eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Proiektua

Itzulpen-sistema hibridoen eraikuntza EBMT bidezko itzulpen partzialak erabiliz

---

Egilea

Mikel Artetxe

informatika fakultatea
facultad de informática

2014

Eskerrak

Eskerrik asko Kepa eskerren atala ez egiteko zure oniritzia emateagatik. Eta eskerrik asko, bide batez, proiektuko zuzendari aparta izateagatik. Eskerrak, nola ez, Gorkari ere zure gidaritza paregabeagatik eta hainbeste arazorekin arazorik gabe laguntzeagatik. Eta Norari, Iñakiri eta beste guztiei. Baina sekula ez naiz koipea emateko lerro hauen zalea izan eta, falta zaretenon barkamenaz, ez noa gehiago luzatzera. Eskerrik asko Kepa eskerren atala ez egiteko zure oniritzia emateagatik.

Laburpena

Proiektu honetan EBMT tekniken bidez itzulpen partzialak sortzen dituen aurreprozesu batean oinarritutako itzulpen automatikorako hibridazio-mekanismo bat garatu da, entitateen eta esaldia baino txikiagoak diren unitate sintaktikoen bidezko orokortzea darabilena. Sistema oso arina eta eskalagarria izan dadin diseinatua izan da, eta inplementazio modular, hedagarri eta eraginkor bat eskaini zaio, baliabide eta tresna ugarirekin integratuz. Egindako esperimentuetan oso emaitza positiboak eskuratu dira, proposatutako sistemak abiapuntukoarekiko hobekuntza nabarmenak ekar ditzakeela erakusten dutenak.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera

Kapitulu honetan memorian zehar landuko denaren sarrera bat egiten da. Honela, lehen azpiatal batean proiektua zertan datzan deskribatzen da, nola antolatu den azaldu hurrengoan eta, amaitzeko, azken azpiatal batean bere irismena zehazten da, helburu, betekizun, emangarri eta mugarri guztiak definituz. Proiektuaren kudeaketari buruzko xehetasunetarako, A eranskinean ematen den proiektuko plangintzara jotzea gomendatzen da.

1.1 Proiektuaren deskribapena

Itzulpen-memoriak giza itzulpenen datu-baseak dira. Segmentu deritzaien oinarrizko unitateekin egiten dute lan, oro har esaldiak izan ohi direnak, hauetako bakoitzari dagozkion jatorrizko testua eta testu itzulia jasoz etorkizunean itzulpen hauek berrerabiltze aldera. Itzulpen-memoriak konputagailuz lagunduriko itzulpen-tresnekin batera erabiltzen dira nagusiki, jada itzuliak izan diren segmentuak berriro itzultzeko lana aurrez dadin. Proiektu honen jomuga, itzulpen-memoriez baliatuz itzultzaile automatikoen portaera hobetzea da.

Itzulpen automatikoa hizkuntzalaritza konputazionala delakoaren aplikazio ezagun eta garrantzitsuenetariko bat da. Konputagailu bidez hizkuntza batetik beste baterako itzulpena automatizatzea du helburu, herri ezberdinen arteko hesiak gainditzeko ametsa egi bihurtzen lagun dezakeena geroz eta globalizatuagoa den mundu honetan. 20. gizaldiko bigarren zatian kokatzen dira bere hastapenak eta, hasiera batean urte gutxiren buruan erabat ebatzi ahalko zela pentsatzen bazen ere, erronka uste baino askoz handiagoa izaten ari

da. Nolanahi ere, aurrerapauso handiak eman dira azkenaldian eta, eskuratutako emaitzak perfekziotik oso urrun badaude ere, eginkizun batzuetarako erabilgarriak diren hainbat sistema ditugu eskuartean.

Itzulpen-memorien erabilera ez da berria itzulpengintza automatikoaren arloan. Alta, itzulpen automatikorako metodoak bi multzo nagusitan banatu ohi dira: erregeletan oinarrituak, zeinak jatorri eta xede hizkuntzen gaineko ezagutza linguistikoan funtsatzen baitira, eta corpusetan oinarrituak, zeinak jada eginiko itzulpenei atxikiriko ezagutza enpirikoan funtsatzen baitira, horretarako abiapuntutzat itzulpen-memorien moduko corpus paraleloak erabiliz hain zuzen ere. Azken hauetan, halaber, bi hurbilpen nagusi jarraitu izan dira: corpus paraleloen azterketatik erauziriko eredu estatistikoetan oinarritzen diren metodo estatistikoak batetik, eta jada eginiko itzulpenak berrerabiltzen dituzten adibideetan oinarrituak bestetik. Honela, itzuli beharreko testuaren eta corpusaren arteko bat-etortze maila altua denean eta, honen baitan, corpuseko itzulpen hauek modu zentzudun batean berrerabiltzeko aukera dagoenean, adibideetan oinarrituriko sistemek oso emaitza onak ematen dituzte, jatorrizko giza itzulpenen doitasun eta naturaltasunari eusteko gai baitira gainerako hurbilpenen arazo nagusiari aurre eginez. Bat-etortze maila txikiagoa denean eta, honenbestez, corpuseko itzulpenak behar bezala berrerabiltzea posible ez denean, baina, emaitzak oso eskasak izaten dira halabeharrez, erregeletan nahiz metodo estatistikoetan oinarrituriko sistemen estaldura eta malgutasuna askozaz handiagoa delarik zentzu honetan.

Gauzak honela, proiektu hau honako planteamendutik abiatzen da: itzulpen-memoriara joz modu egokian itzul daitezken testu zatiak itzuli lehendabizi eta, ondoren, era honetara itzuli ezin izan den testua itzultzaile automatiko nagusiaren bidez itzuli. Beste hitz batzuetan, itzulpen automatikoko sistema hibrido bat proposatzen da, adibideetan oinarrituriko itzulpen automatikoa aurreprozesu gisara txertatzen duena itzultzaile automatiko nagusiak, izan estatistikoa ala izan adibideetan oinarritua, osatuko dituen itzulpen partzialak sortuz.

Bistakoa denez, itzulpen-memoriako segmentuak bere horretan baino berrerabiliko ez lituzkeen oinarrizko hurbilpenak zehaztasun-maila oso altuko itzulpen partzialak sortuko lituzke baina, horrekin batera, bat-etortzeen indizea hain izango litzateke txikia ezen hobekuntza honen inpaktu erreala oso eskasa izango bailitzateke. Hori dela eta, jarraituriko planteamendua arrakastatsua izan dadin sorturiko itzulpen partzialen kopuruaren eta kalitatearen arteko oreka bat bilatu behar da. Gauzak honela, proiektu honen eginkizun nagusia aipatu berri den oinarrizko hurbilpena abiapuntutzat hartuz itzulpen-memoriako sarrerak orokortzea izango da itzulpen partzial gehiago sor daitezen hauen zehaztasunari

ahal den neurrian eutsiz. Horretarako, honako bi bideak aztertuko dira:

- **Itzulpen-memoriako sarreren orokortzea entitateen bidez.** Honen baitan, entitate deritzaien izaera bereziko hitzak ala hitz-multzoak ezagutuko lirateke itzulpenetan (izen bereziak, zenbakiak...), halako beste entitateez ordezkatuak izan litezkeenak modu askean.
- **Itzulpen-memoriako sarreren orokortzea segmentua baino txikiagoak diren unitate sintaktikoen bidez.** Honi esker, esaldien osagaien itzulpenak berrerabiltzeko aukera egongo litzateke.

Proiektuaren helburua hori eskuratzeko tekniken inguruan ikertu eta haien portaera esperimentalki neurtzea da, horren arabera diseinaturiko metodoarentzat inplementazio funtzional eta eraginkor bat emanez.

