batzuen mende dagoenean.

HPko ataza hauetan erabili izan da arrakastaz: testuinguruaren araberako zuzenketa ortografikoan, etiketatze morfosintaktikoan, azaleko analisi sintaktikoan, dokumentuen sailkapenean eta adieren desanbiguazioan (Màrquez, 2002).

III. kapituluan, kateen eta perpausen identifikazioa aztertzerakoan, *pertzeptroi*ei buruz gehiago jardungo gara, *pertzeptroi*etan oinarritutako algoritmo bat erabiliko baitugu ataza horietarako.

#### II.1.5.4 *Support Vector Machines* edo *sostengu-bektoreen makinak*

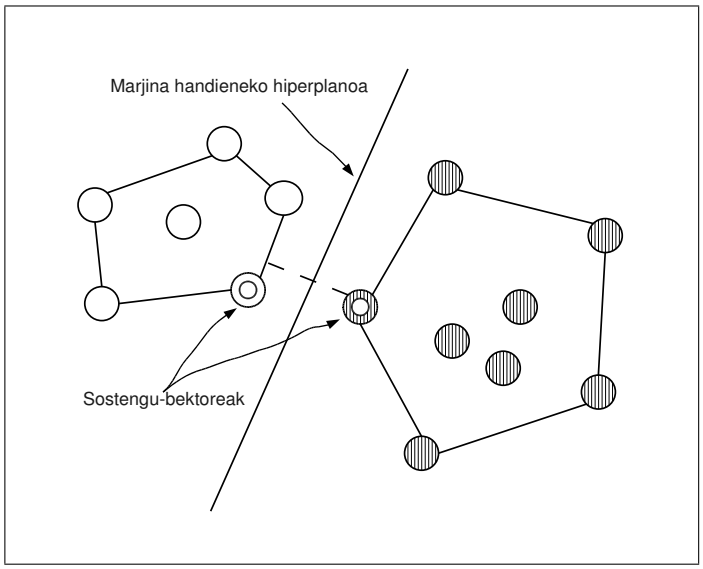
Ikasketa automatikoko algoritmo honek eredu linealek dituzten desabantailak konpontzen ditu. Izan ere, linealak ez diren datu multzoentzat soluzio bat ematen du. Bere forma sinpleenean, ordea, eredu linealetan oinarritzen da, marjina handieneko hiperplanoa[6] deitzen zaion eredu lineal berezi bat baliatzen baitu. Hainbat atazatan erabiltzen da eredu lineal hau.

Har dezagun, adibidez, bi klaseko datu multzo bat, zeina linealki banagarria den; beste modu batean esanda, bada hiperplano bat —zuzen bat, alegia— instantzien espazioan, zeinak instantzia guztiak sailkatzen dituen zuzenaren alde batera eta bestera. Witten eta Frank-en (2005) liburuko adibide batean oinarritutako II.2 irudian, argiago uler dezakegu azaltzen ari garena. Kontuan izan marraz beteak dauden zirkuluak klase batekoak direla; hutsak, beste klasekoak.

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

Esan dugun moduan, ikasketarako datuak linealki banagarriak ez diren kasuetarako ere orokortu daiteke eskema hau, *kernel* deituriko funtzioen bitartez. Sarrerako atributuen espazioa dimentsio handiagoko espazio batean bilaka daiteke, eta hori sailkatzaile polinomikoen edo hiru geruzako neurona-sareen bidez adierazia geratzen da azkenean. Praktikan, goi-muga antzeko

[6] *maximum margin hyperplane*.




Irudia II.2: Marjina handieneko hiperplano bat, eta dagozkion sostengu-bektoreak.

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

Amaitzeko, esan beharra dago *support vector machines (SVM)* izeneko ikasketa automatikoko eskema hau ez dela batere azkarra, adibide asko dituen ikasketa corpusekin lan egiterakoan, batik bat. Gainera, ez da sinbolikoa; beraz, ezin da eskuratutako ezagutza gizakiarentzat ulergarria den adierazpide batera ekarri. Hala eta guztiz ere, emaitza onak lortzen ditu oro har, erabaki-muga konplexuak eta finak eskuratzen dituelako, eta portaera bereziki ona dauka atributu askoko atazetan, eta baita linealki banagarriak ez diren problemetan ere.

Hala ere, tesi-lan honetan eredu lineal gisa soilik erabili dugu, *Weka* paketearen inplementazioan, eta parametroek besterik adierazi ezean dituzten balioekin ($C$ konplexutasun-konstantea = 1).

Patroien identifikazioarekin zerikusia duten hainbat arlotan erabilia izan da ikasketa automatikoko eskema hau; hala nola, bioinformatikan. HParen baitan, ataza hauetan erabilia izan da arrakastaz, besteak beste: azaleko sintaxiaren analisi automatikoan eta dokumentuen sailkapenean (Màrquez, 2002).

Aztertutako lau ikasketa-eskema hauen sailkapena, arestian aipatutako

irizpideen arabera, II.1 taulan ikus daiteke. Ikasketa-algoritmo hauen eta HPan gehien erabiltzen diren algoritmoen xehetasun gehiago nahi izanez gero, jo Witten eta Frank-en (2005) eta Manning eta Schutze-n (2003) liburuetara. Ikasketa automatikoa baliatuz HPa landu nahi duen edonorentzat, ikuspegi zabala dakar Màrquez-en (2002) lanak ere.

| | **Estatistikoa** | **ML** | **Sinbolikoa** | **Azpisinbolikoa** |
|---|---|---|---|---|
| **Naive Bayes** | √ | | | √ |
| **Erabaki-zuhaitzak** | | √ | √ | |
| ***Pertzeptroi*ak** | | √ | | √ |
| **SVM** | | √ | | √ |

Taula II.1: Tesi-lan honetan erabilitako oinarrizko ikasketa-eskemen sailkapena.

## II.2 Sintaxiaren tratamendu automatikoa

XX. mendearen bigarren erdialdean Noam Chomsky hizkuntzalariak sintaxiari buruz egindako azterketa formalei esker, bidea egin zitzaion sintaxiaren tratamendu konputazionalari (Aranzabe, 2008). Analisi sintaktiko automatikoaren garrantzia auzitan inork gutxik jartzen badu ere, kontuan hartu behar da, gainera, funtsezko pausoa dela esaldien barruko erlazio semantikoak zehazteko.

Analisi sintaktikoaren tratamendua, oro har, testuko esaldi bakoitzari egitura sintaktiko bat esleitzean datza (Aranzabe, 2008). Esaldien egitura sintaktikoa ezaugarritzeko era bat baino gehiago dago, ordea. Teoria linguistikoak irtenbide ugari eskaintzen jarraitzen du hizkuntzen egitura sintaktikoa deskribatzeko, eta ez dago adostasunik egitura hori erarik egokienean irudikatzeko erabil daitezkeen irizpideei buruz: osagaietan oinarritutako analisi-egituretan pentsa daiteke, dependentzia-gramatiketan, funtzio-gramatiketan, eta abar. Hizkuntzalaritza teorikoan ematen den desberdintasun hau tratamendu konputazionalean ere islatzen da: sintaxia formalizatzeko modu asko daude eta askotan ez dira bateragarriak.

Aranzaberen (2008) iritziz, hala ere, osagai-egitura eta dependentzia-gramatika ereduek markatu dute sintaxia ulertzeko modua. Berrogei bat urtez formalismo ugari garatu dira, modu ezberdinetan, bi ikuspegi hauen ba-

rruan. II.2.1 atalean, hain zuzen, analisi sintaktiko automatikorako dauden joera nagusien ikuspegi orokorra aurkeztuko dugu, hizkuntzaren ezagutzan oinarritutako tekniken bidez, batez ere; II.2.2 atalean, berriz, corpusetan oinarritutako tekniken bidez, eta zehazkiago, ikasketa automatikoko tekniken bitartez, sintaxiaren tratamendu automatikoan egin diren lanak deskribatuko ditugu. Azkenik, II.2.3 atalean, IXA taldean, orain arte, sintaxiaren tratamendu automatikoan eman diren urratsak aztertuko ditugu.

### II.2.1 Sintaxiaren tratamendu automatikoaren joera nagusiak

Testu baten edo esaldi baten analisi sintaktikoa gauzatzeko estrategia desberdinak daude. Sintaxiaren prozesamenduan, duela urte batzuk arte, hizkuntzaren ezagutzan oinarritzen diren teknikak erabili izan dira, gehienbat. Lan hauetan, hizkuntzalariek definitutako gramatiketan kodetzen da ezagutza linguistikoa. Gramatika horiek idazteko hainbat irizpide baldin badaude ere (Gojenola, 2000), testuingururik gabeko gramatikak eta egoera finituko mekanismoak bereizten dituena hartu dugu guk aintzat, sailkapen hau egiteko.

- Testuingururik gabeko gramatiketan oinarritutako sistemak

  Mota honetako sistema batzuek eragiketa nagusi gisa baterakuntza erabiltzen dute. Egitura konplexuetako informazioa konbinatzea ahalbideratzen duen operazio logiko bat da baterakuntza. Hasiera batean, operazio hau programazio-lengoaietarako definitu zen, eta ondoren, tratamendu linguistikoari aplikatu zaio. Izen honen pean, teoria eta formalismo linguistiko ugari multzokatu dira (Shieber, 1986). Besteak beste, *Lexical Functional Grammar* (LFG) (Bresnan, 1982), *Head-driven Phrase Structure Grammar* (HPSG) (Pollard eta Sag, 1994), *Government and Binding* (GB) (Chomsky, 1981), *Generalized Phrase Structure Grammar* (GPSG) (Gazdar *et al.*, 1985) eta *Word Grammar* (Hudson, 1990). Analisi sintaktikoa gauzatzeko hainbat proiektu egin dira mota honetako gramatiken bidez: TACAT azaleko analizatzaileak, esaterako, testuingururik gabeko gramatikak darabiltza gaztelaniako testuak analizatzeko (Atserias *et al.*, 1998); RASP[7] (Briscoe eta Carroll, 2002; Briscoe *et al.*, 2006) analizatzailearen abiapuntua, berriz, GPSG formalismoa izan zen.

[7] *Robust Accurate Statistical Parser*

Aipatutako gramatika horietako batean oinarritutako analizatzaile sintaktikoak honako ezaugarriak ditu (Aranzabe, 2008):

- Ezagutza linguistikoaren irudikatze- eta prozesatze-sistemaren arteko banaketa zorrotza egiten da. Ezagutza linguistikoa gramatikaren bitartez formalizatzen da.
- Ezagutza linguistikoa kategoria konplexuekin irudikatzen dute gramatikek.
- Analizatzaileek edozein motatako estrategia jarrai dezakete analisirako (goitik beherakoa, behetik gorakoa edo mistoa), nahiz eta behetik gorako analizatzaile mistoak nagusitzen joan diren.
- Analizatzaileek metodo zehatz bat gehitu beharra daukate kategoria konplexuen informazioa tratatu ahal izateko, baterakuntza izenekoa. Horrek eraginda, eraginkortasun txikiagokoak dira; izan ere, informazioaren tratamendu hau motela da.
- Gramatikan baino informazio gehiago kodetzen da lexikoan; hau da, ahalik eta erregela gutxien izatea eta ezagutza linguistikoa lexikoan adierazita egotea da bilatzen dena.
- Oso modu naturalean gehi dakioke semantikaren tratamendua. Beste sistema batzuetan, semantika (erabat analisi sintaktikoaren mende dagoena) ezaugarri laguntzailetzat hartzen da, edo sintaxiaren ondoren datorren analisitzat (analisi sintaktikoaren atzetik aplikatu behar dena). Baterakuntza-gramatiketan oinarritutako sistemek, ordea, informazio semantikoa sintaktikoarekin batera osatzea onartzen dute, eta honek are egokiago bihurtzen ditu.

- Egoera finituko mekanismoetan oinarritutako sistemak

  Morfologia tratatzean, egoera finituko teknikek —hau da, automatek eta transduktoreek— izan zuten arrakasta ikusita, teknika berak sintaxiaren atal batzuetan erabil ote zitezkeen egiaztatu nahi izan zen. Testu errealak sintaktikoki —partzialki bazen ere— tratatzeko gai izango zen analizatzaile sintaktikoa sortzea zen helburu nagusia.

  Analizatzaile sintaktikoak sortzeko formalismorik ezagunena *Constraint Grammar (CG) parser*[8] dugu, guk MG (Murriztapen Gramatika) ize-

[8] *Constraint Grammar* formalismoak (Karlsson *et al.*, 1995; Tapanainen, 1996) patroiak identifikatzeko eta etiketak jarri, kendu edo aldatzeko aukera ematen du.

narekin ezagutzen duguna. Analizatzaile hau berraztertu egin zen eta berria —sendoagoa eta azkarragoa— garatu zen: *Constraint Grammar Parser CG-2* (Tapanainen, 1996).

Mota honetako analizatzaile sintaktikoak ezaugarri hauek ditu (Aranzabe, 2008):

- Esaldiei ez die egitura sintagmatikoa esleitzen; informazio sintaktikoa esaldietako hitzetan markatzen da funtzio-etiketen bidez. Esaterako, "*Arrazoia zuen Kurt-ek*" esaldian, "*arrazoia*" hitzak *Objektua* funtzio-etiketa izango luke; "*zuen*" hitzak, *Aditz nagusia* funtzioa, eta "*Kurt-ek*" hitzari *Subjektua* funtzio-etiketa esleituko litzaioke.
- Kodetze honek, berez, dependentzia-egitura bat zehazten du inplizituki.
- Prozesatze sintaktikoaren abiapuntua morfologikoki etiketatutako esaldiak dira.
- Funtsezko bi urrats daude prozesatze sintaktikoan:
  1. Islapenarena: etiketa morfologiko bakoitzari (izen, adjektibo, adberbio, aditz...) dagozkiokeen funtzio-etiketa guztiak esleitzen zaizkio (izenei, esaterako, ondoko funtzio-etiketa hauek: *Subjektu*, *Objektu* eta *Zehar-objektu*, gehi *Adizlagun* eta *Predikatiboa* zenbait kasutan). Islapenaren urrats honetan, hurrengo pausoa errazteko helburuarekin islatzen diren etiketen kopurua muga daiteke.
  2. Desanbiguazioarena: hitz bati funtzio sintaktiko bat baino gehiago esleitu zaizkionean, urrats honetan egokiena zein den erabakitzen da, eta beste guztiak ezabatu egiten dira, ahal bada behintzat.
- Mendekotasun handia dago erregelekin esan daitekeen eta nola esan daitekeen ideien artean; hots, ezagutza linguistikoaren eta erabiltzen dugun prozesatze-sistemaren artean.
- Gramatikaren konplexutasuna oso handia da, erregelen kopurua oso handia delako eta euren artean elkarreragin dezaketelako.

Analizatzaile sintaktiko hauek dituzten mugen artean, nabarmenenak hauek dira:

- Egiteko konplexuak dira eta mantentzeko (edo aldatzeko) zailak.
- Ematen duten informazioa mugatua da (esaterako, kasurik gehienetan ez dute postposizio-sintagmen arteko dependentzia konpontzen), eta ez da beti erraza izaten benetako dependentzia-zuhaitza lortzea.
- Azaleko sintaxia da tratatzen dutena, eta ez dituzte ukitzen analisi sintaktikoaren oinarrizko zenbait kontu; esaterako, anafora, elipsia edota erlatibozko perpausetako erreferentzia-izenak.

Eta dituzten abantailak, berriz, hauek dira:

- Eraginkorrak eta azkarrak dira, egoera finituko teknikak erabiltzen dituztelako eta konputazionalki konplexua den elementurik ez dutelako.
- Edozein motatako testua trata dezakete.
- Zenbait aplikaziotarako ez dira beharrezkoak analisi sintaktiko osoak; beraz, murriztapen-gramatikaren emaitza nahikoa izaten da.

Ikuspegi konputazionala duen teknika hau Helsinkiko Unibertsitatean garatu zen lehendabizi; egun, ordea, dituzten abantailak direla eta, gero eta gehiago erabiltzen dira hizkuntzalaritza konputazionalean; zenbaitetan, maila altuagoko prozesatze baten aurreko urrats moduan.

(Voutilainen *et al.*, 1993) lanean deskribatzen dena eta Abney-rena (1995) dira automata finituak erabiliz sintaxiaren tratamendu automatikoa tratatzen den lan esanguratsuenak: analizatzaile sintaktiko partzialak garatu zituzten. Gerora, lan ugari egin dira automata finituetan oinarritutako formalismoak —*XFST*[9] eta *CG*, batez ere— erabiliz, eta hainbat hizkuntzatarako analizatzaileak sortu dira horrela (Gala, 1999; Hagen *et al.*, 2000; Bick, 2000; Mrisep, 2001; Alsina *et al.*, 2002; Schiehlen, 2003; Oflazer, 2003; Bick, 2006; Loftsson, 2007; Dhonnchadha, 2009).

[9] *Xerox Finite State Tool* (Karttunen *et al.*, 1997; Beesley eta Karttunen, 2003; Aït-Mokhtar eta Chanod, 1997): adierazpen erregularrak jaso, eta hauek transduktore bihurtzen dituen tresna. Egoera finituko kalkulua ahalbidetzen duen eragiketa multzo aberatsa du.

### II.2.2 Sintaxiaren tratamendu automatikoa ikasketa automatikoko tekniken bidez

Corpusetan oinarritutako sistemak corpus handietatik informazio sintaktikoa automatikoki eskuratzen saiatzen dira estatistika erabiliz, gerora testu berrien gainean ikasitakoa aplikatzeko (Charniak, 2000). Zenbait teknikek hainbat erregela lortzen dituzte, eta probabilitate bat eransten zaio erregela bakoitzari; informazio hori erabiltzen da, ondoren, aztertzen den esaldiaren analisia eskuratzeko. Analizatzaile estatistikoek probabilitate-konbinazio asko aztertu behar izaten dituzte eta bilaketa-estrategia on bat behar izaten dute.

Etiketatzaile estatistikoen punturik ahulena zera da: esaldi osoko ezagutza linguistikoa ez dela erabiltzen. Izan ere, bilaketa-espazioa handiegia denean, leherketa estatistikoa gertatzen da.

Hizkuntzaren ezagutzan oinarritutako lan ugari egin den arren (ikus II.2.1 atala), azken urteetan, corpusetan oinarritutako lanak gailendu dira arlo honetan ere. Hala, sintaxiaren analisiaren baitakoak diren lanak ataza sinpleagoetan banatu dira, teknika estatistikoak eta ikasketa automatikokoak baliatzeko: kateen, perpausen eta entitateen identifikazioa landu da, besteak beste, modu horretan.

CoNLL batzarrean, lau ataza partekatu antolatu ziren, aipatu ditugun hiru eginkizun horiek ikertzeko (Sang eta Buchholz, 2000; Sang eta Déjean, 2001; Tjong Kim Sang, 2002; Tjong Kim Sang eta De Meulder, 2003), eta, gaur egun, oraindik mugarri dira eta ezinbesteko aurrekari, eginkizun horiek aztertu eta landu nahi dituen ororentzat. Tesi-lan honetan, luze jardungo dugu —III. kapituluan— kateen eta perpausen identifikazioari buruz, eta, beraz, kapitulu horretan jorratuko ditugu aipatutako arlo honetako aurrekariak.

Sintaxiaren tratamendu sakonagoen alorrean, berriz, testuingururik gabeko gramatiketan edota dependentzietan oinarritutako analizatzaileen garapena landu da, teknika estatistikoak edo ikasketa automatikoko algoritmoak erabiliz; CoNLL batzarreko beste bi ekintza partekatuetan (Buchholz eta Marsi, 2006; Nivre *et al.*, 2007a) argitaratutako lanek laburbiltzen dituzte ataza honetan egindako aurrerapenak. Batzar horietan ateratako ondoriorik esanguratsuenetako bat zera da: analizatzaileen doitasuna asko aldatzen dela tratatzen den hizkuntzaren arabera (Nivre *et al.*, 2007a).

Bestalde, (Nivre *et al.*, 2007b) lanean aurkezten den kode irekiko analizatzaile-sortzailea (*MaltParser*) da, beharbada —(Kübler *et al.*, 2009) liburuan

azaltzen den MST-rekin[10] batera—, dependentzietan oinarritutako analizatzaile automatikoen sorkuntzan mugarria izan den lana, eta gehien erabiltzen dena, hainbat arrazoi direla medio:

- Batetik, hizkuntza-independentea da; alegia, dependentzien formatuan dagoen *treebank* bat emanda, *treebank*aren hizkuntzako analizatzaile bat sortzeko gai da.
- Bestetik, ez du *treebank* handiegirik behar emaitza onak lortzeko.

Hala, hainbat hizkuntzatako analizatzaileak sortzeko baliatu da. Analizatzaile-sortzaile tradizional batek (hizkuntzaren ezagutzan oinarritutako batek) gramatika bat behar du analizatzailea sortzeko; aitzitik, corpusetan oinarritutako analizatzaile-sortzaile batek *treebank* bat *besterik* ez du behar. Kontuan izan behar da kostu handiko lana dela *treebank* bat sortzea.

*MaltParser* analizatzailea hiru moduluz osatuta dago:

- Dependentzia-grafoak sortzeko, analisi-algoritmo deterministak.
- Analizatzailearen hurrengo ekintza iragartzeko, historian oinarritutako ezaugarri-ereduak[11].
- Historia horiek analizatzaileen ekintza bilakatzeko, ikasketa automatikoko algoritmo diskriminatzaileak: *memorian oinarritutako ikasketa* (MBL[12]) edo SVM (ikus II.1.5.4 atala) erabiltzeko aukera ematen du sistemak.

Ikuspuntu estatistiko hutsean oinarritutako gramatikekin lortutako analisiak linguistikoki interpretatzea, ordea, ez da erraza (algoritmoaren arabera errazagoa edo zailagoa den arren). Horren ondorioz, zaildu egiten dira ondorengo prozesuak; esaterako, interpretazio semantikoa. Hizkuntzalariek

[10] *Maximum spanning tree (MST)*: dependentzia-zuhaitz guztien artean puntuazio altuena daukana topatzeko aurkitu behar den zuhaitza.

[11] *History based feature models*: algoritmo honetan, lau aukeren arteko aukera egin behar du sistemak: bi hitz elkartzea ezkerreranzko arku batekin, bi hitz elkartzea eskuineranzko arku batekin, desplazatzea edo murriztea.

[12] *Memory based learning* (Daelemans eta Bosch, 2005): adibideetan oinarritutako ikasketa ere deitua, ikasketa adibide guztiak memorizatzen saiatzen den teknika da, batere erregelarik edo bestelako orokortzerik egin gabe. Adibide berri bat sailkatzeko, adibideen memoriatik sailkatu nahi dugunaren antzekoena den adibide-multzoa hartzen da, eta adibide-multzo horretan gehien ematen den klasea esleitzen zaio (Màrquez, 2002).

idatzitako gramatiketan, aldiz, maila altuko gertaera linguistikoak deskribatu ohi dira, sintagmak edota esaldi osoak konbinatzeko; baina garrantzi txikia ematen zaio esaldi errealetan agertzen den zenbait fenomenori; adibidez, maiztasun txikiko egitura jakin batzuei. Hori dela eta, hizkuntzaren ezagutzan oinarritutako teknikak eta corpusetan oinarritutakoak uztartzeko saioak egin dira, bakoitzaren abantailak aprobetxatzeko helburuarekin.

### II.2.3 Sintaxiaren tratamendu automatikoa IXA taldean

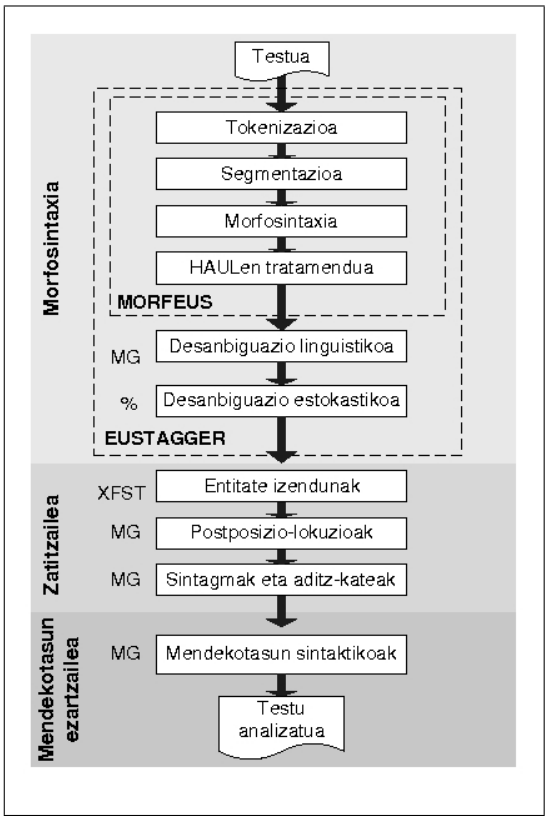
Atal honetan IXA taldean euskararen analisi linguistiko konputazionalerako erabiltzen diren baliabideak aurkeztuko ditugu. Analisi-katearen baitan, azaleko sintaxiaren tratamenduari jarriko diogu arreta, batik bat, horixe izan baita tesi-lan honetan landu dugun atala.

#### II.2.3.1 Analisi-katea

Euskararen analisi linguistiko konputazionalerako, IXA taldean, mendekotasun- edo dependentzia-egituretan oinarritutako sintaxi-analizatzaile sendo bat garatu zen (Aduriz *et al.*, 2004). Sintaxi-analizatzaileak geruzaka egiten du analisia; geruzetako bakoitzean, hizkuntza-ezagutza sakonagoa edo azalekoagoa erabiltzen da, beharraren arabera. Analisi-geruzak katean erabiltzen dira modu sekuentzialean eta moduluetan bilduta. Moduluetako batzuk, mendekotasun-ezartzailea kasu, trukagarriak dira. Analizatzailearen ezaugarriak Oronozen (2009) tesi-txostenean azalduta datoz, zehatz-mehatz. Horregatik, laburpen bat soilik egingo dugu hemen.

Analisi-kateko geruza bakoitzak, sarrera moduan, aurreko geruzak eskaintzen dion informazioa erabiltzen du, eta jasotako analisia informazio linguistiko berriarekin aberasten du. Sintaxi-analisia urratsez urrats egiten da honela, eta erabiltzailearen esku geratzen da erabili nahi duen hizkuntza-ezagutzaren mailaren aukeraketa. Geruzetako bakoitzean bereizketa argia egiten da gramatiken eta hauek aplikatuko dituzten programen artean. CG eta XFST tresnek definitutako erregela-formatuetan kodetzen dira gramatikak. Ordena askeko elementuekin lan egiteko metodologia eta tresna egokiak eskaintzen dituztelako aukeratu dira, hain zuzen, bi formalismo horiek. Analisi-kateko moduluak eta haien analisi-geruzak II.3 irudian ikus ditzakegu.

Analisi-prozesua *Eustagger*-en (Ezeiza, 2002) baitan dagoen *Morfeus* (Aduriz *et al.*, 1998) analizatzaile morfosintaktikoarekin hasten da. *Morfeus*ek sarrerako testua jaso eta tokenetan banatu ondoren, horietako bakoitzarentzat




Irudia II.3: Geruza anitzeko euskarako sintaxi-analizatzailea.

lema eta morfema konbinazio posible guztiak ematen ditu, eurei buruzko informazio morfologiko guztiarekin batera. Hitz anitzeko unitate lexikalak (HAUL) detektatzeko saioa ere egiten du. Gero, *Morfeus*ek emandako analisia sarreratzat harturik, sarrerako hitz-forma bakoitzari testuinguru horretan dagokion analisi morfosintaktiko egokia ematen zaio. Hau da, hitz-forma bati dagozkion analisi morfosintaktiko posible guztiak ematen ditu *Morfeus*ek, eta aukera guztien artean, testuinguruko informazioa begiratuta onartezinak diren interpretazioak baztertzen ditu *Eustagger*-ek, egokiarekin geratzen saiatuz, jakina.

*Ixati* zatitzaileak, aldiz, testua kateetan banatzea du helburu. Sintagma kategoriako zatia da katea. III. kapituluan ikusiko dugun legez, gainjartzen ez diren eta elkarrekin sintaktikoki erlazionaturik dauden hitz multzoak atzematean datza testua kateetan zatitzea. Analisi-katean sintaktikoki erlazionaturiko hitz multzo hauek kontsideratu ziren: sintagmak eta aditz-kateak.

Sintagmaren baitan sartzen dira postposizio-lokuzioak, entitate izendunak, adjektibo-sintagmak eta adberbio-sintagmak (Aranzabe, 2008), eta, beraz, sintagma osoa hartuko da katetzat (esaterako, "*Xabierren amonaren lagunari buruz*" kate bakar bat izango da). III. kapituluan azalduko dugu hau, zehaztasun handiagoarekin. Horiek etiketatu ahal izateko, dagoeneko morfosintaktikoki analizatutako eta desanbiguatutako testua jasotzen du zatitzaileak. Aipatu beharra dago baterakuntzan oinarritutako euskarako zatitzailea ere garatu zela: PATRIXA (Aldezabal *et al.*, 2003c), hain zuzen ere.

Zatitzaileak emandako emaitza sarrera gisa hartuta, mendekotasunak markatzea da azken urratsa. Esaldien egitura dependentzia edo mendekotasun izeneko hitzen arteko erlazioen bidez adieraz daitekeelako ideia, mendekotasun-egituretan oinarria duten gramatika-teoria eta -formalismo multzoen azpian dago. Gobernatzailea (edo burua, gurasoa) izeneko hitz baten eta, mendekoa (edo modifikatzailea, umea) izeneko beste hitz baten artean ezarritako erlazio bitar asimetrikoa da mendekotasun-erlazioa (Lin, 1998). Esaldiko hitz guztiak lotzen dituen mendekotasun-zuhaitz bat eratzen dute, normalean, erlazio horiek.

Aipatu dugun moduan, zatitzailean —hau da, kateen identifikazioan— hobekuntzak egiteko saioak egin dira tesi-lan honetan. Horretaz gain, analisi-katean agertzen ez den modulu berri bat osatzen saiatu gara: perpausen identifikazioa egiten duen modulua, hain zuzen. Modulu berri horren kokapena eztabaidagarria da. Batzuek kateen identifikazioa egin aurretik kokatzen dute (Tanev eta Mitkov, 2002); besteek, berriz, kateen informazioa darabilte perpaus-identifikatzailea hobetzeko: horrela egiten da, hain zuzen, perpausak identifikatzeko antolatu zen ataza partekatuan (Sang eta Déjean, 2001).

Analisi-katean hobekuntza hauek txertatzeko, ikasketa automatikoko teknikak baliatu ditugu, eta dagoeneko garatuta zeuden hizkuntza-ezagutzan oinarritutako gramatikekin konbinatu ditugu. III. kapituluan azalduko ditugu euskararako garatutako kateen eta perpausen identifikatzaile berriak.

Bestalde, lan hauekin batera, euskarako datuetan oinarritutako (*data-driven*) analizatzaile sintaktikoa (*Maltixa*) lortzeko lehen pausoak eman dira (Bengoetxea eta Gojenola, 2007) IXA taldean. Tresna hau dependentzia bidez etiketatutako *treebank*[13] batean oinarritzen da, eta MaltParser (Nivre *et al.*, 2007b) analizatzaile sintaktiko determinista egokitzen du (ikus II.2.2 atala). Eskuz etiketatutako euskarako *treebank*-a erabilita, % 77,12ko *LAS*-

[13]Analizatutako corpus bat da *treebank*a edo *zuhaitz-bankua*, non esaldi bakoitzaren egitura sintaktikoa adierazia datorren, zenbait etiketaren bidez.

neurria[14] lortzera iritsi dira dagoeneko Bengoetxea eta Gojenola (2009); hau da, literaturan lortzen diren emaitza onenetatik hurbil (Nivre *et al.*, 2007a).

Aipatu dugun *treebank* hori, azken urteetan IXA taldean eraiki den euskararen prozesamendurako erreferentzia-corpusaren (EPEC) baitan kokatzen da. EPEC corpusak, besteak beste, ikasketa automatikoko teknikak aplikatzeko aukera eman digu, ez soilik aipatu berri dugun euskarako datuetan oinarritutako analizatzaile sintaktikoa garatzeko, baita tesi-lan honetan egin diren ML atazetarako ere. Hurrengo atalean azalduko ditugu corpus honen eraketaren eta etiketatzearen nondik norakoak.

#### II.2.3.2 EPEC: euskararen prozesamendurako erreferentzia-corpusa

Dagoeneko aipatu dugun moduan, hizkuntza bakoitzak, HPan aurrera egin nahi badu, erreferentziazko corpus bat behar du, are gehiago ikasketa automatikoko teknikak erabili nahi badira.

Gauzak horrela, azken urteetan, lan handia egin da IXA taldean eskuz etiketatutako euskarako corpus bat sortzen. Corpus honek EPEC izena du, eta euskararen prozesamendu automatikorako erreferentzia-corpusa izateko asmoz jaio zen. Euskara batuan idatzitako testuz osatutako corpusa da EPEC, eta hainbat mailatan etiketatua izan da: morfologia eta azaleko sintaxi mailan, lehendabizi, eta sintaxi maila sakonagoan, gero. Corpus honen zati bat *XX. mendeko euskararen corpus estatistikoa* izeneko corpusetik hartu zen[15]; beste zatia, aldiz, *Euskaldunon Egunkaria* berripaperekoa da, euskara batuan egun dagoen egunkari bakarraren (*Berria*ren[16]) aitzindaria, dakigun moduan. Guztira, 200.000 token inguruko corpusa bildu zen.

Corpusa etiketatzeko, hasieran, EPEC corpuseko 50.000 hitzeko testuen multzoa aukeratu zen. Testu hauek guztiak erdi-automatikoki etiketatu ziren. Hasieran, Morfeus (Alegria *et al.*, 1997) analizatzaile morfologiko automatikoaren bitartez, hitz bakoitzaren analisi morfosintaktiko posible guztiak lortzen ziren. Horren ostean, eskuzko desanbiguazioa egiten zen; alegia, hitz bakoitzarentzat etiketa morfologiko eta sintaktiko egokia aukeratzen zen. Sintagmak eta aditz-kateak etiketatzeko ere, antzeko prozesua jarraitu zen: erregeletan oinarritutako *Ixati* zatitzaileak automatikoki detektatzen zituen kateok lehendabizi, eta eskuz zuzentzen ziren gero. Esaldi eta perpausen

[14] *Labeled Attachment Score*: etiketez gain, dependentzia-arku guztiek adierazten dituzten erlazioen zuzentasuna ere neurtzen du.

[15] www.euskaracorpusa.net

[16] www.berria.info

mugak ere modu berean etiketatu ziren.

```
"<Azterketa>"<HAS_MAI>"  S:95/0
        <Correct!> "azterketa"  IZE ARR  @KM>  AORG HAS_MAI S:95 %SIH
"<zehatzagoak>"  S:1340/0
        <Correct!> "zehatz"  ADJ IZO GRA KONP DEK ERG NUMS MUGM  @SUBJ  %SIB
"<ordea>"
        <Correct!> "ordea"  LOT LOK AURK  @LOK  AORG
"<,>"<PUNT_KOMA>"
        PUNT_KOMA
"<benetan>"  S:1954/0
        <Correct!> "benetan"  ADB ALGARR  @ADLG  S:1954 %SINT
"<ez>"  S:65/0
        <Correct!> "ez"  PRT EGI  @PRT  S:65 %ADIKATHAS
"<dela>"
        <Correct!> "izan"  ADT A1 NOR NR_HU ERL MEN KONP  @+JADNAG_MP
"<horrela>"  S:2196/0
        <Correct!> "horrela"  ADB ADOARR  @ADLG  AORG S:2196 %SINT
"<erakusten>"  S:154/0
        <Correct!> "erakutsi"  ADI SIN AMM ADOIN ASP EZBU  @-JADNAG  NOTDEK %ADIKATHAS
"<digu>"  S:390/0
        <Correct!> "*edun"  ADL A1 NOR_NORI_NORK NR_HU NI_GU NK_HU  @+JADLAG  %ADIKATBU
"<$.>"<PUNT_PUNT>"
        PUNT_PUNT
```

Irudia II.4: Eskuz desanbiguatutako euskarako corpusetik hartutako adibide bat: "*Azterketa zehatzagoak ordea, benetan ez dela horrela erakusten digu.*"

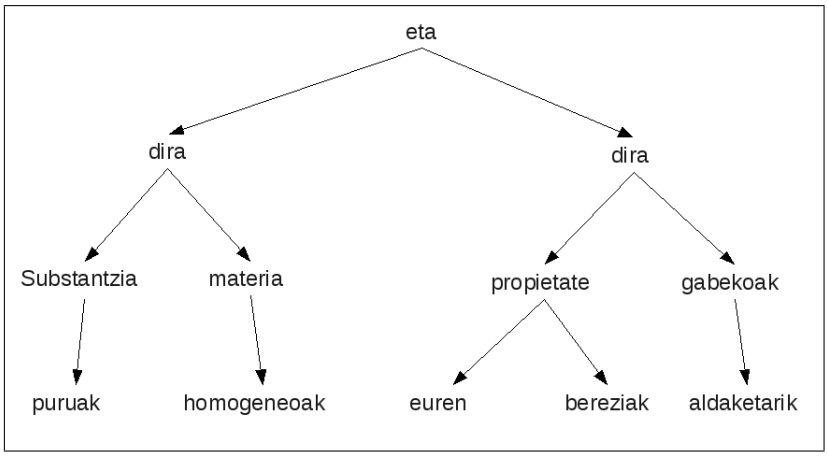
II.4 irudian ikus daiteke jatorrizko corpuseko esaldi bat, eta esaldiko token bakoitzak daukan informazio linguistikoa. Token bakoitzerako bi lerro ditugu: lehenengoan, tokena bera daukagu; bigarrenean, besteak beste, tokenari dagokion lema, kategoria (IZE, izenarentzat; ADJ, adjektiboarentzat...), azpikategoria (ARR, izena arrunta dela adierazteko; IZO, izenondoa; ALGARR, aditz-lagun arrunta...), deklinabide-kasua (ABS, absolutiborako; ERG, ergatiborako...) eta funtzio sintaktikoa, "@" markaz adierazia (@SUBJ, subjekturako; @OBJ, objekturako...). Mendeko perpausen bat tartean bada, mendekoa dela adierazten duen hitzak ere izango du mendekotasun-marka: esaterako, "*dela*" hitzaren analisian, *KONP* marka agertzen da, perpaus osagarri bat dugula adierazten duena.

EPEC corpusean, dependentziak ere etiketatu ziren gerora, 200.000 token ingurutan (Aduriz *et al.*, 2006b). Modu honetan, sintaxi maila sakonagoko informazioa izatea lortu zen. Horretarako, esaldiko elementu guztien arteko loturak zehazten dira, gobernatzaileen eta beren mendekoen arteko dependentzia-erlazioak markatuz (ikus II.5 irudia). Lehen lerroko *ncmod* erlazioak, esaterako, adjektibo batek ("*puruak*") izen bat ("*substantzia*") modifikatzen duela adierazten du; bigarren lerroko *ncsubj* erlazioak, berriz, "*dira*" adi-

Substantzia puruak materia homogeneoak dira eta euren propietate bereziak aldaketarik gabekoak dira.

ncmod (-, Substantzia-[w941], puruak-[w942], puruak-[w942])
ncsubj (abs, dira-[w945], Substantzia-[w941], puruak-[w942], subj)
ncmod (-, materia-[w943], homogeneoak-[w944], homogeneoak-[w944])
ncpred (-, dira-[w945], materia-[w943], homogeneoak-[w944])
lot (emen, eta-[w946], dira-[w945])
lot (emen, eta-[w946], dira-[w952])
ncmod (gen, propietate-[w948], euren-[w947], euren-[w947])
ncmod (-, propietate-[w948], bereziak-[w949], bereziak-[w949])
ncsubj (abs, dira-[w952], propietate-[w948], bereziak-[w949], subj)
ncpred (-, dira-[w945], gabekoak-[w951], gabekoak-[w951])
postos (-, gabekoak-[w951], aldaketarik-[w950])

Irudia II.5: Dependentzietan oinarritutako etiketatzearen EPEC corpuseko adibide bat.




Irudia II.6: Gobernatzaileen eta buruen arteko dependentzia-erlazioak grafikoki adierazita.

tzaren eta "*substantzia puruak*" hitzen arteko dependentzia adierazten du, bigarren hauek lehenengoaren subjektu direla adieraziz, gainera. II.6 irudian, dependentzia-erlazio hauek ikus daitezke grafikoki adierazita (dependentzien etiketatzeak dituen erlazio sintaktikoak gehi litezke, gainera, eta zuhaitz sintaktiko osatua lortuko genuke horrela).

## II.3 Erroreen detekzio automatikoa

Sintaxiaren tratamendu automatikoak bidea ematen du, besteak beste, erroreen detekzio automatikoa egiteko. Izan ere, analisi morfosintaktiko on batek emandako informazioan oinarrituta, zenbait errore automatikoki detektatu ahal izango dira. Atal honetan, euskarako erroreen detekzioan egin dugun oinarrizko lana azalduko dugu: euskarako erroreen sailkapena eta bilketa, hain zuzen ere, eta ikasketa automatikoko teknikak erroreen detekzioan nola balia daitezkeen ikusiko dugu. Aurretik, errorearen izaeraren inguruan jardungo gara, eta erroren detekziorako erabiltzen diren hainbat teknika deskribatuko ditugu.

### II.3.1 Erroreak eta desbideratzeak

Errore bat zer den definitzea ez da erraza. Alor honen literaturan, definizio asko erabili izan dira, eta hainbat errore mota bereizi izan dituzte ikertzaileek. Hala ere, hizkuntza-komunitate batean ezarritako arauetatik at dagoen guztia hartzen da erroretzat, oro har (Uria, 2009).

Corder-ek (1967) sistematikoki egiten diren erroreak (*systematic errors*), hutsegiteak (*mistakes*) eta lapsusak (*lapsus*) bereizten ditu. Hizkuntza-ikasleak ezjakintasunagatik (arauak edo forma zuzenak oraindik ikasi ez dituelako) erabiltzen dituen egitura edo forma okerrei deitzen die errore sistematiko Corder-ek (1967). Behin eta berriz errepikatzen dira normalean, eta ikaslearen gaitasun linguistikoari lotuta daude. Hutsegite deitzen die, berriz, araua edo forma zuzena ezagutu arren, ikasleak nahi gabe egiten dituen akats ezsistematikoei. Lapsusak, azkenik, kontzentrazio ezagatik, arreta faltagatik, nekeagatik edo antzeko arrazoiengatik egiten direnei deitzen die. Hizkuntza-ikasleak gai dira hutsegiteak eta lapsusak zuzentzeko; ez, ordea, hizkuntza-gaitasunari lotutakoak.

Corder-en (1967) ekarpen honek eragin handia izan zuen hizkuntzen ikasketaren eta irakaskuntzaren alorrean. Ordura arte nagusi zen teoria kon-

duktistak, izan ere, ez zituen erroreak aintzat hartzen; Corder-en (1967) lan honek, baina, erroreek hizkuntzaren ikasketa-prozesuan daukaten garrantzia nabarmendu zuen (Oronoz, 2009). *Erroreen analisia* deitzen zaion arloa indarra hartzen hasi zen honela, eta hainbat ikertzailek bat egin zuten ikuspegi honekin (Norrish, 1983; James, 1998; Torijano, 2004; Alexopoulu, 2005).

Bestalde, *errore* terminoa erabiltzearen kontrakoak dira bigarren hizkuntzaren ikasketaren eta irakaskuntzaren alorreko ikertzaile batzuk. Izan ere, *errore* terminoak konnotazio negatiboa dauka, porrota baitakarkigu burura (Torijano, 2004), bai ikaslearena (behar besteko arreta jarri ez duelako, behar adina ez delako saiatu...), bai irakaslearena (gauzak behar bezain ongi ez azaltzeagatik, ikasleei gaiak barneratzeko behar beste denbora ez uzteagatik...). Hori dela eta, *desbideratze* terminoa hobetsi izan da arlo honetan. Izan ere, errorea arautik aldentzen dena baita, arautik *desbideratzen* dena, nolabait.

Hizkuntza-ikasleei dagokienez, gainera, kontuan hartu behar da ez dituztela arau guztiak ezagutzen; alegia, daukaten hizkuntza mailaren arabera, arau —eta arauen salbuespen— gehiago ala gutxiago ezagutzen dituztela. Bigarren hizkuntza bateko ikasleek egiten dituzten *akatsak* —Corder-en (1991) iritziz— akats horiek ekiditeko jakin beharreko arauak ez dakizkitelako egiten dituzte, hain zuzen. Ezin zaie, beraz, *akats* edo *ez-gramatikal* deitu, koherenteak baitira ikasleek une horretan dakiten *tarteko* hizkuntza horrekin.

Maritxalarren (1999) irizpideei jarraiki, Uriak (2009) beste zentzu batean desberdintzen ditu erroreak eta desbideratzeak. Gramatika- eta ortografia-arauetatik aldentzen diren egitura guztiak errore gisa hartzen ditu; desbideratze deitzen die, berriz, testuinguru jakin batean egokiak ez diren egiturei. Desbideratzeen artean bi multzo bereizten ditu:

- Gehiegi errepikatzen diren egiturak edo akatsak egiteko beldurrez ekiditen direnak.

- Komunikazio-egoera edo eremu geografikoaren arabera gaizki erabilitako formak; hau da, hizkuntza estandarrak eta hari lotutako kode dialektalak nahasteak dakartzanak.

Hizkuntzaren ikasketan eta irakaskuntzan oso erabilia den kontzeptu bat da aipatu berri dugun *tarte-hizkuntza*rena (Selinker, 1969, 1974). Ikasle batek bigarren hizkuntza ikasten duen bitartean bere baitan sortzen den sistema edo kode linguistikoa da, hain zuzen, tarte-hizkuntza, etengabe aldatzen

doana ikasleak xede-hizkuntza *eskuratzen* duen arte. Ikasleak bereganatzen dituen lexiko eta egitura berriek markatzen dute, ikasle bakoitzaren sormen-prozesuari esker, urratsez urrats garatuz doan tarte-hizkuntza (Rey, 2004). Esaldi zuzenez eta erroredunez osatua egon arren, erroreak izango dira tarte-hizkuntzaren adierazle nagusiak. Bestalde, tarte-hizkuntza horretan ikasleak estrategia komunikatibo batzuk asmatu eta baliatu ohi ditu, xede-hizkuntzan sortzen zaizkion komunikazio-arazoak gainditzeko. Estrategia horiek ikertzea ezinbestekoa da hizkuntzaren ikasketa-prozesua osotasunean ezagutu ahal izateko, Uriaren (2009) iritziz. Gainera, maila bereko ikasleen sistema linguistikoa oso antzekoa da, eta, hortaz, errepikatu egiten dira egiten dituzten akatsak ere; egongo dira desberdintasunak, nolanahi ere, norberaren ikasketa-esperientziaren ondorioz sortutakoak. Tarte-hizkuntzaren azterketarekin, beraz, ikasle bakoitzak xede-hizkuntza ikasteko izan duen bilakaera uler daiteke.

Gisbert-en (1998) arabera, erroreen analisi on bat egin nahi bada, ikasle bakoitzaren *hizkuntza-esperientzia* ere kontuan hartu behar da; hau da, ikasle bakoitzaren ama-hizkuntza, dakizkien beste hizkuntzak eta antzeko informazioa garrantzitsua da erroreen analisia egitean. Tarte-hizkuntza bakoitzari dagozkion erroreak identifikatzeko, hizkuntza-ikasleen corpusaren azterketa sakon bat egitea proposatzen du Uriak (2009), bere tesi-lanean; ikasleen testuak mailaka sailkatu eta testuotan erroreak detektatu eta aztertu behar dira, bere iritziz.

### II.3.2 Erroreen detekziorako teknikak

Erroreak detektatzeko eta zuzentzeko, teknika ugari erabiltzen dira, HPko gainontzeko alorretan baliatzen diren ia teknika guztiak erabil baitaitezke eginkizun zehatz honetarako ere (Oronoz, 2009). Teknika guztiek eredu formaletan dute oinarria. Jurafsky eta Martin-ek (2000), bost eredu formal bereizten dituzte: egoera-makinak, erregela-sistemak, analisi logikoa, probabilitatearen teoria eta ikasketa automatikoa.

Oinarrian erabiltzen duten informazio mota irizpidetzat hartuta, berriz, HPan erabiltzen den sailkapen bera baliatzen da, arestian ikusi duguna, hain zuzen:

- Hizkuntza-ezagutzan oinarritutako teknikak edo teknika sinbolikoak.
- Corpusetan oinarritutako teknikak edo teknika enpirikoak.

Gaur egun, HParen gakoa teknika sinboliko eta enpirikoen konbinazio egokian dagoela onartzen dute ikertzaile gehienek (Oronoz, 2009).

#### II.3.2.1 Hizkuntza-ezagutzan oinarritutako teknikak

Arestian aipatu dugun moduan, hizkuntza-ezagutzan oinarritutako teknikatan, eskuz garatzen dira erregelak. Hortaz, hizkuntzaren ezagutza handia behar da hizkuntzaren fenomenoak deskribatuko dituzten erregela formalak idazteko. Hala eta guztiz ere, erroreen detekziorako maiz erabili izan dira hizkuntza-ezagutzan oinarritutako teknikak. Dena dela, beharrezkoa den informazio linguistikoaren arabera, estrategia desberdinak erabiltzen dira. Hiru azpimultzotan banatu zituen Oronozek (2009):

1. Ezagutza morfologikoa edo hitz mailako informazioa behar duten teknikak.

   Hitzekin maila morfologikoan lan egiteko adierazpen erregularrak eta automatak erabiltzen dira normalean. Hizkuntzaren ezaugarriek erabakitzen dute maiz teknikaren aukeraketa. Esate baterako, ingelesaren antzeko hizkuntzetan, errore ortografikoak lantzeko, adibidez, hitz baten forma flexionatu guztiak zerrenda batean gorde daitezke, eta zerrenda horrekin hizki-zuhaitz moduko egoera finituko automata bat eraiki. Horrela, hitz zuzenak ezagutuko lirateke eta hitz erroredunak baztertu. Suomieraren, hungarieraren, turkieraren eta euskararen moduko flexio maila handiko hizkuntzetan, ordea, forma flexionatu asko sortzen dira, eta forma horiek guztiak hiztegian gordetzea ez da izaten soluziorik onena. Arazoa konpontzeko beharrezkoa da hitzak morfologikoki lantzen dituzten teknikak erabiltzea. Euskarako zuzentzaile ortografikoan (Agirre *et al.*, 1992), adibidez, hitzen analisirako Koskenniemi-k (1983) garatutako bi mailatako morfologian oinarritutako analizatzailea erabiltzen da. Analizatzaile hau gai da analisi eta sorkuntza morfologikoa egiteko.

2. Ezagutza sintaktikoa edo esaldi mailako informazioa behar duten teknikak.

   Esaldi bat sintaktikoki analizatzen denean, egituraren bat esleitzen zaio. Egitura horrek esaldiko osagai linguistikoak errepresentatzen ditu, eta beren arteko harreman gramatikalak azaleratzen ditu. Erroreen

detekzioaren lan-eremuan, gramatikalak edo ez-gramatikalak izan daitezke hizkuntza-egiturak. Sintaxi mailako erroreak detektatzeko teknikak hiru multzo nagusitan banatu zituen Oronoz-ek (2009): egoera finituko mekanismoak erabiltzen dituzten teknikak, gramatiketan oinarritzen direnak eta gainontzeko guztiak.

Egoera finituko mekanismoak eta patroiak identifikatzeko erregelak baliokidetzat jotzen ditu Oronozek (2009). Izan ere, erregelak automatetan edo transduktoreetan bihurtzen dira, normalean. Errorea identifikatzeko patroi konkretu batzuk ematean datza teknika hau, labur esanda. Formalismoren bat erabiliz garatzen diren erregelen bitartez idazten dira patroiak. Beste hainbat atazatarako baliagarriak badira ere (hala nola, desanbiguazioko eginkizunetarako), erroreen detekzioan erregelak idazteko erabili izan dira maiz CG eta XFST formalismoak. (Uria, 2009; Badia *et al.*, 2004; Johannessen *et al.*, 2002; Birn, 2000; Arppe, 2000) lanetan, erroreen detekziorako CG erabili dute, eta (Hashemi *et al.*, 2003; Oronoz, 2009) lanetan, XFST. Beste lan batzuetan (Naber, 2003; Carlberger *et al.*, 2002), formalismo propioak eraikitzen dituzte eginkizun honetarako.

3. Esaldi mailatik harago dagoen ezagutza —semantika, pragmatika eta diskurtsoa— behar duten teknikak.

   Ortografia eta sintaxi mailan zuzenak diren baina kokatuta dauden testuinguruan zentzurik ez duten hitzek edo esaldiek errore semantiko bat osatzen dutela esan ohi da (adibidez, "*bilatu*" hitza erabiltzea, "*aurkitu*" esan nahi denean). James-ek (1998) zehatzago sailkatu zituen errore (semantiko) hauek. Halakoekin, hizkuntzaren ezagutzan oinarritutako teknikak erabiltzea ez da oso usua. Aitzitik, teknika estatistikoak eta ikasketa automatikoari dagozkionak erabiltzen dira errore hauek detektatzeko. Hurrengo atalean aztertuko ditugu, hortaz.

#### II.3.2.2 Corpusetan oinarritutako teknikak

HPko alor askotan erabiltzen dira corpusetan oinarritutako teknikak; erroreen detekzioan, baina, ez dira maiz erabili. Izan ere, erroreen etiketatzea ataza konplexua eta garestia da. Oronozen (2009) iritziz, informazio zuzena duen corpusa biltzea zaila bada, interesatzen zaigun atazari dagokion erroredun corpusa biltzea are neketsuagoa da.

Badira zenbait corpus etiketatu (hala nola, *Brown Corpus*a, *British National Corpus*a edo ikasleen testuez osatutako *Cambridge Learner Corpus*a), baina hauek ez dituzte ikerlarien behar denak asetzen, zenbait arrazoi direla medio:

- Etiketatzea ez izatea egokia edo nahikoa, ebatzi nahi den atazarako.
- Ebatzi nahi den atazaren hizkuntzarako ez egotea corpusik.

Arazo hauek direla eta, eskuz etiketatutako corpus berriak sortzen ari dira hizkuntza askotarako.

Badira ataza batzuk, ordea, etiketatze berezirik behar ez dutenak. Testu digitalak edo Internet sarean pilatutako testuak erabili izan dira, esate baterako, terminoren baten zuzentasuna neurtzeko (Moré *et al.*, 2004): oso maiz azaltzen diren terminoak zuzenak direla suposatzea da gakoa. Informazio gordailu erraldoi bat da Internet, eta informazio andana horri zukua ateratzea da, gaur egun, arlo desberdinetako hainbat ikerlariren erronka. HPan ere, Interneteko testuak corpus moduan erabiltzeko geroz eta joera handiagoa dago.

Testuinguruaren araberako zuzenketa ortografikoa[17] izango da, ziurrenik, corpusetan oinarritutako tekniken bidez gehien landu den arloa, erroreen detekzioaren baitan. Izan ere, ez da etiketatze berezirik behar ataza konkretu horretarako. Eskuarki, etiketatu gabeko corpusak erabiltzen dira. Adibidez, "*ebatzi/ebatsi*" hitzen artean aukeratzeko, euskarako corpus ustez zuzenak baliatzea nahikoa izango litzateke, bi hitzon artean —dagokion testuinguruan— zuzena zein den erabakitzeko. Inguruko hitzen nolabaiteko zerrenda bat gorde eta antzekotasun gehien dituena aukeratzea da problema hau ebazteko modua. Asmatutako adibidearekin jarraituz, "*ebatzi*" hitzaren inguruan gehien agertzen diren hitzen zerrenda izango genuke, alde batetik: "*arazo, problema...*"; "*ebatsi*" hitzaren ingurukoak, bestetik: "*lapur, auto, euro...*". Probarako corpusean, zalantzazko kasu baten aurrean ("*ebatzi*" vs "*ebatsi*"), bere inguruko hitzak hartuko lirateke aintzat, eta aurrez osatutako hitz-zerrendekin konparatuko lirateke. Erabaki beharreko hitzaren ingurukoek "*Lapur, auto, euro...*" hitz-zerrendakoekin antzekotasun gehiago balituzte, "*ebatsi*" hitza aukeratuko litzateke; aitzitik, "*arazo, problema...*" zerrendako hitzekin bat baletoz, orduan "*ebatzi*" hitza aukeratuko litzateke.

[17] *context sensitive spelling correction*: antzeko bi hitzen artean zuzena aukeratzea helburu duen HPko arloa.

Ikasketa automatikoko algoritmo desberdinak erabili dira eginkizun honetarako: bayesiar metodo hibridoak (Golding, 1995), *winnow*[18] algoritmoa (Golding eta Roth, 1996, 1999) edota *Ezkutuko Semantikaren Analisia* (LSA)[19] delakoa (Jones eta Martin, 1997).

Reynaert (2004), aldiz, maizen agertzen diren *bigram*en[20] zerrenda berezi batean eta agerkidetzetan[21] oinarritzen da. Zerrenda horrek, datu horiez gain, corpusetik erauzitako hitz bateko edo biko anagrama guztiak biltzen ditu hash-taula erraldoi batean; hash-taulako gakoak nola osatzen dituen, horixe da errepresentazio modu honen berezitasuna. Izan ere, taulako elementu bakoitzaren karaktere bakoitzeko bere *ISO Latin-1* kodea lortu eta karaktere guztienak batuz lortzen du elementu horren gako-zenbakia. Adierazpen bitxi honek antzeko hitzen kodeak, nolabait, kontrolatuta izatea dakar.

(Agirre *et al.*, 1998) lanean, berriz, errore ortografikoen zuzentzailea hobetzen saiatu ziren, ingeleserako; zuzentzaileak ematen dituen proposamenen artean zuzena den bakarra aukeratzea zen helburua. Horretarako, hitzen desanbiguazioko teknikak zenbait heuristikorekin, maiztasunen estatistikekin eta *Constraint Grammar* formalismoaren abantailekin uztartu zituzten.

Komak ikasteko ataza ere multzo honetan sar daiteke *a priori*. Izan ere, zenbait testutan komak ondo jarrita daudela suposa daiteke ikasketa automatikoari ekiteko. IV. kapituluan ikusiko dugun legez, ordea, pentsa daitekeena baino korapilatsuagoa da kontua.

Corpusaren etiketatzea beharrezkoa den atazetarako, berriz, ikasi nahi diren errore moten adibideak eskuz etiketatzea da biderik zuzenena, baina ez hargatik eraginkorrena, kostu handia baitauka. Horregatik, hainbat teknika erabili dira azken urteetan —eta indarra hartzen ari dira pixkanaka—

[18] Banatzaile linealen familiakoa, *on-line* egiten den ikasketa-algoritmo sinple bat da. Ezaugarri edo atributu bitarrekin egiten du lan, eta 2 emaitza-klaseko problemekin (0/1). Adibide berriak sailkatzeko, sarrerako ezaugarri edo atributuen batura haztatu bat egiten da (konbinazio lineala). Emaitza mugaren azpitik badago, 0 itzultzen da; bestela, 1. Gaizki iragarritako adibideek ezaugarri bakoitzari ematen zaion pisua aldatzen dute, ikasketa multzora ahalik eta gehien egokitzeko (Màrquez, 2002).

[19] *Latent Semantic Analysis*: Testu idatzien semantika adierazteko gaitasuna duen tresna bat da. Modu matematikoan adierazten ditu testuko paragrafo eta hitzak. Ondoren, adierazpen matematiko horren gainean zenbait eraldaketa burutzen ditu, eta horrela, testuen eta bertan dauden hitzen arteko erlazio semantikoak neurtzeko gai da (Zelaia *et al.*, 2003).

[20] *Bigram* deitzen zaie corpusean elkarren segidan agertzen diren hitz-pareei.

[21] Agerkidetza (*co-ocurrence*): Dokumentu batean bi termino edo gehiago elkarren ondoan izateari deritzo, zoriz lortutakoa baino handiagoa den maiztasun batekin.

etiketatzearen kostua jaisteko: *active learning*, errore artifizialen sorkuntza automatikoa eta ikasketa partzialki gainbegiratua. Banan-banan aztertuko ditugu, jarraian.

- *Active learning* edo *ikasketa bizi*a:

  *Active learning* edo *ikasketa bizi* delakoaren muina, etiketatu gabeko ikasketa-corpus batean, etiketatzeko adibide multzo egokia aukeratzean datza. Izan ere, ikasketa-corpuseko adibide guztiak ez dira neurri berean erabilgarriak (Banko eta Brill, 2001). Ikasketa automatikoko algoritmoarentzat baliagarrienak diren adibideak eskuz etiketatuz, emaitza hobeak lor litezke, zoriz aukeratutako adibideak etiketatuz lortutakoak baino.

  Banko eta Brill-en (2001) iritziz, adibide baten sailkapen-etiketa geroz eta zalantzagarriago izan, orduan eta erabilgarriagoa izango da ikasketarako. Adibide zalantzagarrienak aukeratzeko modu desberdinak egon daitezke.

  (Dagan eta Engelson, 1995) lanean, esaterako, corpus txiki bat darabilte kategoria-etiketatzaile bat ikasteko. Etiketatzaile-familia bat sortzen dute gero, etiketatzailearen probabilitateak zoriz aldatuz. Ondoren, etiketatzaile desberdinak probatzen dituzte corpus handian, eta sailkatze-emaitza desberdinenak ematen dituzten adibideak aukeratzen dituzte. Hauek izango dira eskuz etiketatuko direnak, gerora, berriz, ikasketa-corpusean erabiltzeko.

  Izan ere, zenbait hizkuntzatarako —eta hizkuntza batzuen zenbait domeinutarako edo zenbait problematarako— corpus etiketatuak egon baldin badaude ere, baliabide gutxiagoko beste hizkuntza askotarako edota domeinu eta problema ez hain ezagunetarako falta izaten dira. Corpusen eskuzko etiketatzea, izan ere, lan neketsu eta garestia da, esan dugun bezala. Hala, *active learning* erabiltzeak etiketatzearen kostua murrizteko balio dezakeela frogatu izan da azken urteotan, etiketatzearen kalitatea mantenduz betiere (Ringger *et al.*, 2009). NAACL'09[22] batzarraren baitan egindako mintegi batek (*Active Learning for Natural Language Processing*[23]) gai hau aztertu du duela gutxi, eta HPko hainbat atazatan teknika honen erabilpena azaldu dute hainbat lanek.

[22]North American Association for Computational Linguistics.
[23]http://nlp.cs.byu.edu/mediawiki/index.php/Workshop_on_Active_Learning_for_NLP

- Erroreak automatikoki sortzea:

  Erroreak ikasterakoan dugun arazo handiena ez da corpusaren tamaina, baizik corpus horretan dauden erroreen proportzioa. Cermenoren (2008) lanean, adibidez, euskarako determinatzaile-erroreak eta komunztadura-erroreak landu ziren ikasketa automatikoa erabiliz. Determinatzaile-erroreei dagokienez, corpuseko hitzen % 1,5 soilik ziren erroreren baten parte, 113.290 hitzeko corpus batean. Komunztadura-erroreei dagokienez, berriz, hitz guztien % 1,2 ziren erroreren baten parte, 13.591 hitzeko corpus batean. Bi corpus hauek euskarako corpus erroredunetik hartutako zatiak dira (ikus II.3.3.1 atala corpus horren eraketaren inguruko informazio gehiago jasotzeko). Arazo hauek direla eta, erroreak automatikoki sortzen saiatzen dira hainbat lanetan. Determinatzaile-erroreak sortzea, adibidez, ez dirudi horren zaila. "*Zenbait lagun etorri dira*" adibidean, "*Zenbait lagunak etorri dira*" sortzea bideragarria iruditzen zaigu, esate baterako. Hau automatikoki egitean, ordea, adibide ez horren errealistak sortzeko arriskua dago, eta kontuz egin beharreko lana da.

  Foster eta Andersen-ek (2009) bide hau darabilte, esaterako. Hitz bat kenduz, edo berri bat gehituz, edo hitz bat mugituz edo beste batengatik aldatuz sortzen dituzte gramatika-errore artifizialak. Hala, ikasketa-corpus artifiziala osatzen dute, gerora gramatika-erroreak detektatzeko atazan erabiltzeko. Beren ustez, gramatika-erroreen detektatzaile bat lantzeko modurik onena corpus handi bat etiketatzea bada ere, horrek kostu handia dauka. Euren iritziz, erroreak detektatzeko aplikazio batek ikasketa automatikoko teknikak erabiltzen baditu, geroz eta ikasketa-corpus handiagoa izan, orduan eta emaitza hobeak lortuko dira (goi-muga batera iritsi bitartean, betiere). Ezaugarri asko maneiatzen dituzten ikasketa algoritmoetan (entropia handienekoan[24] edo SVM algoritmoan, kasu), ikasketa-corpusak bereziki handia behar du, *overfitting* edo *gehiegizko egokitzea* deitzen den fenomenoa gerta ez dadin. Gainera, hizkuntza batzuetarako ikasleen corpusak —erroredunak, alegia— egon baldin badaude ere —ingeleserako, *Cambridge Learner Corpus* da ezagunena, baina ez da librea—, hizkuntza gutxituetan arazoa are nabarmenagoa da. Halakoetan, errore artifizialen sorkuntza

[24]Entropia handienaren printzipioak, neurri batean, aldagai batek har ditzakeen balio guztien probabilitate berdintsua bilatzen du; alegia, ez egotea balio bat beste bat baino gehiago gertatzeko probabilitaterik.

izan daiteke soluzio bat.

Erroreak sortzea, hala ere, ez da lan erraza, eta eskuzko lana beharrezkoa da gehienetan. Zein motatako erroreak sortu, eta zein modutara, erabaki beharra dago, eta horrek benetako erroreen azterketa bat eskatzen du. Prozesu hori amaituta, ordea, automatikoki egin daiteke hortik aurrerako guztia, eta behar adinako corpusa sor daiteke.

Errore artifizialak erabili dira, besteak beste, honako lan hauetan ere:

- Hitz elkartuen banaketan eta hitz-ordenan egindako erroreak detektatzeko suedierako testuetan. Errore mota horien adibide artifizialak gehitu zituzten testuetan, eta corpus hori erabili zuten euren errore-detektatzailea entrenatzeko (Sjöbergh eta Knutsson, 2005).
- Ingeleseko izen zenbakarrien eta zenbaezinen erroreak detektatzeko (Brockett *et al.*, 2006).
- Aditzen errore artifizialak sartuta, zuhaitz sintaktikoak aztertu zituzten Lee eta Seneff-ek (2008), aditzen erabilera okerrak zuzentzen saiatzeko.
- Esaldi gramatikal eta ez-gramatikalen artean bereizten duen sistema bat garatzeko errore artifizialez osatutako corpus bat darabilte (Okanohara eta Tsujii, 2007), (Wagner *et al.*, 2007) eta (Foster eta Andersen, 2009) lanetan.
- Erroreak detektatzeko sistemen ebaluazio automatikoa egiteko (Bigert, 2004).
- Ikasketa ez-gainbegiratua egiteko ere erabili izan dira errore artifizialdun testuak, (Smith eta Eisner, 2005a) eta (Smith eta Eisner, 2005b) lanetan, kasu.

Laburbilduz, HPko hainbat alorretarako baliagarria da artifizialki sortutako erroreez osatutako corpusa, baita erroreen detekziorako eta errore-detektatzaileen ebaluaziorako ere.

- Ikasketa ez-gainbegiratua edo ikasketa erdi-gainbegiratua:

  Oinarrian, ideia hau dago: gainbegiratutako ikasketa bidez ahalik eta ondoen etiketatutako corpus txiki bat eta etiketatu gabeko corpus handi bat erabiliz, emaitza hobeak lor daitezke kostu txikiagoarekin. Alegia, oinarrizko corpus bat etiketatuz, ikasketa gainbegiratua egin, eta

lortutako sailkatzailea erabiltzea corpus berri erraldoia (etiketatu gabea) etiketatzeko. Gero, corpus erraldoi hori (sailkatzaileak etiketatua) erabil liteke berriz ikasketa-corpus gisa. Banko eta Brill-ek (2001) eta Charniak-ek (1996), ordea, modu honetan egindako esperimentuetan oso hobekuntza txikiak lortu zituzten. Banko eta Brill-en (2001) iritziz, halere, ikasketa ez-gainbegiratua eta *active learning* teknika konbinatuz, bi tekniken propietateak aprobetxa litezke, eta emaitzak gehiago hobetu.

Arlo hau aztertu eta garatzeko pauso garrantzitsua izan da NAACL batzarraren baitan berriki (2009an, hain zuzen) egindako mintegi hau: *workshop on semi-supervised learning for natural language processing*[25]. Bertan, ikasketa erdi-gainbegiratua erabili duten hainbat lan erakutsi dira, eta teknika honek —ikasketa guztiz gainbegiratuaren eta ez-gainbegiratuaren aldean— dituen abantailak azaleratu dira. Alde batetik, etiketatutako testuen kopuru txikiagoa behar da (gainbegiratuaren aldean), eta, bestalde, ikasketa ez-gainbegiratuak dauzkan zailtasunak ekiditen dira neurri batean. Gainera, ikasketa ez-gainbegiratuak ez ditu emaitza onak lortzen iragarri beharrekoa egitura konplexua den problemetan. Hala, azkenaldian, HPan, garrantzitsua bilakatu da datu etiketatuak eta ez-etiketatuak izatea ataza askotan hobekuntzak lortzeko. Beraz, hazi egin da ikasketa erdi-gainbegiratuarekiko komunitatearen interesa.

Erroreen detekzio ez-gainbegiratua lantzen da, esate baterako, (Tsao eta Wible, 2009; Quixal eta Badia, 2008; Chodorow eta Leacock, 2000) lanetan. Bestalde, Abney-k (2008) ikasketa erdi-gainbegiratuarekin hizkuntzalaritza konputazionalean jorratu diren eta jorra daitezkeen bideak xeheki azaltzen ditu bere liburuan.

Aukera hauez gain, gaur egun indarra hartzen ari dira, era berean, datuen eskuratzea errazteko helburua duten beste alor hauek ere: giza-konputazio banatua (*distributed human computation*) (Gentry *et al.*, 2005) eta kostuarekiko sentibera den ikasketa automatikoa (*cost-sensitive machine learning*) (Elkan, 2001).

Corpus egokia lortzea —modu batean edo bestean— ikasketa automatikoa egiteko ezinbesteko aurre-urratsa baldin bada ere, pauso horren ostean hasten da ikasketa automatikoko algoritmoen edo sailkatzaile desberdinen

[25] http://sites.google.com/site/sslnlp/

erabilera. Hurrengo atalean, orain arte erroreen detekzioan egin diren lanak aztertuko ditugu, zein ikasketa-eskema baliatu dituzten ikusteko. Corpus erroreduna nola osatu duten ere aztertuko dugu.

### Ikasketa automatikoa erabiliz erroreen detekzioan eginiko lan esanguratsuenak

Ikasketa automatikoa gehiegi erabili ez den HPko alorra da erroreen detekzioa. Testuinguruaren araberako zuzenketa ortografikoa da, lehen ikusi dugun bezala, salbuespen bakarrenetarikoa. Halere, badira beste lan batzuk, eta horiek aztertuko ditugu jarraian.

(Sjöbergh eta Knutsson, 2005) lanean, automatikoki txertatu zituzten erroreak suedierazko corpus batean; ez zen, beraz, eskuzko etiketatzerik egin: errorea txertatzean, errore gisa etiketatzen zen, eta gainontzeko guztia zuzentzat ematen zen. Bi errore konkretu landu zituzten: hitz-ordenako erroreak eta hitz-konposatuetan egindakoak. Sjöbergh eta Knutsson-en (2005) ustez, biak ala biak erraz sor zitezkeen automatikoki, kontuan harturik hitz-ordenako erroreak alboko bi hitzen ordena trukatzean soilik datzala. Hitz-ordenari dagozkion erroreak detektatzeko heuristiko sinple batzuk baliatu baldin baziren ere, *transformation based learning* (TBL)[26] ikasketa-algoritmoan oinarritu zen lan hau. Ikasketa sinbolikoko algoritmo bat izatea garrantzitsua zen egileentzat, sortzen zituzten erregelak ulergarriak —eta, beraz, aldagarriak— izan zitezen adituentzat. Hitz-konposatuen erroreen detekzioan, % 95eko doitasuna eta % 67ko estaldura lortu zuten; hitz-ordenako erroreetan, berriz, % 60ko doitasuna eta % 30eko estaldura.

(Kárason, 2005) lanean, berriz, hitz-konposatuen erroreak detektatzen saiatu ziren, islandierako corpus bat erabiliz. Erroreak automatikoki txertatu zituzten, betiere heuristiko batzuk aplikatuz. *Memorian oinarritutako ikasketa-algoritmoa* (MBL) baliatu zuten, eta % 81eko doitasuna eta % 65eko estaldura lortu zuten.

(Izumi *et al.*, 2003) lanean, ingelesaren ikasle japoniarrek hitz egitean egiten dituzten lexiko- eta gramatika-erroreak antzematen saiatu ziren. El-

[26] *Transformation based learning* (transformazioan oinarritutako ikasketa): Ikasketa automatikoko algoritmo hau problemaren soluzio sinple batekin hasten da (kategoria etiketatzaile batean, maizen ematen den kategoria esleitzea, esaterako), eta transformazioko erregela batzuen bidez, soluzio hobea bilatzen da. Transformazioko erregelak sortzeko, soluzio zuzenarekin konparatzen da uneko soluzioa; urrats bakoitzean, aurrekoan sortutako erroreak ondoen konpontzen dituen erregelak sortzen ditu. Transformazio hauek behin eta berriz egin ohi dira, hobekuntzarik lortzen ez den arte (Brill, 1995).

karrizketa-saioetan grabatutako audioak erabili ziren ikasketa-corpus gisa. Eskuz etiketatu ziren 13 errore motari lotutako erroreak 5.599 esalditan, eta 617 esaldiko corpus batean ebaluatu zen sistema. Errore mota bakoitzari etiketa bat esleitu zitzaion, eta etiketa horiek erabili ziren gero corpuseko erroreak etiketatzean. Corpuseko errore bakoitzari errore motaren etiketa eta errore horri zegokion zuzenketa esleitzen zitzaion etiketatze-prozesuan. Entropia handienaren metodoa baliatuta egindako probetan ez zituzten oso emaitza onak lortu, eta ikasketa-corpus txikiari egotzi zioten errua (errore mota batzuk ez ziren ikasketa-corpusean inoiz ematen). % 30eko estaldura eta % 50eko doitasuna lortu zuten. Esaldi zuzenak gehitu zizkioten gero, ikasketa corpusari, eta doitasuna igo zen arren, estaldura jaitsi egin zen. Errore artifizialak gehituta egin zuten azken proba, eta doitasuna % 80ra igotzea lortu zuten (estalduraren datuak % 30ean mantendu ziren).

(Shi eta Zhou, 2005) lanean, ahotsaren ezagutza-sistema baten emaitza hobetzeko TBL ikasketa-algoritmoa erabili zen, zenbait ezaugarri lexiko eta sintaktiko konbinatuz eta hitzaren konfiantza-puntuazioak (*word confidence score*) erabiliz. Eskuz etiketatu zituzten erroreak ikasketa-corpusean. Ahotsaren ezagutza-sistemaren errore-tasa % 12an murriztu zuten modu honetan. Etorkizunerako, informazio semantikoa gehitzea aurreikusten zuten. Ahotsaren ezagutzaren alorrean antzeko lan gehiago ere badira (Antal *et al.*, 2002).

