Anafora leiala eta pronominala etiketatzen laguntzeko sistema

2011. Iraila

Egilea: Iakes Goenaga Azkarate

Proiektuaren zuzendaria: Olatz Arregi

Hizkuntzaren Azterketa eta Prozesamendua Masterreko titulua lortzeko bukaerako proiektua

Laburpena

LNP alorreko lanaren zati bat testuak etiketatzea da, ondoren lortutako corpusarekin bestelako teknika batzuk erabiliz ataza ezberdinak burutzeko.

IXA taldean etiketatzeko erabiltzen den tresna MMAX2 izeneko kode irekiko aplikazioa da. Honekin hizkuntz fenomenoak markatzeko lana grafikoki egin daiteke modu azkarrago eta errazago batean. Etiketatze lan horretan arazo handiak ematen dituzte anaforek, zenbait kasutan hauek identifikatzea oso zaila delako eta behin identifikatuta, hauen aurrekariak zeintzuk diren zehaztea ez delako erraza.

Proiektu honekin anafora fidel eta pronominal gehienak automatikoki markatzea lortu da hizkuntzalari batek eskuz MMAX2 erabilita markatu izan balitu bezala. Diseinatu dugun sistemak anafora fidelak eta pronominalak markatu dituenean, hizkuntzalariak egin beharko duen gauza bakarra proposamenak onartzea ala zuzentzea da.

Gaien Aurkibidea

Irudien Zerrenda

Kapitulua 1

Sarrera

IXA taldeko hizkuntzalariek izugarrizko lana egiten dute beraien etiketatzeko tresnarekin testu gordinetik abiatuta bertan aurkitzen dituzten hinkuntz fenomenoak banan banan markatzen. Etiketatzen dituzten elementuak mota askotakoak diren arren, anaforak markatzea lan astunenetarikoen artean dago. Anaforak testuan zehar aurkitzea ez da lan xamurra eta hauei beraien aurrekaria esleitzeak ere sekulako lana du.

1.1 Anafora eta korreferentzia

Anafora eta korreferentzia askotan lotuta daude eta elkarrekin tratatzen dira beraien artean antzekotasun eta zerikusi handia dutelako. Hala ere, bi termino horiek gauza bera izango balira bezala hartzea ez da zuzena, ez baitira gauza bera inolaz ere.

Bi horien arteko desberdintasunak zehazteko autore batek (Hirst, 1981) Lengoaia Naturalen Prozesamendua alorrean eman zuen anaforaren definizioarekin hasiko gara:

> *Anafora diskurtsoan entitate bati edo gehiagori erreferentzia laburra egiten dion elementua da, ondoren jasotzaileak erreferentzia horretatik abiatuz entitatearen identitatea asmatuko duen intentzioarekin. Erreferentzia ANAFORA deitzen da eta honek erreferentziatzen duen entitatea ERREFERENTEA edo AURREKARIA. Erreferentzia edo anafora eta honen erreferentea korreferenteak direla esaten da. Anafora baten erreferentea zehazteko prozesuari EBAZPENA deritzo.*

Irakurleak anaforaren kontzeptua hobeto ulertuko duelakoan anafora bat eta bere aurrekaria ageri diren ondorengo esaldiari gainbegiratu bat botatzea proposatzen dugu:

> ***Esperanza Agirre eta Mariano Rajoy*** *helikopteroan erori eta bizirik atera ziren.* ***Hauek*** *bai dutela sortea!!!*

Aurreko esaldian anafora pronominal bat ageri da, *Hauek* izenordaina anafora da eta *Esperanza Agirre eta Mariano Rajo*y bere aurrekaria.

Anafora bat bere aurrekariaren menpe dago beti, semantikoki hutsa delako eta aurrekariaren beharra duelako hutsune hori betetzeko. Korreferentzia, berriz, erlazio pragmatikoagoa da eta testuinguruaren araberakoa. "Erreferentziaren identitatea" erlazioa duten bi hizkuntz elementuren artean ematen da. Bi entitateren artean korreferentzia erlazioa dagoela edo korreferenteak direla esaten da bi entitate horiek kontzeptu bera erreferentziatzen badute (Recasens, 2008). Har dezagun adibide bezala hurrengo esaldia:

> *Batzuek diote* ***Elvis*** *handia bizirik dagoela eta Las Vegas-en bizi dela. Beste batzuek, ostera,* ***abeslaria*** *estralurtar bat zela eta bere planetara itzuli dela diote.*

Esaldi horretan argi ikusten da *Elvis* eta *abeslaria* hitzek *Elvis Presley* abeslaria erreferentziatzen dutela, ondorioz korreferentzia erlazio bat dutela esan dezakegu, *abeslaria* hitzak *Elvis*-en beharra izanik horretarako.

Nahiz eta batzuek testuaren kohesioan fenomeno honek duen garrantzia goraipatzen duten, bera bakarrik ez da gai kohesio hori lortzeko. Errepikapenak ekidinez testua estilistikoki aberasten duen diskurtso mekanismo bat bakarrik da.

Anafora eta korreferentziaren arteko antzekotasunak ugariak dira, anaforak eta bere aurrekariak gehienetan korreferentzia erlazio bat dute, baina beti ez da gertatzen azken hau. Izan ere, posible baita erlazio anaforiko bat korreferentzia elazioa ez izatea. Berdina gertatzen da alderantziz ere, korreferentzia erlazio guztiak ez dira erlazio anaforikoak.

Bukatzeko, ohitura eta erraztasunagaitik anafora fidelei anafora deituko diegu nahiz eta benetan anafora ez diren kontsideratzen (Recasens, 2008).

1.2 Motibazioa eta helburuak

Hizkuntzalariei hizkuntza fenomenoak markatzen dituztenean lanaren zati bat erraztuko dien sistema baten beharrak bultzatuta eraiki dugu esku artean dugun proiektua. Euskararako anaforak automatikoki markatzen dituen sistema bat eraikiz gero, hizkuntzalariek anaforak markatzeko erabiltzen duten denbora asko murriztuko litzateke. Izan ere, anafora gehienak markatuta egongo

bailirateke eta hizkuntzalariek egin beharko luketeen gauza bakarra zuzenketak egitea da, hots, sistemak markatu ez dituenak markatu eta gaizki markatu dituenak zuzendu.

Motibazioaren zati handi bat hizkuntzalariei lana erraztuko dien sistema bat egitea bada ere, baditugu arrazoi gehiago proiektu hau aurrera eramateko. Arrazoi horietan garrantzitsuena LNP osatzen duten hainbat alorretan aurrera pausoak emateko grina da. Anaforen ebazpena oso lotuta dago LNPko beste hainbat alorrekin (*Question answering*, *Machine translation*...) eta ondorioz, anaforak ebazteko sistema bat eraikiko bagenu, beste alor horietan aurrerakuntzak egiteko aukera izango genuke.

Aurreko paragrafoetan gure sistemak anaforak ebazten dituela esan dugun arren, ez ditu anafora mota guztiak ebazten. Izan ere, oso zaila baita anafora mota guztiak ebazten dituen sistema bat eraikitzea. Nahiz eta ezberdin tratatzen diren, gure proiektua anafora fideletan eta pronominaletan zentratzen da, anafora fidelak anaforen artean errazenetarikoak direlako eta anafora pronominalak IXA taldean aurretik zertxobait landuta daudelako.

Ondorioz, gure proiektuaren helburu nagusia euskararako anafora pronominalak eta fidelak automatikoki markatzen dituen sistema bat inplementatzea da. Sistema honekin hizkuntzalariei lana zertxobait erraztuko zaie eta LNPan aurrera pausoak eman ahal izango ditugu. Helburu nagusia orain aipatu berri dugun hori den arren, etorkizunari begira badugu beste helburu bat: euskarazko corpus ezberdinetan anafora pronominalak eta fidelak automatikoki markatzea.

1.3 Egitura

Gure lanaren memoria ondorengo ataletan banatu dugu: Hasteko, gaian sartzen joateko sarrera bat idatzi dugu. Hurrengo kapituluan artearen egoera eta aurrekariak aipatuko ditugu. 3. kapituluan proiektua burutzeko erabili diren baliabideak azalduko ditugu. Horren atzetik MMAX2 aplikazioaren nondik norakoei buruz mintzatuko gara. 5. kapituluan proiektuan barrena murgilduko gara anafora pronominalak eta fidelak ebazteko sistemen diseinua eta inplementazioa azalduz. Proiektuaren ebaluazioari buruzko zehaztapenak 6. kapituluan emango ditugu. Horren atzetik bidean aurkitu ditugun arazoak azalduko ditugu. Eta bukatzeko proiektuari buruz atera ditugun ondorioak eta etorkizunerako lanak kontatuko ditugu.

Kapitulua 2

Artearen egoera eta aurrekariak

Esku artean dugun proiektua aurrera eramateko hainbat lan hartu ditugu irizpide bezala. Horietatik gurearekin zerikusi handia duen lan bat (Versley et al., 2008) da. *BART* izeneko proiektu horretan, ingeleserako hainbat anafora mota ezberdin ebazteko tresna bat garatu dute. Tresna horrek, testu gordinetik abiatuta, bertan aurkitzen dituen anafora mota guztiak grafikoki eta modu erakusgarri batean aurkezten ditu. Emaitzak MMAX2 formatura pasatzeko aukera ere ematen du, MMAX2 (Muller and Strobe, 2006) anotazio tresnarekin emaitzak ikusi eta aldatu ahal izateko. Gure asmoa horrelako tresna bat egitea izan da euskarazko anafora mota guztiak ebazteko, baina lan horrek inplikatzen duen zailtasunaren aurrean proiektu honetarako asmoak mugatu behar izan ditugu. Ingeleserako egin diren lan gehiago azalduz, oso interesgarria da (Soon et al., 2001) artikulua. Bertan ikasketa automatikoaren bidez etiketatutako corpus batetatik ikasten dute eta edozein izen-sintagmen arteko korreferentziak ebazten dituzte. Ingeleserako lanekin bukatzeko, (Ng and Cardie, 2002) lanean (Soon et al., 2001) artikuluan lortutako emaitzak hobetzen dituzte azken hauen lanean zenbait aldaketa eginez. Adibidez, ikasketa automatikoan erabiltzeko ezaugarri ezberdinen kopurua gehituta (12tik 53ra).

Artearen egoerarekin jarraituz, beste hizkuntza batzuetarako gure gaiarekin zerikusia duten hainbat lan egin dira. (Nguy and Zabokrtský, 2007) lanean erregeletan oinarritutako sistema bat proposatzen dute Txekierarako anaforak ebazteko. Lan horretan, eskuz etiketatuta dauden ia 50.000 esalditan banatutako 45.000 korreferentzia lotura inguru dituen Treebank bat erabiltzen dute. (Versley, 2006) lanean, berriz, egileak Alemanieraz idatzitako testuetan eredu estatistikoa erabilita ebazten ditu izen-sintagmen arteko korreferentziak. Bestalde, (Moosavi and Ghassem-Sani, 2008) eta (Moosavi and Ghassem-Sani, 2009) proiektuetan ikasketa automatikoko teknikak erabiltzen dituzte Persierarako izenordainak ebazteko.

Gure proiektuarekin lotura duten beste bi lan interesgarri azaltzen dira (Zelaia et al., 2010) eta (Arregi et al., 2010) artikuluetan. Lan horietan, euskarazko anafora pronominalak ikaske-

ta automatikoa erabilita tratatu dituzte lehenengo aldiz. Sekulako lana egin dute sailkatzaile ezberdinekin eta ezaugarri ezberdinekin emaitzarik hoberenak zein konbinaziok ematen dituen zehazteko. Lan horietatik, erabili duten ikasketarako corpusa aprobetxatu dugu eta bertan jasotako emaitzak eta erabilitako sailkatzaile eta ezaugarriak oso kontutan izan ditugu.

Beste zenbait lan ere erabili ditugu erreferentzi bezala. (Jimeno, 2010) karrera bukaerako proiektuan euskararako anafora pronominalak ebazten dira testu gordinetik abiatuta ikasketa automatikoa erabiliz. Proiektu honetatik oso baliagarria izan zaigu ikasketa automatikoa erabiltzen duen modulua nola inplementatuta dagoen ikustea.

Anafora fidelak tratatzeko oso lagungarria izan zaigu (Aduriz et al., 2005) artikulua, bertan anafora mota hau identifikatzen laguntzeko izen-sintagmek izan behar dituzten ezaugarriak zehazten baitira. Bukatzeko, korreferentziaren inguruan katalanerako eta gaztelerarako egindako tesi bat (Recasens, 2008) ere oso lagungarria izan zaigu. Bertan ikasketa automatikoa aplikatzen da bi hizkuntza horietako korreferentziak ebazteko.

Kapitulua 3

Erabilitako baliabideak

Gure sistema eraikitzeko erabili ditugun baliabideen artean aipagarrienak Weka ikasketa automatikorako tresna eta Perl eta Java programazio lengoaiak izan dira.

Ikasketa automatikorako tresna zuzenean bere interfazearekin erabili ez dugun arren, honek eskaintzen dituen liburutegiak erabili dira anafora pronominalak automatikoki ebazteko. Ondorengo puntuetan azalduko dugun bezala, ikasketa automatikoa aplikatu ahal izateko Wekako liburutegiak erabilita, corpus bat beharrezkoa da programak bertatik ikasteko eta corpus hori formatu berezi batean egon behar da, arff formatuan alegia.

Java eta Perl programazio lengoaiei aipamen berezi bat egitea gustatuko litzaiguke, hauek erabili gabe proiektu hau aurrera eramatea oso zaila izango litzatekelako. Gure proiektuan Java ikasketa automatikoarekin lotuta dauden eragiketa guztiak egiteko erabili da eta beste edozertarako (formatu aldaketak, irakurketak, informazioren eskurapena, idazketak...) Perl erabili da.

3.1 Java programazio lengoaia

Nahiz eta egun *Java* programazio lengoaia *Oracle*-rena izan, objektuei zuzendutako lengoaia ahaltsu hau *Sun Microsystems*-ek sortu zuen 90. hamarkadaren hasieran. *Java*-k C eta *C++* lengoaien sintaxiaren zati handi bat dauka barneratuta baina hainbeste ezaugarritan oso desberdinak dira lengoaia hauek, adibidez errore asko ematen dituen memoriaren manipulazio zuzenik ez du erabiltzen *Java*-k eta horregatik errazagoa da lengoaia hau menperatzea.

Gure proiektuan anafora pronominalen aurrekari probableenak aurkitzeko ikasketa automatikoa erabili dugu eta lan hori aurrera eramateko Java erabiltzea pentsatu da ondorengo lau arrazoi hauengatik:

1. Gure programan ikasketa automatikoa erabili ahal izateko liburutegiak *Java*-z idatzita daude.
2. *Java*-z egindako aplikazio bat *Windows*-en, *Linux*-en nahiz *Mac*-en exekutatu daitekeelako.
3. Lengoaia honi buruzko informazio ugari aurkitu daitekeelako interneten.
4. *Java*-rekin dugun esperientziagaitik.

Behin erabaki dugunean zein lengoaiatan idatzi gure aplikazioa, komenigarria izaten da programazio ingurune batean programatzea honek ematen dituen erraztasunak direla eta. Guk Java-rako daudenen artean NetBeans erabiltzea erabaki dugu karrera bukaerako proiektuan erabili genuelako eta ongi ezagutzen dugulako honek eskaintzen duen guztia.

3.2 Perl programazio lengoaia

Perl (Practical Extraction and Report Language) programazio-lengoaia *Larry Wall*-ek *UNIX* sistemetako administrazioko lanak sinplifikatzeko helburuarekin sortu zuen arren, gaur egun helburu orokorreko lengoaia bihurtu da eta hainbat lanetarako ere oinarrizko tresna. Gainera, denboran zehar jasan duen eboluzioari esker, gaur egun *UNIX* ez diren beste hainbat ingurunetan ere erabil daiteke: *Windows*, *Amiga*, *MacOS*...

Perl lengoaia interpretatua dela esaten da bere interpretatzaileak gaur eguneko interpretatzaile moderno gehienak bezala, programak egikaritu aurretik konpilatzen dituelako. Horregatik Perl ingurunean script terminoa erabiltzen da eta ez programa. Testuak prozesatzeko erosotasuna da gure proiekturako programazio-lengoaia honen ezaugarri interesgarriena. Besteak beste, fitxategiekin lan egiteko erraztasuna eskaintzen du eta espresio erregularrekin lan egiteko oso kudeaketa ona.

Hainbat ataza burutzeko aurredefinitutako funtzio multzo bat eskaintzen duela da beste ezaugarri aipagarri bat, programatzaileari lana erraztuko diona. Gainera aldagaiak ez dira definitu behar, balio lehenetsia dute, eta array-en luzera ez da finkatu behar, dinamikoki handitu eta txikitu daitekeelarik momentuko beharren arabera.

Guk erabili ditugun funtzio aipagarrienak bektoreak atzitu eta maneiatzekoak, fitxategiak irakurri eta bektore batean modu azkar eta sinplean gordetzekoak eta adierazpen erregularrek erabiltzen dituzten funtzioak izan dira batik bat. *Perl*-etik komando interpretatzailetik exekutatuko genituzkeen bezala aginduak exekutatzeko funtzioek ere sekulako erraztasunak eman dizkigute gure lana burutzeko.