1.2 Proiektuaren antolaketa orokorra

Proiektu hau hizkuntzaren prozesamenduaren inguruan aritzen den IXA taldearen barnean burutu da. Proiektua bera hutsetik abiatzen bazen ere, bada, ikerketa-talde horrek itzulpen automatikoaren arloan egiten duen lanaren testuinguruan ulertu behar da.

Bere antolaketa, baina, ez da batere ohikoa izan gradu amaierako proiektu batentzat. Izan ere, 2012-2013 ikasturtean hasi zen 400 orduko lan-poltsa baten harira, eta 2013-2014 ikasturtean jarraitu 450 orduko lankidetza-beka baten bidez. Bai lan-poltsan bai eta lankidetza-bekan, baina, proiektu honetaz gain bestelako eginkizunetan ere aritu da, malgutasun handiz jokatuz zentzu horretan.

1.3 Irismena

Atal honetan proiektuaren irismena zehazten da, eta azpiatalez azpiatal bere helburua eta betekizunak, emangarriak eta mugarriak biltzen ditu.

1.3.1 Helburua eta betekizunak

Proiektuaren helburua EBMT tekniketan oinarrituz itzulpen partzialak sortzen dituen aurreprozesu bat darabilten itzulpen-sistema hibridoen inguruan ikertzea da. Bide horretan,

ondoko gutxieneko betekizunak zehaztu dira, eurak betetzea ezinbestekoa izango delarik proiektuaren arrakastarako:

- Itzulpen automatikoaren egungo egoera eta hurbilpenak ezagutzea, bai eta horretarako erabiltzen diren hizkuntzalaritza konputazionaleko oinarrizko baliabide, tresna eta kontzeptuak ere.
- EBMT tekniketan oinarrituz itzulpen partzialak sortzen dituen aurreprozesu baten bidez itzulpen-sistema hibridoak sortzeko aukera aztertzea, eta hurbilpen hori jarraitzen duen oinarrizko sistema bat diseinatzea.
- Gutxien-gutxienez itzulpen-memoriako sarrerak bere horretan berrerabiltzeko gai den aurreprozesatzaile bat garatu eta itzultzaile automatiko batekin txertatzea.
- Ondorengo bi orokortze-mekanismoak aztertzea, ahal dela garatu beharreko sisteman barneratuz:
  - Itzulpen-memoriako sarreren orokortzea entitateen bidez.
  - Itzulpen-memoriako sarreren orokortzea segmentua baino txikiagoak diren unitate sintaktikoen bidez.
- Garaturiko sistema euskarazko tresnekin bateragarria izatea, gutxien-gutxienez gaztelania-euskara edo ingelesa-euskara bikoteetako itzultzaileren batekin txertatuz.
- Garaturiko sistemaren portaera ebaluatu eta aldagai ezberdinen eragina neurtzeko esperimentu bat diseinatu eta aurrera eramatea, bertan eskuratutako emaitzak aztertu eta interpretatuz.
- Proiektuan zehar egindako lana eta ateratako ondorioak jasotzen dituen dokumentazioa sortzea.

1.3.2 Emangarriak

Proiekturako ondoko bost emangarriak finkatu dira, azken biak Gradu Amaierako Proiektu batean berezkoak eta derrigorrezkoak, eta gainerakoak proiektu honen eginkizun zehatzei lotuak:

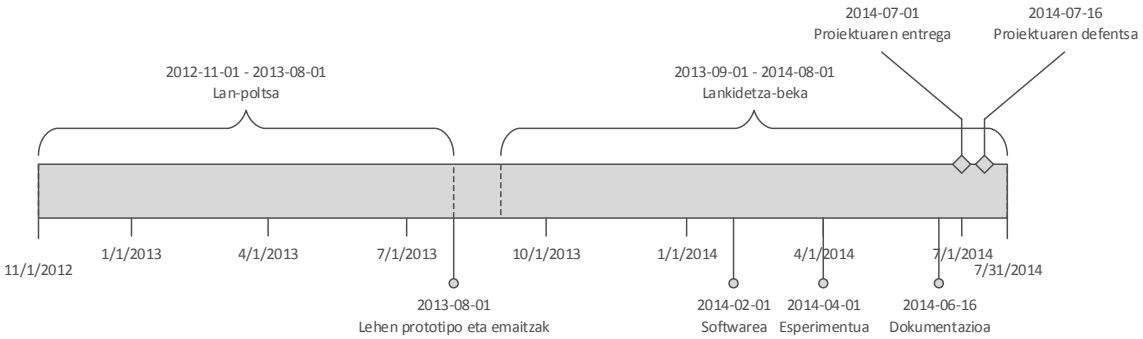
- **Softwarea.** Emangarri honek proiektuaren ardatza den itzulpen-sistema bera hartzen du beregain. Exekutagarriaz gain, bere iturburu-kodea ere emangarriaren parte izango da.
- **Iturburu-kodearen API dokumentazioa.** Emangarri hau garaturiko softwarea aldatu, egokitu, hedatu edota liburutegi modura erabili nahi duten garatzaileek erreferentzia bezala erabiltzeko dokumentazioa izango da, bere APIa zehaztasunez deskribatuko duena.
- **Erabiltzailearen gida.** Emangarri honek garaturiko softwarea erabiltzeko jarraibideak bilduko ditu. Ahal bezain trinko eta argia izango da, erabiltzailearekiko elkarrekintzan soilik zentratuz (agindu posibleak, parametroak...) eta gainerako alderdiak albo batera utziz.
- **Memoria.** Emangarri hau proiektuan egindako lan guztia deskribatzen duen dokumentu bat izango da, eta proiektuaren beraren zehaztapena, problemaren analisia, soluzioaren diseinua, inplementazioaren xehetasunak, esperimentua eta bertan eskuratutako emaitzak, eta proiektuan zehar ateratako ondorioak bilduko ditu.
- **Aurkezpena.** Emangarri hau proiektuaren bukaerako defentsarako prestatu beharreko aurkezpenari dagokio, proiektuan egindako lanaren, eskuratutako emaitzen eta ateratako ondorioen sintesi bat izan beharko dena.

1.3.3 Mugarriak

Proiekturako mugarriak bi multzotan bereizi dira. Alde batetik, barne-mugarriak aurkitzen dira, proiektua bide onetik joan dadin erreferentzia modura erabiliko direnak baina derrigorrezko izaerarik ez dutenak. Eta, bestetik, kanpo-mugarriak daude, Gradu Amaierako Proiektuen egutegi ofizialak zehaztuak eta asmoa den deialdian aurkezteko derrigor bete beharrekoak. Horretaz gain, gogoan hartu behar da 1.2 atalean zehaztu bezala proiektua bi fasetan garatuko dela, bata 2012-2013 ikasturtean eta bestea 2013-2014 ikasturtean. 1.1 irudiko mugarri diagramak horiek guztiak jasotzen ditu, jarraian banan-banan azaltzen direnak.

Proiektuaren faseak

- **Lan-poltsa (2012/11/01 - 2013/08/01).** Lehen fase hau 2012-2013 ikasturtean burutuko da 400 orduko lan-poltsa baten bitartez, non proiektuaz gain beste eginkizun



1.1 Irudia: Mugarri-diagrama

batzuetan ere arituko baita.

- **Lankidetza-beka (2013/09/01 - 2014/08/01).** Bigarren fase hau 2013-2014 ikasturtean burutuko da, oraingoan 450 orduko lankidetza-beka baten bidez. Lan-poltsarekin bezala, proiektuaz gain beste eginkizun batzuetan ere lanean arituko da.