Laburbilduz, esan liteke erroreen detekzioa ez dela batere ataza erraza. Ikasketa automatikoa egiteko beharrezkoa den corpus erroredun egokia osatzeko dauden zailtasunak dira arazoaren muina, askotan. Hau dela eta —aztertu ditugun lanetan ikus daitekeen moduan—, erroreak automatikoki sortzera jotzen da. Modu horretara sortutako corpusarekin ikasitako ereduak, ordea, ez dira oso fidagarriak izaten, eta emaitza kaskarrak lortzen dituzte errore errealekin ebaluatzerakoan.

Bestalde, adierazgarria iruditzen zaigu ikasketa automatikoa egiteko errore oso konkretuak aukeratu izana aztertutako lan hauetan, testuinguru oso mugatua dutenak, gainera (hitz-konposatuetan eginiko erroreak, bi hitzen arteko ordena erroreak...). Izan ere, ez bakarrik detekziorako, errore artifizialen sorkuntza automatiko txukuna egiteko ere, komeni da problema ahalik eta gehien mugatzea. Hala eta guztiz, lortzen diren emaitzak —oro har— ez dira tresna fidagarriak eraikitzeko behar bezain onak izaten.

### II.3.3 XUXENg: euskarako gramatika-zuzentzaile automatikoa

Erroreak edo gaizki erabilitako egiturak detektatzea oso ataza garrantzitsua da, besteak beste, hizkuntzaren tratamendu automatikoko ondorengo bi alorretan: ortografia- eta gramatika-zuzentzaileetan eta ordenagailuz lagundutako hizkuntzen ikaskuntzan. IXA taldean, bi alor horietan aurrerapenak egiteko asmoarekin lantzen dihardugu euskarako erroreen detekzioa.

Normala den bezala, zuzentzaile ortografikoa lortzeko bideratu ziren lehen urratsak, informazio linguistiko gutxiago behar delako horretarako.

Hala, 1992. urtean sortu zen Xuxen[27], euskarako egiaztatzaile/zuzentzaile ortografikoa, euskarako analizatzaile morfologikoaren garapenaren ondorioz (Urkia, 1997; Alegria, 1995; Agirre *et al.*, 1992). Kukich-en (1992) ideia hau zuen oinarrian: deskribapen morfologikoa egina dagoenean, erraza da ortografia-zuzentzaile bat garatzea; alegia, analizatzaile morfologikoak estandar moduan analizatzen dituen hitzak zuzenak lirateke, eta ezagutzen ez dituenak, berriz, okerrak. Orduz geroztik, zuzentzailea datu berriekin elikatua izan da eta testu-prozesadore desberdinetarako, web-erako eta OCR-rako[28] bertsioak sortu dira. 2008. urtean, GNU/Linux inguruneko aplikazioetan erabilgarria den Xuxenen bertsio libre bat jarri zen eskuragarri (Alegria *et al.*, 2008b). Xuxen zuzentzaile ortografikoaren mugen artean, aipagarria da testuinguruaren araberako zuzenketa ortografikorako (*ebatzi* vs *ebatsi*, kasu)[29] ez dagoela prestatuta.

Euskarako Xuxen egiaztatzaile/zuzentzaile ortografikoa garatu ondoren, azken urteotan XUXENg gramatika-zuzentzailea sortzea izan dugu helburu IXA taldean. Errore sintaktikoak detektatzea, dena dela, errore ortografikoak detektatzea baino konplexuagoa da. Izan ere, anbiguotasun handiagoa dago, oro har. Gainera, informazio linguistiko gehiago behar izaten da errore horiek detektatzeko, eta askotarikoak direnez, baliteke errore sintaktiko desberdinak detektatu ahal izateko teknika desberdinak behar izatea. Gainera, analisi- eta inplementazio-lan sakona eskatzen dute.

Hala eta guztiz ere, azken urteotan, IXA ikerketa-taldean lan handia egin da sintaxiaren esparruan, eta gramatika-zuzentzailerako bidean urrats

[27] http://www.xuxen.com helbidean topa daitezke Xuxenen bertsio guztiak.

[28] *Optical Character Recognition* edo karaktere-ezagutze optikoa. Besteak beste, eskanerrek darabilte.

[29] *Real-word error* deritze ingelesez errore hauei, eta, lehenago aipatu dugun moduan, *context sensitive spelling correction* edo *testuinguruaren araberako zuzenketa ortografikoa* deitzen zaio errore hauen detekzioaz eta zuzenketaz arduratzen den arloari.

handiak eman dira horrela. Besteak beste, euskararen sintaxia lantzeko oinarrizko baliabideak garatu dira (Gojenola, 2000), euskararen desanbiguazio morfologikoa eta azaleko sintaxia landu da (Aduriz eta Díaz de Ilarraza, 2003) CG bidez, euskarako aditzen azpikategorizazioaren azterketa burutu da (Aldezabal, 2004) eta dependentzia-gramatiken formalismoa jarraituz garatutako analizatzaile sintaktikoa sortu da (Aranzabe, 2008). Sintaxi-lan horiek baliatuz, gramatika-zuzentzailea sortzeko bidean lehen urrats garrantzitsuak ere eman dira jada. Izan ere, Oronozek (2009) postposizio-lokuzio okerrak, data okerrak eta komunztadura-erroreak detektatzeko tesi-lana egin zuen, eta Uriak (2009) determinatzaile-erroreak detektatzeko CG erregelak sortu zituen.

XUXENg, euskarako gramatika-zuzentzailearen lehen prototipoa, ordea, 2003. urtean garatu zen. Otegik (2003) informazio linguistikorik gabe detekta zitezkeen estilo- eta puntuazio-erroreak integratu zituen *Microsoft Word* testu-prozesadorean (honi buruzko informazio zabalagoa emango dugu IV. kapituluan); hala nola, puntuazio-marken ondorengo espazioak, esaldi luzeegiak, letra larriaren erabilera puntuazio-marken ondoren... Bigarren prototipo batean (Otegi, 2006), informazio linguistikoa behar duten errore batzuen integrazioa egin zen; Oronoz-en (2009) tesi-lanean sortutako postposizio-lokuzioetan gertatzen diren erroreak detektatzeko gramatikaren lehen bertsioa txertatu zen bertan. Horretarako, informazio linguistikoa ematen zuten tresnak integratu behar izan ziren Otegiren (2006) aplikazioan, liburutegi dinamiko batean bilduta: tokenizatzailea, analizatzaile morfosintaktikoa, desanbiguaziorako modulua eta errore sintaktiko sinple batzuk detektatzeko garatutako murriztapen-gramatika.

Baina, esan bezala, Oronozek (2009) egindako lana izan da, orain arte, euskarako gramatika-zuzentzailea lortzeko bidean egindako ekarpenik garrantzitsuena. Hizkuntza-ezagutzan oinarritutako teknikak erabiliz, erroreen detekzioa landu zuen Oronozek (2009); hots, hizkuntzari buruzko ezagutza erregeletan edo beste adierazpideetan —baina beti modu esplizituan— kodetzen duten teknikak baliatu zituen. Hala, bi esparrutako erroreak detektatzen saiatu zen: testuinguru mugatuko erroreak eta testuinguru zabalekoak. Lehenengo multzoaren baitan, daten eta postposizio-lokuzioen erroreekin jardun zuen, patroiak baliatuz; bigarrenean, berriz, komunztadura-erroreak detektatzeko ahalegina egin zuen, eta horretarako, mendekotasun-zuhaitzetan informazioa kontsultatzeko tresna bat (*Saroi*) garatu zuen.

Esku artean dugun tesi-lana Oronozen (2009) tesi-lanaren osagarria dela esan daiteke neurri batean, lehenago aipatu dugun moduan. Izan ere,

erroreen detekziorako berak hizkuntzalaritza konputazionaleko hurbilpen bat erabili bazuen (hizkuntza-ezagutzan oinarritutakoa), beste hurbilpena erabili dugu guk (corpusetan oinarritutakoa). Ikasketa automatikoa baliatu dugu, zehatzago esanda, euskarako errore batzuen detekzioa lantzeko, beste zenbait gauzaren artean. Are, hurbilpen bateko eta besteko teknikak konbinatzera ere jo dugu, aukera hau izan dugun guztietan.

IV. kapituluan ikusiko dugun moduan, erroreen detekzioari dagokionez, estilo eta puntuazio-kontuak landu ditugu tesi-lan honetan.

Atal honetako hurrengo puntuetan, aipatutako euskarako gramatika-zuzentzailea garatu ahal izateko egin dugun oinarrizko lana deskribatuko dugu. Batetik, erroreen detekzioan aurrerapausoak eman ahal izateko hain beharrezkoa den baliabidea aurkeztuko dugu: euskarako corpus erroreduna (II.3.3.1). Gero, corpus hori aztertzearen ondorioz sortutako euskarako erroreen sailkapena (II.3.3.2). Euskarako erroreak ondo kudeatzeko eta gordetzeko tresnak —datu-baseak eta web-interfazeak— aurkeztuko ditugu ondoren (II.3.3.3), eta bukatzeko, euskarako erroreak detektatzeko ikasketa automatikoa erabiliz zer egin daitekeen azalduko dugu (II.3.3.4).

### II.3.3.1 Euskarako corpus erroreduna

Lehen ere aipatu dugu corpus handi eta sendoak izatearen garrantzia HPan. Uriaren (2009) eta Oronozen (2009) tesi-lanetan, argi uzten da corpusen beharra (ikus euren tesi-txostenak, corpus motei edo corpusgintzari buruzko informazio xeheagoa izateko), eta erroreen detekzioa lantzeko corpus erroredun baten beharra azpimarratzen da, gainera. Hala sortu zen euskarako corpus erroreduna; behar horri erantzuteko, hain zuzen. Hartara, bi alor konkretuetan aurrerapausoak eman ahal izatea zen gure xedea: erroreen tratamendu automatikoan eta ordenagailuz lagundutako hizkuntzen ikasketan eta irakaskuntzan.

IXA taldean egun eskura dagoen euskarako corpus erroredunaren bilketari Maritxalarrek ekin zion 1990. urtean (Maritxalar *et al.*, 1997). Euskaltegi batekin harremanetan jarri eta ikasleek idatzitako testuak biltzen hasi zen paperean, eta banan-banan transkribatzen. Geroztik, ahalik eta corpus handiena, heterogeneoena eta sendoena lortzeko asmoz, honako iturri hauetatik bildu genituen testu erroredunak:

- Hizkuntza-ikasleen testuak:

  Hainbat euskaltegitako testuak jaso, euskarri elektronikora pasa eta zenbait ezaugarriren arabera sailkatu ziren: i) irakasleak proposatutako ariketa motaren arabera (laburpena, idazlana edo gutuna den, esaterako), eta ii) egilearen arabera (betiere, anonimotasuna bermatuz).

  1990. urtean hasi ginen testu hauek jasotzen eta gaur egun, oraindik, horretan dihardugu. Guztira, 113.290 hitzeko corpusa osatu dugu oraingoz, testu hauekin bakarrik. Hizkuntza-ikasleen corpus honetan, testu batetik bestera zailtasun maila eta errore-tasa asko alda daitekeenez —ikasleen jakintza mailaren araberakoak izango baitira—, hiru mailatan banatu genituen testuok, *Helduen Euskalduntzearen Oinarrizko Kurrikulua* (HABE, 1981) aintzat hartuta: behe mailan, garai bateko 1-6 urratsetako testuak (39.117 hitz); maila ertainean, 7-9 urratsetakoak (42.219 hitz); eta goi mailan, 10-12 urratsetakoak (31.954 hitz).

  Hizkuntzaren ikasketa-prozesua edota ikasle bakoitzaren tarte-hizkuntza ezagutu ahal izateko, beharrezkoa da erroreen eta desbideratzeen diagnosi linguistikoaz gain, diagnosi psikolinguistikoa ere egitea. Horretarako, adibide erroredunekin batera, testuingurua (testua), testuaren ezaugarriak eta hizkuntza-ikasleari berari dagozkion datuak ere gordetzea komeni da (Uria, 2009). Hau guztia egin ahal izateko, testu bakoitza kode konkretu batekin identifikatzea erabaki zen. Interesatzen zaigun informazio guztia biltzen du bere baitan kode horrek: testuaren jatorria (euskaltegia) eta urtea, hizkuntza maila, ikaslearen identifikazio hizkia(k) eta ariketa mota.

- EuskaraZ zerrenda:

  EuskaraZ zerrenda[30] 1996an sortu zen eta euskararekin loturiko informazioa trukatzea zuen helburu. Zerrenda horretatik 2000 eta 2001 urteetan idatzitako posta elektronikoko 3.049 mezu (542.866 hitz) jaso genituen. Corpus hau erraz atzigarria da, modu elektronikoan lor baitaiteke, eta errore linguistikoak ditu. Desabantaila bat du, ordea: hizkuntza informalean idatzita dago, eta batzuetan laburtzapenak eta bukatu gabeko hitzak egon ohi dira. Horrek analisia zailtzen du, nahitaez.

[30] http://www.sarean.com/artxiboak/000401.html

- Euskara teknikoko ikasleen testuak:

  Euskal Herriko Unibertsitatean euskara teknikoa izeneko irakasgaiko ikasleen lanak. Lan horiekin 19.391 hitzeko testu-bilduma osatu dugu.

- Karrera bukaerako proiektuak:

  Informatika Fakultateko ikasleek, karrera amaitu ahal izateko, proiektu bat garatu eta honi buruzko txosten bat idatzi behar izaten dute. Euskaraz idatzitako hainbat txosten jaso genituen erroreen bila. Txostenen zuzendariek zuzendu aurretiko bertsioak gorde genituen, akatsak izan zitzatela nahi genuelako. Testu hauetan 305.796 hitz daude.

Hizkuntza-ikasleen testuetan euskara-ikasleen erroreak baldin baditugu ere, gainontzeko hiru testu multzoetan hiztun osoek egiten dituzten erroreak bildu genituen. Beraz, corpus erroredun orokor bat daukagula esan daiteke (Oronoz, 2009). II.2 taulan euskarako corpus erroredunaren osaera ikus daiteke laburbilduta. Kontuan hartu behar da hizkuntza-ikasleen eta euskara teknikoko ikasleen testuetan determinatzaile-erroreak etiketatuta daudela. Horretaz gain, komunztadura-erroreak etiketatu ziren hizkuntza-ikasleen corpusaren 13.591 hitzeko zati batean, II.3.2.2 atalean aipatu dugun moduan.

| | **Hitz kopurua** |
|---|---|
| **Hizkuntza-ikasleen testuak** | 113.290 |
| **EuskaraZ zerrendako testuak** | 542.866 |
| **Euskara teknikoko ikasleen testuak** | 19.391 |
| **Karrera bukaerako proiektuen txostenak** | 305.796 |
| **GUZTIRA** | 981.343 |

Taula II.2: Euskarako corpus erroredunaren osaera.

### II.3.3.2 Euskarako erroreen sailkapena

Euskarako errore mota guztiak identifikatuta eta sailkatuta izatea ezinbestekoa da erroreen tratamendu automatikoa egin ahal izateko. Euskarako erroreen sailkapena finkatzea, ordea, ez da lan erraza. Hain zuzen, prozesu luze baten ondorioz soilik lortu ahal izan genuen euskaraz egiten diren erroreen sailkapen sendoa eta osatua:

1. Hasierako bertsioa garatzeko iturri anitz erabili genituen: hasteko, II.3.3.1 atalean deskribatutako euskarako corpus erroreduna baliatu genuen, eta horretaz gain, zenbait tesi-lanetako (Maritxalar *et al.*, 1997; Gojenola, 2000; Guinovart, 1996a) ikuspuntu zeharo desberdinetatik abiatuta zehaztutako sailkapenak konbinatu genituen; euskararen gramatika aztertu eta ohiko erroreak azaltzen dituen liburu bat ere baliatu genuen (Zubiri eta Zubiri, 1995), eta baita beste zenbait hizkuntzatarako egindako lanak ere (Becker *et al.*, 1999). Gainera, IXA taldeko zenbait hizkuntzalarik eta hainbat euskara-irakaslek ere parte hartu zuten euskarako erroreen sailkapenaren diseinuan.

2. Sailkapena ebaluatzeko asmoarekin, inkesta bat pasa zitzaien 14 adituri, erroreen sailkapenari euren ekarpenak egiteko. Aurkeztutako errore-kategoria bakoitzaren maiztasuna markatzeko eskatu zitzaien adituei. Aztertzen zituzten testuetan errore-kategoria zehatz bat askotan, batzuetan edo gutxitan gertatzen ote zen adierazi behar zuten, eta, horretaz gain, errore motaren bat falta ote zen edo soberan zegoen esan behar zuten. Hiru multzotan banatu genituen inkesta jasotzaileak: egunkari-zuzentzaileak eta editoreak, euskara teknikoko irakasleak, eta euskaltegietako irakasleak. Aditu hauek egindako proposamenekin, erroreen sailkapena osatu eta hobetu genuen.

   Aditu batzuei, egitura batean errorea dagoela jakinda ere, batzuetan ez zitzaien erraza egin errorea sailkapeneko kategoria batean kokatzea. Izan ere, egitura batzuk kategoria batean baino gehiagotan koka daitezke. Kasu horretan ez genuke errore bakarra izango, kategoria adina errore baizik. Oronozek (2009) "*Bonboiak aitor jan du.*" adibidea darabil fenomeno hau azaltzeko. Esaldi horretan zein kategoria esleituko genioke erroreari? Semantikoa? Sintaktikoa? Kontuan hartzekoa da, era berean, esaldia zuzentzat jo daitekeela ikuspuntu literario batetik hartuko bagenu, adibidez. Honek guztiak erroreen tratamendu automatikoak daukan zailtasuna agerian uzten du.

3. Erroreen sailkapena hobeto zehazteko ariketa bat prestatu genuen, eta sei hizkuntzalariri eman genien. 28 esaldi eman zitzaizkien, gutxienez errore bat zutenak. Erroreak detektatu, sailkatu eta zuzentzea zen euren lana. Ondorio interesgarriak lortu genituen: i) hizkuntzalariak, oro har, nahiko konforme zeuden sailkapenarekin; ii) zailtasun handienak errore morfosintaktikoak eta semantikoak sailkatzean aurkitu zituzten;

iii) errore bat baino gehiagoko egiturak kategoria bakar batean sailkatu zituzten normalean; iv) errore askotako esaldietan, esaldi osoa zuzentzera jo zuten, eta ez erroreetako bakoitzari bere kategoria propioa esleitzera (Aldabe *et al.*, 2005a).

Hauek dira lan honen ondorioz sortutako euskarako erroreen sailkapenaren kategoria nagusienak:

1. Ortografikoak: **zuaitz*
2. Lexikoak: **afaltzaile*
3. Gramatika-erroreak:
   (a) Morfologikoa: **gordetu*
   (b) Deklinabidea: **Bakearengandik egin dut.*
   (c) Determinatzailea: **Txokolate nahi dut.*
   (d) Izenordainak: **Bera ikusi da.*
   (e) Adjektiboak eta adberbioak: **Bera onagoa da.*
   (f) Aposizioak: **Zure lagunari, Dublinen bizi dena, sari bat eman diote.*
   (g) Postposizio-lokuzioak: **Zu bidez etorri da.*
   (h) Aditza: **Goaz mendira?*
   (i) Komunztadura: **Gizonak egin dute.*
   (j) Mendeko perpausak: **Bera da gurekin etorri dela.*
   (k) Perpausen egitura: **Jakin dudanez auzokide baten bitartez Udalak dirua eskaintzen du; arrantza motak erabiltzen zirenak; oso erle fina ez baitzen..*
   (l) Juntagailuak eta lokailuak: **Bera baina hobea da.*
   (m) Bestelakoak
4. Semantikoak: *hura* eta *ura* nahastea, adibidez.
5. Puntuazio-markak: komaren erabilpen okerra, edo puntu eta komaren ondoren maiuskula jartzea, kasu.

6. Estilo-kontuak: esaldi luzeegiak erabiltzea, esaterako.

Esan gabe doa kategoria hauek hainbat azpikategoriatan banatu zirela. Uriak (2009) bere tesi-lanean osatu eta aberastu zuen sailkapena, eta bertara jo dezake kategoria-sistema osoaren berri izan nahi duenak.

Tesi-lan honetan, arestian aipatu bezala, azken bi errore-kategoriak landu ditugu: puntuazio-markak eta estilo-kontuak.

#### II.3.3.3 Euskarako erroreen datu-baseak

Corpusetan erroreak etiketatzea ez da nahikoa azterketa sakonak egin ahal izateko; adibide erroredunekin batera, hauen informazio osagarria biltzea ere beharrezkoa da (Uria, 2009). Horretarako, bi datu-base garatu genituen: *Erroreak* eta *Ikasleak*.

*Erroreak* datu-basearen (Arrieta *et al.*, 2003) helburua, esaterako, erroreen inguruko informazio linguistikoa eta teknikoa gordetzea da, erroreen tratamendu automatikoaren alorrean gramatika- eta estilo-zuzentzaile sendo bat garatzeko. Datu-base honetan, beraz, errore-adibidea eta dagozkion zuzenketa posible guztiak gordetzen dira, eta baita errore-adibideari dagokion kategoria ere (II.3.3.2 atalean ikusitako sailkapenean oinarrituta). Horretaz gain, informazio teknikoa gordetzen da; esaterako, errore bakoitza detektatzeko zein teknika erabil daitekeen. Datu-base honen helburua, hain zuzen, erroreen tratamendu automatikoa bideratzea da, datu linguistiko eta tekniko hauetan oinarrituta.

*Ikasleak* datu-basea (Aldabe *et al.*, 2005a), berriz, euskara-ikasleen tarte-hizkuntzari buruzko informazio psikolinguistikoa gordetzeko darabilgu; alegia, ikasleek egindako erroreen diagnosi linguistikoa eta ikasleen inguruko informazio psikolinguistikoa gordetzeko. Informazio linguistikoaz gain, beraz, bestelako informazioa ere gordetzen da bertan: testuari buruzko datuak (zenbat hitzeko testua den, zein ariketa motari dagokion...), errorea eragin ahal izan du(t)en sakoneko arrazoia(k) eta ikasleari berari buruzko informazioa (datu pertsonalak, hizkuntza maila, ama-hizkuntza...). Datu-base honetan gordetako datuek informazio-iturri aberatsa osatzen dute euskara-ikasleen tarte-hizkuntza —eta, bide batez, hizkuntzaren ikasketa-prozesua— ezagutu ahal izateko. Horrela, bertan jasotzen den informazioa baliagarria da, esate baterako, euskararen ikasketa- eta irakaskuntza-prozesuaren alorrean ekarpenak egiteko, ikasle bakoitzaren beharren edota zailtasunen araberako laguntza-tresnak garatu ahal izateko eta abar.

Bi datu-base hauek kontsultatzeko, edo txertaketak eta aldaketak egiteko, web-interfaze bana sortu genituen: *Erreus*, *Erroreak* datu-basearentzat; eta *Irakazi*, *Ikasleak* datu-basearentzat. Datu-baseetan, gaur egun, 1103 errore erreal daude, baina corpusetatik —esportazio prozesu baten bidez— gehiago sartzea aurreikusita dago etorkizun hurbilean.

#### II.3.3.4 Euskarako erroreen detekzioa ikasketa automatikoa erabiliz

Dagoeneko esan dugun moduan, euskarako erroreen detekzioa hizkuntza-ezagutzaren ikuspegitik landu da orain arte IXA taldean. Tesi-lan honetan, aitzitik, corpusean oinarritutako teknikak baliatu nahi izan ditugu eginkizun berbererako. Ikasketa automatikoko teknikak baliatzeko, ordea, corpus handi samarra izan beharra dago, ikasi nahi den kontzeptu hori bera etiketatua duena, gainera, eta gaur egun bilduta dugun euskarako erroredun corpusak milioi bat hitz inguru baldin baditu ere, testu guztiak ez dira kalitate berekoak, II.3.3.1 atalean ikusi dugun moduan. Gainera, corpus honetan ez ditugu identifikatuta errore mota guztiak; komunztadurari eta determinatzaileei dagozkien erroreak soilik etiketatu ziren, eta ez corpus osoan, zati batean baizik, arestian aipatu dugun eran.

Zehazki, 113.290 hitzeko hizkuntza-ikasleen corpusean, 788 determinatzaile-akats identifikatu ziren (horietatik 374 (% 47,46) behe-mailako testuetan, 244 akats (% 30,97) erdi-mailako testuetan eta 170 (% 21,57) goi-mailakoetan). Bestalde, 458 komunztadura-errore etiketatu ziren 13.591 hitzeko beste zati batean. Hau guztia kontuan harturik, ikasketa automatikoko teknikak lantzeko corpus etiketatu txikiegia dugula iruditzen zaigu. Hipotesi hau baieztatzeko, dena dela, proba txiki bat egin genuen. Komunztadura-erroreak eta determinatzaileenak detektatzeko saio batzuk egin genituen *Weka*[31] (Witten eta Frank, 2005) paketeko hainbat ikasketa-algoritmorekin, baina lortutako emaitzak —aurreikusi bezala— ez ziren onak izan (Cermeno, 2008): urrun zeuden tresna fidagarriak sortzeko behar izaten diren emaitzetatik.

Landu nahi genuen beste kontu baterako, ordea, eskura geneukan nahi adina corpus, dagoeneko aipatu dugun moduan: komak zuzentzen ikasteko, hain zuzen. Estilo- eta puntuazio-zuzentzailea —zeina gramatika-zuzentzailearen baitan txertatu nahi baita— garatzeko bidean, oso gai delikatua eta era berean esanguratsua iruditzen zitzaigun komaren auzia. Hortaz, pentsatu

[31] *Datu-meatzaritza*ko atazetarako erabiltzen diren ikasketa automatikoko algoritmoen multzoa da WEKA (http://www.cs.waikato.ac.nz/ml/weka/).

genuen ikasketa automatikoko teknikak erabil genitzakeela komak zuzentzeko. Horretarako, komak zuzen etiketatuta zituen corpus handi bat behar genuen; hots, egunkarietako, liburuetako, aldizkarietako testuak balia genitzakeen, baldin eta testu horietan komak zuzen jarrita zeudela suposatzen bagenuen. Hipotesi hori geure egitea zen kontua: "egunkarietan, aldizkarietan edota liburuetan jarrita dauden komak zuzenak dira". Hipotesi hori onartzeak nahi beste ikasketa-corpus lortzea zekarren, eta halaxe egin genuen. IV. kapituluan komari buruz kontatzen diren esperimentu gehienak hipotesi horretatik abiatuz egin ziren. Honek baditu bere mugak. Izan ere, ez dakigu ikasten ari garena zenbateraino datorren bat ikasi behar genukeenarekin. Hau da, aukeratutako corpuseko jatorrizko komak okerrak izan daitezke, kontrakoa suposatu arren. Hori dela eta, ebaluazio kualitatibo bat egin behar izan genuen, IV. kapituluan ikusiko dugun moduan. Kapitulu horretan bertan azalduko ditugu komaren ikasketaren ebaluazioari buruzko kontuak, modu xeheago batean.

Ikasketa-corpus txikia den kasuetarako, II.3.2.2 atalean aipatutako irtenbideren batetik jotzea aurreikusten dugu etorkizunean: ikasketa bizia (*active learning*) delakoa erabiltzea, edo errore artifizialak automatikoki sortzea, edota ikasketa erdi-gainbegiratua egitea. Tesi-lan honetatik kanpo geratu diren lanak dira hauek, halabeharrez.

## II.4 Ondorioak

Kapitulu honetan, azaleko sintaxiaren tratamendu automatikoan eta erroreen detekzioan aritzeko egindako azterketa eta sortutako oinarrizko baliabideak aurkeztu ditugu, eta orain arte IXA taldean arlo hauetan egindako lanen artean kokatu dugu tesi hau. Nagusiki ikasketa automatikoa baliatu dugunez, teknika honekin arlo horietan egin dena eta egin daitekeena aztertu dugu. Azkenik, euskarako erroreen detekziorako sortutako oinarrizko baliabideak aurkeztu ditugu: euskarako erroreen corpusa eta erroreen sailkapena, besteak beste. Daukagun corpus erroreduna, ordea, txikia ez bada ere, errore mota bakoitzeko bertan ditugun errore-adibideak gutxitxo dira ikasketa automatikoko teknikak baliatuz erroreen detekzioa jorratzeko. Aitzitik, badira akats batzuk erroreak esplizituki etiketatua izatea behar ez dutenak; koma da horietako bat, eta horregatik, tesi-lan honetan, komak zuzentzen saiatu gara ikasketa automatikoko tekniken bidez.

Aurrez, ordea, azaleko sintaxiaren baitan teknika hauekin jorratutako

kate- eta perpaus-identifikatzaileak aurkeztuko ditugu (III. kapituluan). Bi tresna hauek lortzen duten informazioa, gainera, erabilia izango da, gerora, koma-zuzentzailean (ikus IV. kapitulua).

# III. KAPITULUA

## Azaleko sintaxiaren tratamendua euskaraz: kateen eta perpausen identifikazioa

Kapitulu honetan, euskararen azaleko sintaxiaren tratamendu automatikoan egin ditugun aurrerapenak aurkeztuko ditugu. Zehatzago esanda, kateen eta perpausen identifikazio automatikoan egindako lana azalduko dugu.

90. hamarkadaren hasieratik, azaleko sintaxiaren tratamendu automatikoa burutzeko hainbat hurbilpen erabili dira, dagoeneko aipatutako bi multzo hauetan sailka daitezkeenak: hizkuntza-ezagutzan oinarritutako teknikak (edo teknika sinbolikoak), eta corpusean oinarritutako teknikak (edo teknika enpirikoak).

Teknika batzuekin zein besteekin, sintaxiaren tratamenduan egindako lan esanguratsuenen deskribapen zabal bat dakar Molina-k (2003), bere tesi-txostenean. Azaleko analisi sintaktikoaren atazaren bat landu dutenen artean, hauek aipatzen ditu, besteak beste: MBL[1] algoritmoan oinarritutako (Daelemans *et al.*, 1999) lana, *Winnow* algoritmoan oinarritutako lanak (Li eta Roth, 2001; Zhang *et al.*, 2002), zenbait sailkatzaile konbinatzen dituen *boosting* algoritmoan oinarritutako lana (Carreras eta Màrquez, 2001) eta SVM algoritmoan oinarritutako lana (Kudo eta Matsumoto, 2001).

II. kapituluan aipatutako moduan, IXA taldean ere egin da lanik euskarako azaleko analizatzaile automatiko fidagarri bat lortzeko helburuarekin. (Arriola, 2000) eta (Aranzabe, 2008) lanetan, besteak beste, kateen —sintagmen eta aditz-kateen— identifikazio automatikoa landu zen. Adu-

[1] *Memory Based Learning* edo memorietan oinarritutako ikasketa.

rizen (2000) lanean, aldiz, perpausak eta esaldiak[2] detektatzeko lehen saioa egin zen, eta gerora ere saio hura hobetu duten hainbat ahalegin egin dira (Aduriz *et al.*, 2006c), betiere erregeletan oinarritutako tekniken bidez.

Bestalde, literaturako zenbait lanetan zehazten den moduan, hizkuntza-ezagutzan oinarritutako teknikek eta corpusean oinarritutako hurbilpenek lortzen dituzten emaitzak hobe daitezke, baldin eta bi hurbilpenak konbinatzen badira (Muresan *et al.*, 2001; Bigert eta Knutsson, 2002; Ayan *et al.*, 2004; Sjöbergh, 2005; Zribi *et al.*, 2007; Aranzabe, 2008; Oronoz, 2009; Cendejas *et al.*, 2009). Kapitulu honetan, beraz, corpusean oinarritutako hurbilpenak lantzeaz gain, hauek hurbilpen linguistikoekin nola konbinatu ditugun azalduko dugu.

Hala, ikasketa automatikoa baliatuz, eta erregeletan oinarritutako teknikekin uztartuz, bi ataza burutu ditugu, sintaxiaren azaleko analisi automatikoaren baitan koka daitezkeenak:

- **Kateen identifikazio automatikoa**: sintagmak eta aditz-kateak hartu ditugu kate gisa.

- **Perpausen identifikazio automatikoa**: perpaus bakunez gain, esaldi konposatuen perpaus bakoitza identifikatzea da helburua.

Kapitulu honetan, hortaz, lehenik eta behin, azaleko sintaxiaren tratamendu automatikoak dakartzan onurak aztertuko ditugu sarrera gisa, eta analisi sintaktiko osoarekin konparatuko dugu, bataren eta bestearen abantailak eta desabantailak azalduz (III.1 atala). Ondoren, literaturan alor honetan —kateen eta perpausen identifikazioan— egin diren ikerketak laburbilduko ditugu (III.2 atala). Gero, *pertzeptroi*etan[3] oinarritutako *FR-Perceptron*[4] ikasketa automatikoko algoritmoa azalduko dugu (III.3 atalean), horixe izan baita erabili duguna euskarako kateen eta perpausen identifikazioan. Ondoren, egindako esperimentuetan erabilitako oinarrizko baliabideez eta ebaluazio-moduaz arituko gara, III.4 atalean. Horren ostean, egindako saioak azalduko ditugu, bai kateen identifikazioan (III.5 atala), bai perpausen identifikazioan (III.6 atala). Laburpen gisako batekin eta lortutako emaitzen interpretazioarekin bukatuko dugu kapitulu hau (III.7 atala).

[2]Hemendik aurrera, perpausen identifikazioa egingo dugula esango dugu laburtzeko, baina, egiazki, perpausen eta esaldien identifikazioaz ariko gara.

[3]II. kapituluan ikusi dugun moduan, neurona-sare artifizial moduko bat da *pertzeptroi*a; sailkatzaile lineal gisa erabiltzen da. Informazio gehiagorako, ikus II.1.5.3 azpiatala.

[4]*Filtering and Ranking with Perceptrons* (*iragazketa eta sailkapena, pertzeptroiekin*).

## III.1 Sarrera

Edozein hizkuntzatan idatzitako esaldi baten analisi sintaktiko osoa egitea, labur esanda, esaldi horri dagokion zuhaitz sintaktikoa lortzea da. Horretarako, eskuarki, prozesatu nahi dugun hizkuntzaren egitura sintaktikoa deskribatuko duen gramatika bat erabiltzen da, eta, horrekin batera, algoritmo bat definitzen da, zeinak esaldiaren zuhaitz sintaktikoa zein den erabakiko baitu, aurretik definitutako gramatikan oinarrituz sortutako bilaketa-estrategia baten bidez.

Analisi sintaktiko osoa egiten duten algoritmoek, alta, esaldiari dagokion zuhaitz-egitura lortu ahal izateko, esaldi horrek —bistan da— gramatikak definitutako lengoaiaren parte izan beharko du; bestela, analisi-prozesuak ez du emaitzarik emango. Allen-en (1995) liburuan, analisi sintaktiko osoa egiten duten zenbait algoritmo *klasiko*ren nondik norakoak azaltzen dira. Hauek emaitza onak lortzen dituzte lengoaia mugatuekin, hots, domeinu espezifikoko lengoaiekin, non testuinguru lexikoa eta semantikoa lokalagoa izan ohi den; hizkuntza arruntekin, ordea, hainbat analisiren artetik analisi sintaktiko bakarra —zuzena dena, noski— aukeratzearen arazoa azaleratzen da, hizkuntzaren maila orotan agertzen den anbiguotasuna dela-eta (Molina, 2003); anbiguotasun hau ebazteko, kasu batzuetan, informazio semantikoaren beharra ere izaten da. Horregatik, halakoetan, ez dituzte emaitza onak lortzen analisi sintaktiko osoa egiten duten algoritmoek (Civit, 2003).

Gainera, HPko aplikazio guztiek ez dute analisi sintaktiko osorik behar. Hori dela eta, azken urte hauetan, ikerketa ugari egin da azaleko sintaxiaren analizatzailea edo analizatzaile sintaktiko partziala —*shallow* edo *partial parser* (Motkhtar *et al.*, 2002) izenarekin ere ezagutzen dena— lantzeko helburuarekin. Mota honetako analizatzaileek informazioaren zati bat, ez guztia, aztertzen dute. Hara, nola definitzen duen Abney-k (1997) analisi sintaktiko partziala:

> "*[...] mugatu gabeko testu batetik modu eraginkor eta fidagarri batean informazio sintaktikoa lortzen duen analisi-teknika bat da, zeinak analisi osoaren sakontasuna eta osotasuna baztertzen dituen*"

Molina-k (2003) honela borobiltzen du aurreko definizioa:

> "*[...] testu batetik ahalik eta informazio sintaktiko gehien lortzen duen tresna sendoa da azaleko analizatzaile sintaktikoa, nahiz eta ez den zuhaitz sintaktiko osoa emateko gai izango*"

Berak dioenez, analizatzaile osoaren emaitza zuhaitz sintaktiko osoa den moduan, azaleko analizatzaileak elkar lotu gabeko *azpizuhaitzak* emango ditu. Osagaion arteko loturak, hala eta guztiz ere, *a posteriori* egin litezke, heuristikoak, eredu probabilistikoak eta abar aplikatuz, baldin eta aplikazioak halakorik beharko balu, betiere. Gainera, analisi sintaktiko osoak ematen duen informazio zehatza galdu arren, analisi partzialak eraginkortasun eta fidagarritasun handiagoz egiten dio aurre edozein testu erreal analizatzeko erronkari.

(Li eta Roth, 2001) lanean, hain zuzen, analizatzaile sintaktiko partzial bat (Punyakanok eta Roth, 2001) eta oso bat (Collins, 1997) konparatzen dira, eta analizatzaile sintaktiko partzialak doitasun eta estaldura handiagoa lortzen du (% 94.64 vs % 91.96, $F_1$ neurrian[5]) eta gainera sendoagoa da. Ondorio horietara iristeko, *Wall Street Journal* corpusa erabiliz entrenatu ziren bi analizatzaileak eta bi corpus baliatuz ebaluatu ziren biak ala biak: *Wall Street Journal* bera, eta *Switchboard* corpusa, zeinak errore sintaktikoak baititu (telefonozko elkarrizketen transkripzioak baitira). Bai corpus batekin, bai bestearekin ebaluatuta, bi-bietan emaitza hobeak lortu zituen azaleko analizatzaileak. Aipagarria da, era berean, corpus erroredunean ebaluatutakoan, bi analizatzaileen emaitzek txarrera egin arren, beherakada handiagoa izan zutela analizatzaile osoaren emaitzek, analizatzaile partzialarenak baino. Horrek adieraz lezake sendotasun handiagoa luketela analizatzaile sintaktiko partzialek, kalitate gutxiko testuekin —hala nola, ahozko elkarrizketen transkripzioak, errore gramatikalak dituzten testuak...— lan egiterakoan (Molina, 2003).

Beraz, analisi sintaktiko partzialaren ideia nagusia, laburbilduz, informazio morfosintaktikoa abiapuntutzat hartuta testuak *hitz multzo*tan zatitzean datza —esaldiko *hitz multzo*ak identifikatzea, alegia—, analisi osoa hurrengo urratserako utziz. Gerora jorratu nahi den arloaren arabera, era bateko ala besteko *hitz multzo*en identifikazioa landuko da: kateak, perpausak, entitateak... Edo guztiak. Analisi partziala, hortaz, esaldiaren analisi osoa lortzeko aurreko urrats gisa ere uler daiteke. Izan ere, esaldiak *hitz multzo*en segida gisa mugatuta izanik, gutxiago dira elkarrekin lotu beharreko unitateak, eta ondorioz anbiguotasuna gutxitu eta bilaketa-eremua murriztu egiten da (Collins, 1996). Gainera, corpus handi bat eskuz etiketatu nahi bada ere —sintaxi mailan—, azaleko analizatzailea erabil liteke aurre-urrats

[5]Doitasunaren eta estalduraren batezbesteko moduko bat da $F_1$ neurria. Jo III.4.2 atalera, neurri hauen azalpen zehatzak irakurtzeko.

moduan, hizkuntzalariei lana aurreratzeko edo errazteko gailu gisa. Modu honetan etiketatu ziren, hain zuzen, *Wall Street Journal* corpusa —*Penn Treebank* (Marcus *et al.*, 1993) proiektuaren baitan—, eta Cat3LB, Cast3LB eta Eus3LB —katalanerako, gaztelaniako eta euskarako corpusak, hurrenez hurren—, 3LB (Palomar *et al.*, 2004) proiektuaren baitan.

Horretaz gain, analizatzaile partzialek ematen duten informazio sintaktikoa baliagarria da HPko hainbat aplikaziotarako: informazioaren berreskurapena, informazioaren erauzketa, laburpenen sorkuntza eta galde-erantzun sistemak, besteak beste. (Srihari eta Li, 1999; Vicedo, 2002) lanetan, azaleko analizatzaileak erabili izan dira corpus handiak aztertzeko eta oinarrizko erlazio sintaktikoak baliatuz euren helburuak lortzeko. Izan ere, batzuetan, nahikoa da zenbait elementu sintaktiko edota semantiko aurkitzea: agentea, objektua, lekua edo denbora, esate baterako. *Verbmobil*[6] (Wahlster, 2000) izeneko proiektuan, adibidez, analizatzaile partzialak baliatu ziren itzulpen-sistema sendotzeko.

*Hitz multzo*z osatutako errepresentazio sintaktikoa, hortaz, oso erabilgarria den tarteko errepresentazioa dela esan genezake —bai eraginkortasunarengatik, eta baita fidagarritasunarengatik ere—, corpus handiak analizatu behar dituzten aplikazioentzat.

Laburbilduz, HParen helburu konkretuaren mende dago analizatzaile sintaktiko automatikoek sortu beharreko analisien xehetasun edota sakontasun maila (Aranzabe, 2008). Beste modu batean esanda, analisi sintaktiko horrek osoa edo azalekoa behar duen, azken helburuaren mende dago. Akatsik gabeko hizkuntzaren prozesamendua egiterik bagenu —emaitza perfektuak lortuko lituzkeena, alegia—, analisi sintaktiko osoa egitea litzateke egokiena, esaldien egitura sintaktikoak oso-osorik eta ahalik eta zehaztasun handienarekin izango genituzkeelako. Agertoki hori, ordea, utopikoa da, dakigun moduan. Izan ere, analisi sintaktiko automatiko osoan, hainbat arazo sortzen dira, arestian ikusi dugun gisan. Sarri, egokiagoa izaten da, beraz, analisi sintaktiko partziala egitea, bai aurre-urrats gisa, edo baita behar adina informazio eskaintzen digulako, osoak baino zehaztasun eta sendotasun handiagoarekin, gainera.

[6] http://www.dfki.de/pas/f2w.cgi?iuic/verbmobil-e

## III.2 Testuingurua

Azken urteetan, kateen eta perpausen identifikazioan, ikasketa automatikoko teknikak erabili dira gehienbat. Hala eta guztiz, hizkuntza-ezagutzan oinarritutako hurbilpenak ere baliatu izan dira —eta oraindik baliatzen dira—. (Abney, 1991) eta (Aït-Mokhtar eta Chanod, 1997) lanak dira, seguru asko, mugarriak, arlo honetan, teknika hauen erabilpenean: hurrenez hurren, testuingururik gabeko gramatikak (CFG) eta egoera finituko transduktoreak (FST) darabiltzate kateen identifikaziorako eta analisi sintaktiko partzialerako. Gala-k (1999) espainierako azaleko analizatzaile sintaktikoa sortzeko baliatu zuen egoera finituko transduktoreen hurbilpena, eta (Atserias *et al.*, 2006) lanean, testuingururik gabeko gramatikak erabili ziren, besteak beste, katalanerako eta espainierarako.

Azken urteetako joera, ordea, ikasketa automatikoko teknikak erabiltzekoa izan da, II. kapituluan ikusi dugun moduan. CoNLL batzarrean antolatutako ataza partekatuetan —kateen identifikazioa 2000. urtean, eta perpausen identifikazioa 2001. urtean— ikasketa automatikoko sistemek, esate baterako, nagusitasun ia erabatekoa izan zuten, eta emaitza onak lortu zituzten (Sang eta Buchholz, 2000; Sang eta Déjean, 2001): $F_1$ neurrian % 94 inguruko emaitzak lortu zituzten kateen identifikazioaren eginkizunean, eta % 84 ingurukoak perpausen identifikazioarenean.

Ikasketa-corpus bera baliatu zelarik bi aldiotan, kateen identifikazioan lortzen diren emaitza hobeak modu batean soilik arrazoi daitezke: perpausak baino errazagoa da kateak automatikoki detektatzea. Baina galdera, orduan, beste bat litzateke: zergatik da perpausak detektatzea kateak detektatzea baino zailagoa? Galdera hau erantzuteko, kateak eta perpausak nola definitzen diren aztertu beharko genuke, eta horixe da hurrengo ataletan ikusiko duguna.

### III.2.1 *Hitz multzo*ak: definizio formal bat kateen eta perpausen deskribapenerako

Kateen eta perpausen identifikazioa beste hainbat zereginekin batera tratatu izan da zenbaitetan, zeregin orokor baten parte kontsideratuz: esaldi batean egitura sintaktiko partzial batzuk identifikatzea, hain zuzen ere. Beste modu batean esanda, *hitz multzo*ak identifikatzean datzan zeregin orokor baten parte lirateke, bai kateen identifikazioa, eta baita perpausen identifikazioa

ere (Carreras, 2005). Mota desberdineko *hitz multzo*ak dira, ordea, kateak eta perpausak, jarraian ikusiko dugun moduan. Ñabardura batek bereiziko ditu, baina ñabardura garrantzitsua, ezbairik gabe.

(Carreras, 2005) lanean deskribatutako *hitz multzo*en[7] definizioa da, hain zuzen, tesi honetan gure egin duguna. Laburbilduz, *hitz multzo* gisa hartzen da elkarren segidan doazen hitzen edozein sekuentzia, zeinak bi propietate betetzen dituen: *hitz multzo* batek beste *hitz multzo* bat hartu ahal izango du bere baitan, baina *hitz multzo* bat ezingo zaio beste bati gainjarri. Kateek eta perpausek, besteak beste, propietate hauek betetzen dituzte, eta *hitz multzo*ak izango dira, beraz.

**Adibidea III.2.1**
"*Krisia, antzinatik bazetorren ere, abenduaren 20an areagotu zen.*" esaldian, hauek lirateke sintagmei, aditz-kateei eta perpausei dagozkien *hitz multzo*ak:

- "*Krisia*", "*antzinatik*" eta "*abenduaren 20an*" sintagmak.
- "*bazetorren*" eta "*areagotu zen*" aditz-kateak.
- "*antzinatik bazetorren ere*" perpausa, eta "*Krisia, antzinatik bazetorren ere, abenduaren 20an areagotu zen.*" esaldia[8]. Azken kasu honetan ikus daitekeen moduan, *hitz multzo* bat beste baten baitan joan daiteke.

Formalki, honela definitzen ditu Carreras-ek (2005) *hitz multzo*ak:

Izanik $x$ esaldi bat, zeina $n$ hitzez osatua dagoen $[x_1, x_2, \ldots, x_n]$; izanik $\mathcal{K}$, *hitz multzo*en kategoria posibleak[9]:

$\mathcal{P}$ *hitz multzo*ak —$(s, e)_k$ moduan adierazia, $s$ izanik *hitz multzo*aren hasiera (*start*) eta $e$ bukaera (*end*)— adierazten du $x_s$ hitzetik $x_e$ hitzera doazen elkarren segidako hitzen sekuentzia, $s <= e$ izanik, eta $k \in \mathcal{K}$. Formalki:

$$\mathcal{P} = \{(s, e)_k | 1 <= s <= e, k \in \mathcal{K}\}$$

*Hitz multzo*ak detektatzeko ataza batean, beraz, $x$ esaldiarentzat, soluzio egoki bat izango da *hitz multzo*en sekuentzia bat ($y$), zeina koherentea izango den murrizketa batzuekiko.

Lehenengo murrizketa hauxe da: soluzioko *hitz multzo*ak ezin dira elkarren artean gainjarri. Formalki, bi *hitz multzo* —$p_1 = (s_1, e_1)$ eta $p_2 =$

[7] *Phrase* gisa definitzen ditu Carreras-ek (2005).

[8] Esaldiak ere perpaustzat hartuko ditugu, problemaren ebazpenean bi kontzeptu horiek desberdintzeak ez baitu ezer berririk ekartzen.

[9] Aurrerago ikusiko dugun moduan, euskararen kasuan, sintagmak eta aditz-kateak izango dira kateen kategoriak; perpausen kasuan, perpausa soilik izango dugu, ez baititugu bereiziko perpaus mota desberdinak.

$(s_2, e_2)$— gainjarri direla esango dugu, baldin $s_1 < s_2 <= e_1 < e_2$ edo $s_2 < s_1 <= e_2 < e_1$; laburtzeko, honela adieraziko dugu $p_1$ eta $p_2$ gainjarri egiten direla: $p_1 \sim p_2$. Gainjarritako bi *hitz multzo* honela adieraziko lirateke grafikoki:

$$(s_1 \qquad (s_2 \qquad )e_1 \qquad )e_2$$

Bigarren murrizketak *hitz multzo* batek beste bat *bere baitan har* dezakeen edo ez adierazten du; alegia, *hitz multzo hierarkikoak* edo *errekurtsiboak* onar daitezkeen ala ez. Aurreko $p_2$ *hitz multzo*ak $p_1$ *hitz multzo*a *bere baitan hartzen* duela esango dugu, baldin $s_2 <= s_1 <= e_1 <= e_2$; laburtzeko, honela adieraziko dugu $p_2$ *hitz multzo*ak $p_1$ *hitz multzo*a hartzen duela bere baitan: $p_1 \preceq p_2$.

Bi *hitz multzo* errekurtsibo honela adieraziko lirateke grafikoki:

$$(s_2 \qquad (s_1 \qquad )e_1 \qquad )e_2$$

Lehen murrizketa soluzio posible guztiek bete beharrekoa bada ere (ez dira onartuko gainjartzen diren *hitz multzo*ak), bigarrenak esku artean daukagun ataza bi azpi atazatan banatzea dakar: *hitz multzo*ak bata bestearen baitan onartzen dituzten atazak (*hitz multzo errekurtsiboak*), eta onartzen ez dituztenak (*hitz multzo jarraituak*).

Beste modu batean esanda, *hitz multzo jarraitu*en identifikazioko atazetan, *hitz multzo*ak elkar ezin gainjartzeaz gain, *hitz multzo*ek ezin dituzte beren baitan beste *hitz multzo*ak hartu. Honela definituko litzateke ataza mota hauen soluzioen espazioa, formalki:

$$\mathcal{Y} = \{y \subseteq \mathcal{P} | \forall p_1, p_2 \in y(p_1 \not\sim p_2 \wedge p_1 \not\preceq p_2)\}$$

*Hitz multzo errekurtsibo*en atazetan, berriz, *hitz multzo*ek beren baitan beste *hitz multzo*ren bat har dezakete, baina ezingo dira elkar gainjarri, halere. Formalki, honela definituko litzateke ataza mota hauen soluzioen espazioa:

$$\mathcal{Y} = \{y \subseteq \mathcal{P} | \forall p_1, p_2 \in y(p_1 \not\sim p_2)\}$$

Kateak, beraz, *hitz multzo jarraituak* izango dira: ezin dira gainjarri eta ez dira errekurtsiboak. Perpausak, aldiz, *hitz multzo errekurtsiboak* dira: ezin dira gainjarri, baina perpaus batek beste bat har dezake bere baitan.

## III.2.2 Kateak

Ikasketa automatikoa darabilte azken urteetan kateen identifikazio automatikoan egin diren ikerketa gehienek (Sang eta Buchholz, 2000). Euskarako orain arteko hurbilpenek, ordea, erregela bidezko formalismoak darabiltzate. Lan hauetatik harago joan nahi izan dugu tesi-lan honetan, eta ikasketa automatikoa baliatu dugu euskarako kateen identifikazio automatikorako; ez hori bakarrik: erregela bidezko teknikak ikasketa automatikokoekin konbinatu nahi izan ditugu, III.5.2 atalean ikusiko dugun moduan. Atal honetan, berriz, kateen identifikazio automatikoan guretzat mugarri izan diren lanak aztertuko ditugu; hots, gure ikerketak nondik abiatu diren azalduko dugu.

### III.2.2.1 Kateak: hurbilpen linguistikoa

Hurbilpen linguistikoa jorratu zuten lanentzat, mugarria izan zen Abney-k (1991) egin zuen kateen definizioa, bat datorrena, gainera, III.2.1 atalean ikusi dugun definizio formalarekin (Aduriz *et al.*, 2006a):

> *"[...]* Chunk *terminoa Abney-ri zor zaio(Abney, 1991). Horren ordez, esan bezala* kate *erabiliko dugu.* Katea *sintagma kategoriako zatia da eta, sintaktikoki erlazionaturiko hitzez osatua dago. [...] Horrela, bada, testua* kate*etan zatitzea gainjartzen ez diren eta elkarrekin sintaktikoki erlazionaturik dauden* hitz multzo*ak atzematean datza.* Hitz multzo *horiek, beraz, ez-errekurtsiboak izango dira, hau da, ezin dute beren baitan beste* hitz multzo*rik edota katerik izan."*

Definizio honi jarraiki, etiketen zerrenda jarraitu baten moduan errepresenta daitezke esaldia osatzen duten kateak.

Hizkuntza bakoitzak, halere, bere berezitasunak ditu, eta Abney-ren (1991) definizioa bakoitzak berera ekarri behar izan du nolabait. Gaztelaniaren kasuan, adibidez, hala egin zen (Civit, 2003), gaztelaniak ingelesarekiko zituen berezitasunak zirela-eta. Hain zuzen ere, gaztelaniaz adjektiboek izenarekin —bai generoan, bai numeroan— daukaten komunztadura dela eta, biak sintagma berean bateratu daitezkeela ebatzi zuen Civit-ek (2003), baldin eta *auzokidetasun-erlazio* bat baldin badute. III.2.2 adibidean, esaterako, kateen banaketa desberdina dela ikus daiteke esaldi beraren ingeleseko eta gaztelaniako itzulpenetan. "*proud*" hitzak berak bakarrik kate bat osatzen du ingelesekoan, eta, aldiz, bere pareko "*orgulloso*" hitza "*un hombre*" hitzekin elkartuta dator kate bakar batean gaztelaniako itzulpenean, aipatu berri dugun *auzokidetasun-erlazioa* dela-eta.

**Adibidea III.2.2**

1. *(a man) (proud) (of his son)*

2. *(un hombre orgulloso) (de su hijo)*

### III.2.2.2 Euskarako kateen identifikazioa hurbilpen linguistikoa erabiliz

Orain arte, IXA taldean, erregela bidezko hurbilpenen bitartez landu da kateen identifikazio automatikoa. Honela, Arriola (2000) lanean eman ziren lehen urratsak zentzu horretan. Lan haietan, CG formalismoa erabiliz, sintagmak eta aditz-kateak identifikatzen saiatu ziren, beste zenbait gauzaren artean. 2008an aurkeztutako tesi-lanean, Aranzabek (2008) aurretik zeuden erregela horiek hobetu eta osatu zituen, formalismo bera erabiliz.

Horretarako, (Aduriz *et al.*, 2006a) artikuluan laburbilduta datorren irizpidea (III.2.1 atalean azaldutako kateen deskribapen formalarekin bat datorrena) hartu zen aintzat. Euskarak ere, ordea, baditu bere berezitasunak, eta, oro har, aipatutako kateen definizioa baliatu izan bada ere, egin behar izan da moldaketarik. Esate baterako, preposizioen ordez postposizio-sistema dauka euskarak, dakigun moduan. Alegia, gaztelaniaz edo ingelesez preposizioekin aditzera ematen dena, euskaraz postposizioekin adierazten da (postposizio-lokuzioekin edo deklinabide-kasuekin). Adibidez:

**Adibidea III.2.3**

1. *Sus reflexiones sobre la poesía universal [...]*
   *Poesia unibertsalari buruzko bere gogoetak [...]*

2. *La casa de mi amigo [...]*
   *Nire lagunaren etxea [...]*

III.2.3 adibideko lehen esaldian ikus daitekeen moduan, gaztelaniako "*sobre*" preposizioari, euskarako "*buruzko*" postposizioa dagokio. Bigarren esaldian, berriz, "*de*" preposizioaren ordain gisa, euskarakoan, "*noren*" deklinabide-kasua erabiltzen da.

Euskararen berezitasun honek, jakina, moldaketa batzuk eskatzen ditu. Gaztelaniaz, adibidez, erregela bidezko kateen identifikazioa lantzerakoan, preposizio-sintagmak eta izen-sintagmak bereizi egiten dituzte (Civit, 2003). Guztira, bost kate mota identifikatzen ahalegintzen dira, III.2.4 adibidean ikus daitekeen moduan: izen-sintagmak (NP), preposizio-sintagmak (PP), adjektibo-sintagmak (ADJP), adberbio-sintagmak (ADVP) eta aditz-kateak (VP).

**Adibidea III.2.4**

1. $(Sus\ reflexiones)_{NP}$ $(sobre\ la\ poesía\ universal)_{PP}$ $(dejaron)_{VP}$ $(boquiabiertos)_{ADJP}$ $(a\ todos)_{PP}$.
2. $(La\ casa)_{NP}$ $(con\ vistas)_{PP}$ $(de\ mi\ amigo)_{PP}$ $(es)_{VP}$ $(estupenda)_{ADJP}$.
3. $(Ciertamente)_{ADVP}$ $(se\ salió)_{VP}$ $(con\ la\ suya)_{PP}$.

Buru-lehen diren hizkuntzetako preposizioen ordez, euskarak, buru-azken izanik, postposizioak behar dituela aski kontu argia da. Horregatik, bestelako forma bat hartzen dute izen-sintagmek, eta genitiboen eta postposizio-lokuzioen bidez egindako loturak sintagma bakartzat hartzen dira, eta hala hartu zituen Aranzabek (2008) ere, bere tesi-lanean erregela bidez sintagmak detektatzeko egin zuen ahaleginean. Genitiboarekin eta postposizio-lokuzioekin osatutakoez gain, koordinazio batzuk ere sintagma bakartzat hartu zituen. Deklinabideko kasu-marka bera daramaten eta hitz bakar bat duten sintagmen koordinazioa soilik hartu zuen, hain zuzen ere, sintagma bakartzat: "*udaberrian eta udazkenean*".

Hala, euskaraz, bi kate —*chunk*— mota soilik kontsideratu zituen Aranzabek (2008): sintagmak eta aditz-kateak. Eta sintagmen baitan sartu zituen preposizio-sintagmak (postposizionalak, alegia, euskararako), adjektibo-sintagmak eta adberbio-sintagmak.

Ingeleserako, berriz, are unitate txikiagoa kontsideratu zuten katea, 2000. urteko CoNLL batzarreko zeregin partekatuan. Azal dezagun adibide batekin:

**Adibidea III.2.5**

$(He)_{NP}$ $(reckons)_{VP}$ $(the\ current\ account\ deficit)_{NP}$ $(will\ narrow)_{VP}$ $(to)_{pp}$ $(only\ 1.8\ billion)_{NP}$ $(in)_{PP}$ $(September)_{NP}$.

III.2.5 adibidean ikus daitekeen moduan, preposizioa bera (eta bera bakarrik) hartzen zuten preposizio-sintagmatzat (ikus "*to*" eta "*in*" preposizioak, aurreko adibidean). Preposizio horiek jarraian zetozen izen-sintagmekin lotzea, izan ere, ondorengo urrats batean modu errazean egin zitekeela aurreikusi zuten.

Tesi-lan honen zereginen artean ez da sartzen kate baten izaerari buruz jardutea, baina argi utzi nahiko genuke Aranzabek (2008) finkatutako irizpide linguistiko berak erabili ditugula, IXA taldearen baitan egindako lanen koherentzia bermatze aldera.

Arestian aipatutako definizioari jarraiki, beraz, jarraian agertzen den moduan markatuko genituzke kateak (adibidean agertu ez arren, kate bakoitzak

kategoria bat ere izango du, kate hori sintagma bat den edo aditz-kate bat den argituko duena orokorrean):

**Adibidea III.2.6**
*(Antimilitaristen eskaerekin) (bat egin du) (Jaurlaritzak).*

III.2.6 adibidean ikus daitekeen moduan, kate linguistikoak ez dira euren artean gainjartzen; are, ez daude bata bestearen baitan. Bi propietate horiek (elkar ez gainjartzea eta bata bestearen baitan ez egotea) ez ditu soilik adibide honek betetzen; aitzitik, kate guztietara orokor daitezke. Propietate hau inguruko hizkuntza gehienek ere (hala nola, ingelesak, gaztelaniak edo katalanak) betetzen dute, *chunk*-aren edo katearen definizioa bakoitzak bere erara aplikatzen duen arren (arestian ikusi dugun legez). Euskaraz, baina, bada salbuespen argi bat —jarraian ikusiko dugun adibidearekin azalduko duguna—, non kateen arteko izaera jarraitua galtzen den:

**Adibidea III.2.7**
*(Ez dut (horretan) sakondu nahi).*

Batik bat ezezko esaldietan gertatzen den fenomeno hau, ordea, sinplifika daiteke, eta hala egin dugu guk gainerako hizkuntzekiko halako homogeneotasuna bilatze aldera, eta ataza errazteko asmoz:

**Adibidea III.2.8**
*(Ez dut) (horretan) (sakondu nahi).*

Jakitun ginen erabaki honekin informazio linguistikoa galtzen genuela. Izan ere, ez da gauza bera aditz-kate baten bi parteek kate bat bakarra osatzen dutela esatea, edo bi elementu guztiz independente direla adieraztea. Hala ere, erabaki honekin, arazoaren ebazpena sinplifikatzeaz gain, gainerako hizkuntzekin egin denarekin halako parekidetasun bat lortzen genuen, kate jarraituak soilik kontsideratu baitira, oro har, inguruko hizkuntza gehienetan. Gainera, galtzen den informazio linguistikoa —bi elementuen artean dagoen lotura— ez zitzaigun erabakiorra iruditu. Galtzen den informazio hau, halere, baliagarria da analisi sintaktiko osorako, eta etorkizunean kontuan hartu beharrekoa izango da.

Aranzabek (2008) ere, batik bat kate jarraituak landu zituen. 560 erregelaz osatutako gramatika bat egin zuen, eta emaitza onak lortu zituen kate jarraituen identifikazioan. III.1 taulan ikus daitezke kate jarraituen identifikazioan lortutako emaitzak, erregeletan oinarritutako kate-identifikatzailearekin. Kontuan hartu behar da eskuz desanbiguatutako informazio lin-

guistikoa eta analisi guztietan zuzena zena bakarrik erabiliz ebaluatu zela kate-identifikatzaile hau.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **Sintagmak** | 86,92 | 80,68 | 83,68 |
| **Aditz-kateak** | 84,19 | 87,77 | 85,94 |
| **Kateak** | 85,92 | 83,46 | 84,67 |

Taula III.1: Erregeletan oinarritutako kate-identifikatzailearen emaitzak, eskuz desanbiguatutako informazio linguistikoa erabiliz: kate jarraituen identifikazioa.

Lehen aipatu dugun moduan, ordea, kate ez-jarraituek konplikazioak dakartzate. Aranzabe (2008) soluzio-bide bat ematen saiatu zitzaion gisa honetako kateei ere.

**Adibidea III.2.9**
*Ez dut horretan sakondu nahi.*

III.2.9 adibidean, esaterako, aditz-katea "*ez dut sakondu nahi*" izango litzateke, (Aranzabe, 2008) lana oinarritzat hartuz gero, eta III.2.10 adibidean ikus daiteke aditz-kate hori testuan nola etiketatuko litzatekeen, irizpide horren arabera:

**Adibidea III.2.10**
$(Ez_{Adikatetenhas}\ dut_{Adikateten})\ (horretan)_{Sint}\ (sakondu_{Adikateten}\ nahi_{Adikatetenbu})$.

Beste modu batean esanda, aditz-katean eten bat dagoela adierazten da, hiru etiketa berri gehituz: *adikatetenhas* (aditz-kate etendunaren hasiera markatzeko), *adikatetenbu* (aditz-kate etendunaren bukaera markatzeko) eta *adikateten* (etenaren hasiera eta bukaera markatzeko).

Ezezko esaldietan gertatzen den arazo hau, gainera, are gehiago konplika daiteke, bi ezezko gainjarri daitezkeela kontuan hartzen badugu ("*Ez duzu aitortu nahi ez duzun hori aitortu beharrik.*").

Honek, bistan denez, zaildu egiten ditu kontuak. Aranzabek (2008) berak aparte tratatu zuen aditz-kate ez-jarraituen fenomenoa, baina ez zuen ebaluaziorik egin.

Aipatu dugun moduan, honelako egiturak etiketatzeko guk hartutako bidea, ordea, sinpleagoa da. Izan ere, gure jokabidea, hasiera batean, aditz-kate ez-jarraituak bi aditz-kate gisa kontsideratzea izan da. Alegia, "*Ez dut*"

aditz-kate bat litzateke, "*horretan*" izen-sintagma bat, eta "*sakondu nahi*" beste aditz-kate bat (ikus III.2.8 adibidea).

### III.2.2.3 Kateen identifikazioa ikasketa automatikoa erabiliz

Aranzaberen (2008) laneko definizioari jarraiki eta aipatutako sinplifikazioa kontuan hartuz (aditz-kate ez-jarraituak bi aditz-kate gisa tratatzearena, hain zuzen), kateen identifikazio automatikoa egin daiteke ikasketa eredu sekuentzialak edo jarraituak erabiliz. Eredu hauek, esaldi bat emanda, kateen etiketen konbinazio egokiena asmatu behar dute. Azken batean, hitz bakoitzari dagokion katearen etiketa asmatu behar dute gisa honetako sailkatzaileek, hitz honen eta inguruko hitzen informazio linguistikoa erabiliz.

Azken urteetan egindako lanetan, ikasketa automatikoa erabili izan da gehienbat kateen identifikaziorako. Ikasketa automatikoaren paradigma orokorraren baitan, hainbat algoritmo baliatu izan dira, baina algoritmo diskriminatzaileak darabiltzate onenek.

Kateen identifikazioari lotutako *CoNLL 2000* batzarreko zeregin partekatua izan zen ataza honen mugarrietako bat (Sang eta Buchholz, 2000), baliabideak jarri baitzituen ingeleseko kate-identifikatzaileak lantzeko. Modu horretan, ikasketa- eta test-corpus hauek erabiltzen zituzten lan guztien emaitzak konparagarriak izatea lortu zen. III.4.1.1 atalean ematen da formatu honen berri.

Oso emaitza onak lortzen dituzte, besteak beste: Lee eta Wu-k (2007) eta Kudo eta Matsumoto-k (2001), *Support Vector Machines (SVM)* algoritmoa erabiliz (ikus II.1.5.4 atala SVM algoritmoaren azalpenak ikusteko); Shen eta Sarkar-ek (2005) eta Molina eta Pla-k (2002), Markov-en eredu ezkutuen[10] (Rabiner eta Juang, 1986) nolabaiteko aldaerak erabiliz; baita (Zhang *et al.*, 2002) lanean ere, *Winnow* algoritmoaren orokortze bat baliatuz; eta, azkenik, pareko emaitzak lortzen dira (Carreras *et al.*, 2005) lanean ere, *pertzeptroi*etan oinarritutako algoritmoa baliatuz. Sha eta Pereira-k (2003),

[10]Markov-en eredu ezkutuak (*Hidden Markov Models (HMM)*): Markov-en ereduek Markov-en propietatea betetzen dute: gertaera bat betetzeko probabilitatea bere berehalako aurreko gertaeraren mende soilik dago. Automata finitu baten moduan ikus daiteke Markov-en eredu ezkutu bat. Egoerek ereduaren aldagaiak errepresentatzen dituzte, eta arkuek egoera batetik bestera joateko dagoen probabilitatea gordetzen duen etiketa bat dute. Alfabeto bateko sinboloak idaztea baimentzen dute egoerek, probabilitate-funtzio baten arabera. Ezkutua dela esaten da ezin delako jakin ereduak aukeratzen duen egoeren segida, ez bada egoeren segidaren funtzio probabilistiko bat.

*Conditional Random Fields* (CRF[11]) izeneko algoritmoa erabiliz izen-sintagmak soilik landu zituzten, baina hauen emaitzak —$F_1 = \%$ 94,38— algoritmo onenen antzekoak dira.

III.2 taulan ikus daitekeen moduan, ingeleseko emaitzarik onenak % 94ko bueltan dabiltza $F_1$ neurrian (III.4.2 atalean azalduko dugu zer den neurri hau eta nola kalkulatzen den). Lee eta Wu-k (2007) SVM algoritmoa erabiliz eta ezaugarri berriak —ortografikoak— gehituz lortu zituzten emaitzarik onenak. Hala ere, ezin frogatu ahal izan zuten ea eurek lortutako hobekuntza estatistikoki esanguratsua den ala ez.

Algoritmo hauek guztiek propietate garrantzitsuak dituzte: hasteko, datuak errepresentatzeko hainbat iturritatik lortutako ezaugarriak erabil ditzakete; bestalde, oso algoritmo eraginkorrak dira: milaka eta milaka adibiderekin lan egin dezakete, eta baita hamaika ezaugarrirekin ere; gainera, badira teoriak ziurtatzen dutenak ikasitako eredu hauek behatu gabeko datuetan aplikatzean emaitza onak lortzen dituztela, erabilitako atributuak asko izanagatik ere.

| | **Teknika** | **Doit.** | **Est.** | $F_1$ |
|---|---|---|---|---|
| **(Lee eta Wu, 2007)** | *SVM* | 94,22 | 94,23 | 94,22 |
| **(Zhang *et al.*, 2002)** | *Winnow* orokortua | 94,29 | 94,01 | 94,13 |
| **(Shen eta Sarkar, 2005)** | *Voted HMM* | 94,12 | 93,89 | 94,01 |
| **(Kudo eta Matsumoto, 2001)** | *SVM* | 93,89 | 93,92 | 93,91 |
| **(Carreras *et al.*, 2005)** | *FR-perceptron* | 94,19 | 93,29 | 93,74 |
| **(Molina eta Pla, 2002)** | *Spec. HMM* | 93,25 | 93,24 | 93,25 |
| ***Oinarrizko neurria*** | kategoriaren usuena | 72,58 | 82,14 | 77,07 |

Taula III.2: *CoNLL 2000* batzarreko zeregin partekatuaren baldintzetan lortutako emaitzarik onenak, bertan emandako *oinarrizko neurri*arekin erkatuta (token bakoitzari dagokion kategoriak corpusean gehienetan daukan etiketa esleituta lortzen da oinarrizko neurria).

[11] *Conditional Random Fields (CRF)*: Markov-en eredu ezkutuen antzeko eredu estokastiko bat da. Oro har, datu-sekuentzia bat emanik $(O_1, O_2...O_n)$, eredu honek $S_i$ etiketa bana esleitzen dio $O_i$ osagai bakoitzari. Markov-en eredu ezkutuek etiketen eta behaketen probabilitateen distribuzioa batera kalkulatzen dute: P(S,O); *Conditional Random Fields* izenekoetan, berriz, behaketek baldintzatzen dute etiketen sekuentzia zuzenaren probabilitatea: $P(S|O)$. Hala, Markov-en eredu ezkutuetan egiten den independentzia-hipotesia erlaxatzen dute *Conditional Random Fields* izenekoek.

Bestalde, aipatu beharra dago emaitza hauek guztiak ikasketa gainbegiratua eginez lortutakoak direla. Ikasketa erdi-gainbegiratuari dagokionez, (Ando eta Zhang, 2005) lanak hauek guztiak hobetu zituen ($F_1$ =% 94,39). Lan honetan, kateen identifikazioko atazarako garrantzitsuak diren milaka azpiataza laguntzaile sortzen ziren (esaterako, hitz bat *jendea* den ala ez erabakitzea, testuinguruko hitzak soilik hartuz kontuan), eta azpiataza bakoitzerako sailkatzaile bana ikasten zen, etiketatu gabeko corpusetik azpiataza bakoitzerako automatikoki sortutako ikasketa-corpusa baliatuz. Azpiataza hauek sinpleak izanik, automatikoki sor daiteke nahi adina ikasketa-corpus, azpiataza hauentzat, etiketatu gabeko corpusetik abiatuz. Azkenik, azpiataza hauek ikasteko sortutako sailkatzaileek elkarren artean konpartitzen zituzten iragarpen-egiturak bilatzen ziren, kateen identifikazioan erabiltzeko. *Egiturazko ikasketa* deitu zioten autoreek ikasketa-modu honi. Lortzen dituzten emaitzak ikusita, etorkizunean kontuan hartzeko bide bat dela uste dugu.

### III.2.3 Perpausak

Perpausen identifikazioa azaleko analisi sintaktikoaren baitan kokatzen den beste eginkizun bat da. Hala, oso baliagarria izan daiteke hizkuntzaren prozesamenduko zenbait aplikaziotarako (Ejerhed, 1996):

- Testuen analisia: esaldien analisia egiteko aurreko urratsa izan daiteke esaldia perpausetan banatzea.
- Ahotsaren sintesia: intonazioa hobe daiteke perpausen identifikazioarekin.
- Itzulpen automatikoa: perpausak har daitezke itzuli beharreko unitate gisa.

Kateen identifikazioan nola, perpausen identifikazioan ere egin dira hainbat ahalegin azken urteetan, HPan.

Ikasketa automatikoak kateen identifikazioan izandako emaitza onek metodo hauek perpausen identifikazioan erabiltzera eraman dute komunitate zientifikoa. Perpausen identifikazioa, ordea, kateen identifikazioa baino lan zailagoa da, arestian aipatutako moduan, perpausen izaera errekurtsiboa delako, hurrengo adibidean ikus daitekeen moduan:

**Adibidea III.2.11**
*((Urteko helburu nagusia (bakea lortzea) izango dela) adierazi du.)*

Emaitzak, perpausen izaera errekurtsibo hau dela-eta, ez dira kateen identifikazioan lortutakoak bezain onak izango.

### III.2.3.1 Perpausak: hurbilpen linguistikoa

Hurbilpen linguistikoa jorratzeko, definizio formalez gain, garrantzitsua izaten da tratatu nahi den fenomenoaren definizio linguistikoa egitea. Perpausaren eta esaldiaren definizio linguistiko erabilienak laburbilduko ditugu, hortaz, jarraian.

(Zubiri eta Zubiri, 1995) lanean, esate baterako, perpausaren definizio hau ematen da:

> *"[...] osagai desberdinak (izena, aditza, adjektiboa...) konbinatu egiten dira unitate handiagoak osatuz. Unitate horiei sintagma deitzen diegu. Aditzaren inguruan biltzen diren osagaiek osatzen duten unitateari aditz-sintagma deritzagu eta izenaren inguruan biltzen diren osagai multzoari izen-sintagma. Sintagma desberdinak konbinatuz, perpausak osatzen ditugu. Bestalde, osagai guztiak aditzaren inguruan biltzen dira; izan ere, aditza dugu perpausaren ardatza. Beraz, perpaus bakoitzak aditz bat izango du. Bestela esanda, perpaus bakuna aditz bakarra duena da [...]"*

Collins-ek (1992) ere antzeko definizioa ematen digu: *aditz bat daukan hitzen multzoa da perpausa.*

Aurreko definizioari zera gehitu behar zaio, aditza ez dela beti testuan esplizituki agertuko; hots, aditza perpausaren ardatza izanagatik ere, aditzaren beraren elipsia egon daitekeela eta horrek ez diola perpausari perpaus-izaera kentzen:

**Adibidea III.2.12**
*Bi lagun larri, auto-istripuan.*

Adibide honetan, esaterako, aditzik ez da agertzen; edo beste modu batean esanda, aditzaren elipsia dago. Alabaina, perpaus bakun bat dela esan genezake.