Orain arte *Perl*-ek dituen abantailak besterik ez dira aipatu baina esate baterako, programazioan metodologiari arreta handia jartzen ez bazaio script-ak ez dira oso dotoreak eta irakurgarriak izango, *Perl*-en helburu nagusia atazak modu azkarrean egikaritzea baita, itxuran erreparatu gabe.

Lengoaia hau aurretik inoiz ez dugu erabili, baina, gure kasuan behintzat, oso erraza izan da ikasketa prozesua eta uste dugu oso baliagarria izango zaigula ikasitakoa etorkizunean, batez ere hizkuntzen tratamendurako erraztasun ugari ematen dituelako.

Bukatzeko, *Perl*-erako dauden programazio inguruneak asko ez diren arren, badaude batzuk nahiko interesgarriak direnak. Guk horien artean *Padre* programazio ingurunea erabiltzea pentsatu dugu funtzio asko dituelako eta doakoa delako.

3.3 Weka ikasketa automatikorako tresna

Zelanda Berriko Waikato unibertsitatean garatutako ikasketa automatikorako tresnen bilduma da Weka. Edozein plataformatan (Windows, Linux, Mac Os...) erabil daiteke eta software librea da. Tresna honek hainbat aukera ematen ditu ikasketa automatikoa ataza ezberdinetan aplikatzeko: sailkatzaile ugari, ezaugarrien trataera, interfaze grafiko egoki bat, etab.

Weka-k lana aurrera eramateko beharrezkoa du corpus bat. Corpusa honako baldintzak betetzen dituen testu-bilduma bat da: hizkuntza bateko erabilpen errealen adibide multzoa da, irizpide batzuen arabera bildua dagoena, formatu elektronikoan biltegiratua eta informazio linguistikoz osatua dagoena. Weka-k arff formatuan dauden fitxategiak bakarrik onartzen dituenez, corpus hori formatu horretan eduki behar dugu.

3.3.1 Sailkatzaileak

Wekak erabiltzen dituen sailkatzaile desberdinei buruz hitz egingo dugu puntu honetan. Sailkatzaile horiek 7 multzo desberdinetan banatuta daude:

- **Bayes:** Bayes-en funtzioaren aldaerak erabiltzen ditu sailkapenerako.
- **Functions:** Funtzio desberdin ugari daude aukeran sailkapenak egiteko.
- **Lazy:** Multzo honetan, instantzietan oinarritutako zenbait sailkatzaile daude, adibidez n auzokide hurbilenak (KNN) algoritmoaren inplementazioa.
- **Meta:** Sailkatzaile desberdinak konbinatzen dira hemen.
- **Trees:** Zuhaitzetan oinarritzen diren algoritmoak daude atal honetan.
- **Misc:** Bertan Hiperpipes eta VFI bezalako sailkatzaileak aurkitu ditzakegu.

- **Rules:** Erregeletan oinarritutako algoritmoak daude atal honetan.

Goazen orain anafora pronominalak ebazteko probetan erabili ditugun sailkatzaile desberdinen artean garrantzitsuenak azaltzera:

- **NaiveBayes**: Bayes funtzioaren bertsioetako bat da.
- **SMO:** Sailkatzaile hau oinarri bektoreetan (SVM) oinarritzen da bere sailkapena burutzeko.
- **IB1:** Wekak kasu bakoitzean ikasketako instantzien artekoetatik gehien hurbiltzen denaren balioa esleitzen dio. Knn algoritmoan oinarritzen da K 1 izanik.
- **J48:** C4.5 zuhaitzaren inplementazioa da, zenbait arlotan oso emaitza onak lortzen direlarik berarekin.
- **RandomForest:** Atributuen ausazko azpimultzoak dituzten zuhaitzak sortuta egiten da sailkapena.
- **Logistic:** Erregresio logistiko multinomialaren eredua erabiltzen duen sailkatzailea.
- **VFI:** *Misc* taldean kokatzen den algoritmoa da hau, *Voting Feature Intervals*. Ezaugarri tarteak sortzen ditu. Tarte batek ezaugarri bati dagokion balio multzo bat adierazten du, non azpimultzo berdineko klase balioak aztertzen diren.

3.3.2 Arff formatua

Ondorengo kapituluetan formatu hori hainbat alditan aipatuko dugunez, ezinbestekoa iruditzen zaigu horren nondik norakoak azaltzea puntu honetan. Proiektu honetan anafora pronominalen aurrekariak identifikatzeko ikasketa automatikoa erabili dugu. Ikasketa automatikoa erabiltzeko garatu dugun aplikazioak arff formatua duen fitxategia behar du ikasteko, baita ebaluaketa egiteko ere.

Formatu berezi hau duten fitxategiak 3 zatitan daude banatuta:

- **Burukoa:** Hemen fitxategiaren izenburua definitzen da.

  Bere formatua: @RELATION <erlazio izena>

- **Atributuen deklarazioa:** Hemen gure fitxategiak dituen atributuen deklarazioa egiten da eta hauen mota adierazten da. Azken atributua klasea edo kategoria izaten da, hau da, iragarri nahi den balioa.

  Bere formatua: @ATTRIBUTE <atributu izena> <**mota**>

Atributuaren **mota** honako bost aukeretako bat izan daiteke:

1. Zenbaki errealak.
2. Zenbaki osoak.
3. Datak.
4. Enumeratuak.
5. String-ak.

- **Datuak:** Atal honetan aurrekoan definitutako atributu edo ezaugarri zehatzak dituzten instantziak edo adibideak izango ditugu. Instantzia bakoitza lerro batean doa eta lerro bukaeran klasea. Hau izango da ikasteko erabiliko den informazioa, gure kasuan anafora pronominalen trataerarako. Ezaugarri baten balioa ezezaguna bada, bere ordez "?" ikurra jarriko da.

Bere formatua:

@DATA

<instantziaren ezaugarrien balioak><klasea>

.

.

.

<instantziaren ezaugarrien balioak><klasea>

Orain badakigu arff formatua duen fitxategi batek 3 atal nagusi dituela eta atal bakoitzean ze informazio ematen den, baina irudi batek 1000 hitzek baino gehiego balio duenez 3.1 irudiarekin argi geldituko da nolakoa den arff formatuko fitxategi bat.

```
@RELATION Anaforen_ikasketa

% izen sintagmaren ezaugarriak
@ATTRIBUTE azpikategoria {ADK, ADOARR, ADOGAL, ADP, ALGGAL, ALGARR, ARR, BAN, DZG, DZH, EGI, ELK, ERKARR}
@ATTRIBUTE deklinabide_kasua_is {ABL, ABS, ABU, ALA, ABZ, DAT, DES, ERG, GEL, GEN, INE, INS, MOT, PAR, PRO, SOZ}
@ATTRIBUTE numeroa {MG, NUMP, NUMS}
@ATTRIBUTE sintagmak_is {SIB, SINT}
@ATTRIBUTE funtzio_sintaktikoa_is {+JADLAG_IZLG>, +JADNAG_MP, ADLG, ATRIB, IZLG>, LOK, MP, OBJ, PJ, ZOBJ, +JADLAG_MP}

%loturazko ezaugarriak
@ATTRIBUTE distantzia {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
@ATTRIBUTE esaldi_bera {0,1}
@ATTRIBUTE Numero_Bera {0,1,2,3}
@ATTRIBUTE entitatea {ENTI_ORG, ENTI_LOC, ENTI_PER, ENTI_HAS_PER, ENTI_BUK_PER, ENTI_HAS_LOC, ENTI_BUK_LOC,
 ENTI_HAS_ORG, ENTI_BUK_ORG}

% anaforaren ezaugarriak
@ATTRIBUTE deklinabide_kasua_ana {ABL, ABS, ABU, ALA, ABZ, DAT, DES, ERG, GEL, GEN, INE, INS, MOT, PAR, PRO, SOZ}
@ATTRIBUTE funtzio_sintaktikoa_ana {+JADLAG_IZLG>, +JADNAG_MP, ADLG, ATRIB, IZLG>, LOK, MP, OBJ, PJ, ZOBJ, +JADLAG_MP, SUBJ}
@ATTRIBUTE sintagmak_ana {SIB, SINT, SIH}
@ATTRIBUTE numeroa_ana {MG, NUMP, NUMS}

% kategoria
@ATTRIBUTE erreferentea {0,1}

@DATA
SIN ABS MG SINT OBJ 1 0 0 ? ABS SUBJ SINT NUMS 0
ARR ABS NUMS SIB SUBJ 2 0 3 ? ABS SUBJ SINT NUMS 0
ARR ABS NUMS SIB PRED 3 0 3 ? ABS SUBJ SINT NUMS 1
IZO ABS MG SIB PRED 1 1 0 ? ABS OBJ ? NUMS 0
ARR ? NUMS ? KM> 2 1 3 ? ABS OBJ ? NUMS 1
DZH ABS MG SIB SUBJ 1 1 0 ? ERG SUBJ ? NUMS 0
ARR ? NUMS ? KM> 2 1 3 ? ERG SUBJ ? NUMS 1
```

Irudia 3.1: Arff formatua

Aurreko irudian aipatutako 3 atalak garbi ikusten dira. Hasteko burukoa dugu (@RELATION Anaforen_ikasketa). Ondoren erabiliko diren atributu guztiak daude definituta (@ATTRIBUTE <atributuaren izena>{atributuaren balio posibleak}), atal honen amaieran klasea dugularik. Bukatzeko instantzia ezberdinez osatutako datuen atala dugu (@DATA <instantziak>) instantzia bakoitzaren amaieran klasearen balioa dugularik sistemak ikasi dezan. Gure kasuan, klaseak har ditzakeen balioak 0 eta 1 dira eta erabili diren ezaugarri kopurua 14 (13 + klasea). Ezaugarri horiei buruzko zehaztapenak hurrengo kapituluetan azalduko dira.

Kapitulua 4

MMAX2 tresna

Tresna horri buruz hainbat alditan hitz egin dugun arren ez dugu berari buruzko informazio askorik eman. Goazen ba horretara.

4.1 Sarrera

Tresna hau software libre kategoriaren barruan sartzen da eta Javaz inplementatuta dagoenez edozein plataformatan (Windows, Linux, Mac...) erabil daiteke. Testuaren gainean hainbat oharpen egiten uzten du, korreferentzi mailatik hasita esaldi mailakoetara arte. IXA taldeko hizkuntzalariek korreferentzi mailako anotazioak egiten dituztenez horietan zentratuko gara gu. Taldean markatzen diren korreferentzi mailakoen artean kontzeptuzko anaforak, anafora fidelak, pronominalak, lekuzko adberbioak, izen bereziak eta bestelakoak deritzenak daude.

Tresna horrekin anotazioak egiteari buruz mintzatzen ari gara, baina testu bat markatu ahal izateko, testu horrek formatu berezi bat izan behar du. Demagun *fitx1.txt* izeneko fitxategian dugun testuan zenbait anafora pronominal nahi ditugula markatu MMAX2-arekin. Gure helburua lortzeko ezin dugu anotazio tresnarekin fitxategia ireki eta bertan anafora pronominalak markatu. Lehenengo formatu aldaketa bat egin beharko litzateke MMAX2-ak behar dituen formatua (.mmax) eta direktorio sistema lortzeko.

4.2 MMAX2 formatua

Gorago esan dugun modura, MMAX2-ak fitxategi baten gainean anotazioak egin ahal izateko, fitxategi horrek formatu berezi eta direktorio sistema jakin bat behar ditu izan.

4.2.1 Direktorio sistema

Direktorio sistema berezi hori hainbat karpetek eta fitxategik (ikus A eranskina) osatzen duten arren, guretzat garrantzitsuenak markables karpeta eta hemen kokatzen diren fitxategiak dira. Izan ere, MMAX2 tresnarekin hizkuntza fenomenoak grafikoki ikusi ahal izateko fitxategi horietan egin behar baitira idazketak. MMAX2 tresna erabiltzen duten hizkuntzalarientzat gardena da prozesu hori tresna bera arduratzen delako fitxategi horietan idazketak egiteaz, baina gure kasuan guk sortutako aplikazioak arduratu dira horretaz. Ondorengo lerroetan aipatutako direktorio eta fitxategiak azalduko ditugu:

- **Markables karpeta:** karpeta honen barnean korreferentzi mailako nahiz esaldi mailako loturak egiteko fitxategiak daude.
  - **Fitxategiaren izena_coref_level.xml fitxategia:** guretzat oso garrantzitsua da fitxategi hau, bertan egin behar izan ditugulako anafora fidel nahiz pronominalen arteko loturak. Hemen zehazten da zein hitz edo izen-sintagma dagoen zeinekin lotuta 4.1 irudian ikusten den bezala (aurreko kasuan *fitx1.txt_coref_level.xml* izena izango luke fitxategiak).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE markables SYSTEM "markables.dtd">
<markables xmlns="www.eml.org/NameSpaces/coref">
<markable id="markable_62" span="word_22..word_22" type="anaphoric" coref_class="set_120" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_59" span="word_11..word_14" type="none" coref_class="set_120" mmax_level="coref" number="S" />
<markable id="markable_3" span="word_17..word_17" type="none" coref_class="set_120" mmax_level="coref" />
<markable id="markable_60" span="word_17..word_18" type="none" coref_class="set_120" mmax_level="coref" number="S" />
<markable id="markable_69" span="word_38..word_38" type="anaphoric" coref_class="set_121" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_4" span="word_29..word_29" type="none" coref_class="set_121" mmax_level="coref" />
<markable id="markable_5" span="word_33..word_33" type="none" coref_class="set_121" mmax_level="coref" />
<markable id="markable_67" span="word_33..word_34" type="none" coref_class="set_121" mmax_level="coref" number="S" />
<markable id="markable_82" span="word_83..word_83" type="anaphoric" coref_class="set_122" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_78" span="word_73..word_74" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
<markable id="markable_78" span="word_73..word_74" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
<markable id="markable_80" span="word_77..word_77" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
</markables>
```

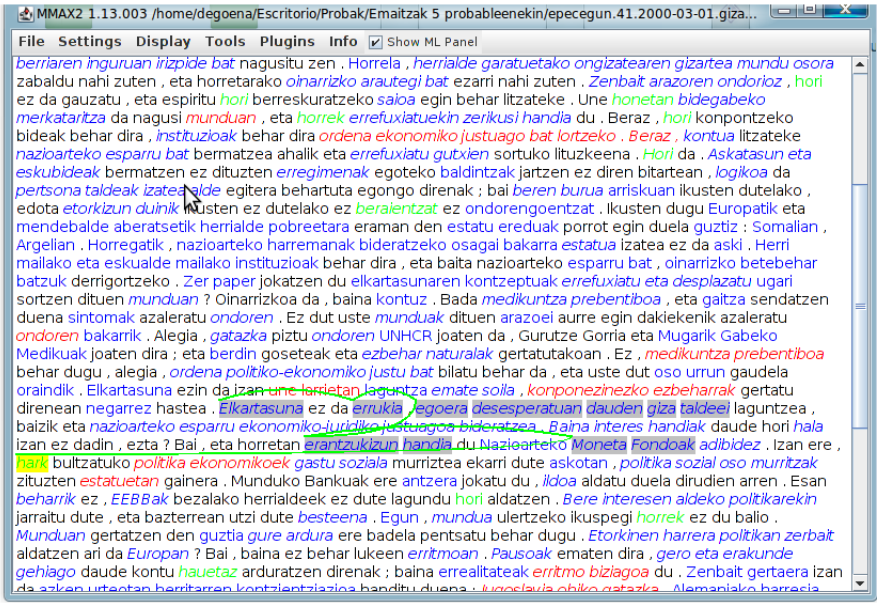
Irudia 4.1: Korreferentzien fitxategia

  - **Fitxategiaren izena_sentence_level.xml fitxategia:** esaldi mailako loturak zehazteko erabiltzen da fitxategi hau (aurreko kasuan *fitx.txt_sentence_level.xml* izena izango luke fitxategiak).
  - **markables.dtd fitxategia:** fitxategi honek aurreko biek izen behar duten formatua adierazten du.

Behin direktorio eta fitxategi garrantzitsuenak azaldu ditugula, MMAX2 aplikazioren zenbait irudi erakustea komenigarria dela uste dugu. 5.15 irudian anafora fidelen arteko loturak ageri dira lerro berde batez koloreztatuta. Lotuta ageri diren izen-sintagmek *Manuel Montero* pertsona erreferentziatzen dute, denen artean anafora fidel bat osatzen dutelarik. Lotura hori ikusi ahal izateko nahikoa da anafora fidel hori osatzen duten ataletariko baten gainean klik egitea. Hala ere, anafora fidel bat markatuta dagoela adierazteko, hori osatzen duten izen-sintagmetariko bat kolore gorriz dago markatuta. 5.24 irudian, berriz, anafora pronominalen arteko loturak ageri dira. Kasu honetan,

anafora pronominalak berdez daude markatuta eta hauen gainean klik eginda beraien 5 aurrekari probableenen arteko loturak ageri dira kolore berean. Irudian ageri den adibidean *hark* da anafora eta bere 5 aurrekari probableenak *Moneta Fondoak*, *erantzukizun handia*, *egoera desesperatuan dauden giza taldeei*, *errukia* eta *Elkartasuna*.

Irudia 4.2: Anafora fidelak



Irudia 4.3: Anafora pronominalak

Kapitulua 5

Proiektuan barrena

5.1 Sarrera: Anaforaren arazoa

Proiektu hau Lengoaia Naturalen Prozesamendua arloan kokatzen da eta bere helburua anafora pronominal eta fidelei beraien erreferentea esleitzea eta emaitzak MMAX2 tresnanarekin maneiatu ahal izateko modura prestatzea da.

Anaforak eta aurrekariak aipatu dira aurreko kapitulo batean, baina oraindik ez ditugu azaldu proiektuan landu ditugun anafora motak, hots, anafora pronominalak eta anafora fidelak. Demagun pertsona bati buruz hitz egiten gabiltzala eta hainbat alditan bere izena aipatu eta gero bere izenetik deitu beharrean *hura* izenordaina erabiltzen dugula berari buruz gabiltzala adierazteko, ba *hura* anafora pronominala da eta pertsona erreferentziatzen duen aurrekaria. Ikus dezagun adibide bat:

Jose *etxera joan da.* **Hura** *bai dela langilea*

Kasu honetan *hura* anafora pronominala da eta *Jose* aurrekaria. Anafora mota honetan elementu anaforikoa determinatzailea edo izenordaina izan daitezke. Hala ere, anafora papera duen determinatzaileak izenordain funtzioa hartuko du esaldiko beste elementu bati erreferentzia eginez. Erreferentearen eta anaforaren arteko tartea ez da oso zabala izango, irakurleak bestela ez du asmatuko perpauseko ze elementuri egiten ari zaion erreferentzia. Erreferentea elementu pronominalaren atzetik agertu daiteke, kasu honetan anaforari katafora deituko zaio. Adibidez:

Aimarrek **hau** *esan zuen,* **etxera joango zela euria egiten bazuen**

Kataforen agerpena oso urria denez eta prozesamendu guztiz ezberdina behar duenez, gure proiektuan ez ditugu haintzat hartu.

Anafora fidelak, berriz, izenen errepikapen bezala definitu ditzakegu, hots, izen (pertsona nahiz objektuena) bera bi alditan edo gehiagotan ageri bada testuan zehar, izen horiek anafora fidela osatzen dute. Adibidez:

Mikel *atzo gaueko ordubatean joan zen tabernara lanera...*

Mikeli *lana egitea ez zaio asko gustatzen, baina ikasketak ordaintzeko dirua behar duenez ez zaio beste irtenbiderik geratzen...*

Mikel *donostian jaio zen 1983. urtean...*

Bi anafora motak hain ezberdinak izateak bakoitzari trataera guztiz ezberdin bat ematera eraman gaitu. Horrenbestez, gure proiektua 2 ataletan banatu dugu. Lehenengo fasean, corpusean agertzen diren anafora fidel guztiak markatu ditugu. Bigarren fasean, ikasketarako corpus bat erabilita, ikasketa automatikorako *Weka* tresnaren liburutegiak erabiliz anafora pronominalak markatu ditugu corpusean.

5.2 Corpusa

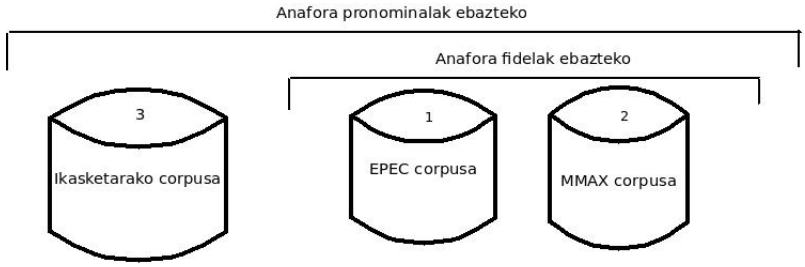
Corpus terminoa era askotara definitu dute jakitunek denboran zehar. Autore batzuek (McEnery and Wilson, 1996) lau ezaugarri eskatzen dizkiote testu-bilduma bati corpustzat hartzeko: lagin adierazgarria izatea, tamaina finitukoa, formatu elektronikoan egotea eta erreferentzia estandarra izatea. Beste batzuek, berriz, (Bach et al., 1997) are baldintza gehiago eskatzen dituzte testu-bilduma bat corpus bezala definitu ahal izateko: hizkuntza batean ematen diren kasu erakusgarri errealen multzo handia izatea, irizpide batzuen arabera bildua, formatu elektronikoan biltegiratua eta informazio linguistikoz hornitua. Biber eta beste batzuek (1998), aldiz, aurrez zehaztutako irizpide batzuen arabera bildutako testu-bilduma gisa definitzen dute. Beste autore batzuek (Kilgarriff and Grefenstett, 2003) definizio are irekiagoa egiten dute, corpusa edozein testu-bilduma dela esanez, eta erabilera askotan beste eskakizun murriztaileago horiek ez direla derrigorrezkoak azpimarratuz. Azken definizio horien arabera paperezko testu-bildumak corpusak badira ere, gaur egun formatu elektronikoan dauden testu-bildumak izendatzeko erabiltzen da batez ere corpus hitza.

5.2.1 Erabilitako corpusak

Gure helburuak betetzeko erabili ditugun corpusek jatorri bera duten arren, EPEC corpusa, hiru corpus ezberdin bezala tratatzea erabaki dugu 5.1 irudian ikusten den modura. Corpus horien arteko

ezberdintasuna idatzita dauden formatua eta EPEC corpusetik hartu den informazio kopurua da. Hona hemen corpus horiei buruzko azalpen laburra:

1. 1000 fitxategi inguruk osatzen duten corpus etiketatua (hemendik aurrera *EPEC* corpusa deituko diogu).
2. Aurrekoaren berdina den MMAX2-rako bertsioa. Aurreko corpusaren fitxategi bakoitzeko direktorio bat dagokio corpus honi (hemendik aurrera *MMAX* corpusa deituko diogu).
3. Anafora pronominalak ebazteko ikasketarako corpusa. Eskuz etiketatutako *EPEC* corpusaren zatia da 50.000 hitz inguru dituelarik.



Irudia 5.1: Corpusak

Lehenengo biak anafora fidelak ebazteko erabili dira eta anafora pronominalak ebazteko, aldiz, 3 corpusak erabili dira.

5.2.1.1 EPEC corpusa

Corpus hau gure lanaren abiapuntua dela esan genezake, bertatik hasten baikara behar dugun informazioa ateratzen anafora fidel zein pronominaletarako. Aurretik azpimarratu dugun bezala corpus hau 1000 fitxategi inguruk osatzen dute eta informazio hori guztia egunkari bateko 2001. urteko testuetatik aterata dago. Fitxategi bakoitza gai bati buruzkoa da eta beraietan jorratzen diren gai aipagarrienak ondorengoak dira: Kirola, Politika, Mundua, Europa, Ekonomia, Gizartea, Nazioartekoa, Agenda eta Udagiroa.

Corpus honek IXA taldeko analisi katetik sortzen den informazioa du, hau da, hitz bakoitzeko analisi morfosintaktikoa du 5.2 irudiak erakusten duen bezala. Bertan hitzaren lema, kategoria, azpikategoria, deklinabide kasua, hitzaren numeroa eta funtzio sintaktikoa daude markatuta besteak beste.