Barne-mugarriak

- **Lehen prototipo eta emaitzak (2013/08/01).** Lan-poltsaren amaierarako itzulpen-sistemaren oinarrizko prototipo bat izan beharko litzateke, bai eta bigarren fasean jarraitu beharreko bidea argitasunez zehazteko moduko lehen emaitza batzuk ere.
- **Softwarea (2014/02/01).** Data honetarako itzulpen-sistemaren garapenak bukatuta egon beharko luke.
- **Esperimentua (2014/04/01).** Egun honetarako esperimentuak amaituta beharko luke, bai eta bertan lorturiko emaitzen azterketa eta interpretazioak ere.
- **Dokumentazioa (2014/06/16).** Data honetarako dokumentazioa bukatuta egon beharko litzateke, bai iturburu-kodearen API dokumentazioa, bai erabiltzailearen gida, bai eta memoria ere.

Kanpo-mugarriak

- **Proiektuaren entrega (2014/07/01).** Egun honetan ADDI bitartez edo Fakultateko Idazkaritzan proiektua entregatu behar da.

- **Proiektuaren defentsa (2014/07/16-18).** Egun hauetako batean proiektua epaimahaiaren aurrean aurkeztu eta defendatu beharko da.

2. KAPITULUA

Baliabideak eta aurrekariak

Kapitulu honetan proiektu honen baitan erabiliriko hizkuntzaren prozesamenduaren arloko baliabide eta aurrekariak zehazten dira. Honela, atal banatan hizkuntzaren analisirako tresnak, baliabide linguistikoak, hitz-lerrokatzea eta itzulpen automatikoa lantzen dira, euretako bakoitzaren inguruko oinarrizko kontzeptuak azaldu eta proiektuan zehar baliatuko den software espezifikoa aurkeztuz. Horretarako erreferentzia nagusiak Jurafsky and Martin (2008), Manning and Schütze (1999) eta Aldezabal et al. (2005) izan dira.

2.1 Hizkuntzaren analisirako tresnak

Hizkuntzaren analisirako tresnek testu idatzietatik nolabaiteko informazio linguistikoa erauztea dute jomuga. Informazio honen izaera eta abstrakzioaren arabera, analisia maila ezberdinetan egin daiteke. Hauetako bakoitza modu automatizatuan burutu ahal izateko, hainbat aplikazio informatiko garatu izan dira hizkuntza ezberdinentzat, analizatzaile deritzen software-paketeen modulu gisara integratu ohi direnak.

Gauzak honela, 2.1.1 azpiataletik 2.1.5 azpiatalera bitartean maila ezberdinetan diharduten analisi-tresnak aurkezten dira testutik hurbilen aritzen direnetatik hasi eta abstrakzio maila handiena eskaintzen dutenetaraino, hurrenez hurren segmentazioa, analisi morfologikoa, etiketatzea, entitate-izenen ezagutzea eta analisi sintaktikoa jorratuz. Azkenik, 2.1.6 azpiatalean proiektuan zehar helburu honetarako baliaturiko softwarea aurkezten da.

2.1.1 Segmentazioa

Segmentazioa testu bat berau osatzen duten unitate adierazkorretan banatzean datza. Unitate hauen izaeraren arabera, segmentazioa maila ezberdinetan egin daiteke:

- **Hitz-segmentazio edo tokenizazioak** testua tokenetan, hots, testuinguru jakin batean unitate semantikotzat azter daitezen multzokatzen diren karaktere-sekuentzietan, banatzen du. Puntuazio-ikurrak, kontrakzioak nahiz *multiword* bezalako fenomenoak medio, honetarako irizpide linguistiko argi bat finkatzea ez da erraza, baina token eta hitz kontzeptuen arteko elkarrekikotasuna ia erabatekoa dela esan daiteke. Honela, latindar alfabetoa darabilten hizkuntzekin jarraituriko oinarrizko hurbilpena zuriuneen moduko karaktere banatzaileez baliatzean datza. Txinerak ala japonierak bezala halako kontzepturik ez duten idazketa sistemetan, berriz, tokenizazioaren arazoa nabarmenki konplexuagoa suertatzen da.
- **Esaldi-segmentazioak** testua esalditan banatzen du. Puntuazioa hartu ohi da horretarako abiapuntutzat, idazketa-sistemaren arabera beti ere. Hizkuntza gehienetan, baina, puntuazio-ikurrak anbiguoak suertatzen dira, laburduretan nahiz balio dezimaletan ager baitaitezke kasu. Honi aurre egiteko, erregela konplexuagoak definitu ohi dira, ikasketa automatikoko teknikekin uztartuz.

2.1.2 Analisi morfologikoa

Analisi morfologikoak token bakoitzaren azaleko forma hartu eta honek izan litzakeen forma lexikal posible guztiak itzultzen ditu. Azaleko forma testuan bere horretan ageri denari dagokio eta forma lexikala, berriz, tokenaren lema edo oinarrizko formak eta honi atxikiriko informazio gramatikalak (kategoria lexikala, generoa, numeroa...) osatzen dute.

Adibide modura, 2.1 taulak Apertium plataformak euskara-gaztelania bikoteko baliabideak erabiliz *hil artean bizi* esaerarentzat eginiko analisi morfologikoa erakusten du. Bertan ikus daitekeenez, eginiko analisiaren baitan *hil* azaleko formak bi forma lexikal ditu, izena nahiz aditza izan baitaiteke. *artean* forma, berriz, adberbio zein izen bat izan daiteke eta *bizi*, azkenik, izena, adjektiboa ala aditza.

Analisi morfologikoa egiteko hiztegi morfologikoak erabiltzen dira, hizkuntza jakin bateko azaleko forma eta forma lexikalen arteko baliokidetzak biltzen dituztenak. Honenbestez, analizatzaile morfologikoaren egiteko bakarra sarrerako azaleko forma bakoitzeko

hil	**artean**	**bizi**
hil<n>	artean<adv><gen>	bizi<n>
hil<vblex><inf>	arte<n>+a<det><art><sg>+an<post>	bizi<adj><izo>
		bizitu<vblex><inf>

2.1 Taula: Analisi morfologikoaren adibide bat Apertium erabiliz

hiztegian honi loturik ageri diren forma lexikal guztiak itzultzea baino ez da. Hau modu eraginkorrean egin dadin, hiztegi hauek transduktore finituetan konpilatu ohi dira, sarrerako testua denbora linealean prozesa daitekeelarik hauen bidez.

Alderantzizko prozesuari, hots, forma lexikal bat hartuta horri dagokion azaleko forma emateari, sorkuntza deritzo. Horretarako, analisian baliaturiko hiztegi morfologiko berberak erabili ohi dira aurkako noranzkoan, eraginkortasunaren alde transduktore finituetan konpilatuak direnak hemen ere. Analisian ez bezala, baina, sorkuntzan irteerako forma bakar bat izatea da zentzuzkoena. Ohikoena hiztegi morfologikoek baldintza hau betetzea da, forma lexikal bakoitzari azaleko forma bakar bat esleituz eta, halakorik betetzen ez duten kasu berezientzat, azaleko forma posibleen arteko nolabaiteko lehentasunak ezarri ohi dira. Adibide bat jartzearren, euskararen kasuan etorkizuneko adizki perifrastikoetako lehendabiziko osagai partizipioari *-ko*/*-go* nahiz *-en* atzizkia eransteko aukera onartzen da kasuan kasu eta, honela, *hilgo*, *hilko* eta *hilen* azaleko formek, hirurak zuzenak, forma lexikal berbera partekatzen dute, sorkuntza egiterakoan hiruren artetik bat aukeratu beharko litzatekeelarik nolabaiteko lehentasun baten arabera.