(Quirk *et al.*, 1985) lanean kontuan hartzen dute arazo hau, eta hiru perpaus mota definitzen dituzte:

1. Perpaus *finituak*: "*You can borrow my car* ***if you need it.***"
2. Perpaus *ez-finituak*: "***Visiting many cities*** *makes me tired.*"

3. Aditzik gabeko perpausak (komunikazioa errazteko erabiltzen dira): "***If necessary***, *he will take notes for you.*"

Ejerhed-ek (1988) ikuspegi sintaktikotik nahiz semantikotik garrantzitsua den esaldi-egitura gisa definitzen du oinarrizko perpausa:

> *"[...] Erraz identifika daitekeen azaleko unitate-egitura egonkorra da oinarrizko perpausa, eta tarteko emaitza garrantzitsua izanik, diskurtso-egituraren elementu sintaktiko eta semantikoak bere baitan hartzen dituen errepresentazio linguistiko aberatsagoen sorkuntzan osagai garrantzitsua da [...]"*

Esaldia, berriz, komunikazio-mezu oso bat osatzen duen hitzen multzoa da, eta perpaus batez edo gehiagoz osatua egon daiteke. Eginkizun konputazionaletarako, puntuazio-markak hartzen dira aintzat esaldia mugatzeko (Leech *et al.*, 1996). Hala, esaldia kontsideratzen da puntuazio *gogor* artean dagoen oro, puntuazio *gogor* gisa harturik puntua, harridura-marka eta galdera-marka (hiru puntuak, puntu eta koma eta bi puntuak ere hala kontsideratzen dira, kasu batzuetan).

Esaldi barruko aditz bakoitzeko, alabaina, perpaus bat dugula esango dugu. Hortaz, aditzak eta aditzari dagozkion elementuek osatzen dute perpausa. Bestalde, bi perpaus mota definitzen ditugu: markatuak (menderatuak) eta markatu gabeak. Menderagailua daramatenak izango dira markatuak edo mendeko perpausak, eta markarik ez dutenak perpaus bakunak izango dira. Ikus dezagun adibide bat:

**Adibidea III.2.13**

1. *Oposizioko alderdiek kritikatu egin dute EAEko lehendakariaren urteburuko mezua.*
2. *Hiriko agintari militarrek aditzera eman dutenez, suziri batek eragin zuen leherketa.*

III.2.13 adibideko 1. esaldian, perpaus bakarra daukagun moduan (perpaus bakuna litzatekeena), 2. esaldian mendeko perpaus bat dugu esaldiaren baitan. Hona hemen, parentesi bidez adierazia, azaldutakoa:

**Adibidea III.2.14**

1. *(Oposizioko alderdiek kritikatu egin dute EAEko lehendakariaren urteburuko mezua.)*
2. *((Hiriko agintari militarrek aditzera eman dutenez,) suziri batek eragin zuen leherketa.)*

Hala, definizio linguistiko hauekin, perpausen identifikazio automatikorako egindako formalizazioa baieztatzen da; perpaus-egiturak, elkar gainjarri ez, baina bata bestearen baitan egon daitezkeen *hitz multzo* jarraituak dira.

#### III.2.3.2 Euskarako perpausen identifikazioa hurbilpen linguistikoa erabiliz

Arazoak arazo, IXA taldean egin zen saiakera bat, CG formalismoa baliatuz, batez ere perpausen bukaera-mugak identifikatzeko erregelak idaztekoa. Lehen saioak 2000. urtean egin baziren ere (Aduriz, 2000), duela gutxi —lan hura abiapuntutzat hartuta— perpausak detektatzeko gramatika sendoago bat egin zen. Lan honetan, CG formalismoa erabiliz, hainbat erregela egin ziren (Aduriz *et al.*, 2006c).

Esaldien identifikazioari dagokionez, problema ez da oso konplexua. Izan ere, puntuazio-markez mugatuta daude esaldiak, eta erregela gutxi batzuk nahikoak dira esaldiak taxuz identifikatzeko:

- Baldin puntuazio-marka *gogorra*[12] topatzen bada, orduan puntuazio-markari jarri esaldi-bukaera etiketa eta hurrengo hitzari esaldi-hasiera etiketa.

- Baldin koma bat topatzen bada eta hurrengo hitza juntagailua bada, orduan komari jarri esaldi-bukaera etiketa eta juntagailuari esaldi-hasiera etiketa.

Bigarren kasu honetan, arazoren bat sor daiteke, baldin eta analizatzaile automatikoak juntagailua ez den hitz bat juntagailu gisa etiketatzen badu, edo alderantziz: baldin eta juntagailua den hitz bat ez badu hala etiketatzen. Dena dela, aplikatzea erabaki genuen.

Perpausei dagokienez, berriz, oro har, erregela bidezko saiakera hauetan, perpaus bakoitzaren hasiera eta bukaera identifikatzen saiatu beharrean, perpaus biren arteko mugak jartzen ahalegindu ziren. Horretarako, mendekotasun-markak baliatu zituzten. Izan ere, mendeko perpaus askotan azken hitzak mendekotasun-marka eraman ohi du, perpaus horrek perpaus nagusiarekiko daukan mendekotasuna adierazten duena, hain zuzen. Horregatik, normalean, perpaus-bukaera adierazten dute muga hauek. III.2.15 adibidearekin errazago uler daiteke esaten ari garena (kontuan izan *&MUGA* etiketa erabili dela perpausak mugatzeko).

**Adibidea III.2.15**

1. *Hiriko agintari militarrek aditzera eman dutenez&MUGA suziri batek eragin zuen leherketa.*

[12]Hauek dira puntuazio-marka *gogor* gisa kontsideratu ditugunak, IXA taldean: puntua, bi puntu, puntu eta koma, harridura-marka, galdera-marka eta hiru puntuak.

2. *Bospasei ustezko islamistek autobusaren aurka tiro egin zuteneanℰMUGA gertatu zen erasoa Eukaliptoak auzoan.*

III.2.15 adibideko esaldiak, hala eta guztiz ere, idatz daitezke hitz berak erabiliz baina hitzen ordena aldatuta. Halakoetan —menderagailua perpaus nagusiaren eta mendekoaren artean ez dagoenean—, erregelen bidez mugak detektatzeko zailtasunak areagotzen dira:

**Adibidea III.2.16**

1. *Suziri batek eragin zuen leherketa,ℰMUGA hiriko agintari militarrek aditzera eman dutenez.*

2. *Erasoa Eukaliptoak auzoan gertatu zenℰMUGA bospasei ustezko islamistek autobusaren aurka tiro egin zutenean.*

III.2.16 adibidean ikus daitekeen eran, esaldi hauetan menderagailuak ez liguke mendeko perpausaren mugak identifikatzeko pistarik emango. III.2.16 adibidearen 1. esaldiaren gisako kasuetan koma lagungarria izan daitekeen arren, kontuan hartu behar dugu, alde batetik, 2. esaldiaren gisakoak ere izango ditugula, eta, bestetik, komak behar diren tokian ez izatea ere gerta dakigukeela. Gainera, guztiak ez dira adibideok bezain esaldi sinpleak izango.

Ingelesez eta gaztelaniaz ere, esate baterako, antzeko arazoa dago. Hala, (Quirk *et al.*, 1985) lanean ere, perpausen identifikazioan, batzuetan (adberbio-perpausetan edo erlatibozkoetan, kasu), zenbait marka oso lagungarriak izan daitezkeela esaten da. Adibidez, "*You can borrow my car if you need it*" esaldian, "*if you need it*" perpausa "*if*" konjuntzioaren bidez markatuta dator.

Alabaina, kasu askotan ez da gisa honetako markarik izaten. Halakoetan, hainbat murriztapen sintaktiko edo semantiko kontuan hartu behar dira. Esate baterako, "*The book he had described Rome (Berak zeukan liburuak Erroma deskribatzen zuen)*" esaldia zuzen analiza daiteke, jakinik "*had*" eta "*described*" aditzek subjektu bat izan behar dutela. III.2.17 adibidean, berriz, bi esaldiek egitura sintaktiko oso antzekoa daukatenez, informazio semantikoa behar-beharrezkoa da perpausak zuzen identifikatzeko. Izan ere, lehen esaldia perpaus bakun bat den arren, bigarren esaldiak beste perpaus bat dauka bere baitan:

**Adibidea III.2.17**

*(The teacher is teaching)*
*(The problem is (teaching))*

Bukatzeko, beste esaldi bat aztertuko dugu ("*Problems show up if you love money*"), zeinak interpretazio bakarra daukan ("*Dirua maite baduzu, arazoak izango dituzu*"). Alta, mendeko perpausa esaldiaren hasieran jartzen badugu, interpretazio anitz izan ditzake, puntuazioa ez bada —zuzen— erabiltzen:

**Adibidea III.2.18**

*If you love money* ***problems show up*** *(Dirua maite baduzu, arazoak izango dituzu).*
*If you love* ***money problems show up*** *(Maite baduzu, diru-arazoak izango dituzu).*

Adibide hauekin, argi geratzen da perpausen identifikazioa ez dela batere ataza erraza eta puntuazioarekin lotura zuzena duela.

Bestalde, kontuan izan behar dugu perpausen identifikatzailea zertarako erabili nahi dugun. HPko ataza askotarako izango da baliagarria, zalantzarik gabe. Ataza horietarako, beraz, ahalik eta perpausen identifikazio onena beharko litzateke, eta horretarako, noski, komak ere aintzat hartu beharko dira, iruditzen baitzaigu komek ematen duten informazioa esanguratsua dela perpausen identifikazio automatikorako.

Tesi-lan honetan, ordea, komak zuzentzeko ere erabili nahi ditugu perpaus-mugak. Honek, ordea, bi perpaus-identifikatzaile desberdin egitera garamatza: batetik, komen informazioa aprobetxatzen duen perpaus-identifikatzailea (ahalik eta onena), HPko hainbat atazatarako baliagarria izango dena (kapitulu honetan bertan azalduko duguna); eta bestetik, komen informazioa ez darabilen perpaus-identifikatzailea, IV. kapituluan azalduko dugun koma-zuzentzailea egiteko erabiliko duguna. Izan ere, uste dugu perpaus-mugen informazioa komak zuzentzeko garrantzitsua dela (komak perpausak identifikatzeko baliagarriak diren moduan). Hori dela eta, komak detektatzeko perpaus-mugak erabili nahi baldin baditugu, koma guztiak kendu beharko genituzke perpausak identifikatzeko; alegia, ezin ditugu komak erabili perpausak identifikatzeko, gero perpaus-muga horiek komak zuzentzeko erabili nahi baldin baditugu, egokiak ez diren komek informazio okerra eman baitezakete. IV. kapituluko IV.6.2.9 atalean, ekidin beharreko gurpil zoro honen berri emango dugu.

Orotara, 70 bat erregela garatu ziren (Aduriz *et al.*, 2006c), eta baita emaitza onak lortu ere (ikus III.3 taula). Emaitza hauek aztertzerakoan, haatik, kontuan hartu behar dira bi gauza:

1. Bi perpausen arteko mugak soilik ebaluatu dira, kontuan hartu gabe muga horiek perpaus baten bukaera edo hurrengoaren hasiera adierazten duten.

2. Ez dira perpausak beren osotasunean ebaluatu; hots, esaldiaren puntu konkretu batean bi, hiru edo lau perpaus bukatuko balira, erregelek perpaus-muga bukaera bat (eta bakarra!) jarri izana ontzat emango litzateke.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **mugak** | 91,9 | 82,9 | 87,17 |

Taula III.3: *Constraint Grammar* erregeletan oinarritutako euskarako mugen detektatzailearen emaitzak.

III.2.19 adibidean, esate baterako, esaldi hasieran bi perpaus-hasiera egonagatik, CG formalismoko mugen gramatikak etiketa bakarra jarriko luke, eta hori zuzentzat emango litzateke ebaluazioan. Izan ere, gramatikak perpaus-muga jartzen duenean, berez mugaren bat badago, ontzat ematen da. Hortaz, ebaluazioan ez dira perpausen identifikazioak beren osoan ebaluatzen, perpaus-muga soilak baizik. Beraz, ebaluazio hau, bere horretan, ez da konparagarria (Sang eta Déjean, 2001) lanean deskribatutako perpausen identifikazio osoaren ebaluazioarekin.

**Adibidea III.2.19**
*((Urteko helburu nagusia (bakea lortzea) izango dela) adierazi du.)*

Konparagarria izan zedin, hain zuzen, hiru heuristiko inplementatu eta probatu genituen. III.2.19 adibidea baliatuko dugu heuristikoak hobeto azaltzeko. Eman dezagun erregeletan oinarritutako gramatikak honela etiketatzen duela adibide horretako esaldia:

**Adibidea III.2.20**
*&MUGA Urteko helburu nagusia bakea lortzea &MUGA izango dela &MUGA adierazi du. &MUGA*

Hiru modu desberdin hauek probatu genituen perpaus edo esaldi arteko muga soil diren hauek balio zezaten perpausak beren osoan identifikatzeko; alegia, muga-detektatzailea, programa sinple batekin, perpaus-identifikatzaile bihurtzeko:

- 1. heuristikoa: esaldi-hasiera edo esaldi-bukaera ez den perpaus-muga bakoitzean, perpaus bat bukatzen dela suposatzea; perpaus-bukaera

honen hasiera, berriz, aurreko perpaus-bukaeraren hurrengo hitzari esleitzen zaio (*heuristiko1* deituko diogu honi). Esku artean darabilgun adibidean, perpaus hauek identifikatuko lituzke heuristiko honek:

**Adibidea III.2.21**
*((Urteko helburu nagusia bakea lortzea) (izango dela) adierazi du.)*

Modu honetan ebaluatuta, % 49,71 lortzen zen $F_1$ neurrian.

- 2. heuristikoa: 1. heuristikoaren emaitza behatuz, ordea, beste heuristiko bat ez ote zen hobea izango otu zitzaigun: &MUGA etiketa bakoitzean perpaus-bukaera jartzea, baina bukaera horri dagokion hasiera beti esaldi-hasierari esleitzea (*heuristiko2*). Izan ere, darabilgun adibiderako, behinik behin, heuristiko honekin lortzen da emaitza zuzena:

  **Adibidea III.2.22**
  *(((Urteko helburu nagusia bakea lortzea) izango dela) adierazi du.)*

  Bigarren heuristiko honekin ebaluatuta, ordea, emaitza kaxkarragoa lortzen zen: $F_1 =$% 48,35.

- 3. heuristikoa: perpaus-muga bakoitzean perpaus bat bukatzen dela suposatzea, eta perpaus-bukaera honen hasiera, berriz, aurreko perpaus-bukaeraren hurrengo hitzari esleitzea; mugatu gabe geratzen den hitzen multzoa ere perpaustzat hartzen da (*heuristiko3*).

  **Adibidea III.2.23**
  *((Urteko helburu nagusia bakea lortzea) (izango dela) (adierazi du).)*

  III.2.23 adibidean ikus daitezke 3. heuristiko hau aplikatuta lortzen diren perpausak, eta hauek ez dira esaldi horrek izan behar lituzkeenak; hala ere, kasu batzuetarako heuristiko ona zela jakinik, proba egin nahi izan genuen. Emaitzarik kaxkarrenak lortu genituen, ordea: $F_1 =$ % 45, 88.

III.4 taulan ikus ditzakegu heuristiko desberdinen emaitzak laburbilduta.

IXA taldean CG bidez egindako mugatzailearen emaitzak, beraz, bi perpausen arteko mugak detektatzeko onak baldin badira ere ($F_1 =$ % 87,17), perpausak beren osoan identifikatzeko eskasak dira ($F_1 =$% 49,71). Dena dela, ikasketa automatikoa erabiliz lortzen ditugun emaitzak hobetzeko balioko digutela ikusiko dugu III.6.2 atalean.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **muga-detektatzaile** | 91,9 | 82,9 | 87,17 |
| **perpaus-identifikatzaile: heuristiko1** | 50,84 | 48,63 | 49,71 |
| **perpaus-identifikatzaile: heuristiko2** | 49,45 | 47,29 | 48,35 |
| **perpaus-identifikatzaile: heuristiko3** | 42,75 | 49,51 | 45,88 |

Taula III.4: *Constraint Grammar* erregeletan oinarritutako euskarako mugen detektatzailearen emaitzak, muga-detektatzaile soil gisa, edota perpaus-identifikatzaile gisa.

#### III.2.3.3 Perpausen identifikazioa ikasketa automatikoa erabiliz

Ataza honen garapenean, mugarria izan zen CoNLL batzarrean, 2001. urtean, antolatutako eginkizun partekatua (Sang eta Déjean, 2001).

Bertan, aurreko urteko CoNLL batzarreko eginkizun partekatua oinarritzat hartuta (non kateak edo *chunk*ak detektatzea baitzen helburua), perpausak identifikatzeko helburua jarri zen. Perpausaren izaera errekurtsiboak eginbeharra zailtzen zuela-eta, hiru atazatan banatu zen eginkizuna: perpausen hasiera aurkitzea, perpausen amaiera aurkitzea eta perpaus osoak identifikatzea. Hirugarren atazarako lehen bi eginkizunen emaitzak erabil zitezkeen, jakina.

**Adibidea III.2.24**
*The deregulation of railroads and trucking companies that began in 1980 enabled shippers to bargain for transportation.*

*(S The deregulation of railroads and trucking companies*
*(S that*
*(S began in 1980)*
*)*
*enabled*
*(S shippers to bargain for transportation)*
*.)*

Erreferentzia gisa erabilitako datuak ingeleseko *Penn Treebank* corpusetik lortu ziren (Marcus *et al.*, 1993). Corpus horretan erabilitako etiketatzearauak eta esaldia perpausetan banatzeko irizpideak zehaztuta datoz (Bies *et al.*, 1995) lanean[13]. CoNLL 2001eko ataza partekaturako, ordea, eta egin-

[13]Zenbait perpaus mota desberdintzen dira, bi taldetan banatuta, *Penn Treebank* corpusean: oinarrizkoak eta konbinatuak (koordinazio edo mendekotasun bidez erlazionatutako zenbait perpaus).

kizuna sinplifikatze aldera, perpaus mota guztiak bateratu egin ziren, denei perpaus-izaera bera emanez, baina perpausen hasierako berezko egitura inolaz ere aldatu gabe (ikus III.2.24 adibidea).

III.2.25 adibidean ikus daiteke CoNLL 2001eko ataza partekaturako corpusaren formatuaren itxura. Lehen zutabean, hitzari dagokion informazioa dago; bigarrenean, kategoriari dagokiona; hirugarrenean, kateari dagokiona; eta, laugarrenean, perpausari dagokiona. III.4.1.1 atalean ikus daitezke kateen eta perpausen etiketen esanahiak.

**Adibidea III.2.25**
CoNLL 2001eko ataza partekatuko formatua:

| | | | |
|---|---|---|---|
| *The* | *DT* | *B-NP* | *(S** |
| *deregulation* | *NN* | *I-NP* | * |
| *of* | *IN* | *B-PP* | * |
| *railroads* | *NNS* | *B-NP* | * |
| *and* | *CC* | *O* | * |
| *trucking* | *NN* | *B-NP* | * |
| *companies* | *NNS* | *I-NP* | * |
| *that* | *WDT* | *B-NP* | *(S** |
| *began* | *VBD* | *B-VP* | *(S** |
| *in* | *IN* | *B-PP* | * |
| *1980* | *CD* | *B-NP* | **S)S)* |
| *enabled* | *VBD* | *B-VP* | * |
| *shippers* | *NNS* | *B-NP* | *(S** |
| *to* | *TO* | *B-VP* | * |
| *bargain* | *VB* | *I-VP* | * |
| *for* | *IN* | *B-PP* | * |
| *transportation* | *NN* | *B-NP* | **S)* |
| *.* | *.* | *O* | **S)* |

Ataza partekatuan parte hartu zuten guztiek ikasketa automatikoa zerabilten: *boosting* algoritmoa, erabaki-zuhaitzak edo memorietan oinarritutako ikasketa (MBL) erabili ziren, besteak beste. Emaitzarik onenak Carreras eta Màrquez-ek (2001) lortu zituzten, *AdaBoost*[14] metodoarekin ($F_1 = \%\ 78{,}63$).

Gerora, *pertzeptroi*ekin egindako *iragazketa eta sailkapena* arkitektura sortu eta baliatu zuten euren emaitza propioak hobetzeko (Carreras eta Màrquez, 2003). Eurenak izan dira, orain gutxi arte, perpausen identifikazio automatikoan, ingeleserako lortu diren emaitzarik onenak ($F_1 = \%\ 85{,}03$).

[14]AdaBoost (Freund eta Schapire, 1997): sailkatzaile *ahulak* uztartuz zehaztasun handiko sailkatze-erregelak eskuratzen dituen metodo orokorra da.

| | Teknika | Doit. | Est. | $F_1$ |
|---|---|---|---|---|
| **(Ram eta Devi, 2008)** | *CRF + erregela ling.* | 92,06 | 87,89 | 89,04 |
| **(Carreras *et al.*, 2005)** | *FR-Perceptrons* | 88,17 | 82,10 | **85,03** |
| **(Nguyen *et al.*, 2009)** | *joint-CRF* | 91,03 | 79,13 | 84,66 |
| **(Carreras eta Màrquez, 2001)** | *Boosting* | 84,82 | 78,85 | 81,73 |
| **(Molina eta Pla, 2001)** | *HMM* | 70,85 | 70,51 | 70,68 |
| **(Tjong Kim Sang, 2001)** | *MBL* | 76,91 | 65,22 | 70,58 |
| **(Patrick eta Goyal, 2001)** | *Erabaki-grafoak* | 73,75 | 64,56 | 68,85 |
| ***Oinarrizko neurria*** | *Esaldiak mugatu* | 98,44 | 31,48 | 47,71 |

Taula III.5: *CoNLL 2001* batzarreko zeregin partekatuaren baldintzetan lortutako perpaus-identifikatzailerik onenak, bertan emandako *oinarrizko neurri*arekin erkatuta.

Algoritmo horrek zeukan berezitasun garrantzitsuena zera zen, ikasketa automatikoko sailkatzaileak aplikatzen zituela eginkizunaren hiru atazetan. Ordura arteko beste lan guztiek lehen bi atazetan soilik erabiltzen zituzten ikasketa automatikoko teknikak; alegia, perpausen hasiera- eta bukaera-puntuak identifikatzeko soilik erabiltzen zuten ikasketa automatikoa, eta hirugarrenerako heuristikoak zerabiltzaten, oro har. Ez zituzten hain emaitza onak lortu, III.5 taulan ikus daitekeen eran.

Bestalde, (Carreras *et al.*, 2005) lanean kontuan hartu zen perpausen izaera errekurtsiboa, eta arazoa orokortuz, *hitz multzo*ak identifikatzeko sistema bat garatu zuten, *hitz multzo* hauek izaera errekurtsiboa izan ala ez.

Zehazki, bi mailatako arkitektura bat proposatu zuten, iragazketa moduko bat egiten zuena lehendabizi (*filtering*), eta iragazitako emaitza posible horiek egokitasunaren arabera sailkatzen zituena ondoren (*ranking*). HPko ataza garrantzitsu gehienetan (kateen identifikazio automatikoan, perpausen identifikazio automatikoan edo rol semantikoen etiketatzean) onenen pareko emaitzak lortu zituzten (Carreras, 2005).

Berriki, (Carreras *et al.*, 2005) lanak ematen dituen emaitzen parera iristen den beste lan bat argitaratu da (Nguyen *et al.*, 2009). Bertan, CRF algoritmoaren aldaera bat —*joint-CRF* (Shi eta Wang, 2007)— darabilte ikasketa-algoritmo gisa, baina euren ekarpen garrantzitsuena ezaugarri linguistiko berriak gehitzean datza. *CoNLL 2001* batzarreko zeregin partekatuan erabiltzen direnez gain (hitza, *Eustagger*-ek emandako kategoria eta III.5 atalean aurkeztutako kate-identifikatzailea erabiliz lortutako kateei buruzko informazioa) atributu hauek baliatzen dituzte:

- Hitz mailan: esaldi batean agertzen diren zenbait elementu linguistikoren kopuruak, esaldiaren hasieratik eta esaldiaren bukaeratik uneko tokenerako tartean:
  - Erlatibozko izenordainak (*that, who, what...*)
  - Puntuazio markak (. ; : ,)
  - Aipuak
  - Aditz-kateak
  - Erlatibozko kateak
- Esaldi mailan:
  - Hautagaiaren baitan topatutako perpausen kopurua
  - Patroi moduko batzuk, esaldi-egitura egokienaren elementu garrantzitsuenak errepresentatzen dituztenak

C++ lengoaian idatzitako algoritmo dinamiko bat darabilte, eta (Carreras *et al.*, 2005) laneko emaitzetara iristen ez badira ere, oso gertu geratzen dira. Gainera, euren sistemaren denbora konputazionala (Carreras *et al.*, 2005) laneko sistemarena baino askoz txikiagoa da; alegia, euren sistema azkarragoa da.

Duela gutxi, ordea, CRF ikasketa-algoritmoa erabiliz lau puntuko hobekuntza lortu dute (Ram eta Devi, 2008) lanean. Horretarako, CRF ikasketa-algoritmoa aplikatzeaz gain, erregela linguistikoak sortu eta erabili zituzten. CRF algoritmoan oinarritutako moduluaren portaera aztertu zuten, eta honek egiten zituen erroreak aintzat hartuz, akats hauen atzean zeuden gabeziak erregela linguistikoekin osatu zituzten. *Stacking* teknika erabiliz, ezaugarri berri baten gisa gehitu zuten erregelen informazioa euren sisteman. III.5 taulan ikus daitezke CoNLL 2001 batzarreko ataza partekatuaren baldintzetan lortu diren emaitzarik onenak.

Hala ere, *FR-Perceptron* algoritmoa erabiltzearen alde egin genuen tesilan honetan. Batetik, jorratu nahi genituen atazetarako emaitza oso konpetitiboak lortzen dituelako, eta bestetik, soluzio global bat proposatzen duelako azaleko sintaxiko hainbat ataza ebazteko. Dena dela, Nguyen *et al.* (2009) eta Ram eta Devi (2008) lanetan erabilitako estrategiak kontuan izan genituen: informazio linguistiko konplexuagoa erabiltzea, lehenari dagokionez, eta erregela linguistikoek emandako informazioa uztartzea, bigarrenari dagokionez.

Hurrengo atalean, beraz, gure egin dugun eta (Carreras, 2005) tesi-txostenean datorren *FR-Perceptron* algoritmoaren nondik norakoak azalduko ditugu.

## III.3 *Hitz multzo*en identifikazioa: iragazketa eta sailkapena, *pertzeptroi*ekin

HPko zenbait atazatarako beharrezkoa da ikasketa automatikoko zenbait sailkatzaile konbinatzea. Màrquez-ek (2002) dioenez, horixe da ikasketa automatikoaren gaur egungo erronketako bat:

> *"Otro de los retos actuales es la aplicación de técnicas de aprendizaje para resolver problemas complejos que no se reduzcan al simple esquema de clasificación. Un posible enfoque es la descomposición del problema en subproblemas básicos de clasificación y la definición de un esquema de combinación de los clasificadores básicos, cumpliendo las restricciones semánticas o estructurales impuestas por el tipo de problema, para obtener la solución deseada."*

(Carreras, 2005) eta (Carreras *et al.*, 2005) lanetan, zenbait sailkatzaile konbinatzen dituen ikasketa-estrategia orokor bat proposatzen da, *hitz multzo* batzuen izaera errekurtsiboa kontuan izanda (perpausak, kasu). Sistemak, zehatzago esanda, *hitz multzo*en egiturak identifikatzen ditu esaldian, eta bi mailatan edo bi geruzatan lan egiten du:

- Lehenengoan, iragazketa egiten da hitz mailan (*filtering*): esaldiko *hitz multzo* posible guztiak detektatzen dira, hau da, *hitz multzo hautagaiak*. Beste modu batean esanda, hitz bakoitza *hitz multzo* baten hasiera edo bukaera izan daitekeen ala ez erabakitzen da geruza honetan. Aukeratutako *hitz multzo* hautagai guztiek ez dute zertan koherente izan esaldiarentzat.

- Bigarrenean, *hitz multzo* mailan lan egiten da. Geruza honetan, lehen geruzan iragazitako *hitz multzo hautagaiak* puntuatzen dira (*ranking*), eta esaldiarentzat *hitz multzo*en segida onena aukeratzen da. Alegia, *hitz multzo* hautagai bakoitzari puntuazio bat ematen zaio —zenbaki erreal bat—, testuinguru horretan *hitz multzo* hori esaldian zenbaterainoko hautagai sendoa den adierazten duena.

Esaldiaren azken puntuazioa, beraz, aukeratutako *hitz multzo* hautagaiek duten puntuazioen batura izango da. *Hitz multzo*ekin aritzeak, ordea, badu desabantaila bat: aztertu beharreko hautagaien konbinazioak asko izan daitezkeela. Hori dela eta, hitz mailan egiten den lanak —hau da, *hitz multzo*en hasierak eta bukaerak aukeratzeak— garrantzi handia du; izan ere, geroz eta *hitz multzo*en hasiera eta bukaera posible gehiago aukeratu, orduan eta *hitz multzo* hautagai gehiago izango ditugu, eta, beraz, baita hautagaien konbinazio posible gehiago ere.

Hortaz, hiru ikasketa-funtzio daude guztira: iragazketako *start* eta *end* funtzioak, hitz bakoitza *hitz multzo* baten hasiera edo bukaera izan ote daitekeen erabaki beharko dutenak hurrenez hurren, eta *score* deiturikoa, *hitz multzo* hautagai bakoitzari puntuazio bat emango diona, hautagaitza horren sendotasunaren arabera.

Iragazketa (*filtering*) eta sailkapena (*ranking*) izeneko bi geruzatan egiten duelako lan eta *pertzeptroi*en algoritmoaren halako orokortze bat baliatzen duenez hiru ikasketa-funtzioak inplementatzeko, *FR-Perceptron* izena jarri zion Carreras-ek (2005) bere algoritmoari. Jarraian xehetasun handiagoz azalduko dugu.

### III.3.1 *FR-Perceptron* algoritmoa: iragazketa eta sailkapena, *pertzeptroi*ekin

*Pertzeptroi*en algoritmo tradizionalaren (ikus II.1.5.3 atala) halako orokortze bat da Carreras-en (2005) algoritmoa. *Pertzeptroi*en algoritmoen familiakoa izanik, erroreak gidatutakoa dela esan daiteke. *Hitz multzo*en identifikazio-prozesuan, algoritmoak iteratu egiten du $n$ aldiz; alegia, ikasketa-corpuseko adibide bakoitza $n$ aldiz bisitatzen da (*epoch-zenbakia* deitzen zaio parametro honi). *Start* eta *end* funtzioak esaldiko hitz bakoitzeko aplikatzen dira lehendabizi, eta *score* funtzioaren sarrera izango diren *hitz multzo*en hautagaiak definitzen dira honela. Gero, *score* funtzioa aplikatzen zaio, modu errekurtsiboan, *hitz multzo*en hautagai bakoitzari. Honela, *hitz multzo*en konbinazio onena aukeratzen da esaldiko. Egindako iragarpena okerra baldin bada, sailkatzaileak zuzentzen dira hurrengo iteraziorako, erregela sinple batzuen bidez. Esaldiarentzat soluzio onena bilatzen duenez, algoritmo globala dela esaten da (Carreras, 2005).

Ebatzi beharreko problemaren arabera, *hitz multzo*en egitura *jarraituak* edo *errekurtsiboak* bilatuko ditu algoritmoak. Hala, kateen kasuan, *hitz mul-*

*tzo*en egitura *jarraituak* izango ditugu, sintagmak eta aditz-kateak egitura jarraituak baitira (ikus III.2.1 eta III.2.2 atalak); perpausen kasuan, aldiz, *hitz multzo*en egitura *errekurtsiboak* izango ditugu, perpaus bat beste baten baitan joan daitekeelako, hain zuzen ere (ikus III.2.1 eta III.2.3 atalak).

Sailkatzaile guztiak *pertzeptroi*en algoritmoaren hiru aldaerarekin probatu zituen Carreras-ek (2005): *last*, *voted* eta *averaged*. Emaitza onenak *averaged perceptron* (Freund eta Schapire, 1999) delakoak eman zizkion, zeina *pertzeptroi*aren algoritmo klasikoaren hobekuntza sinple bat baita: ikasketa egiterakoan, algoritmo honek zenbait sailkatzaileren konbinazio moduko bat —batez besteko moduko bat— kalkulatzen du. Emaitza onak lortu dira algoritmo honekin HParen alorrean (Collins, 2002). Hori dela eta, hauxe izan da gure esperimentuetan erabili dugun aldaera: *averaged perceptron* delakoa, alegia.

Algoritmo honekin, literaturako emaitzarik onentsuenak lortu zituzten, bai kateen identifikazioan (Sang eta Buchholz, 2000), bai perpausen identifikazioan (Sang eta Déjean, 2001). Bi atazotan duen portaera onaz gain, ordea, erabili beharreko algoritmoak beste bi baldintza edo ezinbesteko ezaugarri ere izan behar zituen, ikasketa automatikoko algoritmo gehientsuenek betetzen dituztenak, bestalde:

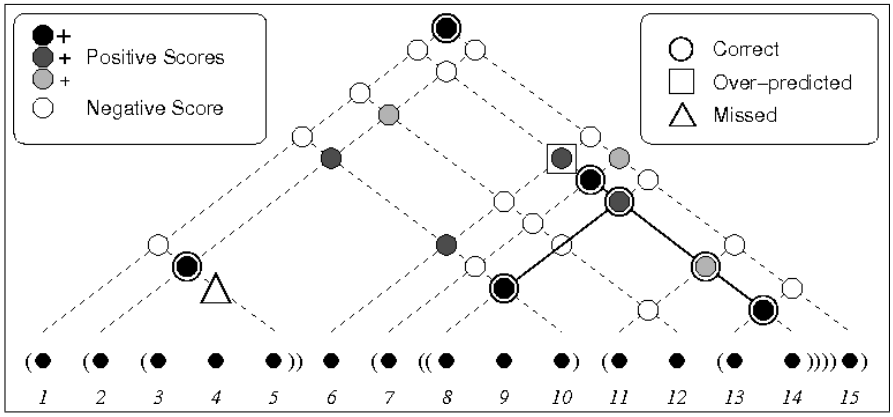
- Batetik, algoritmoak hizkuntza desberdinetara egokitzeko gaitasuna izan behar zuen, eta horretarako erraztasunak eskaini behar zituen.

- Bestetik, ikasketarako ezaugarri edo atributu berriak gehitzeko aukera eman behar zuen. Ezaugarri hau ezinbestekoa zen guretzat. Izan ere, ingeleserako erabiltzen zen corpusaren tamaina euskarakoa baino handiagoa zen, eta urritasun hau ekiditeko, euskarako ezaugarri linguistiko gehiago erabili nahi genituen. Gainera, erregeletan oinarritutako teknikak ere baliatu nahi genituen ikasketa automatikoaren emaitzak hobetzeko. Hain zuzen, erregelek emandako informazioa ezaugarri gisa gehitu nahi genuen (*stacking* edo *pilaratzea* deitzen den teknika erabiliz, alegia).

Ezaugarri hauek guztiak hartu genituen kontuan *FR-Perceptron* algoritmoa aukeratzerakoan. Datorren azpiatalean, algoritmoa nola dabilen erakusten duen adibide bat azalduko dugu.

### III.3.2 *Iragazketa eta sailkapena* algoritmoaren adibide bat

III.1 irudian, *iragazketa eta sailkapena* algoritmoa nola dabilen ikus daiteke. Azalpena sinplifikatzeagatik, *hitz multzo* oro hartzen da kontuan; berdin balio du azalpenak, hortaz, *hitz multzo* horiek kateak izan edo perpausak izan, baina adibidean hitz multzo errekurtsiboak agertzen direnez, perpausentzako egokiagoa dela esan genezake.




Irudia III.1: *Iragazketa eta sailkapena* algoritmoaren adibide bat, Carreras-en (2005) tesi-txostenetik egilearen baimenarekin hartutakoa.

Sarrerako esaldia irudiaren behealdean dago, hitz bakoitza ($x_1 \ldots x_{15}$) borobiltxo beltz txiki banarekin adierazia. Esaldiaren egitura zuzena —*hitz multzo*en konbinaketa egokia, algoritmoak identifikatzen saiatu behar duena, alegia— parentesi bidez adierazia dator borobiltxoetan. Hauek dira, hain zuzen ere, esaldiko *hitz multzo* zuzenak:

(1, 15) (2, 5) (3, 5) (7, 14) (8, 10) (8, 14) (11, 14) (13, 14)

Azaldu dugun moduan, *hitz multzo*ak identifikatzeko, hitz bakoitzari *start* eta *end* funtzioak aplikatzen zaizkio lehendabizi. Irudian, marra etenez adierazten dira hasiera eta bukaera posibleak. Piramidean gora (ezkerretik eskuinera) doazen marra etenek algoritmoak hasiera gisa markatutako hitzak adierazten dituzte ($x_1, x_2, x_6, x_7, x_8, x_{11}, x_{13}$); piramidean gora (baina eskuinetik ezkerrera) doazen marra etenek, aldiz, algoritmoak bukaera gisa markatutako hitzak adierazten dituzte ($x_5, x_{10}, x_{12}, x_{14}, x_{15}$). Kontuan izan iragarpen guztiak ez direla zuzenak. Adibidez, $x_3$ hitzak berez duen hasiera-marka ez da detektatzen, eta $x_6$ hitzak hasierarik ez badu ere, hasierako gisa hartu du

*start* ikasketa-funtzioak.

Hasierako eta bukaerako hitz posibleetatik ateratako marra etenen elkarguneek *hitz multzo* hautagaiak markatzen dituzte (zirkuluz marraztuak irudian). Esaterako, (7, 14) hautagaia osatzen dute $x_7$ hasierako hitzak eta $x_{14}$ bukaerakoak. Kontuan hartzekoa da, bestalde, 120 hautagai posibleetatik 27ra murriztu dela hautagaien espazioa, *start-end* funtzioen iragazpen-lana dela eta.

27 *hitz multzo* hautagaiak ebaluatu edo puntuatzen dira jarraian, *score* funtzioaren bitartez. Zirkuluetako grisaren eskala desberdinek hautagai bakoitzaren iragarpena adierazten dute. Zirkulu zuriak baztertu beharrekoak dira; grisek, berriz, hautagai bakoitza zenbaterainoko ziurra den adierazten dute: geroz eta beltzago, orduan eta ziurrago. Esaldia behetik gora aztertzen da, hautagai bakoitzari puntuazio bat emanez eta esparru bakoitzeko egitura onena mantenduz. Hala, *hitz multzo* bati aplikatutako puntuazioak erabil dezake bere baitan bildutako *hitz multzo*ena ere. (7, 14) *hitz multzo*a puntuatzean, adibidez, baztertu gabeko lau *hitz multzo*z osatutako hierarkia bat topatzen du bere baitan (marra jarraituaz markatua, irudian): (8, 10), (8, 14), (11, 14) eta (13, 14).

Hautagai guztiak puntuatu ondoren eta esaldiaren azterketa amaitutakoan, azken soluzio orokorra lortzen da. Adibidearekin jarraituz, hau litzateke algoritmoak lortzen duen *hitz multzo*en konbinazio egokiena, puntuazio orokor handiena lortzen duena:

(1, 15) (2, 5) (6, 14) (7, 14) (8, 10) (8, 14) (11, 14) (13, 14)

Irudian, zuzen iragarritako *hitz multzo*ak zirkulu batekin markatuak daude (zirkulu bikoitz gisa markatuak, alegia):

(1, 15) (2, 5) (7, 14) (8, 10) (8, 14) (11, 14) (13, 14)

Algoritmoak *hitz multzo* gisa markatu arren hala ez zirenak lauki batean bilduta daude: (6,14).

Algoritmoak markatu ez arren *hitz multzo* zuzenak zirenak, berriz, hiruki batekin markatuak: (3,5).

Doitasuna, beraz, 7/8 litzateke (iragarritako 8etatik 7tan asmatu baita); estaldura ere, kasu honetan, 7/8 da (zuzenak ziren 8etatik, 7 asmatu baitira).

Kontuan hartzekoa da, bestalde, hiru ikasketa-funtzioen errore lokal guztietatik ($x_{12}$ hitzari bukaera-marka jartzea edo (2, 10) *hitz multzo*ari puntuazio positiboa ematea, kasu), hiru soilik izan direla kritikoak, azken soluziorako kaltegarriak izan diren heinean: $x_3$ hitzak berez duen hasiera ez

detektatzea, $x_6$ hitzak hasierarik ez badu ere hasiera gisa markatzea, eta (6, 14) *hitz multzo*ari puntuazio positiboa ematea.

Hurrengo azpiatalean, algoritmoan egindako aldaketak azalduko ditugu.

### III.3.3 *FR-Perceptron* algoritmoan egindako egokitzapenak

Bi egokitzapen egin behar izan genituen *FR-Perceptron* algoritmoan.

Batetik, atributu berriak eransteko moldaketa batzuk egin behar izan genituen. Lehendik zerabiltzan atributuei (hitza eta kategoria), beste hauek gehitu genizkion hasiera batean: lema, azpikategoria, deklinabide-kasua eta mendeko perpaus mota (ikus III.5.1.3 eta III.6.1.2 atalak). Gerora, hizkuntzaren ezagutzan oinarritutako teknikak erabiliz lortutako informazioa (erregela bidez lortutakoa) erantsi genion *FR-Perceptron* algoritmoari. Honela, HPko atazak ebazteko erabiltzen diren bi hurbilpenak (hurbilpen linguistikoa eta hurbilpen estatistikoa) konbinatu genituen, III.5.2 eta III.6.2 ataletan ikusiko dugun moduan. *Stacking* teknika baliatu genuen honetarako, zeinaren jokabidea informazio berria atributu berri gisa gehitzean baitatza.

Bestetik, perpausen identifikazio automatikoari begira, *FR-Perceptron* algoritmoan esplizituki agertzen ziren ingeleseko zenbait termino euskarara ekarri behar izan genituen; alegia, *that, which* edo *who* gisako izenordain erlatiboak euskarako euren baliokideekin ordezkatu genituen: *non, zeina, zeinaren...*.

Euskarako forma hauek —ingelesekoen ordez— algoritmoan txertatuta lortutako hobekuntza adierazgarria izan ez arren, esperimentu guztiak forma hauekin egin genituen.

Jakina denez, euskaraz perpaus erlatiboak egiteko, ordea, izenordainak ez ezik, menderagailuak ere erabil daitezke ("*Zurekin etorri deN gizona*"). Hauek ez ditugu gehitu algoritmoan, txertaketa ez baitzen horren sinplea, baina etorkizunean halakoak tratatzeko modua bilatzea interesgarria litzatekeela iruditzen zaigu.

## III.4 Esperimentuen prestaketa

Atal honetan ikasketarako eta ebaluaziorako erabili dugun corpusa deskribatuko dugu. Bestalde, ebaluatzeko zein neurri erabili ditugun ere azalduko dugu, eta bukatzeko, esku artean darabiltzagun atazetarako lortutako *oinarrizko neurri*ak aurkeztuko ditugu, eta baita neurri hauek nola kalkulatu

diren azaldu ere.

*FR-Perceptron* algoritmoak, ikusi dugun gisan, esaldi osorako soluzio global bat proposatzen du, eta, beraz, esaldiko hitz guztiak hartzen dira kontuan soluzio onena bilatzeko. Hala ere, *start* eta *end* funtzioen emaitzak kalkulatzeko, leihoak markatzen dion hitzen informazioa hartzen du kontuan. Leiho hori, besterik adierazi ezean, (-2,+2) da; alegia, uneko hitzaren aurreko eta ondorengo bi hitzak hartzen dira kontuan. Kapitulu honetan aurkeztuko ditugun esperimentu guztietan, leiho hauxe erabili dugu. Leihoaren tamaina aldatuz probak egitea ez zaigu, kasu honetan, interesgarria iruditu, *FR-Perceptron* algoritmoak esaldi osoa hartzen duelako kontuan, azken batean.

### III.4.1 Corpusa

II.2.3.2 atalean esan dugun legez, euskara batuan idatzitako testuz osatutako corpusa da EPEC, eta hainbat mailatan etiketatua izan da: morfologia eta azaleko sintaxi mailan, lehendabizi, eta sintaxi maila sakonagoan, gero. 2006. urtean, hain zuzen, dependentziak etiketatu ziren, 200.000 hitz ingurutan (Aduriz *et al.*, 2006b). Dependentzietan oinarritutako etiketatzetik abiatuz, gainera, osagaietan oinarritutako etiketatzea lortu zen automatikoki. Osagaietan oinarritutako etiketatzean, jakina denez, kateak eta perpausak ondo mugatuta agertzen dira, nahiz eta esplizituki ez diren etiketatuak izan. Beraz, corpus hau erabili dugu, gure probak egin ahal izateko.

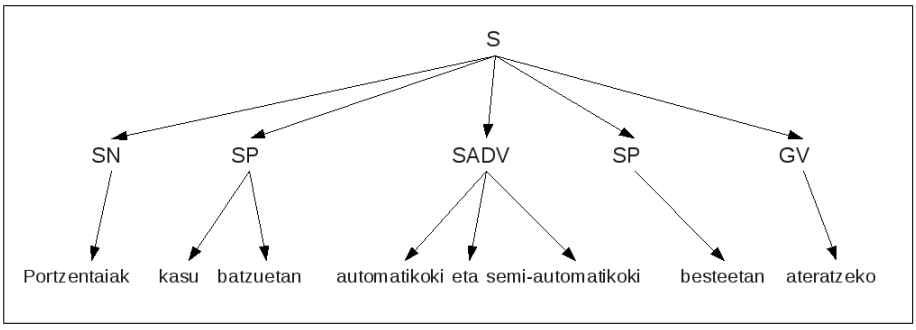
EPEC corpusean, esan dugun moduan, dependentziak etiketatu ziren, eta, programa informatiko baten bidez, osagaietan oinarritutako etiketatzera bihurtu zen corpusa. Dependentzietatik osagaietara pasatzeko urrats horretan, esaldi batzuk aparte utzi ziren —luzeegiak eta konplexuegiak zirelako, batik bat—. Corpusak, gainera, beste bihurketa bat behar zuen, *FR-Perceptron* algoritmoa erabili ahal izateko. Izan ere, *FR-Perceptron* algoritmoaren sarrera CoNLL batzarretan (Sang eta Buchholz, 2000; Sang eta Déjean, 2001) erabilitako bera da (*CoNLL formatua* deituko dioguna, hemendik aurrera). Osagaietatik *CoNLL formatu*ra pasatzerakoan ere, era bateko edo besteko akatsak medio —eskuzko etiketatzearen akatsak, normalean—, gauza bera gertatu zen: bihurketa prozesuan esaldi batzuk galdu egin ziren.

Izandako arazoen adibide gisa, azter dezagun III.2 irudia eta irudi honi dagokion III.3 irudiko zuhaitza, non osagaien formatutik *CoNLL formatu*rako bihurketa zuzena ezinezkoa dela erakusten baita; izan ere, irudi horretan ikus daitekeen moduan, osagaien etiketatzea ez da zuzena: "*automatikoki kasu batzuetan, eta semi-automatikoki besteetan*" hitzak ez daude behar be-

```
@@450321741,0069,11 Corpusaren bidez , maila bakoitzean ikasleek erabiltzen dituzten
 egitura linguistikoak aztertu eta portzentaiak automatikoki kasu batzuetan , eta
 semi-automatikoki besteetan , ateratzeko tresna informatikoak lantzen ari gara .

        ...
        (S
          (sn =func:CD=
            (grup.nom
              (w1201 portzentaiak portzentaje)))
          (sp =func:CC=
            (grup.nom
              (w1203 kasu kasu))
            (espec
              (w1204 batzuetan batzuk)))
          (sadv =func:CC=
            (sadv =func:CC=
              (w1202 automatikoki automatikoki))
            (coord
              (w1206 eta eta))
            (sadv =func:CC=
              (w1207 semi-automatikoki semi-automatiko+!ki)))
          (sp =func:CC=
            (espec
              (w1208 besteetan beste)))
          (gv
            (w1210 ateratzeko atera))))
        ...
```

Irudia III.2: Osagaien etiketatzean erroreak.




Irudia III.3: Osagaien etiketatzean errorea: zuhaitza.

zala etiketatuta: "*automatikoki kasu batzuetan*" hitzek adberbiozko sintagma bat osatu beharko lukete, eta "*semi-automatikoki besteetan*" hitzek beste bat; horren ordez, irudian ikus dezakegun moduan, hiru kate independente markatu dira: "*kasu batzuetan*", "*automatikoki eta semi-automatikoki*" eta "*besteetan*".

Modu honetako etiketatze-akatsak eta bestelako arazoak zirela eta, azkenean, 150.000 token ingururekin geratu ginen. Prozesu honek, bestalde, EPEC corpuseko etiketatze-arazoak azaleratzen lagundu zuen, eta EPEC corpusa zuzentzeko balio izan du.

150.000 token inguruko corpus hau, ikasketa automatikoko esperimentuetan ohi den moduan, hiru zatitan banatu genuen: % 70a, ikasketarako (*train*); % 15a, garapeneko probak (*develop*) egiteko; eta gainerako % 15a, azken proba (*test*) egiteko (ikus III.6 taulan, corpus bakoitzaren token kopuru zehatza). Gure ikasketa-eredua fintzeko, ordea, esperimentuak gehiegi ez luzatzearren, ikasketa-corpusaren % 25a erabili genuen. Emaitzarik onenak lortzen genituen *parametroak* ikasketa-corpus osoarekin baliatu genituen, gerora.

| | **Token kopurua** | **train** | **develop** | **test** |
|---|---|---|---|---|
| *EPEC* | 150.128 | 104.956 | 22.548 | 22.624 |

Taula III.6: Kateen eta perpausen identifikaziorako erabilitako corpusaren neurria.

Bestalde, ikasketa automatikoa egiterakoan, ikasi nahi dugunak eskuz etiketatua izan behar duen gisan, ikasteko darabilgun gainerako informazio linguistikoa komeni da analizatzaile automatikoen bidez lortutakoa izatea, emaitza errealistak lortu nahi badira. Kontuan hartu behar baita, hain zuzen ere, ikasi duguna, testu berri baten gainean aplikatzerakoan, testu berri horren analisi automatikoa soilik erabili ahal izango dugula, eta ez aditu batek eskuz etiketatua edo zuzendua.

Ikasketa-prozesuan erabilitako informazio linguistikoa automatikoki lortua izatea emaitzen kalterako izango da, noski; literaturan askotan aipatu den moduan, geroz eta informazio linguistiko hobea, orduan eta ikasketa-eredu hobea lortzen da. Baina, esan bezala, emaitzarik errealistenak lortuko ditugu modu horretan. Esan nahi baita automatikoki lortutako informazioa baliatuko dela edozein aplikazio praktiko egiteko, eta informazio hau erabiltzen

duen jokalekua dela, honenbestez, egoera errealena.

Ikasketa-prozesuan eskuzko informazio linguistikoa erabiltzeak, bestalde, gure ikasketa-ereduaren *benetako* ahalmena erakusten du: informazio linguistiko erabat zuzena lortuko luketen analizatzaile automatikoak bagenitu, erdietsiko genituzkeen emaitzak, hain zuzen.

Gauzak honela, ikasketa-ereduaren fintzea automatikoki lortutako informazio linguistikoa baliatuz egin dugu, eta emaitzarik onenak eman dizkigun *parametro*ekin errepikatu dugu ikasketa-prozesua, eskuz etiketatuko corpusa eta beraz eskuz etiketatutako informazio linguistikoa baliatuz[15]. Honela, analisi morfosintaktiko automatiko perfektu bat bagenu gure ikasketa-ereduaren emaitza optimoa —ideala— zein izango litzatekeen ikusi ahal izan dugu (ikus III.5.5 eta III.6.4 atalak).

CoNLL batzarreko esperimentuetan, bestalde, 260.000 token inguruko corpusa erabili zutela; zehatz esanda, 211.727 token ikasteko, eta beste 47.377 token probarako. Hau da, ingeleserako erabilitako ikasketa-corpusa gurearen bikoitza da gutxi gorabehera. Emaitzak konparatzeko orduan kontuan hartzekoa izango da datu hau. Bestalde, ingeleseko probetan erabilitako informazio linguistikoa ere, automatikoki lortutakoa da.

Hurrengo azpiatalean *CoNLL formatua* zertan datzan ikusiko dugu.

#### III.4.1.1 *CoNLL formatua*

*FR-Perceptron* algoritmoa erabili ahal izateko, CoNLL batzarreko 2000 eta 2001 urteetako ataza partekatuetan erabili zen *CoNLL formatu*ra ekarri behar izan genuen gure corpusa. Formatu honek baditu argitu beharreko zenbait berezitasun.

Kateen identifikaziorako, ikasketarako eta probarako corpusek, hasiera batean, espazio batez banaturiko hiru zutabe —hiru ezaugarri linguistiko— soilik dituzte: hitza (edo forma), kategoria eta katearen informazioari buruzko etiketa. Azken hau da, noski, sistemak ikasi beharrekoa, eta kate mota bakoitzerako bi balio izan ditzake:

- B-KATE, katearen hasiera adierazteko (adibidez, B-NP, (*begin noun-phrase*), sintagmaren hasiera adierazteko).

[15]Eskuz etiketatua diogunean, berez, eskuz desanbiguatua ulertu behar da; izan ere, analisi morfosintaktikoa automatikoki egina da, eta analisiaren emaitza posibleen artean zuzena aukeratzean datza eskuzko lana.

- I-KATE, katearen barnean —eta ez hasieran— dagoela adierazteko (adibidez, I-VP (*in verb-phrase*), aditz-kate barneko parte dela adierazteko).

Katearen informazioari buruzko etiketak "O" balioa ere har dezake (*out*), token hori ez baldin bada kate baten parte.

Bestalde, lerro bakoitza token bati dagokio, hau da, adibide bat izango da lerro bakoitza —instantzia bat— eta aipatutako hiru ezaugarrientzat balio bana izango du. III.4.1 adibidean, corpusaren formatua ulertzeko esaldi bat ikus daiteke. Zutabe bakoitzak informazio linguistiko desberdina dauka: lehenengoak, hitza; bigarrenak, berriz, kategoria; eta azken zutabean izango dugu ikasi beharrekoa: kasu honetan, kateei buruzko etiketa.

**Adibidea III.4.1**
Kateak ikasteko corpusaren formatua (CoNLL 2000 batzarrean erabilia), oinarrizko ezaugarri linguistikoekin (hitza eta kategoria), "*Niregana abiatu zen.*" esaldiarentzat:

| | | |
|---|---|---|
| *Niregana* | *IOR* | *B-NP* |
| *abiatu* | *ADI* | *B-VP* |
| *zen* | *ADL* | *I-VP* |
| *.* | *PUNT* | *O* |

Perpausen identifikaziorako ere formatu bera erabili zen. Kasu honetan, ordea, lau ezaugarri linguistiko erabili ziren ikasketarako eta probarako, CoNLL 2001 batzarrean: hitza (forma), kategoria, kateari buruzko informazioa eta perpausari dagokion etiketa. Azken hau da, noski, sistemak ikasi beharreko kontzeptua. Perpaus-muga etiketak honako balio hauek edo hauen konbinazioak izan ditzake ("*" zeinua, konbinazio bakoitzean, behin bakarrik agertuko da):

- (S*: perpausaren hasiera adierazteko.
- *S): perpausaren amaiera adierazteko.
- *: tokena ez da perpausaren hasiera, ezta perpausaren bukaera ere.

Hala nola, (S(S*S) etiketak zera esan nahi du: bi perpausen hasiera dela token hori, eta beste perpaus baten bukaera dela, era berean.

III.4.2 adibidean, ikasketa-corpusetik hartutako esaldi bat dugu. Zutabe bakoitzean informazio linguistiko desberdina daukagu: lehen zutabean, hitza dugu; bigarrenean, kategoria; hirugarrenean, kateari dagokion informazioa; eta azken zutabean ikasi beharreko *kontzeptua* izango dugu: kasu honetan, perpaus-mugari dagokion etiketa.

**Adibidea III.4.2**
Perpausak identifikatzen ikasteko corpusaren formatua (CoNLL 2001 batzarrean erabilitakoa), oinarrizko ezaugarri linguistikoekin (hitza, kategoria eta katea) "*Ogia egunekoa al den galdetzen du.*" esaldiarentzat.

| | | | |
|---|---|---|---|
| *Ogia* | *IZE* | *B-NP* | *(S(S** |
| *egunekoa* | *ADJ* | *B-NP* | * |
| *al* | *PRT* | *B-VP* | * |
| *den* | *ADT* | *I-VP* | **S)* |
| *galdetzen* | *ADT* | *B-VP* | * |
| *du* | *ADL* | *I-VP* | * |
| *.* | *PUNT* | *O* | **S)* |

## III.4.2 Ebaluaziorako neurriak

Ebaluazioari dagokionez, analizatzaile sintaktiko automatikoak ebaluatzeko PARSEVAL neurri hauek (Black *et al.*, 1991) erabili ohi izan dira: doitasuna (*precision*) eta estaldura (*recall*). Hartutako erabakien zuzentasuna neurtzen du doitasunak; estaldurak, berriz, zuzenak direnetatik asmatzen direnen portzentaia ematen du. HPan eskuz etiketatutako osagaiak hartzen dira zuzentzat. Hala, analizatzaile sintaktiko automatiko batek osagai bat zuzen etiketatu duela esango dugu, baldin eskuz etiketatutako osagaiaren kategoria sintaktiko bera izateaz aparte, token-sekuentzia bera hartzen badu bere baitan.

Neurri hauek guztiak III.7 gisako kontingentzia-taula batean oinarrituz kalkulatzen dira, bi klaseko emaitzak (Y eta N) ditugunean.

| | **Zuzena=Y** | **Zuzena=N** |
|---|---|---|
| **Esleitua=Y** | a | b |
| **Esleitua=N** | c | d |

Taula III.7: Estatistikak kalkulatzeko kontingentzia-taula.

"a" zenbakiak Y klasekoak diren eta Y klasea esleitu zaien elementuen kopurua adierazten du; "b" zenbakiak N klasekoak diren baina Y klasea esleitu zaien elementuen kopurua; "c" zenbakiak Y klasekoak diren baina N klasea esleitu zaien elementuen kopurua; eta "d" zenbakiak, berriz, N klasekoak diren eta N klasea esleitu zaien elementuen kopurua.

Doitasuna eta estaldura emaitzaren klase posible bakoitzeko kalkulatzen dira. Oro har, honela definitzen dira bi neurri hauek HPan, analizatzaileak

neurtzen gabiltzanean ($A_z$ analizatzaile automatikoak *zuzen* etiketatutako osagai kopurua izanik; $A_e$ analizatzaile automatikoak *etiketatutako* osagai kopurua izanik; $E_e$ *eskuz* etiketatutako osagai kopurua (zuzentzat hartutakoak) izanik):

$$Doitasuna = A_z/A_e$$
$$Estaldura = A_z/E_e$$

III.7 taulako kontingentzia-taula kontuan harturik, Y klaseko datuak, esaterako, honela kalkulatuko lirateke:

$$Doitasuna = a/(a+b)$$
$$Estaldura = a/(a+c)$$

Doitasunaren eta estalduraren artean erlazio matematiko zuzenik ez dagoen arren, estudio enpirikoetan ikusi ahal izan denez, alderantziz erlazionatuta daude; alegia, sistemak detektatutako elementuen kopurua handitzen bada —hots, estaldura handitzen bada—, doitasuna txikitzen da, eta alderantziz. Ondorioz, bi neurri hauek batera konparatzea ez da erraza. Horregatik, bi neurriok kontuan hartzen dituzten zenbait neurri proposatu izan dira. Gehien erabiltzen dena $F_\mathcal{B}$ neurria da.

$$F_\mathcal{B} = \frac{(\mathcal{B}^2+1)*Doitasuna*Estaldura}{(\mathcal{B}^2*Doitasuna+Estaldura)}$$

Normalean, $\mathcal{B} = 1$ erabiltzen da, doitasunari eta estaldurari pisu bera emanez:

$$F_1 = \frac{2*Doitasuna*Estaldura}{(Doitasuna+Estaldura)}$$

Kasu batzuetan, zehaztasuna edo *accuracy* izeneko neurria ere erabiltzen da: hartutako erabaki guztietatik zuzenak izan direnen portzentaia neurtzen du. Kontingentzia-taulako datuekin kalkulatzen da neurri hau ere (ikus III.7 taula):

$$zehaztasuna = (a+d)/(a+b+c+d)$$

Neurri hau, ordea, zenbait kasutan ez da nahikoa esanguratsua; izan ere, garrantzi bera ematen dio emaitza-klase bati edo besteari. Ataza batzuetan, ordea —klase bateko balio askoz gehiago ditugunetan, batez ere— normala izaten da askotan gertatzen den klaserako emaitza onak lortzea eta txarrak, berriz, gutxitan gertatzen den klaserako. Zehaztasunak emaitza-klaseak kontuan hartzen ez dituenez, aipatutako kasuetan ez da neurri esanguratsua izaten. Hori dela eta, klase bakoitzarekiko kalkulatzen den $F_1$ neurria erabiltzen da, oro har, eta hala egin dugu guk geuk ere.

Kateen identifikazioaren kasuan, beraz, honako hau adierazten dute neurriok:

- Doitasuna: automatikoki detektatuko kateetatik zenbat diren zuzenak (kate bat zuzentzat joko da eskuz etiketatuaren osagai berak eta kate-etiketa berdina baldin badauka).
- Estaldura: detektatu beharreko kateetatik (zuzenetatik, alegia) zenbat detektatu diren automatikoki.

Perpausen identifikazioaren kasuan, berriz, honako hau adierazten dute:

- Doitasuna: automatikoki detektatuko perpausetatik zenbat diren zuzenak.
- Estaldura: detektatu beharreko perpausetatik (zuzenetatik, alegia) zenbat detektatu diren automatikoki.

Neurri hauek berak erabili ziren ingeleseko kateen eta perpausen identifikazioko atazetan (Sang eta Déjean, 2001; Sang eta Buchholz, 2000).

### III.4.3 *Oinarrizko neurri*ak

Jakina denez, HPan, abiapuntu bat eman ohi da hasieran, ebatzi nahi den problemaren halako soluzio sinple bat, eta soluzio sinple honek lortutako emaitza gainditu beharrekoa izaten da. Horri deitzen zaio *baseline* edo *oinarrizko neurri*a, hain zuzen ere. Azpiatal honetan, kapitulu honi dagozkion bi atazetarako (kateak eta perpausak) lortutako *oinarrizko neurri*ak eta hauek lortzeko erabilitako heuristikoak azalduko ditugu.

Hala, kateen identifikazioan, *oinarrizko neurri*a kalkulatzeko, CoNLL 2000ko batzarrean erabilitako heuristiko bera baliatu genuen. Token bakoitzaren kateei buruzko informazioa honela kalkulatzen zen: hartu uneko

tokena, ikusi zein den bere kategoria (analizatzaile morfosintaktiko batek automatikoki lortua), eta kategoria horrek corpusean maizen zeukan katearen etiketa esleitu tokenari. Esaterako, III.4.1 adibideko *zen* tokena hartuko bagenu, bere kategoria *ADL* denez, kategoria horri ikasketa-corpusean maizen esleitutako kate-etiketa emango litzaioke: *I-VP* etiketa, alegia.

Heuristiko hau baliatuz, % 52,0 lortu genuen $F_1$ neurrian, ingeleserako emaitzak baino 25 puntu gutxiago (ikus III.8 taula).

| | | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|---|
| ***Oinarrizko neurri*a euskarako corpusarekin** | **Sintagmak** | 31,50 | 47,08 | 37,74 |
| | **Aditz-kateak** | 77,65 | 80,63 | 79,11 |
| | **Kateak** | 45,76 | 60,21 | 52,00 |
| ***Oinarrizko neurri*a ingeleseko corpusarekin** | **Kateak** | 72,58 | 82,14 | **77,07** |

Taula III.8: Kateen identifikazioko *oinarrizko neurrien* konparaketa, euskarako eta ingeleseko corpusen artekoa.

Perpausen identifikazioan, berriz, hartutako abiapuntua CoNLL 2001eko batzarrean erabilitako heuristikoa izan zen: esaldiaren hasierako eta bukaerako tokenei soilik jartzen zitzaien perpaus-muga, esaldia puntutik puntura doan unitate gisa ulertuta. Modu honetan, % 48,79ko $F_1$ neurria lortu genuen. Ingeleserako, aldiz, % 47,71ko $F_1$ neurria lortzen zen (ikus III.9 taula).

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **Euskarako *oinarrizko neurri*a** | 91,41 | 33,27 | **48,79** |
| **Ingeleseko *oinarrizko neurri*a** | 98,44 | 31,48 | **47,71** |

Taula III.9: Perpausen identifikazioko *oinarrizko neurri*en konparaketa, euskarako eta ingeleseko corpusen artekoa.

Heuristiko berak aplikatuz bi hizkuntzen garapen-corpusean, esaldien identifikazioarekin lortutako emaitzak antzekoak direla ikus dezakegu, baina kateen identifikazioan alde nabarmena dagoela euskarako eta ingeleseko emaitzen artean (25 puntu). *Oinarrizko neurri*en emaitzak arretaz begiratuta, badirudi sintagmak identifikatzearen ataza dela konplikatuagoa euskararen kasuan. Kontuan hartu behar da sintagmaren barruan sartzen direla —euskaraz— izen-sintagmak, adberbio-sintagmak eta adjektibo-sintagmak,

eta, gainera, sintagma bakartzat hartzen direla postposizio-lokuzioekin osatutakoak eta baita zenbait koordinazio ere. Euskara hizkuntza eranskaria izateak eta horrek dakarren kasuistika zabalak ere izango du zerikusia, ziur aski.

Hurrengo ataletan aztertuko dugu *oinarrizko neurri* hauek nola eta zein mailatan hobetu ditugun, eta zein bide hartu dugun euskarako ikasketa-corpus —ingelesekoa baino— txikiagoak emaitzetan eragin handirik izan ez zezan.

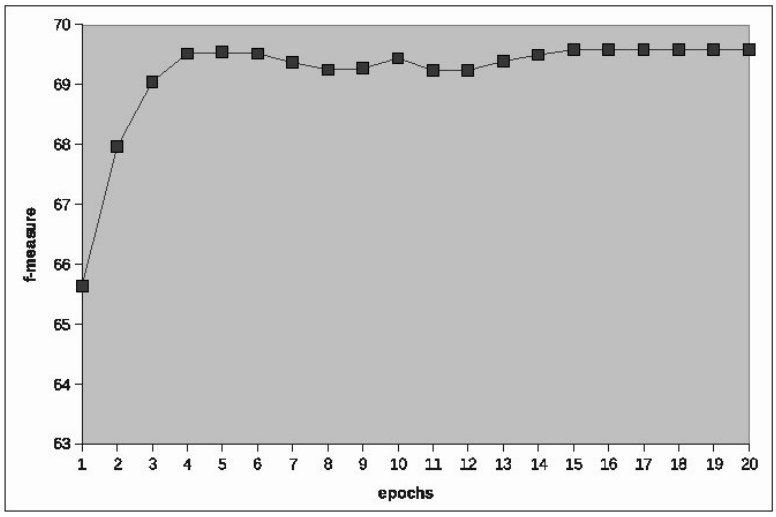
## III.5 Kateen identifikazio automatikoa

Esan dugun legez, kateen identifikazio automatikoa helburu, ikasketa automatikoko teknikak landu ditugu tesi-lan honetan, literaturan azken urteetan beste hizkuntzetarako egindako aurrerapenak geure egin nahian euskararako. Honekin, kateen identifikazioan IXA taldean aurrez egindako lanen jarraipena egin nahi izan dugu, azaleko sintaxiaren emaitzak hobetzeko asmoz.

Esan gabe doa erregela bidez eskuratutako emaitzak hobetzea zela gure helburuetako bat, euskarazko azaleko analizatzaile sintaktiko automatikoa hobetze aldera. Horretarako, ikasketa automatikoko ohiko algoritmo batzuez gain, *FR-Perceptron* algoritmoa baliatu dugu batez ere, EPEC corpusa erabiliz (% 70 ikasketarako, % 15 garapenerako eta % 15 testerako).

Sistemaren doitzea egiteko, dena dela, ikasketa-corpusaren % 25a erabili dugu hasieran (azken probak corpus osoarekin egin ditugun arren, III.5.4 atalean ikus daitekeen gisan). Gainera, ezaugarrien aukeraketa egin dugu, automatikoki etiketatutako corpusetan oinarritutako informazio linguistikoa baliatuz (emaitza errealistak lortze aldera). Horretaz gain, ordea, eskuz desanbiguatutako corpusarekin ere egin dugu proba bat, gure kate-identifikatzailearen benetako maila zein den jakitearren, eta analizatzaile morfosintaktiko automatikoaren desanbiguazio-prozesua hobetuz gero noraino irits gintezkeen jakiteko (ikus III.5.5 atala). Esperimentu guztiak garapen-corpusean ebaluatu ditugu, eta emaitzarik onenak eman dizkiguten *parametroekin*, test-corpusean egin dugu ebaluazio definitiboa (ikus III.5.6 atala).

Hurrengo atalean, beraz, kateen identifikazio automatikoan ikasketa automatikoa erabiliz lortutako aurrerapenak aztertuko ditugu.




Irudia III.4: *Epoch-zenbakiaren* eragina *FR-Perceptron* algoritmoaren bidez egindako ikasketa automatikoan (*f-measure* = F1).

## III.5.1 Kateen identifikazioa ikasketa automatikoa erabiliz

Aipatu dugun moduan, CoNLL 2000 batzarreko ataza partekatuan finkatutako corpusaren formatua erabili dugu, *FR-Perceptron* algoritmoarekin ikasketa automatikoa burutzeko.

### III.5.1.1 Epoch-zenbakiaren eragina

Ezer baino lehen, *epoch-zenbakia*ren eragina zenbatekoa den ziurtatu behar izan genuen, eta gure esperimentuetarako *epoch-zenbaki* egoki bat aukeratu modu horretan. *Pertzeptroi*en bidezko ikasketan, ikasketa-corpuseko adibide bakoitza zenbat aldiz aztertzen den erabakitzen da parametro honen bidez, arestian aipatu dugun moduan. Zenbait proba egin genituen, *epoch-zenbakia*ri 1etik 20rako balioak emanez. Proba hauetarako, CoNLL 2000 batzarreko ataza partekatuan erabilitako informazio linguistiko bera erabili zen: hitza eta kategoria.

III.4 irudian ikus daitekeen gisara, hamabosgarren *epoch-zenbaki*tik aurrera $F_1$ neurriak ez du ia aldaketarik jasaten. Carreras (2005) tesi-lanean egindako probetan ere, 15 *epoch-zenbaki*tik aurrera lortzen diren hobekuntzak minimoak dira. Kontuan hartu behar da, gainera, geroz eta *epoch-zenbaki*

handiagoa erabili, orduan eta denbora gehiago behar duela ikasketa-algoritmoak. Arrazoi hauek direla medio, esperimentu guztiak 15 *epoch-zenbakia* baliatuta egin genituen.

#### III.5.1.2 Lehen probak, oinarrizko ezaugarriekin

Lehen probetan, CoNLL batzarreko ataza partekatuan erabili zen informazio linguistiko bera (hitza eta *Eustagger*-ek emandako kategoria) baliatu genuen guk ere, kateei buruzko informazioa ikasteko. *Oinarrizko ezaugarriak* deituko diegu abiapuntu gisa erabilitako ezaugarri linguistiko hauei. Esan bezala, erabilitako *epoch-zenbakia* 15ekoa izan zen. III.10 taulan ikus daitekeen moduan, oinarrizko neurriarekiko ia 28 puntuko hobekuntza lortzen du.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **Sintagmak** | 60,54 | 55,49 | 57,90 |
| **Aditz-sintagmak** | 84,98 | 88,40 | 86,66 |
| **Kateak** | **70,85** | **68,36** | **69,58** |

Taula III.10: *FR-Perceptron* algoritmoan oinarritutako kateen identifikatzailearen emaitzak, ikasketa-corpusaren % 25arekin eta *oinarrizko ezaugarriak* erabilita, eta *epoch-zenbakia* 15 izanik. Garapen-corpusaren gainean egindako ebaluazioa (automatikoki desanbiguatutako informazio linguistikoa erabilita).

Deigarria da zer emaitza onak lortzen diren aditz-kateen identifikazioan, sintagmenekin alderatzen baditugu: ia hogeita bederatzi puntu inguruko aldea dago (% 86,66 vs % 57,90: ikus III.10 taula). Kontuan hartu behar da, hala ere, oinarrizko neurrietan 40 puntu baino gehiagoko aldea zegoela (ikus III.8 taula). Ikasketa automatikoa egiteko erabilitako informazio linguistikoak ere, badu honetan zerikusirik. Izan ere, uste dugu hitzen kategoria linguistikoa aditz-kateak detektatzeko oso lagungarria dela. Sintagmak detektatzeko ere, laguntzen du kategoriak, baina badirudi ez dela nahikoa. Aditz-kateen kasuan, alta, nabarmena da kategoriaren garrantzia.

Azter dezagun hau polikiago, corpusetik ateratako adibide batzuen bidez. Parentesi artean, hitz bakoitzari analizatzaile morfosintaktiko automatikoak esleitutako kategoria jarri dugu (IZE: izena; ADI: aditza; ADL: aditz-laguna; LOT: lokailua; ADB: adberbioa):

**Adibidea III.5.1**

1. *Kosovoko (IZE) hauteskundeen(IZE) emaitzak(IZE) onartzeko(ADI) eskatu(ADI) diote(ADL) EEBBek(IZE) Jugoslaviako(IZE) Gobernuari(IZE).*
2. *Izan ere (LOT), atzealdea (IZE) oso(ADB) kaltetua(ADI) dago(ADT).*
3. *Ezin (IZE) dut (ADT) gehiago(DET).*

III.5.1 adibidean ikus daitekeen eran, *ADI*, *ADT* eta *ADL* etiketek adierazten dituzte, oro har, aditz-kateetako elementuak. Sintagmen kasuan, hasteko, etiketa gehiago hartu behar dira kontuan (IZE, ADJ, ADB...), eta, gainera, postposizio lokuzioen bidez edo koordinazio bidez osatutako sintagmak detektatu beharrak zailtasun handiagoa eransten dio atazari.

Bestalde, aditz-kateko elementuak identifikatzea ez da dirudien bezain erraza. Ikus, esaterako, III.5.1 adibideko hirugarren esaldia: *"Ezin dut"* aditz-kate gisa identifikatu beharko lukeen arren, analizatzaile morfosintaktikoak *"Ezin"* hitzari izen kategoria esleitzen dionez, sailkatzaileak izen-sintagma gisa etiketatzen du. Antzekoa gertatzen da beste forma hauekin ere: *"behar izan", "uste izan", "kontuan hartu", "ahal izan"*.... Hitz anitzeko unitate gisa ondo identifikatuz gero, arazo hau ekidin egingo litzateke.