```
"<Sanchez>"<HAS_MAI>"
        <Correct!> "Sanchez" IZE IZB ABS NUMS MUGM HAS_MAI w23,L-A-IZE-IZB-77,lsfi191  @SUBJ %SIB
        <Correct!> "Sanchez" IZE IZB ABS NUMS MUGM HAS_MAI w23,L-A-IZE-IZB-77,lsfi190  @PRED %SIB
        <Correct!> "Sanchez" IZE IZB ABS NUMS MUGM HAS_MAI w23,L-A-IZE-IZB-77,lsfi189  @OBJ %SIB
"<aukeratu>"
        <Correct!> "aukeratu" ADI SIN PART BURU NOTDEK w24,L-A-ADI-SIN-183,lsfi193  @-JADNAG %ADIKATHAS
"<dute>"
        <Correct!> "*edun" ADL A1 NOR_NORK NR_HURA NK_HAIEK-K w25,L-A-ADL-38,lsfi198  @+JADLAG %ADIKATBU
"<berriro>"
        <Correct!> "berriro" ADB ARR ZERO w26,L-A-ADB-ARR-22,lsfi200  @ADLG %SINT
"<idazkari>"
        <Correct!> "idazkari" IZE ARR ZERO w27,L-A-IZE-ARR-412,lsfi203  @KM> %SIH
"<nagusi>"
        <Correct!> "nagusi" ADJ ARR ABS MG w28,L-A-ADJ-ARR-135,lsfi209  @SUBJ %SIB
        <Correct!> "nagusi" ADJ ARR ABS MG w28,L-A-ADJ-ARR-135,lsfi208  @PRED %SIB
        <Correct!> "nagusi" ADJ ARR ABS MG w28,L-A-ADJ-ARR-135,lsfi207  @OBJ %SIB
"<$.>"<PUNT_PUNT>"
```

Irudia 5.2: EPEC corpusa

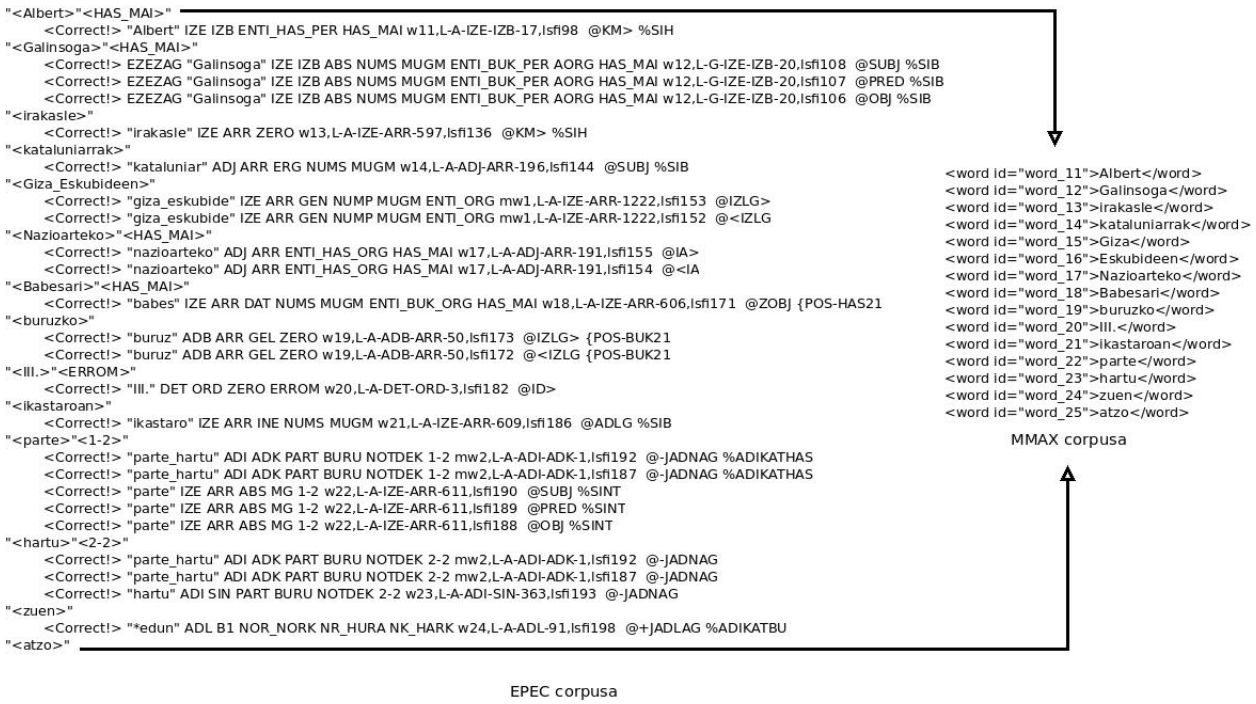
5.2 irudian *Sanchez* hitzaren analisiari erreparatzen badiogu, ikus daiteke 3 analisi posible daudela, hots, desanbiguatu gabe dago. Hala ere, analisi ezberdinen arteko desberdintasun bakarra funtzio sintaktikoa da, beste ezaugarri guztiek balio bera izaten baitute analisi horietan. Guk honelako kasuetan lehenengo analisia hartu dugu beti. Bertan ikusten da *Sanchez* hitzaren kategoria izena (IZE) dela, izen berezia (IZB) dela, absolutiboan (ABS) dagoela, singularra (NUMS) dela, 23. hitza dela (w23), subjektua (@SUBJ) dela eta sintagma bukaera (SIB) dela.

Idazkari hitzaren analisian, berriz, analisi bakarra ageri zaigu eta ez dugu zalantzarik zein hartu behar dugun.

5.2.1.2 MMAX corpusa

1000 direktorio inguruk osatutako multzo honi *corpus* kategoria ematea zalantzan jartzeko modukoa da, batez ere informazio linguistiko gutxi izan duelako gure sistemak anafora fidelak eta pronominalak markatu arte, hots, hasieran izen-sintagmak bakarrik zeuden markatuta. Hasiera batean, MMAX2 tresnarekin direktorio sistema horri dagokion *mmax* luzapena duen fitxategia zabalduz gero, pantailan testu soila ageri da (izen-sintagmak urdinez koloreztatuta ageri dira) beste inolako informaziorik gabe. Hala ere, guk corpus gisa tratatuko ditugu direktorio horiek aurrerantzean.

Aurreko corpuseko fitxategi bakoitzeko honetako karpeta bat dugu, bietan ageri den testua (informazio linguistikoa alde batera utzita) berdina izanik. Alderaketa hori 5.3 irudian ikus daiteke garbiago. Bertan *EPEC* corpuseko testua eta *MMAX* corpuseko testua (*izena.words.mmx.xml* fitxategian biltegiratuta dagoena) konparatzen dira.



Irudia 5.3: EPEC eta MMAX corpusen alderaketa

Bi corpus hauek gure lanaren momentu ezberdinetan erabili dira, lehenengoa abiapuntutzat jo dezakegu eta bigarrena bukaeratzat. Izan ere, gure lana *EPEC* corpusarekin hasi baitugu beharrezkoa zaigun informazio jasoaz eta *MMAX* corpusarekin amaitu hemen idazketak eginez.

5.2.1.3 Ikasketarako corpusa (train.arff)

Corpus hau anafora pronominalen aurrekariak identifikatzeko erabili dugu, hauek ebazteko soilik erabili baitugu ikasketa automatikoa. Bertan 349 anafora pronominalei buruzko informazioa dugu. Anafora bakoitzeko anafora horren eta bere aurrekariaren artean dauden izen-sintagmei buruzko informazio linguistikoa dugu, hau da, anafora bakoitzeko anafora horren informazioa, bere aurrekariarena eta bi hauen arteko izen-sintagmena dugu gordeta. Aipatutako ezaugarri linguistikoetatik at anafora eta gertuko (anafora eta aurrekariaren artean daudenak) izen-sintagma bikote bakoitzeko bestelako informazioa ere gordetzen da, adibidez: bien arteko izen-sintagma distantzia, ea numero bera duten eta ea esaldi berean dauden. Ezaugarri hauei loturazko ezaugarriak deritze.

Corpus hau arff formatuan gordeta dago eta instantziak osatzeko erabili den eredua (Soon et al., 2001) artikuluan proposatzen dutena da. Kontuan izan behar dugu ikasketa automatikoa

sailkatze prozesu bat bezala planteatzen dela eta ondorioz gure ataza horrela diseinatu behar dugula. Anaforaren kasuan, bikoteen eredua erabiltzen da non bi elementuak anafora eta bere erreferentea (edo erreferente posiblea) diren. Demagun gure corpus originalean (arff formatura pasatu baino lehen) ondorengo esaldia dugula:

Jugoslaviarrak bi urte *eman ditu jokatu* ***gabe Erroman*** *eta ikusten da* ***oraindik*** *ez dagoela* ***bere*** *momenturik onenean.*

Esaldi horretan anafora *bere* da eta honen aurrekaria *Jugoslaviarrak*, baina bi hauen artean badaude beste lau izen-sintagma (*bi urte, gabe, Erroman* eta *oraindik*). Ondorioz, kasu honetan arff formatuan gordeko litzatekeen informazioa bost instantziatan edo adibidetan banatuko litzateke. Lehenengo instantzia, *oraindik* hitzaren eta *bere* anaforaren ezaugarriek osatzen dute eta kategoria 0koa da *oraindik* ez delako anaforaren aurrekaria. Bigarren instantzia, *Erroman* hitzaren eta *bere* anaforaren ezaugarriek osatzen dute eta kategoria 0koa da *Erroman* ez delako anaforaren aurrekaria. Hirugarren instantzia, *gabe* hitzaren eta *bere* anaforaren ezaugarriek osatzen dute eta kategoria 0koa da *gabe* ez delako anaforaren aurrekaria. Laugarren instantzia, *urte* hitzaren eta *bere* anaforaren ezaugarriek osatzen dute eta kategoria 0koa da *urte* ez delako anaforaren aurrekaria. Bukatzeko, bostgarren instantzia, *Jugoslaviarrak* hitzaren eta *bere* anaforaren ezaugarriek osatzen dute eta kategoria 1koa da *Jugoslaviarrak* anaforaren aurrekaria delako.

Puntu hau egokia dela iruditzen zaigu gure arff formatuko corpuseko instantzia bat nola dagoen antolatuta azaltzeko. Instantzia bat lau ataletan banatzen da:

1. Izen-sintagmaren informazioa.
2. Izen-sintagmaren eta anaforaren arteko loturazko informazioa
3. Izen-sintagmak ondoren duen anaforaren informazioa.
4. Izen-sintagma hori anafora horren aurrekaria den ala ez zehazten duen balioa (1 baietz, 0 ezetz) kategoria edo klasea deritzona.

Orain badakigu instantzia batek lau atal dituela, baina oraindik ez dugu azaldu atal horietako bakoitzean ze ezaugarri gordetzen diren. Orokorrean guretzat ezaugarri interesgarrienak honako hauek dira: kategoria, azpikategoria, deklinabide kasua, funtzio sintaktikoa, anaforaren eta izen-sintagmaren arteko distantzia, ea izen-sintagmak eta anaforak numero bera duten, ea izen-sintagma eta anafora esaldi berean dauden, izen-sintagma entitate bat bada ea ze entitate mota den, numeroa eta izen-sintagma bada ea hasiera edo bukaera den. Izan ere, anafora pronominalak nahiz fidelak

ebazteko ezaugarri horiek baitira esanguratsuenak (Zelaia et al., 2010), (Arregi et al., 2010) eta (Aduriz et al., 2005) artikuluen arabera.

Izen-sintagma bakoitzeko ondorengo ezaugarriak gorde dira:

- Azpikategoria: ezaugarri honentzat 47 balio ezberdin ditugu aukeran, leku izen berezia (LIB), izen berezia (IZB), izen arrunta (ARR)...
- Deklinabide kasua: 16 balio ezberdin ezaugarri honentzat, absolutiboa (ABS), genitiboa (GEN), datiboa (DAT)...
- Numeroa: 3 dira balio posibleak, mugagabea (MG), singularra (NUMS) ala plurala (NUMP).
- Izen-sintagma mota: 2 aukera posible, bakuna (SINT) ala sintagma bukaera (SIB).
- Funtzio sintaktikoa: 36 aukera ezberdin daude ezaugarri honentzat, predikatiboa (PRED), subjektiboa (SUBJ), objektiboa (OBJ)...
- Entitate mota: 9 kasu ezberdin ditugu ezaugarri honetarako, entitatea lekua denean (ENTI_LOC), entitatea erakunde bat denean (ENTI_ORG) eta entitatea pertsona bat denean (ENTI_PER). Instantzietan ezaugarri hau loturazko ezaugarrien ondoren doan arren izen-sintagmari dagozkion ezaugarrien artean sartzen da.

Anafora eta izen-sintagma bikote bakoitzeko loturazko ezaugarriak ondorengoak dira:

- Distantzia: anaforaren eta izen-sintagmaren artean beste 4 izen-sintagma badaude, distantzia 5 izango da, adibidez. Distantzia maximoa 15 izen-sintagmara mugatu dugu normalean anaforaren eta aurrekariaren arteko distantzia hori baino txikiagoa izaten delako.
- Esaldi berean: 1ekoa baiezko kasuan, 0 ezezkoan.
- Numero bera: anaforaren eta izen-sintagmaren numeroa berdina bada 3, desberdinak badira 0. Izen-sintagmaren numeroa ezezaguna bada, balioa 1 da. Izen-sintagma entitatea bada, bere numeroa mugagabea eta anafora singularra, balioa 2 da.

Anafora bakoitzari buruz, aldiz, ondorengo informazioa interesatzen zaigu:

- Deklinabide kasua: 16 aukera ezberdin izen-sintagmen kasuan bezala.
- Funtzio sintaktikoa: 36 aukera ezberdin izen-sintagmen kasuan bezala.
- Izen-sintagma mota: 3 aukera posible, bakuna (SINT), sintagma hasiera (SIH) ala sintagma bukaera (SIB).
- Numeroa: 3 dira balio posibleak, mugagabea (MG), singularra (NUMS) ala plurala (NUMP).

Bukatzeko, instantzia baten azken atala geratzen zaigu azaltzeko:

- Klasea edo kategoria: atal hau elementu bakarrak osatzen du eta instantzia horretako izen-sintagma anaforaren aurrekaria den ala ez adierazteko erabiltzen da.

Laburbilduz, corpusa 349 instantzia positibok eta 600 instantzia negatibo inguruk osatzen dute eta instantzia hauetako bakoitza 4 ataletan dago banatuta: izen-sintagmaren ezaugarriak, loturazko ezaugarriak, anaforaren ezaugarriak eta klasea. Azaldutakoa argi geratu dadin Jugoslaviarraren adibidearekin bukatuko dugu bertan sortuko liratekeen 5 instantziak irudikatzen dituen irudia (5.4 irudia) erakutsiz:

Irudia 5.4: Arff formatuko instantziak

5.3 Anafora fidelak

5.3.1 Sarrera

Anafora fidelei beraien erreferenteak esleitzeko prozesua *EPEC* corpusean hasten da eta eman beharreko urratsak lau dira:

- Testuan agertzen diren izen-sintagmak identifikatu.
- Izen-sintagma bakoitzaren gunea identifikatu eta honen lema gorde.
- Gune berdina duten izen-sintagmak anafora fidel beraren osagai bezala markatu.
- Idazketak egin *MMAX* corpusean.

5.3.2 Anafora fidelak ebazteko aplikazioaren diseinua

Aurretik esan dugu anafora fidelei beraien erreferenteak esleitzeko prozesua *EPEC* corpusean hasten dela, corpus horretan daudelako markatu nahi ditugun anaforak. IXA taldeko hizkuntzalariek

corpus horretako zenbait fitxategi markatuta dauzkate jadanik. Horrenbestez, gure aplikazioak markatu gabe dauden fitxategiak bakarrik aukeratu beharko ditu. Horretarako markatuta daudenen izenak idatzita dauden zerrenda bat aztertzen dugu uneko fitxategia aukeratu behar dugun ala ez erabakitzeko.

Orain markatu behar ditugunak filtratuta ditugunez, eman behar dugun hurrengo pausoa fitxategiak deskonprimitzea da corpus hori osatzen duten fitxategiak konprimituta daude eta.

Behin aztertu nahi ditugun fitxategiak deskonprimituta daudenean, fitxategi hauetako bakoitza banan-banan irakurri behar dugu beraietan anafora fidelak eta aurrekariak identifikatzeko. Ataza hori burutzeko eman beharreko lehenengo urratsa fitxategi bakoitzeko **izen-sintagmak identifikatzea** da, anafora fidelak izen-sintagmez osatuta daudelako.

Fitxategi bateko izen-sintagma bat identifikatuta dugunean, sintagma horren ***gunea* zein den zehaztu** behar dugu heuristiko baten bidez. Har dezagun ondorengo adibidea:

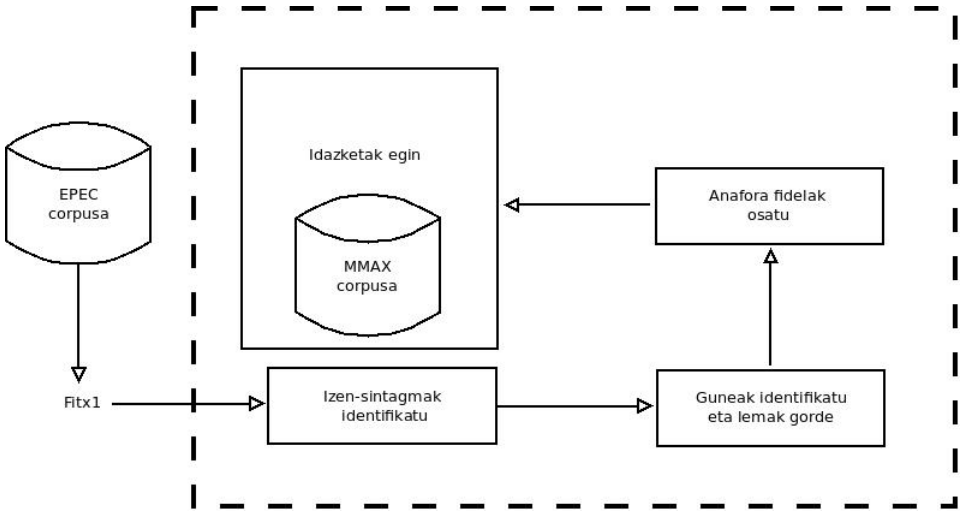
> *Zure alabaren etxeko* ***atea*** *oso polita da. Etxea oso handia da eta leihoak ere oso politak dira, baina* ***atea*** *benetan harrigarria da...*

Aurreko esaldian *Zure alabaren etxeko atea* izen-sintagma hartzen badugu, argi ikusten dugu bere gunea *atea* dela esaldiaren atal hori ate bati buruz dabilelako eta ez alaba bati buruz, ezta etxe bati buruz ere. Esaldian beranduago agertzen den izen-sintagmaren gunea ere atea da izen-sintagma hau izen bakarrak osatzen du baitu. Kasu honetan, gune berdina duten bi izen-sintagmek anafora fidel bat osatzen dute.

Izen-sintagma baten *gunea* zein den zehaztu dugunean egin behar dena *gune* horren lema gordetzea eta zenbatgarren hitza den gordetzea da. Fitxategi bateko izen-sintagma guztien lemak eta hauen hitz zenbakiak gorde ditugunean konparaketak egin behar ditugu, hots, lema guztiak konparatu behar dira berdinak direnak **anafora fidel beraren atal bezala gorde**tzeko.

Fitxategi bateko anafora fidel bakoitza osatzen duten *guneen* hitz zenbakiak gordeta ditugunez, fitxategi horren izen bera duen *MMAX* corpuseko karpetan dagokion lekuan **idazketak egiten dira**. Corpuseko aukeratutako fitxategi guztiekin prozesu berdina jarraitu ondoren amaitu egiten da anafora fidelak ebazten dituen fasea.

Atal honetan adierazitakoa 5.5 irudian argiago ikus daiteke:

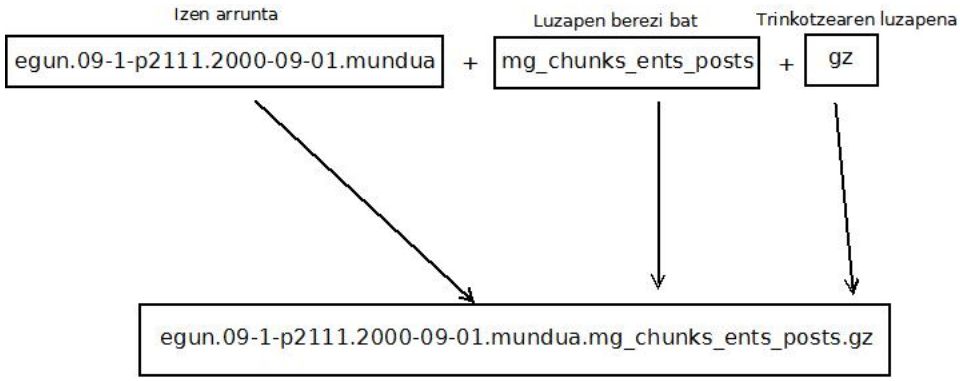


Irudia 5.5: Anafora fidelak ebazteko sistemaren diseinua

5.3.3 Anafora fidelak ebazteko aplikazioaren inplementazioa

Anafora mota hau tratatzeko sistema eraikitzeko abiapuntua IXA taldeko zerbitzarietan gordeta dagoen *Korreferentzia_originala* direktorioan aukitzen da. Karpeta horren barruan dago biltegiratuta 1111 fitxategik osatzen duten *EPEC* corpusa. Hala ere, karpeta horren barruan badaude beste hainbat fitxategi interesatzen ez zaizkigunak eta horrenbestez hautaketa bat egiten dugu denen artean guri interesatzen zaizkigunak soilik hartzeko.

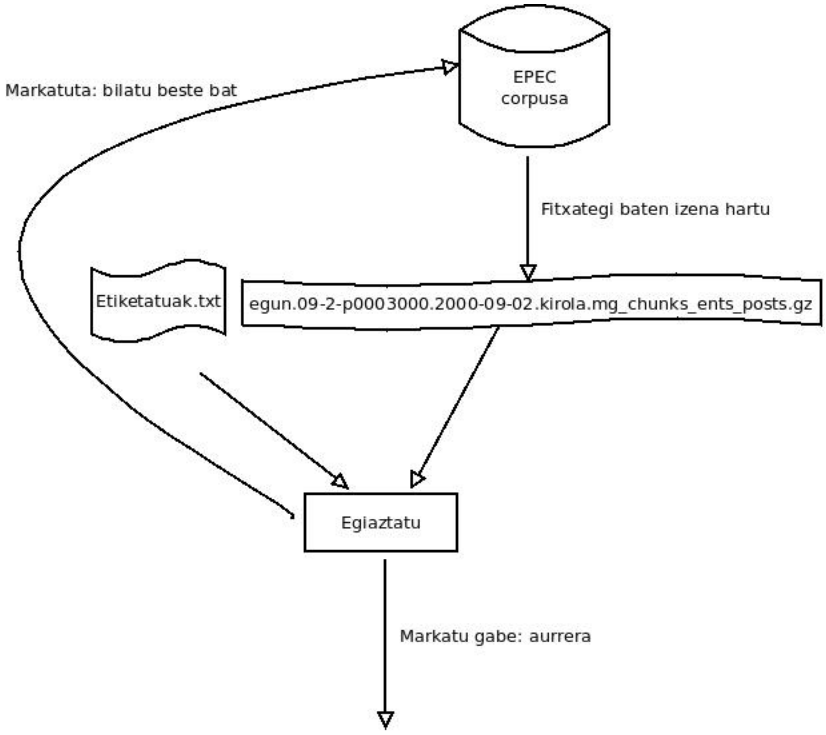
Corpus hori osatzen duten fitxategi guztien izenak hiru ataletan daude banatuta 5.6 irudian ageri den bezala. Izen banaketa horrek bereizten ditu interesatzen zaizkigun fitxategiak eta gainerakoak.



Irudia 5.6: EPEC corpuseko fitxategien izenak

Fitxategi bakoitzaren izena izen arrunt batekin hasten da, ondoren luzapen berezi bat dator eta trinkoketa luzapen batekin amaitzen da. Gure sistemak *Korreferentzia_originala* izeneko karpeta osoa iragaten du eta izenen patroi hori betetzen duen fitxategi bat aurkitzen duenean hurrengo urratsa ematen du.

Gure sistemak patroi jakin hori betetzen duen fitxategi bat aurkitzen duenean, beste egiaztapen bat burutu behar du. Zerrenda batean (*Etiketatuak.txt*) fitxategi horri dagokion *MMAX* corpuseko fitxategia hizkuntzalariek eskuz markatu duten ala ez egiaztatzen du. Fitxategi horren gainean anafora fidelak edo pronominalak markatuta badaude, fitxategi hori baztertu egiten da eta beste bat bilatzen da 5.7 irudiak erakusten duen bezala.



Irudia 5.7: Egiaztapena

5.3.3.1 Izen-sintagmak identifikatu

Aplikazioak markatu gabe dagoen fitxategi bat aurkitzen duenean, prest dago fitxategi horretako anafora fidelak identifikatzeko. Horretarako, fitxategiaren informazio guztia irakurtzen du bertan agertzen diren izen-sintagmak bilatuz. *EPEC* corpuseko izen-sintagmak bi motatakoak izan daitezke: hitz bakar batek edo hitz elkarketa batek osatutakoak eta hitz batek baino gehiagok osatutakoak. Ikus 5.8 irudia.

Hitz batek baino gehiagok osatutako izen-sintagma

```
"<ZALEEN>"<DEN_MAI>"
      <Correct!> "zale" ADJ ARR GEN NUMP MUGM ZERO DEN_MAI w24,L-A-ADJ-ARR-100,lsfi114  @IZLG> {POS-HAS14 %SIH
      <Correct!> "zale" ADJ ARR GEN NUMP MUGM ZERO DEN_MAI w24,L-A-ADJ-ARR-100,lsfi113  @<IZLG {POS-HAS14 %SIH
"<aurreko>"
      <Correct!> "aurre" IZE ARR GEL NUMS MUGM w25,L-A-IZE-ARR-6254,lsfi124  @IZLG> {POS-BUK14
      <Correct!> "aurre" IZE ARR GEL NUMS MUGM w25,L-A-IZE-ARR-6254,lsfi123  @<IZLG {POS-BUK14
"<aurkezpen>"
      <Correct!> "aurkezpen" IZE ARR ZERO w26,L-A-IZE-ARR-6256,lsfi138  @KM>
"<partidua>"
      <Correct!> "partidu" IZE ARR ABS NUMS MUGM w27,L-A-IZE-ARR-6260,lsfi148  @SUBJ %SIB
      <Correct!> "partidu" IZE ARR ABS NUMS MUGM w27,L-A-IZE-ARR-6260,lsfi147  @PRED %SIB
      <Correct!> "partidu" IZE ARR ABS NUMS MUGM w27,L-A-IZE-ARR-6260,lsfi146  @OBJ %SIB
```

Hitz batekin osatutako izen-sintagma