2.1.3 Etiketatzea edo desanbiguazio morfologikoa

Izenak berak dioenez, etiketatzearen edo desanbiguazio morfologikoaren egitekoa analisi morfologikoak azaleko forma bakoitzarentzat emaniko forma lexikal guztien artean testuinguru jakin horretan egokien suertatzen dena hautatzea da. Horretarako, token bakoitzak ingurukoekin duen harremana aztertzen da Markoven Eredu Ezkutuak bezalako baliabide estatistikoak edota murriztapen-gramatiken moduko erregelak erabiliz.

2.2 taulak ikusi berri den analisi morfologikoarentzat Apertiumek berak eginiko desanbiguazioa erakusten du. Ikus daitekeenez, testuinguru horretan *hil* formak aditz bezala, *artean* formak adberbio bezala eta *bizi* formak aditz bezala jokatzen duela ondorioztatzen du Apertiumek hitz hauen arteko erlazioa aintzakotzat hartuz.

hil	artean	bizi
hil<vblex><inf>	artean<adv><gen>	bizitu<vblex><inf>

2.2 Taula: Etiketatzearen adibide bat Apertium erabiliz

2.1.4 Entitate-izenen ezagutzea

Entitate-izenen ezagutzea kategoria aurrezarri jakin batzuei dagozkien izaera bereziko testu-elementu atomikoak identifikatzean datza. Honi hertsiki loturik, entitate-izenen sailkapen deritzona identifikaturiko entitate bakoitzari dagokion kategoria esleitzeaz arduratzen da. Bide horretan, aurrez definituriko entitateen biltegiak, hauetan ageri ez direnak identifikatzeko nolabaiteko erregelak, edota ikasketa automatikoko teknikak erabiltzen dira.

Proiektu honetan ondorengo kategoriekin egingo da lan:

- **Pertsona-izen bereziak**. Hala nola, *Barack Obama*, *Aita Santu*, *Joselontxo* edo *Lehendakari* modukoak izan daitezke.

- **Erakunde-izen bereziak**. Adibidez, *Nazioarteko Diru Funtsa*, *Euskal Irrati Telebista* ala *Athletic Club* bezalakoak.

- **Leku-izen bereziak**. *Euskal Herria*, *Zarautz* edo *Musika Plaza* bezalakoak, hala nola, kategoria honetakoak izango lirateke.

- **Bestelako izen bereziak**. Kategoria hau entitate sailkatzaile batetik bestera nabarmenki alda daiteke, baina hasiera batean aurreko hiruretan lekurik ez luketen izen bereziak bilduko lituzke. Adibide modura, *Bigarren Mundu Gerra*, *Lagun Izoztua* ala *Windows* modukoak kategoria honetan sailka litezke.

- **Zenbaki dezimalak**. Kategoria honek *3,1415*, *1915756*, *1.915.756*, ala *1.915.756,15* moduko zenbakiak bilduko lituzke. Entitate sailkatzailetik entitate sailkatzailera alde handiak egon ohi dira zenbakiei loturik, besteak beste *5,4kg* moduko neurri, *15€* moduko moneta nahiz *urtarrilak 13* moduko datentzat kategoria bereziak sortzea ohikoa delarik. Hori dela eta, sailkatzaile ezberdinak erabilita ere portaera koherente bat eskuratu ahal izateko halako kategoriak erabat baztertu eta zenbakien tratamendu berezi bat egingo da, adierazpen erregularren bidez ezagutuz hauek. Bide

horretan, aintzakotzat hartu behar da hizkuntzatik hizkuntzara milakoen eta dezimalen banatzaileak ezberdinak izan daitezkeela. Euskarak eta gaztelaniak, adibidez, puntua erabiltzen dute milakoak banatzeko eta koma, berriz, dezimalentzako. Ingelesaren kasuan, baina, alderantziz da hain justu ere, komak milakoak banatzen dituelarik eta puntuak, berriz, dezimalak. Gauzak honela, zenbakiak ezagutzeko erabiliko den adierazpen erregularra ondokoa da, non `T` ikurrak milakoen banatzailea adierazten baitu eta `D` ikurrak dezimalena:

```
[+-]?([0-9]{1,3}(M[0-9]{3})+|[0-9]+)(D[0-9]+)?
```

2.1.5 Analisi sintaktikoa

Analisi sintaktikoaren egitekoa testu bat bere osagaietan banatzea da. Osagai hauen izaera eta mailakatzearen arabera, analisi hau sakona nahiz azalekoa izan daiteke.

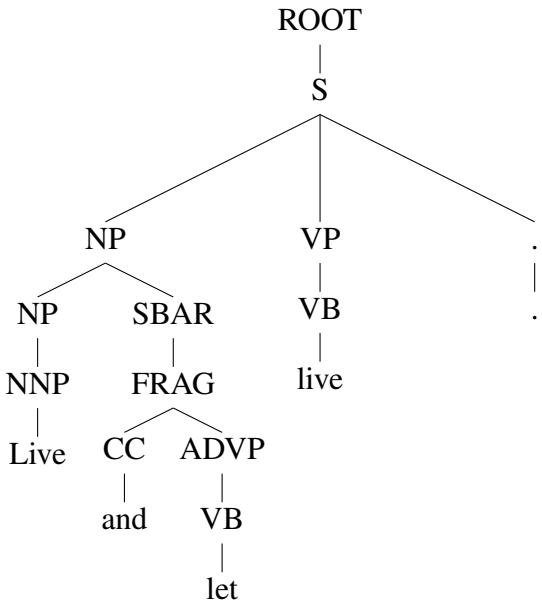
Analisi sintaktiko sakona

Analisi sintaktiko sakonean perpausen osagaiak modu mailakatu eta egituratuan ezagutzen dira, zuhaitz sintaktiko deritzona sortuz. Zuhaitz itxurako egitura honen erroa esaldiari dagokio, hostoetan tokenak kokatzen dira, eta tarteko barne-adabegiak maila ezberdinetako unitate sintaktikoei dagozkie, euren funtzioa adierazten duten etiketen bidez identifikatuak direnak. 2.1 irudiak Stanford CoreNLP analizatzaileak *Live and let live* esaldiarentzat eraikiriko zuhaitz sintaktikoa erakusten du adibide gisa.

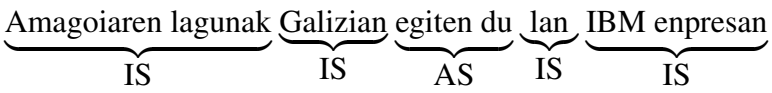
Azaleko analisi sintaktikoa

Azaleko analisi sintaktikoan *chunk* deritzen maila bereko elementuak ezagutzen dira. Era honetara, esaldiaren oinarrizko osagaiak identifikatzen dira (izen multzoak eta aditz multzoak, oro har), baina euren barne egitura nahiz funtzioa zehaztu gabe. Adibide modura, 2.2 irudiak Eustagger analizatzaileak *Amagoiaren lagunak Galizian egiten du lan IBM enpresan* esaldian ezaguturiko *chunk*ak erakusten ditu.