Izen-sintagmen kasuan, berriz, III.5.1 adibideko lehen esaldian bertan ikus daitezke hauek identifikatzeko zailtasunak. *"Kosovoko hauteskundeen emaitzak"* elkarren segidan datozen hiru izen hauek, esate baterako, kate bakar gisa identifikatu beharko lituzke sailkatzaileak; *"EEBBek Jugoslaviako Gobernuari"* hiru izenen segidak, ordea, bi kate bereizi gisa: *"EEBBek"* alde batetik, eta *"Jugoslaviako Gobernuari"* bestetik.

15 *epoch-zenbakia* eta oinarrizko ezaugarriak erabilita lortutako emaitzek ($F_1$ neurria = % 69,58), III.4.3 puntuan aipatutako *oinarrizko neurri*arenak ($F_1$ neurria = % 52,00) hobetzen badituzte ere (ia 18 puntuko hobekuntza lortu dugu), ezaugarri berak erabiliz ingeleserako lortutako *FR-Perceptron* bidezko emaitza onenekin konparatuz gero ($F_1$ neurria = % 93,74; ikus III.2 taula), 24 puntuko aldea dagoela ikus dezakegu. Gogora dezagun, ingeleseko ikasketa-corpusa, une honetan, zortzi aldiz handiagoa dela.

### III.5.1.3 Ezaugarri linguistikoak gehituz

Corpus txikiagoa izatearen eragin negatiboa, ordea, informazio linguistiko aberatsagoa erabiliz konpentsatu nahi izan genuen. Honela, hitzaren eta kategoriaren informazioaz gain, euskarako analizatzaile/desanbiguatzaile morfosintaktiko automatikoak (*Eustagger*-ek) emandako informazio linguistiko

gehigarria ere erabili genuen: lema, azpikategoria, deklinabide-kasuari buruzko informazioa eta mendeko perpausen marka.

Ezaugarri hauek baliatuz, *feature selection* edo *ezaugarrien aukeraketa* deritzona egin genuen: ezaugarri bakoitzak sistemaren hobekuntzan zenbat laguntzen zuen aztertzea, alegia.

Hasiera batean, oinarrizko ezaugarriei, aldi bakoitzean, ezaugarri linguistiko berriak banan-banan gehitu genizkion, aztertzeko zeintzuk ziren emaitzak gehien hobetzen zituztenak. III.11 taulan ikus daitekeen moduan, ezaugarri berri bakoitzarekin, oinarrizko ezaugarriekin lortutako emaitzak hobetu ziren. Hori dela eta, beste proba bat egin genuen ezaugarri linguistiko guztiak batera gehituta.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **oin** | 70,85 | 68,36 | 69,58 |
| **oin + ak** | 70,78 | 69,15 | 69,96 |
| **oin + d** | 77,42 | 80,17 | **78,77** |
| **oin + l** | 71,53 | 68,91 | 70,20 |
| **oin + men** | 70,77 | 68,48 | 69,61 |
| **oin + ak + d + l + men** | 77,67 | 79,70 | 78,67 |

Taula III.11: *FR-Perceptron* algoritmoan oinarritutako kateen identifikatzailearen emaitzak garapen-corpusean, *epoch-zenbakia*= 15 izanik, ikasketa-corpusaren % 25arekin, eta *oinarrizko ezaugarri*ez (oin) gain, informazio linguistiko gehigarria erantsita (ak: azpikategoria; d: deklinabide-kasua; l: lema; men: mendeko perpaus mota).

Emaitzarik onenak (ikus III.11 taula) deklinabide-kasua gehitzean lortutakoak dira (ia hamar puntuko hobekuntza, oinarrizko ezaugarriekin lortutako neurriekiko). Izan ere, euskaraz, sintagma batean, mugatasun morfologiko bakarra dugu, oro har. Mugatasun hori sintagma-bukaeran doa ("*Zakur handi beltzarekin*"). Aipatutako adibideko sintagma aztertzen badugu, ikus dezakegu marka morfologikoa sintagma hori osatzen duen azken osagaiak daramala, "*beltz*arekin" hitzak, alegia. Horixe da, dakigun moduan, euskarako sintagmek betetzen duten propietate esanguratsuena, eta propietate horretxen ondorioa da deklinabidea kateen detekziorako —sintagmen detekziorako, zehazki— hain erabakigarria izatea.

Edutezko genitiboak eta leku-genitiboak (*noren* eta *nongo*) tarteko direnean, ordea, sintagmako hitz bakar batek baino gehiagok izan dezake mugatasun morfologikoa ("*Zakur handiaren belarriak*"). Analizatzaileak, hortaz,

kasu-marka egokiak eskuratu beharko ditu, sailkatzaileak kasu hauek ere kontuan har ditzan.

**Adibidea III.5.2**
*(Kosovoko (IZE) hauteskundeen(IZE) emaitzak(IZE)) onartzeko(ADI) eskatu(ADI) diote(ADL) (EEBBek(IZE)) (Jugoslaviako(IZE) Gobernuari(IZE)).*

Hala, deklinabidearen informazioarekin, lehen detektaezinak ziren kateak zuzen identifikatzen ditu sailkatzaileak. III.5.2 adibidean, genitiboarekin etiketatuta datozen hitzak hurrengoarekin kate berean bilduta ikus daitezke. Alegia, sailkatzailea gai da honelako kateak ere zuzen identifikatzeko.

III.12 taulan, sintagmen eta aditz-kateen datu zehatzak paratu ditugu, deklinabideak sintagmetan daukan eragina argiago ikustearren. Deklinabidearen informazioa gehituta soilik, hamabost puntu hobetu ziren sintagmen emaitzak; aditz-kateenak, aldiz, puntu bat baino gutxiago.

| | **Sint. $F_1$ neurria** | **AK $F_1$ neurria** | **$F_1$ neurria** |
|---|---|---|---|
| **Oin** | 57,90 | 86,66 | 69,58 |
| **Oin + d** | **73,28** | 87,33 | 78,77 |

Taula III.12: *FR-Perceptron* algoritmoan oinarritutako kateen identifikatzailearen emaitzen desberdintasuna, deklinabidearen informazioa erabilita eta erabili gabe (oin: *oinarrizko ezaugarriak*; d: deklinabidea; *epoch-zenbakia*: 15; ikasketa-corpusaren % 25arekin; Sint.: sintagmak; AK: aditz-kateak). Garapen-corpusaren gainean egindako ebaluazioa; automatikoki desanbiguatutako corpusa.

Genitiboak sintagmen identifikazioa zailtzen duen arren, deklinabide-kasuak sintagmak detektatzeko daukan garrantzia argia da, eta, beraz, ikasketa automatikoarekin lortutako emaitzak guztiz logikoak direla esan genezake. Edo, alderantziz; ikasketa-automatikoaren emaitzak erabil genitzake ondorioztatzeko hitz baten kategoriak eta deklinabide-kasuak osatzen dutela nagusiki euskarako sintagmak eta aditz-kateak detektatzeko behar-beharrezkoa den informazio linguistikoa. Hitzaren kategoria funtsezkoa da aditz-kateak detektatzeko: % 87ra hurbiltzen dira aditz-kateen identifikazioaren $F_1$ neurriko emaitzak, hitza eta kategoria soilik erabiliz. Sintagmen identifikazioan maila ona ziurtatzeko, berriz, kategoriaz gain, deklinabide-kasuari buruzko informazioa behar da. III.12 taulan ikus daitekeen moduan, % 73 lortzen da honela, sintagmen identifikazioan.

### III.5.2 Kateen identifikazioa, erregelak eta ikasketa automatikoa konbinatuz

Aurreko atalean aipatutako informazio linguistikoa erabiliz lortutako emaitzak nahiko onak izanik ere, beste proba bat geratzen zitzaigun egiteko: erregeletan oinarritutako kateen identifikatzailea eta ikasketa automatikoan oinarrituz guk sortutakoa konbinatzea, hain zuzen ere.

Honela, kateak detektatzeko garatutako erregelak hartu (Aranzabe, 2008), eta *stacking* edo pilaratzeko teknika erabiliz, ikasketa automatikoko algoritmoari informazio hori gehitu genion. Teknika honen bidez, informazio berria (kasu honetan, erregelen bidez lortutako kateei buruzko informazioa) ikasketa-corpusari, eta ondorioz baita garapen- eta test-corpusari ere, beste atributu baten gisan gehitzen zaio. Corpusek zutabe bat gehiago izango dute, eta honela, zutabe horretan, erregelek esleitutako balioa gordeko da, adibide edo instantzia bakoitzeko (lerro bakoitzeko, azken batean).

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **Erreg** | 50,06 | 52,98 | 51,48 |
| ***FR-Perceptron* oin** | 70,85 | 68,36 | 69,58 |
| ***FR-Perceptron* oin + Erreg** | 75,51 | 77,61 | 76,54 |
| ***FR-Perceptron* oin + dek** | 77,42 | 80,17 | 78,77 |
| ***FR-Perceptron* oin + dek + Erreg** | 77,68 | 80,80 | **79,21** |

Taula III.13: Erregeletan oinarritutako kateen identifikatzailearen emaitzak (Erreg) eta *FR-Perceptron* algoritmoan oinarritutako kateen identifikatzailearen emaitzak, *oinarrizko ezaugarri*ez gain (oin), deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) baliatuz. Automatikoki desanbiguatutako corpusak eta ikasketa-corpus osoaren % 25a erabilita. Garapen-corpusaren gainean egindako ebaluazioa.

III.13 taula aztertu aurretik taularen lehen lerroko neurriak nondik atera ditugun azaldu beharra dago. Emaitzak konparagarriak izateko, ebaluazioa corpus beraren gainean egin beharra dago. Hortaz, IXA taldean kateak detektatzeko egindako CG gramatika gure garapen-corpusaren gainean ebaluatu genuen, gure esperimentuen emaitzekin konparagarria izan zedin. Horrela lortu genituen lehen lerro honetako emaitzak. III.1 taulako emaitzekin konparatuz gero, oso emaitza baxuak dira hauek, baina kontuan hartu behar da III.1 taulako emaitzak eskuzko corpus ez-anbiguoaren gainean ebaluatu zire-

la, eta proba honetan erabili zen corpusa automatikoki lortutako informazio linguistikoz osatua dela.

Azter ditzagun, orain, III.13 taulako emaitzak:

- Erregelekin lortutako emaitzak baino dezente hobeak lortzen dira *FR-Perceptron* ikasketa algoritmoarekin.

- *FR-Perceptron* ikasketa-algoritmoari erregelek emandako informazioa gehitzen badiogu, emaitzak are gehiago hobetzen dira, nahiz eta hobekuntza ez izan guk espero bezain handia kasu batzuetan. Hala ere, McNemar testa[16] egin genuen, lortutako hobekuntzak estatistikoki esanguratsuak ziren ala ez ikusteko, eta bi kasuetan hala zela frogatu genuen: bai oinarrizko informazio linguistikoari erregelen informazioa eranstean, bai "oinarrizko informazioa + deklinabidea" konbinazioari erregelen informazioa gehitzean, bi-bietan erdiesten den hobekuntza esanguratsua da ($p < 0,05$), bigarren kasuan puntu erdi eskaseko aldea dagoen arren.

Beraz, ondorio hauek atera daitezke, datu hauetan oinarrituz:

- Erregela bidezko kate-identifikatzailearen emaitzak erruz hobetu ditugu (% 51,48 vs % 79,21). Izan ere, eskenatoki idealerako diseinatu ziren erregelak, token bakoitzeko analisi bakarra eta zuzena geneukanerako, alegia. Hala, eskenatoki erreal batean aplikatzerakoan, erregelak egiterakoan kontuan hartu ez zen anbiguotasunak eta errore-tasak eragin negatiboa du.

- Ikasketa automatikoko teknikak eta erregeletan oinarritutakoak konbinatzeak hobekuntzak dakartza, baina hobekuntza horiek handiagoak dira beti ere goi-mugatik urrun baldin badaude uneko emaitzak. Beste hitzetan esanda, geroz eta gertuago egon lor daitekeen emaitza onenetik, orduan eta hobekuntza txikiagoa lortzen da.

[16]McNemar testa (Everitt, 1992) bi sailkatzaileen arteko aldea esanguratsua den ala ez erabakitzeko erabiltzen da. Horretarako, corpusa bi zatitan banatu behar da: ikasketa-corpusa eta test-corpusa. Bi sailkatzaileak (A, B) ikasketa-corpus bera erabiliz ikasi ondoren, test-corpus beraren gainean ebaluatu behar dira. Hipotesi nuluaren arabera, A sailkatzaileak ondo eta B sailkatzaileak gaizki sailkatutako adibideen kopuruak A sailkatzaileak gaizki eta B sailkatzaileak ondo sailkatutako adibideen kopuruaren berdina izan behar du. Datu hauen arabera, $\chi^2$ testan oinarritzen da McNemar testa, hipotesi nulu hau uka daitekeen edo ez erabakitzeko. Hipotesia errefusatu baldin badaiteke, bi sailkatzaileen arteko aldea esanguratsua dela esaten da (Dietterich, 1998).

Ingelesarekin *FR-Perceptron* bidez lortutako emaitzetara iristen ez bagara ere (hamabost puntura dago, une honetan, gure sistema: % 93,74 vs % 79,21), gogoratzeko modukoa iruditzen zaigu proba hauek egiterakoan erabilitako euskarako eta ingeleseko ikasketa-corpusen tamainen desberdintasuna: zortzi aldiz handiagoa zen ingelesekoa.

### III.5.3 Ikasketa automatikoko algoritmoa baloratuz

Puntu honetan, erabiltzen ari ginen algoritmoari euskarako kate-identifikatzailea sortzeko aurreikusi genion nagusitasuna ziurtatu nahi izan genuen, neurri batean.

Proba gisa, Weka paketeko erabaki-zuhaitzak (C4.5 inplementazio ezaguna) eta *support vector machines* algoritmoak baliatu genituen; erabaki zuhaitzak, batetik, algoritmo sinple eta HPrako erabilerrazak direlako, eta SVM, bestetik, gaur egun erabilienetakoa izateaz gain, emaitza onentsuenak lortzen dituelako azaleko sintaxiari dagozkion hainbat atazatan.

Konparazioa ahalik eta ondoen egiteko, baldintza berdinetan egiten saiatu ginen: *FR-Perceptron* algoritmoan erabilitako ezaugarri berak baliatu genituen, batetik (oinarrizko ezaugarriak, deklinabidea eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa); eta bestetik, *FR-Perceptron* algoritmoan *start* eta *end* funtzioetan aplikatzen den (-2,+2) leihoa baliatu genuen. Gainera, garapen-corpus beraren gainean egin genituen ebaluazio guztiak.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| ***FR-Perceptron* oin + dek + Erreg** | 77,68 | 80,80 | **79,21** |
| **C4.5 oin + dek + Erreg** | 73,71 | 80,80 | 77,09 |
| **SVM oin + dek + Erreg** | 66,16 | 79,91 | 72,39 |

Taula III.14: Kate-identifikatzailearen emaitzak ikasketa-algoritmo desberdinen arabera, automatikoki desanbiguatutako ikasketa-corpusaren % 25a erabiliz, eta *oinarrizko ezaugarri*ez (oin) gain, deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) eta (-2,+2) leihoa baliatuz. Garapen-corpusaren gainean egindako ebaluazioa.

III.14 taulan ikus daitekeen moduan, *FR-Perceptron* algoritmoak euskarako ikasketa-corpusa baliatuz sortzen duen sailkatzailea erabaki-zuhaitzek eta SVM-k sortutakoak baino hobea da.

### III.5.4 Ikasketa-corpusa handituz

Gauzak horrela, 105.000 token inguruko corpusa baliatu genuen ondoren, ikasketa automatikorako: corpusaren tamaina handitzeak ekar zitzakeen onurak aztertu nahi genituen. Emaitzak konparatu ahal izateko, esan gabe doa ikasketa-corpusaren % 25a erabilita emaitzarik onenak eman zizkiguten atributuak erabili genituela proba hauetan ere: oinarrizko atributuak, deklinabide-kasua eta erregelek emandako informazioa. Garapen-corpus bera erabili genuen ebaluaziorako.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **Ikasketa-corpusaren % 25arekin** | 77,68 | 80,80 | 79,21 |
| **Ikasketa-corpusaren % 50arekin** | 79,46 | 83,06 | 81,22 |
| **Ikasketa-corpusaren % 100arekin** | 81,09 | 84,24 | **82,64** |

Taula III.15: Oinarrizko atributuak eta deklinabide-kasua baliatuta, *FR-Perceptron* ikasketa-algoritmoa eta erregeletan oinarrituak konbinatuz lortutako kate-identifikatzaileak garapen-corpusean erdietsitako emaitzak, ikasketa-corpusaren tamainaren arabera (% 100 = 104.956 token izanik). Automatikoki desanbiguatutako corpusak baliatuta.

III.15 taulan ikus daitekeen moduan, ikasketa-corpusa lehen aldiz bikoiztean (% 25etik % 50era), 2 puntuko hobekuntza lortzen da, eta bigarren aldiz bikoiztean (% 50etik % 100era), 1,42koa. Beraz, hobekuntza txikitzen ari den arren, badirudi marjina badagoela oraindik, ikasketa-corpusa handituz, emaitzak hobetzekoa. Etorkizuneko eginkizun gisa utzi dugu corpusa handituz emaitzak hobetzeko ahalegina egitea.

### III.5.5 Emaitza idealak, eskuz desanbiguatutako informazio linguistikoa erabiliz

Aurreko azpiatalean azaldu dugun moduan, orain arteko probetan kateak ikasteko erabilitako informazio linguistikoa (deklinabidea, azpikategoria edo kateen erregela bidezko informazioa) automatikoki lortua izan da, ikasi behar genuena izan ezik, noski: kateen informazioa. Hala, lortutako emaitzak errealistak direla aipatu dugu; izan ere, testu errealekin gabiltzanean, eskura izango dugun informazio linguistikoa *Eustagger*-ek lortutakoa izango da.

Hala eta guztiz ere, interesgarria iruditu zitzaigun eskuz desanbiguatutako informazioarekin —informazio linguistiko *zuzenarekin*— eta automatikoki

erauzitako informazio linguistiko ez hain fidagarriarekin lortutako emaitzak konparatzea; modu honetan, *Eustagger*-en desanbiguazio-prozesua hobetuz gero lortuko genituzkeen emaitzak azter genitzake. III.16 taulan ikus daitezke emaitza hauek. Kontuan izan izen-sintagmen eta aditz-kateen $F_1$ neurriak ere eman ditugula taula horretan, ohiko $F_1$ neurri orokorrarekin.

| **Corpusa nola desanbiguatua** | **Doit.** | **Est.** | **Sint.** $F_1$ | **AK** $F_1$ | $F_1$ |
|---|---|---|---|---|---|
| *Autom* | 81,09 | 84,24 | 78,00 | 89,84 | **82,64** |
| *Eskuz* | 89,61 | 91,46 | 87,85 | 94,52 | **90,52** |

Taula III.16: Automatikoki edo eskuz desanbiguatutako corpusa (% 100) erabiltzearen eragina kate-identifikatzailean, oinarrizko informazioa, deklinabidearena eta erregeletan oinarritutako kateena erabiliz (Sint.: sintagmak; AK: Aditz-kateak; Eskuz: eskuz desanbiguatutako corpusa; Autom: automatikoki desanbiguatutako corpusa). Garapen-corpusaren gainean kalkulatutako neurriak (automatikoki ala eskuz desanbiguatua, proba bakoitzean erabilitako ikasketa-corpusaren modu berean).

Emaitza hauek aztertuz gero, zenbait ondoriotara irits gaitezke:

- Geroz eta informazio linguistiko hobea izan, orduan eta emaitza hobeak lortzen dira. Hala, IXA taldearen analizatzaile morfosintaktikoaren eragina nabarmena da eta estatistikoki esanguratsua, McNemar testaren arabera ($p < 0,05$). Analizatzaile morfosintaktikoaren erroreak direla-eta, ia zortzi puntu jaisten dira emaitzak (% 82,64 vs % 90,52). Emaitza logikoak, inondik inora, deklinabidea barne hartzen duen desanbiguazio automatikoak (3. maila) hamar puntu inguruko errore-tasa duela (Ezeiza, 2002) kontuan izanik (ikus III.17 taula). Informazio linguistiko sofistikatua automatikoki lortzea ez da erraza, eta tresna baten errore-tasak eragin zuzena du tresna hori darabilen tresnaren errore-tasan.

| **Desanbiguazio maila** | $F_1$ |
|---|---|
| 1. mailan | 95,92 |
| 2. mailan | 95,42 |
| 3. mailan | 90,36 |

Taula III.17: *Eustagger*-en emaitzak, desanbiguazio mailaren arabera.

- Eskuz desanbiguatutako corpusa erabiltzetik *Eustagger*-ek desanbiguatutakoa erabiltzera pasatzean, sintagmen emaitzek beherakada handiagoa dute (hamar puntuko galera sintagmek; 5 puntukoa aditz-kateek). III.17 taulari erreparatuta aurki daiteke honen arrazoia: 1. mailako desanbiguazioak (aditz-kateentzat garrantzitsua den kategoriari buruzko informazioa desanbiguatzen duenak) bost puntuko errore-tasa daukala ikus dezakegu; 3. mailakoak (sintagmentzat garrantzitsua den deklinabideari buruzko informazioa desanbiguatzen duenak), berriz, hamar puntukoa.

### III.5.6 Azken emaitza, test-corpusean

Proba guztiak garapeneko corpusarekin ebaluatu genituen, eta bukatzeko, emaitzarik onenak lortzeko baliatutako *parametroak* erabiliz, test-corpusa erabili genuen azken ebaluazioa egiteko. III.18 taulan ikus daitekeen moduan, pareko emaitzak lortzen dira, test-corpusean lortutako emaitzak apur bat hobeak diren arren (test-corpusa garapen-corpusa baino sinpleagoa izango dela pentsatzera garamatza honek). Lortutako emaitzak bi ebaluazio-corpusekin antzekoak izateak, dena dela, emaitza hauek fidagarriago bihurtzen ditu.

| **Proba-corpusa** | **Doit.** | **Est.** | $F_1$ |
|---|---|---|---|
| Garapen-corpusa | 81,09 | 84,24 | **82,64** |
| Test-corpusa | 81,35 | 85,07 | **83,17** |

Taula III.18: Automatikoki desanbiguatutako corpusa (% 100), oinarrizko informazioa, deklinabidearena eta erregeletan oinarritutako kateena erabiliz, automatikoki desanbiguatutako garapen- eta test-corpusean lortutako emaitzak.

## III.6 Perpausen identifikazio automatikoa

Esaldien eta perpausen identifikazioa ez da kateen identifikazioa bezain erraza. Izan ere, aurreko atalean ikusi dugun legez, kateak bata bestearen baitan sar ez daitezkeen *hitz multzo* jarraitu gisa soilik kontsideratu ditugu; beste hitzetan esanda, kateei ez diegu izaera errekurtsiboa aitortu; hau da, kate baten

baitan beste kate bat ezin egon daitekeela suposatu dugu. Alta, perpausekin eta esaldiekin, ezin dugu suposizio hori egin. Baiki, perpausak definizioz errekurtsiboak dira. Alegia, egon daiteke perpaus bat beste perpaus baten barruan. Hala eta guztiz, perpausak ere ezin dira elkar gainjarri (ikus III.2.1 eta III.2.3 atalak, definizio zehatz eta formalak gogora ekartzeko).

## III.6.1 Perpausen identifikazioa ikasketa automatikoa erabiliz

Gauzak horrela, kateen identifikazioan baliatutako algoritmo bera erabili genuen perpausak identifikatzeko ere: ikasketa automatikoko *iragazketa eta sailkapena* teknika, *pertzeptroi*ekin garatutakoa, hain zuzen ere (Carreras *et al.*, 2005). Izan ere, arestian esan dugun moduan, algoritmo honek *hitz multzo*en identifikazioa lantzen du, edozein delarik ere *hitz multzo* horren izaera: kateak, perpausak, entitateak... Arrazoi hau ere funtsezkoa izan zen algoritmo honen aldeko erabakia hartzeko.

Gainera, *CoNLL 2001* batzarreko zeregin partekatuan —perpausen identifikazioan (Sang eta Déjean, 2001), hain zuzen— frogatuta geratu zen ikasitako sailkatzaileak identifikazio-prozesuko etapa guztietan aplikatzea dela emaitza onak lortzeko modurik eraginkorrena, *FR-Perceptron* algoritmoak egiten duen moduan.

*Pertzeptroi*ekin garatutako *iragazketa eta sailkapena* teknika erabili ahal izateko, *CoNLL 2001* biltzarreko zeregin partekatuan erabili zen formatura bihurtu genuen gure corpusa. Kasu honetan ere erabilitako *epoch-zenbakia* 15 izan zen. Kateekin egindako esperimentuak eta Carreras (2005) tesi-lanean egindakoek zenbaki egokia dela erakutsi dute.

### III.6.1.1 Lehen probak, oinarrizko ezaugarriekin

Lehen probetan, CoNLL 2001eko batzarrean baliatu zen informazio linguistiko bera erabili zen (hitza, *Eustagger*-ek emandako kategoria eta deskribatu berri dugun kate-identifikatzaileak emandako kateei buruzko informazioa). *Oinarrizko ezaugarriak* deituko diegu abiapuntu gisa erabilitako ezaugarri linguistiko hauei.

III.19 taulan oinarrizko ezaugarriekin lortutako emaitzak ikus daitezke, *oinarrizko neurri*ekin konparatuta. Puntutik punturako hitzen segidak esaldi gisa hartuta kalkulatu ziren *oinarrizko neurri*ak (ikus III.4.3 atala xehetasun gehiagorako).

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| ***Oinarrizko neurri*a** | 91,41 | 33,27 | 48,79 |
| ***FR-Perc.* oinarrizko ezaugarriekin** | 76,72 | 64,34 | **69,99** |

Taula III.19: *FR-Perceptron* algoritmoan oinarritutako perpaus-identifikatzailearen emaitzak (*oinarrizko ezaugarriak* erabilita, *epoch-zenbakia* 15 izanik eta ikasketa-corpusaren % 25 —automatikoki desanbiguatua— erabilita) eta *oinarrizko neurri*ekin konparaketa. Garapen-corpusaren gainean egindako ebaluazioa.

III.19 taulan ikus daitekeen moduan, *oinarrizko neurri*ak 22 puntutan gainditu genituen arren, urrun samar geratzen zitzaizkigun ingeleserako baldintza berdinetan lortutako 85 puntuak (ikus III.5 taula).

Hori dela eta, kateak detektatzean egin dugun gisan, ingeleserako erabili zuten ikasketa-corpus zortzi aldiz handiagoaren eragina, nolabait, informazio linguistiko gehiagorekin ordezkatu nahi izan genuen.

#### III.6.1.2 Informazio linguistikoa gehituz

Hala, aipatutako *oinarrizko ezaugarri* linguistikoez gain, eskura geneukan informazio linguistikoa ere erabili genuen: azpikategoria, deklinabide markari buruzko informazioa, lema eta mendekotasuna adierazten duen marka.

| | **Doitasuna** | **Estaldura** | $F_1$ **neurria** |
|---|---|---|---|
| **oin** | 76,72 | 64,34 | 69,99 |
| **oin + ak** | 76,52 | 66,49 | 71,15 |
| **oin + d** | 76,41 | 66,10 | 70,89 |
| **oin + l** | 75,66 | 67,58 | 71,39 |
| **oin + men** | 76,39 | 65,68 | 70,63 |
| **oin + ak + d + l + men** | 76,82 | 69,94 | **73,22** |

Taula III.20: *FR-Perceptron* algoritmoan oinarritutako perpausen identifikatzailearen emaitzak, *oinarrizko ezaugarri*ez gain (oin), bestelako informazio linguistikoa ezaugarri gisa gehituta (ak: azpikategoria; d: deklinabide-kasua; l: lema; men: mendeko perpaus-marka) eta ikasketa-corpusaren % 25 erabilita —automatikoki desanbiguatua—. Garapen-corpusaren gainean egindako ebaluazioa.

III.20 taulan dakartzagu ezaugarri hauek —banan-banan lehendabizi, eta

denak batera gero— erantsi ondoren lortutako emaitzak. Aipatutako informazio linguistiko guztia batera eranstean lortu ziren emaitzarik onenak: hiru puntuko hobekuntza erdietsi genuen perpausen *oinarrizko ezaugarriekin* lortutakoarekin konparatuta. Kateetarako hain garrantzitsua den deklinabidearen informazioa, perpausak identifikatzeko ez dirudi hain erabakiorra denik; ez, behintzat, beste ezaugarri linguistikoak baino gehiago.

### III.6.2 Perpausen identifikazioa, erregelak eta ikasketa automatikoa konbinatuz

Arestian aipatutako informazio linguistikoa erabiliz, ez genuen nahi adinako hobekuntzarik lortu. Emaitzak hobetzeko, ordea, erregela bidez lortutako informazioa ikasketa-automatikoarekin konbinatzea geratzen zitzaigun.

Honela, mugak detektatzeko garatutako erregelak hartu (Aduriz *et al.*, 2006c), eta berriz ere *stacking* edo pilaratzeko teknika erabiliz, ikasketa automatikoko algoritmoari informazio hori gehitu genion. Teknika honen bidez, informazio berria (kasu honetan, erregelen bidez lortutako mugei buruzko informazioa) ikasketa-corpusari, eta ondorioz baita garapen- eta test-corpusei ere, beste atributu baten gisan gehitzen zaio. Hala, corpusek zutabe bat gehiago izango dute, eta adibide edo instantzia bakoitzak atributu horrentzat daukan balioa —erregelek emandakoa— izan beharko du.

| | **Doitasuna** | **Estaldura** | $F_1$ |
|---|---|---|---|
| **erreg** | 50,84 | 48,63 | 49,71 |
| **oin** | 76,22 | 64,34 | 69,99 |
| **oin + ak + d + l + men** | 76,82 | 69,94 | 73,22 |
| **oin + ak + d + l + men + erreg** | 78,03 | 71,35 | **74,54** |

Taula III.21: *FR-Perceptron* algoritmoan oinarritutako perpausen identifikatzailearen emaitzak, *oinarrizko ezaugarri*ez gain (oin), bestelako informazio linguistikoa erabilita (ak: azpikategoria; d: deklinabide-kasua; l: lema; men: mendeko perpaus mota) eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (erreg) baliatuz; automatikoki desanbiguatutako ikasketa-corpus osoaren % 25 erabilita. Garapen-corpusaren gainean egindako ebaluazioa.

III.21 taulan ikus daitekeen gisan, emaitzak hobetu genituen; puntu bat baino gehixeagoko hobekuntza lortu genuen $F_1$ neurrirako, eta garrantzi-

tsuagoa dena, McNemar testa egin ondoren, hobekuntza hau estatistikoki esanguratsua dela frogatu genuen ($p < 0,05$). Hala, euskarako perpausen identifikatzaile automatikorako ere, erregelak eta ikasketa automatikoko teknikak uztartzeak bataren eta bestearen emaitzak hobetzen dituela frogatu genuen, eta hobekuntza hori estatistikoki esanguratsua dela, gainera.

CoNLL 2001 batzarreko ataza partekatuan ingelesarekin *FR-Perceptron* algoritmoa erabiliz egindako saioetan lortutako emaitzen azpitik daude, ordea, gure emaitzak (% 85,03 vs % 74,54).

### III.6.3 Ikasketa-corpusa handituz

Emaitzen aldea handi samarra zela ikusirik, eta ordura arte generabilen corpusa ingeleserako erabili zutena baino zortzi aldiz txikiagoa zela jakinik, geneukan corpusa bere osoan aprobetxatzea deliberatu genuen. Corpusa handituz zenbaterainoko hobekuntza lor zitekeen egiaztatu nahi izan genuen perpausen identifikazio automatikoaren eginkizunean ere. 105.000 token inguruko ikasketa-corpusa bere osoan baliatuz egin genuen proba, beraz. III.22 taulan ikusten den moduan, bi puntu inguruko hobekuntza lortzen da corpusa bikoizten dugun bakoitzean.

| | **Doitasuna** | **Estaldura** | $F_1$ |
|---|---|---|---|
| **Ikasketa-corpusaren % 25arekin** | 78,03 | 71,35 | 74,54 |
| **Ikasketa-corpusaren % 50arekin** | 78,70 | 74,21 | 76,39 |
| **Ikasketa-corpusaren % 100arekin** | 80,13 | 76,18 | **78,11** |

Taula III.22: Ikasketa automatikoan oinarritutako teknikak eta erregeletan oinarrituak konbinatuz lortutako perpaus-identifikatzailearen emaitzak, ikasketa-corpusaren tamainaren arabera (% 100 = 104.956 token izanik). Garapen-corpusaren gainean egindako ebaluazioa (corpus automatikoki desanbiguatuarekin).

Azken emaitza, hala eta guztiz ere, ez da ingelesekoaren mailara iristen (zazpi puntu gehiago lortzen dira ingeleserako). Arrazoi asko egon daitezke honetarako. Begi bistan dagoenetik hasita, euskarazko ikasketa-corpusaren tamaina ingelesekoaren erdia dela esan beharko genuke (105.000 token inguru euskararako vs 212.000 token inguru ingeleserako). Hala ere, darabilgun ikasketa-corpusaren tamaina ingelesekoaren parekoa balitz (orain duguna bikoiztuta), gehienez beste bi puntuko hobekuntza lor genezakeela ondoriozta

liteke, III.22 taulako progresioari erreparatzen badiogu.

Bestalde, *Eustagger*-ek ematen duen informazio linguistikoaren kalitateari erreparatu behar diogu. Esan dugun moduan, orain arte egindako probetan erabilitako informazio linguistikoa *Eustagger*-ek lortutakoa da; honen emaitzek, ordea, okerrera egiten dute pixkanaka, geroz eta desanbiguazio sakonagoa ematearekin batera, eta, ondorioz, baita eskaintzen duen informazio linguistikoaren kalitateak ere. Beste modu batean esanda, geroz eta informazio linguistiko konplexuagoa eskatu, orduan eta emaitza txarragoak lortzen ditu desanbiguatzaileak, eta honek ondorioak izan ditzake perpaus-identifikatzailearen emaitzetan. Horixe aztertuko dugu hurrengo atalean, perpaus-identifikatzailea sortzeko —*Eustagger*-ek lortutako informazioaren ordez— eskuz desanbiguatutakoa baliatuz.

### III.6.4 Emaitza idealak, eskuz desanbiguatutako informazio linguistikoa erabiliz

Orain arteko probetan perpausak identifikatzen ikasteko erabilitako informazio linguistikoa (kategoria, deklinabidea, azpikategoria, lema...) automatikoki lortua izan da, ikasi behar genuena izan ezik: perpausen informazioa. Hala, lortutako emaitzak errealistak direla aipatu dugu; izan ere, testu errealekin gabiltzanean, eskura izango dugun informazio linguistikoa *Eustagger*-ek lortutakoa izango da.

Hala eta guztiz ere, interesgarria iruditu zitzaigun eskuz desanbiguatutako analisi morfosintaktikoaren informazioa erabiltzea, eta hauen emaitzak automatikoki erauzitako informazio linguistikoarekin lortutakoekin konparatzea; modu honetan, analizatzaile morfosintaktiko automatikoaren desanbiguazio-prozesua hobetuz gero lortuko genituzkeen emaitzak aurreikus genitzakeen.

III.23 taulako emaitzak aztertuz gero, ikus daiteke *Eustagger*-en eta kate-identifikatzailearen errore-tasak ez duela eragin handirik perpaus-identifikatzailean: lortzen den hobekuntza ez da estatistikoki esanguratsua, McNemar testaren arabera ($p < 0,05$). Honek badu bere logika; izan ere, oinarrizko ezaugarri linguistikoak erabiltzetik, gerora gehitutako ezaugarri linguistikoak erabiltzera, lortu den hobekuntza hiru puntukoa soilik izan da (ikus III.20 taula), erregelekin lortutako hobekuntza kontuan hartu gabe. Dirudienez, informazio horren kalitatea ez da erabakigarria perpaus-identifikatzailearentzat.

| Corpusa nola desanbiguatua | Doit. | Est. | $F_1$ |
|---|---|---|---|
| *Autom* | 80,13 | 76,18 | 78,11 |
| *Eskuz* | 80,81 | 76,11 | 78,39 |

Taula III.23: Automatikoki edo eskuz desanbiguatuaren corpusa (% 100) erabiltzearen eragina perpaus-identifikatzailean, oinarrizko informazioa, azpikategoriarena, deklinabidearena, lemarena, mendekoena eta erregeletan oinarritutako mugena erabiliz (eskuz: eskuz desanbiguatutako corpusa; autom: automatikoki desanbiguatutako corpusa). Garapen-corpusaren gainean egindako ebaluazioa (automatikoki ala eskuz desanbiguatua, proba bakoitzean erabilitako ikasketa-corpusaren modu berean).

### III.6.5 Azken emaitza, test-corpusean

Proba guztiak garapeneko corpusarekin ebaluatu genituen, eta bukatzeko, emaitzarik onenak lortzeko baliatutako *parametroak* erabiliz, test-corpusa erabili genuen azken ebaluazioa egiteko.

| Proba-corpusa | Doit. | Est. | $F_1$ |
|---|---|---|---|
| Garapen-corpusa | 80,13 | 76,18 | 78,11 |
| Test-corpusa | 79,22 | 75,36 | 77,24 |

Taula III.24: Automatikoki desanbiguatutako garapen- eta test-corpusean lortutako emaitzen aldea, automatikoki desanbiguatutako ikasketa-corpusa (% 100), oinarrizko ezaugarriak, lema, azpikategoria, deklinabidea, mendekotasuna eta erregeletan oinarritutako perpausen informazioa erabiliz.

III.24 taulan ikus daitekeen moduan, kasu honetan test-corpusean lortzen ditugun emaitzak garapen-corpusekoak baino apur bat txikiagoak dira. Hala eta guztiz ere, desberdintasuna ez da handiegia, eta euskarako perpaus-identifikatzailearen neurria % 77 punturen bueltan finkatzeko balio digu proba honek.

## III.7 Ondorioak

Kapitulu honetan, laburbilduz, ikasketa automatikoko teknikak erabili ditugu euskarako azaleko sintaxiaren tratamenduan aurrera egiteko, bai kateen

identifikazio automatikoan, bai perpausen identifikazio automatikoan. Hala, *pertzeptroi*ekin egindako iragazketa eta sailkapena algoritmoa (*FR-Perceptron* izenekoa) baliatu dugu, ingeleserako emaitzarik onenetarikoak algoritmo horrekin lortzen zirela ikusita, eta mota desberdineko *hitz multzo*ak identifikatzeko bidea ematen duen algoritmo global bat delako. Modu honetan, orain arte geneuzkan euskarako kate- eta perpaus-identifikatzaileak (hizkuntza-ezagutzan oinarritutakoak) hobetu ditugu (ikus III.25 eta III.26 taulak).

| | **Desanbiguatua** | $F_1$ |
|---|---|---|
| **Erreg** | Autom | 51,48 |
| ***FR-P* oin + dek + Erreg** | Autom | **82,64** |

Taula III.25: Garapen-corpusean ebaluatutako kate-identifikatzailearen emaitza konparatiboak: erregeletan oinarritutakoa (Erreg) vs *FR-Perceptron* (*FR-P*) algoritmoan oinarritutakoa (*oinarrizko ezaugarri*ez (oin) gain, deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) eta 104.956 tokeneko ikasketa-corpusa baliatuz).

| | **Desanbiguatua** | $F_1$ |
|---|---|---|
| **Erreg** | Autom | 49,71 |
| ***FR-P* oin+ak+d+l+m+Erreg** | Autom | **78,11** |

Taula III.26: Garapen-corpusean ebaluatutako perpaus-identifikatzailearen emaitza konparatiboak: erregeletan oinarritutakoa (Erreg) vs *FR-Perceptron* (*FR-P*) algoritmoan oinarritutakoa (*oinarrizko ezaugarri*ez (oin) gain, azpikategoria (ak), deklinabidea (d), lema (l), mendekoen informazioa (m) eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (Erreg) eta 104.956 tokeneko ikasketa-corpusa baliatuz).

Emaitza hauek lortzeko bidean, ordea, zenbait eragozpeni aurre egin behar izan diegu. Alde batetik, etiketatutako corpus txikiaren desabantaila informazio linguistiko gehiagorekin orekatu dugu, nolabait. Hala, kateak identifikatzen ikasteko, kategoria —aditz-kateentzat— eta kategoria eta deklinabidea —sintagmentzat— behar-beharrezko informazioa dela frogatu dugu. Perpausen identifikazioan, berriz, gehitutako informazio linguistiko guztia baliagarria izan da: hitza, lema, kategoria, azpikategoria, deklinabidea eta mendeko perpausen informazioa.

| | **Ikasketa-corpusaren tamaina** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| ***Oinarrizko neurria*** | - | Autom | 52,00 |
| ***FR-P* oin** | *% 25* | Autom | 69,58 |
| ***FR-P* oin + dek** | *% 25* | Autom | 78,77 |
| ***FR-P* oin + dek + Erreg** | *% 25* | Autom | 79,21 |
| ***FR-P* oin + dek + Erreg** | *% 100* | Autom | **82,64** |
| ***FR-P* oin + dek + Erreg** | *% 100* | Eskuz | 90,52 |

Taula III.27: Garapen-corpusean ebaluatutako kate-identifikatzailearen emaitzen eboluzioa: *oinarrizko neurri*ekin hasi eta *FR-Perceptron* (*FR-P*) algoritmoan oinarritutako kate-identifikatzailearen emaitzetara, *oinarrizko ezaugarri*ez (oin) gain, deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) baliatuz; eskuz desanbiguatutako corpusa (Eskuz) edo automatikoki lortua (Autom) erabiliz; ikasketa-corpusaren tamaina osoa (% 100) 104.956 tokenekoa izanik.

Bestalde, ikasketa automatikoko teknikak hizkuntza-ezagutzan oinarritutako teknikekin —erregelekin, alegia— uztartu ditugu; hots, hizkuntzalaritza konputazionalean erabiltzen diren bi teknikak —hizkuntzaren ezagutzan oinarritutakoak eta corpusetan oinarritutakoak— konbinatu ditugu, eta emaitzak hobetu ditugu honela (ikus III.27 eta III.28 taulak). *Stacking* teknika erabilita gehitutako informazio honekin lortu dugun hobekuntza, gainera, bi kasuetan estatistikoki esanguratsua dela frogatu dugu, McNemar testaren bidez ($p < 0,05$).

Aipatzekoa da, halaber, eskuz etiketatuko corpusak erabiltzerakoan, emaitzak are gehiago hobetzen direla; izan ere, zenbat eta informazio linguistiko hobea, orduan eta emaitza hobeak lortzen dira (ikus III.27 eta III.28 tauletako azken bi lerroak). Hobekuntza hau estatistikoki esanguratsua da kateen identifikazioaren atazan, baina ez perpausen identifikazioarenean ($p < 0,05$). *Eustagger*-ek ematen duen informazio linguistikoa badirudi ez dela oso fina, zenbait kasutan. Kateen identifikazioari dagokionez, sintagmen eta aditz-kateen emaitzak bereiziz gero (gogoratu III.10 eta III.12 tauletako emaitzak), argi ikusten da sintagmetarako behar den informazioaren (deklinabidearena, seguruenik) desanbiguazioan daukan errore-tasa handiagoa dela *Eustagger*-en eragozpen handienetakoa; *Eustagger*-en portaera, hain zuzen, bost puntu inguru jaisten da, deklinabidearen informazioa ere eskatzen baldin bazaio.

| | **Ikasketa-corpusa** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| ***Oinarrizko neurri*ak** | - | Autom | 48,79 |
| ***FR-P* oin** | *% 25* | Autom | 69,99 |
| ***FR-P* oin+ak+d+l+m** | *% 25* | Autom | 73,22 |
| ***FR-P* oin+ak+d+l+m+Erreg** | *% 25* | Autom | 74,54 |
| ***FR-P* oin+ak+d+l+m+Erreg** | *% 100* | Autom | **78,11** |
| ***FR-P* oin+ak+d+l+m+Erreg** | *% 100* | Eskuz | 78,39 |

Taula III.28: Garapen-corpusean ebaluatutako perpaus-identifikatzailearen emaitzen eboluzioa: *oinarrizko neurri*ekin hasi eta *FR-Perceptron* (*FR-P*) algoritmoan oinarritutako perpausen identifikatzailearen emaitzetara, *oinarrizko ezaugarri*ez gain (oin), azpikategoria (ak), deklinabidea (d), lema (l), mendekoaren marka (m), eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (erreg) baliatuz; eskuz desanbiguatutako corpusa (Eskuz) edo automatikoki (Autom) lortua erabiliz; ikasketa-corpusaren tamaina osoa (% 100) 104.956 tokenekoa izanik.

Gainera, desanbiguatzailearen errore-tasa ia hamar puntukoa da maila honetan (ikus III.17 taula). Kateen identifikazioko emaitzei arreta jarriz gero, berriz, zortzi puntuko aldea dagoela ikus daiteke, desanbiguatzaile automatikoak emandako informazioa edo eskuz desanbiguatutakoa erabiltzearen artean. Beraz, esan dezakegu desanbiguatzaile automatikoa hobetzeak kateen identifikatzailearen hobekuntza zuzena ekar lezakeela.

Bestalde, esan beharra dago lortu ditugun euskarako kate- eta perpaus-identifikatzaileak aski erabilgarriak direla, nahiz eta ingeleseko emaitzen azpitik egon (ikus III.29 eta III.30 taulak).

| **Hizkuntza** | **Teknika** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| *Euskara* | *FR-P* oin + dek + Erreg | Autom | 83,17 |
| *Ingelesa* | *FR-P* | Autom | 93,74 |

Taula III.29: FR-Perceptron bidezko euskarako (test-corpuseko) eta ingeleseko kateen identifikatzaileen emaitzarik onenen konparazioa (automatikoki analizatutako eta desanbiguatutako corpusa erabiliz (Autom)). Euskarakoan, *oinarrizko ezaugarri*ak (oin), deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) baliatuta, eta ikasketa-corpusaren tamaina osoa (% 100 = 104.956 token) erabilita.

Emaitza onak lortu diren arren, ingeleseko kate- eta perpaus-identifikatzaile onenen emaitzekin erkatuta, apur bat azpitik geratu dira gureak: hamar puntu gutxiago eta zazpi puntu gutxiago, hurrenez hurren. Hainbat arrazoi topatu ditugu honetarako:

- Euskarako ikasketa-corpus txikiagoa (ingelesekoaren erdia).

- Kate-identifikatzailearen kasuan, *Eustagger*-en errore-tasa; izan ere, informazio linguistiko zuzenarekin —eskuz desanbiguatuarekin—, hobekuntza estatistikoki esanguratsuak lortzen dira kate-identifikatzailearentzat ($p < 0,05$).

- Kate-identifikatzailea egiteko zailtasun handiagoa euskararentzat. Baiki, hizkuntza bakoitzaren *oinarrizko neurri*ak konparatzen baditugu, ingelesekoak 25 puntu hobeak direla ikus dezakegu (% 77,07 vs % 52,00).

- Perpaus-identifikatzailea egiteko zailtasun handiagoa euskararentzat. Perpaus-identifikatzailearen euskarako eta ingeleseko *oinarrizko neurri*ak alderatuta, desberdintasuna ez da adierazgarria (puntutik punturako hitz multzoak esaldi gisa kontsideratzea antzekoa baita bi hizkuntzetan); dena dela, euskarak ezaugarri duen ordena libreak perpausen identifikazioa ingelesekoa baino zailagoa egiten duelakoan gaude.

| **Hizkuntza** | **Teknika** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| *Euskara* | *FR-P* oin+ak+d+l+m+e+Erreg | Autom | 77,24 |
| *Ingelesa* | *FR-P* oin | Autom | 84,36 |

Taula III.30: *FR-Perceptron* bidezko euskarako (test-corpuseko) eta ingeleseko perpaus-identifikatzaileen neurriak (automatikoki analizatutako eta desanbiguatutako corpusa erabiliz (Autom)). Euskarakoan, *oinarrizko ezaugarri*ak (oin), azpikategoria (ak), deklinabidea (d), lema (l), mendekoen informazioa (m) eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (Erreg) erabilita, eta ikasketa corpusaren % 100 (104.956 token) baliatuta.

Gainontzeko hizkuntza gehienetan ere, % 90etik gorako emaitzak lortzen dira kateen identifikazioan, $F_1$ neurrian.

Italierarako, esate baterako, $F_1$ = % 91,13 lortzen dute (Lenci *et al.*, 2001); koreerarako, berriz, % 95,36 eskuratzen dute (Lee *et al.*, 2005); txinerarako, % 95,23 (Liang *et al.*, 2007); arabierarako % 96,33 (Diab, 2007); portugeserako, izen-sintagmetan, % 89,14 (Milidiu *et al.*, 2008); eta hebreerarako, izen-sintagmetan, % 93,2 (Goldberg *et al.*, 2006). Testuingururik gabeko gramatika probabilistikoak eta 10 milioi hitzeko ikasketa-corpusa erabiliz, % 92 inguruko $F_1$ neurria lortu zuten (Schmid eta Walde, 2000) lanean, alemaneko izen-sintagmak identifikatzen. Darabiltzaten teknikei dagokienez, ikasketa automatikokoak nagusitzen dira orokorrean, eta, hauen artean, SVM eta CRF algoritmoak.

Gure emaitzen antzekoak lortzen dituzte, berriz, beste zenbaitek. Kroazierarako, erregeletan oinarritutako teknikak erabili dituzte, ikasketa-automatikoa egiteko nahikoa corpus etiketatu ez izateagatik, eta, hala, $F_1$ = % 90,80 lortzen dute, baina eskuz etiketatutako informazio linguistikoa erabiliz. Hindi hizkuntzarako egindako esperimentuetan, berriz, % 80,97ko $F_1$ neurria eskuratzen da; bengalerarako % 82,74; eta telugurako % 79,15 (Avinesh eta Karthik, 2007).

Perpausen identifikazioaren atazan, zailagoa izanik, emaitza eskasagoak lortzen dira, oro har. Hala, portugeserako, adibidez, % 69,31 besterik ez da lortzen $F_1$ neurrian, TBL algoritmoaren hedapena den entropiak gidatutako transformazioko ikasketa algoritmoa erabiliz (Fernandes *et al.*, 2010). Txinerarako % 78,15 eskuratzen dute SVM algoritmoaren aldaera batekin, baina eskuz etiketatutako informazio linguistikoa erabiliz (Zhou *et al.*, 2010). Koreerarako egindako lanean, perpausen hasiera- eta bukaera-markak identifikatzen soilik saiatzen dira, perpausak beren osoan identifikatu beharrean (Lee *et al.*, 2006). Errumanierarako lortzen dira ingelesez besteko emaitzarik onenak: $F_1$ = % 88,76 (Puscasu, 2004). Horretarako, MBL ikasketa-algoritmoa errumanierarako espresuki egindako erregela linguistikoekin uztartzen dute. Ingeleserako ere probatzen dute euren sistema (ingeleseko erregela espezifikoak sortuz), baina ez dira (Carreras *et al.*, 2005) lanaren emaitzetara iristen (% 82,36 vs % 85,03).

Hizkuntza bakoitzerako erabiltzen diren errekurtsoak desberdinak izanik, eta baita hizkuntza bakoitzaren ezaugarriak ere, lan hauen konparaketa objektibo bat egitea oso zaila da. Hala eta guztiz, analizatzaile morfosintaktiko egoki bat eta nahikoa corpus etiketatu izateak ezinbesteko baldintza dirudi ataza hauetan emaitza onak eskuratzeko.

Etorkizunean, beraz, ikasketa-corpusaren tamaina handitzea izango da euskarako kate- eta perpaus-identifikatzaileak hobetzeko eman beharreko urra-

tsetako bat. *Eustagger* hobetzeak —eta, honela, informazio linguistiko zuzenagoa izateak— ere, emaitzak hobetzeko balioko luke, batez ere, kateen identifikazioan.

Azken urteetan, ingeleseko kate- eta perpaus-identifikatzaileen emaitzak gehiegi hobetu ez izanak, bestalde, ataza horien ingeleseko goi-mugatik gertu daudela esan nahi du; euskarako tresna hauen goi-muga, berriz, informazio linguistiko zuzenarekin —eskuz desanbiguatuarekin— lortutakotik gertu ibil daitekeela uste dugu.

# IV. KAPITULUA

## Euskarako estilo- eta puntuazio-zuzentzailerantz: komaren zuzenketa automatikoa

Duela urte batzuk arte, hizkuntzalaritza konputazionalean, puntuazioa ez da kontuan hartua izan. Nunberg-en (1990) lan monografikoaz geroztik, ordea, puntuazioari buruzko lan konputazionalak ugaldu egin ziren, eta, egun, geroz eta gehiago jorratzen da puntuazioa, HPan.

IXA taldean, zuzentzaile ortografikoa garatu zen duela urte batzuk (Agirre *et al.*, 1992), eta gramatika-zuzentzailea osatzera bideratu dira azken urteetan hainbat ahalegin, bai zuzentzaile gramatikalerako beharrezkoak diren oinarrizko tresnak garatuz (Aranzabe, 2008; Aldabe *et al.*, 2005b; Arrieta *et al.*, 2003; Aduriz *et al.*, 2002), bai gramatika-erroreak detektatzeko eta zuzentzeko tresnak sortuz (Oronoz, 2009; Uria, 2009; Ansa *et al.*, 2004). Testuinguru honetan kokatzen da kapitulu honetan aurkeztuko dugun estilo- eta puntuazio-zuzentzailea: gramatika-zuzentzailea osatzeko egindako lana da, batetik; eta hizkuntzaren ulermen ahalik eta osoena lortzeko bidean urrats garrantzitsua, bestetik. Gainera, HPko zenbait aplikazio hobetzeko lagungarria dela uste dugu. Hala, kapitulu honetan, euskarako estilo- eta puntuazio-zuzentzaile bat lortzeko helburuarekin egin ditugun aurrerapenak aztertuko ditugu.

Lehen atalean, puntuazioari buruzko sarrera bat egingo dugu: puntuazioaren garrantzia nabarmenduko dugu, eta puntuazioak daukan rola aztertuko dugu, bai euskaran, bai beste zenbait hizkuntzatan ere. Bigarrenean,

HPan puntuazioari buruz egin diren lanak deskribatuko ditugu. Hirugarrenean, ezagutza linguistikorik gabe detekta daitezkeen puntuazio-erroreak aztertuko ditugu. Laugarrenean, komaren arauak formalizatzeko azterketa teorikoa egingo dugu. Bosgarrenean, berriz, komen zuzenketa jorratuko dugu, hizkuntza-ezagutzan oinarritutako tekniken bidez, eta, seigarrenean, ikasketa-automatikoa erabiliz koma-zuzentzaile bat garatzeko egin ditugun saioak azalduko ditugu. Ondorioen azalpenarekin bukatuko dugu kapitulua.

## IV.1 Sarrera

Puntuazio-zuzentzaile bat egitea ez da lan erraza. Hasteko, puntuazio-arauak hizkuntza askotan ez daude guztiz zehaztuta, eta zehaztuta daudenean ere, ez daude ondo zabalduta erabiltzaileen artean. Oro har, ez dago arazo handirik puntuarekin, galdera-markarekin edo harridura-markarekin. Kontuan hartzekoa da, hala ere —puntuaren detekzioan, esaterako—, laburduretan edo zenbakiekin erabilitako puntuek atazari eransten dioten zailtasuna. Hala eta guztiz, Santos-en (1998) iritziz, hiru hauek —puntua, galdera-marka eta harridura-marka— puntuazio-marka *fidagarriak* dira; beste guztiak, aldiz, ezin dira hala kontsideratu. Are gehiago, ikur *fidagarri* hauei dagozkien zenbait errore detektatzea (hizkuntzaren arabera galdera-markaren hasiera-marka dagoen ala ez, kasu), ez da hain zaila. Aitzitik, koma da arazo handienak sortzen dituen ikurra. Izan ere, koma da puntuazio-marketan gehien erabiltzen dena, eta baita modu zabalenean ere.

Meyer-ek (1987), adibidez, frogatu zuen *Brown* corpusean (Francis eta Kucera, 1979) ageri diren puntuazio-marken % 45a komak direla, eta beste % 45a, puntuak. Datu hauei erreparatuz gero, pentsa daiteke komak daukan garrantzi bera izan beharko lukeela puntuak ere. Alabaina, Jones-ek (1996b) frogatu zuen puntuak ez daukala, inondik ere, komaren *malgutasuna*. Bere lanean, puntuazio-marka bakoitzarentzat, rol sintaktiko posible guztiak zerrendatu zituen; bada, komak dauzkan rol sintaktikoak, hain zuzen ere, puntuazio-marken rol sintaktiko guztien % 73 dira.

Komaren anbiguotasuna frogatua izan da beste zenbait lanetan ere (Bayraktar *et al.*, 1998; Beeferman *et al.*, 1998; Delden eta Gomez, 2002). Lan hauek guztiek komari buruzko erregela finkoen eta normalizatuen hutsunea nabarmendu dute. Hots, koma da erabilera desberdin gehien dituen puntuazio-marka, eta baita gutxien arautua dagoena ere (Bayraktar *et al.*, 1998; Hill eta Murray, 1998). Badira nahiko modu orokorrean onartuak dauden

zenbait erregela, baina beren erabilera ez da oso estandarra.

> *"[. . . ] Puntuazioari dagozkion arazoen artean, komarena dugu korapilotsuena zalantzarik gabe. Berori ez dagokio soilik euskarari. Bestelako hizkuntzetan ere, erregela finkorik ez egon eta dagoenekoetan guztiz edo idazle guztiek mantentzen ez dituztela esan genezake. "*
>
> (Odriozola eta Zabala, 1993)

Ingelesaren moduko hizkuntza normalizatu batentzako ere, puntuazioaren erabilera deskribatzen duten hainbat liburu (Ehrlich, 1992; Jarvie, 1992; Paxson, 1986) ez datoz bat zenbait kontutan (Bayraktar *et al.*, 1998).

Hizkuntza guztiek duten gaitza izanik ere, euskaraz are nabariagoak dira azaldutako arazoak. Euskararen estandarizazio- eta normalizazio-prozesu berantiarrak badu honetan zerikusirik. Hala, maila lexikoan ez bezala, sintaxi mailan, esaterako, badira oraindik finkatu beharrekoak. Beste modu batera esanda, ortografia zeharo araututa dagoen arren, gramatika-arauak ez daude guztiz estandarizatuta eta normalizatuta. Badira liburuak euskal gramatika deskribatzeko saioak egin dituztenak (Amundarain *et al.*, 2003; Alberdi eta Sarasola, 2001; Garzia, 1997; Laka, 1996; Zubiri eta Zubiri, 1995; Zubimendi eta Esnal, 1993; Goenaga, 1980), baina Euskaltzaindiari dagokio arauak ematea, eta gramatikari buruzko zenbait liburu argitaratu dituzten arren (Euskaltzaindia, 1999, 1994, 1993, 1990, 1987, 1985), ezin esan daiteke gramatika osoa arautua dutenik, oraindik orain.

Puntuazioa, berriz, aipatutako gramatika-liburuetatik gutxi batzuetan lantzen den gaia da. Zubiri eta Zubirik (1995) eta Zubimendi eta Esnalek (1993), adibidez, ortografiari eskainitako kapituluaren baitan lantzen dute puntuazioaren gaia, eta zenbait arau ematen dituzte, puntuazioaren erabilerari buruzkoak. (Azkarate *et al.*, 2006) eta (Aranguren *et al.*, 2006) lanetan ere, puntuazio-arau oinarrizkoenak ematen dituzte, modu argi eta laburrean.

Odriozola eta Zabala (1993), puntuazio-markei buruz, oro har, eta komaren erabilerari buruz, bereziki, mintzatzen dira. Euren iritziz, perpausaren eduki semantikoa alda dezaketen elementuak dira puntuazio-markak eta gertaera sintaktikoekiko harreman estuan daude:

> *"[. . . ] koma, puntua, puntu eta koma eta antzeko baliabide grafikoak, gertaera sintaktikoekiko harreman estuetan daude edo bestela esanez, sintagmen edo sintagma-taldeen arteko muga sintaktikoak hizkuntza idatzian adierazteko baliabide grafikoak ditugu"*
>
> (Odriozola eta Zabala, 1993)

Gainera, puntuazio-marken artean *erabilera-zailtasun handiena* azaltzen duena koma dela nabarmentzen dute.

Garziak (1997) semantikaren eskakizunekin lotzen ditu puntuazio-konbentzioen arrazoiak, aitorturik alderdi fonikoak ere baduela zerikusirik puntuazioarekin, eta sintaxi-arazoen sintomatzat hartzen ditu puntuazioaren erabilera okerrak:

> *"[...] Izan ere, intonazioaren proiekziotzat jo daiteke, lege onez, puntuazioa. Zer da, ordea, intonazioa, zentzuaren proiekzio bat baino? Hala, puntuazioaren eginkizuna zera litzateke: esaldien zentzua argitzen laguntzea, intonazioa nola edo hala islatuz. Kontutan har bedi intonazioaren gorabeherak infinitu samarrak direla, eta puntuazio-kodea, berriz, oso mugatu eta txiroa"*
>
> (Garzia, 1997)

Bestalde, euskarako puntuazio-kode autonomo, egoki eta finkatu samar baten falta eta premia sumatzen du Garziak (1997), eta premia horri erantzun bat ematen saiatzen da bere liburuan.

Nunberg-ek (1990) dioenez, puntuazioaren jatorria intonazioan bilatu behar bada ere, intonazioa eta puntuazioak aparteko bideak egin zituzten, eta, egun, sistema linguistiko oso bat da puntuazioa bera.

(Bayraktar *et al.*, 1998) lanean esaten denez, berriz, badira koma batzuk —*egiturazkoak* deitzen diete eurek—, muga sintaktikoak nabarmentzen dituztenak; koma horiek esaldiaren gramatika-egituraren mende daudela diote.

Puntuazioari baino ortotipografiari dagozkion arauak soilik ditu onartuta Euskaltzaindiak, gaur gaurkoz (Zubimendi, 2004): siglak, laburdurak, marratxoa eta enparauak nola idatzi behar diren azaltzen da liburu honetan.

Euskaltzaindiaren araurik gabe, beraz, aipatu berri ditugun erreferentzia hauek dira gure egitekorako erabili eta aztertu ditugunak, eta batez ere (Odriozola eta Zabala, 1993) eta (Garzia, 1997) lanak, hauek baitira puntuazioaren halako teorizazio bat egiten saiatu direnak. Bi lan hauetan azaltzen den teorizazioa desberdina izanik ere, ematen dituzten arauei dagokienez, bat datoz biak ala biak, kontu askotan. Esan gabe doa komaren erabilera dela bi lanen artean desberdintasun gehien sortzen duena, baina horretan ere, ñabardurak ñabardura, bat datoz *emaitzetan*, ez ordea emaitzotara iristeko darabiltzaten arrazoi-bideetan.

Ez da gure asmoa, halere, bi lanon azterketa eta konparazio zabala egitea tesi-lan honetan; ikuspuntu pragmatikoagoa da gurea. Izan ere —ez dezagun

ahantz—, puntuazio-zuzentzaile automatikoa egitea da gure helburua. Hala, puntuazioari dagozkion arau ahalik eta zehatzenak aplikatzea litzateke egokiena guretzat. Bi lanon arteko zalantzazko kasuetan, arestian aipatutako gainerako liburuak (Zubiri eta Zubiri, 1995; Zubimendi eta Esnal, 1993; Azkarate *et al.*, 2006) aztertu ditugu, erabaki zuzenena hartzeko asmoarekin. IV.4 atalean aztertuko dugu komari dagozkion arauak zertan geratu diren.

Hurrengo atalean, ostera, HPan puntuazioa jorratu duten lanak aztertuko ditugu.

## IV.2 Puntuazioaren garrantzia HPan

Nunberg-en (1990) monografikoaz geroztik, esan dugun legez, puntuazioari buruzko lan konputazionalak ugaldu egin ziren. Komunitate zientifikoa orduan ohartu zen puntuazioak —eta, zehatzago, komak— HPan izan zezakeen garrantziaz. Izan ere, komak esaldiaren sintaxian zeukan garrantzia azaleratu zuten Nunberg-ek (1990) eta beste hainbat lanek (Odriozola eta Zabala, 1993; Garzia, 1997; Bayraktar *et al.*, 1998). Honekin batera, HParen alorreko zenbait tresna (analizatzaile morfosintaktikoak edo erroreak detektatzeko sistemak, esaterako) hobetzeko edo berriak garatzeko bidea eman zuen komaren azterketak (Briscoe eta Carroll, 1995; Jones, 1996a). Beste modu batera esanda: argi geratu zen ondo puntuatutako testu batek —edo zehatzago esanda: komak zuzen jarrita dituen testu batek— esaldiaren analisi morfosintaktikoan lagundu zezakeela. Laurogeita hamargarren hamarkadaren hasieran, HPan puntuazioarekin zerikusia duten lanen ugalketaren erakusgarri nagusia da ACL[1] biltzarrean egindako lantegia: *workshop on punctuation in computational linguistics*[2]. Urte horietan egindako lanei buruzko laburpena dakar, bestalde, Say eta Akman-en (1996) lanak.

Laurogeita hamaseigarren urtetik aurrera egindako lanen artean, askotarikoak daude. Badira komaren rol sintaktiko desberdinak aztertzen dituztenak (Bayraktar *et al.*, 1998) edota rol sintaktiko egokia automatikoki esleitzen saiatzen direnak (Delden eta Gomez, 2002). Beste batzuek ahotsaren ezagutzarako lagungarri gisa erabiltzen dute puntuazioa (Beeferman *et al.*, 1998; Shieber eta Tao, 2003). Perpausak identifikatzeko komak —edota bestelako puntuazio-markak— erabiltzen dituztenak ere badira, analizatzailea hobetzeko azken helburuarekin batzuek (Li *et al.*, 2002a; Jin *et al.*, 2004),

[1] *Association for Computational Linguistics.*

[2] http://www.hcrc.ed.ac.uk/publications/wp-2.html

edota itzulpen automatikoko sistemak hobetzeko asmoarekin besteek (Zong *et al.*, 2002; Heyan eta Zhaoxiong, 2002; Tillmann eta Ney, 2003). Ingeleseko ikasleen testuetan preposizio-erroreak automatikoki detektatzen zituen tresna batek ere, komak zuzentzeko heuristiko batzuk sortu zituen sistemaren emaitzak hobetzeko (Chodorow *et al.*, 2007). Azkenik, erlazio semantikoak erauzteko darabiltzate (Srikumar *et al.*, 2008) lanean. Azken helburua esaldi batetik erator daitezkeen esaldi berriak identifikatzea bada ere, komek adierazten dituzten erlazio motak detektatzen saiatzen dira, lehendabizi. Horretarako, koma bakoitzak adieraz ditzakeen erlazio motak bost multzotan sailkatzen dituzte, eta ikasketa automatikoko algoritmo propio bat sortzen dute, koma bakoitzak adierazten duen erlazioa identifikatzeko, batetik, eta horretan oinarrituz jatorrizko esalditik erator daitezkeen esaldi berriak sortzeko, bestetik.

Koma-zuzentzaile edo berreskuratzaile automatikoak sortzeko, ordea, ez dira saiakera asko egin. Hardt-ek (2001) danierarako koma okerrak detektatzeko saioak egin zituen, *transformazioan oinarritutako ikasketa* Brill (1995) erabiliz. Horretarako, corpus batean, komak zoriz gehitu zituen lehendabizi. Zoriz gehitutako koma horiek koma oker gisa etiketatu zituen, eta beste guztiak koma zuzen gisa. Modu honetan, koma okerrak detektatzen saiatu zen Hardt (2001). % 91ko doitasuna lortu zuen zeregin horretan, hain zuzen ere; estaldura, berriz, % 77koa. Sistema hau gai izango da, neurri batean, gaizki jarritako komak detektatzeko, baina inolaz ez jarri gabe egonik jarri beharko liratekeenak. Gainera, erroreak automatikoki sortzean, errore oso artifizialak sortzen dira maiz, eta halako sistemek gero arazoak izan ohi dituzte errore errealekin. Baldwin eta Joseph-ek (2009), berriz, puntuazioa (ez soilik komak) eta kasua berreskuratzeko saioak egin dituzte duela gutxi, ikasketa automatikoko teknikak erabiliz. Zehazki, SVM algoritmoan oinarritutako sailkatzaileen arkitektura bat darabilte, eta $F_1 = \%\ 62$ erdiesten dute. Bestalde, txekierako puntuazioa detektatzeko helburuarekin, sakoneko analizatzaile sintaktiko bat baliatu dute berriki (Jakubicek eta Horak, 2010) lanean. $F_1 = \%\ 83{,}5$ lortzen dute ataza horretan, baina eskuz etiketatutako corpus bat erabiliz.

Ahotsaren ezagutzarako sistemek ere puntuazioa berreskuratu behar izaten dute. Shieber eta Tao-k (2003) egindakoa da lanik esanguratsuena, arlo honetan. Osagaien informazioa baliatzen dute, komak non jarri erabakitzeko. Zehatzago esanda, osagaien mugak erabiltzen dituzte komen kokalekua asmatzeko; hau da, token bakoitza zenbat osagairen hasiera eta bukaera den kontatzen dute. Izan ere, euren iritziz, token bat geroz eta osagai gehiagoren

muga izan, token horren inguruan koma bat izateko orduan eta probabilitate handiagoa dago. (Beeferman *et al.*, 1998) lanean azaldutako hiru egoerako *Markov-en eredu ezkutuak (HMM)* erabiltzen dituzte. Euren ereduan, aipatutako informazio linguistikoa txertatzen dute (osagaiei dagozkien mugena, hain zuzen), eta hobekuntzak lortzen dituzte informazio gehigarri horrekin. III. kapituluan deskribatutako kateen eta perpausen identifikatzaileek ematen diguten informazioa, ondorioz, badirudi informazio esanguratsua izango dela euskarako koma-zuzentzailea hobetzeko.

Arazo batekin egiten dute topo proba horiek egiterakoan: osagaien informazioa nondik atera. Izan ere, hasieran, *Penn Treebank* corpusa baliatzen dute, bai ikasketarako, bai probarako. Honela, % 68,4 lortzen dute komak berreskuratzeko atazan, $F_1$ neurrian. Halere, datu hauek ez dira guztiz errealistak. Izan ere, eskuz etiketatutako corpus bat erabiltzen da. Metodoa errealista eta beraz erabilgarria izan dadin, eskuz analizatutako corpusaren ordez, analizatzaile automatiko bat erabiltzea beharrezkoa da; Collins-en (1997) analizatzaile estatistikoa erabili zuten helburu horrekin. Analizatzaile estatistikoa erabiltzean ere, zenbait arazori egin behar izan zioten aurre. Izan ere, test-corpusean komarik ez zutela izango jakinik, ez zuten garbi zerekin entrenatu behar zuten. Aukera bat komadun corpusa analizatu eta analizatzaileak emandako datuekin entrenatzea zen. Horren arriskua zen test-corpusak ez zeukala komarik eta beraz portaera desberdina izan zezakeela. Ikasketa-corpusari komak kendutakoan analizatzea zen beste aukera, test-corpusarekin egingo zena imitatuz. Honen desabantaila zera zen: analizatzailearen portaera okerragoa izango zela, eta ondorioz baita analizatzailearekin lortutako informazioa linguistikoa ere (guri ere gauza bera gertatu zaigu, IV.6.4.2 atalean ikus daitekeen moduan). Test-corpusaren ahalik eta antz handiena duen ikasketa-corpus bat erabiltzeak emaitzak hobetu zitzakeela pentsatu zuten, eta hala zela frogatu zen, bi aukerekin proba egin eta gero. Hala eta guztiz ere, beste proba bat egin zuten: berreskuratu nahi zituzten koma horiek test-corpusean jarrita zeudela suposatuz baliatu zuten Collins-en (1997) analizatzailea, ikasketa-corpusean ere komak mantenduz kasu honetan. Modu honetan lortutako emaitzak komarik gabeko analisiarekin lortutakoak baino hobeak izan ziren, baina eskuzko analisiarekin lortutakoak baino txarragoak, jakina. Zera ondorioztatu zuten honekin: geroz eta informazio sintaktiko hobea izan, orduan eta emaitza hobeak lortuko liratekeela komaren berreskurapenean.

Bestalde, euren ereduari informazio linguistiko berria gehituz (token bakoitzaren kategoria), koma-berreskuratzailearen emaitzak are gehiago hobetu

ziren (eskuzko analisiarekin: $F_1 = \%\ 74{,}8$; analizatzaile automatikoarekin: $F_1 = \%\ 70{,}1$). Gainera, prosodiaren informazioa baliatuz gero, emaitzak hobetuko lituzketela diote, hainbat lanek erakutsi duten moduan (Christensen *et al.*, 2001; Kim eta Woodland, 2001).

Guri dagokigunean, ostera, arazoa bestelakoa da, idatzizko testuak baitira gure abiapuntu. Hala, testu horietatik koma guztiak kentzea aukera bat izan daitekeen arren, erabiltzaileak jarritako komak erabilgarriak izan daitezke, kasu batzuetan, analisi hobea lortzeko. Dena dela, sistemaren portaera homogeneoa izatea nahi badugu, ikasketa- eta test-corpusean komak kendu beharko genituzke ezer baino lehen. Aurrerago eztabaidatuko dugun gai bat izango da hau.

(Shieber eta Tao, 2003) lanaren beste ekarpen garrantzitsua ebaluazioari buruzkoa da. Izan ere, token mailako ebaluazioak emateaz gain, esaldi mailako ebaluazioak ere ematen ditu. Neurri horretan, esaldi guztiko koma guztiak ondo badaude, esaldia ondo puntuatua izan dela kontsideratuko da; komaren bat gaizki badago, ordea, esaldia gaizki puntuatutzat hartuko da. Komaren ebaluaziorako, badu honek zentzua, esaldi beraren barruan koma bat ondo jarri baina hurrengoa gaizki jartzeak, esaterako, esaldi guztiaren zentzua alda baitezake. Tesi-lan honetan token mailako ebaluazioa egin bada ere, esaldi mailako azken ebaluazio bat ere egin dugu (ikus IV.6.5 atala).

Bestalde, arestian aipatutako lanetatik, komaren rol sintaktikoa aztertzen saiatzen direnak ere interesatzen zaizkigu. Izan ere, etorkizunean, komak jartzen *ikasten laguntzen* duen modulu batekin uztartu nahi genuke komen zuzentzailea. Horretarako, ordea, ezinbestekoa da, koma non jarri behar den jakiteaz aparte, ezagutzea zein den koma leku horretan jartzeko arrazoia. Hori dela eta, koma bakoitzari dagokion klasea zein den detektatu beharko genuke automatikoki; alegia, koma bakoitzaren rol sintaktikoa identifikatu beharko litzateke automatikoki. (Bayraktar *et al.*, 1998) eta (Delden eta Gomez, 2002) dira eginkizun hori aztertu eta landu dutenetako batzuk.