```
"<Alavesek>"<HAS_MAI>"
      <Correct!> "Alaves" IZE LIB ERG NUMS MUGM ENTI_ORG HAS_MAI w64,L-A-IZE-LIB-1826,lsfi383  @SUBJ %SINT
```

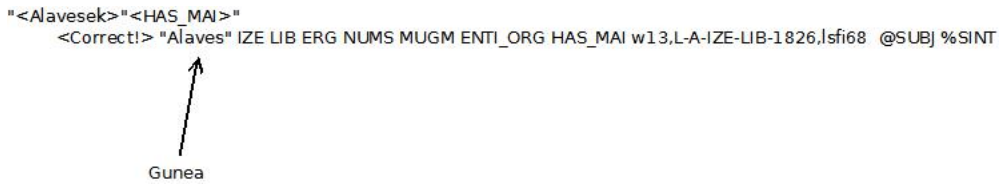
Irudia 5.8: Izen-sintagma motak

Sistemak izen-sintagma bat identifikatzeko, hitz bakoitzaren informazioa ageri den lerroan bilaketa bat egiten du *SINT* edo *SIH* markak aurkitzeko asmoz. Hauek adierazten baitute izen-sintagma baten hasiera. *SINT* marka aurkitzen duenean ez du sintagmaren bukaera bilatzen jarraitzen, sintagma hori hitz bakar batez edo hitz elkarketa batez osatuta dagoelako. *SIH* marka aurkitzen duenean, aldiz, izen-sintagma hori hitz batek baino gehiagok osatzen dutela esan nahi duenez, *SIB* marka aurkitu arte jarraitzen du tarteko hitz guztien informazioa gordez.

5.3.3.2 Guneak identifikatu

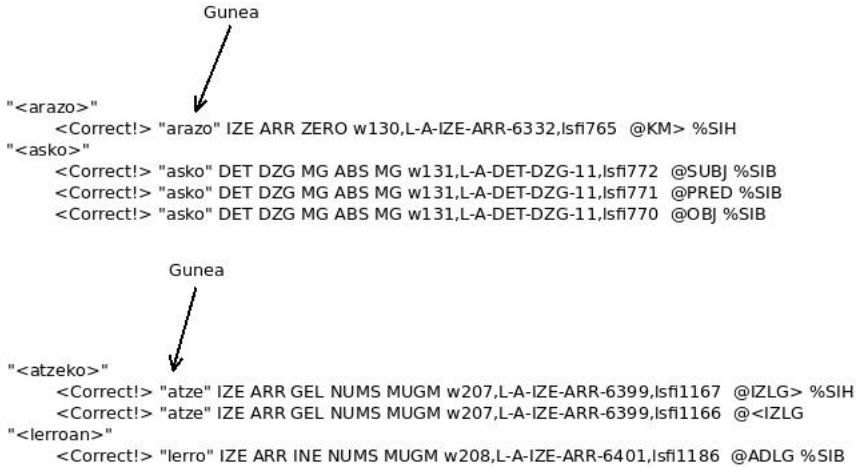
Fitxategian idatzita dagoen izen-sintagma bat identifikatuta dagoenean, sistemak horren *gunea* identifikatu behar du. Horretarako, hiru kasu bereiztuko ditugu:

- **Izen-sintagma hitz bakar batek edo hitz elkartu batek osatzen dute:** Kasu honetan sintagmaren *gunea* hitz bakar edo hitz elkartu hori da. Adibidez:



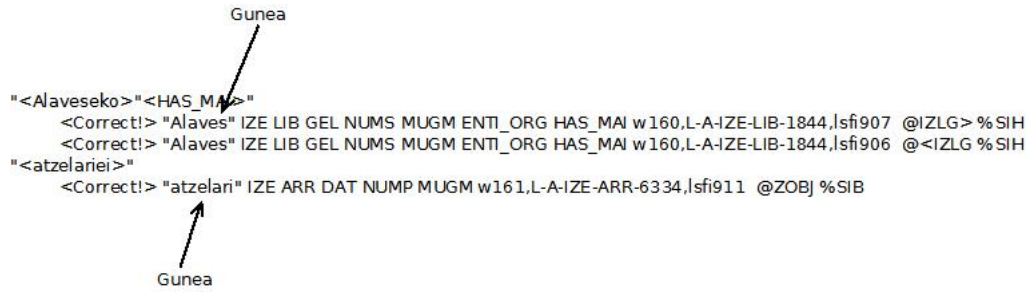
Irudia 5.9: Izen-sintagma bakunen gunea

- **Izen-sintagma hitz batek baino gehiagok osatzatzen dute:** Kasu honetan sintagmaren *gunea* heuristiko baten bidez bilatzen da. Heuristikoa honetan datza:
  - Sintagma osatzen duten hitzen datuen artean *@KM* marka ageri bada, marka hori duen hitza da *gunea*.
  - Bestela, bere datuen artean ">" edo "<" markak ez dituen hitza da *gunea*.



Irudia 5.10: Izen-sintagma konposatuen gunea

- **Hitz batek baino gehiagok osaturiko izen-sintagma baten hitzen artean entitate bat dagoenean:** Kasu honetan entitate hori *gune* bezala markatzen da. Gure heuristikoan irizpide hau erabiltzea erabaki dugu testu batean entitate bat agertzen denean oso probablea delako berriro entitate hori errepikatzea testuan anafora fidel bat eratuz. Modu honetara posible da izen-sintagma batek *gune* bat baino gehiago izatea posible delako 2. kasua eta azken hau biak batera betetzea.

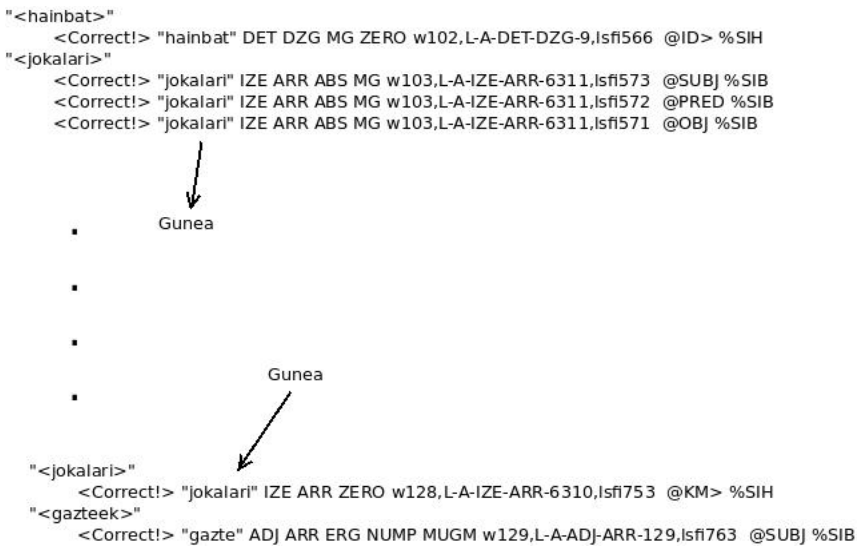


Irudia 5.11: Izen-sintagma konposatuen gunea entitatea denean

Izen-sintagma baten *gunea* (edo *guneak*) identifikatu ondoren, horren lema eta hitz zenbakia gordetzen dira. Lema beste izen-sintagmen guneen lemekin konparatzeko gordetzen da gero anafora fidelak bilatzeko. Hitz zenbakia, ostera, *MMAX* corpusean idazketak egiteko behar izango dugulako gordetzen da.

5.3.3.3 Anafora fidelak osatu

Sistemak fitxategi bateko izen-sintagmen *gune* guztien lemak eta hitz zenbakiak gordeta dituenean, lema guztien artean konparaketa bat egiten du. Lema bera duten *guneak* anafora fidel beraren atal bezala kontsideratuko ditu sistemak eta ondorioz, *gune* horietaz osatuta dauden izen-sintagmak ere anafora fidel beraren atal kontsideratuko ditu 5.12 irudian ikusten den bezala.

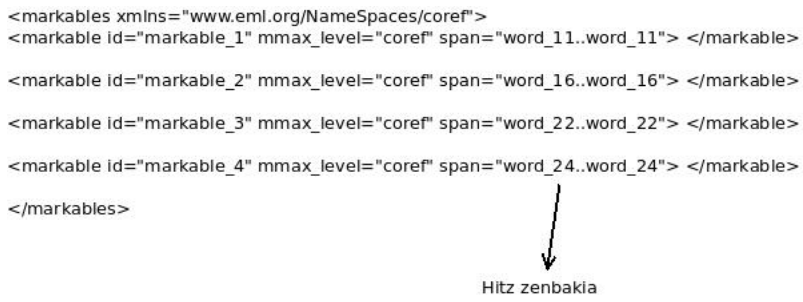


Irudia 5.12: Gune bera duten izen-sintagmak

Puntu honetan, fitxategi bateko anafora fidel guztiak identifikatuta ditugu, baina gure anafora fidelen ebazpena guztiz bukatzeko fitxategi bakoitzari dagokion *MMAX* corpuseko karpetan zenbait idazketa egitea falta da.

5.3.3.4 Idazketak egin *MMAX* corpusean

Aurretik aipatu dugu *EPEC* corpuseko fitxategi bakoitzeko *MMAX* corpusean izen berdina duen karpeta bat dagoela *EPEC* corpuseko fitxategien hitzak *MMAX2* anotazio tresnarekin tratatu ahal izateko. Horrenbestez, *EPEC* corpuseko fitxategi batean identifikatutako anafora fidelak fitxategi horri dagokion *MMAX* corpuseko fitxategian isladatzeko, izen berdina duen karpeta aurkitu behar du sistemak. Karpeta zuzena aurkitu ondoren, honen barnean kokatuta dagoen *Markables* karpeta barruan sartu behar da aplikazioa fitxategi egokia atzitzeko. Idazketak egin behar diren fitxategi horietan hainbat lerro daude idatzita *<markables>* eta *</markables>* xml-ko mugen artean 5.13 irudian ikusten den bezala.



Irudia 5.13: coref_level fitxategia

Lerro horietatiko bakoitzean ageri den informazio guztia hartu behar da kontuan, baina hasiera batean guri interesatzen zaigun informazioa *span* hitzaren ondoren kakotx artean dagoena da. Atal horretan baitago adierazita izen-sintagma bakoitza ze hitz zenbakitik ze hitz zebakira doan *EPEC* corpuseko gure aukeratutako fitxategian. Modu honetara, *MMAX* corpuseko fitxategi bat *MMAX2* tresnarekin irekitzean fitxategi horretako izen-sintagma guztiak markatuta agertuko dira kolore urdinarekin. Hau jakinda, aukeratutako fitxategiko anafora fidelen *guneek* ze hitz zenbaki duten gordeta dugunez, erraza da jakitea *MMAX* corpusean ze izen-sintagmaren atal diren.

Fitxategian anafora fidel bakoitza eratzen duten izen-sintagmak identifikatuta ditugunean, multzo berean daudela idaztea eta anafora fidelak direla adieraztea bakarrik falta da 5.14 irudian ikusten den bezala.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE markables SYSTEM "markables.dtd">
<markables xmlns="www.eml.org/NameSpaces/coref">
<markable id="markable_34" span="word_1..word_2" type="none" coref_class="set_0" mmax_level="coref" number="S" />
<markable id="markable_65" span="word_115..word_116" type="anaphoric" coref_class="set_0" exp_type="anafora fidelak"
mmax_level="coref" number="S" />
<markable id="markable_74" span="word_143..word_144" type="none" coref_class="set_0" mmax_level="coref" number="S" />
<markable id="markable_87" span="word_184..word_185" type="none" coref_class="set_0" mmax_level="coref" number="S" />
<markable id="markable_112" span="word_290..word_290" type="none" coref_class="set_0" mmax_level="coref" number="S" />
<markable id="markable_194" span="word_604..word_604" type="none" coref_class="set_0" mmax_level="coref" number="S" />
<markable id="markable_35" span="word_3..word_3" type="none" coref_class="set_1" np_form="pper" mmax_level="coref"
number="none" />
<markable id="markable_183" span="word_541..word_545" type="anaphoric" coref_class="set_1" np_form="pper"
exp_type="anafora fidelak" mmax_level="coref" number="none" />
<markable id="markable_36" span="word_6..word_7" type="none" coref_class="set_2" np_form="defnp" mmax_level="coref"
number="S" />
<markable id="markable_66" span="word_121..word_122" type="anaphoric" coref_class="set_2" np_form="defnp"
exp_type="anafora fidelak" mmax_level="coref" number="S" />
<markable id="markable_1" span="word_11..word_11" type="none" coref_class="set_3" mmax_level="coref" />
<markable id="markable_7" span="word_63..word_63" type="anaphoric" coref_class="set_3"
exp_type="anafora fidelak" mmax_level="coref" />
<markable id="markable_71" span="word_136..word_138" type="none" coref_class="set_3" np_form="ne"
mmax_level="coref" number="S" />
<markable id="markable_15" span="word_162..word_162" type="none" coref_class="set_3" mmax_level="coref" />
<markable id="markable_18" span="word_207..word_207" type="none" coref_class="set_3" mmax_level="coref" />
<markable id="markable_125" span="word_355..word_355" type="none" coref_class="set_3" np_form="ne"
mmax_level="coref" number="S" />
<markable id="markable_26" span="word_384..word_384" type="none" coref_class="set_3" mmax_level="coref" />
<markable id="markable_29" span="word_493..word_493" type="none" coref_class="set_3" mmax_level="coref" />
<markable id="markable_30" span="word_528..word_528" type="none" coref_class="set_3" mmax_level="coref" />
```

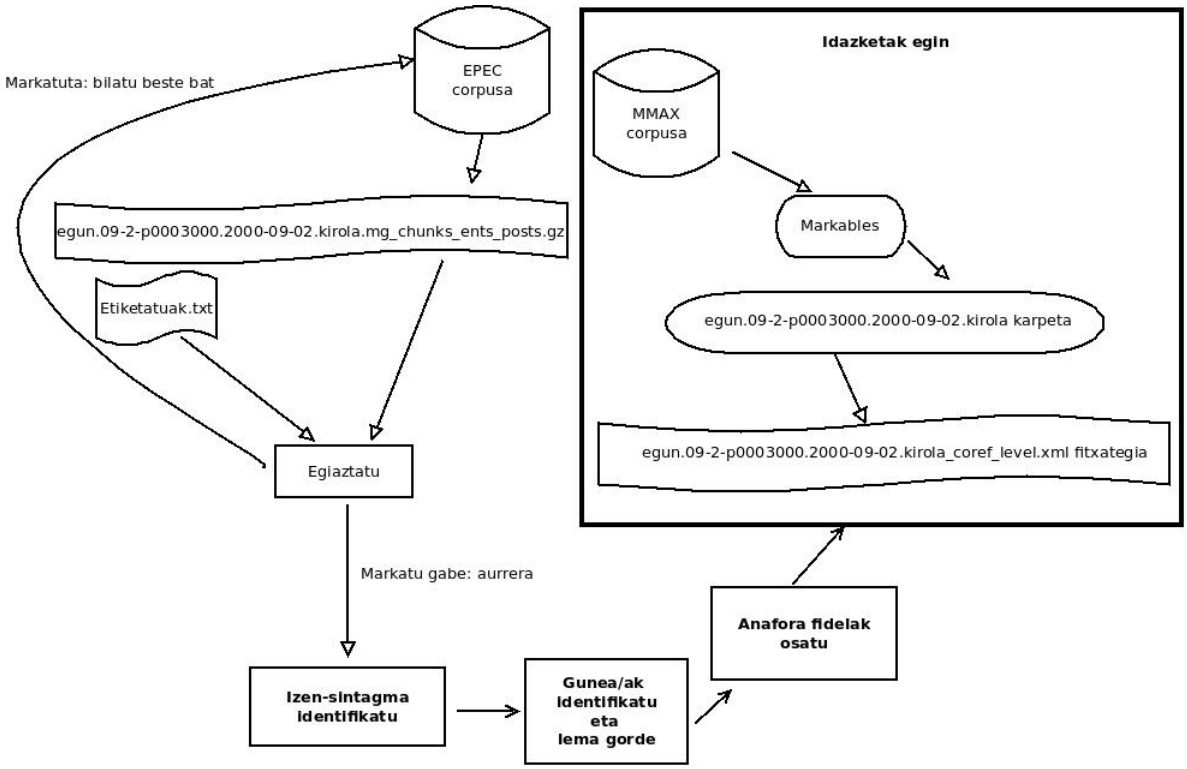
Irudia 5.14: Anafora fidelak eratzen

Aurreko irudian ageri dena azaltzearren, lau anafora fidel daude markatuta, lau lerro ezberdinetan dagoelako *exp_type="anafora fidelak"* idatzita. Anafora fidel horietako bakoitza set edo multzo desberdin batekoa da: gure kasuan 0,1,2 eta 3 multzoak ageri dira guztira lau anafora fidel daudelako. Anafora fidel bat atal batekin baino gehiagorekin dago osatuta. Ondorioz, anafora fidela osatzen duten atalak multzo berekoak direla adierazi behar da multzo zenbaki edo set zenbaki berdinarekin identifikatuz. Gure adibidean, lehenengo anafora osatzen duten sei atalek 0 zenbakia dute multzoaren zenbaki bezala (*coref_class="set_0"*).