Proiektu honetan analisi sintaktiko sakonarekin egingo da lan baina, zoritxarrez, erabiltzeko asmoa den euskarazko tresnak azaleko analisi sintaktikoa baino ez du egiten. Hori dela eta, azaleko analisi sintaktikoaren irteera sakonak sorturiko zuhaitzen kasu partikulartzat hartzearen alde egin da, non *chunk*ak maila bakarreko barne-adabegiak izango bailirate-



2.1 Irudia: Analisi sintaktiko sakonaren adibide bat Stanford CoreNLP erabiliz

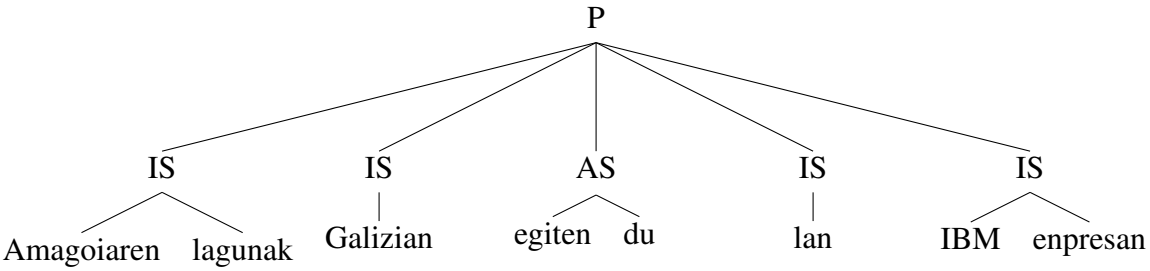


2.2 Irudia: Azaleko analisi sintaktikoaren adibide bat Eustagger erabiliz

ke. Modu honetara, aurreko adibidean ikusiriko analisia 2.3 irudiko zuhaitz sintaktikoa bailitzan interpretatuko litzateke.

2.1.6 Baliatutako softwarea

Proiektu honek oraindaino ikusiriko segmentazioa, analisi morfologikoa, etiketatzea, entitate-izenen ezagutzea eta analisi sintaktikoa burutzeko tresnak darabiltza oinarrizko baliabide gisara. Nolanahi ere, eskuragarri diren analizatzaileen artean ez da aurkitzen eskakizun horiek landu gura diren hizkuntza guztientzat betetzen dituenik. Gauzak honela, eta proiektuaren izaera hedagarriarekin bat etorriz, hizkuntzaren analisirako hainbat software integratzearen alde egin da. Honela, Freeling, Stanford CoreNLP eta Eustagger analizatzaileekin egin da lan, ondorengo lerroetan sakonago aztertzen direnak, etorkizunean beste pakete batzuk erraztasunez integratzeko aukera emanez era berean. 2.3 taulak analizatzaile horietako bakoitzak hizkuntza ezberdinentzat eskainiriko euskarria laburbiltzen du.



2.3 Irudia: 2.2 irudiko azaleko analisiaren interpretazioa zuhaitz sintaktikoaren kasu partikularrtzat

	Freeling										CoreNLP	Eustagger
	as	ca	cy	en	es	fr	gl	it	pt	ru	en	eu
Hitz-segmentazioa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Esaldi-segmentazioa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Analisi morfologikoa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Etiketatzea	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Entitate-izenen ezagutzea (erregelak)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Entitate-izenen ezagutzea (ikasketa automatikoa)		✓		✓	✓				✓		✓	✓
Entitate-izenen sailkapena	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Analisi sintaktiko sakona	✓	✓		✓	✓		✓				✓	
Azaleko analisi sintaktikoa	✓	✓		✓	✓		✓		✓			✓

2.3 Taula: Analizatzaile bakoitzak hizkuntza ezberdinetarako eskainiriko euskarria

FreeLing

FreeLing (Padró and Stanilovsky, 2012) hizkuntzaren analisirako tresna multzo bat da, Universitat Politècnica de Catalunyako TALP ikerketa-zentroak garatua Lluís Padróren gidaritzapean eta GNU General Public License lizentziapean eskainia. Komando-lerroko interfaze bat izan arren, hirugarren aplikazioak garatzeko liburutegi gisara erabil dadin dago diseinatua. Honela, C++ programazio lengoaian idatzita baldin badago ere, Java, Perl, PHP, Python eta Ruby lengoaietarako loturak ere eskaintzen ditu.

FreeLing hainbat hizkuntzarekin lan egin ahal izateko diseinaturik dago bere egitura modular eta hedagarriari esker eta, honela, hastapenetako ingelesa, gaztelania eta katalanari frantsesa, galiziera, italiera, errusiera, portugesa, galesa eta asturiera gehitu zaizkie komunitatearen ekarpenen bidez, eta txekiera eta esloveniera ere bidean dira. Nolanahi ere, hizkuntza guztien kasuan euskarria ez da erabatekoa, modulu jakin batzuetara mugatuz horietako batzuetan 2.3 taulak zehaztu bezala. Hori dela eta, aurrerantzean ikusiko denez Freelingek darabiltzan hizkuntza batzuk ez dira erabat erabilgarri izango proiektu honetan, eta beste batzuk, berriz, erabilgarri izan arren erabateko euskarria dutenen aldean kalitate baxuagoko emaitzak ematea espero daiteke.

Amaitzeko, aipatzekoa da 2.3 taulan jasoriko eta proiektu honetan erabiliko diren moduluez gain FreeLingek beste hainbat tresna ere eskaintzen dituela, besteak beste hizkuntza hautemateko modulua, antzeko hitzak iradokitzekoa, hitzen adiera posible guztiak etiketatzekoa, hitzen adiera desanbiguatzekoa, kodetze fonetikoa ematekoa edota korreferentziak ebaztekoa.

Stanford CoreNLP

Freelingen antzera, Stanford CoreNLP hizkuntzen analisirako tresna multzo bat da, Stanford Unibertsitateko Natural Language Processing Group taldeak garatua eta GNU General Public License lizentziapean eskainia. Besteak beste proiektu honetan erabiliko den etiketatzaile bat (Toutanova et al., 2003), entitate-izenen ezagutzaile bat (Finkel et al., 2005) eta analizatzaile sintaktiko bat (Socher et al., 2013) biltzen ditu, baina aipatzekoa da horietaz nahiz 2.3 taulan jasoriko gainerako funtzioez gain beste hainbat ere eskaintzen dituela, korreferentzien ebazpena eta sentimenduen analisia adibidez. Komando-lerroko interfaze bat izanagatik, Stanford CoreNLPren erabilera nagusia Java liburutegi modura da.

2.3 taulak erakusten duenez, Stanford CoreNLP ingelesera soilik mugatzen da eta, proiektu honetan baliatuko diren tresnei dagokienez, Freelingek bezala erabateko euskarria eskaintzen du hizkuntza honentzat. Hori dela eta, Freeling edukita Stanford CoreNLPren ekarpena hutsala litzatekeela pentsa zitekeen baina, egiari zor, baditu hainbat abantaila lehenaren aldean:

- **Analisien kalitatea.** Ingelesean soilik zentratuz, Stanford CoreNLP hizkuntza honen neurrira diseinatu eta eraikia izan da, IXA taldearen barne ebaluazioek erakutsirikoaren arabera emaitza nabarmenki hobeak eskuratuz hizkuntza honentzat.

- **Integrazioa.** Javaz idatzirik dagoenez, hobekiago txerta daiteke proiektuan, plataforma ezberdinetan funtzionatzen duten exekutagarri eramangarriak modu erraz eta erosoan sortzeko aukera ematen duena. Freelingen kasuan, berriz, instalazio prozesua eta Javarekiko lotura konplexuagoak suertatzen dira, batik bat GNU/Linux ingurunetik kanpo lan egin nahi bada.

Edozelan ere, badira beste hainbat puntu aurrekoei loturik Freelingen alde jokatzen dutenak:

- **Beste hizkuntzekiko batasuna.** Hizkuntza ezberdinen analisiak uztartzerakoan, tresna berak sortuak izatea lagungarri izan daiteke, irizpide bertsuak jarraitu izana espero baitaiteke.
- **Errendimendua.** Eginiko proben arabera, Freeling Stanford CoreNLP baino azkarragoa suertatzen da, bai kargatze-denboran, bai eta, batez ere, analisian. Era berean, bere memoria-eskakizunak ere nabarmenki baxuagoak dira.