(Bayraktar *et al.*, 1998) lanean, komaren funtzio edo erabilera desberdinen azterketa egiten da. Hasteko, komaren erabilera desberdinen araberako sailkapen bat egiten dute hainbat liburutan oinarrituta (Ehrlich, 1992; Jarvie, 1992; Paxson, 1986): zazpi erabilera mota sailkatu zituzten (jo IV.4.1 atalera, sailkapena ikusteko). Gero, koma bat edo gehiagoko egituren patroi sintaktikoak eraiki zituzten, *Prolog* programazio-lengoaia erabiliz (1.978 patroi, guztira). *Penn Treebank* (Marcus *et al.*, 1993) corpuseko zati bat hartuta (*Wall Street Journal* egunkariko artikuluak, hain zuzen ere), patroi horiek erabiliz corpuseko komadun adibideak sailkatzea zen helburua. Patroi

sintaktiko garrantzitsuenak soilik erabiltzea erabaki zuten, corpuseko komen % 80 sailkatua geratzea bermatuz betiere. Horretarako, nahikoak izan ziren patroi-sintaktiko guztietatik % 11 (211 patroi, 1.978 patroietatik). Beste modu batean esanda: nahikoak izan ziren patroi guztien % 11, corpus osoko komen % 80 sailkatzeko. Bukatzeko, 211 patroi horiek aurrez aipatutako zazpi erabilera mota horietako bati esleitu zizkioten. Komadun adibide bakoitza patroi bati lotuta zegoenez, patroi bakoitza komaren zazpi erabilera mota horietako bati esleituz, komadun adibide bakoitzari zein erabilera mota zegokion ondorioztatu zuten; alegia, komadun adibide guztien % 80 zazpi erabilera mota horietan sailkatu zituzten. Atera zituzten ondorioen artean, hiru iruditzen zaizkigu garrantzitsuenak (gogoratu ingeleserako ateratako ondorioak direla hauek):

1. Koma *egonkorrenak* aposizioenak[3], mintzagaienak[4] eta izenen ondoko modifikatzaile ez-murriztaileenak dira.

2. Erabilera mota aldakorrenak aposizio ez diren tartekienak[5] eta esaldibukaerako elementuak bereizten dituztenak dira (koma hauek hain estandarizatuta ez daudelako, agian).

3. Geroz eta estaldura handiagoko analizatzaileak izan, orduan eta puntuazio-zuzentzaile finagoak sortzea posible izango da. Gainera, gramatika-zuzentzaileekin integratu eta, testu bakoitzaren estiloaren arabera, puntuazioari dagozkion aholku zehatzagoak edo orokorragoak eman ahal izango dira.

[3]Aposizioa: bi osagai elkarren segidan jarriz egindako eraikuntza, bigarrenak aurrekoa azaltzen edo zehazten duelarik. Azalpenezko sintagma bat izan ohi da, eta bi motakoak izan daitezke: murriztaileak eta ez-murriztaileak. Aposizio murriztailea da izen sintagmaren barnean gertatzen dena, izen bati beste izen bat –edo gehiago– lotzen zaionean. Hortaz, izen sintagma bakarra osatzen du hitz-segida guztiak ("*Unibertsitateko Errektoreorde Edurne Mendiluzek...*"). Aposizio ez-murriztaileak, berriz, izen sintagma osoak beste hitz-segida batekin nolabait parekatuz egiten dira ("*Matematikako irakasle berria,* iazkoaren ordez etorri dena*, oso atsegina da.*").

[4]Mintzagaia: informazio zaharra edo jadanik ezaguna biltzen duten elementuek osatzen dute, edo solasaren gaia finkatzen dutenek. Gure ikuspuntutik, esaldia (edo perpausa) abiatzeko elementu egoki gisa ikusiko dugu mintzagaia; esaldiari (edo perpausari) sarrera egiten diona.

[5]Tartekia: lokailu bat edo aposizio ez-murriztaile bat izan daiteke, baina baita esaldiak edo perpausak adierazi nahi duen mezu nagusiari tartekatzen zaion azalpen gehigarria ere ("*Heriotza da,* Saramagoren iritziz*, Jainkoaren asmatzailea.*").

Delden eta Gomez-en (2002) lanean, berriz, komen rol sintaktikoak aztertzeaz aparte, testuetan koma bakoitzari bere rol sintaktikoa automatikoki esleitzeko ahalegina egiten da. Bi fasetan banaturiko prozesu bat deskribatzen dute horretarako: automata finituak (Abney, 1995; Roche, 1999) eta *transformazioan oinarritutako ikasketa*[6] (Brill, 1995; Daelemans, 1999) teknikan oinarritutako algoritmo *jale*ak baliatzen dituzte, hurrenez hurren, fase bakoitzean. Lehenengo fasean, koma bakoitzari, har ditzakeen etiketa guztiak —hots, rol sintaktiko posible guztiak— esleitzen zaizkio, automata finituen bidez. Automata finitu bakoitza independentea denez, koma bakoitzak etiketa bat baino gehiago izan ahalko du. "*John likes apples, oranges, and bananas.*" esaldia jartzen dute adibidetzat. Esaldi horretan "*apples*" hitzaren ondoren datorren komari bi rol sintaktiko esleituko zaizkio: enumerazioarena eta aposizioarena. Izan ere, informazio semantikorik gabe, zaila da "*oranges*" hitza aposizio bat ez dela ebaztea. Hortaz, behar-beharrezkoa zaie bigarren fase bat, koma bakoitzarentzat, aurreko fasetik dituen rol sintaktikoen artetik, zuzenak bakarrik aukeratzeko. Zuzenak, gainera, bat baino gehiago izan daitezkeela arrazoitzen dute. "*In the Fall of 1992, a great year for sports, my favorite team won the World Series.*" adibidean, esate baterako, lehendabiziko komak preposizio-sintagma baten bukaera adierazten du, baina, aldi berean, aposizio baten hasiera markatzen du. Etiketa zuzen hauek, esan bezala, *transformazioan oinarritutako ikasketa* teknikan oinarritutako algoritmo *jale* baten bidez aukeratzen dira, lehenengo fasean jarritako etiketa guztien artetik. Horretarako, 250.000 tokeneko corpus bat darabilte —15.000 koma inguru dituena; token guztien % 6—. Corpusean, kategoriak Brill-en (1993) etiketatzailea erabiliz etiketatu ziren, eta eskuz markatu ziren gerora koma bakoitzaren rol sintaktikoak. Test-corpusean ebaluatu zuten sistema, eta komen % 90 inguru ondo etiketatu zituen makinak. Hala eta guztiz ere, hobekuntzak egin daitezkeela aipatzen zuten, i) darabilten kategoria-etiketatzailea hobetuz, ii) ikasketa-corpusa handituz, edo iii)automata finituei erregela semantikoak gehituz —WordNet (Fellbaum, 1998) gisako baliabideren bat erabiliz, kasu—.

Bestalde, puntuazioak analizatzaile sintaktiko automatiko batean daukan eragina neurtzeko, interesgarria da Briscoe eta Carroll-ek (1995) egindako azterketa. Susanne (Sampson, 1995) corpuseko 2.500 esaldi inguru analizatu zituzten. Esaldi bakoitzeko 225 analisi lortu zituzten, batez bestean. Esaldi barruko puntuazio-marka guztiak kendu zituzten gero, eta berriz analizatu

[6] *Transformation based learning.*

zuten testua. Esaldi guztien % 8a analisirik gabe geratu zen. Analisia eman zuen kasuetan, bestalde, 310 analisi eman zituen batez beste, esaldiko; alegia, lehen baino % 38 analisi gehiago. Are gehiago, 100 esaldiren eskuzko azterketa egin ostean, horietatik herenek, euren analisien artean, ez zuten analisi-zuzena. Ondorioz, desanbiguatzailearen emaitzek ere behera egin zuten.

Hurrengo ataletan, puntuazio- eta estilo-*akatsak* detektatzeko egindako lana aurkeztuko dugu. Hala, IV.3 atalean, ezagutza linguistikorik erabili gabe detekta daitezkeen *akatsak* harrapatzeko egindako lanaren berri emango dugu; IV.4 atalean, berriz, euskaraz egiten den komaren erabileraren formalizazioa deskribatuko dugu; IV.5 atalean, hizkuntzaren ezagutzan oinarritutako teknikekin komak zuzentzeko egindako saioak aurkeztuko ditugu; eta, azkenik, ikasketa automatikoa baliatuz komak zuzentzeko egindako ahaleginak deskribatuko ditugu IV.6 atalean.

## IV.3 Estilo-zuzentzailearen lehen hurbilpena: ezagutza linguistikorik behar ez duten erregelak

Badira puntuazioari edo estiloari dagozkion *akats* tipiko batzuk[7], ezagutza linguistikorik gabe detekta daitezkeenak; hau da, hizkuntzalaritza konputazionaleko tresnak —analizatzaileak, batez ere— erabili gabe detekta daitezkeen puntuazio- edo estilo-akatsak badira. Atal honetan, era horretako akatsak detektatzen egindako lana deskribatuko dugu.

Hauek dira landu ditugunak:

1. Zuriunea zuzen jartzea (Aranguren *et al.*, 2006):

    - puntuazio-marka hauen aurretik ez da zuriunerik jarriko: puntua, koma, bi puntuak, puntu eta koma, hiru puntuak, harridura-marka, galdera-marka, marratxoa eta ixteko parentesia.
    - puntuazio-marka hauen ondotik zuriunea utzi behar da nahitaez (paragrafo bukaera ez den kasuan): puntua, koma, bi puntuak, puntu eta koma, hiru puntuak, harridura-marka eta galdera-marka. Salbuespenak:

[7] Arautik desbideratzen direla esan beharko genuke, eta ez *akatsak* direla. Informazio gehiago II.3.1 atalean.

  - komarekin: zenbaki hamartarrak (54,34)
  - puntuarekin: zenbakizko milakoak (17.834)
  - bi puntuekin: orduak (8:30)
  - harridura- eta galdera-markekin: tartekiak ("*Baina, kontuz!, ni ez nago-eta zuekin ados.*")

- Hasierako kakotxen eta parentesien ondoren, ez da zuriunerik jarriko; eta aurretik, beti jarriko da zuriunea.
- Bukaerako kakotxen eta parentesien aurretik, ez da zuriunerik jarriko; eta ondoren, beti jarriko da zuriunea.

2. Letra larriaren eta xehearen erabilera (Zubimendi eta Esnal, 1993):

   - Puntuaren ondoren, letra larriz hasi behar da.
   - Bi puntuen ondoren lerro berean segitzen denean —salbuespenak salbuespen (Zubimendi, 2004)—, letra xehez hasi behar da.
   - Bi puntuen ondoren paragrafo berria hasten denean —salbuespenak salbuespen (Zubimendi, 2004)—, letra larriz hasi behar da.
   - Puntu eta komaren ondoren, oro har, letra xehez hasi behar da.
   - Harridura- eta galdera-marken ondoren eta hiru puntuen ondoren, letra larriz hasi behar da, oro har (Aranguren *et al.*, 2006).

3. Hasierako parentesiak eta komatxoak bukatu egin behar dira; bukaerako parentesiek eta komatxoek ere hasierakoak behar dituzte.

4. Euskaraz —gaztelaniaz ez bezala—, harridura-markek eta galdera-markek ez dute hasierakorik onartzen (Zubiri eta Zubiri, 1995).

5. Apostrofoa, oro har, ez dago baimenduta euskaraz; bi kasu hauetan, ordea, maiz erabiltzen da:

   - *agur t'erdi* esapidean (Azkarate *et al.*, 2006).
   - *lau t'erdiko* pilota-txapelketari buruz jardutean (Aranguren *et al.*, 2006).

6. Siglak punturik gabe idatziko dira, letra larriz idatzitako esaldietan tartekatuta datozenean izan ezik (Zubimendi, 2004).

7. Ehunekoaren sinboloa eta zenbakiaren artean, zuriunea utzi behar da (Zubimendi, 2004): *Kontinentearen % 5 izotza da.*

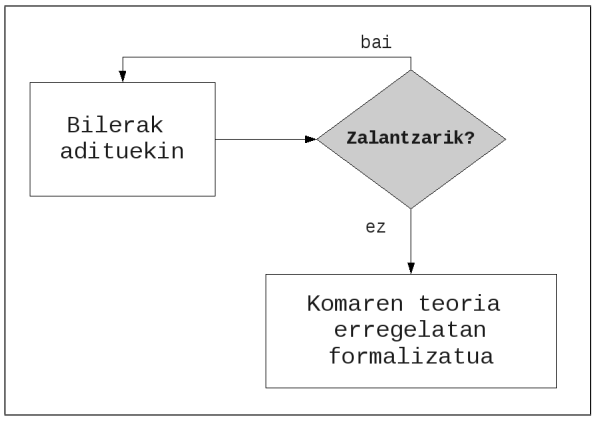
(Otegi, 2003) lanean jasota dago arau hauei dagokien programazio-lana. Aipatu dugun moduan, lan hau hizkuntzalaritza konputazionaleko tresnarik erabili gabe egin zen, baina tokenizatzaile baten erabilerak erraztu egingo lituzke gauzak. Izan ere, lehenengo puntuko salbuespenak tratatzeko, adibidez, ez genuke zenbaki artean dauden puntuazio-markak ekiditen ibili beharko, tokenizatzaileak zenbaki edo ordu gisa ulertuko bailituzke tartean ikurren bat izanagatik ere. Gainera, zerrenda honetako *akats* batzuk testu-editoreek eurek detektatzen dituzte, gaur egun. Dena dela, testu-editorearen mende ez egotearren sortu genituen guk ere algoritmo hauek.

Hurrengo ataletan, gehien interesatzen zaigun puntuazio-marka aztertuko dugu: koma. Izan ere, arestian ikusi dugun moduan, hizkuntzalaritza konputazionaleko zenbait atazatarako informazio garrantzitsua eman dezake komak.

## IV.4 Komaren erabilera: azterketa linguistikoa

Duela urte batzuk, euskarako gramatika-zuzentzailerako lanak bideratzen ari ginela, Juan Garziaren[8] bisita izan genuen IXA taldean. Bere ustea zen une hartan geneuzkan tresnekin komaren zuzenketa automatikoa egingarria izan zitekeela. Hala, komari buruzko bere teorizazioa (Garzia, 1997) geurera ekartzeko saioak antolatu genituen; beste modu batean esanda, bere teorizazioa nolabait formalizatzeko bilerak egin genituen: ikuspuntu informatiko batetik laburtu eta eskematizatu nahi genuen Garziaren komari buruzko teoria. Horretarako, metodologia zikliko bat diseinatu eta bost pertsonako lan-talde bat osatu genuen (IXA taldeko hiru informatikari eta bi hizkuntzalarirekin). Hala, bilera bakoitzean, adituaren azalpenak entzuten genituen, eta bileraren ostean, azalpen horiek informatikaren ikuspuntutik formalizatzen saiatzen ginen: komaren arauak zehaztuko zituen erregelen multzoa osatzea zen azken helburua. Garbi geratzen ez zitzaizkigun kontuak edo zalantzazko kasuak hurrengo bileran argitzeko uzten genituen. IV.1 irudian ikus daiteke erabilitako metodologia. Modu horretan, sei bilera egin genituen, Garziaren teorizazioa hainbat erregelatan formalizatzea lortu genuen arte.

[8]Hizkuntzalaria da Garzia, sintaxian eta puntuazioan aditua.




Irudia IV.1: Komen teoria formalizatzeko erabilitako metodologia.

Horren ostean, formalismo hori puntuazioaren arloko beste zenbait adituri erakutsi genien (Joxe Ramon Etxebarria[9], Igone Zabala[10] eta Juan Carlos Odriozola[11]). Arestian aipatu dugun eran, ñabardurak ñabardura, bat zetozen hauek ere Garziarekin formalizatutako komaren arauekin. Etxebarriak, esaterako, malgutasun apur bat gehiago baimentzen zuen, agian, zenbait erregelatan; Zabalak eta Odriozolak, berriz, ez zuten lokailuak koma bikoitzaz (bai aurretik, bai ondoren) markatzearen beharra ikusten.

Azkenean, Garziarekin bat egitea erabaki genuen lokailuaren kasuan, honako arrazoi hauengatik:

- Kontsultatu genituen gainerako liburuak (Zubimendi eta Esnal, 1993; Zubiri eta Zubiri, 1995; Aranguren *et al.*, 2006; Azkarate *et al.*, 2006) bat datoz Garziaren irizpidearekin, lokailuen kasuan.

- Garziaren teorizazioa zabalduta dago zenbait erakundetan (*Euskal Herriko Unibertsitatea*n, adibidez) eta zenbait komunikabidetan (*Berria* egunkarian, kasu).

- Garziaren teorizaziotik abiatuta sortu genuen gure formalizazioa, eta arau batengatik bere teorizazioari izkina egiteak ez dauka zentzurik.

Hauek dira, beraz, Garziaren (1997) teorizazioa oinarritzat hartuz bildutako arauak. Kontuan hartu behar da, aipatu dugun legez, Odriozola eta

[9]UEUko euskara-zuzentzailea izan zen Joxe Ramon Etxeberria.
[10]Hizkuntzalaritzan doktorea, eta sintaxian eta puntuazioan aditua.
[11]Hizkuntzalaritzan doktorea, eta sintaxian eta puntuazioan aditua.

Zabalaren (1993) irizpideak ere bat datozela hauekin ia kasu guztietan, eta baita aztertu ditugun gainerako puntuazioari buruzko idatziak ere (Zubimendi eta Esnal, 1993; Zubiri eta Zubiri, 1995; Azkarate *et al.*, 2006; Aranguren *et al.*, 2006). Hauexek erabili ditugu, hain zuzen, gure sailkapena osatzeko.

Batetik, koma noiz jarri behar den adierazten diguten arauak ditugu; eta, bestetik, koma noiz *ez* den jarri behar adierazten digutenak.

**Koma noiz jarri behar den:**

1. Esaldi koordinatuak lotzeko: bi esaldi juntagailu batez koordinatzen baditugu, koma jarriko dugu juntagailuaren aurretik.

   **Adibidea IV.4.1**
   (a) *Euria ari zuen**, eta** etxean geratu nintzen.*
   (b) *Denboraldi baterako joango da**, baina** baliteke ez itzultzea.*

2. Enumerazioetan, osagaien artean koma jarri behar da; azken biak juntagailuarekin lot daitezke.

   **Adibidea IV.4.2**
   (a) *Etxean gordeta zeuzkan **diskoak, liburuak eta bideoak**.*
   (b) *Afaltzera denak datoz: **Ane, Miren, Jon eta Mikel**.*

3. Deikiak komaz markatzen dira.

   **Adibidea IV.4.3**
   (a) *Ongi etorri, **Jon**.*
   (b) *Kaixo, **Ane**.*

4. Estilo zuzeneko *esaldiak* mugatzeko ere, koma erabili ohi da.

   **Adibidea IV.4.4**
   (a) ***“Agirretxe da faboritoa”,** aitortu du Arregik.*
   (b) *Alderdi politikoen legeari **“zorakeria”** iritzi zion lehendakariak.*

5. Aposizio ez-murriztaileak koma artean idazten dira; aposizio murriztaileak, aldiz, ez.

   **Adibidea IV.4.5**
   (a) *Matematikako irakasle berria, **iazkoaren ordez etorri dena**, oso atsegina da.*
   (b) *Jon Pereira **lehendakaria** izan zen kazetarien aurrean azaltzen azkena.*

6. Tartekiak, lokailuak eta diskurtso-antolatzaileak koma artean idazten dira (a,b,c); esaldiaren hasieran edo bukaeran daudenean, koma bat eta beste puntuazio-marka *gogor* baten artean (d). Batzuetan, koma hauek ken daitezke; puntuazio arina erabil daiteke, esaldiko puntuazio orokorrak eraginda, edo, besterik gabe, anbiguotasun arriskurik ez dagoelako. "e" adibidean, esaterako, mendeko perpausaren hasiera adierazten duen koma da "*batez ere*" esapidearen aurretik dagoena; hala, koma hau markatzearen garrantziak esapideari dagozkion koma bikoitzak kentzera garamatza.

**Adibidea IV.4.6**

(a) *Barrez erantzun zien,* ***pozik****, galdera guztiei.*

(b) *Jokalariak,* ***pozez zoratu beharrean****, kopa hartu zuen esku artean.*

(c) *Jonek,* ***beraz****, etxean geratzea erabaki zuen.*

(d) *Uste baino gutxiago iraun zuten,* ***ordea****.*

(e) *Baskoniak talde bezala indar handiagoa duela frogatu du hainbatetan,* ***batez ere*** *defentsan ondo ezartzen denean.*

7. Aditz nagusiaren aurreko mintzagaiaren ondoren koma jartzea gomendatzen da (a), mintzagaia subjektua ez denean betiere (b). Aitzitik, behar-beharrezkoa da batzuetan, anbiguotasuna ekiditeko (c). Mintzagaia edo galdegaia[12] (bietako bat, gutxienez) mendeko perpausa bada, mintzagaiaren ondoren koma jarri behar da derrigor (d,e).

**Adibidea IV.4.7**

(a) ***Azkenean****, zurekin joango naiz.*

(b) ***Aita*** *atzo iritsi zen.*

(c) ***Batzuetan****, irabazteko gogor jokatzea beharrezkoa da.*

(d) ***Euria ari zuenez****, etxean geratzea erabaki genuen.*

(e) ***Gaur iritsiko zela esan zuen arren****, atzo iritsi zen azkenean.*

8. Aurretik markatzailerik ez duten perpaus zirkunstantzialak-eta, perpaus nagusiaren atzetik badatoz, komaz bereizi behar dira printzipioz (a); anbiguotasuna egon daitekeenean, derrigor (b,c). Aitzitik, perpaus nagusiaren atzetik datorrena galdegaia baldin bada, ez da komaz bereiziko (d).

**Adibidea IV.4.8**

(a) *Ez nuen eztabaidatu nahi izan,* ***ados ez nengoen arren****.*

[12]Galdegaia: oro har, aditzaren aurretik doan sintagma. Esaldian azpimarratu nahi dena adierazten du.

(b) *Jan zuen,* ***andereñoa konturatu ere gabe****.*

(c) *Jan zuen andereñoa,* ***konturatu ere gabe****.*

(d) *Ez dakigu* ***etorriko den****.*

9. Aditza isildua dagoenean, koma baliatzen dugu elipsia markatzeko.

**Adibidea IV.4.9**

(a) *Atzo euria egin zuen; gaur, ez.*

### Koma noiz *ez* den jarri behar:

1. Sintagma eta aditz-kate barruan, oro har, ez da komarik izango.

**Adibidea IV.4.10**

(a) ***Zarauzko hondartzaren luzerak*** *harritu gintuen.*

(b) *Hemen* ***geratuko gara****.*

2. Mendeko perpaus baten barruan, oro har, ezin da koma bakarra jarri; koma pareak bai, ordea.

**Adibidea IV.4.11**

(a) ***Azkenaldian gai beraren inguruan eztabaidatzen ari garela*** *ezin dugu ahaztu.*

(b) *Iluntzera arte geratu behar izan genuen,* ***guk,*** **jakina*****, lehenbailehen alde egin nahi genuen arren****.*

3. Galdegaia eta aditza ez dira komaz bereiziko, oro har (a,b). Galdegaia perpaus nagusiaren ondoren datorrenean, eta horrek anbiguotasuna ekar dezakeenean, koma jarri beharko da (c,d,e), edo esaldiaren egitura aldatu, bestela.

**Adibidea IV.4.12**

(a) ***Atzo*** *iritsi ginen.*

(b) ***Gurekin geratuko zela*** *esan zigun.*

(c) *Andereñoa haserretu zaio,* ***klasean jende guztiaren aurrean izozkia jan duelako****.*

(d) *Andereñoa haserretu zaio klasean,* ***jende guztiaren aurrean izozkia jan duelako****.*

(e) *Andereñoa haserretu zaio klasean, jende guztiaren aurrean,* ***izozkia jan duelako****.*

4. Subjektuaren eta aditzaren artean, oro har, ezin da koma bakarra jarri (a,b); tartean tartekiren bat edo lokailuren bat denean, koma parea jarri ahal izango da (c).

   **Adibidea IV.4.13**
   (a) *Gurera etorriko da* ***Imanol****.*
   (b) ***Imanol*** *gurera etorriko da.*
   (c) ***Imanol****, azkenean, gurera etorriko da.*

### IV.4.1 Komaren erabileraren konparaketa: euskara eta ingelesa

Xehetasun handietan sartu gabe, eta betiere ikuspuntu pragmatiko batetik, komaren erabileran euskara eta ingelesa konparatu nahi izan ditugu. Batez ere, jakin nahi izan dugu euskarako komen zuzentzailea garatzeko erabilitako irizpideak ingelesaren gisako hizkuntza batean balia ote daitezkeen.

Santos-en (1998) arabera, puntuazioa —neurri handi batean— hizkuntzaren mende dago. Ondorio horretara iritsi zen ingelesaren, portugesaren eta norvegieraren puntuazioa aztertu ondoren. Batik bat, estilo zuzena eta bi puntuak aztertzen baditu ere, komaren erabilera desberdinak ere aipatzen ditu:

> - *A comma is obligatory before an integrating clause in German, forbidden in Portuguese or English.*
> - *A comma is obligatory after a restrictive relative clause in Norwegian, forbidden in Portuguese and English.*
> - *A comma obligatorily surrounds rethorical sentential adverbs in Portuguese, but it is forbidden in Norwegian.*
> - *A comma is required after a sentence initial adverb or prepositional phrase in English, but forbidden in Norwegian and optional in Portuguese.*
>
> (Santos, 1998)

Santosek (1998) erakusten duen moduan hizkuntza batetik bestera desberdintasunak badiren arren, aztertu ditugun bi hizkuntzen komaren erabilerari dagokionez, irizpide berdintsuak dituztela ikusi ahal izan dugu. Konparatzen diren hizkuntzen araberakoa izango da hau ziur aski, baina euskara eta ingelesaren gisako hizkuntza desberdinetan antzekotasun handiak topatu

baditugu, jatorri bera duten hizkuntzen artean are handiagoak izango direla esatea ez zaigu zentzugabekeria iruditzen. Esan gabe doa baieztapen honek azterketa sakonagoa beharko lukeela.

IV.4 atalean azaldutako arauak ingelesenekin konparatuko ditugu orain, euskararen eta ingelesaren arteko komen erabilpena zehatz-mehatz aztertzeko. (Bayraktar *et al.*, 1998) lanean aipatzen duten sailkapena hartuko dugu oinarritzat. Sailkapen hau, eurek diotenez, Ehrlich-en (1992) liburuan oinarritua dago, batez ere, baina (Jarvie, 1992; Paxson, 1986) liburuak ere hartu dituzte aintzat. Gainera, egiturari ez dagozkion erabilerak (zenbakietako komak, esaterako) ez dira kontuan hartu sailkapen honetan. Euren sailkapeneko arauak banan-banan azaldu, eta euskarakoekin bat datozen ala ez aztertuko dugu segidan:

1. Enumerazioak: hiru elementuz edo gehiagoz osatutako zerrendetako osagaiak banatzeko koma erabiltzen da. Osagai horiek hitzak, sintagmak edo mota sintaktiko bereko perpausak izan daitezke. Azken elementua, eskuarki, juntagailu batez banatuko da (*eta* edo *edo*, adibidez), eta batzuetan *koma+juntagailua* egituraz (batez ere, gaizki-ulertuak ekiditeko eta esaldiak banatzeko):

   **Adibidea IV.4.14**

   (a) *Elsewhere, share prices closed higher in* ***Amsterdam, Brussels, Milan and Paris****.*

   (b) *You may order anything you want at my dinner as long as you order* ***sausage and eggs, ham and eggs, or bacon and eggs****.*

2. Mintzagaiak: esaldi-hasieran sarbide gisa agertzen diren sintagmak edo perpausak komaz mugatuta joan daitezke, irakurlea nahasteko aukerarik baldin badago. Sarbide gisako modifikatzaileak (hala nola adjektiboak, adberbioak edo partizipioak), eskuarki hitz bakar batez osatutakoak, komaz jarraituak dira normalean.

   **Adibidea IV.4.15**

   (a) ***Under the new features****, participants will be able to transfer money from the new funds to other investment funds [. . . ].*

   (b) ***Although the action removes one obstacle in the way of an overall settlement to the case****, it also means that Mr. Hunt could be stripped of virtually all of his assets if the Tax Court rules against him in a 1982 case heard earlier this year in Washington, D.C.*

   (c) ***Clearly****, the judge has had his share of accomplishments.*

3. Esaldi-bukaerako elementuak: esaldi-hasierako sarbide gisako elementuak legez, esaldi-bukaerako elementu osagarriak komaz bereizita joango dira, desanbiguatzeko beharra baldin badago. Elementu hori sintagma bat, mendeko perpaus bat edo ideia oso bat adierazten duen perpaus bat izan daiteke.

**Adibidea IV.4.16**

(a) *A bomb exploded at a leftist union hall in San Salvador,* ***killing at least eight people and injuring about 30 others, including two Americans****, authorities said.*

(b) *A face-to-face meeting with Mr. Gorbachev should damp such criticism,* ***though it will hardly eliminate it****.*

(c) *She ran faster,* ***her breath coming in deep gasps****.*

4. Izenen ondoko modifikatzaile ez-murriztaileak: izenen ondoren doazen modifikatzaileak —sintagmak edo perpausak izan daitezkeenak—, ez-murriztaileak badira, koma artean doaz.

**Adibidea IV.4.17**

(a) *The man* ***at the left*** *is taller.*

(b) *He was the only student* ***who answered all the questions in the exam****.*

(c) *A Western Union spokesman,* ***citing adverse developments in the market for high-yield junk bonds****, declined to say what alternatives are under consideration.*

(d) *At one point, almost all of the shares in the 20-stock Major Market Index,* ***which mimics the industrial average****, were sharply higher.*

5. Aposizio ez-murriztaileak koma artean doaz.

**Adibidea IV.4.18**

(a) *Alexander* ***the Great*** *was a powerful emperor.*

(b) *The new company, called Stardent Computer Inc., also said it named John William Poduska,* ***former chairman and chief executive of Stellar****, to the posts of president and chief executive.*

6. Tartekiek ere koma artean joan behar dute. Hitzak, sintagmak edo perpausak izan daitezke tarteki.

**Adibidea IV.4.19**

(a) *The Brookings and Urban Institute authors caution,* ***however****, that most nursing home stays are of comparatively short duration, and reaching the Medicaid level is more likely with an unusually long stay or repeated stays.*

(b) *The new bacteria recipients of the genes began producing pertussis toxin which,* ***because of the mutant virulence gene****, was no longer toxic.*

(c) *Rebuilding that team,* ***Mr. Lee predicted****, will take another 10 years.*

7. Estilo zuzeneko esaldiak: aipu zuzenak, beste norbaiten hitzak zehatzmehatz errepikatzen dituztenak, komaz bereizita doaz.

**Adibidea IV.4.20**

(a) *"****The absurdity of the official rate should seem obvious to everyone****", the afternoon newspaper Izvestia wrote in a brief commentary on the devaluation.*

Koma jarri behar ez den kasuak ez dira aipatzen sailkapen honetan, baina zenbait liburutan arakatu ondoren, jarraian azalduko ditugun ondorioetara iritsi gara.

Ingeleseko esaldiek nahiko ordena konkretua daukate[13]: SVO[14] deiturikoa, alegia. Ordena konkretu hori errespetatzen da esaldi gehientsuenetan. Bada, subjektua eta aditzaren artean, eta aditzaren eta objektuaren artean, ezin da komarik jarri (Huddleston eta Pullum, 2002), salbuespenak salbuespen:

**Adibidea IV.4.21**

(a) *He went on foot.*

(b) *You'll have to train every day to have any chance of winnning.*

(c) *The right of the people to keep and bear arms shall not be infringed.*

Bi arau hauekin pareka ditzakegu euskarako "*subjektua eta aditzaren artean koma bakarra ez*" araua eta "*galdegaia eta aditzaren artean koma bakarra ez*" araua. Komak ezin jar daitezkeen euskarako beste bi arauei dagokienez, berriz, zera esan daiteke: sintagmen eta aditz-kateen barruan ingelesez ere ez dela komarik onartzen, eta mendeko perpausen barruan ere ez, jakina den legez (tartekien koma bikoitzak izan ezik).

Beraz, esan dezakegu koma ez jartzeko euskarako arauak ingelesak ere badituela. IV.1 taulan ikus daiteke, berriz, koma jarri beharreko arauen artean, bi hizkuntzen arteko parekidetasuna zenbaterainokoa den.

Euskarako hirugarren eta hamargarren arauetarako izan ezik, beste guztietarako badugu pareko ingeleseko arauren bat. Bi salbuespenak, gainera, (Bayraktar *et al.*, 1998) lanean datorren sailkapenaren hutsuneak direla esan dezakegu, topatu baitugu euskarako bi arau horiek ingelesean ere betetzen direla dioen libururik (Truss, 2003).

[13]Inguruko gainerako erdarek ere bai, oro har; baina ez gara kontu horietan sartuko.

[14]SVO = Subject+Verb+Object; alegia, subjektua+aditza+objektua.

| **Euskarakoak** | **Ingelesekoak** |
|---|---|
| Esaldi koordinatuak lotzean, juntagailuaren aurretik koma (1) | Enumerazio batzuetan koma+juntagailua egitura (1) |
| Enumerazioetan, osagaien artean koma (2) | Enumerazioetan, osagaien artean koma (1) |
| Deikiak komaz markatu (3) | - |
| Estilo zuzeneko esaldiak komaz bereizi (4) | Estilo zuzeneko esaldiak komaz bereizi (7) |
| Aposizio ez-murriztaileak koma artean (5) | Aposizio ez-murriztaileak koma artean (5) |
| Tartekiak koma artean (6) | Tartekiak koma artean (6) |
| Lokailuak eta diskurtso-antolatzaileak koma artean (7) | Izenen ondoko modifikatzaile ez-murriztaileak (4) eta tartekiak (6) koma artean |
| Aditz nagusiaren aurreko mintzagaiaren ondoren, koma (8) | Esaldiaren hasieran, sarbide gisa doazen sintagmak edo perpausak komaz mugatuta (2) |
| Aditz nagusiaren ondorengo perpaus zirkunstantzialen aurretik, koma (9) | Esaldi bukaerako elementu osagarriak komaz bereizi (3) |
| Aditza isildua dagoenean, komaz markatu (10) | - |

Taula IV.1: Koma jartzeko arauen konparazioa: euskara vs ingelesa.

Laburbilduz, esan dezakegu euskarako eta ingeleseko komaren arauak bat datozela, oro har. Bi hizkuntza hauen artean —komari dagokionez—, beraz, ez da betetzen Santos-en (1998) adierazpena, non puntuazioa, neurri handi batean, hizkuntzaren mende zegoela esaten baitzen. Gauza jakina da, halere, arauak berak izanagatik, esaldien ordenamendu desberdina dela-eta, baterako sortutako erregela hipotetikoek ez luketela besterako, bere horretan, balioko. Dena dela, komaren arauak inguruko hizkuntzetan antzekoak direla frogatu nahi genuen, eta ingelesarenak eta euskararenak hala direla ziurta dezakegu gutxienez.

## IV.5 Komen zuzenketa hizkuntzaren ezagutzan oinarritutako tekniken bidez

Komaren erabilera-arauak aztertu eta formalizatu ondoren (IV.4), koma-zuzentzaile bat lortzeko saioak egin genituen, hizkuntzaren ezagutzan oinarritutako teknikak erabiliz hasieran. Hala, CG formalismoa baliatu genuen, komen arauei zegozkien erregelak idazteko.

Gisa honetako erregelak, ordea, testuinguru txikiko arauak formalizatzeko dira egokiak, Oronozen (2009) iritziz. Horregatik, komen arauen artetik CG formalismoarekin inplementatzeko egokiak zirenak soilik aukeratu genituen. Izan ere, badira arau batzuk testuinguru zabala behar dutenak; alegia, arau batzuk automatizatzeko, koma behar duen hitzaren inguruko hitz asko aztertu beharko lirateke. Gainera, euskararen hitzen ordenaren malgutasunak are gehiago zailtzen du eginkizun hau. Bestalde, alarma faltsuak ekiditera jo genuen, eta ziurtasun minimo batekin detekta genitzakeen arauak soilik inplementatu genituen (doitasun handiagoa bilatu genuen, beraz, estalduraren kaltetan).

Guztira, hemeretzi erregela idatzi genituen, IV.4 atalean azaldutako zenbait arau nolabait formalizatzen saiatzeko. CG formalismoaren bidez modu errazean inplementa zitezkeenak egin genituen: hamabi orduko lana besterik ez zen izan. IV.2 irudian ikus daiteke erregela baten adibidea, eta A eranskinean ikus daitezke guztiak.

IV.2 irudiko adibidean dugun erregelak honako hau adierazten du:

Jarri "&OKER_KOMA_FALTA_1_1" etiketa hitz bati —edozein delarik ere bere kategoria—, zeinaren hurrengoa "baina" juntagailua den. Beste modu batean esanda, hitz bat aurkitzen badugu eta bere ondoren datorren

```
MAP (&OKER_KOMA_FALTA_1_1)
                TARGET EDOZEIN_KAT
                IF (1 BAINA + JNT);
```

Irudia IV.2: Komen arauak formalizatzen saiatzeko egindako CG erregelen adibide bat.

hitza "baina" juntagailua bada, tartean koma bat falta da.

Erregela hauek ebaluatzeko, *Euskaldunon Egunkaria*ko garapen-corpusa erabili zen (ikasketa automatikoko probak egiteko erabili zen garapen-corpus bera, hain zuzen). Erregelek jarritako etiketak, berez testuan jarritako komekin konparatuta atera ziren emaitzak; bitan banatuta aurkezten dira, beraz: 0 klasea (ondoren komarik ez duten tokenak) eta 1 klasea (ondoren koma duten tokenak). Bi klase hauen gaineko ohiko neurriak ematen dira: doitasuna, estaldura eta $F_1$ neurria. IV.6.1 atalean, corpusari eta ebaluazioari buruzko xehetasun gehiago irakur daitezke.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Hizkuntza-ezagutzan oinarrituta** | 93,1 | 96,7 | 94,9 | 56,9 | 27,2 | 36,8 |

Taula IV.2: Komen identifikazioaren emaitzak, CG formalismoa baliatuz.

IV.2 taulan paratu ditugu erregela hauen bidez lortutako emaitzak. 0 klaseko emaitzak onak dira, espero moduan. 1 klaseko emaitzak onegiak ez diren arren (36,8ko $F_1$ neurria lortu genuen), aipatzekoa da arau guztientzat ez ditugula erregelak egin, eta, beraz, estalduraren emaitzak logikoak direla. Bestalde, doitasunari erreparatzen badiogu (% 56,9), emaitzak ez dira hain txarrak; erregela bidez jartzen diren kometatik, erdia baino gehiago ondo leudeke. Agian, emaitzek, orokorrean, ez dirudite oso onak, baina ikasketa automatikoarekin konbinatzeko onak izan daitezkeela uste dugu; alegia, erregela hauek ikasketa automatikoaren emaitzak hobetzen lagunduko digutela iruditzen zaigu.

## IV.6 Komen zuzenketa ikasketa automatikoan oinarrituta

Jarraian deskribatuko ditugun probetan, corpusetan oinarritutako hurbilpenak erabili genituen koma-zuzentzailea garatzeko; zehatzago esanda, ikasketa automatikoko teknikak baliatu genituen. Kasu honetan, hitz bakoitzaren ondoren koma jarri behar den (1) edo ez (0) da ikasi beharreko kontzeptua. Instantziak edo adibideak, berriz, eskuragarri ditugun corpusetatik lortu genituen (IV.6.1.1 atalean azalduko dugu zein diren erabilitako corpusak). Ikasketa-prozesurako, gainera, hainbat ezaugarri linguistiko baliatu genituen; hau da, mota desberdineko informazio linguistikoa erabili genuen ikasketa-prozesuan atributu edo ezaugarri gisa.

Informazio linguistikoa lortzeko, *Eustagger* erabili genuen, IXA taldearen analizatzaile/desanbiguatzaile morfosintaktikoa. Honek komak ere erabiltzen ditu, ahalik eta analisi onenak lortu ahal izateko; komak darabiltzan analizatzailea erabiltzea zalantzazkoa da, ordea, gerora analizatzaile honek ematen duen informazio linguistikoa koma-zuzentzailea sortzeko erabili behar bada. IV.6.4.2 atalean azalduko dugu nola egin genion aurre arazo honi. Dena dela, hasierako esperimentuak ohiko analizatzailearekin egin genituen (komak darabiltzanarekin, alegia).

IV.6.1 atalean, esperimentuen prestaketaz arituko gara. Erabilitako corpus mota desberdinak aurkeztuko ditugu. Horrekin batera, ebaluazioa nola egin genuen zehaztu eta *oinarrizko neurri*ak nola kalkulatu genituen kontatuko dugu. Azkenik, ikasteko hasieran baliatu genituen ezaugarri linguistikoak aurkeztuko ditugu.

IV.6.2 atalean, berriz, egindako esperimentuen berri emango dugu; besteak beste, zein leiho den egokiena erabakitzeko egindako probak deskribatuko ditugu, edota ikasketa-algoritmo egokiena aukeratzeko saioak, edo ezaugarri linguistiko berriak gehituz egindako ahaleginak. Komak zuzentzeko corpus motak daukan garrantzia ere azalduko dugu, eta euskararako egindako saioak ingeleserako errepikatuko ditugu, ingelesaren gisako hizkuntza normalizatu batean emaitza hobeak lortzen ote diren ikusteko. Aurreko kapituluan deskribatutako kate- eta perpaus-identifikatzaileek emandako informazioa koma-zuzentzailea hobetzen saiatzeko nola baliatu dugun kontatuko dugu azkenik.

IV.6.3 atalean, hizkuntzaren ezagutzan oinarritutako teknikak corpusetan oinarrituekin uztartuta lortzen diren emaitzak aztertuko ditugu. IV.6.5 atalean, egindako ebaluazio kualitatiboa azalduko dugu, eta bukatzeko, komarik

gabeko analizatzailea erabiltzeko beharra aztertuko dugu, aipatu dugun moduan, IV.6.4.2 atalean.

## IV.6.1 Esperimentuen prestaketa

Atal honetan, corpusaren aukeraketa, ebaluazio-moduaren azalpena, *oinarrizko neurri*ak zein izan ziren, baliatu genituen ikasketa-algoritmoak eta ikasketan —hasiera batean— erabilitako ezaugarri linguistikoak azalduko ditugu, besteak beste.

### IV.6.1.1 Corpusaren aukeraketa

Ezaugarri desberdinetako lau corpus mota erabili genituen probetarako, corpus desberdinen eragina aztertzearren. Gogoratu beharra dago komak ikasteko erabili genituela corpusok, eta horretarako corpus hauetan komak zuzen jarrita daudela suposatu genuela. Horregatik, corpus desberdinekin egin nahi izan genuen proba. Izan ere, corpusaren arabera komak hobeto edo okerrago —edo, zehatzago esanda, modu homogeneoagoan edo libreagoan— jarrita egon daitezkeela iruditzen zitzaigun. Esate baterako, autore bakar batek idatzitako corpusean komak modu homogeneoago batean jarrita izango zirela pentsatzen genuen. Arrazoi hauengatik egin dugu proba, egile eta jatorri desberdinetako corpusekin.

Dena dela, proba gehienak egiteko, *Euskaldunon Egunkaria*[15] berripapereko testuez osatutako corpusa baliatu genuen. 135.000 hitzez osatutako corpusa erabili genuen proba gehienetan; hala eta guztiz ere, corpus handiagoarekin ere egin genituen saio batzuk (ikus IV.6.2.8 atala). Corpus honek, handia izateaz aparte, beste dohain garrantzitsu bat dauka: bertako komak arestian aipatutako irizpideei jarraiki jarri ziren, teorian; izan ere, egunkariaren estilo-liburuan azaltzen diren komari buruzko arauak bat datoz gureekin, Garziaren (1997) jarraibideak segitzen baitituzte. Corpus honetako testu multzo bat gainbegiratu eta orokorrean hala zela ziurtatu genuen.

Beste corpus batzuk ere erabili genituen, hala ere. Batetik, *autore bakar batek idatzitako filosofia-testuak* baliatu genituen. 25.000 tokeneko corpus hau autore bakar batek idatzia denez, koma modu homogeneoan erabilia izango zela pentsatu genuen.

[15] *Euskaldunon Egunkaria* eta *Berria* (*www.berria.info*) egunkariekin IXA taldeak daukan elkarlana dela-eta lortutako corpusa.

Bestetik, *autore bakar baten literatura-testuak* ere erabili genituen. Hau ere, 25.000 tokeneko corpusa da, eta komari dagokionez, homogeneotasuna espero liteke[16].

Azkenik, *Zientzia eta Teknika* corpusa baliatu genuen, Elhuyar Fundazioan sortua[17]. Corpus etiketatua da, bai testuaren egiturari eta formatuari dagokionez, baita linguistikoki ere. Etiketatze linguistikoa egiteko, *Eustagger* erabili zen. Testuko hitz bakoitzaren lema eta kategoria/azpikategoria etiketatu ziren. Corpusaren lehen bertsioan, 8,5 milioi hitz daude, eta horietatik 1,9 milioi hitz eskuz berrikusi, desanbiguatu eta zuzendu ziren, lema eta kategoria mailan. Testu teknikoz osatutako corpus hau baliatu genuen, literaturari eta filosofiari *a priori* aitortzen ez zaizkion ezaugarriak izango zituelakoan: argitasuna eta zehaztasuna.

#### IV.6.1.2 Ebaluazioa

Aurreko kapituluan deskribatutako neurri estandar berberak erabili ziren: doitasuna, estaldura eta $F_1$ neurria, garapen-corpusaren gainean, lehendabizi, eta test-corpusaren gainean, azkenik, kalkulatzen direnak. *Euskaldunon Egunkaria* corpusaren 135.000 tokenak zoriz banatu ziren ikasketa eta ebaluazioa egiteko. Horietatik, % 75 ikasketa-corpus gisa eta *cross-validation* probak egiteko (ikasketa-corpusa hamar zatitan banatuz); gainerako % 25a, berriz, garapen- eta test-corpus gisa. *Euskaldunon Egunkaria*ko corpus hau erabili zen ia proba guztietan. Beste corpusekin egindako ebaluazioak *cross-validation* teknika baliatuz egin ziren (corpusa 10 zatitan banatuz). IV.3 taulan ikus daitezke corpusaren banaketaren datu zehatzak. Tesi-txosten honetan, kontrakorik esaten ez den bitartean, corpus hau erabili da.

Ebaluazio-modu honetan, aipatu dugun moduan, corpusean jarritako komak —eta horiek bakarrik— ematen dira ontzat. Honek baditu bere mugak.

[16]Bada erabili ezin izan dugun corpus erraldoi bat ere, *Ereduzko Prosa Gaur* izenekoa (http://www.ehu.es/euskara-orria/euskara/ereduzkoa/), eta 25,1 milioi hitzez osatutakoa. Hauetatik, 13,1 miloi hitz literatur-obrenak dira; gainerako hamabi milioiak, *Berria* egunkarikoak (hamar milioi) eta *Herria* astekarikoak (bi milioi). Corpus hau Euskal Herriko Unibertsitateko Euskara Institutuak kudeatzen duen arren, IXA taldeak ezin izan du ustiatu gaurdaino, baimen-kontuak direla medio (*Berria* egunkaria izan ezik, lehen aipatu dugun moduan, Berriaren eskutik). Ikusi bai, baina ukitu ezin den altxorra da, beraz, literatur-obrei dagokien corpusa.

[17]Zientzia eta euskara uztartzeko helburuarekin jaiotako irabazi-asmorik gabeko erakundea da Elhuyar fundazioa (*www.elhuyar.org*). HPa da lantzen duen alorretako bat, eta IXA taldearekin elkarlanean dabil horretan, beste batzuekin batera.

| | Token kopurua |
|---|---|
| **Ikasketa-corpusa** | 101.250 |
| **Garapen-corpusa** | 28.500 |
| **Test-corpusa** | 5.250 |
| **Corpus osoa** | 135.000 |

Taula IV.3: Komak ikasteko eta ebaluatzeko erabilitako *Euskaldunon Egunkaria*ko corpusaren banaketa.

Izan ere, ikasten ari garena zenbateraino zuzena den ez dakigu. Gainera, esaldi batean komak jartzeko konbinazio zuzen posible bat baino gehiago egin daiteke, eta guk zuzentzat emandakoa —testuen egileek jarritakoa— aukera bat baino ez da. HPko beste zenbait alorretan ere gertatzen den arazo hau aintzat hartuta (Mayor *et al.*, 2009), ebaluazio kualitatibo bat egitea erabaki genuen (ikus IV.6.5 atala), non ontzat ematen baitira aukera bat baino gehiago.

Azken testa egiteko corpus gisa, eta eskuzko etiketatzea —hizkuntzalariek eginikoa— eta ebaluazio kualitatiboa egiteko erabili genuen 5.500 hitzeko test-corpus txikiagoa.

#### IV.6.1.3 Ikasi beharreko kontzeptua

Emaitza-atributua edo ikasi beharreko kontzeptua bitarra da kasu honetan; alegia, 0 edo 1 balioak soilik har ditzake. 0 balioak esan nahi du adibide edo instantzia horren ondoren ez datorrela komarik; 1 balioak, aldiz, koma datorrela adibide horren ostean. Beste hitz batzuetan esanda, garapen- edo test-corpuseko token bakoitzari zein balio dagokion erabaki behar du sailkatzaileak; alegia, token bakoitza 0 klasekoa den edo 1 klasekoa den erabaki beharko du. Hemendik aurrera, beraz, 0 eta 1 klaseaz arituko gara, emaitza-atributuaren balio posibleez ari garenean.

Horretarako, corpusa prestatu behar izan genuen. Komak, hain zuzen, ez ziren adibide edo instantzia gisa gehitu, aurreko tokenaren emaitza-atributu gisa baizik. Hau da, token baten ondoren koma bat baldin badator, emaitza-atributuan 1 balioa izango du token horri dagokion adibideak; bestela, 0 balioa.

Garrantzi handien eman diogun neurria doitasuna izan da. Izan ere, Guinovart-ek (1996b) dioen moduan, doitasunaren eta estalduraren arteko oreka moduko bat lortu beharko litzatekeen arren, horretarako aukerarik ezean,

doitasunak izan beharko luke lehentasuna. Honela arrazoitzen du esandakoa Guinovart-ek (1996b): gramatika-zuzentzaileen eta antzeko tresnen erabiltzaileek nahiago dituzte okerrak zuzentzat hartzen dituzten akats informatikoak, zuzenak okertzat hartzen dituztenak baino; hots, alarma faltsuak dira erabiltzaileek gehien gorrotatzen dituzten errore informatikoak.

Komari dagokionez, gainera, emaitza-atributuko 1 klasearen doitasunari eman behar izan genion garrantzia, batez ere: sailkatzaileak (makinak) koma behar dela dioenean, benetan koma izatea, alegia. Sailkatzaileak koma ez dela dioenean, koma ez izatea ere garrantzitsua da (emaitza-atributuko 0 klasearen doitasuna), baina, aurrerago ikusiko dugun moduan, corpusean mota honetako adibide anitz daudenez, errazagoa da 0 klaseko adibideak iragartzea. Hortaz, gure esperimentuetan, 1 klaseko emaitzei erreparatu genien, batik bat; 1 klaseko doitasunari, batez ere, baina estaldura ere kontuan hartuta; izan ere, zertarako balio du esaldi batean koma bat ondo jartzeak, esaldiko gainerako komak ez badira ondo jartzen? IV.6.5 atalean ikusiko dugun moduan, galdera honek ez dauka erantzun errazik, koma bakoitzaren arabera hartu beharreko erabakia baita sarritan. Beraz, oreka moduko bat bilatu behar izan genuen, doitasunari garrantzia gehixeago emanez.

#### IV.6.1.4 *Oinarrizko neurriak*

Hiru modu baliatu genituen oinarrizko neurriak kalkulatzeko.

1. Testuko komen proportzioa erabiliz lortutakoa:
   Ikasketa-corpusean zeuden komen proportzioa kalkulatu genuen (koma dira token guztien % 8), eta zoriz jarri genituen komak, gero, garapen-corpusean, proportzio hori erabiliz (*baseline_%8* deitu dioguna IV.4 taulan).

2. Ikasketa-corpusean komaz jarraituak maizen agertzen diren $x$ hitzak kontuan hartuz ($x = 100$, $x = 200$ edo $x = 300$ balioak erabili genituen):

   - *baseline_100*: ikasketa-corpusean komaz jarraituak maizen agertzen diren 100 hitzak hartu, eta garapen-corpuseko hitz horien agerpen guztiei koma jarrita.
   - *baseline_200*: ikasketa-corpusean komaz jarraituak maizen agertzen diren 200 hitzak hartu, eta garapen-corpuseko hitz horien agerpen guztiei koma jarrita.

- *baseline_300*: ikasketa-corpusean komaz jarraituak maizen agertzen diren 300 hitzak hartu, eta garapen-corpuseko hitz horien agerpen guztiei koma jarrita.

3. Ikasketa-corpuseko *komadun* hitzak kontuan hartuz:
   Ikasketa-corpusean komaz jarraitutako hitz guztiak zerrendatu genituen. Garapen-corpusean, zerrenda horretako hitzen agerpen bakoitzari jarri genion koma (*baseline_ikasketakoak* deitu dioguna IV.4 taulan).

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| ***baseline_%8*** | 92,7 | 92,4 | **92,6** | 7,6 | 7,9 | 7,8 |
| ***baseline_100*** | 94,1 | 80,2 | 86,6 | **12,5** | 35,8 | 18,5 |
| ***baseline_200*** | 94,4 | 75,6 | 84,0 | 12,1 | 42,7 | **18,9** |
| ***baseline_300*** | 94,5 | 72,4 | 82,0 | 11,2 | 46,5 | 18,7 |
| ***baseline_ikasketakoak*** | 94,6 | 55,6 | 70,0 | 9,6 | **59,6** | 16,5 |

Taula IV.4: *Baseline-neurriak* edo *oinarrizko neurriak*.

IV.4 taulan ikus daitezke *baseline* bakoitzarekin lortutako emaitzak. *Oinarrizko neurriok* aztertzen baditugu, bi ondorio nagusi atera ditzakegu:

1. Emaitza onak lortzen dira 0 klaserako; hau da, komak noiz ez diren jarri behar ondo ikasten du sistemak. Jarri behar diren komak jartzen, ordea, ez du batere ondo asmatzen (ikus 1 klaseko emaitzak). Jarri behar diren komak ondo jartzen asmatzea, beraz, ez da lan erraza izango.

2. Oso alde handia dago bi klaseen (0,1) arteko emaitzen artean. Izan ere, corpusa ez da *orekatua* zentzu horretan: askoz adibide gehiago ditu 0 klasekoak, 1 klasekoak baino. Gogoan izan aztertu dugun corpusean token guztien % 8a komak direla. Alegia, komaz jarraituriko token askoz gutxiago dago ikasketa-corpusean, komaz jarraitu gabekoak baino. Desoreka horrek arazoak ekarriko ditu 1 klasean emaitza onak lortzeko.

#### IV.6.1.5 Ikasketa-algoritmoak

Hiru ikasketa-algoritmo hauen WEKA inplementazioak erabili genituen: *Naive Bayes*, erabaki-zuhaitzak (C4.5 algoritmoa) eta *Support Vector Machine* (SVM). *Naive Bayes* erabili genuen algoritmo sinpleenetako bat delako;

erabaki-zuhaitzak, berriz, morfosintaxiari dagozkion atazetan emaitza onak lortu izan dituelako eta lortzen den ezagutza interpretagarria delako; SVM erabili genuen ($C$=1), azkenik, gaur egun gehientsuen erabiltzen den ikasketa-algoritmoa delako eta HPko atazatan emaitza onak erdietsi ohi dituelako (II.1 atalean, ikasketa-algoritmo hauen azalpen zabalagoak paratu ditugu).

#### IV.6.1.6 Atributuak edo ezaugarri linguistikoak

Adibide edo instantzia bakoitzerako erabil daitezkeen atributuak edo ezaugarri linguistikoak nahi adina eta nahi bezain konplexuak izan daitezke. Bestalde, erabaki beharrekoa da, modu berean, corpusa adibideetan banatzeko zein izango den aukeratutako unitatea: tokena, sintagma, perpausa... Alegia, erabaki behar da zerk osatuko duen adibide edo instantzia bakoitza: token, sintagma, perpaus edo esaldi bakoitzak.

Gure kasuan, unitatea tokena izan da. Alegia, token bakoitzerako erabaki behar izan dugu ea ondoren koma datorren ala ez. Dena dela, ebaluazioa esaldika egitea ere interesgarria delakoan gaude, arestian ikusi dugun legez. Azken ebaluazio kualitatiboa egiterakoan, kontuan hartu dugu hau, IV.6.5 atalean.

Adibide bakoitzerako —token bakoitzerako, beraz, gure kasuan— baliagarriak iruditu zitzaizkigun ezaugarri linguistikoak aukeratu genituen, komari buruz egindako teorizazioa aintzat hartuta (ikus IV.4). Hala, hasiera batean, 33 atributu hauek kontuan hartzea erabaki genuen; *Eustagger*-ek emandako datuak dira horietako asko (ezaugarri morfosintaktikoak: lema, kategoria, deklinabide-kasua...); besteak *Ixati* zatitzaileak emandakoak (aditz-kate baten hasiera edo bukaera den, sintagma baten hasiera edo bukaera den...); beste batzuk CG erregelez osatutako perpaus-mugatzaileak emandakoak dira (esaldiaren hasiera edo bukaera den edo perpaus-muga bat den), eta badira batzuk daukagun informaziorekin kalkula daitezkeenak, kontaketa sinple batzuen bidez gehienetan (atributu *kalkulatu* deitu diegu hauei). Jarraian dauzkagu zerrendatuta, hasieran erabilitako ezaugarri guztiak (parentesi artean, ezaugarri bakoitzari eman genion atributu-izena, IV.3 irudian ikus daitekeen moduan):

- hitza (*word*)
- lema (*lemma*)
- kategoria (*kat*): izena, adjektiboa, aditza...

- azpikategoria (*azpkat*): izenetan, leku-izen bereziak edo pertsona-izen bereziak, esate baterako; determinatzaileetan, determinatzaile erakusleak, adibidez; adjektiboetan, izenondoak eta izenlagunak...
- deklinabide-kasua (*decls*): absolutiboa, ergatiboa...
- mendeko perpaus mota (*mendekoak*): erlatibokoa, moduzkoa, kausazkoa...
- aditz-kate baten hasiera den ala ez (*aditz_kate_has*): aditz-kate baten hasiera den ala ez adierazten da (atributu bitarra).
- aditz-kate baten bukaera den ala ez (*aditz_kate_buk*): aditz-kate baten bukaera den ala ez adierazten da (atributu bitarra).
- sintagma baten hasiera den ala ez (*sint_has*): sintagma baten hasiera den ala ez adierazten da (atributu bitarra).
- sintagma baten bukaera den ala ez (*sint_buk*): sintagma baten bukaera den ala ez adierazten da (atributu bitarra).
- entitate-hasiera ote den (*enti_has*): entitate baten hasiera den ala ez adierazten da (atributu bitarra).
- entitate-bukaera ote den (*enti_buk*): entitate baten bukaera den ala ez adierazten da (atributu bitarra).
- postposizio-hasiera ote den (*post_has*): postposizio baten hasiera den ala ez adierazten da (atributu bitarra).
- postposizio-bukaera ote den (*post_buk*): postposizio baten bukaera den ala ez adierazten da (atributu bitarra).
- hitz anitzeko unitate baten parte ote den (*haul*): hitz anitzeko unitate baten parte den ala ez adierazten da (atributu bitarra).
- tarteki baten parte ote den (*tarteki*): marra luzeez edo parentesiez mugatutako tarteki baten parte den ala ez adierazten da (atributu bitarra).
- puntua (*puntu*) den ala ez (atributu bitarra).

- hiru puntuak (*hiru_puntu*) puntuazio-marka den ala ez (atributu bitarra).
- bi puntuak (*bi_puntu*) puntuazio-marka den ala ez (atributu bitarra).
- puntu eta koma (*puntu_eta_koma*) den ala ez (atributu bitarra).
- harridura-marka (*harridura_ikurra*) den ala ez (atributu bitarra).
- galdera-marka (*galdera_ikurra*) den ala ez (atributu bitarra).
- esaldi hasiera den ( *esaldi_muga_has*) ala ez (atributu bitarra).
- esaldi bukaera den ( *esaldi_muga_buk*) ala ez (atributu bitarra).
- perpaus-muga den ( *perpaus_muga*) ala ez (atributu bitarra).
- uneko tokenetik esaldiaren hasierara dagoen aditz-kate kopurua (*zenbat_AK_ezk* atributu kalkulatua).
- uneko tokenetik esaldiaren bukaerara dagoen aditz-kate kopurua (*zenbat_AK_esk* atributu kalkulatua).
- uneko tokenetik esaldiaren hasierara dagoen sintagma kopurua (*zenbat_IS_ezk* atributu kalkulatua).
- uneko tokenetik esaldiaren bukaerara dagoen sintagma kopurua (*zenbat_IS_esk* atributu kalkulatua).
- uneko tokenetik esaldiaren hasierara dagoen mendeko perpausen marka kopurua (*zenbat_mendeko_EM_ezk* atributu kalkulatua).
- uneko tokenetik esaldiaren bukaerara dagoen mendeko perpausen marka kopurua (*zenbat_mendeko_EM_esk* atributu kalkulatua).
- uneko tokenetik esaldiaren hasierara dagoen token kopurua (*dist_EM_ezk* atributu kalkulatua).
- uneko tokenetik esaldiaren bukaerara dagoen token kopurua (*dist_EM_esk* atributu kalkulatua).

IV.3 irudian ikus daiteke arff formatuko fitxategi baten itxura, eta aipatu berri ditugun atributu guztiak. *@Relation* hitzaz hasitako lerroak ikasi beharreko kontzeptuari izena ematen dio. *@Atributte* hitzaz hasitako lerro bakoitza atributu bat izango da, ikasketarako ezaugarri bat; azken atributua (*koma*) izango da ikasi beharreko kontzeptua. *@Data* hitzaren ondorengo lerro bakoitza ikasketarako edo testerako datu bat —adibide bat— izango da, zeinak balio bat izango baitu goian azaldutako atributu bakoitzarentzat.

#### IV.6.1.7 Leihoa

Token bakoitzaren atributuen artean, komeni izaten da inguruko tokenen informazioa ere kontuan hartzea. Token bakoitzerako inguruko zenbat token hartzen diren kontuan adierazten du leihoak, hain zuzen ere.

Hasierako gure leihoa (-5,+5) izan zen; alegia, token bakoitzerako, token horren aurreko bost tokenen eta ondorengo bosten informazioa hartzen genuen kontuan. Beste modu batean esanda, token bakoitzak 33 atributu dauzkanez, inguruko hamar tokenak kontuan hartuz gero, 363 atributu (11 * 33) izango genituzke token bakoitzarentzat (uneko tokenarena berarena eta bere inguruko hamarrena). Alabaina, arestian aipatutako atributuen artean, azken zortziak *kalkulatuak* dira. Kalkulatuak deritzegu, token bakoitzaren inguruko osagaien zenbait ezaugarri kontatuz kalkulatzen direlako. Esaterako, *Zenbat_AK_ezk* atributu kalkulatuak uneko tokenetik esaldiaren hasierara dagoen aditz-kate kopurua gordetzen du. Uneko tokenetik esaldiaren hasierara dauden aditz-kate kopurua jakinda, erraz jakin liteke inguruko tokenek (leihoak mugatutakoek) atributu horrentzat luketen informazioa; informazio hori leihoko token guztiei jartzea gauza bera errepikatzea litzateke, alegia. Beraz, kalkulatutako ezaugarri hauen informazioa ez da gehitzen leihoko atributu gisa. Hala, (-5,+5) leihoarekin, 283 ezaugarri izango genituzke: uneko tokenaren 33 ezaugarriak gehi ezkerreko eta eskuineko 10 tokenen 25 ezaugarriak (10 * 25 = 250). Hauei, ikasi beharreko emaitza-atributua gehitu beharko genieke.

#### IV.6.1.8 Jatorrizko komen eragina saihesten

Komak ikasteko saioetan, beharrezkotzat jotako ezaugarri linguistikoak (eskuragarri geneuzkanak) erabili genituen, IV.6.1.6 atalean ikusi berri dugun moduan: hala nola, token bakoitzaren kategoria eta azpikategoria, edo kateei buruzko informazioa. Aipatu dugun moduan, informazio linguistiko hau

```
@RELATION komak_eu
@ATTRIBUTE word REAL
@ATTRIBUTE lemma REAL
@ATTRIBUTE kat
{,-,ADB,ADI,ADJ,ADL,ADT,AMM,ASP,ATZ,AUR,BST,DEK,DET,ELI,ERL,
FLX,GRA,HAOS,IOR,ITJ,IZE,LAB,LOT,MAR,PRT,SIG,SNB,IZB,LIB}
@ATTRIBUTE azpkat
{,-,DZH,BAN,DZG,ORO,ORD,ERKARR,ERKIND,NOLGAL,NOLARR,BIH,
ELK,PERARR,PERIND,IZGMGB,IZGGAL,ARR,IZB,LIB,ZKI,SIN,ADK,ADP
,FAK,GAL,JNT,MEN,LOK,IZO,IZL,ADOARR,ADOGAL,ALGARR,ALGGAR
,DATA,ALGGAL,ZNB,ARRIZE}
@ATTRIBUTE decls
{,-,ABL,ABS,ABU,ABZ,ALA,BNK,DAT,DES,DESK,ERG,EZZ,GEL,GELGE
L,GEN,INE,INS,MOT,PAR,PRO,SOZ,KK,IDENT}
@ATTRIBUTE mendekoak
{,-,BALD,DENB,ESPL,HELB,KAUS,KONT,MOD,MOT,ERLT,KONP,ZHG,
MOS,MOD/DENB,EMEN,ONDO,AURK,HAUT}
@ATTRIBUTE aditz_kate_has {0,1}
@ATTRIBUTE aditz_kate_buk {0,1}
@ATTRIBUTE sint_has {0,1}
@ATTRIBUTE sint_buk {0,1}
@ATTRIBUTE enti_has {0,1}
@ATTRIBUTE enti_buk {0,1}
@ATTRIBUTE postp_has {0,1}
@ATTRIBUTE postp_buk {0,1}
@ATTRIBUTE haul {0,1}
@ATTRIBUTE tarteki {0,1}
@ATTRIBUTE puntu {0,1}
@ATTRIBUTE hiru_puntu {0,1}
@ATTRIBUTE bi_puntu {0,1}
@ATTRIBUTE puntu_eta_koma {0,1}
@ATTRIBUTE harridura_ikurra {0,1}
@ATTRIBUTE galdera_ikurra {0,1}
@ATTRIBUTE esaldi_muga_has {0,1}
@ATTRIBUTE esaldi_muga_buk {0,1}
@ATTRIBUTE perpaus_muga {0,1}
@ATTRIBUTE zenbat_AK_ezk REAL
@ATTRIBUTE zenbat_AK_esk REAL
@ATTRIBUTE zenbat_IS_ezk REAL
@ATTRIBUTE zenbat_IS_esk REAL
@ATTRIBUTE zenbat_mendeko_EM_ezk REAL
@ATTRIBUTE zenbat_mendeko_EM_esk REAL
@ATTRIBUTE dist_EM_ezk REAL
@ATTRIBUTE dist_EM_esk REAL
@ATTRIBUTE koma {0,1}

@DATA
100,0,-,-,-,-,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,3,0
1,1,IZE,IZB,ABS,-,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0
100,0,-,-,-,-,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,2,1,0
45,0,-,-,-,-,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0
...
```

Irudia IV.3: Arff formatuaren adibide bat, leihorik gabe.

guztia lortzeko, IXA taldearen tresnak baliatu genituen, hasieran: zehazki, *Eustagger* analizatzaile/desanbiguatzaile morfosintaktikoa, *Ixati* zatitzailea eta CG erregeletan oinarritutako mugatzailea.

*Eustagger*-ek testu soila den fitxategi bat hartzen du sarrerako datu gisa, eta testu horri dagokion analisi morfosintaktikoa itzultzen du, desanbiguatua. *Eustagger*-ek ematen duen informazioa, orduan, *Ixati*k darabil, kateen etiketak non jarri erabakitzeko. Azkenik, CG erregeletan oinarritutako mugatzaileak perpausen arteko mugak markatzen ditu. Laburbilduz, testu hutsetik komak ikasteko beharrezkoa zaigun hainbat informazio erdiesten dugu tresna hauen bidez.

Tresna hauek guztiek komak ere erabiltzen dituzte ahalik eta emaitza onenak lortu ahal izateko. *Eustagger*-ek, batez ere, desanbiguazio-prozesuan erabiltzen ditu komak. Desanbiguazio-prozesua bi mailatan egiten da: lehenengoan, CG erregelak baliatzen dira anbiguotasuna txikiagoa izan dadin; bigarrenean, eredu estokastiko bat erabiltzen da analizatzaileak desanbiguatzen ikas dezan. Erregelei dagokienez, batzuek koma kontuan hartzen dute; estokastikoari dagokionez, berriz, ikasketa egiterakoan komadun corpusa baliatu zen. *Ixati*k eta CG erregeletan oinarritutako mugatzaileak ere badituzte erregela batzuk, non komak baliatzen diren kateen etiketak finkatzeko.

Tresna hauek, esan bezala, komak erabiltzeko prestatuta daudenez, sarrera gisa komarik gabeko corpus bat ematen badiegu, tresna hauen emaitzek okerrera egitea besterik ezin da espero. Komak bere horretan uzteak ere, arazoak ekar ditzake, ordea. Koma-zuzentzailea izango den sailkatzailea sortzeko baliatzen dugun ikasketa-corpusa jatorrizko komekin analizatzeak[18] ez dakar berez arazorik, baina bai ikasitako sailkatzaile hori aplikatzeko beste edozein test-corpus jatorrizko komekin analizatzeak. Izan ere, ikasketa-corpusa —atributuen balioak lortzeko— jatorrizko komekin analizatu badugu, test-corpus gisa tratatzen diren hauek ere jatorrizko komekin analizatu beharra dauzkagu: ikasketa- eta test-corpuseko atributuei dagozkien balioek baldintza berdinetan lortutakoak izan behar baitute, emaitzek behera egin ez dezaten. Eta hortxe dago gakoa: test-corpuseko komak zuzendu nahi ditugunak baitira, eta, beraz, okerrak izan baitaitezke. Hala, test-corpuseko komen zuzentasun zalantzagarria dela-eta, analizatzailearen emaitza eta, ondorioz, test-corpuseko atributuen kalitatea ikasketa-corpusekoena baino txa-

[18] Azalpen honetan, laburtzeko, analizatzen ari garela esango dugu, baina analisi morfosintaktikoaz gain, desanbiguazioaz eta CG erregeletan oinarritutako zatitzaileak eta perpausen mugatzaileak egiten duten kateen eta mugen etiketatzeaz ere ari garela ulertu behar da.

rragoa izan daiteke, eta, hortaz, baita komak zuzentzen dituen sailkatzailearen emaitza ere, test-corpus horren gainean. Beste modu batean esanda, komak berreskuratzea bada gure helburua, testuaren egileak jarritako komak desegokiak izan daitezke analisi automatikoa egiteko.

Hau konpontzeko aukera bat IXA taldeko analizatzaile, zatitzaile eta mugatzaileak aldatzean datza, halako gisan non komak ezertarako erabil ez ditzaten. Alegia, test-corpuseko komek eragin negatiborik izan ez dezaten, koma horiek kentzea litzateke aukera bat. Horretarako, ordea, komak zuzentzeko darabilgun ikasketa-corpusetik ere komak kendu beharko lirateke, eta, ondorioz, baita *Eustagger*-en desanbiguatzaileak —desanbiguatzaileak daukan eredu estokastikoak— darabilen ikasketa-corpusetik ere. Modu honetan, komarik kontuan hartu gabe ikasiko luke desanbiguatzaileak. Gainera, testuan komarik ez bagenu, komak kontuan hartzen dituzten CG erregelak ere ez lirateke aplikatuko.

Komak kontuan hartzen ez dituen analizatzaile/desanbiguatzaile automatikoari *Eustagger komagabe*a deitu genion. IV.6.4.2 atalean emango dugu modu honetan lortutako emaitzen berri.

Beste bide bat ere jorratu genuen, ordea. Komak zuzentzeko sailkatzailerako darabiltzagun ikasketa- eta test-corpusei jatorrizko komak kentzea, baina *Eustagger komaduna* —komak darabiltzana— baliatzea; hau da, eredu estokastikoen bidez garatutako desanbiguatzaileak darabilen ikasketa-corpusean komak bere horretan uztea, nahiz eta desanbiguatzaileari pasatuko zaizkion testuak komagabeak izan. IV.6.4.1 atalean aurkeztuko ditugu modu honetan lortutako emaitzak.

Bestalde, antzeko arazoa aurreikusi genuen informazio linguistiko gisa aurreko kapituluan aurkeztutako —ikasketa automatikoko *FR-Perceptron* algoritmoan oinarritutako— kate- eta perpaus-identifikatzaileen iragarpenak gehitzean. Kate- eta perpaus-identifikatzaile hauek, berez, komak baliatzen dituzte ahalik eta informazio osoena lortzeko; izan ere, HPko hainbat atazatan erabili nahi dira. Komen berreskurapenerako, aldiz, test-corpuseko komen zuzentasuna zalantzagarria izanik, koma hauen erabilerak kate- eta perpaus-identifikatzaileen emaitzetan eragin negatiboa izan dezake. Hau ekiditeko, ez genituzke test-corpuseko komak erabili beharko, eta, ondorioz, ezta ikasketa-corpusekoak ere.

Hori dela eta, kate- eta perpaus-identifikatzaile bereziak prestatu genituen (ikus IV.6.2.9 atala): komak erabiltzen ez dituzten euskarako kate- eta perpaus-identifikatzaileak. Komak berreskuratzeko atazan ebaluatzeko orduan, halere, biak ala biak erabili ditugu: kate- eta perpaus-identifikatzaile

komadunak eta komagabeak.

Jatorrizko komen eragin negatiboa, ordea, zalantzagarria da. Izan ere, erabiltzaileak jarritako komak, batzuetan —eta betiere erabiltzailearen hizkuntza mailaren arabera—, baliagarriak izan daitezke eta haiekin lortzen den analisia eta analisi horrek ematen duen informazio linguistikoa egokiagoa izan daiteke, komak erabili gabe lortutakoa baino. Gauzak horrela, esperimentu guztiak jatorrizko komak mantenduz egitea erabaki genuen, hasiera batean.

## IV.6.2 Egindako saioak

Gure sistema fintzeko helburuarekin proba desberdinak egin genituen. Kontrakorik esan ezean, *Euskaldunon Egunkaria* corpusa erabili genuen.

### IV.6.2.1 Leihoaren aukeraketa

*Euskaldunon Egunkaria*ko ikasketa-corpusarekin (ikus IV.6.1.2 atala), beraz, aplikazio-leihoa erabakitzeko esperimentuak egin genituen lehendabizi. Token konkretu bati atzetik koma datorkion ala ez erabakitzeko, izan ere, aurreko eta ondorengo tokenen informazioa baliagarria izan zitekeela pentsatu genuen. Hala, leiho desberdinekin probatu genuen sistema, token bakoitzarentzat aurreko eta ondorengo zenbat tokenen informazioa kontuan hartu erabakitzeko.

IV.5 taulan ikus daitekeen moduan, ez dago alde handirik leihoaren tamainaren arabera. 0 klaserako ez dago ia alderik. 1 klaserako, berriz, $F_1$ neurrirako emaitzak 3 punturen bueltan dabiltza guzti-guztiak, eta zazpi leiho daude emaitza onenetik ($F_1 = \%\ 52$) puntu bakar baten bueltan. Antzeko $F_1$ neurria dutenen artetik, doitasun handiena lortu zuena aukeratu genuen, doitasunak molde honetako zuzentzaileetan daukan garrantziaren jakitun (Guinovart, 1996b): (-5,+2) leihoa, hain zuzen ere. Alegia, token bakoitzaren aurreko bost tokenen eta ondorengo biren informazioa hartu genuen kontuan.