Anafora fidelak ebazten dituen sistemak bera lana burutzen duenean posible da honek markatu dituen anafora fidelak MMAX2 tresnarekin grafikoki ikustea 5.15 irudian ageri den bezala. Anafora fidel bakoitzaren atal bat gorriz ageri da hizkuntzalariak errazago identifikatu dezan. Atal horren gainean klik eginez gero anafora hori osatzen duten atalen arteko loturak ikus daitezke.

Irudia 5.15: Anafora fidelak

Ondorengo puntuan anafora pronominalen ebazpena jorratuko da. Hala ere, ezer baino lehen, anafora fidelak ebazten dituen sistemaren arkitektura osoa aurkeztu nahi diogu irakurleari 5.16 irudiarekin, egin dugunaren ideia argiagoa izan dezan.



Irudia 5.16: Anafora fidelak ebazteko sistemaren arkitektura

5.4 Anafora pronominalak

5.4.1 Sarrera

Ikasketa automatikoa erabilita anafora pronominalei beraien aurrekariak esleitzeko prozesua ere *EPEC* corpusean hasten da. Kasu honetan ere fitxategiak banaka tratatu behar dira eta fase hau 4 azpifasetan banatzen da:

- **Testerako fitxategia sortu:** Fitxategi bakoitzeko ikasketarako corpusak (train.arff) dituen atributu berdinak dituen test fitxategi bat sortu.
- **Train fasea:** Ikasketarako corpusarekin eredu bat (.model bat), hots, sailkatzaile bat sortu gure aplikazioak behin eta berriro hasieratik ikas ez dezan. Modu honetara ikasketa prozesua behin bakarrik egiten da eta denbora aurrezten da.

- **Test fasea eta emaitzen interpretazioa:** Aurreko faseetan lortutako test fitxategiarekin eta sailkatzailearekin, gure aplikazioak emaitzak bueltatzen dizkigu. Emaitza horiek formatu berezi batean daudenez idatzita, formatu horretatik interesatzen zaizkigun atalekin geratuko gara eta formatu egokiago batean idatziko ditugu.
- **Emaitzen idazketa:** Azken fase honetan, aurreko fasean lortutako emaitzak *MMAX* corpusean idatziko ditugu.

Ondorengo puntuetan urrats bakoitzaren diseinu eta erabilera zehatza azalduko da, dagozkion sarrera eta irteera formatuak ere azalduz.

5.4.2 Anafora pronominalak ebazteko aplikazioaren diseinua

Anafora fidelen kasuan gertatzen den bezala, anafora pronominalei beraien aurrekaria automatikoki esleitzeko prozesua *EPEC* corpusean hasten da. Kasu honetan ere iragazki bat pasatu behar da hizkuntzalariek markatu ez dituzten fitxategiekin soilik geratzeko, zenbait fitxategitan jadanik anafora fidelak eta pronominak markatuta baitaude. Uneko fitxategiaren izena jakinda, markatuta dauden fitxategien izenak idatzita dauden zerranda batean begiratu behar da ea uneko fitxategi hori markatuta dagoen ala ez. Markatuta badago hurrengo fitxategia hartzen da eta markatu gabe badago prozesuarekin jarraitzen da.

EPEC corpusa osatzen duten fitxategietako bat markatu gabe dagoela egiaztatu dugunean, hau deskomprimitu egin behar da. Orain gure sistemak eskuragarri du fitxategi horren informazioa eta hurrengo urratsa eman dezake.

Gure sistemak anafora pronominalak ebazteko ikasketa automatikoa erabiltzen duenez, beharrezkoak ditu ikasteko corpus bat eta testerako beste bat, biak arff formatuan. Ikasketan erabiltzeko corpusa aurretik aipatu dugun *Ikasketarako corpusa (train.arff)* da eta testerakoa *EPEC* corpuseko deskonprimitutako fitxategitik abiatuta sortu behar da. Argiago esateko, markatu behar diren fitxategi guztiak sailkatuko dira ikasketarako corpusetik ikasi duen sailkatzaile berarekin, baina testerako fitxategia fitxategi bakoitzetik interesatzen zaizkigun ezaugarriekin sortuko da bakoitzarentzako berea.

Deskonprimitutako fitxategi bakoitzari dagokion testerako corpusa (edo fitxategia) lortzeko sistemak burutu behar duen lehenengo pausoa **anafora pronominalak identifikatzea** da. Hori lortzeko inplementazio atalean azalduko diren zenbait irizpide jarraitu behar dira. Behin anafora pronominal bat identifikatuta dugunean, honi buruz interesatzen zaigun **informazioa eskuratzen dugu** fitxategitik (ez ahaztu fitxategi hauetan hitz bakoitzari buruzko informazio ugari dugula).

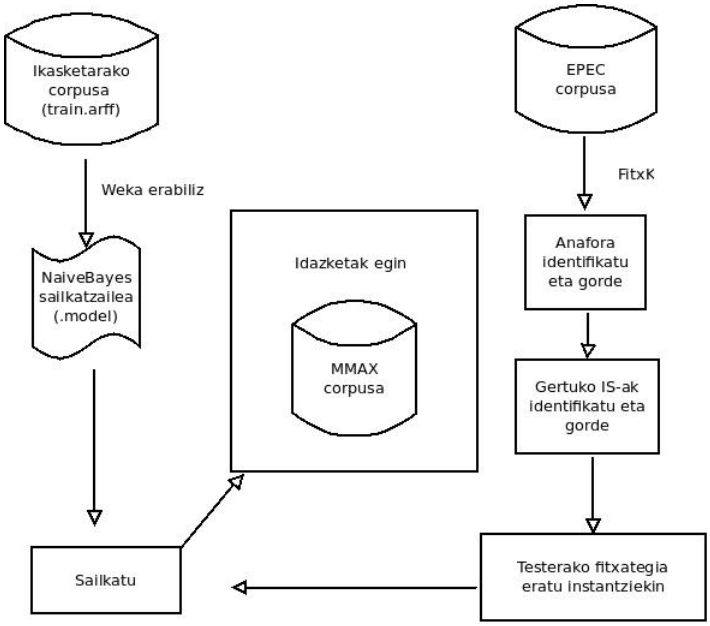
Sistemak anafora pronominal baten informazioa biltegiratu duenean eman behar duen hurrengo urratsa anafora pronominal horretatik **distantzia jakin baten** barruan dauden **izen-sintagmen informazioa** eskuratzea da. Izen-sintagmen informazioa lortzea erabaki dugu anafora pronominal baten aurrekaria gehienetan izen-sintagma bat izaten delako.

Puntu honetan anafora pronominal bakoitzeko distantzia jakin baten barruan dauden izen-sintagmen informazioa gorde dugunez, posible dugu orain anaforen eta beraien zortzi aurrekari posibleak diren izen-sintagmen informazioarekin arff formatua duen **testerako fitxategia prestatzea** bertan **instantziak osatuz**.

Ikasketa automatikorako modulua Javaz inplementatuta dago eta bere abiapuntua *Ikasketarako corpusean* oinarrituta Weka tresnarekin sortu dugun sailkatzailea da. Sailkatzaile hori NaiveBayes algoritmoarekin eraiki dugu eta bere eginbeharra testerako fitxategian aurkitzen dituen instantziak edo adibideak sailkatzea da, hau da, instantzien klaseak iragartzea. Sailkatzaileak ematen duen iragarpen horrek hainbat atal ditu, baina guri benetan interesatzen zaiguna izen-sintagma anafora pronominalaren aurrekaria izateko probabilitatea da (ez ahaztu instantzia bat izen-sintagma baten eta anafora baten datuek eratzen dutela).

Anafora pronominal bakoitzeko aurrekariak izan daitezkeen izen-sintagma guztien probabilitateak kalkulatu direnean probabilitate handiena ematen duten 5 izen-sintagmak hartzen dira.

Bukatzeko, sistemak egin behar duena ikasketa automatikorako moduluak fitxategi bakoitzerako lortu dituen emaitzak **MMAX corpusean idaztea** da. Emandako azalpenak ondo barneratzeko, komenigarria dela uste dugu 5.17 irudia aztertzea.



Irudia 5.17: Anafora pronominalak ebazteko diseinua

Irudi honekin bukatuko ditugu gure sistemaren diseinuari buruzko azalpen guztiak. Ondorengo puntuetan proiektuan barrena gehiago murgilduko gara eta honi buruzko informazio zehatzagoa azalduko dugu, proiektuaren inplementazioari buruzko informazioa emateko garaia da eta.

5.4.3 Anafora pronominalak ebazteko aplikazioaren inplementazioa

Anafora fidelak eta pronominalak ebazteko aplikazioek ematen dituzten lehenengo urratsak berdinak direnez ez dira hemen berriro azalduko urrats horiek.

Sistemak uneko fitxategia markatu gabe dagoela (*Etiketatuak.txt* fitxategiaren bitartez) egiaztatu duenean eta hau deskonprimitu duenean hasten dira anafora fidelak ebazteko sistemaren eta anafora pronominalak ebazteko sitemaren arteko ezberdintasunak. Aurrerago aipatu dugu anafora pronominalak tratatzen dituen sistema lau azpiataletan edo azpifasetan banatuta dagoela. Ondorioz, honen inplementazioari buruzko azalpena ere lau ataletan banatu dugu.

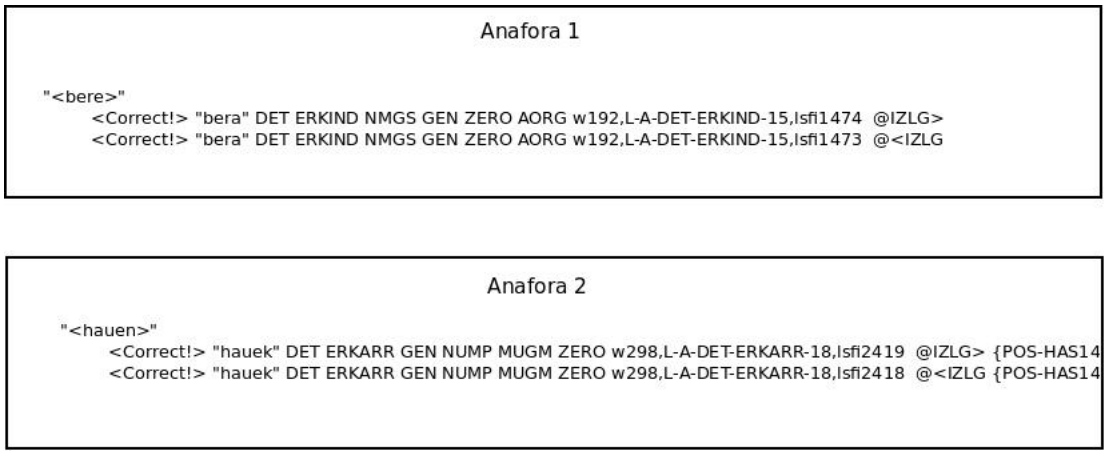
5.4.3.1 Testerako fitxategia sortu

Prozesu hau burutzeko sistemak ematen duen pausoa *EPEC* corpuseko fitxategian anaforak bilatzea da. Horretarako, fitxategia lerroz lerro irakurtzen du ondorengo baldintza betetzen duen hitz bat

aurkitu arte:

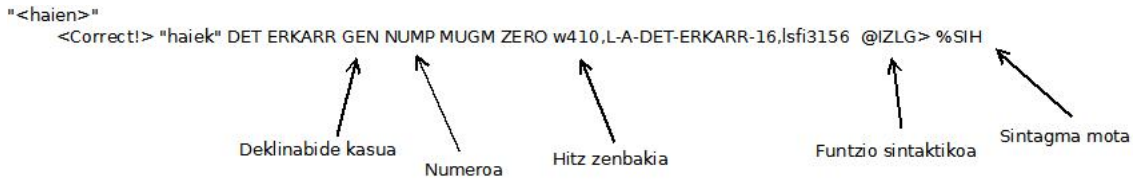
- Bere lema *hau*, *hori*, *hura*, *hauek*, *horiek*, *haiek*, *bera*, *beraiek* edo *eurek* izatea eta bere ezaugarrien artean *DET ERKARR*, *IOR PERARR* edo *DET ERKIND* izatea, hau da, determinatzaile erakusle arruntak, erakusle indartuak edo izenordain pertsonal indartuak direnean.

5.18 irudian adibide pare bat ageri da:



Irudia 5.18: Anafora pronominala izateko baldintzak

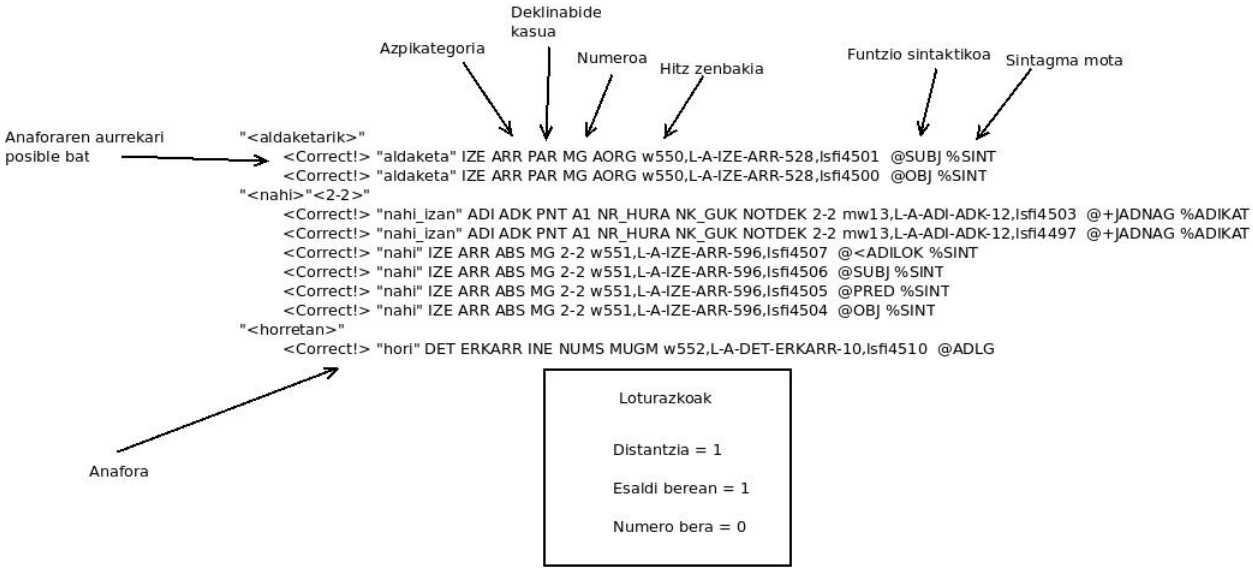
Aplikazioak anafora pronominal bat identifikatu duenean, berari buruz interesatzen zaizkigun datuak gordetzen ditu. Datu horiek anaforaren deklinabide kasua, funtzio sintaktikoa, hitz zenbakia, sintagma mota (*SIH*, *SINT* ala *SIB*) eta numeroa dira:



Irudia 5.19: Anafora pronominalaren datuak

Hurrengo urratsa anafora pronominalaren aurrekari posibleak bilatzea da. Aurrekari posibleak dira anaforatik 8 izen-sintagma distantziaren barruan dauden izen-sintagma guztiak. 8 izen-sintagmako distantzia aukeratu dugu aurrekaria kasuen %97tan (*Ikasketarako corpusetik* atera da datu hau) distantzia horren barruan egoten delako.

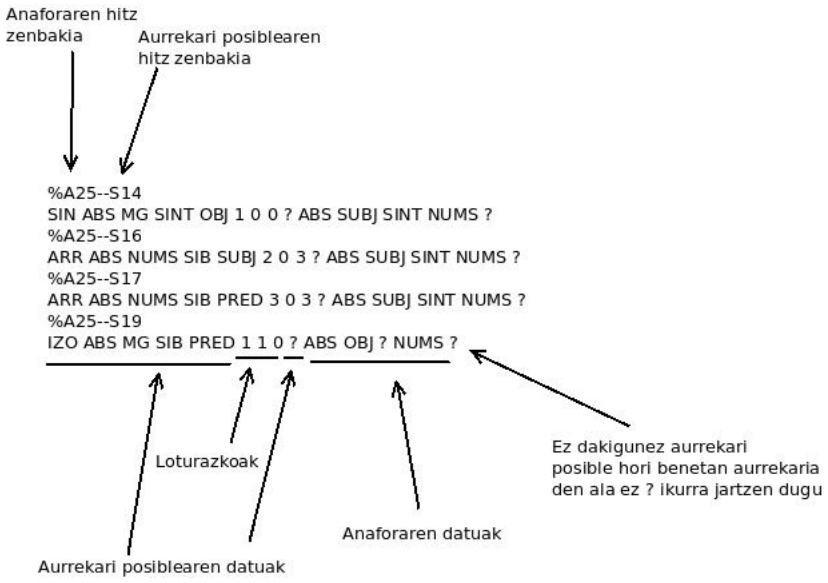
Aurrekari posible bakoitzeko ondorengo informazioa gordetzen du sistemak: azpikategoria, deklinabide kasua, numeroa, sintagma mota, funtzio sintaktikoa eta izen-sintagma entitatea bada ze entitate mota den. Gainera aurrekari posiblearen eta dagokion anaforaren arteko loturazko datuak ere gordetzen ditu sistemak: aurrekari posibleak anaforarekiko duen distantzia, ea anafora eta izen-sintagma esaldi berean dauden eta ea numero bera duten eta. Azaldutako ezaugarriak 5.20 irudian ikus daitezke:



Irudia 5.20: Aurrekari posibleen datuak

Irudian lauki baten barruan ageri diren datuak anafora pronominalaren eta aurrekari posiblearen arteko loturazko ezaugarriak dira. Ezaugarri hauek, anaforaren ezaugarriak eta aurrekari posiblearen ezaugarriak kontutan hartuta kalkulatu behar ditu sistemak. Irudiko adibidean, beraien arteko distantzia izen-sintagma bateko dela ikusten da. Ildo beretik jarraituta, esaldi berean daudela eta numero ezberdinak (MG-NUMS) dituztela ere argi ikusten da.