Gauzak honela, bataren ala bestearen aldeko hautua kasuan kasu aztertu beharreko zerbait da. Proiektu honen ikuspegitik, baina, Stanford CoreNLPk emaitza hobeak eman ditzakeela espero da, bere integrazio aukerak ere hobeak dira, eta errendimenduari garrantzia handia emango bazaio ere, ez da baliabide mugatuko inguruneetan (hala nola, plataforma mugikorretan) izan zitekeen besteko puntu kritikoa. Hori dela eta, Freelingez landa Stanford CoreNLPrako euskarria ere eskaintzea deliberatu da proiektu honetan eta, biak erabiltzeko aukera emango bada ere, berau izanen da ingelesarekin lan egiteko hobetsiriko analizatzailea.

Eustagger

Euskararen analisirako IXA taldearen barne *pipeline*a baliatuko da, Morfeus analizatzaile morfologikoak (Aduriz et al., 1999), Eustagger etiketatzaileak (Aduriz and Díaz de Ilarraza, 2004), Eihera entitate-izenen ezagutzaileak (Alegria et al., 2006) eta Ixati azaleko analizatzaile sintaktikoak (Aduriz et al., 2004) osatua. *Pipeline* hau taldearen barne erabilerarako baino ez da oraingoz, eta ez dago publikoki eskuragarri. Nolanahi ere, bere bertsio arin eta oinarrizkoago bat argitaratzeko asmoa da, Eustagger izenez ezagutuko dena eta, hori dela eta, termino hau erabiliko da aurrerantzean *pipeline* osoari erreferentzia egiteko. Gogoan izan beharko da, baina, bertsio publikoak ez lituzkeela emaitza berdinberdinak emango, eta moldaketa batzuk eska litzakeela behar bezala funtzionatzeko.

Honetaz gain, aipatzekoa da Freeling eta Stanford CoreNLP tresnen kasuan ez bezala, *pipeline* honen erabilera nagusia aplikazio independente modura dela. Honela, C++ liburutegi bat garapen-prozesuan badago ere, behe mailan funtzionatzeko diseinatua izaten ari da, eta ez du proiektu honetan erabiltzeko moduko heldutasunik gaur-gaurkoz. Hori dela eta, euskarazko testuak tratatzeko *pipeline*a eskuz exekutatu eta honen irteerako fitxategien gainean lan egitearen alde egin da. Erabaki honen ondorioz, ez da posible izanen euskarazko testuak denbora errealean tratatzea eta, honenbestez, xede hizkuntza bezala soilik erabili ahalko da hau, eta ez abiapuntuko bezala.

2.2 Baliabide linguistikoak

Aurreko atalean ikusiriko analisi-tresnek testu gordinetik era ezberdinetako informazio linguistikoa ateratzeko aukera ematen dute, hauek nolabait landu ahal izateko funtsezkoa dena. Atal honetan, berriz, baliabide linguistikoei, hots, datuei, helduko zaie, hau bezalako proiektuetan ezinbestekoak suertatzen direnak oinarrizko ezagutza iturri modura. Honela, azpiatal banatan hiztegi elebidunak, Wikipedia, eta corpus paraleloak izanen dira hizpide, proiektu honetan ematen zaien erabileraren arabera euren inguruko oinarrizko kontzeptuak azalduko direlarik.

2.2.1 Hiztegi elebidunak

Hiztegi elebidunek hizkuntza bateko hitzek beste hizkuntza batean izan litzaketen adierak biltzen dituzte, noranzko bakarrean nahiz bietan. Gizakiei zuzenduriko hiztegiak informazio hau modu errazean aurkitu eta uler dadin maketatu ohi dira, baliokidetza posibleak adibide nahiz azalpen gehigarriekin osatuz. Proiektu honetan, baina, testu lauzko fitxategi soilekin eginen da lan, lerroz lerro abiapuntu eta xede hizkuntzako hitzen arteko baliokidetzak bilduko dituztenak tabulazioz bereizirik. Zehazki, 2.4.3 atalean sakonago aztertuko den Matxin itzultzailetik hartutako entitate-hiztegiak erabiliko dira, Elhuyar eta Euskaltzaindiaren baliabideetatik abiatuta sortuak IXA taldearen moldaketa eta egokitzapen propioekin.

2.2.2 Wikipedia

Wikipedia eduki askeko Interneteko entziklopedia eleanitz bat da, milaka boluntariok elkarlanean garatua eta Wikimedia Fundazioak sostengatua. 2014ko otsailaren 2ko datuen arabera, 30 milioi artikulu baino gehiago ditu orotara 287 hizkuntzatan, besteak beste 4.438.940 ingelesez, 1.077.202 gaztelaniaz eta 165.647 euskaraz. 2009ko maiatzaz geroztik, bere eduki guztia Creative Commons Attribution-ShareAlike 3.0 Unported License (CC-BY-SA) *copyleft* lizentziapean eskuragarri da, eta hauxe bera da proiektuak egun darabilen lizentzia nagusia ere.

Artikulu hauek testu iturri zinez aberatsa izatetik haratago, metadatuek egiten dute Wikipedia hain baliabide erabilgarria hizkuntzaren prozesamenduaren arloan. Izan ere, artikulu bakoitzeko beste hizkuntzetakoekiko loturak nahiz berbideraketen bidezko ordezko

izenburuak biltzen ditu Wikipediak besteak beste. Gauzak honela, entziklopedia bat izaki artikulu hauetako asko eta asko pertsona, leku, erakunde nahiz bestelako entitateei dagozkiela jakinik, entitateak lerrokatu eta itzultzeko hiztegi moduan erabiltzeko asmoa da proiektu honetan.

Wikipediak MediaWikiren web API estandarra eskaintzen du, hirugarren aplikazioetatik bertako informazioa modu errazean atzitzeko aukera ematen duena web bidezko eskaeren bidez. Nolanahi ere, proiektu honetan bezala datuak modu masiboan prozesatu behar direnean, hurbilpen hau ez da bideragarria suertatzen. Halakoetan, Wikimediak berak eskaintzen dituen *dump*ak erabili ohi dira, XML fitxategi ezberdinetan Wikipediaren eduki guzti-guztia biltzen dutenak eta periodikoki argitaratzen direnak. Proiektu honetan, zehazki, halako *dump*ak IXA taldearen tresnen bidez iragazi eta prozesatuz sorturiko XMLen gainean egin da lan.

2.2.3 Corpus paraleloak

Corpus paraleloak bi hizkuntza ala gehiagotan eskuragarri diren testuak dira, non hizkuntza bateko elementu bakoitzak gainerako hizkuntzetan dagozkion elementuen esanahi bera baitu. Bestela esanda, maila jakin batean lerrokaturiko testu eleanitzak dira, lerrokaturiko osagai hauek elkarren itzulpenak izanik. Oro har, eta proiektu honetan, esaldimailan egingo da lan.

Corpus paraleloen abiapuntua elkarren itzulpentzat hartzen diren testu gordinak izan ohi dira, kasuan kasu corpus elebidun ala eleanitz deritzena. Abiapuntu honetatik corpus paraleloa eskuratzeko, bada, bi urrats behar dira eman: esaldi-segmentazioa, 2.1.1 atalean landua, eta esaldi-lerrokatzea, behin hizkuntza bakoitzean esaldi-unitateak identifikatuta euren arteko lerrokatzeak ezartzea xede duena. Azken honetarako oinarrizko hurbilpena esaldien luzera eta posizioari erreparatzean datza, baina esaldien edukia aintzakotzat hartzen duten algoritmo sofistikatuagoak ere izan badira. Esaldi ez beste maila batean lan egin nahiko balitz, jarraitu beharreko urratsak berdin-berdinak lirateke funtsean, segmentazio- eta lerrokatze-prozesuak helburu den mailako elementuekin egin beharko baina.