### IV.6.2.2 Ikasketa-algoritmo egokienaren aukeraketa

IV.6.2.1 atalean erabakitako (-5,+2) leihoarekin, ikasketa-algoritmoa aukeratzeko probak egin genituen ondoren. Aipatu bezala, WEKA paketeko hiru ikasketa-algoritmo probatu genituen: erabaki-zuhaitzak (C4.5 inplementa-

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **(-2,+5)** | 95,6 | 98,2 | 96,9 | 64,8 | 43,1 | 51,8 |
| **(-3,+5)** | 95,7 | 97,9 | 96,8 | 62,7 | 44,1 | 51,8 |
| **(-4,+5)** | 95,7 | 98,0 | 96,8 | 63,4 | 44,6 | **52,0** |
| **(-5,+5)** | 95,5 | 98,1 | 96,8 | 63,5 | 41,7 | 50,3 |
| **(-5,+4)** | 95,5 | 98,2 | 96,8 | 64,0 | 41,7 | 50,5 |
| **(-5,+3)** | 95,6 | 98,1 | 96,9 | 64,3 | 43,2 | 51,7 |
| **(-5,+2)** | 95,6 | 98,2 | 96,9 | **65,0** | 42,4 | 51,4 |
| **(-6,+2)** | 95,6 | 98,2 | 96,9 | 64,5 | 42,1 | 50,9 |
| **(-6,+3)** | 95,6 | 98,2 | 96,9 | 64,6 | 42,6 | 51,4 |
| **(-8,+2)** | 95,6 | 98,2 | 96,9 | 64,5 | 42,5 | 51,3 |
| **(-8,+3)** | 95,6 | 97,9 | 96,7 | 61,5 | 43,1 | 50,7 |
| **(-8,+8)** | 95,6 | 97,8 | 96,7 | 60,4 | 42,2 | 49,7 |

Taula IV.5: Garapen-corpusean kalkulatutako emaitzak, leihoaren arabera (C4.5 algoritmoa erabilita).

zioan), *Naive Bayes* eta *Support Vector Machine* (ikus emaitzak, IV.6 taulan).

0 klaserako emaitzak oso antzekoak dira ikasketa-algoritmo guztientzat. 1 klaserako, aldiz, alde handiak daude. Zalantzarik gabe, erabaki-zuhaitzak dira emaitzarik onenak lortzen dituztenak. Hala ere, deigarria da *Support Vector Machine* algoritmoak lortzen duela doitasun onena (% 67,2), baina oso gutxi arriskatuz, estaldurak adierazten digun moduan (% 14,3). Badirudi SVM algoritmoarekin emaitza onak lortzeko askoz ezaugarri gehiago erabili beharko genituzkeela. Matematikako artikuluak sailkatzeko lan batean, hobekuntza adierazgarriak erdietsi zituzten, hain zuzen, SVM algoritmoarekin, 500 ezaugarri *soilik* erabiltzetik, 20.000 ezaugarri erabiltzera pasatzerakoan (Rehurek eta Sojka, 2010).

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **C4.5** | 95,6 | 98,2 | 96,9 | 65,2 | 42,4 | **51,4** |
| ***Naive Bayes*** | 94,8 | 95,6 | 95,2 | 37,6 | 33,5 | 35,5 |
| ***SVM*** | 93,6 | 99,4 | 96,5 | **67,2** | 14,3 | 23,6 |

Taula IV.6: Garapen-corpusean ebaluatutako emaitzak, ikasketa-algoritmoaren arabera ((-5,+2) leihoa erabilita).

*Naive Bayes* algoritmoa izan zen baztertu genuen lehenengoa. Aipatzekoa da, dena dela, algoritmo honekin ere *oinarrizko neurri*ak gainditu genituela[19]. Bestalde, hurrengo probetarako erabaki-zuhaitzak erabiltzea deliberatu genuen, $F_1$ neurrirako emaitza onenak lortzeaz gain, *SVM* algoritmoa baino askoz azkarragoa baita. Hala eta guztiz ere, *SVM* ez genuen erabat baztertu. Izan ere, corpus handiagoarekin, eta batez ere atributu askorekin, emaitza onak lor ditzakeela esaten baita literaturan (Milenova *et al.*, 2005; Joachims, 1998).

#### IV.6.2.3 Adibideen aukeraketa

Proba hauek egin ostean, *oinarrizko neurri*ekin aurreikusitakoa betetzen ari zela ohartu ginen: 0 klaserako eta 1 klaserako lortutako emaitzen desberdintasunak handiak ziren; $F_1$ neurrirako 45 puntuko aldea zegoen bi klaseen emaitzen artean: $F_1$ = % 96,9 lortzen zen, 0 klaserako; 1 klaserako, berriz, $F_1$ =% 51,4.

Arestian esan dugun eran, honen arrazoia begi-bistakoa da: ikasketa-corpusean 0 klaseko askoz adibide gehiago daude, 1 klasekoak baino; alegia, corpusa ez da *orekatua* emaitza-atributuko bere bi klaseekiko. Modu argiagoan esanda: ikasketa-corpusean askoz adibide gehiago daude atzetik komarik ez dutenak, atzetik koma dutenak baino. Errazago ikasiko du, beraz, sistemak, komarik gabeko adibideak sailkatzen, koma dutenak baino.

Adibide kopuruen arteko desorekak dakartzan arazoak konpontzeko bide bat adibideak gehitzean edo kentzean datza, halako moduan non bi klaseko adibide kopuruen arteko oreka bilatzen den (Zhang eta Mani, 2003). Honela, komarik gabeko adibideak kendu genituen bi klaseen maiztasunak konpentsatzeko. Proba desberdinak egin genituen, komadun adibide bakoitzeko, komarik gabeko $x$ adibide utzita. $x$ aldagaiaren balioa aldatuz, lortu genituen IV.7 taulako emaitzak. $x = 1$ denean, esaterako, komadun adibide bakoitzeko, esaldian komarik gabeko adibide bakarra utzi genuela esan nahi du.

IV.7 taulan ikus daitekeen moduan, komadun eta komarik gabeko adibideen kopurua geroz eta berdinduago, orduan eta estaldura handiagoa lortzen da 1 klasean. Doitasuna, halere, nabarmen txikiagotzen da klase horretan bertan. Izan ere, adibide batzuk —esaldi barruko token batzuk— kendu egiten dira. Horrek, informazio-galera dakar nahitaez. Hala eta guztiz ere,

[19]Zenbait atazatan, *oinarrizko neurri*ak kalkulatzeko erabili ohi da *Naive Bayes*, bere sinpletasunarengatik.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **x=1** | 98,9 | 63,3 | 77,2 | 16,4 | 91,2 | 27,7 |
| **x=2** | 97,7 | 90,2 | 93,8 | 36,7 | 72,5 | 48,7 |
| **x=3** | 96,9 | 93,4 | 95,1 | 42,7 | 62,1 | 50,6 |
| **x=4** | 96,6 | 95,2 | 95,9 | 48,4 | 57,5 | 52,6 |
| **x=5** | 96,6 | 96,1 | 96,3 | 53,4 | 56,8 | **55,0** |
| **x=6** | 96,3 | 96,6 | 96,4 | 55,0 | 52,4 | 53,7 |
| **guztiak** | 95,6 | 98,2 | 96,9 | **65,2** | 42,4 | 51,4 |

Taula IV.7: Garapen-corpusean ebaluatutako emaitzak, kendutako adibide komagabe kopuruaren arabera. x aldagaiak adierazten du komadun adibide bakoitzeko komarik gabeko zenbat adibide izango diren, esaldiko; *guztiak* deitu diogu adibiderik batere kendu gabeko corpusari (erabaki-zuhaitzak eta (-5,+2) leihoa erabilita).

leihoarekin inguruko tokenen informazioa mantentzen denez, komadun bakoitzeko lauzpabost komarik gabeko uzten ditugunean, apur bat hobetzen dira $F_1$ neurriko emaitzak. Doitasun onena, dena dela, komarik gabeko adibideak kendu gabeko corpusarena da, alde handiarekin (ikus IV.7 taulako *guztiak* lerroko emaitzak). Hori dela eta, gainerako probak ere corpuseko adibide *guztiak* erabiliz egitea erabaki genuen.

#### IV.6.2.4 Corpus motaren eragina

Arestian aipatu dugun moduan, corpus mota desberdinekin probak egitea deliberatu genuen. Ikasketa automatikoan sarri egin ohi den proba bat da hau, eta komaren kasuan are arrazoituago dagoena, komaren erabilera ez baitago guztiz normalizatua, arestian aipatu bezala. Gainera, susmatzen genuen corpus literario batean, esate baterako, ez direla komak egunkari-corpus batean bezala erabiltzen. Horretaz gain, egile bakar batek idatzitako testuetan komaren erabilera homogeneoagoa izango zela pentsatzen genuen. Arrazoi hauek zirela medio, eskura genituen corpus desberdinekin konparazio-proba bat egitea komenigarritzat jo genuen. Konparagarritasuna ziurtatzeko, ordea, token kopuru bereko corpusekin egin behar izan genituen probak. Hala, egile bakar baten corpus handiena 25.000 tokenekoa lortu genuenez, corpus guztiak tamaina horretakoak hartu genituen. *Cross-validation* bidez ebaluatu genituen (ikasketa-corpusa hamar zatitan banatuz) corpus desberdin

bakoitzarekin ikasitako sailkatzaileak.

Kontuan hartu behar da filosofiako testua idazle bakar batek idatzitakoa dela, eta itzulpen bat dela gainera. Literaturako corpusa ere idazle bakar batek idatzitakoa da, baina *Euskaldunon Egunkaria* eta Elhuyar Fundazioaren *Zientzia eta Teknika* corpusak idazle anitzenak dira.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Egunkaria** | 93,0 | 97,9 | 95,4 | 48,7 | 21,4 | 29,7 |
| **Filosofia-itzulpena** | 93,0 | 97,0 | 95,0 | 60,4 | 38,8 | **47,3** |
| **Literatura** | 92,7 | 97,5 | 95,0 | 50,4 | 25,0 | 33,4 |
| **Zientzia eta Teknika** | 94,9 | 98,5 | 96,6 | 49,5 | 22,4 | 30,8 |

Taula IV.8: *Cross-validation* emaitzak, corpus motaren arabera (25.000 tokeneko corpusak, erabaki-zuhaitzak eta (-5,+2) leihoa erabilita).

IV.8 taulan ikus daitekeen moduan, pertsona bakar batek —itzultzaile batek, kasu honetan— itzulitako filosofia-testu batean lortzen dira 1 klaseko emaitza onenak. Badirudi testu motak baino, eragin handiagoa duela testu hori pertsona bakar batek idatzi eta pertsona horrek halako homogeneotasun bat mantendu izanak komak jartzerakoan. Alegia, komak modu homogeneoago batean izango dira jarriak, eskuarki, idazle bakar baten corpus batean, idazle anitzek idatzitako corpusetan baino; baina, era berean, idazlearen mende egongo da homogeneotasun hori. Hala, proba honetarako erabili dugun literatur testuak, idazle bakar batek idatzi izanagatik ere, badirudi ez daukala horrenbesteko homogeneotasuna, edo testuaren konplexutasuna handiagoa —eta, beraz, tratatzen zailagoa— dela: nahiz eta bigarren emaitza onenak literatur testuarenak izan, idazle anitzek idatzitako corpusarekin ikasitako sailkatzaileei ez die alde handirik ateratzen. Pentsa daiteke, era berean, itzultzaile batek zorrotzago beteko dituela gramatika-arauak. Idazle batek, konparazioan, libreago joka dezakeela uste dugu, eta lexiko aberatsagoa eta sintaxi konplexuagoa erabil ditzakeela. Honek ikasketa-prozesua zailagoa egingo luke. Bestalde, egunkariko emaitzetan argi ikusten da esperimentu honetan darabilgun corpusa txikiegia dela; izan ere, 1 klaseko emaitzetan galera handia dago, orain arte erabilitako 100.000 token pasatxoko ikasketa-corpusarekin konparatuta (ikus C4.5 emaitzak, IV.6 taulan).

#### IV.6.2.5 Ingeleseko corpusarekin komak ikasten

Emaitzak guk nahi bezain onak ez zirela ikusita, beste proba bat egin nahi izan genuen. Pentsatu genuen, agian, ingelesaren gisako hizkuntza normalizatu batean —non komak jartzeko arauak euskaraz baino finkatuago egongo zirela iruditzen zitzaigun—, emaitza hobeak lortuko genituzkeela. Ingeleseko hitzen ordena zurrunagoak ere lagunduko zuelakoan geunden. Hala, ingeleseko corpus konparagarri bat aukeratu genuen, bai tamaina aldetik, bai corpus motaren aldetik: *Reuters*-eko kazetaritza-corpusa, hain zuzen. Baliatu genuen informazio linguistikoa ere ahalik eta antzekoena izan zedin saiatu ginen. *Freeling*[20] (Atserias *et al.*, 2006) erabili genuen, hain zuzen ere, token bakoitzaren informazio linguistikoa lortzeko, eta informazio horretatik abiatuz atributu berrien balioak kalkulatzeko. Hauek izan ziren, azkenik, erabili genituen atributuak:

- hitza
- lema
- kategoria: izena, adjektiboa, aditza...
- kate baten hasiera den: aditz-kate, sintagma, hitz anitzeko unitate edo postposizio baten hasiera den ala ez adierazten da (atributu bitarra).
- kate baten bukaera den: aditz-kate, sintagma, hitz anitzeko unitate edo postposizio baten bukaera den ala ez adierazten da (atributu bitarra).
- puntua den ala ez (atributu bitarra)
- puntu eta koma den ala ez (atributu bitarra)
- hiru puntuak den ala ez (atributu bitarra)
- bi puntuak den ala ez (atributu bitarra)
- harridura-marka den ala ez (atributu bitarra)
- galdera-marka den ala ez (atributu bitarra)

[20]Hizkuntza-analisirako kode irekiko tresna bat da Freeling (http://garraf.epsevg.upc.es/freeling/).

- uneko tokenetik esaldiaren hasierara dagoen aditz-kate kopurua (atributu kalkulatua)
- uneko tokenetik esaldiaren bukaerara dagoen aditz-kate kopurua (atributu kalkulatua)
- uneko tokenetik esaldiaren hasierara dagoen sintagma kopurua (atributu kalkulatua)
- uneko tokenetik esaldiaren bukaerara dagoen sintagma kopurua (atributu kalkulatua)
- uneko tokenetik esaldiaren hasierara dagoen token kopurua (atributu kalkulatua)
- uneko tokenetik esaldiaren bukaerara dagoen token kopurua (atributu kalkulatua)

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Euskara** | 95,6 | 98,2 | 96,9 | 65,2 | 42,4 | 51,4 |
| **Ingelesa** | 97,8 | 99,7 | 98,7 | 83,3 | 38,7 | 52,8 |

Taula IV.9: Bi hizkuntzen garapen-corpusetan ebaluatutako emaitzak, hizkuntzaren arabera (erabaki-zuhaitzak eta (-5,+2) leihoa erabilita).

IV.9 taulan ikus daitekeen moduan, ingeleserako emaitzak, ahalegin berezirik egin gabe, euskarakoak baino hobeak dira. Gainera, kontuan hartzekoa da erabili dugun leihoa euskararako egokiena zena dela; hau da, litekeena da ingelesekoa beste leiho batekin hobeto ibiltzea. Horretaz aparte, baterako eta besterako erabili dugun informazio linguistikoa —hots, atributuak— ez dira berberak; euskaraz, ingeleseko guztiez gain, beste batzuk ere erabili ditugu (deklinabide-kasua, mendeko marka...). Beraz, uste dugu gure hipotesia baieztatzekotan gaudela: ingeleserako komak ikastea euskararako ikastea baino errazagoa da, eta honen arrazoia hizkuntza biek normalizazio eta estandarizazio mailan daukaten aldean bilatu behar da batez ere, baina baita euskararen esaldiaren antolamendu libreagoan ere.

Hala eta guztiz, ingeleserako lortzen diren emaitzak ere ez dira onak, eta honek atazaren zailtasuna erakusten duela uste dugu.

#### IV.6.2.6 Atributu berrien gehikuntza

Emaitzak uste bezain onak ez zirenez, informazio berria gehitzea erabaki genuen; hots, atributu edo ezaugarri berriak eranstea. Hala, komaren aurretik maizen agertzen diren hitzak atributu bitar gisa gehitu genituen. Hauek izan ziren egin genituen probak:

1. Gure ikasketa-corpusa aztertu genuen komaren aurretik maizen agertzen ziren ehun hitzak, ehun hitz-bikoteak (*bigram*ak) eta ehun hitz-hirukoteak (*trigram*ak) lortu eta atributu gisa erabiltzeko (300 atributu berri, guztira).

2. Aipatutako 300 atributu horien ordez, 3 atributu soilik erabiltzea. Lehenengoak uneko tokena maiztasun handieneko ehun hitzen artean zegoen ala ez adierazten zuen; bigarrenak, berriz, uneko tokena maiztasun handieneko ehun *bigram*en parte ote zen; eta, hirugarrenak, ostera, maiztasun handieneko ehun trigramen parte ote zen.

IV.10 taulan ikus daitezke lortutako emaitzak. Espero bezala, 1 klaseko emaitzak nabarmen hobetu ziren.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Atributu berririk gabe** | 95,6 | 98,2 | 96,9 | 65,2 | 42,4 | 51,4 |
| **(1) 300 atributu berri** | 96,0 | 98,3 | 97,2 | 69,6 | 48,6 | **57,2** |
| **(2) 3 atributu berri** | 96,0 | 98,1 | 97,0 | 66,5 | 48,1 | 55,8 |

Taula IV.10: Garapen-corpusean ebaluatutako emaitzak, atributu berriak gehituta ala kenduta (erabaki-zuhaitzak eta (-5,+2) leihoa erabilita).

300 atributuen ordez hiru atributu gehituta, 1 klaseko emaitzak lau puntu baino gehiago hobetu baziren ere, 300 atributuak gehituta lortu ziren emaitzarik onenak (ia sei puntuko hobekuntza, 1 klaseko $F_1$ neurrian). Beraz, lehen puntuko 300 atributuekin jarraitzea erabaki genuen.

#### IV.6.2.7 Ikasketa-algoritmoa finkatuz

Puntu honetan, lehendik genituen atributuak baino 300 atributu gehiago izatera pasa ginelarik, *support vector machines* algoritmoa baliatu nahi izan genuen, jakinik atributu askorekin ondo dabilen ikasketa-algoritmoa dela

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **C4.5** | 95,6 | 98,2 | 96,9 | 65,2 | 42,4 | 51,4 |
| **C4.5, 300 atributu berriekin** | 96,0 | 98,3 | 97,2 | 69,6 | **48,6** | **57,2** |
| ***SVM*** | 93,6 | 99,4 | 96,5 | 67,2 | 14,3 | 23,6 |
| **SVM, 300 atributu berriekin** | 94,1 | 99,6 | 96,8 | **79,5** | 21,0 | 33,2 |

Taula IV.11: Erabaki-zuhaitzak (C4.5) eta *support vector machines* (SVM) ikasketa-algoritmoekin lortutako emaitzen konparaketa (garapen-corpusaren gainean ebaluatuta), 300 atributu berriak gehituta eta gehitu gabe ((-5,+2) leihoa erabilita).

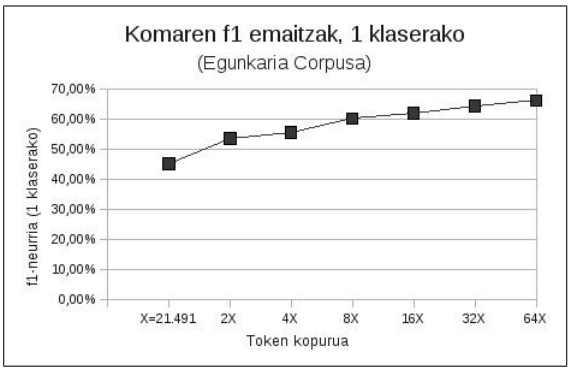
(Rehurek eta Sojka, 2010), lehen aipatu dugun moduan. IV.6.2.2 atalean egindako ikasketa-algoritmoen arteko konparaketaren emaitzak atributu berriak erantsita ere mantentzen ote ziren egiaztatu nahi genuen.

IV.11 taulan ikus daitekeen moduan, 300 atributuak gehituta SVM algoritmoarekin lortutako emaitzak erabaki-zuhaitzekin lortutakoak baino gehiago hobetu ziren arren (hamar puntu hobetu ziren SVM algoritmoarekin; erabaki-zuhaitzekin, sei), erabaki-zuhaitzekin lortu ziren emaitzarik onenak. Hau dela eta, gure gainerako esperimentuak ere, erabaki-zuhaitzekin egin genituen.

Klase bateko eta besteko kopuruen desberdintasunak —hau da, datuen desorekak— eragin negatiboa du, antza, SVM algoritmoaren portaeran (Lewis *et al.*, 2004; Li eta Shawe-Taylor, 2003). Hala ere, badira moduak desoreka honek dakarren arazoa konpontzeko. Esaterako, SVM algoritmoa marjina aldagarriekin baliatu dute duela gutxi (Li *et al.*, 2009) lanean, eta arrakasta lortu dute klaseen arteko desoreka handiko problemetan; soluziobide honetan, bi klaseen arteko marjina handieneko hiperplanoa bilatu beharrean, marjina bana bilatzen dute klase bakoitzerako; klase baten nagusitasuna orekatzen da modu honetan. Klaseen adibide kopuruaren desorekak sortutako arazoari aurre eginez eta atributu gehiago erantsita are hobekuntza handiagoak lor daitezke SVM algoritmoarekin. Etorkizunean aztertzeko bide egokia izan daiteke hau.

#### IV.6.2.8 Corpusaren tamainaren eragina

Corpusaren tamainaren eragina ere aztertu nahi izan genuen, corpusa handitzearekin emaitzak zenbat hobetzen ziren ikusteko. Horretarako *Euskaldunon*

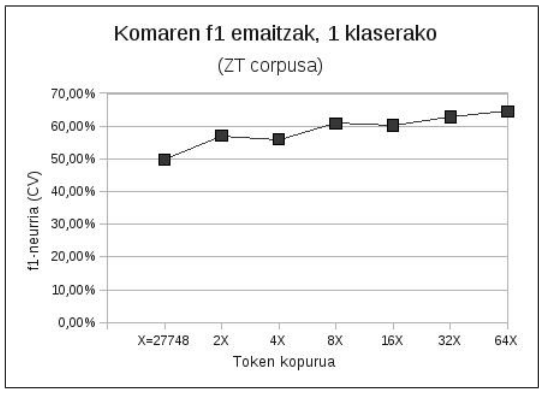



Irudia IV.4: Corpusaren tamainaren eragina koma zuzenen identifikaziorako, *Euskaldunon Egunkaria* corpusa baliatuta (hamar zatiko *cross-validation* baliatuta lortutako emaitzak).

*Egunkaria*ko eta *Zientzia eta Teknika*ko corpus handiak baliatu genituen. Erabaki-zuhaitzak (C4.5) erabili genituen ikasketarako, (-5,+2) leihoa, eta 300 atributu gehigarriak.

IV.4 eta IV.5 irudietan ikus daitekeen moduan, corpusa gero eta handiagoa izan, emaitzak orduan eta hobeak dira. Gainera, badirudi emaitzak gehiago hobe daitezkeela corpusaren tamaina are gehiago handituta; alegia, corpusaren tamaina handituz lor daitekeen goi-mugara ez garela heldu iruditzen zaigu. Dena dela, kontuan hartu behar da proba bakoitzean aurreko proban erabilitako corpusaren bikoitza erabili genuela. Normala den bezala, emaitzen hobekuntza geroz eta mantsoagoa dela ikus daiteke. 0 klaseko emaitzak ez zaizkigu esanguratsuak iruditu, eta ez ditugu jarri. Baiki, 0 klaseko emaitzak oso onak dira hasiera-hasieratik, eta ez daukate hobetzeko tarte handirik. Esanguratsua da, era berean, antzeko emaitzak lortzen direla *Euskaldunon Egunkaria* edo *Zientzia eta Teknika* corpusak erabilita (% 66ren bueltan dabiltza biak ala biak, corpusaren zati handienarekin).

Goi-muga kalkulatzeko, dena dela, beste modu bat aurreikusi genuen: puntuazioan adituak diren bi hizkuntzalarik corpus txiki bat eskuz etiketatzea; berek lortutako emaitza hartuko genuen goi-mugatzat (ikus IV.6.5 atala). Hala ere, ikasketan pasatako denbora larregi handitzen da corpusa handitzearekin. Beraz, hemendik aurrerako probak ere, orain artekoak bezala, 100.000 token pasatxoko ikasketa-corpusarekin egitea erabaki genuen.




Irudia IV.5: Corpusaren tamainaren eragina koma zuzenen identifikaziorako, *ZT corpusa* baliatuta (hamar zatiko *cross-validation* baliatuta lortutako emaitzak).

#### IV.6.2.9 Kateen eta perpausen identifikatzaileen informazioa koma-zuzentzailea hobetzeko

Aurreko kapituluko III.5 eta III.6 ataletan azaldutako kateen eta perpausen identifikatzaileek ematen duten informazioa balio handikoa iruditzen zaigu koma-zuzentzailearentzat, are gehiago (Shieber eta Tao, 2003) lanean esandakoa kontuan hartuz gero: osagaien mugei dagokien informazioa erantsiz, beren koma-berreskuratzailea hobetzea lortu zutela, hain zuzen. Intuitiboki ere hala dela esatea ez da zentzugabekeria. Izan ere, arestian ikusi dugun moduan, kate baten barruan ez da oro har komarik izango; bestalde, badira perpaus-muga batzuk komaz markatu behar direnak. Horretaz ohartzeko, IV.4 atala —eta zehatzago, mintzagaiaren araua, adibidez— errepasatzea besterik ez dago, esate baterako: mintzagaia perpaus bat baldin bada eta aditz nagusiaren aurretik baldin badoa, koma behar du atzetik.

Arrazoi hauek direla medio, *FR-Perceptron* bidez sortutako kate- eta perpaus-identifikatzaileek emandako informazioa ikasketarako atributu gisa gehitzea erabaki genuen, ordura arte erabilitako CG erregelek emandako kateen eta perpausen informazioaren ordez. Izan ere, III. kapituluan ikusi dugun moduan, *FR-Perceptron* bidez sortutako kate- eta perpaus-identifikatzaileak emaitza hobeak eskuratzen dituzte; gainera, CG erregelen informazioa ere baliatzen dute.

Arestian aipatu dugun moduan, ordea, zalantzagarria da komak darabiltzaten kate- eta perpaus-identifikatzaileak erabiltzea komak ikasteko. Ho-

rregatik, kateen eta perpausen identifikatzaileak apur bat aldatu behar izan genituen: kateen eta perpausen identifikatzaile berriak sortu behar izan genituen, komak kontuan hartzen ez zituztenak beren eginbeharretan. Kate- eta perpaus-identifikatzaile gisa sortutako sailkatzaileei komarik gabeko test-corpus bat pasatzean, emaitzen beherakada handirik ez izatea zen helburua. Horretarako, sailkatzaile hauek sortzeko erabilitako ikasketa-corpusetik komak kendu, eta berriz sortu genituen kate- eta perpaus-identifikatzaileak. Komarik ez darabiltzaten hauei, kate- eta perpaus-identifikatzaile *komagabea*k deitu diegu.

IV.12 eta IV.13 tauletan ikus daitezke komarik gabeko kateen eta perpausen identifikatzaileen emaitzak, aurreko kapituluan aurkeztutako kateen eta perpausen identifikatzaile komadunekin konparatuta (test-corpusean egindako ebaluazioa).

| **Ikasketa-corpusa** | **Test-corpusa** | **Desanb.** | $F_1 neurria$ |
|---|---|---|---|
| *Komaduna* | *Komaduna* | autom | 83,17 |
| *Komagabea* | *Komagabea* | autom. | 82,24 |

Taula IV.12: Komaren eragina, *FR-Perceptron* bidezko euskarako kateen identifikatzailean.

| **Ikasketa-corpusa** | **Test-corpusa** | **Desanb.** | $F_1 neurria$ |
|---|---|---|---|
| *Komaduna* | *Komaduna* | autom. | 77,24 |
| *Komagabea* | *Komagabea* | autom. | 73,66 |

Taula IV.13: Komaren eragina, *FR-Perceptron* bidezko euskarako perpausen identifikatzailean.

Datu hauen arabera, badirudi perpausak identifikatzeko garrantzitsuagoa dela koma, kateak identifikatzeko baino.

Komarik gabeko corpusean oinarritutako kate- eta perpaus-identifikatzaile hauek emandako informazioa erabili genuen komen ikasketa hobetzen saiatzeko, emaitzen beherakada ez zitzaigulako handiegia iruditu, batetik, eta gaizki jarritako komen eragin negatiboa —tresna hauetan, behintzat— saiheste aldera, bestetik.

IV.14 taulan ikus dezakegun eran, 1 klaseko emaitzak zazpi puntu hobetu ziren kate- eta perpaus-identifikatzaile *komagabe*ek emandako informazioare-

| | 0 | | | 1 | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Kate-info. eta perpaus-info gabe** | 96,0 | 98,3 | 97,2 | 69,6 | 48,6 | 57,2 |
| **Kate-ident *komagabearen* info. gehituta** | 96,0 | 98,4 | 97,2 | 70,4 | 48,5 | 57,4 |
| **Kate- eta perpaus-ident *komagabeen* info. gehituta** | 96,6 | 98,7 | 97,6 | 76,6 | 55,7 | **64,5** |
| **Kate-ident *komadunaren* info. gehituta** | 96,2 | 98,5 | 97,3 | 73,0 | 50,7 | 59,8 |
| **Kate- eta perpaus-ident *komadunen* info. gehituta** | 96,9 | 98,7 | 97,8 | 78,4 | 59,8 | **67,9** |

Taula IV.14: Garapen-corpuseko emaitzak, *FR-Perceptron* algoritmoaren bidez sortutako kate- eta perpaus-identifikatzaile *komagabe*ak edo *komadun*ak emandako informazioa gehitu aurretik eta gehitu ondoren (erabaki-zuhaitzak, (-5,+2) leihoa eta 300 atributu gehigarriak erabilita).

kin; hamar puntu baino gehiago, kate- eta perpaus-identifikatzaile komadunekin. Hobekuntza hauek, gainera, esanguratsuak direla ziurtatu ahal izan genuen, McNemar testa eginez ($p < 0,05$), bi kasuetan.

Kateen informazioarekin lortu zen hobekuntza oso txikia den arren, hobekuntza handiak lortu ziren perpaus-identifikatzaileak emandako informazioa gehituta (ikus IV.14 taula). Guztira, zazpi puntu inguruko hobekuntza lortu zen, kate- eta perpaus-identifikatzaile *komagabe*ak erabiliz. Kate- eta perpaus-identifikatzaile komadunak erabilita, berriz, are emaitza hobea erdietsi zen. Zehazki, hiru puntu eta erdiko aldea dago kate- eta perpaus-identifikatzaile *komadun*ak erabiltzetik *komagabe*ak erabiltzera (% 64,5 vs % 67,9). Bi arrazoi egon daitezke horretarako: bata, perpaus-identifikatzailearen beraren emaitza hobeak, komaren informazioa kontuan hartuz gero; bestea, kate-identifikatzaile *komadun*aren portaera hobea, *komagabe*arenarekin alderatuta (ikus IV.12 eta IV.13 taulak). Badirudi, hala ere, batarekin eta bestearekin lortu zen aldea ikusita, identifikatzaile *komadun*ekin eskuratzen zen hobekuntza kate-identifikatzaile *komadun*ari zor zaiola gehienbat. Hau da, komak ikasteko atazarako, perpaus-identifikatzaileak ematen duen informazioa kate-identifikatzaileak ematen duena baino garrantzitsuagoa den arren, kate-identifikatzailearen zuzentasunak kritikoagoa dirudi.

Beraz, gure hipotesia betetzen dela baiezta genezake: kateen eta perpausmugen informazioa garrantzitsuak dira koma-zuzentzailerako. Bestetik, argi

dago informazio linguistiko esanguratsua gehitzeak onurak dakartzala, eta informazio linguistiko horrek geroz eta kalitate hobea izan, orduan eta emaitza hobeak lortzen direla.

Hemendik aurrerako probak, hortaz, kateen eta perpausen ikasketa automatikoko informazioa baliatuz egin ziren, emaitzarik onenak horrelaxe lortzen zirelako IV.14 taulan ikus daitekeen gisan.

### IV.6.3 Komen zuzenketa, erregelak eta ikasketa automatikoa konbinatuz

Tesi-lan honetan sarritan aipatu dugun moduan, corpusetan oinarritutako teknikak (kasu honetan, ikasketa automatikokoak) eta hizkuntzaren ezagutzan oinarritutakoak (erregela bidezkoak) konbinatuz, bataren eta bestearen emaitzak hobetzen dira eskuarki. III. kapituluan frogatu ahal izan dugu hori. Izan ere, bi tekniken konbinazioarekin kateen eta perpausen identifikazioan lortu ditugun emaitzarik onenak erdietsi ditugu. Komarekin ere gauza bera egin genuen: ikasketa automatiko bidez lortutako sailkatzaileari IV.5 atalean aurkeztutako CG erregelek emandako informazioa beste atributu baten moduan gehitu genion, *stacking* teknika erabiliz. IV.15 taulan jarri ditugu emaitzak.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **CG erregelak** | 93,1 | 96,7 | 94,9 | 56,9 | 27,2 | 36,8 |
| **Ikasketa automatikoa *KPI-komagabearekin*** | 96,6 | 98,7 | 97,6 | 76,6 | 55,7 | 64,5 |
| **CG erregelak + ikask. autom. *KPI-komagabearekin*** | 96,5 | 98,8 | 97,6 | 77,8 | 55,0 | 64,4 |
| **Ikask. autom. *KPI-komadunarekin*** | 96,9 | 98,7 | 97,8 | 78,4 | 59,8 | 67,9 |
| **CG erregelak + ikask. autom. *KPI-komadunarekin*** | 97,0 | 98,7 | 97,8 | 79,0 | 61,4 | **69,1** |

Taula IV.15: Garapen-corpusaren gainean ebaluatutako koma-zuzentzailearen emaitzak, hizkuntzaren ezagutzan (CG erregelak) eta corpusetan oinarritutako teknikak (erabaki-zuhaitzak, (-5,+2) leihoarekin eta 300 atributu gehigarriekin) konbinatuz, kate- eta perpaus-identifikatzaile (KPI) *komadun*arekin edo *komagabe*arekin.

Kate- eta perpaus-identifikatzaile *komagabe*ak erabiltzen dituguneanean, ikasketa automatikoko teknikak erregelekin konbinatzean lortzen diren emaitzak erregelekin soilik lortutakoak baino askoz hobeak dira, baina ikasketa automatiko hutsarekin lortutakoen oso antzekoak (desberdintasuna, hain zuzen, ez da estatistikoki esanguratsua, McNemar testaren arabera; $p < 0,05$). Izan ere, komak berreskuratzeko egindako CG erregelek oso estaldura apala zuten 1 klasean (% 27,2; ikus IV.2 taula).

Hala eta guztiz, kate- eta perpaus-identifikatzaile *komadun*ak darabiltzan ikasketa automatikoko algoritmoak erregela bidezkoekin konbinatzean, lortzen den hobekuntza estatistikoki esanguratsua da, McNemar testaren arabera ($p < 0,05$), bai erregelekin soilik lortutakoekin konparatuta, bai ikasketa automatikoko teknikekin soilik lortutakoekin erkatuta.

Etorkizunean, dena dela, komak berreskuratzeko erregela multzo osoago bat egiten probatu nahi genuke, emaitzak modu horretan gehiago hobetu ote ditzakegun aztertzeko.

### IV.6.4 Jatorrizko komen eragina saihesten

Orain arte azaldutako proba guztietan, informazio linguistikoa lortzeko, *Eustagger* analizatzaile/desanbiguatzaile morfosintaktikoa, *Ixati* zatitzailea, CG erregela bidezko perpaus-mugatzailea eta ikasketa automatikoko kate- eta perpaus-identifikatzaileak erabili genituen. Tresna hauek, normala den bezala, jatorrizko komak ere erabiltzen dituzte ahalik eta emaitza onenak lortu ahal izateko, eta horretan datza arazoa: besteak beste, *Eustagger*-ek komak erabiltzen ditu analisi morfosintaktikoak lortu eta desanbiguatzeko, eta gero guk informazio linguistiko hori bera erabiltzen dugu koma okerrak detektatu eta zuzenak jartzeko. Jatorrizko komak egokiak baldin badira, lagungarria izan daiteke koma horiek erabiltzea; alabaina, komak gaizki jarrita baldin badaude (maila baxuko euskara-ikasleen testuekin ari bagara, esaterako), koma hauekin lortzen den informazio linguistikoaren kalitatea okerragoa izango da ziur aski, eta, ondorioz, baita lortuko dugun azken emaitza ere.

Arazo hau aztertzeko, bi bide erabili genituen:

- Lehenengoan, komak zuzentzeko sailkatzailea sortzeko generabilen ikasketa- eta test-corpusari jatorrizko komak kendu genizkien; honela, bai *Eustagger*-en, bai *Ixati*n, komak zerabiltzaten erregelak baliogabetuta geratzen ziren. Eredu estokastikoan oinarritutako desanbiguatzailea,

ordea, bere horretan utzi genuen; alegia, bere ikasketa-corpuseko komak ez genituen kendu.

- Bigarrenean, berriz, komak zuzentzeko sailkatzailea sortzeko ibilitako ikasketa- eta test-corpusetik jatorrizko komak kentzeaz gain, komaren informazioa baliatzen ez duen desanbiguatzailea sortu eta erabili genuen.

Banan-banan aztertuko ditugu bi aukerak.

#### IV.6.4.1 Corpus komagabea eta desanbiguatzaile *komaduna* erabiliz

Lehen aukera honetan, esan bezala, komak darabiltzan desanbiguatzailea erabili genuen, baina corpus komagabearekin. Beste modu batean esanda, komak ere aprobetxatzen dituen desanbiguatzailearen emaitza onak aprobetxatu nahi genituen, baina corpusean zetozen komak —okerrak izan zitezkeenez— kontuan hartu gabe. *Eustagger*-ek desanbiguazioko ikasketa-prozesuan komak baliatzen dituenez, komarik gabeko corpus bat desanbiguatzean ezin daitezke emaitza onak espero, baina koma okerrak erabiltzean lor daitezkeenak baino hobeak izan zitezkeela pentsatu genuen; egileak jarritako unean uneko komen mende ez egotea lortu nahi zen. Gainera, modu honetan, komak zerabiltzaten zatitzailearen eta mugatzailearen CG erregelak ez ziren aplikatzen.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Corpus komaduna +** ***Eustagger komaduna*** | 96,5 | 98,8 | 97,6 | 77,8 | 55,0 | 64,4 |
| **Corpus komagabea +** ***Eustagger komaduna*** | 94,9 | 99,0 | 96,9 | 71,5 | 31,3 | 43,5 |

Taula IV.16: Koma-zuzentzailearen emaitzak (garapen-corpusean kalkulatuak), corpus komadun ala komagabeak erabiltzearen arabera (ohiko *Eustagger komaduna*k emandako informazioa baliatuta eta CG erregelak eta ikasketa automatikoko teknikak uztartuz (kate- eta perpaus-identifikatzaile komagabeak erabilita)).

IV.16 taulan ikus daitekeen moduan, emaitza kaskarrak lortu genituen (ikus 1 klasearen emaitzak). Dirudienez, alde batetik, komaren informazioa baliatuz lortutako analizatzailea ez dabil hain fin komaren informaziorik

gabe; eta, bestalde, komaren erabilera, antza, garrantzitsua da, ez soilik analizatzailerako, baita ondoren datozen urratsetarako ere.

B eranskineko B.2 atalean eta bertako B.2 irudian laburbildu ditugu, hain zuzen, *Eustagger komadun*a eta corpus komagabea erabiliz koma-zuzentzailea lortzeko egindako urratsak.

### IV.6.4.2 Komarik gabeko analizatzailearen erabilpena

Ikusirik IV.6.4.1 atalean lortzen ziren emaitza kaskarrak, koma-zuzentzaile bat garatzeko baliatzen den informazio linguistikoa lortzeko, komarik ez darabilen desanbiguatzailea garatzea eta erabiltzea deliberatu genuen. Modu honetan, komak ikasteko behar genuen informazio linguistiko guztia komak kontuan hartu gabe erdiesten zela ziurtatzen genuen. Hala, komak zuzentzeko sailkatzailea komarik gabeko corpus bati aplika geniezaiokeen, ikasketarako erabilitako baldintza berdinetan.

Beraz, corpusa analizatzeko eta desanbiguatzeko tresna *berezi* bat prestatu genuen, *Eustagger*-en egilearen laguntzarekin (*Eustagger komagabe*a deitu geniona): komak kontuan hartzen ez dituen analizatzaile/desanbiguatzailea, hain zuzen. Modu honetan, komak ikasteko prozesuan, *Eustagger komagabe*aren bidez egiten da corpusaren analisia eta desanbiguazio-prozesua (ikus B eranskineko B.3 irudia).

Analizatzaile *berezi* hori inplementatzeko, *Eustagger*-ek komak noiz eta zein zentzutan erabiltzen dituen aztertu behar izan genuen.

IV.6.1.8 atalean azaldu dugun moduan, *Eustagger*-ek, batez ere, desanbiguazio-prozesuan erabiltzen ditu komak. Desanbiguazio-prozesua bi mailatan egiten da: lehenengoan, CG erregelak erabiltzen dira desanbiguazioa fintzeko; bigarrenean, eredu estokastiko bat erabiltzen da analizatzaileak desanbiguatzen ikas dezan. Erregelei dagokienez, batzuek koma kontuan hartzen dute; estokastikoari dagokionez, berriz, ikasketa egiterakoan komadun corpusa baliatzen da. Hori dela eta, bi urrats hauek moldatu behar izan genituen gure analizatzaile berezia inplementatzeko.

Erregelen kasuan, bide sinple bat aukeratu zen: desanbiguazio-gramatikako erregeletatik koma erabiltzen zutenak kentzea, nahiz eta jakin, horrela, analizatzailearen emaitzek okerrera egingo zutela. Gainera, esan bezala, komak dituzten erregelak kendu beharrean, corpusari komak kentzea nahikoa litzateke horretarako; izan ere, desanbiguazio-gramatika aplikatzean ez lirateke koma kontuan hartzen duten erregelak aplikatuko. Baiki, horixe izan zen egin genuena.

Analisiaren emaitzan honek zenbateko eragina izan lezakeen neurtzea erraza ez den arren, zenbaki batzuek lagun lezakete: desanbiguazio-gramatikaren 2055 erregeletatik 220 erregelatan erabiltzen da koma (% 11). Komak darabiltzaten erregelak kendu beharrean, hauek moldatzea litzateke, agian, soluzio hobea; hori, ordea, lan zaila da inondik inora, eta ez da tesi-lan honen esparruan sartzen. Hala, esan bezala, corpusetik komak kendu eta komak darabiltzaten erregelak —baita zatitzailearenak eta mugatzailearenak ere— baliogabetuta geratu ziren[21].

Estokastikoaren kasuan, corpus komagabearekin berriz entrenatzea zen soluzioa. *Eustagger*-ek darabilen ikasketa-corpusari komak kendu eta horrekin entrenatu genuen estokastikoa, eta gero, desanbiguazio-prozesuan txertatu genuen.

Bi urrats horiek eman ondoren, prest geneukan *Eustagger komagabe*a: komak zuzentzeko prozesuan erabiltzeko moduko analizatzaile/desanbiguatzaile automatiko berezia, komak ezertarako erabiltzen ez dituena. Ohiko analizatzailearekin —*Eustagger komadun*arekin— alderatuta, *Eustagger komagabe*aren errore-tasa % 6 inguru handiagoa da, eta komen ikasketa-prozesuan honek eragina izango zuela aurreikusi genuen.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Corpus komaduna +** ***Eustagger komaduna*** | 96,5 | 98,8 | 97,6 | 77,8 | 55,0 | 64,4 |
| **Corpus komagabea +** ***Eustagger komaduna*** | 94,9 | 99,0 | 96,9 | 71,5 | 31,3 | 43,5 |
| **Corpus komagabea +** ***Eustagger komagabea*** | 95,0 | 98,8 | 96,9 | 69,3 | 33,3 | 45,0 |

Taula IV.17: Koma-zuzentzailearen emaitzak (garapen-corpusean neurtuak), analizatzaile/desanbiguatzaile komadunarekin edo komagabearekin ateratako informazio linguistikoa baliatuz; CG erregelak eta ikasketa automatikoko teknikak uztartuz (kate- eta perpaus-identifikatzaile komagabeak erabilita).

IV.17 taulan ikus daitezke *Eustagger komagabe*arekin lortutako informazioarekin ikasitako sailkatzaileak lortzen dituen emaitzak, orain artekoekin

[21]Hala eta guztiz ere, desanbiguatzaile, zatitzaile eta mugatzaile ahalik eta onenak izatea komeni zaigunez, etorkizunean, kendutako erregela horiek ordezkatzea aurreikusten dugu. Modua aurkitu beharko da, komaren bitartez adierazten zena, beste elementu linguistiko batzuen bidez adierazteko.

konparatuta (gainerako baldintza guztiak berdinak izanik bi kasuetan).

Pentsatu bezala, *Eustagger komagabe*aren portaera okerragoa izateak badu eragina, eta ikasketarako beharrezkoa den informazioa biltzeko egiten diren urrats bakoitzak aurrekoaren informazioa darabilenez, *Eustagger komagabe*aren errore-tasa handiagoak, besteak beste, ia 20 puntuko galera sortzen du azkenerako. Hala ere, *Eustagger komagabe*arekin lortzen ditugun emaitzak hobeak dira, *Eustagger komadun*ari (test gisa) corpus komagabea pasata lortzen zirenak baino. Ikasketa automatikoan sarri ikusten den moduan, izan ere, emaitzek okerrera egiten dute ikasketan mota bateko informazioa erabiltzen bada eta testa egitean beste bat.

B eranskineko B.3 irudian laburbildu ditugu, hain zuzen, *Eustagger komagabe*a baliatzen duen koma-zuzentzailea lortzeko egindako urratsak. Irudi horretan ikus daitekeenez, testua analizatzea eta desanbiguatzea da lehen urratsetariko bat; hori dela eta, atzetik datozen prozesu guztietan, *Eustagger komagabe*aren gabeziek geroz eta garrantzi handiagoa hartzen dute.

### IV.6.4.3 Adibideen azterketa

IV.6.4.2 eta IV.6.4.1 ataletan azaldutako soluziobideek emandako emaitza txarrak direla eta, corpus eta analizatzaile/desanbiguatzaile *komadun*ak edota *komagabe*ak erabiliz lortutako koma-zuzentzaileen emaitzak konparatu nahi izan genituen, adibide konkretuetan batzuek eta besteek zeukaten portaera aztertzeko.

Lau adibide hauetan IV.17 taulako hiru aukerek duten portaera aztertuko dugu (aukera bakoitza letra banarekin izendatu dugu, azalpenak errazteko asmoz):

- Corpus komaduna + *Eustagger komadun*a (A aukera)
- Corpus komagabea + *Eustagger komagabe*a (B aukera)
- Corpus komagabea + *Eustagger komadun*a (C aukera)

**Adibidea IV.6.1**

1. *Azken hiru hilabeteetan janaria erosteko dirua bakarrik ematen zietela salatu dute etorkinek, eta euren egoera salatuz gero beren kanporaketa bultzatzeko mehatxua egin zietela enpresaburuek.*
2. *Ez du, ordea, aipatu beste delinkuentzia mota hau.*
3. *Besteak beste, Hitchcock, Godard, Wilder eta Stanley Donenen zenbait maisu lan erakutsiko dira bertan.*

4. *Volker Schlondorff, berriz, Alemaniako zinema garaikidearen bultzatzaile nagusietakoa.*

Lehenengo adibidean, dagoen koma zuzen bakarra ondo identifikatu dute A aukerak eta C aukerak; B aukerak, ordea, ez. A aukerak koma hori identifikatzea logikoa da; izan ere, komaren ondoren datorren "eta" hitzean esaldi-hasierako etiketa dauka, esaldi- eta perpaus-mugen CG gramatikak emana, bere informazioaren artean. B eta C aukerek ez dute informazio hau, CG gramatika horrek komaren informazioa baliatzen duelako esaldi-muga horiek jartzerakoan: hau da, "koma + juntagailua" patroia topatzen duenean, esaldiaren hasiera-marka jartzen dio juntagailuari. B eta C aukeretan, ordea, ez dugu komarik corpusean, eta beraz esaldi- eta perpaus-mugen CG gramatikak ez dio etiketa hori jarriko. Hala, zailagoa izango dute hauek koma hau identifikatzea. Hala eta guztiz, C aukerak koma hau identifikatzea lortzen du. Ikasketa automatikoan askotan gertatzen den moduan, arrazoia ez dago oso argi. B eta C aukeren arteko informazioaren desberdintasun nagusia "euren" hitzak duen perpaus-etiketetan dago (erabili den (-5,+2) leihoa kontuan hartuta, betiere). B aukeran, perpaus baten hasiera dator adierazia; C aukeran, berriz, bi perpausen hasiera. Badirudi etiketa honek bideratzen duela aukera bat eta bestea juntagailuaren aurreko hitzari koma jartzera ala ez jartzera. Laburbilduz, adibide honetan, perpaus- eta esaldi-mugen CG gramatikak eta *FR-Perceptron* bidezko perpaus-identifikatzaileen portaera desberdinak baldintzatzen du erabakia.

Bigarren adibidean, dauden bi koma zuzenak identifikatu dituzte hiru aukerek, baina B aukerak sobran dagoen bat gehitu du "aipatu" hitzaren ondoren. Honen arrazoia "aipatu" hitzaren informazioaren artean, perpaus bat hasi eta bukatu dela dioen etiketa izan daitekeela uste dugu, etiketa hau ez baitute gainerako aukerek. Beraz, kasu honetan ere, *FR-Perceptron* bidezko perpaus-identifikatzaileen portaera okerrak eraman du tresna erabaki oker bat hartzera.

Hirugarren adibidean, A aukerak ondo jartzen ditu koma guztiak; B aukerari "Godard" eta "Wilder" hitzen arteko koma falta zaio eta C aukerari, berriz, "Godard" eta "Wilder" hitzen artekoaz gain, "Hitchcock" eta "Godard" artekoa ere falta zaio. Aukera bakoitzak darabilen informazioa aztertuz gero, kateen CG gramatikek egindako okerrak *FR-Perceptron* bidezko kate-detektatzailearen portaera okerra dakarrela ikusi dugu, eta honek, segur aski, koma horiek ez identifikatzea.

Laugarren adibidean, ostera, A aukerak bi komak ondo jartzen ditu, ziu-

rrenik *Eustagger*-ek asmatu egin duelako "berriz" hitzaren analisian, eta lokailua dela identifikatu duelako; informazio honekin komen CG gramatikak "berriz" hitzari eta aurreko "Schlondorff" hitzari koma bat dagokiela dioen etiketa esleitu die, eta informazio hau ziurrenik esanguratsua izan da bi hitz hauei koma bana esleitzean. B eta C aukeretan, ordea, analizatzaile/desanbiguatzaileak adberbiotzat hartu du "berriz" hitza, eta, beraz, komen CG gramatikak ez ditu A aukerarekin gehitutako etiketak erantsi. Hala eta guztiz ere, B aukeran ondo jarri ditu bi komak; C aukeran, aldiz, "berriz" hitzaren ondorengoa jartzen soilik asmatu du. Test-corpusean ez dugu desberdintasun honen arrazoirik aurkitu: sailkatzaile bakoitzak desberdin jokatu du kasu honetan, informazio bera izanagatik. Kontuan hartu behar da ikasketarako erabilitako informazioa desberdina dela batena eta bestearena: *Eustagger komagabe*a darabil B aukerak eta *komadun*a C aukerak, eta analizatzaile/desanbiguatzaile bakoitzak lortutako informazioan dagoen aldeaz gain, informazio hori baliatzen duten tresnek ere —ondorioz— lortzen duten informazio gehigarria desberdina izango da.

Adibide hauek aztertu ondoren, atera ditzakegun ondorioak hauek dira: batetik, analizatzaile/desanbiguatzaile bat ala bestea erabilita dagoen aldea handiegia ez den arren, lehen urrats honetan ematen diren akatsek beste errore batzuk dakartzatela; eta bestetik, analizatzaile/desanbiguatzaileak akatsak egin gabe ere, tarteko urratsetan aplikatzen diren CG gramatikek eta *FR-Perceptron* bidezko algoritmoek erroreak egiten dituztela corpus komagabea erabiltzen badugu. Eta akats batek beste bat dakarrenez, komaren zuzenketan eragina izatea dakar honek azkenerako.

### IV.6.5 Ebaluazio kualitatiboa

Atal honetan, komak ebaluatzeko egin dugun ebaluazio kualitatiboaren emaitzak aztertuko ditugu. *Eustagger komadun*ak, kate- eta perpaus-identifikatzaile *komagabe*ak eta komak zuzentzeko egindako CG erregelek emandako informazioa gehituta sortutako sailkatzailea ebaluatu genuen.

5.500 tokeneko test-corpusa harturik (gogoratu IV.6.1.2 atalean azaldutako corpusaren banaketa), bi hizkuntzalariri (hizkuntzalari1, hizkuntzalari2) eman genien —aurrez, corpusari, zeuzkan komak kenduta—, eurek komak etiketa zitzaten.

Test-corpusak berez zituen komak zuzentzat emanez, bi hizkuntzalarien etiketatzeak test-corpusaren komekiko zeukan *bateragarritasuna* neurtu genuen. IV.18 taulan ikus daitezke ohiko neurriak.

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Ikask. autom. *KPI-komagabearekin* + CG erregelak** | 95,6 | 98,5 | 97,1 | 77,6 | 52,7 | 62,8 |
| **Hizkuntzalari1** | 98,5 | 97,6 | 98,0 | 79,1 | 85,9 | 82,3 |
| **Hizkuntzalari2** | 97,5 | 97,4 | 97,5 | 76,1 | 76,4 | 76,3 |

Taula IV.18: Sailkatzailearen iragarpena eta hizkuntzalarien etiketatzearen emaitzak, test-corpuseko jatorrizko komekiko.

Aipatzeko modukoa iruditzen zaigu, baldintza berdinetan, test-corpusarekin makinak lortutako datuak (% 62,8) apalagoak direla orain arte erabili dugun garapen-corpusarekin alderatuta (% 64,4). Normala dena, bestalde, orain arteko doitze-prozesua garapen-corpusarekin egin baita; beraz, corpus hari hobeto egokitzea ez da harritzekoa.

Bestalde, deigarria da hizkuntzalarien emaitzak % 80 aren bueltan ibiltzea, eta baten eta bestearen artean sei puntuko aldea egotea. Neurri hauez gain, hizkuntzalari bakoitzaren eta sailkatzailearen arteko *kappa-neurriak* kalkulatu genituen; lehenengo hizkuntzalariaren erabakien eta test-corpuseko jatorrizko komen arteko *kappa-neurria* % 80,36koa da, eta bigarren hizkuntzalariaren eta test-corpuseko jatorrizko komen artekoa, % 73,74koa.

Bi hizkuntzalarien arteko adostasunaren datuak ere kalkulatu genituen (ikus IV.19 taula).

| | **0** | | | **1** | | |
|---|---|---|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ | **Doit.** | **Est.** | $F_1$ |
| **Hizkuntzalari1, hizkuntzalari2-rekiko** | 97,8 | 96,9 | 97,4 | 73,8 | 79,7 | 76,6 |

Taula IV.19: Hizkuntzalarien arteko adostasuna.

Bi etiketatzaileen arteko adostasuna neurtzeko *inter-tagger agreement (ITA)* delakoa ere erabiltzen da (Carletta, 1996): etiketatutako adibide guztien artetik, bi etiketatzaileak bat etorri diren adibideen portzentaia. Gure kasuan, % 95ekoa da neurri hau, baina datu hau ez da oso esanguratsua, zoriz asmatzeko portzentaia handia delako, 0 klaseko adibide askoz gehiago baititugu (komaz jarraitu gabeak, hain zuzen). Horrexegatik, emaitza-atributuko bi klaseen neurrien artean alde handia dago, IV.19 taulan ikus daitekeen

moduan: 0 klaserako emaitzak 1 klasekoak baino hobeak dira, orain arteko esperimentu guztietan ikusi dugun legez.

Halakoetan, *ITA* neurriari zorizko adostasun hori kentzeko, *kappa-neurri*a erabiltzea gomendatzen da (Carletta, 1996): klase bakoitzaren adibideen kopurua eta distribuzioa kontuan hartzen du neurri honek. % 74,02koa da bi hizkuntzalarien arteko *kappa-neurria*. *Kappa-neurri*aren balioak interpretatzeko irizpide anitz egonik ere, Carletta-k (1996) berak % 80tik gorako balioak jotzen ditu fidagarritzat; % 67 eta % 80 artekoak, berriz, zalantzagarriak direla dio.

Datu hauek erakusten digute komak berreskuratzearen ataza ez dela batere erraza. Bi hizkuntzalarien test-corpusarekiko bateragarritasun maila eta hizkuntzalari batek bestearekiko duena ikusita, 1 klasearen goi-muga % 76 ingurukoa dela esatera ausartzen gara. Hau da, koma-zuzentzaile automatiko baten *skyline* edo goi-muga % 76 ingurukoa dela uste dugu.

Hala eta guztiz ere, azken proba gisa, gure sailkatzailearen ebaluazio osoago bat egitea erabaki genuen. Izan ere, orain arte erabilitako neurriek ez dute adierazten komak automatikoki berreskuratzeko gaitasuna, corpuseko jatorrizko komekiko zuzentasuna baizik; hau da, egileak jarri dituen komekiko bateragarritasuna. Beste kontu bat da, ordea, orain arte hala suposatu dugun arren, egileak jarri dituen komak zuzenak izatea; gainera, egileak jarri dituen komak zuzenak izanik ere, koma-konbinazio zuzen posible bat baino gehiago izan daitezke esaldiko; alegia, sailkatzaileak jarri dituen komak, jatorrizkoekin bat etorri ez arren, zuzenak izan daitezke, eta alderantziz: sailkatzaileak jarritako komak, egileak jarritakoekin bat etorriagatik, okerrak izan daitezke.

Hau guztia dela eta, ebaluazio kualitatibo osoago bat egitea erabaki genuen. Egiazki, bi ebaluazio mota egin genituen: bata, tokenka, eta bestea, esaldika; alegia, lehenengo ebaluazioan, token bakoitzaren ondoren koma zihoan ala ez neurtu genuen —orain arteko esperimentuetan bezala—; bigarrenean, berriz, esaldiko koma guztiak ondo zeuden ala ez neurtu genuen (esaldi baten baitan koma bat ondo jartzeak baina hurrengoa gaizki, esaldiaren zentzua erabat alda dezakeelakoan, Shieber eta Tao (2003) lanean ikusi dugun moduan).

Ebaluazio kualitatibo honetan —bai tokenka egindako ebaluazioan, bai esaldika egindakoan—, hiru soluzio ematen ziren ontzat: corpus originalekoa eta bi hizkuntzalarietako bakoitzarena. Hala, sailkatzaileak jarritako komak hiru soluzio horiekin konparatzen ziren, tokenka lehendabizi, esaldi osoa kontuan harturik gero.

Tokenka egindako ebaluazioan, sailkatzaileak jarritako koma bakoitza az-

tertzen zen, alde batetik: koma hori bi hizkuntzalarietako batek jarri bazuen edo corpuseko jatorrizkoan baldin bazetorren, ontzat ematen zen; bestalde, hiru soluzioetan errepikatzen zen koma bakoitza beharrezkotzat jotzen zen, eta sailkatzaileak jarri ez bazuen, okertzat ematen zen.

Tokenka egindako ebaluazioaren emaitzak IV.20 taulan ikus daitezke. Emaitza hauek esperantzagarriak iruditzen zaizkigu; izan ere, sailkatzaileak jarritako bost kometatik, lau ondo egon daitezkeela esan nahi du doitasunaren neurri honek (% 83,01).

| | **1** | | |
|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ |
| **Ikask. autom. *KPI-komagabearekin* + CG erregelak** | 77,6 | 52,7 | 62,8 |
| **Ebaluazio kualitatiboa, tokenka** | 83,01 | 58,46 | **68,61** |

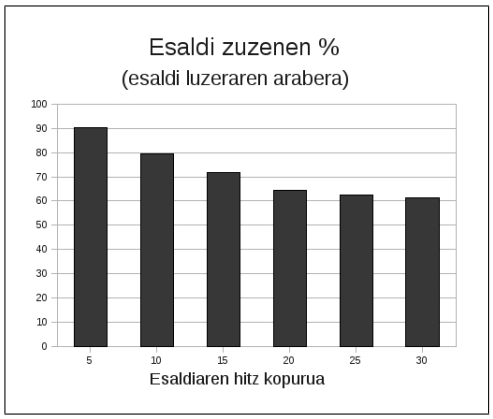
Taula IV.20: Sailkatzailearen bateragarritasuna test-corpusarekiko, eta tokenka egindako ebaluazio kualitatiboaren emaitzak.

Esaldika egindako ebaluazioan, esaldi bat zuzentzat jotzen zen, esaldiko koma guztiak hiru soluzioetako batekin beren osoan bat baldin bazetozen; alegia, ez bazuen komarik, ez sobran, ez faltan (soluzio baten eta bestearen komak nahastu gabe). 380 esaldietatik, 219 esaldi etiketatu ziren koma zuzen guztiekin (% 57,63); komarik gabeko esaldiak kenduz gero, berriz, ia erdira jaitsi zen neurri hau: 244 esaldi komadunetatik, 83 esaldi etiketatu ziren koma zuzen guztiekin (% 34,02).

Pentsaturik, bestalde, esaldiaren luzerak eragina izan lezakeela ataza honen zailtasunean, neurri hauen beste ikuspegi bat izateko, hitz kopuruaren araberako emaitzak kalkulatu genituen. Tokenka egindako neurketetan halako desberdintasunik nabaritzen ez bada ere, esaldika egindako ebaluazioan argi ikusten da esaldiaren luzera handitzearekin okerrera egiten dutela emaitzek ere (ikus IV.6 irudia).

### IV.6.6 Erroreen analisia

Ebaluazio kualitatiboa osatzeko, aurreko atalean ebaluatutako sailkatzailearen portaera aztertu genuen, honek zuzen eta oker etiketatutako test-corpuseko esaldiak behatuz. Jarraian, adibide argigarri batzuk aztertuko ditugu.




Irudia IV.6: Hiru soluzio posibleekin konparatuz kalkulatutako esaldi zuzenen proportzioak, X baino hitz gutxiagoko esaldiak kontuan hartuz (x=5, x=10, x=15...).

"&KOMA" etiketak esan nahi du sailkatzaileak koma jartzea erabaki eta asmatu egin duela; "&FALTAN" etiketak, berriz, sailkatzaileak ez duela toki horretan komarik jarri, baina koma behar zuela, betiere hiru erreferentzien arabera; azkenik, "&SOBRAN" etiketak esan nahi du sailkatzaileak koma jarri duela, komarik behar ez zen tokian.

**Adibidea IV.6.2**
Sailkatzaileak guztiz zuzen etiketatutako esaldiak:

1. *Izan ere&KOMA Iztuetak berak prentsa oharrean adierazi zuenez&KOMA autonomia erkidegoetako Hezkuntza sailburuek bilera egin zuten iragan astelehenean&KOMA Del Castillo ministroarekin&KOMA eta hark ez zien testu berriaren inguruan hitzik ere esan.*
2. *EAEko Sergurtasun Batzordeak hartutako neurriak eta Eudelen mozioa aztertzeko Juan Jose Ibarretxe lehendakariak hil amaierarako deitu duen bileran izango dela ziurtatu zuen atzo PPk&KOMA nahiz eta bezperan uko egin zion segurtasun batzordeak Arkauten eginiko bilerara joateari.*
3. *Bertzalde&KOMA erran du elkarrizketa berean Txillardegik euskararen gaitzen iturburua erdaldunak direla&KOMA hots&KOMA euskara ikasi nahi ez dutenak.*
4. *Cabrera&KOMA berriz&KOMA Kataluniako Generalitateak sortutako Pompeu Fabra eta Universitat Oberta de Catalunya unibertsitate elebakarrak sortzerakoan aholkulari gisa jardundakoa da&KOMA eta aukera legalak aztertuko ditu bere hitzaldian.*

Mota ezberdineko komak zuzen jartzeko gai da sailkatzailea, IV.6.2 adibideko esaldietan ikus daitekeen moduan. Esaterako, lokailuei dagozkien

komak zuzen jarri ditu lehen, hirugarren eta laugarren esaldietan. Aditz nagusiaren aurreko mintzagaiari —perpausa denean— dagokion koma zuzen jarri du, bestalde, lehen esaldian. Aditz nagusiaren ondorengo perpausa banatzen ere asmatu du bigarren esaldian. Laugarren esaldian, ostera, esaldiak banatzen asmatu du, "koma + juntagailua" egitura erabiliz.

**Adibidea IV.6.3**
Sailkatzaileak komaren bat sobran jarritako edo faltan utzitako esaldiak:

1. *Gurean igandeko egunkariek aste osoa ematen dute komunetik&SOBRAN bueltaka eta jiraka&FALTAN orain toalleroan&FALTAN orain erradiadorean.*
2. *Pasa ziren egiazko ospakizunak eta itxurazkoak&FALTAN etorri ziren lehen adierazpenak eta hasierako azterketak&KOMA argazkiak eta gezur ezkutatuak.*
3. *Batzuen ustez&KOMA inora ez doana&KOMA ezer egiten ez duena&KOMA eta beste batzuentzat&FALTAN gehiegi egiten duena&KOMA urrutiegi eta arinegi doana.*
4. *Haien izenak esan ondoren&FALTAN batzordea osatuta geratu zen&KOMA eta ondoren&KOMA alderdietako ordezkariek hartu zuten hitza&KOMA Aitor Gabilondo EA-EAJko alkatea lehenengoa izanik.*
5. *Sobieten Iraultzarekin hasi eta Berlingo Harresiaren birrinketaz bukatu zuen mendetik denboraren abiadura gero eta azkarragoa denez&KOMA epealdi laburrak sekulako garrantzia hartzen omen du &FALTAN Historiaren aro luzeetan orain arte suertatu ohi diren eraldaketa mantsoak gaindituz...*
6. *Gaur onetsiko diot hola dukeela Gipuzkoan edo Nafarroan (kendu Tutera eta bere ingurua)&KOMA baina ez&FALTAN ordea&KOMA Araba-Bizkaietako sartaldeko lurraldeetan (Enkarterri&KOMA Gobiaran...)&FALTAN ezen&SOBRAN alderdi horiek bizitegi ukan zituzten autrigoien hizkuntza zein izandu zen zehaztasunez ez dakigun arren&KOMA badakigu hizkuntza hori galdu ondoan ez zuela euskarak ordezkatu&KOMA latinak lehendabizi eta erromantzeak geroago baizik&KOMA alegia&FALTAN egun gaztelania deritzon erromantzeak&FALTAN gehiegitxo sinplifikatzea zilegi izan balekit.*
7. *UEUk&FALTAN EIREk&FALTAN Euskal Adarrak&KOMA Barrutiak eta Uniekimenak "eztabaidaren erdigunean" jarri nahi dute aldarrikapena.*

Aitzitik, IV.6.3 adibideko esaldietan, akatsen bat egin du beti sailkatzaileak. Kontuan hartu behar da ezen aipatutako hiru soluzioetako bakoitzarekin bat ez datozenak hartu direla akats gisa ebaluazio kualitatiboan; hau da, koma bat sobran dagoela jarri dugun tokian, koma hori ez dago hiru soluzioetako ezeinetan; eta faltan dagoela jarri dugun tokian, hiru soluzioek beharrezkotzat jotako koma da. Akats horiek aztertuko ditugu segidan, esaldiz esaldi.

Lehen esaldian, "*Gurean*" mintzagaiaren ondoren koma bat jartzea zilegi litzatekeen arren, ez jartzea ere ontzat eman da ebaluazio kualitatiboan. "*Komunetik*" hitzaren ondoren, sailkatzaileak jarritako koma sobran dago (hau ere zalantzazkoa dela esan genezake, "*bueltaka eta jiraka*" tarteki gisa uler baitaiteke). Falta direla markatutako bi komak, ordea, ez dira zalantzagarriak, inondik ere.

Bigarren esaldian, esaldiko elementuen ordena ez hain ohikoak eragiten du, gure ustez, falta den komaren akatsa. Komaren ordez ere, jar liteke puntuazio-marka gogorragoren bat (puntu eta koma, adibidez). Bestalde, informazio semantikorik gabe oso zaila deritzogu faltan dagoen koma hori detektatzeari; izan ere, esaldia zentzu honetan ere har liteke: "*Pasa ziren egiazko ospakizunak, eta itxurazkoak etorri ziren...*". Honez gain, aipatu beharra dago esaldi-bukaerako enumerazioari dagokion koma ondo jartzen duela sailkatzaileak.

Hirugarren eta laugarren esaldiak esaldika egindako ebaluazioa zalantzan jartzeko erakutsi ditugu. Izan ere, hirugarren esaldian lau koma zuzen jartzen asmatu du sailkatzaileak, eta koma bakarra falta zaiola ebatzi da ebaluazio kualitatiboan; laugarrenean, berriz, hiru zuzen eta bat faltan. Esaldikako ebaluazioan, txartzat joko litzateke esaldi hau, akats bat duelako. Dena dela, jarritako komak zuzenak izanik, esaldi osoa txartzat jotzea gehiegizkoa dela iruditzen zaigu, kasu hauetan, behintzat. Honekin azaleratu nahi dugun arazoa ebaluazioarena da: kasu batzuetan, koma bakar bateko akatsak esaldi guztia desitxura dezake, baina besteetan, asmatutakoak baliagarriak izan daitezke eta, beraz, ebaluazioan modu positiboan adierazita agertu beharko lirateke. Esaldi mailako ebaluazioa eta token mailakoa, biak egin beharko liratekeela uste dugu, beraz, komen ebaluazio on bat osatzeko.

Bosgarren esaldian koma bat zuzen markatu du sailkatzaileak, eta beste bat falta zaiola ebatzi du ebaluatzaileak. Falta zaion hau, berriz ere, semantikaren ezagutzarik gabe detektatzea ezinezkoa iruditzen zaigu, zentzu handirik ez duen baina sintaktikoki zuzena den modu honetan har baitaiteke esaldia: "*epealdi laburrak sekulako garrantzia hartzen omen du Historiaren aro luzeetan, orain arte suertatu ohi diren eraldaketa mantsoak gaindituz...*".

Seigarren esaldia adibide konplikatu gisa ekarri dugu: 60 hitz baino gehiagoko esaldia, parentesiekin markatutako bi tarteki dituena eta mendeko perpaus nahasiekin idatzia izan dena. Hala eta guztiz ere, sailkatzaileak jarritako sei koma onetsi dira ebaluazioan, eta lau faltan dituela eta bat sobran ebatzi da, horretaz gain.

Bukatzeko, enumerazioko kometan gertatzen dena ikusteko ekarri dugu zazpigarren esaldia. Enumerazioko azken hirugarren elementua komaz markatzen du sailkatzaileak, baina ez da gai enumerazioko gainerakoak (azken hirugarrenaren aurrekoak) komaz markatzeko. Test-corpusean gehiagotan gertatu den fenomeno honek joera bat erakusten digu: badirudi hiru elementuko enumerazioak ezagutu eta komaz mugatzeko gai dela sailkatzailea, baina ez gehiagokoak. Ikasketan erabilitako leihoan egon liteke honen arrazoia: gogora dezagun (-5,+2) leihoa erabili dugula eta horrek token bakoitzaren ondorengo bi tokenen informazioa soilik kontuan hartzea dakarrela. Leihoan tokenaren ondorengo hiru edo lau tokenen informazioa hartzeak arazo hau konpondu lezakeela uste badugu ere, beste batzuk sor ditzakeela ere pentsatzekoa da. Hau guztia baieztatzeko azterketa sakonagoa egin beharko litzateke. Dena dela, gauzak dauden moduan utzita ere, arazo honen aurrean nahikoa litzateke azken hirugarren elementuari sailkatzaileak jartzen dion koma hori baliatzea, enumerazioetan komak nola jartzen diren azaltzeko erabiltzaileari.

Bestalde, hizkuntzalariek egindako etiketatze-lan honetan, bi gauza azpimarratu dizkigute, hemen esatekoak iruditzen zaizkigunak:

1. Esaldi batzuk anbiguoak direla, eta ez dela erraza komak non doazen jakitea.

2. Batzuetan, komak baino, puntuak edo puntu eta komak erabili nahi izan dituztela, etiketatzean.

Esaldiko elementuen ordenamendu eskasak —edo euskara batuaren ordenamendu normalenetik urruntzeak—, esaldiaren konplexutasunak, anbiguotasun semantikoak eta gainerako puntuazio ikurren erabilera ez beti zuzenak, besteak beste, komak berreskuratzearen ataza zaila eta konplexua bilakatzen dute.

## IV.7 Ondorioak

Puntuazioa, duela urte batzuk arte, ez da serioski kontuan hartua izan HPan. Hizkuntzaren ulermen osoa, ordea, ezinezkoa da puntuazioa kontuan hartu gabe. Hala ere, arazo handiak daude puntuazioa jorratzeko, hizkuntza gehienetan puntuazio-arauak ez daudelako oso finkatuta, eta komari dagozkionak are gutxiago. Normalizazio prozesu berantiarra izan duen euskararen moduko hizkuntza batean, gainera, handitu egiten dira arazo hauek.

Hala eta guztiz ere, puntuazio-zuzentzaile bat —koma-zuzentzaile bat, batez ere— sortzeko ahaleginak egin ditugu. Koma da puntuazio-marketan tratatzen zailena, gehien erabilia izateaz gain, erabilera desberdin asko dituelako. Hala, lehendabiziko urrats gisa, komaren erabilerari buruzko azterketa teorikoa egin behar izan dugu; hauxe izan da komaren arauak formalizatzeko eta ikasketa automatikoan erabili beharreko ezaugarri linguistikoak identifikatzeko lehen urratsa. Komaren erabileran, ingelesaren eta euskararen artean antzekotasun handiak daudela ikusi dugu, bestalde, bien arteko konparazioa landu dugunean.

Komaren erabilera formalizatu ostean, ikasketa automatikoko teknikak baliatu ditugu, batez ere, komak zuzen jartzen ikasteko. Ikasketa automatikoa egiteko, ez dugu etiketatze-lanik egin behar izan; erabili ditugun corpusetan komak ondo jarrita zeudela suposatu dugu. Horrek bere abantailak eta desabantailak ditu: abantailen artean, etiketatze-lanean denborarik eman behar ez izatea daukagu; desabantailen artean, komak hala-moduz jarrita egon daitezkeela, kontrakoa suposatu arren. Komak ondo jarrita ez egoteak, koma *txarrak* —zuzenak ez direnak— ikasten aritzeko arriskua izan dugu. Arrisku honen benetakotasuna juzkatzeko, eskuzko etiketatzea eta ebaluazio kualitatiboa egin ditugu.