Sistemak anafora pronominal baten aurrekari posibleen informazioa eskuratzen duenean arff formatuko testerako fitxategia sortzeko instantziak osatzen ditu anaforaren eta bere aurrekari posibleen informazioarekin. Lehenego aurrekari posiblearen informazioa doa instantzia batean, ondoren anaforaren informazioa eta bukatzeko aurrekari posible hori ea anafora pronominal horren aurrekaria den ala ez adierazten duen balioa. Kasu honetan ez dakigunez aurrekari posiblea benetan aurrekaria den ala ez, balio hori *?* ikurrarekin uzten da. Instantziak nola osatuta dauden egiaztatzeko ikusi hurrengo irudia:



Irudia 5.21: Instantziak

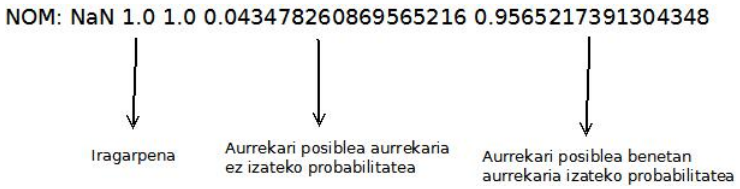
Instantzia guztiak osatuta ditugunean, sistemak testerako fitxategiaren izenburua eta ikasketarako erabiliko diren atributuak (*@ATTRIBUTE*) idatziko ditu testerako fitxategian. Bukatzeko, osatutako instatzia guztiak idatziko dira testerako fitxategiko datuen atalean (*@DATA*) baita instantzia bakoitza osatzen duten aurrekari posiblearen hitz zenbakia eta anaforaren hitz zenbakia ere instantzia bakoitzaren goikaldean 5.21 irudian ikusten den bezala. Anaforen eta aurrekari posibleen hitz zenbakiak idaztea erabaki dugu ikasketa automatikorako moduluari testerako fitxategia pasatzen zaionean erreferentziarik ez galtzeko.

5.4.3.2 Train fasea

Fase honetan sailkatzaile bat eraiki dugu *Ikasketarako corpusean* oinarrituta Weka aplikazioarekin. Sailkatzaile hori NaiveBayes algoritmoarekin sortu dugu hainbat proba egin ondoren emaitzarik onenak algoritmo horrekin sortutako sailkatzaileak eman dituelako.

5.4.3.3 Test fasea eta emaitzen interpretazioa

Fase honetan aurreko bi faseetan sortutako testerako fitxategiarekin eta sailkatzailearekin sistemak instantzia bakoitzeko emaitza bat bueltatzen du. Ondorengo irudian ikusten den bezala emaitza formatu berezi batean dago idatzita:



Irudia 5.22: Instantzia baten iragarpena

Emaitza horretatik interesatzen zaigun datu bakarra aurrekari posiblea benetan aurrekaria izateko probabilitatea da. Izan ere, sistemak bere lana burutzen duenean hizkuntzalariari anafora pronominal bakoitzeko 5 aurrekari probableenak aurkeztuko baitzaizkio grafikoki *MMAX2* tresnaren bidez.

Javaz idatzitako ikasketa automatikorako moduluak anafora pronominal bakoitzeko bere 5 aurrekari probableenak lortu dituenean, modulu nagusiari emaitzak bueltatzen dizkio ondorengo formatuan:

ANAid–PROB1id–PROB2id–PROB3id–PROB4id–PROB5id

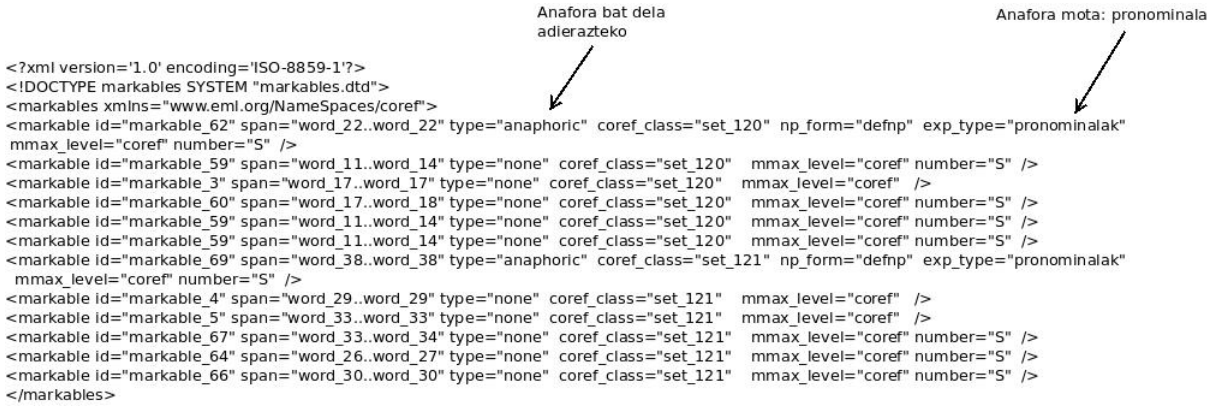
Adibidez:

24–12–11–10–8–7

Lerro horrek adierazten duena hurrengoa da: 24 hitz zenbakia duen anaforaren aurrekari probableena 12 hitz zenbakia duen izen-sintagma dela. Bigarren probableena 11 hitz zenbakia duena dela eta horrela bostgarren probableenera iritsi arte.

5.4.3.4 Emaitzen idazketa

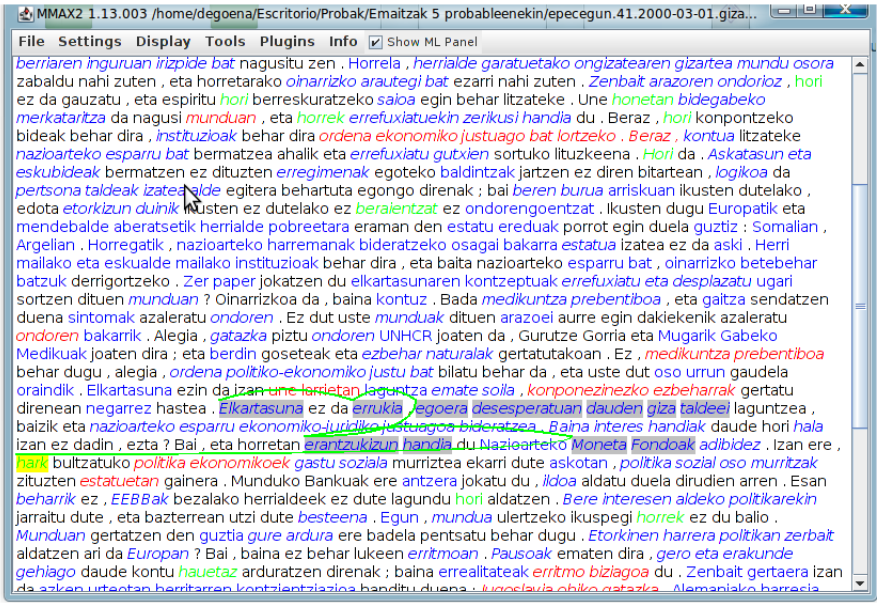
Anafora fidelen idazketa eta anafora pronominalen idazketa prozesuak berdintsuak direnez, ez da anafora pronominalen idazketa prozesu guztia azalduko. Bi prozesu horien artean dagoen desberdintasun bakarra idatzi behar den anafora mota da. *Fitxategiaren_izena_coref_level.xml* fitxategian anafora pronominal bat eta honen aurrekariak idatzi nahi ditugunean, argi adierazi behar da anafora pronominala dela idatziko dena eta bere aurrekari posibleak bere multzo (*SET*) berekoak direla ondorengo irudian argi ikusten den bezala.



Irudia 5.23: Anafora pronominalak coref_level fitxategian

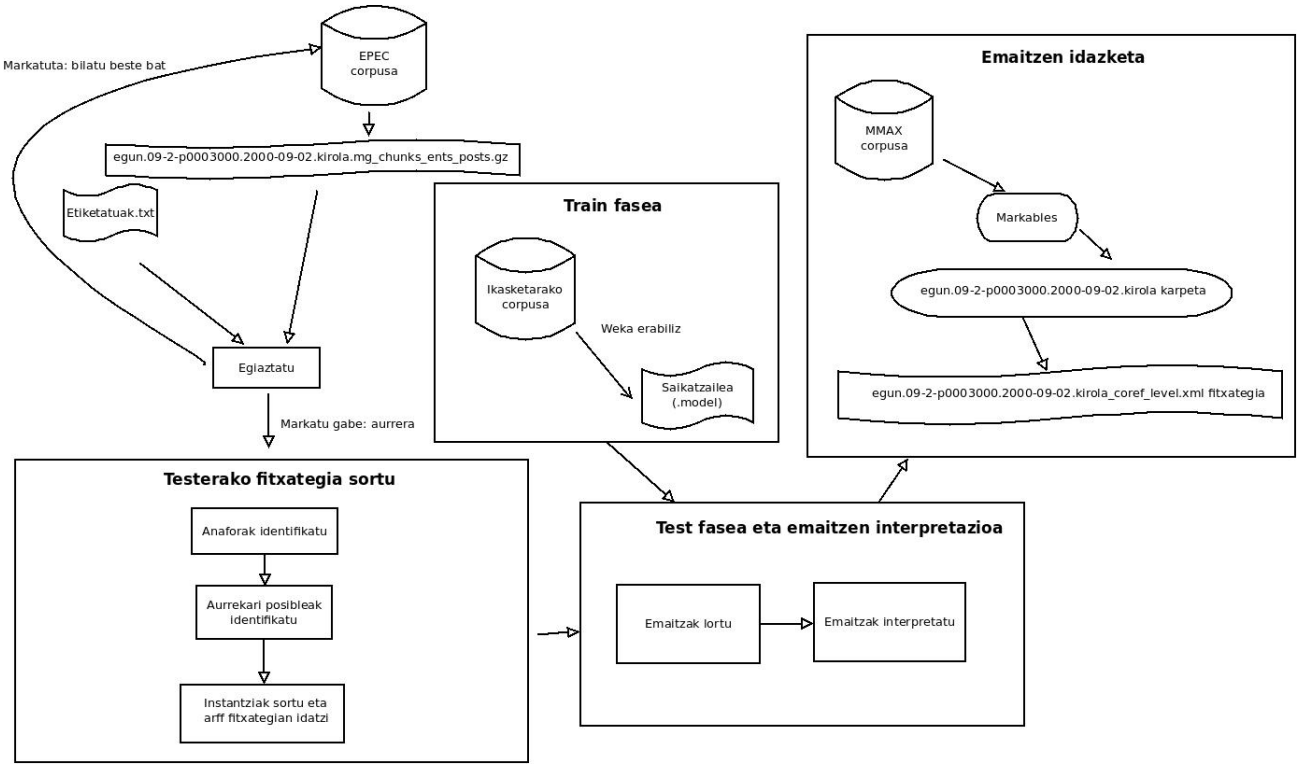
Aurreko irudian ageri dena azaltzearren, bi anafora pronominal daude markatuta, bi lerro ezberdinetan dagoelako *exp_type="pronominalak"* idatzita. Anafora pronominalak eta beraien bost aurrekari posibleak multzo berekoak direla adierazi behar da multzo zenbaki edo set zenbaki berdinarekin identifikatuz. Gure adibidean, lehenengo anafora pronominalak eta bere bost aurrekari posibleek 120 zenbakia dute multzoaren zenbaki bezala (*coref_class="set_120"*). Bigarren anaforak eta bere aurrekari posibleek, berriz, 121 zenbakia (*coref_class="set_121"*).

Sistemak idazketa guztiak egiten dituenean bukatzen da anafora pronominalak ebazten dituen prozesua. Era honetara, hizkuntzalariak *MMAX2* anotazio tresnarekin markatu berri dugun fitxategi bat irekitzen badu, anafora pronominalak eta bakoitzaren 5 aurrekari probableenak grafikoki irudikatuta ikusiko ditu pantailan 5.24 irudian azaltzen den modura.



Irudia 5.24: Anafora pronominalak

Anafora pronominalak ebazten dituen sistemaren inplementazioa osatzen duten 4 atalak edo faseak azaldu diren arren, ezinbestekoa iruditzen zaigu sistema osoaren inplementazio arkitektura erakustea irakurleari. Era honetara, irakurleak aurreko puntuetan azaldu duguna errazago ulertuko du eta sistemaren ideia garbiagoa izango du.



Irudia 5.25: Anafora pronominalak ebazteko sistemaren arkitektura

5.25 irudian anafora pronominalak ebazteko azaldu ditugun 4 faseak ageri dira: lehenengo, testerako fitxategia sortzen da, ondoren train fasean sortutako sailkatzailearekin ikasketa automatikorako moduluak testerako fitxategiko emaitzak bueltatzen ditu. Bukatzeko, emaitzak interpretatu ondoren *MMAX* corpusean idazketak egiten dira.

Irudi horren azalpenarekin bukatzen da anafora pronominalak ebazteko sistemaren inplementazioa, hurrengo kapituluan sistemaren ebaluazioari helduko diogu zer nolako emaitzak lortu ditugun azalduz besteak beste.

Kapitulua 6

Ebaluazioa

Ezin izan dugu sistema osoaren ebaluazioa egin gure sistemak bueltatzen dizkigun anafora fidelak ongi dauden ala ez erabakitzeko zailtasunagaitik eta anafora mota hori ebaluatzeko metodo justu bat aurkitzeko zailtasunagaitik. Izan ere, anafora fidel bakoitzeko izen-sintagmekin kateak osatzen dira eta ez dugu jakin kate horiek nola ebaluatu (kate osoa, bikoteka...). Horrenbestez, anafora pronominalak ebazten dituen sistema ebaluatu dugu eta fidelen ebaluazio zorrotza hizkuntzalarien esku utzi dugu. Hala ere, anafora fidelak ebazten dituen sistemaren azaleko ebaluazio bat egitea lortu dugu %70 inguruko doitasuna lortu duelarik.

6.1 Anafora pronominalak ebazteko sistemaren ebaluazioa

Sistema hau ebaluatzeko *EPEC* corpusetik ausaz hartutako 22 fitxategitan banatutako 130 anafora pronominal erabili ditugu. Probak 3 ezaugarri ezberdinen arabera burutu ditugu: Erabilitako sailkatzailea, anaforaren eta bere aurrekari posibleen arteko izen-sintagma distantzia, eta aurkeztuko diren aurrekari posibleen kopurua.

Probak egiteko erabili ditugun sailkatzaileak 3 izan dira: *NaiveBayes*, *VFI* eta *RandomForest*. Anaforaren eta bere aurrekari posibleen arteko izen-sintagma distantzia, ostera, 5 eta 8 bitartekoa izan da. Azkenik, aurkeztu beharreko aurrekari posibleei dagokienez, 3 aurrekari probableenak aurkeztuta eta 5 aurrekari probableenak aurkeztuta egin dira probak.

Bukatzeko, emaitzak lortzeko erabili dugun formula ondorengoa da:

Aurrekaria probableenen artean aurkitzen den aldi kopurua / Anafora kopuru totala

6.1.1 Anafora pronominalak ebazteko sistemaren emaitzak

Lortutako emaitzak erabilitako sailkatzailearen arabera aurkeztuko ditugu, hots, *NaiveBayes* sailkatzailearekin lortutakoak lehenengo, *VFI* sailkatzailearekin lortutakoak ondoren eta *RandomForest* sailkatzailearekin lortutakoak azkenik. Hainbat distantzia (3-10) eta aurrekari posibleekin (1-5) probak egin ditugun arren, ondorengo tauletan emaitzarik onenak eman dituztenak soilik aurkeztuko dira.

6.1.1.1 NaiveBayes sailkatzailearekin lortutako emaitzak

	3 prob	5 prob
5 dist	%66.15	%73.8
6 dist	%66.15	%73.8
7 dist	%63.8	%74.61
8 dist	%61.5	**%76.9**

Emaitza hauek aztertzen baditugu, konturatuko gara 5 aurrekari probableenak aurkeztuta lortzen diren emaitzak hobeak direla 3 probableenak aurkeztutakoekin lortutakoak baino. Egoera hori guztiz logikoa da, aurrekaria 5 probableenen artean egotea probableago delako 3 probableenen artean egotea baino.