Itzulpen-memoriak

Corpus paraleloen jatorri ohiko eta agerikoena aurkeztu berri den hau izanagatik, badira corpus paraleloen definizioa betez aipamen berezia merezi duten oso bestelako baliabi-

deak ere: itzulpen-memoriak.

Itzulpen-memoriak giza itzulpenen datu-baseak dira, segmentu deritzen oinarrizko unitateei (oro har, esaldiei) dagozkien jatorrizko testua eta testu itzulia jasotzen dituztenak. Arestiko hurbilpenean ez bezala, itzulpen-memorien kasuan segmentuen araberako metaketa hau jatorrian bertan, hots, testua itzuli ahala, egiten da, etorkizuneko itzulpenak egiterakoan lagungarri izan dadin jada eginikoak nolabait berrerabiltzeko aukeraren bidez.

Euren izaera dela eta, itzulpen-memoriak konputagailuz lagunduriko itzulpen-tresnen bidez sortu eta erabiltzen dira nagusiki, eta eginkizun zehatz horretarako diseinaturiko softwareari *itzulpen-memorien kudeatzaile* deritzo. Aplikazio hauek itzuli beharreko testua aurkezten diote itzultzaileari segmentuka banaturik, hauetako bakoitzeko datu-basean topaturiko bat-etortzeen araberako itzulpenak proposatuz halakorik bada. Erabiliriko software eta ezarpenen arabera, bat-etortze hauek zehatzak izan daitezke, hitzez hitz betetzen badira, bai eta lausoak ere, antzekotasun neurri baten araberako hurbiltasun bat betetzen bada. Itzultzaileak era honetara proposaturiko itzulpenak bere horretan onar ditzake, bai eta nolabait moldatu nahiz erabat baztertu ere. Egiten duena egiten duela, itzulpen-memorien kudeatzaileek itzultzaileak eginiko itzulpena datu-basean jasotzen dute, itzulpen-memoria aberastuz etorkizuneko iradokizunak hobetze aldera. Halako tresna aski ezagun bat OmegaT da, Javaz idatzia, multiplataforma, eta GNU General Public License lizentziapean eskainia.

Azkenaldian, prozesamendu sakonago baten bidez itzulpen gehiago nahiz itzulpen partzialak iradoki ahal izateko hainbat saiakera eta proposamen egin dira, eta bigarren zein hirugarren belaunaldiko itzulpen-memoriez mintzo izan da honela (Gotti et al., 2005). Joera honek itzulpen-memorien eta adibideetan oinarrituriko itzulpen automatikoaren arteko aldea txikitu du, azken honetan erabili izan diren hainbat teknika itzulpen-memorien arlora ekarriz (Somers and Fernández Díaz, 2004).

Azken horren harira, proiektu honen jomuga itzulpen automatikoaren kalitatea hobetzea bada ere, horretarako proposaturiko hurbilpenak itzulpen-memorien arloko azken berrikuntzekiko antzekotasun handiak ditu, eta esparru honetan ere aplikazio argia izan lezake.

2.3 Hitz-lerrokatzea

Hitz-lerrokatzearen helburua corpus paralelo batean eta, oro har, esaldi-mailan, hitzen arteko itzulpen-erlazioak zehaztea da. Honela, hitz-lerrokatzearen irteera grafo bipartigarri

bat da, hizkuntza bakoitzeko hitz bakoitza beste hizkuntzan bere itzulpentzat hartzen direnekin lotzen duena.

Hitz-lerrokatzea baliabide oso erabilia da hizkuntzaren prozesamenduaren arlo ezberdinetan eta, bereziki, hurbilpen estatistiko bat jarraitzen denetan. Benetan erabilgarria izan dadin, baina, hitz-lerrokatzea testu bolumen handi baten gainean behar izan ohi da, eskuz egin dadin aukera erabat bideraezin egiten duena. Hori dela eta, eskuz beharrean ikasketa gainbegiratugabearen bidez burutu ohi da eredu estatistiko ezberdinak aplikatuz, ezagunenak IBM 1-5 Ereduak (*IBM Models 1-5*, konplexutasun gehigarriko bost eredu ezberdin direnak) eta Markoven Eredu Ezkutua (HMM edo *Hidden Markov Model*) izanik.

Aipatu berri diren eredu guztiak ere sinplifikazio handi batetik abiatzen dira, F xede hizkuntzako hitz bakoitza E abiapuntuko hizkuntzako beste batekin bakarrik lotzea onartuz. Baliokide zuzenik ez duten xede hizkuntzako hitzak zerbaitekin lerrokatu ahal izateko, berriz, abiapuntuko hizkuntzako 0. posizioan *null* sasi-tokena aurkitzen dela suposatzen dute. Gauzak honela, lerrokatze bat $A = a_1, a_2, \ldots, a_J$ modura adieraz daiteke, non a_j xede hizkuntzako j. hitz edo tokenari lotu zaion abiapuntuko hizkuntzakoaren posizioa izango baita, I eta J abiapuntu eta xede hizkuntzetako esaldien luzerak izanik hurrenez hurren eta $0 \leq a_j \leq I$ betez.

Abiapuntu honekin, eredu bakoitzak E abiapuntuko hizkuntzako token-segida bat emanda A lerrokatzearen bidez F xede hizkuntzako token-segida lortzeko probabilitatea modelatzen du modu batera ala bestera, probabilitate hori maximizatzen duen lerrokatzea aukeratuz:

$$\hat{A} = \arg\max_A P(F, A|E)$$

Guztien artean sinpleena den IBM 1 Ereduak, adibidez, abiapuntuko hizkuntzako $E = e_1, e_2, \ldots, e_I$ token-segidak eta $A = a_1, a_2, \ldots, a_J$ lerrokatzeak $F = f_1, f_2, \ldots, f_J$ xede hizkuntzako token-segida sor dezaten probabilitatea honela kalkulatzen du, $t(f_x|e_y)$ e_y tokenaren itzulpena f_x izateko probabilitatea izanik:

$$P(F|E, A) = \prod_{j=1}^{J} t(f_j|e_{a_j})$$

Bestalde, IBM 1 Ereduak lerrokatze posible guztiek probabilitate berbera dutela suposatzen du. Honenbestez, J xede hizkuntzako luzera bakoitzeko $(I+1)^J$ lerrokatze ezberdin posible daudenez eta benetako luzera J izateko probabilitatea ε konstante txiki bat dela joz, E abiapuntuko hizkuntzako token-segida bat emanda haren lerrokatzea A izateko

probabilitatea honela kalkulatzen du:

$$P(A|E) = \frac{\varepsilon}{(I+1)^J}$$

Eta, orain arte ikusiriko bi berdintzak uztartuz, corpus paraleloko sarrera bakoitzeko ondorengo $\hat{A}$ lerrokatzea hartuko luke:

$$\begin{aligned}\hat{A} = \arg\max_A P(F,A|E) &= \arg\max_A P(F|E,A) \times P(A|E) \\ &= \arg\max_A \frac{\varepsilon}{(I+1)^J} \prod_{j=1}^{J} t(f_j|e_{a_j})\end{aligned}$$

Azkenik, eredu honetan hitz bakoitzaren lerrokatzea ingurukoekiko modu independentean erabakitzen denez, aurreko adierazpena honakora laburbil daiteke:

$$\hat{A} = \arg\max_{a_j} t(f_j|e_{a_j}) \quad 1 \leq j \leq J$$

Gainerako ereduek azaldu berri denak abiapuntutzat dituen sinplifikazioak gainditzeko ahalegina egiten dute eta, ondorioz, nabarmenki konplexuagoak suertatzen dira. Edozelan ere, guztiek uztartzen dituzte, era batera ala bestera, ondorengo bi elementuak:

- **Hitzen arteko lerrokatzea** bera, $A = a_1, a_2, \ldots, a_J$ modura adierazi duguna.
- **Pisu lexikalak** edo hitzen arteko itzulpen-probabilitateak, IBM 1 Ereduaren kasuan $t(f_x|e_y)$ adierazpenari zegokiona.