Hizkuntza-ezagutzan oinarritutako teknikak ere erabili ditugu, eta baita ikasketa automatikoko teknikekin uztartu ere. Kasu honetan, kate- eta perpaus-identifikatzaile *komagabe*ak aplikatu ondoren lortu dugun hobekuntza ez da estatistikoki esanguratsua izan ($p < 0,05$). Erregelen 1 klaseko estaldura txarrari (% 27,2) egozten diogu honen errua. Etorkizunean, komak zuzentzeko, erregelak hobetzea aurreikusten dugu, uztarketa honek fruitua eman dezan.

Erregelek emandako informazioaz gain, askotariko informazio linguistikoa baliatu dugu koma-zuzentzailea hobetzen saiatzeko. Ahotsaren ezagutzarako komak berreskuratzeko Shieber eta Tao-k (2003) egindako lanari jarraiki, kateen eta perpausen informazioa ahalik eta onena lortzen ahalegin berezia egin dugu III. kapituluan. Izan ere, Shieber eta Tao-k (2003) frogatu zutenez, osagaien mugen informazioa oso baliagarria da komen kokalekua asmatzeko; alegia, token bat geroz eta osagai gehiagoren muga izan, token horren inguruan koma bat izateko orduan eta probabilitate handiagoa dela frogatu zuten. Hala, euskarako kate- eta perpaus-identifikatzaile automatiko konpetitiboak eraiki ditugu, eta hauek emandako informazioa komaren ikasketa-prozesuan arrakastaz txertatu dugu: zazpi eta hamar puntu arteko hobekuntzak lortu ditugu (kate- eta perpaus-identifikatzaile *komagabe*ak edo *komadun*ak erabi-

| | 1 | | |
|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ |
| **baseline-200** | 12,1 | 42,7 | 18,9 |
| **Err** | 56,9 | 27,2 | 36,8 |
| **C4.5, -5+2** | 65,2 | 42,4 | 51,4 |
| **C4.5, -5+2, 300 atrib.** | 69,6 | 48,6 | 57,2 |
| **C4.5, -5+2, 300 atrib., KPI komagabearekin** | 76,6 | 55,7 | **64,5** |
| **Err + C4.5, -5+2, 300 atrib., KPI komagabearekin** | **77,8** | 55,0 | 64,4 |
| **C4.5, -5+2, 300 atrib., KPI komadunarekin** | 78,4 | 59,8 | 67,9 |
| **Err + C4.5, -5+2, 300 atrib., KPI komadunarekin** | 79,0 | 61,4 | **69,1** |
| **Err + C4.5, -5+2, 300 atrib., KPI eta *Eustagger* komagabeekin** | 69,3 | 33,3 | **45,0** |
| **Err + C4.5, -5+2, 300 atrib., KPI komagabearekin, koma-ez** | 71,5 | 31,3 | 43,5 |

Taula IV.21: Euskarako koma-zuzentzailearen emaitzen laburpena (zenbait parametroren arabera) garapen-corpusean neurtuta (C4.5: erabaki-zuhaitzak; -5+2: leihoa; 300 atrib.: komaz jarraituak maizen agertzen diren 100 hitz, 100 *bigram* eta 100 *trigram* atributu gisa jarrita; KPI *komagabe*arekin edo *komadun*arekin: *FR-Perceptron* bidez lortutako kate- eta perpaus-identifikatzaile *komadun*aren edo *komagabe*aren informazioa gehituta; err: falta diren komak detektatzeko CG erregelen emaitza atributu gisa gehituta; koma-ez: atributuen informazioa corpus komagabearekin lortutakoa; 101.250 token inguruko ikasketa-corpusa erabiliz, eta, besterik esan ezean, *Eustagger* *komadun*arekin).

liz, hurrenez hurren). Modu honetan, gure hipotesia neurri batean zuzena dela frogatu dugu (ikus IV.21 taula): kate- eta perpaus-identifikatzaileak ematen diguten informazio linguistikoa esanguratsua da koma-zuzentzailerako (estatistikoki esanguratsua, McNemar testaren arabera; $p < 0,05$).

Aipatu bezala, kate- eta perpaus-identifikatzaile *komadun*ez gain, *komagabeak* ere sortu ditugu. Izan ere, tresna hauek ematen diguten informazioa komak ikasteko darabilgunez, kate- eta perpaus-identifikatzaileak hobetzeko koma horiek erabiltzea zalantzagarria da. Modu berean, gainerako informazio linguistikoa lortzeko darabiltzagun tresnetan ere zalantzagarria da jatorrizko komak kontuan hartzea. Arazo hau ebazteko, bi bide probatu genituen, IV.6.4 atalean azaldu bezala: batetik, jatorrizko komak kentzea corpusetik, gainerako informazio guztia lortu bitartean; bestetik, jatorrizko komak kentzeaz gain, *Eustagger komagabe*a inplementatu eta erabiltzea, zeinak ez baititu komak ezertarako kontuan hartzen. IV.21 taulako azken bi lerroetan ikus daitekeen moduan, asko jaisten dira emaitzak, modu honetan. Emaitza txar hauek testuaren egileak jarritako komak kontuan hartzera garamatzate derrigor. Aztertu beharko litzateke, dena dela, egileak komak gaizki jartzen baditu zer den gertatzen dena.

Etorkizunean, euskara-ikasleen testuekin egin nahi genuke proba, eurek jarritako komak mantentzea ala ez komenigarria den ikusteko. Ikaslearen mailaren arabera, erabaki desberdinak hartu beharko direla iruditzen zaigu, baina azterketa sakonago bat egin beharko litzateke uste hauek ziurtatzeko. Bitartean, komak zuzentzeko atazan, kasurik txarrenean lor daitezkeen emaitza gisa interpretatzen ditugu *Eustagger komagabe*arekin lortutako neurriak.

Dena dela, gauza batek asko baldintzatzen du ataza honen garapena: komarik gabeko esaldi batean, batzuetan, ezinezkoa da koma egokiak zein diren jakitea; izan ere, leku bat baino gehiagotan jar daiteke koma hori, esaldiari eman nahi zaion zentzuaren arabera. Esate baterako, har dezagun "*Andereñoa haserretu zaio klasean jende guztiaren aurrean izozkia jan duelako.*" komarik gabeko esaldia (Garzia, 1997), eta saia gaitezen koma zuzenak jartzen. Komak jartzen ditugun lekuaren arabera esaldiaren esanahia aldatzen denez, koma-zuzentzaile automatiko perfektu batek aukera zuzen guztiak eman beharko lituzke. Bi hauek, adibidez:

- *Andereñoa haserretu zaio klasean, jende guztiaren aurrean, izozkia jan duelako.*
- *Andereñoa haserretu zaio klasean, jende guztiaren aurrean izozkia jan duelako.*

Komarik gabeko esaldia baldin bada gure abiapuntua, ezin jakin daiteke esaldia idatzi zuenak zein zentzu eman nahi zion. Ebaluazio kualitatiboan, hain zuzen, kontuan hartu ditugu honelakoak.

Bi hizkuntzalarik eskuz etiketatu zituzten komak, komarik gabeko test-corpus batean. Gero, bi hizkuntzalari hauen etiketatzea ebaluatu genuen, jatorrizko komen arabera, eta sailkatzailearen iragarpenarekin konparatu genituen emaitzok. IV.22 taulan ikus daitekeen moduan, % 76ren bueltan dabil, kasurik txarrenean, hizkuntzalarien etiketatzea (1 klasean). Sailkatzaileak, berriz, kate- eta perpaus-identifikatzaile *komagabe*ek emandako informazioa baliatuta, ia % 63ko $F_1$ neurria lortzen du, jatorrizko komekiko.

| | **1** | | |
|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ |
| **Hizkuntzalari1** | 79,1 | 85,9 | 82,3 |
| **Hizkuntzalari2** | 76,1 | 76,4 | 76,3 |
| **Ikask. autom. KPI *komagabea*rekin + CG erregelak** | 77,6 | 52,7 | 62,8 |
| **Ebaluazio kualitatiboa, tokenka** | **83,01** | 58,46 | **68,61** |

Taula IV.22: Hizkuntzalarien eta sailkatzailearen emaitzak, test-corpuseko jatorrizko komekiko; eta tokenka egindako ebaluazio kualitatiboaren emaitzak, jatorrizko komekiko eta hizkuntzalariek jarritakoekiko.

Baina, esan dugun legez, ez da oso justua sailkatzailearen jarduna emaitza zuzen batekin soilik ebaluatzea. Izan ere, ebaluazio-modu honek corpuseko jatorrizko komekiko zuzentasuna neurtzen du; hau da, egileak jarri dituen komekiko bateragarritasuna. Beste kontu bat da, ordea, orain arte hala suposatu dugun arren, egileak jarri dituen komak zuzenak izatea; gainera, egileak jarri dituen komak zuzenak izanik ere, koma-konbinazio zuzen posibleak bat baino gehiago izan daitezke esaldiko, aipatu dugun moduan; alegia, sailkatzaileak jarri dituen komak, egileak jarritakoekin bat etorri ez arren, zuzenak izan daitezke, eta alderantziz: sailkatzaileak jarritako komak, egileak jarritakoekin bat etorriagatik, okerrak izan daitezke.

Hau guztia dela eta, ebaluazio kualitatibo osoago bat egitea erabaki genuen. Egiazki, bi ebaluazio mota egin genituen: bata, tokenka, eta bestea, esaldika; alegia, lehenengo ebaluazioan, token bakoitzaren ondoren koma zihoan ala ez neurtu genuen —orain arteko esperimentuetan bezala—; bigarrenean, berriz, esaldiko koma guztiak ondo zeuden ala ez neurtu genuen

(esaldi baten baitan koma bat ondo jartzeak baina hurrengoa gaizki, esaldiaren zentzua erabat alda dezakeelakoan, Shieber eta Tao (2003) lanean ikusi dugun eran).

Ebaluazio honetan, hiru soluzio ematen ziren ontzat: corpus originalekoa eta bi hizkuntzalarietako bakoitzarena. Hala, sailkatzaileak etiketatutako komak hiru soluzio horiekin konparatzen ziren, tokenka lehendabizi, esaldi osoa kontuan harturik gero.

Tokenka egindako ebaluazioan, sailkatzaileak jarritako koma bakoitza aztertzen zen, alde batetik: koma hori bi hizkuntzalarietako batek jarri bazuen edo corpuseko jatorrizkoan baldin bazetorren, ontzat ematen zen, eta, bestela, txartzat; bestalde, hiru soluzioetan errepikatzen zen koma bakoitza beharrezkotzat jotzen zen, eta sailkatzaileak jarri ez bazuen, okertzat ematen zen.

Esaldika egindako ebaluazioan, esaldi bat zuzentzat jotzen zen, esaldiko koma guztiak hiru soluzioetako batekin bat baldin bazetozen; alegia, ez bazuen komarik, ez sobran, ez faltan (soluzio baten eta bestearen komak nahastu gabe).

Esaldika egindako ebaluazioan, 380 esaldietatik, 219 esaldi etiketatu ziren koma zuzen guztiekin (% 57,63); komarik gabeko esaldiak kenduz gero, berriz, ia erdira jaitsi zen neurri hau: 244 esaldi komadunetatik, 83 esaldi etiketatu ziren koma zuzen guztiekin (% 34,02). Esaldika egindako ebaluazioa, hala ere, gogorregia iruditzen zaigu askotan. Izan ere, aurrerapen handia izan daiteke, esate baterako, esaldi bateko lau koma zuzen detektatzea, bat detektatu gabe utzita ere. Ikus daitekeen moduan, koma-zuzentzaileen ebaluazioa ez da kontu erraza.

Tokenka egindako ebaluazioan, emaitza hobeak lortzen dira, jakina (ikus IV.22 taulako azken lerroa). Emaitza hauek esperantzagarriak iruditzen zaizkigu; izan ere, sailkatzaileak jarritako bost kometatik, lau ondo daudela esan nahi du doitasun honek (% 83,01), betiere hiru erreferentzietako baten arabera. Tesi-txosten honen hasieran aipatu dugu estilo- eta gramatika-zuzentzaileek doitasunari garrantzi handiagoa eman behar diotela, erabiltzaileek errazago onartzen dutela errore bat detektatu gabe geratu izana, benetan errorea ez dena errore gisa hartzea baino. Doitasun honekin, koma-zuzentzaile erabilgarri bat lortu dugula esan beharrean gaude beraz, euskaraz zuzen samar idazten duen erabiltzaile batentzat zuzendua betiere. Gainera, kontuan harturik goi-muga edo *skyline* deritzona % 76 punturen bueltan dabilela (hizkuntzalarien etiketatzearen arabera), lortzen dugun $F_1$ = % 68,61 neurria ona dela esan genezake.

Esaldiaren luzeraren araberako neurriak ere atera genituen, eta frogatu genuen, esaldika egindako ebaluazioan, esaldiaren luzera handitzearekin okerrera egiten dutela emaitzek ere.

Etorkizunean, koma-zuzentzaile sendoago bat lortzeko, *Eustagger koma-gabe*arekin lortutako emaitzak hobetu beharko genituzke. Hala, edozein delarik ere testuen egilea, komak kentzea litzateke lehen pausoa, eta hauek ematen duten informaziorik gabe koma zuzenak berreskuratzen asmatu beharko luke sailkatzaileak. Hala lortutako emaitzak ez leudeke erabiltzailearen mende, eta geneukakeen koma-zuzentzailea, beraz, sendoagoa litzateke. Jatorrizko komak utzita lortzen diren emaitzak txarragoak direla ziurtatu beharko genuke, bide hau hartu aurretik.

Hobetzeko beste aukera bat, ikasteko informazio aberatsagoa erabiltzean datza. Gure ikasketa automatikoko sistemak ez ditu kontuan hartzen, uneko tokenarentzat, esaldiko bere aurreko tokenek koma duten ala ez. Alegia, uneko tokenari koma jarri ala ez erabakitzeko, (-5,+2) leihoa erabili dugun arren, inguruko token horien informazio linguistiko guztia erabiltzen du sailkatzaileak, token horiek koma duten ala ez izan ezik. Kasu batzuetarako, informazio hori lagungarria dela uste dugu, eta etorkizunean uneko tokenarentzat inguruko tokenek segidan komarik duten ala ez atributu gisa erabiltzea aurreikusten dugu. Horretarako, *ikasketa inkrementala* edo *on-line* ikasketa egin daiteke, II.1.3 atalean azaldu dugun moduan. HMM edo CRF algoritmoek, dituzten berezko ezaugarriengatik, egokiak dirudite modu honetako atazak ebazteko.

*SVM* ikasketa-algoritmoak atributu askorekin emaitza onak lortzen dituenez (Rehurek eta Sojka, 2010), erabili ditugunak baino atributu gehiago erantsita, proba gehiago egin litezke. Komaren aurretik maizen agertzen diren hitzen kopuru handiagoa erabil genezake, adibidez, edota leiho handiagoa.

Beste ikasketa-algoritmo bat baliatzekotan, *FR-Perceptron* bera ere erabil genezake, bi komen arteko *hitz multzo*ak identifikatzen saiatuz. Hala ere, egokiagoa litzateke emaitza-klaseen arteko desorekari aurre egiten dion algoritmo bat baliatzea. Izan ere, arazoak ekarri dizkigu corpusean komaz jarraituak (1 klasekoak) diren adibide gutxiago izateak, komaz ez jarraituak (0 klasekoak) baino. Horregatik lortu ditugu emaitza hobeak 0 klaserako. Arloaren literaturan ikusi dugun moduan, ordea, badira arazo hau ebazteko bideak. Aztertu ditugun batzuek SVMren eta pertzeptroien gisako algoritmoak moldatzen dituzte marjina aldagarriak baliatzeko (Li *et al.*, 2009, 2002b); alegia, bi klaseen arteko marjina handieneko hiperplanoa bilatu ordez, klase bakoi-

tzerako marjina bana bilatzen dute.

Azkenik, ikasketa-corpusa are gehiago handitzea eta corpus horren gainean emaitza onenak lortutako parametroekin probatzea izango litzateke emaitzak hobetzeko beste modu bat.

# V. KAPITULUA

## Ondorioak eta etorkizuneko lanak

Kapitulu honetan, tesi-lan honekin egindako ekarpenak laburbildu eta ateratako ondorioak azalduko ditugu. Bukatzeko, lan honi lotuta etorkizunean egin daitezkeenak zerrendatuko ditugu.

### V.1 Ekarpenak

IXA taldean orain arte gehienbat erabili den hizkuntzaren ezagutzan oinarritutako hurbilpena osatzeko asmoz, ikasketa automatikoko teknikak erabili ditugu, batez ere, tesi-lan honetan, euskararen prozesamenduan zenbait aurrerapauso emateko. Bi arlotan egin dugu lan bereziki, azaleko sintaxiaren tratamendu automatikoan eta erroreen detekzioan. Hala, tesi-lan honetako ekarpenik garrantzitsuenak alor hauetan sortutako hiru tresnak izan dira: euskarako kateen eta perpausen identifikatzaile automatikoak eta koma-zuzentzailea.

#### V.1.1 Euskarako kateen eta perpausen identifikazio automatikoa

Azaleko sintaxiaren baitan, euskarako kate- eta perpaus-identifikatzaile automatiko sendo eta erabilgarriak sortu ditugu ($F_1 = \%$ 83,17 eta $F_1 = \%$ 77,24 emaitzekin, hurrenez hurren). Ekarpen nagusi honez gain, atal honi dagozkion hauek ere nabarmendu nahi genituzke:

1. *FR-Perceptron* ikasketa-algoritmoa arrakastaz egokitu dugu euskararako.

   Algoritmo horrek *hitz multzo*ak identifikatzeko ingelesarekin frogatuta zuen portaera ona berretsi dugu, euskarako pareko atazetan ere emaitza onak lortuz.

2. Ezaugarri linguistiko esanguratsuak gehituta hobetu ditugu kate- eta perpaus-identifikatzaileak.

   Horretarako, ezaugarrien aukeraketa egin behar izan dugu, lehenik. Kateei dagokienez, kategoriaren eta deklinabidearen informazioa izan da emaitzak hobetzen lagundu diguna: kategoria, batez ere, aditz-kateen kasuan, eta deklinabidea, berriz, izen-sintagmen kasuan. Perpausei dagokienez, berriz, probatu dugun informazio linguistiko guztia izan da baliagarria: hitza, lema, kategoria, azpikategoria, deklinabidea eta mendeko perpausen informazioa.

3. Informazio linguistikoa lortzeko darabilgun analizatzailearekiko mendekotasuna aztertu dugu.

   Erabili dugun informazio linguistiko guztia *Eustagger*-ek automatikoki lortutakoa da. *Eustagger*-en portaera, ordea, okerragoa da, eskatzen zaion informazio linguistikoaren konplexutasuna handitzen den heinean. Hala, deklinabidearen informazioa eskatzen zaionean, adibidez, hamar puntuko errore-tasa dauka $F_1$ neurrian. Ondorioz, informazio linguistikoaren kalitatea eragozpen garrantzitsua da kate-identifikatzailearentzat, batez ere. Halaxe frogatuta geratu da, eskuz desanbiguatutako informazio linguistikoa erabiliz egin ditugun probetan: perpaus-identifikatzailearen emaitzak puntu erdia baino gutxiago hobetu badira ere, kate-identifikatzailearenak ia zortzi puntu hobetu dira (ikus III.27 eta III.28 taulak neurri zehatzagoak gogora ekartzeko). Kateen identifikazioan, eskuz desanbiguatutako informazioarekin lortzen den hobekuntza dela eta, esan genezake desanbiguatzaile automatikoaren hobekuntzak —etorkizunean, lortzekotan— kateen identifikatzailearen hobekuntza ia zuzena ekar lezakeela.

4. Ikasketa automatikoko teknikak eta hizkuntzaren ezagutzan oinarritutakoak uztartu ditugu.

   IXA taldean hizkuntza-ezagutzan oinarritutako tekniken bidez sortutako kate-etiketatzailea eta perpaus-mugatzailea aprobetxatu ditugu.

Horretarako, patroi edo erregela bidez lortutako informazioa txertatu dugu ikasketa-algoritmoan, informazio gehigarri gisa (*stacking* edo pilaratzea erabiliz), eta emaitzak hobetu ditugu honela. Gainera, hobekuntza hauek estatistikoki esanguratsuak direla frogatu dugu, McNemar testaren bidez ($p < 0,05$).

### V.1.2 Euskarako koma-zuzentzailea

Euskarako koma-zuzentzaile erabilgarri bat garatu dugu, erroreen tratamendu automatikoaren arloaren baitan. Ekarpen nagusi honez gain, beste hauek nabarmenduko genituzke atal honetan:

1. Komari buruzko azterketa teorikoa egin dugu.

   Sintaxian eta puntuazioan adituak diren zenbait hizkuntzalariren gogoetak bildu eta aztertu ditugu, euskaraz egiten den komaren erabilera formalizatzeko. Gainera, ingelesez egiten den komaren erabilera ere aztertu dugu, eta baita euskarakoarekin konparatu ere.

2. Kate- eta perpaus-identifikatzaileen informazioa baliatu dugu koma-zuzentzailea hobetzeko.

   Erabili beharreko ikasketa-algoritmoa eta leihoa erabaki ondoren, hasierako ezaugarri linguistikoez gain, berriak gehitu ditugu, eta honela lortu ditugu hobekuntzarik handienak. Komaren aurrean usuen agertzen diren hitzak atributu gisa jartzean, ia sei puntuko hobekuntza lortu dugu, eta *FR-Perceptron* bidez sortutako kate- eta perpaus-identifikatzaileek ematen duten informazioa eranstean, beste zazpi puntu baino gehiagoko hobekuntza erdietsi dugu.

3. Koma-zuzentzailearen ebaluazio sakona egin dugu.

   Komak ikasteko, egunkari-corpus batean jarritako komak zuzentzat jo ditugu. Hipotesi horretatik abiatu izanak ebaluazio sakon bat egitera eraman gaitu, ikasi duguna benetan ikasi beharrekoa ote zen ikusteko. Ez da hori arrazoi bakarra izan, gainera: esaldi batean komak zuzen jartzeko aukera bat baino gehiago izateak ere soluzio posible bat baino gehiago ontzat jotzera eraman gaitu.

   Horretarako, bi hizkuntzalariren arteko adostasun-neurriak atera ditugu (% 74koa da *kappa neurria*), eta baita hizkuntzalari bakoitzak

test-corpuseko jatorrizko komekiko duen adostasun maila ere (% 76, kasurik txarrenean, 1 klasean).

| | 1 | | |
|---|---|---|---|
| | **Doit.** | **Est.** | $F_1$ |
| **Hizkuntzalari1** | 79,1 | 85,9 | 82,3 |
| **Hizkuntzalari2** | 76,1 | 76,4 | 76,3 |
| **Ikask. autom. + CG erregelak** | 77,6 | 52,7 | 62,8 |
| **Ebaluazio kualitatiboa, tokenka** | **83,01** | 58,46 | **68,61** |

Taula V.1: Hizkuntzalarien eta gure sailkatzailearen emaitzak, test-corpuseko jatorrizko komekiko; eta tokenka egindako ebaluazio kualitatiboaren emaitzak, jatorrizko komekiko eta hizkuntzalariek jarritakoekiko.

Bestalde, sailkatzaileak jarritako komak test-corpuseko jatorrizkoekiko soilik ebaluatu beharrean, aipatutako bi hizkuntzalariek jarritakoekiko ere ebaluatu ditugu, zuzentzat joaz hiru aukeretako edozein (ikus V.1 taula). Modu honetan egindako neurketetan, sailkatzailearen erabakiak test-corpuseko jatorrizko komekiko soilik egindako ebaluzioko datuek erakusten digutena baino ia sei puntu hobeak direla ikusi ahal izan dugu (% 62,8 vs % 68,61). Azpimarragarria da lortzen den doitasuna (% 83,01): sailkatzaileak jarritako bost kometatik, lau ondo jarri ditu. Estilo- eta gramatika-zuzentzaileetan doitasunak duen garrantzia dela eta, koma-zuzentzaile erabilgarri bat lortu dugula esan beharrean gaude, betiere, euskaraz zuzen samar idazten duen erabiltzaile bati zuzendua.

(Shieber eta Tao, 2003) lanari jarraiki, bestalde, esaldi mailako ebaluazioa ere egin dugu; komaren bat gaizki jartzeak esaldiaren zentzu osoa alda dezakeelako ustean dago oinarrituta ebaluazio hau. Ebaluazio-modu honetan, beraz, esaldi bat zuzentzat jotzen da, esaldiko koma guzti-guztiak baldin badatoz bat aipatutako hiru soluzioetako batekin (soluzio baten eta bestearen komak nahastu gabe). Esaldika egindako ebaluazioan, 380 esaldietatik, 219 esalditan etiketatu dira koma zuzen guztiak eta zuzenak bakarrik (% 57,63).

Esaldiaren luzeraren araberako neurriak ere atera ditugu, eta frogatu dugu, esaldika egindako ebaluazioan, esaldiaren luzera handitzearekin okerrera egiten dutela emaitzek ere.

Corpuseko jatorrizko komekiko egindako ebaluazioaren eta ebaluazio kualitatiboaren aldeak (% 62,8 vs % 68,61) erakusten digu, besteak beste, koma-zuzentzaile bat ebaluatzea ez dela lan erraza, eta ebaluazio kualitatibo bat egitea beharrezkoa dela, nahiz eta egin dugun ebaluazio automatikoa nahikoa izan koma-zuzentzaile on bat garatzeko urratsak emateko.

4. Ikasketarako informazioa lortzeko darabiltzagun tresnetan (*Eustagger*, kate- eta perpaus-identifikatzaileak...), koma erabiltzearen eragina neurtu dugu.

   Komak zuzentzen ikasteko, informazio linguistikoa eman behar zaio makinari. *Eustagger*, *Ixati*, perpaus-mugak lortzeko CG erregelak eta *FR-Perceptron* bidezko kate- eta perpaus-identifikatzaileak darabiltzagu helburu horrekin. Tresna hauek, ordea, koma baliatzen dute, ahalik eta informazio onena emateko. Komak zuzentzeko testu bat dugunean, baina, zalantzagarria da testu horretako komen erabilgarritasuna informazio linguistikoa lortzeko. Izan ere, zuzendu nahi ditugun komek —okerrak izan daitezkeenez— tresna hauen portaeran eragin negatiboa izan dezakete.

   Beraz, hainbat proba desberdin egin ditugu. Besteak beste, komak ez darabiltzan analizatzaile/desanbiguatzaile morfosintaktikoa sortu dugu (*Eustagger komagabea*), komak baliatzen ez dituzten kate- eta perpaus-identifikatzaileekin batera. Bai analisi/desanbiguazio morfosintaktikorako, eta baita kate- eta perpaus-identifikatzaileak sortzeko ere, informazio esanguratsua dira komak, eta, beraz, tresna hauen portaera —komarik gabe— okerragoa da; ondorioz, koma-zuzentzailearena ere jaitsi egiten da tresna hauen bertsio *komagabeak* erabiltzerakoan; izan ere, zenbat eta informazio linguistiko eskasagoa erabili, orduan eta emaitza kaskarragoak lortzen dira.

   Emaitza txar hauek testuko jatorrizko komak kontuan hartzera eraman gaituzte derrigor. Aztertu beharko litzateke, dena dela, jatorrizko komen zuzentasuna espero baino okerragoa bada, zer den gertatzen dena. Bitartean, komak zuzentzeko atazan, kasurik txarrenean lor daitezkeen emaitza gisa interpretatzen ditugu *Eustagger komagabea*rekin lortutako neurriak.

### V.1.3 Bestelakoak

Tesi-lan honetan, ekarpen nagusi hauez gain, beste hauek ere nabarmendu nahi genituzke:

- Ikasketa automatikoko tekniken azterketa, azaleko sintaxian eta erroreen detekzioan.

  Ikasketa automatikoko teknikak aztertu ditugu, eta hauek nola aplikatzen diren HPan. Tesi-lan honetan landu ditugun bi alorretan —azaleko sintaxian eta erroreen tratamenduan—, gaur egun baliatzen diren ikasketa automatikoko teknikak deskribatu ditugu, batez ere.

- Corpuseko erroreen detekzioa.

  Egindako ikasketa automatikoko probetan, eskuz etiketatutako informazioa (kateei eta perpausei buruzkoa) euskararen erreferentzia-corpusetik —EPECetik— eskuratu da automatikoki. Corpus hau, ordea, dependentzietan oinarrituta etiketatu zen, eta, beraz, guk behar genuen informazioa lortzeko zenbait bihurketa egin behar izan ditugu. Azpimarragarria iruditzen zaigu, hortaz, alde batetik, kateak eta perpausak identifikatzen ikasteko beharrezko corpusa guk geuk moldatu dugula, eta bestetik, prozesu horretan, eskuzko etiketatzean egindako zenbait errore topatu ditugula corpusean. Hala, gure lanak euskararen erreferentzia-corpusa hobetzeko balio izan du.

- Erroreen detekziorako oinarrizko tresnen garapena.

  Euskarako erroreen azterketa egin dugu, eta erroreen tratamendu automatikorako beharrezko diren oinarrizko tresnak sortu ditugu (Uria, 2009; Oronoz, 2009; Aldabe *et al.*, 2006, 2005b; Arrieta *et al.*, 2003): erroreak etiketatuta dituzten corpusak eta euskarako erroreen sailkapena, besteak beste.

## V.2 Ondorioak

Hurrengo lerroetan, tesi-lan honekin atera ahal izan ditugun ondorio nagusiak azalduko ditugu.

- Ikasketa automatikoko teknikak baliagarriak dira, euskarako kateen eta perpausen identifikazioan eta komen zuzenketan.

| **Hizkuntza** | **Teknika** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| *Euskara* | *FR-P* oin + dek + Erreg | Autom | 83,17 |
| *Ingelesa* | *FR-P* | Autom | 93,74 |

Taula V.2: Euskarako eta ingeleseko kate-identifikatzaileen emaitzen arteko konparaketa (*FR-Perceptron* (*FR-P*) algoritmoa eta automatikoki analizatutako eta desanbiguatutako corpusa (Autom) erabiliz); eta euskararen kasuan, *oinarrizko ezaugarri*ak (oin), deklinabidea (dek) eta erregeletan oinarritutako kateen identifikatzaileak emandako informazioa (Erreg) baliatuta, eta ikasketa-corpusaren tamaina osoa (% 100 = 104.956 token) erabilita.

Ikasketa automatikoko teknikak guztiz baliagarriak dira HPko alor anitzetan. Azken hogei urteetan teknika hauek HPan izan duten gorakada nabarmena da horren isla nagusia. Tesi-lan honetan, gainera, frogatua geratu da ikasketa automatikoa erabilgarria dela euskarako kateen eta perpausen identifikazioan eta komen zuzenketan.

- Kateak identifikatzea perpausak identifikatzea baino errazagoa da.

  Kate-identifikatzailearen eta perpaus-identifikatzailearen emaitzak konparatuz gero, kateak identifikatzea perpausak identifikatzea baino errazagoa dela ikus daiteke. Ataza bakoitzaren abiapuntuko neurriek ere gauza bera adierazten digute. Zailtasun hori, gainera, ez da hizkuntzaren araberakoa, problemaren izaeraren araberakoa baizik: perpausek izaera errekurtsiboa baitute, baina kateek, oro har, ez.

- Euskarako kateak identifikatzea ingelesekoak identifikatzea baino zailagoa da.

  V.2 taulan ikus daitezke euskarako kate-identifikatzailearen azken emaitzak, ingelesekoekin alderatuta. Ingeleseko emaitzak baino hamar puntu gutxiagoko $F_1$ neurria lortzen da euskararako. Alde batetik, esan beharra dago corpus desberdinak erabili direla, jakina, bi hizkuntzekin egindako esperimentuetan, eta, beraz, emaitzak ez direla zuzenean konparagarriak. Hori kontuan harturik ere, azken emaitzak konparatzen baditugu, abiapuntuko emaitzak ere konparatu beharko genituzke: euskarako *oinarrizko neurri*ak ingelesekoak baino 25 puntu apalagoak dira, eta horrek, gure iritziz, atazaren zailtasuna, euskararako, ingeleserako baino handiagoa dela erakusten du. Kasu honetan, euskara hiz-

| **Hizkuntza** | **Teknika** | **Desanbiguatua** | $F_1$ |
|---|---|---|---|
| *Euskara* | *FR-P* oin+ak+d+l+m+Er | Autom | 77,24 |
| *Ingelesa* | *FR-P* oin | Autom | 84,36 |

Taula V.3: Euskarako eta ingeleseko perpaus-identifikatzaileen emaitzen arteko konparaketa (automatikoki analizatutako eta desanbiguatutako corpusa (Autom) eta *FR-Perceptron* (*FR-P*) baliatuta); eta euskarakoaren kasuan, *oinarrizko ezaugarri*ak (oin), azpikategoria (ak), deklinabidea (d), lema (l), mendekoen informazioa (m) eta erregeletan oinarritutako perpausen mugatzaileak emandako informazioa (er) eta ikasketa-corpusaren tamaina osoa (% 100 = 104.956 token) erabilita.

kuntza eranskaria izateak izen-sintagmen identifikazioa zailtzen duela iruditzen zaigu.

Bestalde, euskararako erabili den kateen definizio bihurriagoak ere ez duela laguntzen uste dugu. Adibide batekin esanda, "*Nire aitaren etxea*" sintagma bakartzat hartu dugu euskaraz, eta honen ingeleseko parekoan ("*The house of my father*") hiru sintagma daudela estimatzen da: "*the house*" izen-sintagma, batetik; "*of*" preposizio-sintagma, bestetik; eta "*my father*" izen-sintagma, azkenik. Euskararako lortutako emaitza kaskarragoek, menturaz, IXA taldean orain arte kateen izaerari buruz hartutako erabakia zalantzan jartzera garamatzate: ez ote da hobea —aurreko adibidearekin jarraituz— "*Nire aitaren*" sintagma bat kontsideratzea, eta "*etxea*" beste bat, eta horien arteko lotura *a posteriori* egitea, ingelesez egiten den moduan.

- Euskarako perpausak identifikatzea ingelesekoak identifikatzea baino zailagoa da.

  Perpausen identifikazio automatikoari dagokionez, V.3 taulan ikus daitezke lortutako azken emaitzak, ingelesekoekin erkatuta. Ingeleseko emaitzak baino zazpi puntu gutxiagoko $F_1$ neurria lortzen da euskararako. Emaitzak, corpus desberdinetan ebaluatuak izanik, ez dira zuzenean konparagarriak. Hala ere, kasu honetan *oinarrizko neurri*ak bi hizkuntzentzat antzekoak direnez, ingeleserako lortzen den hobekuntza euskararako lortzen dena baino handiagoa da. Honen arrazoi nagusietako bat euskararen ordena librea —edo inguruko hizkuntzena baino libreagoa, behintzat— izan daiteke (Odriozola eta Zabala, 1993; Hidal-

go, 1994; Aldezabal *et al.*, 2003a; Erdozia *et al.*, 2009). Iruditzen zaigu ingelesez esaldi batean parte hartzen duten elementuen ordena zurrunagoak baduela zer esanik emaitzetan, eta alderantziz, euskaraz esaldi bat egiteko orduan hitzen ordena hain finkoa ez izateak ez duela laguntzen perpausen identifikazioan, ezen kasuistika zabalago baten aurrean jartzen baitu makina.

- Komaren erabilera arautu beharra dago.

  Ikusi dugun moduan, komaren erabilera ez dago guztiz arautua zenbait hizkuntzatan, eta arautua dagoenetan, arau horiek ez dira betetzen askotan. Komaren erabilera zuzenak, ordea, testuaren ulergarritasunari mesede egiten dio. Euskarak dituen ezaugarriekin —ordena librea, sistema postposizionala...—, gainera, are garrantzitsuagoa iruditzen zaigu koma egoki erabiltzea.

  Euskaraz, komak jartzeko arauak zehazteko garaian, adostasun maila handia dago arloko hizkuntzalarien artean (nahiz eta arau hauek bakoitzak bere modura arrazoitzen dituen). Hala izanik, behar-beharrezkoa iruditzen zaigu Euskaltzaindiak puntuazioari —eta, batez ere, komari— buruzko arau formalak lehenbailehen finkatzea.

- Komaren erabilera antzekoa da euskaraz eta ingelesez.

  Komaren euskarako eta ingeleseko erabilera-arauak aztertu eta konparatu ditugu, eta ikusi ahal izan dugu antzekotasun handiak daudela. Ondorioz, euskararako egin dugun lana errepika liteke ingeleserako ere. Modu horretan, ingeleseko koma-zuzentzailea lortu ahal izango genuke.

- Kate- eta perpaus-identifikatzaileen informazioa esanguratsua da koma-zuzentzailerako.

  Komari buruz egindako teorizazioaren ildotik ateratako ondorioetako bat —Shieber eta Tao-k (2003) egindako lanean ere oinarritzen zena— berretsi dugu: koma-zuzentzaile on bat sortzeko, perpaus-identifikatzaile automatiko on bat behar da. Izan ere, komaren erabilera-arauetan ikusi ahal izan dugun moduan, perpaus-mugekin erabat lotuta dago komaren erabilera. Kate-identifikatzaleak gutxiago laguntzen badu ere, informazio esanguratsua dela ikusi dugu, kate baten baitan ez baita, normalean, komarik joango.

- Corpusean eta hizkuntza-ezagutzan oinarritutako teknikak uztartzeak emaitzak hobetzen ditu, oro har.

  HPan erabiltzen diren bi hurbilpenak —corpusean eta hizkuntza-ezagutzan oinarritutakoak, alegia— bateragarriak direla erakutsi dugu tesi-lan honetan. Are gehiago, bi teknikak konbinatuta, oro har, bataren eta bestearen emaitzak hobetzen direla frogatu ahal izan dugu.

  Kateen eta perpausen identifikazioan, lehendik IXA taldean egina zegoen lana —hizkuntza-ezagutzan oinarritutako teknikak erabiliz garatutako erregelak— ikasketa automatikoko teknikekin konbinatu ditugu, eta hobekuntza estatistikoki esanguratsuak ($p < 0,05$) erdietsi ditugu horrela.

  Komak berreskuratzeko atazan, berriz, guk geuk garatutako erregelak uztartu ditugu ikasketa-automatikoko teknikak baliatuz lortu ditugun sailkatzaileekin. Kasu honetan, komak darabiltzaten kate- eta perpaus-identifikatzaileek emandako informazioa baliatuz soilik lortu da hobekuntza estatistikoki esanguratsua. Komak ez darabiltzaten kate- eta perpaus-identifikatzaileen informazioa erabilita, berriz, ez da hobekuntzarik lortu. Honen arrazoia, erregelekin lortutako estaldura txikian ikusten dugu (% 27,2ko estaldura, 1 klaserako). Estaldura honekin, jarri beharreko lau kometatik bakarra jartzen du gramatikak, eta, beraz, gutxi laguntzen dio ikasketa automatikoko algoritmoari.

- Koma-zuzentzailerako ebaluazio kualitatiboa beharrezkoa da.

  Ikasketa automatikoko tekniken bidez koma-zuzentzailea garatzeko, testuko jatorrizko komekiko egiten den ebaluazioa ez da nahikoa: jatorrizko koma hauek desegokiak izan daitezke, batetik, eta, bestetik, komen konbinazio zuzen bat baino gehiago egon daiteke esaldiko.

  Hau guztia dela eta, ebaluazio kualitatibo bat egitea beharrezkotzat jotzen dugu. Gure hautua bi hizkuntzalarik test-corpus txikiago bat etiketatu eta sailkatzailea hiru erreferentzien arabera ebaluatzea izan da, hirurak zuzentzat hartuz: corpuseko jatorrizko komena, eta bi hizkuntzalarik etiketatutako soluzioena.

- Euskarako corpus erroredun etiketatu handi bat beharrezkoa da, erroreen detekzioan aurrera egiteko.

  Ikasketa automatikoko teknikak erabiltzeko behar-beharrezkoak dira corpusak; ez edozein corpus gainera, ikasi nahi dugun kontzeptu hori

etiketatuta duten corpusak baizik. Hizkuntza baten prozesamenduan, gaur egun atzean ez geratzeko, corpus egokiak izateak berebiziko garrantzia dauka. Bestela, ikasketa automatikoko teknikek bultzatutako hobekuntzei muzin egitea garesti ordain daiteke. Corpus hetereogeneoak behar dira (iturri eta molde desberdinekoak), ahalik eta ondoen etiketatuak (atazaren arabera gauza bat edo bestea etiketatuta dutenak), eta corpus handiak behar dira, gainera. Beste hitzetan esanda, HPan atzera geratu nahi ez duen hizkuntzak corpus sendoak beharko ditu. Euskararen prozesamendua lantzeko —eta hizkuntzaren normalizazioa bultzatzeko—, beraz, corpusgintzan lan handia egin beharra dago.

EPEC corpusaren sorkuntza, zentzu horretan, aurrerapauso handia izan bada ere, uste dugu, erroreen detekzioaren alorrean aurrera egiteko, erroreak etiketatuta dituen corpus sendo bat osatzea dela eginbehar garrantzitsuenetako bat, ikasketa automatikoko teknikak alor horretan ere baliatu ahal izateko. Gramatika-zuzentzaile bat sortzeko beharrezkoa den urratsa iruditzen zaigu, zalantzarik gabe.

Komaren kasua desberdina den arren —ustez komak ondo jarrita dituen corpus batetik ikas baitaiteke, guk geuk egin dugun moduan—, puntuazioan adituak diren hizkuntzalariek komak esplizituki etiketatuz gero, ikasketa hobea egin daitekeela uste dugu; corpus txiki bat eskuz etiketatuta, *active learning* deiturikoa egin daiteke, esate baterako.

## V.3 Etorkizuneko lanak

HPan egiten diren tresnak ez dira inoiz perfektuak, eta, beraz, beti egin daiteke saiakeraren bat tresna horiek hobetzeko. Ikasketa automatikoko tekniken bidez garatutako tresnak hobetzeko, gainera, bada modu bat, goi-mugara iritsi bitartean, emaitzen hobekuntza ziurtatzen duena: ikasketa-corpusa handitzea, hain zuzen. Beste kontu bat da zenbat handitu behar den corpusa hobekuntza adierazgarri bat lortzeko, eta ea ahalegin horrek merezi ote duen.

Atal honetan, tesi-lan honek irekitako bideak aztertuko ditugu, sortutako tresna bakoitza hobetzeko egin daitezkeenetatik abiatuta, eta ikasketa-corpusa handitzearena albo batera utzita.

- Kate-identifikatzailea hobetzea.

  V.2 atalean azaldu dugun moduan, kateen beste definizio bat erabilita, agian, hobekuntzak lor daitezke. Horretaz gain, ez dugu uste ikasketa automatikoko algoritmo gehiago probatu beharko genituzkeenik, are gutxiago *FR-Perceptron* algoritmoa onenen artean egonik eta Banko eta Brill-ek (2001) esandakoa kontuan harturik: ikasketa automatikoko algoritmo desberdinak probatzen edo moldatzen lortzen diren hobekuntzak baino handiagoak lortuko liratekeela ikasketa-corpusak handituz.

  III. kapituluan aipatu dugun moduan, bestalde, kateen identifikazioa egiterakoan ez dugu kontuan hartu kateak ez-jarraiak eta, ondorioz, errekurtsiboak izan daitezkeela euskaraz. Ezaugarri hau aintzat hartzea interesgarria litzateke, kateen identifikatzaile osoago bat izate aldera.

  Gainera, *Eustagger*-ek kate-identifikatzaileak behar duen informazioa —deklinabidearena, batez ere— desanbiguatzean egiten duen errore-tasa txikitzea lortuko bagenu, emaitzak are gehiago hobetuko liratekeela uste dugu. Bestalde, (Ando eta Zhang, 2005) lanaren ildotik, ikasketa erdi-gainbegiratua jorra liteke.

- Perpaus-identifikatzailea hobetzea.

  Informazio linguistiko konplexuagoa erabiltzen saia gintezke: batetik, IXA taldearen analizatzaile berriak ematen digun funtzio sintaktikoen edo dependentzien informazioa erabil liteke; bestetik, (Nguyen *et al.*, 2009) lanean kontuan hartzen diren atributu kalkulatuak balia litezke (uneko hitzetik esaldiaren hasierara eta bukaerara dauden zenbait elementu garrantzitsuren kopuruak, hain zuzen). Bestalde, (Ram eta Devi, 2008) lanean aipatzen den moduan, perpaus-identifikatzaileak egiten dituen akatsak aztertu, eta errore horiek jatorrian duten gabezia konpontzeko erregela linguistiko gehiago egin litezke.

- Koma-zuzentzailea hobetzea.

  Hainbat proba egin litezke koma-zuzentzailea hobetzen saiatzeko.

  – Koma bat jarri ala ez erabakitzeko, aurreko komen informazioa izatea.

    Kasu batzuetan, hitz baten ondoren koma bat jarri behar den ala ez erabakitzeko, esaldi horretan hitz horren aurretik jarritako

komak kontuan hartu behar dira. Esate baterako, tartekiak normalean koma artean joango dira, baina mendeko perpaus baten barruan datozenean, koma gabe joan daitezke testua arintzearren. Hala, mendeko perpaus baten koma identifikatzen badu makinak, eta informazio hori baldin badauka tartekia aztertzerakoan, komarik ez jartzea erabaki dezake. Kasu hauetarako, lehendabizi, ikasketa-corpuseko adibide bakoitzari —token bakoitzari— esaldiko bere aurreko tokenek komarik duten ala ez pasa beharko litzaioke. Gero, proba egiterakoan —test-corpusarekin—, dinamikoki esleitu beharko litzaioke informazio hori token bakoitzari. HMM edo CRF algoritmoek, dituzten berezko ezaugarriengatik, egokiak dirudite modu honetako atazak ebazteko.

– Komaren emaitza-klaseen desoreka konpontzeko marjina aldagarriak erabiltzea.

  Komak berreskuratzeko atazan, emaitza-klaseen arteko desoreka da izan dugun arazo handienetako bat; hau da, corpusean askoz adibide gutxiago ditugu komaz jarraituak (1 klasea deitu diogu-na), komaz ez jarraituak (0 klasea) baino. Horregatik, erabili ditugun algoritmoek emaitza hobeak lortu dituzte 0 klaserako. Klaseen desoreka hau, ordea, arazo nahiko normala da HPan: dokumentuen sailkapenean gertatzen da, besteak beste. Arloaren literaturan ikusi dugun moduan, ordea, badira arazo hau ebazteko bideak. Batzuek SVMren eta *pertzeptroi*en gisako algoritmoak moldatzen dituzte marjina aldagarriak baliatzeko; alegia, bi klaseen arteko marjina handieneko hiperplanoa bilatu ordez, klase bakoitzerako marjina bana bilatzen dute, klase bateko adibideen kopuru handiagoak emandako *abantaila* nolabait orekatzeko. Ebazpide honekin hobekuntza esanguratsuak aurkeztu dituzte zenbait lanek (Li *et al.*, 2009, 2002b).

– *SVM* ikasketa-algoritmoarekin proba gehiago egitea.

  *SVM* ikasketa-algoritmoak atributu askorekin emaitza onak lortzen dituenez (Rehurek eta Sojka, 2010), erabili ditugunak baino atributu gehiago erantsita, proba gehiago egin litezke. Esate baterako, komaren aurretik maizen agertzen diren hitz kopuru handiagoa erabil genezake, edota leiho handiagoa.

– Euskara-ikasleentzat koma-zuzentzailea moldatzea.

  Aipatu dugun moduan, jatorrizko komak kontuan hartuta lortzen dira koma-zuzentzailearen daturik onenak. Koma horiek nork jartzen dituen, ordea, erabakiorra izango da, informazio horrek koma-zuzentzaileari laguntzen edo oztopatzen dion jakiteko. Hortaz, etorkizunean, euskara-ikasleen testuekin proba batzuk egitea interesgarria litzatekeela uste dugu, eurek jarritako komak mantentzea ala ez komenigarria den ikusteko. Ikaslearen mailaren arabera, erabaki desberdinak hartu beharko direla iruditzen zaigu.

– Komak jartzen *ikasten laguntzen* duen modulu batekin uztartzea komen zuzentzailea.

  Horretarako, koma bakoitza zein motatakoa den identifikatu beharko litzateke. Honela, testuetan komak zuzentzeaz gain, koma bakoitza zergatik jarri behar den azaldu ahal izango genuke.

  Esan bezala, ordea, ezinbestekoa da horretarako, koma non jarri behar den jakiteaz aparte, ezagutzea zein den koma leku horretan jartzeko arrazoia. Hori dela eta, koma bakoitzari dagokion klasea zein den detektatu beharko genuke automatikoki; alegia, koma bakoitzaren rol sintaktikoa identifikatu beharko litzateke. Era berean, koma bakoitzaren rol sintaktiko bakoitzari erabiltzailearentzat ulergarria den azalpen bat lotu beharko genioke. Honela, faltan edo soberan dagoen koma bat identifikatzerakoan, bere rol sintaktikoa lortu eta rol sintaktiko horri dagokion azalpena emango genuke.

  Hau egitea ez da erraza, baina badira ikasketa automatikoan oinarritutako teknikak, horretarako erabil daitezkeenak; ikus, adibidez, IV.2 atalean azaldutako (Delden eta Gomez, 2002) eta (Srikumar *et al.*, 2008) lanak.

  Komaren funtzioak edo rol sintaktikoak ikasteko, ordea, ikasketa-corpus bat beharko genuke, non koma bakoitza bere rol sintaktikoarekin etiketatuta egon beharko litzatekeen. Corpus hau sortzea, oraingoz, ez dago gure egitekoen artean.

– Ingeleseko eta gaztelaniako koma-zuzentzaileak garatzea.

  Arestian ikusi dugun moduan, ingeleseko komari buruzko arauak euskarakoen oso antzekoak dira, salbuespenak salbuespen. Gaztelaniakoak ere antzekoak direla iruditzen zaigu, hau ziurtatzeko

azterketa sakon bat egin beharko litzatekeen arren. Dena dela, ikasketa automatikoko teknikak erabiliz, euskararako egindako lana erraz zabal liteke beste bi hizkuntza horietara ere, eta gaztelaniako eta ingeleseko koma-zuzentzaileak lortu horrela.

- Ingeleseko kate- eta perpaus-identifikatzaileak hobetzen saiatzea.

  Tesi-txosten honetan ikusi dugun moduan, informazio linguistiko esanguratsua gehituz identifikatzaileen portaera hobe daiteke. Hala, analizatzaile morfosintaktiko batek eman dezakeen informazioa eransteaz gain, zenbait erregela linguistiko sortu eta hauek ematen duten informazioa ere gehi daiteke *FR-Perceptron* algoritmoan, beste ezaugarri baten moduan.

- Puntuaren zuzentzailea garatzea.

  Puntuaren zuzentzailea garatzea ere garrantzitsua izan liteke komen zuzentzailea osatzeko. Izan ere, komak ondo jartzea askoz zailagoa da puntuak ondo jarri gabe dituen testu batean. Hala, koma-zuzentzailea egiterakoan, puntua ondo jarrita zegoela suposatu baldin badugu ere, hipotesi hau ez da beti betetzen: corpusean, makina bat aldiz ikusi ditugu punturen bat behar zuten esaldiak. Horregatik, puntua non jarri zuzentzen duen tresna bat gara liteke. Puntua non jarri zuzentzen duen tresna, koma-zuzentzailea baino lehenago aplikatu beharko litzaieke testuei.

- *FR-Perceptron* algoritmoan hobekuntzak egitea.

  III. kapituluan, *FR-Perceptron* algoritmoa baliatu dugu euskarako kateen eta perpausen identifikaziorako. Horretarako, zenbait aldaketa egin behar izan ditugu. Bi izan dira garrantzitsuenak: batetik, informazio linguistiko berria, hau da, atributu berriak txertatzeko egin behar izan ditugun aldaketak; bestetik, ikasketa prozesuan lantzen ari garen hizkuntza bakoitzari dagozkion ezaugarri konkretuak ingelesetik euskarara itzultzea.

  Aldaketa hauek guztiak parametriza litezke, etorkizunean norbaitek *FR-Perceptron* algoritmoa erabili nahi izanez gero erraztasunak izan ditzan: nahi adina atributu berri gehitzeko aukera erosoa eta hizkuntzaren arabera beharrezkoak diren hitz konkretuak eransteko modua.

  Ingelesetik euskarara ekarritako ezaugarri konkretuei dagokienez, III. kapituluan aipatu ditugu ezaugarri hauek zeintzuk diren: ingeleserako

erlatibozko perpausak egiteko erabiltzen diren partikulak (*who, whose, where...*). Partikula hauek euskarako ordainekin ordezkatu ditugu (*non, zein, zeinaren...*) *FR-Perceptron* algoritmoan, eta hortxe dago aldaketa honen gabezia. Izan ere, dakigun moduan, erlatibozko perpausak eta gainontzeko mendeko perpausak egiteko, euskaraz menderagailu-atzizkiak erabiltzen dira maiz, hitz osoak baliatu beharrean ("*Zurekin etorri d*en *gizona*"). Etorkizunean, mendeko perpausak markatzen dituzten atzizki hauek *FR-Perceptron* algoritmoan gehitzeak perpausen identifikazioaren atazan emaitzak hobetzea ekar lezake. Horretarako, atzizki hauek erauzi eta aprobetxatu beharko lituzke algoritmoak.

Bestalde, ingeleserako *FR-Perceptron* algoritmoan, erlatibozko partikulez gain, mendeko beste partikulak gehitzen ere proba liteke (*since, because...*).

- Bestelako erroreen detekzioa lantzea, ikasketa automatikoko tekniken bidez.

  Corpus handiagoak lortu bitartean, hiru estrategia erabil litezke ikasketa automatikoa baliatzeko erroreen detekzioan, II. kapituluan aipatu dugun moduan: *active learning* edo *ikasketa bizia* delakoa, erroreak automatikoki sortzea eta ikasketa ez-gainbegiratua edo erdi-gainbegiratua egitea.

  Banko eta Brill-en (2001) iritziz, ikasketa ez-gainbegiratua eta *active learning* teknika konbinatuz, bi tekniken propietateak aprobetxa litezke, eta emaitzak gehiago hobetu, beharbada.

  Euskarako erroreen sailkapena aztertuta, tratagarriak diren erroreak aukeratu beharko dira, etorkizun hurbilean horien tratamendu informatikoa lantzeko. Zehazki, oso errore konkretuak aukeratzea komeni da: esate baterako, determinatzaile-errore guztiak aukeratu beharrean, zenbatzaile zehaztugabeenak soilik (*zenbait, hainbat...*).

  Badira, hala ere, automatikoki tratatzeko etiketatze berezirik behar ez duten erroreak: testuinguruaren araberako errore ortografikoa da horietako bat —beste bat koma dela ikusi dugu tesi honetan—. Ataza honetan emaitzarik onenak lortzen dituztenen ikasketa automatikoko algoritmoak gurera ekartzea izango litzateke egin beharrekoa, kasu horretan.

- Metadatuak erabiltzea.

  Ikasketa automatikoan, sarritan, oso baliagarriak dira datuei buruzko datuak (metadatuak, alegia). Metadatuek, eskuarki, atributuen arteko erlazioak adierazten dituzte. Hiru motatakoak izan ohi dira erlazio hauek: semantikoak, kausalak eta funtzionalak (Witten eta Frank, 2005). Atributuen arteko erlazio hauek ikasketa eskemetan kontuan hartzeak emaitzak hobetzea ekar lezake. Arazo handi bat dago, ordea: ez da batere erraza erlazio hauek —metadatuak— ikasketa algoritmoan txertatzeko modu erraz eta ulergarri batean errepresentatzea.

- Analisi-katean integratzea.

  Ikasketa automatikoko teknikekin garatutako kate- eta perpaus- identifikatzaileak, hizkuntzaren ezagutzan oinarritutako teknikekin uztartu ditugunak, analisi-katean txertatu beharko lirateke. Hala, IXA taldearen analizatzailea hobetuko genuke.

- XUXENg euskarako gramatika-zuzentzailean, koma-zuzentzailea txertatzea.

  Koma-zuzentzailea XUXENg gramatika-zuzentzailean txertatu beharko litzateke etorkizunean. Koma-zuzentzailearen erabakiak justifikatzeko, erabiltzaileari azalpen bat eman beharko litzaioke; horretarako, koma bakoitzari dagokion rol sintaktikoa identifikatu beharko genuke, ingeleserako (Bayraktar *et al.*, 1998) eta (Delden eta Gomez, 2002) lanek abiatutako bideak jorratuz.

# Bibliografia


Abney S. Parsing by chunks. *Principle-Based Parsing*, 1991.

Abney S. Partial parsing via finite-state cascades. *Natural Language Engineering*, 1995.

Abney S. Part-of-speech tagging and partial parsing. *Corpus-Based Methods in Language and Speech Processing. ELSNET*, 1997.

Abney S. *Semisupervised Learning for Computational Linguistics*. Chapman and Hall/CRC, 2008.

Aduriz I. *EUSMG: Morfologiatik sintaxira Murriztapen Gramatika erabiliz. Euskararen desanbiguazio morfologikoaren tratamendua eta azterketa sintaktikoaren lehen urratsak*. Doktoretza-tesia, Filologia eta Historia-Geografia Fakultatea. UPV-EHU, Gasteiz, 2000.

Aduriz I., Agirre E., Aldezabal I., Alegria I., Ansa O., Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar M., Oronoz M., Sarasola K., Soroa A., eta Urizar R. A framework for the automatic processing of Basque. *Proceedings of the Workshop on Lexical Resources for Minority Languages*, Granada, Spain, 1998.

Aduriz I., Aldezabal I., Aranzabe M., Arrieta B., Arriola J.M., Atutxa A., Díaz de Ilarraza A., Gojenola K., Oronoz M., Sarasola K., eta Urizar R. The design of a digital resource to store the knowledge of linguistic errors. *DRH2002 (Digital Resources for the Humanities)*, Edimburgh, 2002.

Aduriz I., Aranzabe M., Arriola J.M., eta Díaz de Ilarraza A. Sintaxi partziala. In Fernández I. eta Laka I., editors, *Andolin Gogoan: Essays in Honour of Professor Eguzkitza*, Bilbo, 2006a. UPV/EHU Argitalpen Zerbitzua.

Aduriz I., Aranzabe M., Arriola J.M., Díaz de Ilarraza A., Gojenola K., Oronoz M., eta Uria L. A cascaded syntactic analyser for Basque. In Gelbukh A., editor, *Computational Linguistics and Intelligent Text Processing: 5th International Conference CICLing2004, Seoul, Korea, February 15-21*, 2945 lib. of *Lecture Notes in Computer Science*, 124–134. Springer-Verlag GmbH, 2004.

Aduriz I., Aranzabe M., Arriola J.M., Atutxa A., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Oronoz M., Soroa A., eta 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., Rayson P., eta Archer D., editors, *Corpus Linguistics Around the World*, 56 lib. of *Book series: Language and Computers*, 1–15, Netherlands, 2006b.

Aduriz I., Arrieta B., Arriola J.M., Díaz de Ilarraza A., Izagirre E., eta Ondarra A. Muga gramatikaren optimizazioa. Barne-txostena UPV/EHU / LSI / TR 9-2006, University of the Basque Country, Informatika Fakultatea, Donostia, 2006c.

Aduriz I. eta Díaz de Ilarraza A. Morphosyntactic disambiguation and shallow parsing in computational processing of Basque. *Inquiries into the lexicon-syntax relations in Basque*, 1–21, 2003.

Agirre E., Aldezabal I., Etxeberria J., Izagirre E., Mendizabal K., Quintian M., eta Pociello E. Improving the Basque WordNet by corpus annotation. *Proceedings of Third International WordNet Conference*, Jeju (Korea), 2006. (2007-07-02an atzitua).

Agirre E., Alegria I., Arregi X., Artola X., Díaz de Ilarraza A., Urkia M., Maritxalar M., eta Sarasola K. XUXEN: A spelling checker/corrector for Basque based on two-level morphology. *Proceedings of ANLP'92*, 119–125, Povo Trento, 1992.

Agirre E., Gojenola K., Sarasola K., eta Voutilainen A. Towards a single proposal in spelling correction. In Boitet C. eta Whitelock P., editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, 22–28, San Francisco, California, 1998. Morgan Kaufmann Publishers.

Aït-Mokhtar S. eta Chanod J.P. Incremental finite-state parsing. *Proceedings of the fifth conference on Applied Natural Language Processing*, 72–79, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

Alberdi X. eta Sarasola I. *Euskal estilo libururantz. Gramatika, estiloa eta hiztegia*. Euskal Herriko Unibertsitatearen Argitalpen Zerbitzua, 2001.

Aldabe I., Aldezabal I., Aranzabe M., Arrieta B., Díaz de Ilarraza A., Gojenola K., Maritxalar M., Oronoz M., Otegi A., eta Uria L. Euskarazko erroreen sailkapena ERROREAK eta DESBIDERATZEAK datubaseetan. Barne-txostena UPV/EHU / LSI / TR 27-2005, University of the Basque Country, Informatika Fakultatea, Donostia, 2005a.

Aldabe I., Arrieta B., de Ilarraza A.D., Maritxalar M., Niebla I., Oronoz M., eta Uria L. Basque error corpora: a framework to classify and store it. *In the Proceedings of the 4th Corpus Linguistic Conference*, 2007.

Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Niebla I., Oronoz M., eta Uria L. The use of NLP tools for Basque in a multiple user CALL environment and its feedback. *TAL & ALAO workshop. Proceedings of the 13th Conference Sur Le Traitement Automatique des Langues Naturelles*, 2 lib., 815–824, Leuven, Belgium, 2006.

Aldabe I., Arrieta B., Díaz de Ilarraza A., Maritxalar M., Oronoz M., eta Uria L. Propuesta de una clasificación general y dinámica para la definición de errores. *Revista de Psicodidáctica*, 10(2):47–60, 2005b.

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K., eta Zabala I. Hitz-hurrenkeraren azterketa masiboa corpusean. Barne-txostena, EHU, 2003a.

Aldezabal I. *Aditz-azpikategorizazioaren azterketa sintaxi partzialetik sintaxi osorako bidean. 100 aditzen azterketa, Levin-en (1993) lana oinarria hartuta, eta metodo automatikoak baliatuz.* Doktoretza-tesia, Euskal Filologia Saila, Euskal Herriko Unibertsitatea, Leioa, 2004.

Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernández G., eta Lersundi M. Edbl: a general lexical basis for the automatic processing of Basque. *IRCS Workshop on Linguistic Databases.*, Philadelphia, USA, 2001.

Aldezabal I., Aranzabe M., Arrieta B., Maritxalar M., eta Oronoz M. Toward a punctuation checker for Basque. *Proceedings of the ATALA workshop of Punctuation*, Paris, France, 2003b.

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., eta Sarasola K. Patrixa: A unification-based parser for basque and its application to the automatic analysis of verbs. *Inquiries into the lexicon-syntax relations in Basque*, 2003c.

Alegria I., Arregi O., Ezeiza N., Fernández I., eta Urizar R. Design and development of a named entity recognizer for an agglutinative language. *First International Joint Conference on NLP (IJC-NLP-04); Workshop on Named Entity Recognition*, 2004.

Alegria I. *Euskal morfologiaren tratamendu automatikorako tresnak*. Doktoretza-tesia, Informatika Fakultatea. UPV-EHU, uztaila 1995.

Alegria I., Arrieta B., Carreras X., Díaz de Ilarraza A., eta Uria L. Chunk and clause identification for basque by filtering and ranking with perceptrons. *Proceedings of SEPLN*, Madrid. Spain, 2008a.

Alegria I., Arrieta B., Díaz de Ilarraza A., Izagirre E., eta Maritxalar M. Using machine learning techniques to build a comma checker for Basque. *Proceedings of Coling-ACL*, 1–8, Sydney. Australia, 2006.

Alegria I., Artola X., eta Sarasola K. Improving a robust morphological analyser using lexical transducers. *Recent Advances in Natural Language Processing. Current Issues in Linguistic Theory (CILT) series*, Vol. 136: 97–110, 1997.

Alegria I., Ceberio K., Ezeiza N., Soroa A., eta Hernandez G. Spelling correction: from two-level morphology to open source. *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Marocco, 2008b.

Alexopoulu A. El error: un concepto clave en los estudios de adquisición de segundas lenguas. *RLA: Revista de Lingüística Teórica y Aplicada*, 43(1): 75–92, 2005.

Allen J. *Natural language understanding (2nd ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1995.

Alsina A., Badia T., Boleda G., Bott S., ngel Gil, Quixal M., eta Valentn O. Catcg: Un sistema de anlisis morfosintctico para el cataln. *Procesamiento de Lenguaje Natural*, 2002.

Amundarain I., Artiagoitia X., Etxepare R., Elordieta G., Hualde J., de Urbina J.O., Oyharcabal B., Trask R., eta Zabala I. *A grammar of Basque*. Berlin/New York: Mouton de Gruyter, 2003.

Ando R.K. eta Zhang T. A high-performance semi-supervised learning method for text chunking. *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 1–9, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

Ansa O., Arregi X., Arrieta B., Ezeiza N., Fernandez I., Garmendia A., Gojenola K., Laskurain B., Martínez E., Oronoz M., Otegi A., Sarasola K., eta Uria L. Integrating NLP tools for Basque in text editors. *Proceedings of the 1st International Workshop on Proofing Tools and Language Technologies*, Patras, Greece, 2004.

Antal P.L., Bosch A.V.D., Krahmer E., eta Swerts M. Improving machine-learned detection of miscommunications in human–machine dialogues through informed data splitting. *ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics*, 2002.

Aranguren J.L., Arrarats I., Goikoetxea A., eta Zabalondo B. *Berria - Estilo liburua*. Berria, 2006.

Aranzabe M.J. *Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitz-bankua eta gramatika konputazionala*. Doktoretza-tesia, Euskal Filologia Saila, Euskal Herriko Unibertsitatea, 2008.

Arppe A. Developing a Grammar Checker for Swedish. *Proceedings from the 12th Nordiske datalingvistikkdager*, Department of Linguistics, Norwegian University of Science and Technology (NTNU), December 9-10 2000. Nordgard.

Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., García E., Lascurain V., Sarasola K., Soroa A., eta Uria L. Semiautomatic conversion of the *Euskal Hiztegia* Basque dictionary to a queryable electronic form. *T.A.L. journal*, 44(2):107–124, 2003.

Arrieta B., Alegria I., de Ilarraza A.D., Aranzabe M., eta Aldezabal I. Using a clause identifier to improve a comma checker for basque: testing the agreement with human judges. *ICETAL: Proceedings of the 7th international conference on Natural Language Processing*, 2010.

Arrieta B., Díaz de Ilarraza A., Gojenola K., Maritxalar M., eta Oronoz M. A database system for storing second language learner corpora. *Learner corpora workshop. Corpus linguistics 2003*, number 1 in 16, 33–41, Lancaster, UK, 2003.

Arriola J.M. *Euskal Hiztegia-ren azterketa eta egituratzea ezagutza lexikalaren eskuratze automatikoari begira. Aditz-adibideen analisia Murriztapen Gramatika baliatuz, azpikategorizazioaren bidean.* Doktoretza-tesia, Filologia eta Historia-Geografia Fakultatea. UPV-EHU, 2000.

Atserias J., Carmona J., Castellon I., Cervell S., Civit M., Marquez L., Marti M., Padro L., Placer R., Rodriguez H., Taule M., eta Turmo J. Morphosyntactic analysis and parsing of unrestricted spanish text, 1998.

Atserias J., Casas B., Comelles E., González M., Padró L., eta Padró M. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), ELRA*, Genoa, Italy, 2006.

Avinesh P. eta Karthik G. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. *Proceedings of the IJCAI2007. Workshop On Shallow Parsing for South Asian Languages*, Hyderabad, India, 2007.

Ayan N.F., Borr B., eta Habash N. Multi-align: Combining linguistic and statistical techniques to improve alignments for adaptable mt. *In Proceedings of AMTA*, 17–26, 2004.

Azkarate M., Kintana X., Mendiguren X., Etxeberria E., eta Gurrutxaga A. *Elhuyar Hiztegia. Euskara-Gaztelania, Castellano-Vasco. 3. argitaralpena.* Elhuyar Fundazioa, Usurbil, 2006.

Badia T., Gil A., Quixal M., eta Valentín O. NLP-enhanced error checking for Catalan unrestricted text. *Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004*, 1919–1922, Lisbon, Portugal, 2004.

Baldwin T. eta Joseph M. Restoring punctuation and casing in english text. *AI '09: Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence*, 547–556, Berlin, Heidelberg, 2009. Springer-Verlag.

Banko M. eta Brill E. Scaling to very very large corpora for natural language disambiguation. *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 26–33, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

Bayraktar M., Say B., eta Akman V. An analysis of English punctuation: the special case of comma. *International Journal of Corpus Linguistics*, 3 (1):33–57, 1998.

Becker M., Bredenkamp A., Crysmann B., eta Klein J. Annotation of error types for german news corpus. *Proceedings of the ATALA Workshop on Treebanks*, Paris, France, 1999.

Beeferman D., Berger A., eta Lafferty J. Cyberpunk: a lightweight punctuation annotation system for speech. *Proceedings of the IEEE International Conference on Acoustics, Speech an Signal Processing*, Seattle, WA. USA, 1998.

Beesley K.R. eta Karttunen L. *Finite State Morphology.* CSLI Studies in Computational Linguistics, 2003.

Bengoetxea K. eta Gojenola K. Desarrollo de un analizador sintáctico estadístico basado en dependencias para el euskera. *Actas del XXIII Congreso de la SEPLN*, Sevilla, Spain, 2007.

Bengoetxea K. eta Gojenola K. Application of feature propagation to dependency parsing. *International Workshop on Parsing Technologies*, 2009.

Bick E. The parsing system palavras automatic grammatical analysis of portuguese in a constraint grammar framework. *In Proceedings of the 3rd Conference on Applied Natural Language Processing*, 2000.

Bick E. A constraint grammar parser for spanish. *In Proceedings of TIL*, 2006.

Bies A., Fergusson M., Katz K., eta MacIntyre R. Bracketing guidelines for treebank ii style penn treebank project. Barne-txostena, University of Pennsylvannia, 1995.

Bigert J. Probabilistic detection of context-sensitive spelling errors. *Proceedings of LREC-04*, 5 lib., 1633–1636, Lisbon, Portugal, 2004.

Bigert J. eta Knutsson O. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. *Romand 2002 (Robust Methods in Analysis of Natural language Data)*, Frascati, Italy, 2002.

Birn J. Detecting grammar errors with Lingsoft's Swedish grammar-checker. *Proceedings from the 12th Nordiske Datalingvistikkdager*, Department of Linguistics, Norwegian University of Science and Technology (NTNU), December 9-10 2000. Nordgard.

Black E., Abney S., Flickenger D., Gdaniec C., Grisham R., Harrison P., Hindle D., Ingria R., Jelinek F., Klavans J., Liberman M., Marcus M., Roukos S., Santorini B., eta Strzalkowski T. A procedure for quantitatively comparing the syntactic coverage of English grammars. *Proceedings of DARPA Workshop on Speech and Natural Language*, 1991.

Bresnan J. *The Mental representation of grammatical relations*. Cambridge Mass; MIT Press, 1982.

Brill E. *A Corpus-Based Approach to Language Learning*. Doktoretza-tesia, University of Pennsylvania, 1993.

Brill E. Transformation-based error-driven learning and Natural Language Processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

Briscoe T. eta Carroll J. Developing and evaluating a probabilistic lr parser of part-of-speech and punctuation labels. *ACL/SIGPARSE 4th International Workshop on Parsing Technologies*, Prague/ Karlovy Vary (Czech Republic), 1995.

Briscoe T. eta Carroll J. Robust accurate statistical annotation of general text. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, 1499–1504, Las Palmas, Gran Canaria, 2002.

Briscoe T., Carroll J., eta Watson R. The second release of the rasp system. *Proceedings of the COLING/ACL on Interactive presentation sessions*, Sidney, Australia, 2006.

Brockett C., Dolan W.B., eta Gamon M. Correcting esl errors using phrasal smt teqhniques. *Proceedings of the 21st COLING and the 44th ACL*, Sydney, Australia, 2006.

Buchholz S. eta Marsi E. CoNLL-X shared task on multilingual dependency parsing. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*. Association for Computational Linguistics, 2006.

Carlberger J., Domeij R., Kann V., eta Knutsson O. A Swedish grammar checker. Unpublished. Submitted 2002 Association for Computational Linguistics, 2002.

Carletta J. Assessing agreement on classication tasks: the kappa statistic. *Computational Linguistics*, 22(2), 1996.

Carreras X. *Learning and Inference in Phrase Recognition: A Filtering-Ranking Architecture using Perceptron*. Doktoretza-tesia, Polytechnic University of Catalunya, 2005.

Carreras X., Màrquez L., eta Castro J. Filtering-ranking perceptron learning for partial parsing. *Machine Learning Journal, Special Issue on Learning in Speech and Language Technologies*, 60(1-3):41–71, 2005.

Carreras X. eta Màrquez L. Boosting trees for clause splitting. In Daelemans W. eta Zajac R., editors, *Proceedings of CoNLL*, 73–75. Toulouse, France, 2001.

Carreras X. eta Màrquez L. Phrase recognition by filtering and ranking with perceptrons. *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP*. Borovets, Bulgaria, 2003.

Cendejas E., Barceló G., Gelbukh A., eta Sidorov G. Incorporating linguistic information to statistical word-level alignment. *CIARP '09: Proceedings of the 14th Iberoamerican Conference on Pattern Recognition*, 387–394, Berlin, Heidelberg, 2009. Springer-Verlag.

Cermeno O. Euskarazko errore sintaktikoen detekzioa ikasketa automatikoa erabiliz. kasu praktikoa: determinatzaile eta komunztadura erroreak. Barne-txostena, Euskal Herriko Unibertsitatea, 2008.

Charniak E. Treebank grammars. *Proceedings AAAI-96*, Menlo Park, California, USA, 1996.

Charniak E. A Maximum-Entropy-Inspired parser. *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, 132–139, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Chodorow M. eta Leacock C. An unsupervised method for detecting grammatical errors. *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, 140–147, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Chodorow M., Tetreault J., eta Han N.R. Detection of Grammatical Errors Involving Prepositions. *4th ACL-SIGSEM workshop on Prepositions*, Prague, 2007.

Chomsky N. *Lectures on Government and Binding*. Foris Publications, The Netherlands, 1981.

Christensen H., Gotoh Y., eta Renals S. Punctuation annotation using statictical prosody models. *Proceedings of the 2001 ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*. International Speech Communication Association, 2001.

Church K. A stochastic parts program and noun phrase parser for unrestricted texts. *Proceedings of the second conference on Applied Natural Language Processing*. Association for Computational Linguistics, 1988.

Civit M. *Criterios de etiquetación y desambiguación morfosintáctica de corpus en español*. Sociedad Española para el Procesamiento de Lenguaje Natural, 2003.

Collins H. *Collins Cobuild ENglish Grammar*. The Cobuild Series from the Bank of ENglish, 1992.

Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *EMNLP*, 2002.

Collins M. A new statistical parser based on bigram lexical dependencies. *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996.

Collins M. Three generative, lexicalised models for statistical parsing. *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, 16–23, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

Corder P.S. The significance of learner's errors. *IRAL (International Review of Applied Linguistics)*, 5:161–170, 1967.

Corder P.S. Idiosyncratic dialects and error analysis. *International Review of Applied Linguistics*, 5(4):147–160, 1991. Traducción al español: "Dialectos idiosincrásicos y análisis de errores. Juana Muñoz Liceras (Ed.) 1991, La adquisición de las lenguas extranjeras, Visor, Madrid, pp. 63–77.

Daelemans W. Machine learning approaches to syntactic wordclass tagging. *Machine Learning*, 285–303, 1999.