Bestalde, emaitzarik onena 8 izen-sintagmako distantziarekin eta 5 probableenekin lortu da, zehazki %76.9ko doitasuna lortu da baldintza horietan.

6.1.1.2 VFI sailkatzailearekin lortutako emaitzak

	3 prob	5 prob
5 dist	%65.3	**%73.8**
6 dist	%66.53	**%73.8**
7 dist	%59.2	%72.31
8 dist	%57.69	%71.5

Sailkatzaile honekin lortutako emaitzarik onena 5 izen-sintagma distantziarekin eta 5 aurrekari probableenekin lortu da (%73.8), emaitza bera lortu da 6 izen-sintagma distantziarekin eta 5 aurrekari probableenekin ere. Sailkatzaile honekin lortutako emaitzek ez dute NaiveBayes sailkatzailearekin lortutakoek duten joera. Izan ere, bi sailkatzaileak alderatuta, bakoitzak lortu dituen emaitzarik hoberena eta kaxkarrena besteak emaitza horiek erdietsi dituen distantzia eta aurrekari probable kopuru desberdinekin eskuratu ditu.

6.1.1.3 RandomForest sailkatzailearekin lortutako emaitzak

	3 prob	5 prob
5 dist	%64.6	%73.8
6 dist	%63.07	%73.8
7 dist	%57.69	%73.07
8 dist	%53.84	**%75.3**

RandonForest sailkatzailearekin lortutako emaitzarik onena NaiveBayes sailkatzailearekin lortutako emaitzarik hoberena baino okerragoa den arren, nahiko emaitza ona da. %75.3ko doitasun hori 8 izen-sintagmako distantziarekin eta 5 aurrekari probableenak aurkeztuz lortu dugu, NaiveBayes sailkatzailearekin emaitzarik hoberena lortu dugun baldintza berdinetan. Ondorioz, hizkuntzalariei aurkeztuko zaien azken sistemak baldintza horiek ditu inplementatuta. Sailkatzaileari buruz, aldiz, emaitzarik onena lortu duena inplementatu dugu, NaiveBayes hain zuzen ere.

Kapitulua 7

Aurkitutako arazoak

Kapitulu hau bi atal nagusitan banatu dugu. Lehenengo atalean anafora fidelak ebazten dituen sistema garatzean eta ebaluatzean aurkitu ditugun arazoak kontatzen ditugu. Bigarren atalean, berriz, gauza bera egin da, baina eredu gisa anafora pronominalak ebazten dituen sistema hartuta.

7.1 Anafora fidelak ebazteko sistemarekin izandako arazoak

Anafora mota hau tratatzen duen sistemarekin izan ditugun arazoak ere bi zatitan banatuko ditugu. Batetik, sistema garatzean aurkitu ditugun zailtasunak azalduko ditugu. Bestetik, zeintzuk arazoren eraginagaitik ez dugun sistema hau ebaluatu azalduko dugu.

7.1.1 Garapenean aurkitutako arazoak

Urrats honetan izandako arazoak ondorengo zerrendan daude laburbilduta:

- *EPEC* corpusa ez dago guztiz ondo etiketatuta
  - Izen-sintagma askoren hasiera (SIH) eta bukaera (SIB) ez dago ongi markatuta: Askotan izen-sintagma baten hasieraren (SIH) ondoren beste izen-sintagma hasiera (SIH) bat dago markatuta izen-sintagma bukaera (SIB) baino lehen.
  - Hitz batek askotan analisi bat baino gehiago du. Zein da zuzena? Guk beti lehenengoa hartu dugu, baina lehenengoa ez da beti zuzena.
- *MMAX* corpusean adierazitako izen-sintagmak batzuetan ez dira *EPEC* corpusean adierazitako berdinak.

7.1.2 Ebaluazioan aurkitutako arazoak

Ebaluatzen saiatu garenean izandako arazoak, sistemak aurkeztu dizkigun emaitzak zuzenak diren ala ez interpretatzeko zailtasunetatik datoz gehienbat. Askotan zalantzak izan ditugu sistemak aurkeztu digun anafora fidelen kate bat benetan anafora fidel bat den ala ez. Guk izan ditugun dudak irakurleari aurkezteko asmoz, zerrendatxo bat osatu dugu bidean aurkitutako zenbait kasurekin:

- Denborazko adberbioak: atzo - atzokoan
- Lekuzko adberbioak: alde batetik - beste aldetik
- Urteak: 50 urtez - 1973. urtean
- Zenbakiak: 6-1 - 6-1 (teniseko emaitzak dira)
- Elipsiak: taldekideak - taldekideenak

Ebaluazioa burutzeko garaian izan dugun beste arazo larri bat nola ebaluatu ez jakitea da. Zein formula erabili emaitzak era fidagarri eta justu batean lortzeko?

7.2 Anafora pronominalak ebazteko sistemarekin izandako arazoak

Anafora mota hau tratatzen duen sistemarekin izan ditugun arazoak ere bi zatitan banatuko ditugu anafora fidelekin egin dugun bezala. Batetik, sistema garatzean aurkitu ditugun zailtasunak azalduko ditugu. Bestetik, sistema ebaluatzerako garaian aurkitu ditugun zailtasunak zehaztuko ditugu.

7.2.1 Garapenean aurkitutako arazoak

Urrats honetan izandako arazoak ondorengo zerrendan daude laburbilduta:

- Ikasketa automatikorako moduluak MultilayerPerceptron sailkatzailearekin bueltatzen dizkigun emaitzak ez zaizkigu egokiak iruditu gure helburua betetzeko nahiz eta sailkatzaile honekin zenbait lanetan (Zelaia et al., 2010) Weka erabiliz oso emaitza onak lortu diren. Ondorioz, sailkatzaile hau ez erabiltzea erabaki dugu.
- Ikasketa automatikorako modulua derrigorrez Javaz inplementatu behar izan dugu, ikasketa automatikoa erabiltzea ahalbidetzen duten liburutegiak lengoaia horretan idatzita daudelako. Modulu nagusia Perl lengoaian idatzi denez, bi moduluen arteko koordinazioa lortu behar izan dugu.
- EPEC corpusa ez dago guztiz ondo etiketatuta anafora fidelen atalean azaldu dugun bezala.

7.2.2 Ebaluazioan aurkitutako arazoak

Sistema hau ebaluatzerako garaian izan dugun arazo bat ebaluaziorako erabili ditugun 130 anafora pronominal horiek aurkitzea izan da *MMAX* corpuseko direktorioen artean. Izan ere, direktorio askotan anafora pronominalik ez baitzegoen. Hala ere, izan ditugun oztoporik handienak gure sisteman zerbait aldatzen izan dugun bakoitzean berriro anafora guzti horien ebaluazioa egiteak daraman denbora eta lana izan dira.

Kapitulua 8

Ondorioak eta etorkizunerako lanak

Esku artean dugun proiektua bukatu ondoren zenbait ondorio ateratzeko garaia da. Hasteko, esan beharra dugu euskararako anaforak automatikoki etiketatzen dituen sistema bat oinarri bezala ez edukitzeak lan asko egitera eraman gaituela, dena edo ia dena berria izan delako guretzat. Hala ere, nahiz eta guk nahi bezain emaitza onak lortu ez ditugun, uste dugu egindako lana aprobetxagarria eta baliagarria dela.

Proiektuarekin martxan jarri ginenean finkatu genituen helburuak bete ditugu: hizkuntzalariei etiketatze lana erraztuko dien sistema bat garatu dugu anafora fidelak eta pronominalak automatikoki etiketatuz. Fidelen kasuan sistemak duen asmatze-tasa zehatza kalkulatu ez dugun arren, egindako ebaluaketa partzialarekin sistemak %70 inguruko doitasuna duela esan daiteke. Pronominalen kasuan, ostera, emaitzarik hoberena %76.9koa da.

Ondorioekin bukatzeko, seguru gaude erabili dugun corpusak etiketatze errorerik izango ez balu eta analisi zuzen bakarra izango balu hitz bakoitzerako emaitza hobeak lortuko genituela. Horregaitik probak errepikatzeko intentzioa daukagu desanbiguatuta dagoen corpus batekin.

Etorkizunari begira, euskararako anafora mota guztiak ebazten dituen sistema bat eraikitzea oso interesgarria izango litzateke, baina jakitun gara horrelako erronka batek sekulako zailtasunak dituela.

Bukatzeko, anafora fidelak ebazten dituen sistemaren ebaluazioa IXA taldeko hizkuntzalarien esku utziko da informatikarientzako honek suposatzen duen zailtasunagaitik.

Bibliografia

[1] Aduriz, I., Ceberio, K., Díaz de Ilarraza, A.: *Pronominal Anaphora in Basque: computational point of view and the development of a corpus* GOGOA, V-1: 91-116 (2005)

[2] Arregi, O., Ceberio, K., Díaz de Ilarraza, A., Goenaga, I., Sierra, B., Zelaia, A.: *Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque* SEPLN XXVI. Vol. 45, pp. 291-294. ISSN: 1135-5948 (2010)

[3] Bach, C., Saurí, R., Vivaldi, J. and Cabré, M.T.: *El corpus de l'IULA*: descripció. Barcelona: Universitat Pompeu Fabra, Institut Universitari de Lingüística Aplicada (1997)

[4] Biber, D., Conrad, S. and Reppen, R.: *Corpus linguistics. Investigating language structure and use*. Cambridge: Cambridge University Press (1998)

[5] Jimeno, A.: *Anafora pronominala identifikatzeko sistema baten garapena* Karrera bukaerako proeiktua (2010)

[6] Kilgarriff, A. and Grefenstette, G.: *Introduction to the Special Issue on Web as Corpus*. Computational Linguistics, 29(3): 1-15 (2003)

[7] Moosavi, N.S., Ghassem-Sani, G.: *Using Machine Learning Approaches for Persian Pronoun Resolution*. In: Workshop on Corpus-Based Approaches to Coreference Resolution in Romance Languages. CBA 2008 (2008)

[8] Moosavi, N.S., Ghassem-Sani, G.: *A Ranking Approach to Persian Pronoun Resolution*. Advances in Computational Linguistics. Research in Computing Science 41, 169–180 (2009)

[9] Ng, V., Cardie, C.: *Improving Machine Learning Approach to Coreference Resolution*. In: Proceedings of the ACL, pp. 104–111 (2002)

[10] Nguy, Zabokrtsk´y: *Rule-based Approach to Pronominal Anaphora Resolution Method Using the Prague Dependency Treebank 2.0 Data*. In: Proceedings of DAARC 2007 (6th Discourse Anaphora and Anaphor Resolution Colloquium) (2007)

[11] Recasens, M.: *Towards Coreference Resolution for Catalan and Spanish* Tesia (2008)

[12] Soon, W.M., Ng, H.T., Lim, D.C.Y.: *A Machine Learning Approach to Coreference Resolution of Noun Phrases.* Computational Linguistics 27(4), 521–544 (2001)

[13] Versley, Y.: *A Constraint-based Approach to Noum Phrase Coreference Resolution in German Newspaper Text.* In: Konferenz zur Verarbeitung Natrlicher Sprache KONVENS (2006)

[14] Versley, Y., Ponzetto, S., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yan, X., Moschitti, A.: *BART: A Modular Toolkit for Coreference Resolution* Proceedings of the ACL-08: HLT Demo Session (Companion Volume), pages 9–12 (2008)

[15] Zelaia, A., Sierra, B., Arregi, O., Ceberio, K., Díaz de Ilarraza, A., Goenaga, I.: *A Combination of Classifiers for the Pronominal Anaphora Resolution in Basque* CIARP 2010. LNCS 6419, pp. 253–260 (2010)

A Eranskina

MMAX2 direktorio sistema

Direktorio sistema berezi hori ondorengo karpetek eta fitxategiek osatzen dute:

- **Erroa:** Fitxategiaren izena duen karpeta bat da (aurreko kasuan *fitx1.txt*).

Erroaren barruan goiko mailan 5 karpeta eta fitxategi 2 daude:

- **common_paths.xml fitxategia:** Fitxategi honetan daude MMAX2 tresnak informazioa pantailaratzeko begiratu behar dituen fitxategi eta karpeten helbideak.

- **Fitxategiaren izena.mmax fitxategia** (aurreko kasuan *fitx1.txt.mmax*)**:** Fitxategi hau da MMAX2 tresnarekin ireki behar dena testuaren gainean anotazioak egiteko. Bertan pantailan erakutsi behar diren hitzak zein fitxategitan dauden dago adierazita 8.1 irudian ikusten den bezala.

```
<?xml version='1.0'?>  <mmax_project>
<!--<sentences>002_htc_text.xml</sentences>-->
<turns></turns> <words>epecegun.36.2000-11-04.kirola13.txt.words.mmx.xml</words>
<gestures></gestures> <keyactions></keyactions> </mmax_project>
```

Irudia 8.1: MMAX fitxategia

- **Schemes karpeta:** Karpeta honen barruan jorratuko ez ditugun konfiguraziorako zenbait fitxategi daude. Adibidez, aplikazioak konfiguratuta ez duen anafora mota berri bat markatzeko karpeta honetako fitxategietan egin beharko genituzke aldaketak.

- **Styles karpeta:** Hau ere, aurrekoaren antzera, konfiguraziorako fitxategiak dituen karpeta bat da.

- **Markables karpeta:** Karpeta honen barnean korreferentzi mailako nahiz esaldi mailako loturak egiteko fitxategiak daude.

  - **Fitxategiaren izena_coref_level.xml fitxategia:** Guretzat oso garrantzitsua da fitxategi hau, bertan egin behar izan ditugulako anafora fidel nahiz pronominalen arteko loturak. Hemen zehazten da zein hitz edo izen-sintagma dagoen zeinekin lotuta **??** irudian ikusten den bezala (aurreko kasuan *fitx1.txt_coref_level.xml* izena izango luke fitxategiak).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE markables SYSTEM "markables.dtd">
<markables xmlns="www.eml.org/NameSpaces/coref">
<markable id="markable_62" span="word_22..word_22" type="anaphoric" coref_class="set_120" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_59" span="word_11..word_14" type="none" coref_class="set_120" mmax_level="coref" number="S" />
<markable id="markable_3" span="word_17..word_17" type="none" coref_class="set_120" mmax_level="coref" />
<markable id="markable_60" span="word_17..word_18" type="none" coref_class="set_120" mmax_level="coref" number="S" />
<markable id="markable_69" span="word_38..word_38" type="anaphoric" coref_class="set_121" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_4" span="word_29..word_29" type="none" coref_class="set_121" mmax_level="coref" />
<markable id="markable_5" span="word_33..word_33" type="none" coref_class="set_121" mmax_level="coref" />
<markable id="markable_67" span="word_33..word_34" type="none" coref_class="set_121" mmax_level="coref" number="S" />
<markable id="markable_82" span="word_83..word_83" type="anaphoric" coref_class="set_122" np_form="defnp"
 exp_type="pronominalak" mmax_level="coref" number="S" />
<markable id="markable_78" span="word_73..word_74" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
<markable id="markable_78" span="word_73..word_74" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
<markable id="markable_80" span="word_77..word_77" type="none" coref_class="set_122" np_form="defnp" mmax_level="coref" number="S" />
</markables>
```

Irudia 8.2: fitxizena_coref_level.xml fitxategia

  - **Fitxategiaren izena_sentence_level.xml fitxategia:** Esaldi mailako loturak zehazteko erabiltzen da fitxategi hau (aurreko kasuan *fitx.txt_sentence_level.xml* izena izango luke fitxategiak).
  - **markables.dtd fitxategia:** Fitxategi honek aurreko biek izen behar duten formatua adierazten du.

- **Customizations karpeta:** Honen barruan korreferentziazko loturetan eta bestelakoetan erabili nahi diren koloreak etab aldatzeko 3 fitxategi daude.
  - **chunk_customization.xml fitxategia:** Chunk mailako estilo aldaketak egiteko erabiltzen da fitxategi hau.
  - **coref_customization.xml fitxategia:** Korreferentzi mailako estilo aldaketak egiteko erabiltzen da fitxategi hau. Adibidez, anafora pronominalak pantailan kolore berdez markatzeko.
  - **sentence_customization.xml fitxategia:** Esaldi mailako estilo aldaketak egiteko erabiltzen da fitxategi hau. Adibidez, anafora fidelak lotzen dituen geziaren kolorea aldatzeko.

- **Basedata karpeta:** Karpeta honen barruan beste fitxategi 2 daude.
  - **Fitxategiaren izena.words.mmx.xml fitxategia:** Fitxategi honetan markatu nahi den testua dago hitzez hitz xml formatuan 8.3 irudian ikusten den bezala (aurreko kasuan *fitx1.txt.words.mmx.xml* izena izango luke fitxategiak).

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE words SYSTEM "words.dtd">
<words>
<word id="word_1">Xakea</word>
<word id="word_2">.</word>
<word id="word_3">Egoera</word>
<word id="word_4">berrira</word>
<word id="word_5">nola</word>
<word id="word_6">moldatuko</word>
<word id="word_7">den</word>
<word id="word_8">ikusi</word>
<word id="word_9">beharko</word>
<word id="word_10">da</word>
</words>
```

Irudia 8.3: izena.words.mmx.xml fitxategia

– **words.dtd fitxategia:** Fitxategi honetan aurreko fitxategiak izan behar duen xml formatua zehazten da.