Daelemans W. eta Bosch A.V.D. *Memory-Based Language Processing*. Cambridge University Press, 2005.

Daelemans W., Buchholz S., eta Veenstra J. Memory-based shallow parsing. *Proceedings of the EMNLP/VLC*, 239–246, Maryland, USA, 1999.

Dagan I. eta Engelson S.P. Committee-based sampling for training probabilistic classifiers. *In Proceedings of the Twelfth International Conference on Machine Learning*, 150–157. Morgan Kaufmann, 1995.

Dale R. Symbolic approaches to Natural Language Processing. In Dale R., Moisl H., eta Sommers H., editors, *Handbook of Natural Language Processing*. Marcel Dekker Inc, 2000.

Delden S.V. eta Gomez F. Combining finite state automata and a greedy learning algorithm to determine the syntactic roles of commas. *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, Washington D.C. USA, 2002.

DeRose S. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 1988.

Dhonnchadha E.U. *Part-of-speech tagging and partial parsing for Irish using finite-state transducers and constraint grammar*. Doktoretza-tesia, Dublin City University, Dublin, Ireland, 2009.

Diab M.T. Improved arabic base phrase chunking with a new enriched pos tag set. *Semitic '07: Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages*, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

Díaz de Ilarraza A., Gojenola K., eta Oronoz M. Reusability of a corpus and a treebank to enrich verb subcategorisation in a dictionary. *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP07)*, 280–284, Borovets, Bulgaria, 27-29 September 2007.

Dietterich T.G. *Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms*, 1895–1924. Number 10 in 7. MIT press journals, 1998.

Ehrlich E. *Theory and Problems of Punctuation, Capitalization and Spelling*. McGraw-Hill, 1992.

Ejerhed E. Finding clauses in unrestricted text by finitary and stochastic methods. *Proceedings of Second Conference on Applied Natural Language Processing*, 219–227, 1988.

Ejerhed E. Finite state segmentation of discourse into clauses. *Proceedings of the ECAI'96 Workshop on Extended finite state models of language*, Budapest, Hungary, 1996.

Elkan C. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

Erdozia K., Laka I., Mestres-Misse A., eta Rodriguez-Fornells A. Syntactic complexity and ambiguity resolution in a free word order language: behavioral and electrophysiological evidences from basque. *Brain and Language*, 1–17, 2009.

Euskaltzaindia. *Euskal Gramatika. Lehen Urratsak-I.* Euskaltzaindia, Burlata, 1985.

Euskaltzaindia. *Euskal Gramatika. Lehen Urratsak-II.* Euskaltzaindia, Bilbo, 1987.

Euskaltzaindia. *Euskal Gramatika. Lehen Urratsak-III.* Euskaltzaindia, Bilbo, 1990.

Euskaltzaindia. *Euskal Gramatika Laburra.* Euskaltzaindia, Bilbo, 1993.

Euskaltzaindia. *Euskal Gramatika. Lehen Urratsak-IV.* Euskaltzaindia, Bilbo, 1994.

Euskaltzaindia. *Euskal Gramatika. Lehen Urratsak-V.* Euskaltzaindia, Bilbo, 1999.

Everitt B. *The analysis of contingency tables.* Chapman and Hall, 1992.

Ezeiza N. *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile sintaktiko sendo eta malgua.* Doktoretza-tesia, University of the Basque Country, Donostia, 2002.

Fellbaum C. A semantic network of English verbs. In Fellbaum C., editor, *WordNet: An Electronic Lexical Data-base.* MIT Press, 1998.

Fernandes E.R., dos Santos C.N., eta Milidiu R.L. A machine learning approach to portuguese clause identification. *Computational Processing of the Portuguese Language*, 2010.

Foster J. eta Andersen O. Generrate: Generating errors for use in grammatical error detection. *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, 82–90, Boulder, Colorado, USA, June 2009. Association for Computational Linguistics.

Francis W.N. eta Kucera H. *Brown Corpus Manual.* Department of Linguistics, Brown University, 1979.

Freund Y. eta Schapire R.E. A decision theoretic generalizationof on-line leraning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):297–336, 1997.

Freund Y. eta Schapire R.E. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

Gala N. Using the incremental finite-state architecture to create a spanish shallow parser. *Procesamiento de Lenguaje Natural*, 1999.

Garzia J. *Joskera lantegi*. Euskal Autonomia Erkidegoko Administrazioa, IVAP, 1997.

Gazdar G., Klein E., Pullum G., eta Sag I. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.

Gentry C., Ramzan Z., eta Stubblebine S. Secure distributed human computation. *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, 155–164, New York, NY, USA, 2005. ACM.

Gisbert J.M.B. Análisis de errores, problemas y categorización. *DICENDA Cuadernos de Filología Hispánica 16*, 1998.

Goenaga P. *Gramatika bideetan*. Erein, Donostia, 1980.

Gojenola K. *Euskararen sintaxi konputazionalerantz. Oinarrizko baliabideak eta beren aplikazioa aditzen azpikategorizazio-informazioaren erauzketan eta erroreen tratamenduan*. Doktoretza-tesia, Informatika Fakultatea. Euskal Herriko Unibertsitatea, Donostia, 2000.

Goldberg Y., Adler M., eta Elhadad M. Noun phrase chunking in hebrew: influence of lexical and morphological features. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Golding A.R. A Bayesian hybrid method for context-sensitive spelling correction. In Yarovsky D. eta Church K., editors, *Proceedings of the Third Workshop on Very Large Corpora*, 39–53, Somerset, New Jersey, 1995. Association for Computational Linguistics.

Golding A.R. eta Roth D. Applying Winnow to Context-Sensitive Spelling Correction. *Proc. 13th International Conference on Machine Learning*, 182–190. Morgan Kaufmann, 1996.

Golding A.R. eta Roth D. A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, 34(1-3):107–130, 1999.

Guinovart F.J.G. Aportaciones a la metodología de evaluación de los sistemas de verificación automática de la sintaxis. *Procesamiento del Lenguage Natural*, 7–13, 1996a.

Guinovart F.J.G. *Fundamentos y límites de los sistemas de verificación automática de la sintaxis y el estilo*. Doktoretza-tesia, Universidade de Santiago de Compostela, 1996b.

HABE. *Helduen euskalduntzerako programazioa*. HABE, 1981.

Hagen K., Johannessen J., eta Noklestad A. A constraintbased tagger for norwegian. *In: Lindberg, C.E. og Lund, S.N. (red.): 17th Scandinavian Conference of Linguistic, Odense. Odense Working Papers in Language and Communication*, 1(19), 2000.

Hardt D. Comma checking in Danish. *Corpus Linguistics*, Lancaster (England), 2001.

Hashemi S.S., Cooper R., eta Andersson R. Positive grammar checking: A finite state approach. *Computational Linguistics and Intelligent Text Processing, 4th International Conference, CICLing 2003, Mexico City, Mexico, February 16-22*, 2588 lib. of *Lecture Notes in Computer Science*, 635–646. Springer, 2003.

Heyan H. eta Zhaoxiong C. The hybrid strategy processing approach of complex long sentence. *In Journal of Chinese Information*, 2002.

Hidalgo B. *Hitzen ordena euskaraz*. Doktoretza-tesia, Euskal Herriko Unibertsitatea, 1994.

Hill R. eta Murray W. Commas and spaces: the point of punctuation. *Annual CUNY conference on Human Sentence Processing*, New Jersey. USA, 1998.

Huddleston R. eta Pullum G.K. *The Cambridge Grammar of the ENglish Language*. Press syndicate of the university of Cambridge, 2002.

Hudson R. *English Word Grammar*. Oxford, 1990.

Izumi E., Uchimoto K., Saiga T., Supnithi T., eta Isahara H. Automatic error detection in the Japanese learners' English spoken data. *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, 145–148, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

Jakubicek M. eta Horak A. Punctuation detection with full syntactic parsing. *Proceedings of CICLing-2010. 11th International Conference on Intelligent Text Processing and Computational Linguistics*, Romania, 2010.

James C. *Errors in Language Learning and Use*. Applied Linguistics and Language Study. Macquarie University, Sydney, 1998.

Jarvie G. *Chambers Punctuation Guide*. UK:Chambers, 1992.

Jin M., Kim M., Kim D., eta Lee J. Segmentation of Chinese long sentences using commas. *Proceedings of ACL*, 2004.

Joachims T. Text categorization with support vector machines: learning with many relevant features. In Nédellec C. eta Rouveirol C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*. Springer, 1998.

Johannessen J.B., Hagen K., eta Lane P. The performance of a grammar checker with deviant language input. *Proceedings of the 19th international conference on Computational linguistics*, 1–8, COLING, Taipei, Taiwan, 2002. Association for Computational Linguistics.

Jones B. Towards a syntactic account of punctuation. *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen (Denmark), 1996a.

Jones B. Towards testing the syntax of punctuation. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz. California. USA, 1996b.

Jones M. eta Martin J. Contextual spelling correction using latent semantic analysis. *Proceedings of the fifth conference on Applied natural language processing*, 166–173, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

Jurafsky D. eta Martin J.H. *Speech and Language Processing. An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, USA, 2000.

Kárason O. Detecting grammatical errors with memory-based learning, 2005.

Karlsson F., Voutilainen A., Heikkila J., eta Anttila A. *Constraint Grammar: Language-independent System for Parsing Unrestricted Text*. Prentice-Hall, Berlin, 1995.

Karttunen L., Gaál T., eta Kempe A. Xerox finite state tool. Barne-txostena, Xerox Research Centre Europe, 1997.

Kim J. eta Woodland P. The use of prosody in a combined system for punctuation generation and speech recognition. *Proceedings of Eurospeech*, Aalborg, Denmark, 2001. International Speech Communication Association.

Koskenniemi K. *Two-level Morphology: a general computational model for word-form recognition and production*. University of Helsinki, Helsinki, 1983.

Kübler S., McDonald R., eta Nivre J. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies; Morgan and Claypool Publishers, 2009.

Kudo T. eta Matsumoto Y. Chunking with support vector machines. *Proceeding of NAACL 2001*, Pittsburgh, PA, USA, 2001.

Kukich K. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, December 1992.

Labaka G. *EUSMT: Incorporating Linguistic information to Statistical Machine Translation for a morphologically rich language. Its use in preliminary SMT-RBMT-EBMT hybridization*. Doktoretza-tesia, University of the Basque Country, Donostia, 2010.

Laka I. *A Brief Grammar of Euskara, the Basque Language*. Euskararako Errektoreordetza, EHU, 1996.

Lee H.J., Park S.B., Lee S.J., eta Park S.Y. Clause boundary recognition using support vector machines. *PRICAI 2006: Trends in Artificial Intelligence*, 2006.

Lee J. eta Seneff S. Correcting misuse of verb forms. *Proceedings of the 46th ACL*, Columbus, USA, 2008.

Lee Y. eta Wu Y. A robust multilingual portable phrase chunking system. *Expert Syst. Appl.*, 33(3), 2007.

Lee Y.H., Kim M.Y., eta Lee J.H. Chunking using conditional random fields in korean texts. *Natural Language Processing IJCNLP05. Lecture Notes in Computer Science*, 2005.

Leech G., Barnett R., eta Kahrel P. Recommendations for the syntactic annotation of corpora. http://www.ilc.cnr.it/EAGLES96/browse.html, 1996.

Lenci A., Montemagni S., eta Pirrelli V. Chunk-it. an italian shallow parser for robust syntactic annotation. *In Linguistica Computazionale*, 2001.

Lewis D., Yang Y., Rose T., eta Li F. Rcvi: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 2004.

Li X. eta Roth D. Exploring evidence for shallow parsing. *Proceedings of 5th Conference on Computational Natuarl Language Learning*, 2001.

Li X., Zong C., eta Hu R. A hierarchical parsing approach with punctuation processing for long Chinese sentences. *Proceedings of ACL*, 2002a.

Li Y., Bontcheva K., eta Cunningham H. Adapting svm for data sparseness and imbalance: a case study in information extraction. *Natural language Engineering*, 2009.

Li Y. eta Shawe-Taylor J. The svm algorithm with uneven margins and Chinese document categorization. *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*, Singapore, 2003.

Li Y., Zaragoza H., Herbrich R., Shawe-Taylor J., eta Kandola J. The perceptron algorithm with uneven margins. *ICML02: proceedings of the 19th international conference on Machine Learning*, San Francisco, USA, 2002b. Morgan Kaufmann Publishers Inc.

Liang Y., Wang N., Qiu Z., Yin-Chen, eta Zhao T. A divide-conquer strategy for both english and chinese text chunking. *Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology*, Washington, DC, USA, 2007. IEEE Computer Society.

Lin D. Dependency-based evaluation of MINIPAR. *1st International Conference on Language Resources and Evaluation*, Granada, Spain, 1998.

Loftsson H. Iceparser: An incremental finite-state parser for icelandic. *In Proceedings of NoDaLiDa*, 2007.

Lopez De Lacalle O. *Domain Specific Word Sense Disambiguation*. Doktoretza-tesia, University of the Basque Country, Donostia, 2009.

Manning C.D. eta Schutze H. *Foundations of statistical natural language processing*. The MIT Press, 2003.

Marcus M., Marcinkiewicz M.A., eta Santorini B. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2), 1993.

Maritxalar M. *MUGARRI: Bigarren hizkuntzako ikasleen hizkuntz ezagutza eskuratzeko sistema anitzeko ingurunea*. Doktoretza-tesia, Informatika Fakultatea. Euskal Herriko Unibertsitatea, Donostia, 1999.

Maritxalar M., Díaz de Ilarraza A., eta Oronoz M. From psycholinguistic modelling of interlanguage in second language acquisition to a computational model. *Proceedings of Computational Natural Language Learning. In conjunction with ACL/EACL*, 50–59, Madrid, Spain, 11-12th July 1997.

Màrquez L. Aprendizaje automático y procesamiento del lenguaje natural. *Tratamiento del lenguaje natural*, page 207, 2002.

Martinez D. *Supervised Word Sense Disambiguation: Facing Current Challenges*. Doktoretza-tesia, University of the Basque Country, Donostia, 2004.

Mayor A., naki Alegria I., de Ilarraza A.D., Labaka G., Lersundi M., eta Sarasola K. Evaluación de un sistema de traducción automática basado en reglas o por qué bleu sólo sirve para lo que sirve. *Revista de la Asociación Española para el Procesamiento del Lenguaje Natural*, 2009.

Meyer C. *A Lingistic Study of American Punctuation*. Peter Lang Publishing Co., 1987.

Milenova B., Yarmus J., eta Campos M. Svm in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. *Proceeding of the 31st VLDB Conference*, Trondheim, Norway, 2005.

Milidiu R.L., Santos C.N.D., eta Duarte J.C. Phrase chunking using entropy guided transformation learning. *Proceedings of ACL08*, Ohio, USA, 2008. Association for Computational Linguistics.

Mitchell M., editor. *Machine Learning*. New York: McGraw-Hill, 1997.

Mitkov R., editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.

Molina A. *Desambiguación en procesamiento del lenguaje natural mediante técnicas de aprendizaje automático*. Doktoretza-tesia, Universidad Politécnica de Valencia, Valencia, 2003.

Molina A. eta Pla F. Clause detection using hmm. In Daelemans W. eta Zajac R., editors, *Proceedings of CoNLL*, 70–72. Toulouse, France, 2001.

Molina A. eta Pla F. Shallow parsing using specialized hmms. *J. Mach. Learn. Res.*, 2:595–613, 2002.

Mooney R.J. Machine learning. *The Oxford Handbook of Computational Linguistics*, 376–394, 2003.

Moré J., Climent S., eta Oliver A. A grammar and style checker based on Internet searches. *Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004*, 1931–1934, Lisbon, Portugal, 2004.

Motkhtar S.A., Chanod J., eta Roux C. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2-3):121–144, 2002.

Muresan S., Tzoukermann E., eta Klavans J.L. Combining linguistic and machine learning techniques for email summarization. *Proceedings of the 2001 workshop on Computational Natural Language Learning*. Association for Computational Linguistics, 2001.

Mrisep K. Parsing estonian with constraint grammar. *In Proceedings of nodalida*, 2001.

Naber D. *A Rule-Based Style and Grammar Checker*. Doktoretza-tesia, Technische Fakultät. Universität Bielefeld, 2003.

Nguyen V.V., Nguyen M.L., eta Shimazu A. Clause splitting with conditional random fields. *Information and Media Technologies*, 4(1):57–75, 2009.

Nivre J., Hall J., Kübler S., McDonald R., Nilsson J., Riedel S., eta Yuret D. The CoNLL 2007 shared task on dependency parsing. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, 915–932, Prague, Czech Republic, June 2007a. Association for Computational Linguistics.

Nivre J., Hall J., Nilsson J., Chanev A., Eryiğit G., Kübler S., Marinov S., eta Marsi E. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007b.

Norrish J. *Language learners and their errors*. Macmillan Press, London, 1983.

Nunberg G. *The linguistics of Punctuation*. Center for the study of language information (CSLI), Lecture notes: no. 18, University of Chicago Press, 1990.

Odriozola J. eta Zabala I. *Hitz-ordena, galdegaia eta komaren erabilera*. Euskal Herriko Unibertsitateko Argitarapen Zerbitzua, Bilbo, 1993.

Oflazer K. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 29(4):515–544, 2003.

Okanohara D. eta Tsujii J. A discriminative language model with pseudo-negative samples. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 112–121, 2007.

Oronoz M. *Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura*. Doktoretza-tesia, Lengoaia eta Sistema Informatikoak Saila, Euskal Herriko Unibertsitatea, 2009.

Otegi A. Estilo kontuak eta puntuazio ikurren erroreak detektatzeko sistema. Barne-txostena, Euskal Herriko Unibertsitatea, 2003.

Otegi A. Zuzentzaile sintaktikoa Word-en integratzeko liburutegi baten sorkuntza. Barne-txostena, Euskal Herriko Unibertsitatea, 2006.

Palomar M., Civit M., Díaz de Ilarraza A., Moreno L., Bisbal E., Aranzabe M., Ageno A., Martí M.A., eta Navarro B. 3LB: Construcción de una base de árboles sintáctico-semánnticos para el Catalán, Euskera y Castellano. *Actas del XX Congreso de la SEPLN*, 2004.

Patrick J.D. eta Goyal I. Boosted decision graphs for nlp learning tasks. In Daelemans W. eta Zajac R., editors, *Proceedings of CoNLL*, 58–60. Toulouse, France, 2001.

Paxson W. *The Mentor Guide to Punctuation*. New York: Mentor Books, 1986.

Pociello E. *Euskararen ezagutza-base lexikala: Euskal WordNet*. Doktoretza-tesia, Euskal Herriko Unibertsitatea, 2008.

Pollard C. eta Sag I.A. *Head-driven phrase structure grammar*. Chicago: University of Chicago Press, 1994.

Punyakanok V. eta Roth D. The use of classifiers in sequential inference. *The Conference on Advances in Neural Information Processing Systems*, 2001.

Puscasu G. A multilingual method for clause splitting. In Lee M., editor, *Proceedings of CLUK 2004*, 199–206. University of Birmingham, UK, 2004.

Quirk R., Greenbaum A., Leech G., eta Svartvik J. *A Comprehensive Grammar of the ENglish Language*. Longman, 1985.

Quixal M. eta Badia T. Exloiting unsupervised techniques to predict EFL learner errors. *CALICO AALL Workshop*, 2008.

Rabiner L. eta Juang B. An introduction to Hidden Markov Models. *IEEE ASSP magazine*, 1986.

Ram R.V.S. eta Devi S.L. Clause boundary identification using conditional random fields. *Computational Linguistics and Intelligent Text Processing*, 4919:140–150, 2008.

Rehurek R. eta Sojka P. Automated classification and categorization of mathematical knowledge. *Intelligent Computer Mathematics*, 2010.

Rey M.A.M.D. *Análisis de Errores de la Interlengua de Español en Estudiantes Italianos*. Elenet.org, 2004.

Reynaert M. Text induced spelling correction. *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 834, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

Ringger E., Haertel R., eta Tomanek K., editors. *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, Boulder, Colorado, June 2009.

Roche E. Finite state transducers: Parsing free and frozen sentences. *Extended Finite State Models for Language*, 1999.

Sampson G. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press, 1995.

Sang E.T.K. eta Buchholz S. Introduction to the conll-2000 shared task: Chunking. *Proceedings of Computational Natural Language Learning*, Lisbon (Portugal), 2000.

Sang E.T.K. eta Déjean H. Introduction to the conll-2001 shared task: Clause identification. *Proceedings of Computational Natural Language Learning*, Toulouse (France), 2001.

Santos D. Punctuation and multilinguality: Some reflections from a language engineering perspective. *Working papers in Applied Linguistics*, Oslo, Norway, 1998.

Say B. eta Akman V. Current approacehs to punctuation in Computational Linguistics. *Computers and the Humanities*, 30(6):457–469, 1996.

Schiehlen M. A cascaded finite-state parser for german. *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, 163–166, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

Schmid H. eta Walde S.I. Robust german noun chunking with a probabilistic context-free grammar. *Proceedings of the 18th conference on Computational linguistics*. Association for Computational Linguistics, 2000.

Selinker L. Language transfer. *General Linguistics*, 1969.

Selinker L. Interlanguage. *Error analysis: Perspectives on Second Language Acquisition*, 1974.

Sha F. eta Pereira F. Shallow parsing with conditional random fields. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, HLT-NAACL*, 2003.

Shen H. eta Sarkar A. Voting between multiple data representations for text chunking. *Proceedings of the eighth meeting of the Canadian society for computational intelligence*, Canada, 2005.

Shi Y. eta Zhou L. Error detection using linguistic features. *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 41–48, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

Shi Y. eta Wang M. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, 1707–1712, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Shieber S.M. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes, Stanford, 1986.

Shieber S.M. eta Tao X. Comma restoration using constituency information. *Proceedings of HLT-NAACL*, 2003.

Sjöbergh J. Chunking: an unsupervised method to find errors in text. *Proceedings of NODALIDA 2005*, Joensuu, Finland, 2005.

Sjöbergh J. eta Knutsson O. Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. *Proceedings of RANLP 2005*, 506–512, Borovets, Bulgaria, 2005.

Smith N.A. eta Eisner J. Guiding unsupervised grammar induction using contrastive estimation. *Proceedings of the IJCAI Workshop on Grammatical Inference Applications*, Edinburgh, Scotland, 2005a.

Smith N.A. eta Eisner J. Training log-linear models on unlabeled data. *Proceedings of the 43rd ACL*, Ann Arbor, USA, 2005b.

Srihari R. eta Li W. Information extraction supported question answering. *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, 1999.

Srikumar V., Reichart R., Sammons M., Rappoport A., eta Roth D. Extraction of entailed semantic relations through syntax-based comma resolution. *Proceedings of ACL-08: HLT*, 1030–1038, Columbus, Ohio, June 2008. Association for Computational Linguistics.

StatSoft. *Electronic Statistics Textbook*. Tulsa, OK: StatSoft, 2007.

Tanev H. eta Mitkov R. Shallow language processing architecture for bulgarian. *Proceedings of the 19th international conference on Computational linguistics*, 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

Tapanainen P. *The Constraint Grammar parser CG-2*. Publications of the University of Helsinki, 27, Helsinki, 1996.

Tillmann C. eta Ney H. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1), 2003.

Tjong Kim Sang E.F. Memory-based clause identification. In Daelemans W. eta Zajac R., editors, *Proceedings of CoNLL*, 67–69. Toulouse, France, 2001.

Tjong Kim Sang E.F. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *Proceedings of CoNLL*, 155–158. Taipei, Taiwan, 2002.

Tjong Kim Sang E.F. eta De Meulder F. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans W. eta Osborne M., editors, *Proceedings of CoNLL*, 142–147. Edmonton, Canada, 2003.

Torijano J. *Errores de aprendizaje, aprendizaje de los errores*. Arco Libros S.L., Madrid, 2004.

Truss L. *Eats, shoots and leaves: the zero tolerance approach to punctuation*. London: Profile books, 2003.

Tsao N. eta Wible D. A method for unsupervised broad-coverage lexical error detection and correction. *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, 51–54, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Uria L. *Euskarazko erroreen eta desbideratzeen analisirako lan-ingurunea. Determintzaile-erroreen azterketa eta prozesamendua*. Doktoretza-tesia, Euskal Filologia Saila, Euskal Herriko Unibertsitatea, 2009.

Urkia M. *Euskal morfologiaren tratamendu informatikorantz*. Doktoretza-tesia, Filologia eta Historia-Geografia Fakultatea. UPV-EHU, uztaila 1997. Miren Azkarate, UPV-EHUko irakaslearen zuzendaritzapean eginiko tesia.

Vicedo J. *SEMQA: Un modelo semántico aplicado a los sistemas de Búsqueda de Respuestas*. Doktoretza-tesia, Universidad de Alicante, Alicante, 2002.

Voutilainen A., Heikkil J., Fries P.I.U., Tottie G., eta Schneider P. An English Constraint Grammar (engcg) a surface-syntactic parser of English, 1993.

Wagner J., Foster J., eta van Genabith J. A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 112–121, 2007.

Wahlster W., editor. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, 2000.

Witten I. eta Frank E. *Data Mining. Practical Machine Learning Tools and Techniques with Java Inplementations*. Morgan Kaufmann Publishers, 2005.

Zelaia A., na I.B., eta Yurramendi Y. LSAren oinarri matematikoa. *Ekaia*, 85–105, 2003.

Zhang J. eta Mani I. knn approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of the ICML workshop on learning form imbalanced datasets*. Association for Computing Machinery, 2003.

Zhang T., Damerau F., eta Johnson D. Text chunking based on a generalization of winnow. *Journal of Machine Learning Research*, 2:615–637, 2002.

Zhou J., Zhang Y., Dai X., eta Chen J. Chinese event descriptive clause splitting with structured svms. *Proceedings of CICLing-2010. 11th International Conference on Intelligent Text Processing and Computational Linguistics*, Romania, 2010.

Zong C., Zhang Y., Yamamoto K., eta Sakamoto M. Utterance paraphrasing for spoken language translation. *In Journal of Chinese Language Computing*, 2002.

Zribi C.O., Mejri H., eta Ahmed M. Combining methods for detecting and correcting semantic hidden errors in arabic texts. *CICLing '07: Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, 634–645, Berlin, Heidelberg, 2007. Springer-Verlag.

Zubimendi J. *Ortotipografia*. Eusko Jaurlaritzaren Argitalpen Zerbitzua, Gasteiz, 2004.

Zubimendi J. eta Esnal P. *Idazkera-liburua*. Eusko Jaurlaritzaren Argitalpen Zerbitzua, Gasteiz, 1993.

Zubiri I. eta Zubiri E. *Euskal Gramatika Osoa*. Didaktiker, Bilbo, 1995.

# Glosategia

**Active learning (*ikasketa bizi*a)**

Active learning edo *ikasketa bizi* delakoaren muina, etiketatu gabeko ikasketa-corpus batean, etiketatzeko adibide multzo egokia aukeratzean datza. Izan ere, ikasketa-corpuseko adibide guztiak ez dira neurri berean erabilgarriak. Ikasketa automatikoko algoritmoarentzat baliagarrienak diren adibideak eskuz etiketatuz, emaitza hobeak lor litezke, zoriz aukeratutako adibideak etiketatuz lortutakoak baino.

**AdaBoost**

Sailkatzaile *ahulak* uztartuz zehaztasun handiko sailkatze-erregelak eskuratzen dituen metodo orokorra da.

**Adibideak**

Ikasketa automatikoaren baitan, adibideei instantzia ere deitzen zaie. Ikasi beharreko kontzeptuari buruzko adibide bakar eta independentea da instantzia bakoitza, eta aurrez definitutako atributuentzat balioak ditu. Hau da, adibide bakoitzak balio bat izango du kontzeptua ikasteko beharrezkotzat jotako ezaugarri —atributu— bakoitzeko (eta baita ikasi beharreko kontzeptuari dagokion balioa ere, oro har.)

**Agerkidetza (*co-ocurrence*)**

Dokumentu batean bi termino edo gehiago —zorizkoa baino handiagoa den probabilitate batekin— elkarren ondoan izateari deritzo.

**Aposizioa**

Bi osagai elkarren segidan jarriz egindako eraikuntza, bigarrenak aurrekoa azaltzen edo zehazten duelarik. Azalpenezko sintagma bat izan ohi

da, eta bi motakoak izan daitezke: murriztaileak eta ez-murriztaileak. Aposizio murriztailea da izen sintagmaren barnean gertatzen dena, izen bati beste izen bat –edo gehiago– lotzen zaionean. Hortaz, izen sintagma bakarra osatzen du hitz-segida guztiak (*Unibertsitateko Errektoreorde Edurne Mendiluzek*). Aposizio ez-murriztaileak, berriz, izen sintagma osoak beste hitz-segida batekin nolabait parekatuz egiten dira (*Matematikako irakasle berria,* iazkoaren ordez etorri dena*, oso atsegina da.*).

**Atributuak**

Ikus *ezaugarriak*.

**Bigram**

*Bigram* deitzen zaie corpusean elkarren segidan agertzen diren hitz-pareei.

**Boosting**

Ikasketa automatikoko algoritmo bat da, zeinak hainbat ikasketa-algoritmo *ahul* konbinatzen baititu, algoritmo *sendo* bat sortzeko.

**Cambridge Learner Corpus**

30 milioi hitzetik gorako ingeleseko ikasleen corpusa da, eta 135.000 ikasle baino gehiagoren ingeleseko azterketez osatuta dago. *Cambridge International Corpus (CIC)* corpusaren parte da. *Cambridge University Press* eta *Cambridge ESOL* elkarteek sortu zuten.

**Clustering (*multzokatzea*)**

Ikasketa ez-gainbegiratua egiteko teknika bat da. Ikasketa-corpuseko instantziak sailkatu gabe ditugunean —euren klasea ezagutzen ez dugunean, alegia—, instantzia horiek euren arteko zenbait antzekotasunen arabera bil daitezke; horrela lortutako multzo bakoitza klase bat edo *cluster* bat dela esaten da.

**Conditional Random Fields**

Markov-en eredu ezkutuen antzeko eredu estokastiko bat da. Oro har, datu-sekuentzia bat emanik, eredu honek etiketa bana esleitzen dio osagai bakoitzari. Markov-en eredu ezkutuek etiketen eta behaketen

probabilitateen distribuzioa batera kalkulatzen dute; *Conditional Random Fields* izenekoetan, berriz, behaketek baldintzatzen dute etiketen sekuentzia zuzenaren probabilitatea. Hala, Markov-en eredu ezkutuetan egiten den independentzia-hipotesia erlaxatzen dute *Conditional Random Fields* izenekoek.

**Cross-validation**

Ikasketa-algoritmoak ebaluatzeko teknika bat da. Datuak bi zatitan banatu beharrean (esaterako, % 70 ikasketarako eta % 30 testerako), norberak erabakitzen du zenbat zatitan banatu; adibidez, hamar zatitan. Orduan, ikasketarako bederatzi zati erabiltzen dira, eta testerako bakarra; baina hamar aldiz errepikatzen da prozesu hau, eta aldi bakoitzean testerako erabiltzen den zatia aldatuz doa. Hala, probaren bukaeran, instantzia bakoitza behin bakarrik erabili da ebaluaziorako.

**Entropia**

Zorizko aldagai batek har ditzakeen balioen ziurgabetasunaren neurria. X zorizko aldagai batek balio desberdinak har ditzake esperimentu baten zenbait errepikapenetan. X aldagaiaren balio batzuk besteak baino probableagoak direnez, X-ren balioen banaketa probabilistikoa esperimentuaren araberakoa da. X aldagaiaren entropia X-ren banaketa probabilistikoari hertsiki lotua dago, eta banaketa zein laua den adierazten du, gainera. Distribuzio bat laua da (entropia handia du), baldin eta X-ren balio guztiek antzeko probabilitatea baldin badute; aitzitik, ez da oso laua, baldin eta X-ren balio batzuek besteek baino probabilitate handiagoa badute. Hala, distribuzio laua dugun kasuetan (entropia handikoetan, alegia), zaila da X-ren balioa zein izango den aurreikustea, X-ren balio guztiek antzeko probabilitatea dutelako, hain zuzen.

**Entropia handienaren printzipioa**

Entropia handienaren printzipioak, neurri batean, aldagai batek har ditzakeen balio guztien probabilitate berdintsua bilatzen du; alegia, ez egotea balio bat beste bat baino gehiago gertatzeko probabilitaterik.

**Erabaki-zuhaitzak (*decission trees*)**

Ikasketa automatikoko eskema hau *zatitu eta irabazi* teknikan oinarritzen da, eta, grafikoki adierazita, zuhaitz baten itxura hartzen du.

Erabaki-zuhaitzak sortzeko, lehendabizi, atributu bat aukeratzen da erro-adabegian kokatzeko, eta bere balio posible bakoitzeko adar bat egiten da. Gero, prozesua errepika daiteke errekurtsiboki, adar bakoitzerako, baina adar bakoitzeko baldintzak bete dituzten adibideekin soilik. Adabegi-ume bakoitzeko adibide guztiek sailkapen bera dutenean amaitzen da prozesua. Eskema honetan, unean uneko atributuaren aukeraketak berebiziko garrantzia dauka: atributu bat erabili ondoren, ez da gerora hartuko diren erabakietan berriz erabiliko. Bestalde, geroz eta atributu gehiago izan, orduan eta denbora gehiago behar du algoritmoak.

**Esaldia**

Komunikazio-mezu oso bat osatzen duen hitzen multzoa da esaldia, eta perpaus batez edo gehiagoz osatua egon daiteke. Eginkizun konputazionaletarako, puntuazio-markak hartzen dira aintzat esaldia mugatzeko. Hala, esaldia kontsideratzen da puntuazio *gogor* artean dagoen oro, puntuazio *gogor* gisa harturik puntua, harridura-marka eta galdera-marka (hiru puntuak, puntu eta koma eta bi puntuak ere hala kontsideratzen dira, kasu batzuetan). Esaldi barruko aditz bakoitzeko, alabaina, perpaus bat dugula esango dugu.

**Estokastikoak (eredu estokastikoak)**

Atazan inplikatutako atributuen dependentzia probabilistikoak deskribatzen dituzte eredu estokastikoek, graforen baten bidez normalean. Grafoko adabegi bakoitzak zorizko aldagai bat adierazten du eta, gainera, probabilitate-banaketa bat du esleituta. Banakako banaketa hauen bidez, behatutako adibide guztien baterako banaketa kalkula daiteke.

**Eustagger**

IXA taldean sortutako euskarako analizatzaile/desanbiguatzaile morfosintaktiko automatikoa.

**Ezaugarriak**

Ikasketa automatikoaren baitan erabiltzen den terminoa da ezaugarriena ere (*features* deituak, ingelesez). Ikasi nahi den kontzeptuarentzat beharrezkotzat jotako informazio mota adierazten dute. Adibide edo instantzia bakoitzak balio bat izango du ezaugarri —atributu— bakoitzeko.

**Ezkutuko semantikaren analisia (*Latent Semantic Analysis*)**

Testu idatzien semantika adierazteko gaitasuna duen tresna bat da. Modu matematikoan adierazten ditu testuko paragrafo eta hitzak. Ondoren, adierazpen matematiko horren gainean zenbait eraldaketa burutzen ditu, eta horrela, testuen eta bertan dauden hitzen arteko erlazio semantikoak neurtzeko gai da.

**FR-Perceptron**

*Pertzeptroi*en algoritmo tradizionalaren orokortze bat da erroreak gidatutako ikasketa automatikoko algoritmo hau, eta testu bateko *hitz multzoak* identifikatzea du xede. Identifikazio-prozesuan, algoritmoak iteratu egiten du $n$ aldiz; alegia, ikasketa-corpuseko adibide bakoitza $n$ aldiz bisitatzen da. *Start* eta *end* funtzioak esaldiko hitz bakoitzeko aplikatzen dira lehendabizi, eta *score* funtzioaren sarrera izango diren *hitz multzo*en hautagaiak definitzen dira honela. Gero, *score* funtzioa aplikatzen zaio, modu errekurtsiboan, hautagai bakoitzari. Honela, *hitz multzo*en konbinazio onena aukeratzen da esaldiko. Egindako iragarpena okerra baldin bada, sailkatzaileak zuzentzen dira hurrengo iterazioako, erregela sinple batzuen bidez.

**Galdegaia**

Oro har, aditzaren aurretik doan sintagma. Esaldian azpimarratu nahi dena adierazten du.

**Hitz multzo (*phrase*)**

Elkarren segidan doazen hitzen edozein sekuentzia, zeinak bi propietate beteko dituen: *hitz multzo* batek beste *hitz multzo* bat hartu ahal izango du bere baitan, baina *hitz multzo* bat ezingo zaio beste bati gainjarri. Kateek eta perpausek, besteak beste, propietate hauek betetzen dituzte, eta *hitz multzo*ak izango dira, beraz.

**Ikasketa erdi-gainbegiratua**

Oinarrizko corpus bat etiketatuz, ikasketa gainbegiratua egiten da lehendabizi, eta horrekin lortutako sailkatzailea corpus berri erraldoia (etiketatu gabea) etiketatzeko erabiltzen da. Gero, corpus erraldoi hori (sailkatzaileak etiketatua) ikasketa-corpus gisa baliatzen da.

**Ikasketa ez-gainbegiratua**

Sailkatu gabeko adibide batzuetatik ikasteko prozesuari deitzen zaio. Horregatik, sarritan, *multzokatze* edo *clustering* ataza moduan ulertzen da ikasketa mota hau.

**Ikasketa gainbegiratua**

Sailkatuta dauden adibide batzuetatik ikasiz, sailkatu gabe dauden adibide berriak sailkatzea.

**Instantziak**

Ikus *Adibideak*.

**Kate (*chunk*)**

Katea sintagma kategoriako zatia da eta, sintaktikoki erlazionaturiko hitzez osatua dago. Gainjartzen ez diren eta elkarrekin sintaktikoki erlazionaturik dauden hitz multzoak dira kateak. Hitz multzo horiek, gainera, ez-errekurtsiboak izango dira; hau da, ezin dute beren baitan beste hitz multzorik edota katerik izan.

**Kontzeptua**

Ikasketa automatikoaren baitan, ikasketa-prozesuaren emaitza da kontzeptua; alegia, ikasi nahi dugun horixe bera.

**Marjina handieneko hiperplanoa**

Marjina handieneko hiperplanoa bi klaseen artean banaketa handiena ematen diguna da; hots, hiperplanoko alde bateko eta besteko instantziak elkarrengandik urrutien jartzen dituena.

**Markov-en eredu ezkutuak (*Hidden Markov Models*)**

Markov-en ereduek Markov-en propietatea betetzen dute: gertaera bat betetzeko probabilitatea bere berehalako aurreko gertaeraren menpe soilik dago. Automata finitu baten moduan ikus daiteke Markov-en eredu ezkutu bat. Egoerek ereduaren aldagaiak errepresentatzen dituzte, eta arkuek egoera batetik bestera joateko dagoen probabilitatea gordetzen duen etiketa bat dute. Alfabetu bateko sinboloak idaztea baimentzen dute egoerek, probabilitate-funtzio baten arabera. Ezkutua dela esaten da ezin delako jakin ereduak aukeratzen duen egoeren segida, ez bada egoeren segidaren funtzio probabilistiko bat.

**McNemar testa**

McNemar testa bi sailkatzaileen arteko aldea esanguratsua den ala ez erabakitzeko erabiltzen da. Horretarako, corpusa bi zatitan banatu behar da: ikasketa-corpusa eta test-corpusa. Bi sailkatzaileak (A, B) ikasketa-corpus bera erabiliz ikasi ondoren, test-corpus beraren gainean ebaluatu behar dira. Hipotesi nuluaren arabera, A sailkatzaileak ondo eta B sailkatzaileak gaizki sailkatutako adibideen kopuruak A sailkatzaileak gaizki eta B sailkatzaileak ondo sailkatutako adibideen kopuruaren berdina izan behar du. Datu hauen arabera, $\chi^2$ testan oinarritzen da McNemar testa, hipotesi nulu hau uka daitekeen edo ez erabakitzeko. Hipotesia errefusatu baldin badaiteke, bi sailkatzaileen arteko aldea esanguratsua dela esaten da.

**Memorian oinarritutako ikasketa (*Memory Based Learning*)**

Adibideetan oinarritutako ikasketa ere deitua, ikasketa adibide guztiak memorizatzen saiatzen den teknika da, batere erregelarik edo bestelako orokortzerik egin gabe. Adibide berri bat sailkatzeko, adibideen memoriatik sailkatu nahi dugunaren antzekoena den adibide-multzoa hartzen da, eta adibide-multzo horretan gehien ematen den klasea esleitzen zaio.

**Mintzagaia**

Informazio zaharra edo jadanik ezaguna biltzen duten elementuek osatzen dute, edo solasaren gaia finkatzen dutenek. Gure ikuspuntutik, esaldia (edo perpausa) abiatzeko elementu egoki gisa ikusiko dugu mintzagaia; esaldiari (edo perpausari) sarrera egiten diona. Euskaraz, galdegaiaren aurretik joan ohi da.

**Murriztapen-gramatika (*Constraint Grammar*)**

Patroiak identifikatzeko eta etiketak jarri, kendu edo aldatzeko aukera ematen duen formalismoa.

**Naive Bayes**

Sailkatzaile estokastiko sinpleena da. Probabilitatearen banakako banaketan oinarritzen da. Adibide baten klasea asmatzeko, behatutako adibidearen probabilitatea maximizatzen duena aukeratzen da. Horretarako, Bayes-en teorematik eratorritako formula sinple bat erabiltzen da,

non atributu guztiei dagozkien balioak emanik emaitza-atributuaren klase probableena aukeratzen baita.

**Perpaus**

Aditz baten inguruan osatzen den hitzen multzoa da perpausa. Hortaz, aditzak eta aditzari dagozkion elementuek osatzen dute perpausa. Aditza, dena dela, ez da beti testuan esplizituki agertuko; hots, aditza perpausaren ardatza izanagatik ere, aditzaren beraren elipsia egon daiteke, eta horrek ez dio perpaus-izaera kentzen. Bestalde, bi perpaus mota definitzen ditugu: markatuak (menderatuak) eta markatu gabeak. Menderagailua daramatenak izango dira markatuak edo mendeko perpausak, eta markarik ez dutenak perpaus bakunak izango dira.

**Pertzeptroi**

Aukeratutako ezaugarrientzat edo atributuentzat pisu multzo bat ikasten du sailkatzaile lineal sinple honek. Pisu horiek atributu bakoitzaren garrantzia adierazten dute. Sailkatzaile bitarra izan ohi da hau. Sailkapena egiteko, atributu multzoaren konbinazio lineal bat egiten da (normalean sailkatzeko dagoen adibidearen atributuen pisuen batura haztatu bat), eta klase positiboa esleitzen zaio, baldin eta emaitzak muga bat gainditzen badu; bestela, klase negatiboa esleitzen zaio.

**Support Vector Machines (*sostengu-bektoreen makinak*)**

Ikasketa automatikoko algoritmo honek eredu linealek dituzten desabantailak konpontzen ditu. Izan ere, linealak ez diren datuen multzoentzat soluzio bat ematen du. Bere forma sinpleenean, ordea, eredu linealetan oinarritzen da, marjina handieneko hiperplanoa deitzen zaion eredu lineal berezi bat baliatzen baitu, eta hainbat atazatan, eredu lineal hau erabiltzen da. Bi klaseko datuen multzo linealki banagarri bat izanik, esaterako, hiperplano bat —zuzen bat, alegia— aurkitzen du instantzien espazioan, zeinak instantzia guztiak sailkatzen dituen zuzenaren alde batera eta bestera.

**Tartekia**

Lokailu bat edo aposizio ez-murriztaile bat izan daiteke, baina baita esaldiak edo perpausak adierazi nahi duen mezu nagusiari tartekatzen

zaion azalpen gehigarria ere ("*Heriotza da,* Saramagoren iritziz*, Jainkoaren asmatzailea.*").

**Testuinguruaren araberako zuzenketa ortografikoa**

Antzeko bi hitzen artean zuzena aukeratzea helburu duen HPko arloa (ingelesez, *context sensitive spelling correction*). Corpus ustez zuzenak baliatzen dira, bi hitzon artean —dagokion testuinguruan— zuzena zein den erabakitzeko. Inguruko hitzen nolabaiteko zerrenda bat gorde eta antzekotasun gehien dituena aukeratzea da problema hau ebazteko modua.

**Transformazioan oinarritutako ikasketa**

Ikasketa automatikoko algoritmo hau problemaren soluzio sinple batekin hasten da (kategoria etiketatzaile batean, maizen ematen zen kategoria esleitzea, esaterako), eta transformazioko erregela batzuen bidez, soluzio hobea bilatzen da. Transformazioko erregelak sortzeko, soluzio zuzenarekin konparatzen da uneko soluzioa; urrats bakoitzean, aurrekoan sortutako erroreak ondoen konpontzen dituen erregelak sortzen ditu. Transformazio hauek behin eta berriz egin ohi dira, hobekuntzarik lortzen ez den arte.

**Treebank (*zuhaitz-bankua*)**

Analizatutako corpus bat da *treebank*a edo *zuhaitz-bankua*, non esaldi bakoitzaren egitura sintaktikoa adierazia datorren, zenbait etiketaren bidez.

**Weka**

*Datu-meatzaritza*ko atazetarako erabiltzen diren ikasketa automatikoko algoritmoen multzoa da WEKA: http://www.cs.waikato.ac.nz/ml/weka/

**Winnow**

Banatzaile linealen familiakoa, on-line egiten den ikasketa-algoritmo sinple bat da. Ezaugarri edo atributu bitarrekin egiten du lan, eta 2 emaitza-klaseko problemekin (0/1). Adibide berriak sailkatzeko, sarrerako ezaugarri edo atributuen batura haztatu bat egiten da (konbinazio lineala). Emaitza mugaren azpitik badago, 0 itzultzen da; bestela, 1. Gaizki iragarritako adibideek ezaugarri bakoitzari ematen zaion pisua aldatzen dute, ikasketa multzora ahalik eta gehien egokitzeko.

**Xerox Finite State Tool**

Adierazpen erregularrak jaso, eta hauek transduktore bihurtzen dituen tresna. Egoera finituko kalkulua ahalbidetzen duen eragiketa multzo aberatsa du.

# A. ERANSKINA

## Komak zuzentzeko CG erregelak

Komak zuzentzeko sortutako CG erregelak. Esan beharra dago, hala eta guztiz, tesi-lan honetan corpus batean falta diren komak berreskuratzeko soilik erabili dugula gramatika hau.

DELIMITERS = "< $. >" ;

**# KATEGORIAK:**

*# 1. LEGERAKO:*
LIST ADI = ADI ;
LIST ADT = ADT ;
LIST ADL = ADL ;
LIST ADITZA = ADI ADT ADL ;

*# juntagailuak (kategoria LOT, azpikategoria JNT):*
LIST JNT = JNT ;
LIST BAINA = "baina";
LIST DEK = DEK ;
LIST NUMP = NUMP ;

*# 2. LEGERAKO:*
LIST IZE = IZE ;
LIST ADJ = ADJ ;

```
LIST ABS = ABS ;
LIST ERG = ERG ;
LIST DAT = DAT ;
LIST ZERO = ZERO ;

# 3. LEGERAKO:
LIST MODUZ = "moduz";
LIST ETORRI = "etorri";
LIST KAIXO = "kaixo";
LIST AGUR = "agur";
LIST DEIKI_AURREKOAK = MODUZ ETORRI KAIXO AGUR ;
LIST IZB = IZB ;

# 7. LEGERAKO:
# Edozein kategoria:
LIST EDOZEIN_kAT = ADB ADI ADJ ADL ADT DET IOR ITJ IZE LOT
PRT BST SIG ;

# lokailuak (kategoria LOT, azpikategoria LOK):
LIST LOK = LOK ;
LIST ERE = "ere";
LIST NAHIZETA = "nahiz_eta";

# Puntuazioenak:
LIST PUNTU = PUNT_PUNT ;
LIST BIPUNT = PUNT_BI_PUNT ;
LIST KOMA = PUNT_KOMA ;
LIST PKOMA = PUNT_PUNT_KOMA ;
LIST PUNT_HIRU = PUNT_HIRU ;
LIST HARRIDURA = PUNT_ESKL ;
LIST PUNT_GALD = PUNT_GALD ;
LIST PAREN = (≪(¿") (≪)¿") ;
LIST EZK_PAREN =(≪(¿") ;
LIST ESK_PAREN = (≪)¿") ;

LIST PUNTUAZIO_MARKA = PUNT_PUNT PUNT_BI_PUNT PUNT_KOMA
PUNT_PUNT_KOMA PUNT_HIRU PUNT_ESKL PUNT_GALD PAREN;
```

*# 8. LEGERAKO*
LIST LOT = LOT ;
LIST ARREN = "arren";
LIST BEREIZ = BEREIZ ;

*# 1. eta 3. LEGERAKO KOMA SOBRAN*
LIST SIH = SIH ;

# **KOMAK FALTAN**

# KOMAK FALTAN: 1. LEGEA

# DESKR.: Esaldi koordinatuak lotzeko: bi esaldi juntagailu batez
# koordinatzen baditugu, koma jarriko dugu juntagailuaren aurretik.

# ADIB:
# Euria ari zuen, eta etxean geratu nintzen.
# Denboraldi baterako joango da, baina baliteke ez itzultzea.
# Euria egin zuen atzo, eta etxean geratzea erabaki genuen.

# TRATAMENDUA:
# Juntagailua "baina"baldin bada, aurretik koma jarri;
# bestela, koma jarri, baldin eta juntagailuaren aurretik edo ondoren
# aditzen bat baldin badago.
# (deklinabide-kasu bereko bi hitz ez baditu banatzen, baldintza hori
# jarri dugu koma jartzeko, baina "egoera ekonomiko eta politikoak"
# gisako asko daude, eta aldatu dugu baldintza)

# SALBUESPENAK:
# 1) Aditzen edo adberbioen enumerazioa.
# 2) "Salerosketa txiki eta ertainetan"gisako egiturak. Halakoetan,
# deklinabide-kasu bera erabiltzea gomendatzen da, bigarren hitza
# pluralean ez doanean behintzat.

# SALBUESPENAREN ADIB.:
# Euria ari zuen, gogorik ez neukan eta ez nintzen ondo sentitzen;

```
# ez nintzen joan, beraz.
# Ondo eta gaizki iruditzen zait.

MAP (&OKER_KOMA_FALTA_1_1) TARGET EDOZEIN_KAT
IF (1 BAINA + JNT) ;

MAP (&OKER_KOMA_FALTA_1_2) TARGET ADITZA
IF (0 ADITZA)
(NOT 1 BAINA)
(1 JNT) ;

MAP (&OKER_KOMA_FALTA_1_3) TARGET EDOZEIN_KAT
IF (NOT 1 BAINA)
(1 JNT)
(2 ADITZA) ;


# KOMAK FALTAN: 2. LEGEA

# DESKR.: Enumerazioetan, osagaiak komaz bereizten dira.

# ADIB:
# Etxean gordeta zeuzkan diskoak, liburuak eta bideoak.
# Afaltzera denak datoz: Ane, Miren, Jon eta Mikel.

# TRATAMENDUA:
# Bi izen edo bi adjektibo topatzerakoan, deklinabide
# kasu bera badute, koma jarri tartean.
# OHARRA: Deklinabide kasu guztiak gehitu behar. # Guk 3 nagusienekin
probatu dugu: absolutiboa, ergatiboa, datiboa.

# SALBUESPENA: Bi izen edo adjektibo elkarren segidan eta
# deklinabide kasu berarekin daudenean, baina enumerazioa ez denean.
# Kasu gutxitan gertatzen denez, ez diogu garrantzirik emango.

# SALBUESPENAREN ADIB.:
# Jonek Mirenek egin zuen gauza bera egin zuen.
```

# GOMENDIOA:
# Estilo aldetik, txukunagoa da aurreko adibidearen aldaera hau:
# Mirenek egin zuen gauza bera egin zuen Jonek.

MAP (&OKER_KOMA_FALTA_2_1) TARGET IZE
IF (0 IZE + ABS)
(1 IZE + ABS);

MAP (&OKER_KOMA_FALTA_2_2) TARGET ADJ
IF (0 ADJ + ABS)
(1 ADJ + ABS);

MAP (&OKER_KOMA_FALTA_2_3) TARGET IZE
IF (0 IZE + ERG)
(1 IZE + ERG);

MAP (&OKER_KOMA_FALTA_2_4) TARGET ADJ
IF (0 ADJ + ERG)
(1 ADJ + ERG);

MAP (&OKER_KOMA_FALTA_2_5) TARGET IZE
IF (0 IZE + DAT)
(1 IZE + DAT);

MAP (&OKER_KOMA_FALTA_2_6) TARGET ADJ
IF (0 ADJ + DAT)
(1 ADJ + DAT);

# KOMAK FALTAN: 3. LEGEA

# DESKR.: Deikiak komaz markatzen dira.

# ADIB:
# Ongi etorri, Jon.
# Kaixo, Ane.

# TRATAMENDUA:

# "Kaixo", "agur", "moduz", "etorri"hitzen ondoren, pertsona-izen
# berezi bat badator, koma beharko du tartean.
# Hitz horien atzetik ez datozen deiki asko harrapatu gabe geratuko dira.
# Adib.: "Zer moduz ibili zineten, Jon?"

# SALBUESPENAK:
# Izen horiek deikiak ez direnean.
# # SALBUESPENAREN ADIB.:
# Zer moduz Jon?
#

MAP (&OKER_KOMA_FALTA_3_1) TARGET KAIXO
IF (0 KAIXO)
(1 IZB);

MAP (&OKER_KOMA_FALTA_3_2) TARGET AGUR
IF (0 AGUR)
(1 IZB);

MAP (&OKER_KOMA_FALTA_3_3) TARGET MODUZ
IF (0 MODUZ)
(1 IZB);

MAP (&OKER_KOMA_FALTA_3_4) TARGET ETORRI
IF (0 ETORRI)
(1 IZB);

# KOMAK FALTAN: 4. LEGEA

# DESKR.: Estilo zuzeneko esaldiak mugatzeko ere, koma erabiltzen da.

# ADIB.: "Agirretxe sasoi betean dago, eta bera da faboritoa", aitortu du Arregik.

# TRATAMENDUA:
# Ixteko gakotxak topatzean, koma jarri ondoren.

# SALBUESPENAK: Gakotxek esaldi osoa markatzen ez dutenean.

# SALBUESPENAREN ADIB.:
# "Primeran"dagoela esan du Benedettik.

# OHARRA: Salbuespenak kasu errealak baino gehiago
# izan daitezkeenez, ez dugu tratamendurik egin honentzat.

# KOMAK FALTAN: 7. LEGEA

# DESKR.: Lokailuak eta diskurtso-antolatzaileak koma artean idazten dira
# (esaldiaren hasieran edo bukaeran daudenean, koma bat eta beste puntuazio-ikur baten artean, noski).

# ADIB: Gu, hala ere, lasai geunden.
# Mehatxuka ari zitzaigun; hala ere, lasai geunden.
# Mehatxuka ari zitzaigun; lasai geunden, baina.
# (kasu honetan "baina"lokailua da, ez juntagailua)

# TRATAMENDUA: Lokailua bada, eta aurrekoa ez bada puntuazio-marka,
# koma jarri aurretik. Lokailua bada, eta ondorengoa ez bada
# puntuazio-marka, koma jarri ondoren.

# SALBUESPENA: "ere"lokailua. Kontuan hartuko dugu erregeletan.
# "ere"lokailua denean, bere aurretik aditza badator,
# perpausa kontsideratuko dugu, eta, beraz, koma jarriko dugu "ere"
# lokailuaren ondoren (aurretik inoiz ez).

# LOKAILUAREN AURREKO KOMA JARTZEKO:

MAP (&OKER_KOMA_FALTA_7_1) TARGET EDOZEIN_KAT
IF (NOT 0 PUNTUAZIO_MARKA)
(1 LOK)
(NOT 1 ERE) ;

# LOKAILUAREN ONDORENGO KOMA JARTZEKO:

```
MAP (&OKER_KOMA_FALTA_7_2) TARGET LOK
IF (0 LOK)
(NOT 0 ERE OR NAHIZETA)
(NOT 1 PUNTUAZIO_MARKA) ;

# ERE-ren KASU BEREZIA:

MAP (&OKER_KOMA_FALTA_7_3) TARGET ERE
IF (-1 ADITZA)
(NOT 1 PUNTUAZIO_MARKA) ;


# KOMAK FALTAN: 8. LEGEA

# DESKR.: Aditz nagusiaren aurreko mintzagaiaren ondoren koma jartzea
# gomendatzen da, mintzagaia subjektua ez denean betiere;
# behar-beharrezkoa da, ostera, anbiguotasuna baldin badago.

# ADIB: Azkenean, gaur iritsi dira.
# Etxean geratu arren, ez nuen batere deskantsatu.

# TRATAMENDUA: "arren"hitza topatzen badugu, eta loturazkoa
# bada eta aurretik aditza badu, koma jarriko zaio ondoren.

# SALBUESPENA:
# a) "arren"hitza "otoi"ren zentzuan erabilia izan bada,
# ez du komarik behar ondoren, aurretik baizik.
# b) "arren"hitza esaldiaren bukaeran doanean
# (beste puntuazio-ikur bat duelako).
# Adib.: Etor zaitez, arren.
# Ez nuen batere deskantsatu, etxean geratu arren.

# ARREN-en KASU BEREZIA:

MAP (&OKER_KOMA_FALTA_8_1) TARGET ARREN
IF (-1 ADITZA)
(0 LOT)
(NOT 1 PUNTUAZIO_MARKA OR BEREIZ);
```

# **KOMAK SOBRAN**

# KOMAK SOBRAN: 1. eta 3. LEGEAK

# DESKR.:
# Lehenengo legeak honela dio: Izen-sintagma eta
# aditz-kate barruan, ezin da koma bakarra jarri;
# tartean tartekiren bat edo lokailuren bat denean,
# koma parea jarri ahal izango da.
# Hirugarren legeak honela dio: # Galdegaia eta aditza ez dira komaz bereiziko, oro har.

# Arau honetan, beraz, bi legeok nolabait konbinatuz,
# edozein aditzen aurreko komak debekatuko ditugu.

# ADIB.: *Jon gurekin, etorri zen.
# *Jon gurekin etorri, zen.

# TRATAMENDUA: Aditzaren aurretik koma baldin badator, # okertzat emango dugu.
# Sintagmaren hasiera bada (eta ez bada hitz bakarrekoa), # eta ondoren koma badator, okertzat emango dugu.
# Oharra: "ize koma ize"egituretan zatiak-ek ize bakoitza # sint bat kontsideratuko du: zatitzaileari komaren erregelak kendu behar!!
#
# SALBUESPENAK: Tartekiak sartzea. Komen ordez, marra luzeak gomenda litezke soluzio gisa, horrelakoetan.
# Adib.: Eta hala, zentroa ireki berria dela,
ikusten ari gara nola kolapsatzen ari den.

MAP (&OKER_KOMA_SOBRA_8_1) TARGET KOMA (1 ADITZA) ;

MAP (&OKER_KOMA_SOBRA_8_2) TARGET SIH (1 KOMA) ;

END

# B. ERANSKINA

## Komen zuzentzailea lortzeko urratsak

### B.1 Komaren ikasketarako eman beharreko urratsak, corpus eta analizatzaile *komadun*ak erabiliz

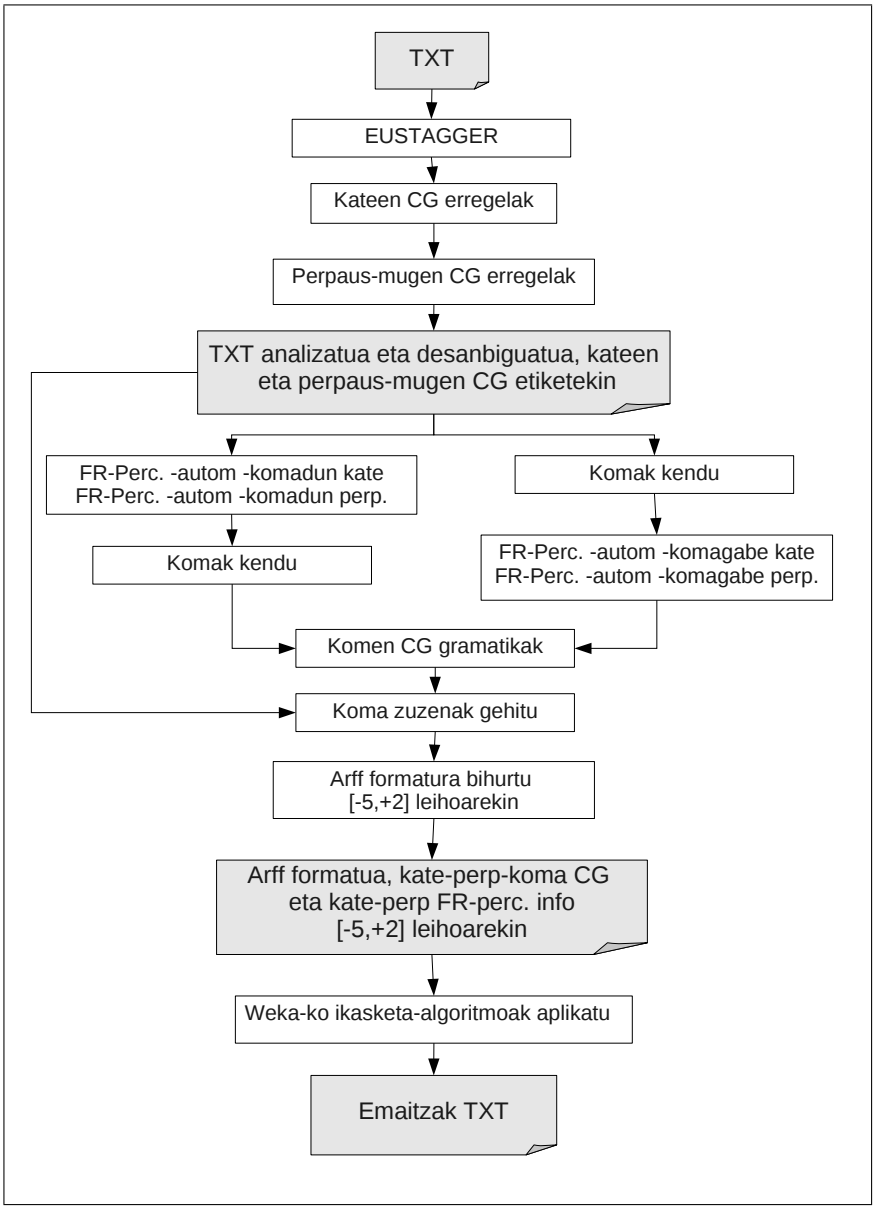
Atal honetan, komaren ikasketarako egin beharreko urratsak laburbilduko ditugu eskema batean (ikus B.1 irudia). Urrats hauek corpusean jatorrizko komak mantenduz eta *Eustagger komadun*a erabiliz egindakoak dira.

Irudian ikus daitekeen moduan, corpusa *Eustagger*-ekin analizatzea da lehendabiziko urratsa. Ondoren, kateen eta perpaus-mugen informazioa lortzeko CG gramatikak aplikatzen dira, hurrenez hurren.

Puntu honetan, bi bide desberdin probatu ditugu: batetik, corpusari komak kendu eta IV.6.2.9 atalean aipatutako kate- eta perpaus-identifikatzaile *komagabe*ak —FR-Perceptron algoritmoarekin lortutakoak— baliatzen dira, kateen eta perpausen informazio ahalik eta onena lortzeko. Bestetik (corpusari komak kendu gabe), kate- eta perpaus-identifikatzaile *komadun*ak baliatzen dira, kateen eta perpausen informazioa lortzeko. Bi bide hauek probatu ditugu, kate- eta perpaus-identifikatzaile *komadun*ak ala *komagabe*ak erabiltzearen arteko aldea aztertzeko.

Ondoren, komen CG gramatika aplikatzen da —aurretik corpusari komak kendu eta gero, dagoeneko kendu ez badira—. Hala, faltan dauden komak detektatzeko CG erregelen emaitza gehitzen dugu, beste ezaugarri baten moduan.

Azkenik, informazio hau guztia fitxategi bakar batean uztartzen da: *Eus-*




Irudia B.1: Komaren ikasketarako —*Eustagger komadun*a erabiliz— eman beharreko urratsen eskema-irudia.

*tagger*-ek eta kateen eta perpausen CG erregelek emandakoa, *FR-Perceptron* bidezko kate- eta perpaus-identifikatzaileek emandakoa eta komen CG erregelek lortutakoa, betiere jatorrizko komak dituen fitxategi batean. Izan ere, koma hauek erabiltzen dira —Weka softwareak ulertzen duen arff fitxategirako bihurketa egindakoan— komen ikasketa automatikoa egiteko.

## B.2 Komaren ikasketarako eman beharreko urratsak, corpus *komagabe*a eta analizatzaile *komadun*a erabiliz

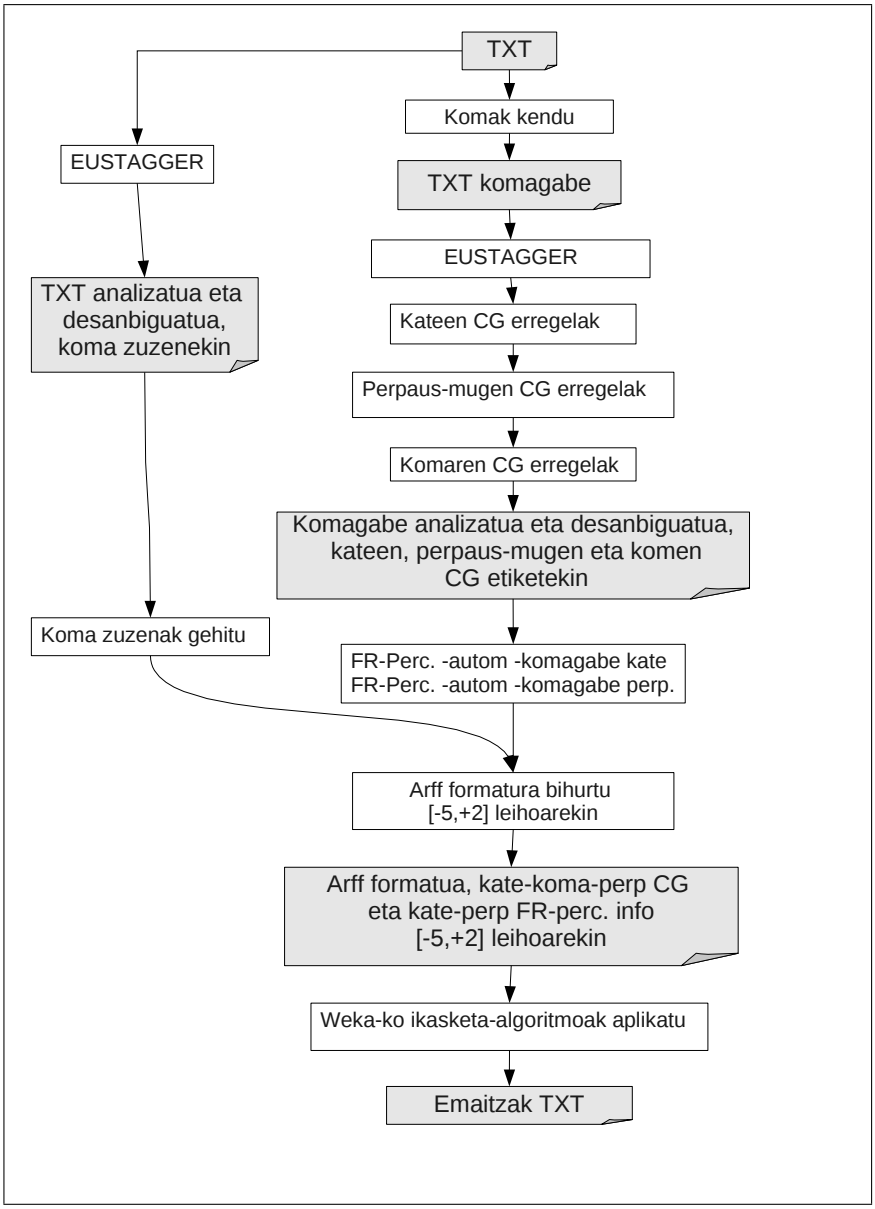
Kasu honetan, *Eustagger komadun*a erabili dugun arren, pasa diogun corpusa jatorrizkoari komak kendu ondoren lortutakoa izan da. Hala, komak ikasteko, B.2 irudiko urratsak jarraituz prestatu dugu corpusa.

Hasieran, bitan banatzen da prozesua: batetik, *Eustagger komadun*a aplikatzen zaio corpusari eta honen emaitza —koma egokiak dituen corpusaren analisi desanbiguatua— aurreragoko urratsetarako gordetzen da. Bestetik, ordea, komak kentzen zaizkio corpusari, baina *Eustagger komagabea*rekin analizatu beharrean (B.3 atalean ikusiko dugun moduan), *Eustagger komadun*arekin analizatzen da. Azken prozesu honetatik lortutako corpusari hiru CG gramatika aplikatzen zaizkio: kateena, perpaus-mugena eta komena. Hala, sarrerako corpusaren komarik gabeko bertsio analizatu eta desanbiguatua izango dugu, kateen eta perpaus-mugen informazio gehigarriarekin. Ondoren, komen CG gramatika aplikatzen da. Hala, faltan dauden komak detektatzeko CG erregelen emaitza gehitzen dugu, beste ezaugarri baten moduan.

Hurrengo urratsa, *FR-Perceptron* algoritmoaren bidez, kateen eta perpausen informazio osoagoa lortzea izango da. Honekin, eta koma egokiak berreskuratu ondoren, prest gaude komak zuzentzeko ikasketa automatikoa egiteko. Horretarako, Weka softwarea darabilgunez, *arff* formatura ekarri beharko dugu gure corpusa, erabaki dugun leihoaren (-5,+2) araberako formatuari eutsiz, gainera.

## B.3 Komaren ikasketarako eman beharreko urratsak, corpus eta analizatzaile *komagabe*ak erabiliz

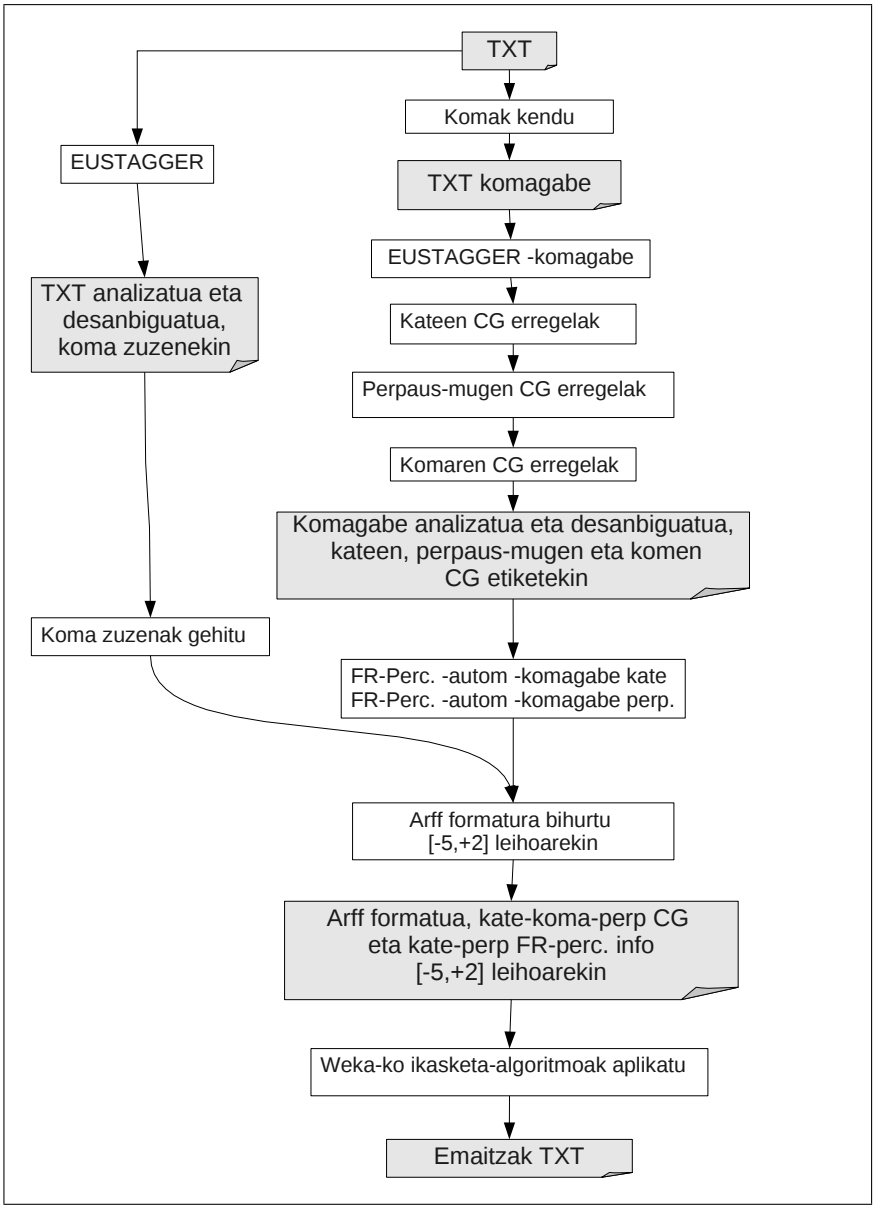
B.3 irudian laburbildu ditugu komaren ikasketarako erabili dugun corpusa prestatzeko egindako urratsak, *Eustagger komagabea* erabiliz.




Irudia B.2: Komaren ikasketarako —corpus *komagabe*a eta *Eustagger koma-dun*a erabiliz— eman beharreko urratsen eskema-irudia.

Hasieratik, bitan banatzen da prozesua: batetik, *Eustagger komadun*a erabiltzen da corpusa analizatzeko eta honen emaitza —koma egokiak dituen corpusaren analisi desanbiguatua— aurreragoko urratsetarako gordetzen da. Bestetik, ordea, komak kentzen zaizkio corpusari eta *Eustagger komagabea*rekin analizatzen da segidan. Azken prozesu honetatik lortutako corpusari hiru CG gramatika aplikatzen zaizkio: kateena, perpaus-mugena eta komena. Hala, sarrerako corpusaren bertsio analizatu eta desanbiguatua izango dugu, kateen eta perpaus-mugen informazio gehigarriarekin. Ondoren, komen CG gramatika aplikatzen da. Hala, faltan dauden komak detektatzeko CG erregelen emaitza gehitzen dugu, beste ezaugarri baten moduan.

Hurrengo urratsa, *FR-Perceptron* algoritmoaren bidez, kateen eta perpausen informazio osoagoa lortzea izango da. Honekin, eta koma egokiak berreskuratu ondoren, prest gaude komak zuzentzeko ikasketa automatikoa egiteko. Horretarako, Weka softwarea darabilgunez, *arff* formatura ekarri beharko dugu gure corpusa, erabaki dugun leihoaren (-5,+2) araberako formatuari eutsiz, gainera.




Irudia B.3: Komaren ikasketarako —*Eustagger komagabea* erabiliz— eman beharreko urratsen eskema-irudia.

*Antzarak zeruan neguan bezala,*
*dana pasaten da.*
*Denbora bezala,*
*mina bezala,*
*dana pasaten da.*
*Bafor bat hodeizintan bezala.*

Josu Aranbarri (Piztiak)

Tesi honen idazketa
2010eko apirilaren 26an
bukatu zen.