Bistakoa denez, pisu lexikalak ezagututa hitzen arteko lerrokatzea erraz jakin liteke, horixe baita, hain justu ere, lerrokatze-ereduak modelatzen duena. Era berean, hitzen arteko lerrokatzea ezagutuz gero pisu lexikalak ere erraz kalkula litezke, hitz bikote bakoitzaren agerpenak zenbatu eta balio hori abiapuntuko hizkuntzako hitzaren agerpen kopuruarekin zatituz normalizatu baino ez bailitzateke egin beharko.

Bai hitzen arteko lerrokatzea bai eta pisu lexikalak ere, baina, ezezagunak dira hasiera batean eta, honenbestez, halako zerbait zuzen-zuzenean egitea ezinezkoa da. Nolanahi ere, bistakoa da bietako baten nolabaiteko estimazio bat izanez gero bestea berrestima zitekeela ikusi berri den bezala. Behin hau eginda, bigarrenaren bidez atzera lehena berrestima zitekeen are estimazio zehatzagoa lortuz eta, nahi izanez gero, prozesua berriro errepika zitekeen, are estimazio hobea emango lukeena. Bada, hauxe da, hain zuzen ere, hitz-lerrokatzaileek darabilten itxaropen-maximizazio (EM ala *expectation-maximization*) algoritmoaren atzean dagoen ideia. Zehatzagoak izanez, algoritmo honek

itzulpen- nahiz lerrokatze-probabilitateak modu uniformean hasieratzen ditu, iterazio bakoitzean lerrokatze-probabilitateak itzulpen-probabilitateen arabera eguneratuz eta, behin hau eginda, hitz bakoitzaren itzulpen-probabilitatea lerrokatze-probabilitate berrien arabera birkalkulatuz. Ikus daitekeenez, sekula bukatzen ez den soka da hau eta, honenbestez, iterazio kopuru aurrezarri bat burutu ohi da, geroz eta handiagoa izan orduan eta emaitza hobeak emango lituzkeena.

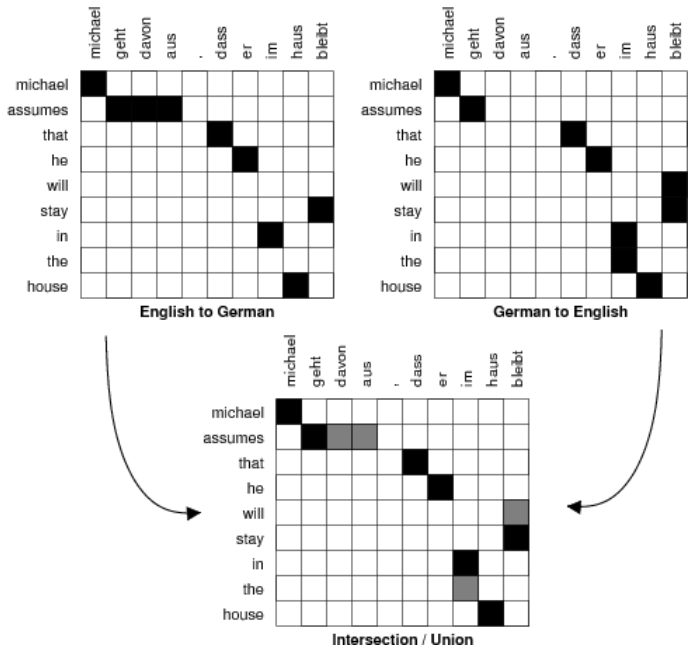
Bukatzeko, aipatu izan diren eredu guztiek abiapuntutik egiten zuten sinplifikazio nagusia gogora ekarriz, modu honetara xede hizkuntzako hitz bakoitza gehienez ere abiapuntuko hizkuntzako beste batekin lerroka daiteke. Lerrokatze-prozesua errazagoa eginagatik, bistakoa da murriztapen honek emaitzen kalitatea nabarmenki mugatzen duela, berez posible baita xede hizkuntzako hitz bat abiapuntuko hizkuntzako beste hainbatekin lerrokatzea komeni izatea. Honi aurre egiteko, lerrokatze-simetrizazio deituriko teknika erabili ohi da, lerrokatzea bi zentzuetan egin eta bien emaitzak konbinatzean datzana. Horretarako, muturreko bi aukerak lerrokatzeen ebakidura edo bildura hartzea lirateke. Lehenak lerrokatze zuzen dezente kanpoan utz litzake, baina harturikoak zehaztasun handikoak lirateke. Bigarrenak, berriz, apenas utziko luke lerrokatze zuzenik kanpoan, baina harturikoen zehaztasuna dezente txikiagoa litzateke aldi berean. Adibide modura, 2.4 irudiak bi aukeren eragina erakusten du kasu jakin batean. Biek ala biek euren mugak dituztela ikusirik, bada, simetrizazio-metodo sofistikatuagoak garatu izan dira, oro har lerrokatzeen arteko ebakiduratik hasi eta irizpide jakin batzuen arabera hari bildurako elementuak eransten funtzionatzen dutenak.

2.3.1 Baliatutako softwarea

Proiektu honek corpus paraleloak hitz mailan lerrokatuz egiten du lan eta, hori dela eta, hitz-lerrokatzaileak funtsezko baliabideak dira bertan. Zentzu honetan, GIZA++ eta Berkeley Aligner lerrokatzaileak integratzeko aukera aztertu zen eta, bakoitzak bere alde onak eta txarrak izan zitzakeela ikusirik, bientzako euskarria eskaintzea deliberatu da. Jarraian, bada, lerrokatzaile hauetako bakoitza nor bere aldetik aztertzen da, erabaki honen zergatia argituz era berean.

GIZA++

GIZA++ lerrokatzailea (Och and Ney, 2003) GIZA deituriko beste baten hedapen modura sortu zen Franz Josef Och-en eskutik, eta GNU General Public License lizentziapean



2.4 Irudia: Ebakidura eta bildura lerrokatze-simetrizazioan

eskaintzen da. Arestian aipatu diren IBM 1-5 Ereduak eta Markoven Eredu Ezkutua inplementatzen ditu hainbat hedapen, hobekuntza eta optimizaziorekin, eta ezbairik gabe lerrokatzaileen artean *de facto* estandarra izatera iritsi da. Bere nagusitasuna dela eta ezinbestekotzat jo da proiektu honetan ere harentzako euskarria eskaintzea, bere sendotasun eta fidagarritasuna ez ezik garaturiko sistema bestelakoekin bateragarri eta alderagarria izan dadin ematen duen aukera biziki interesgarria suertatzen baita. Desabantaila modura, baina, integrazio hau dezente korapilatsua suertatzen da, GIZA++ C++ lengoaian idatzirik baitago, proiektu honetan berau erabiltzeko aukera bakarra kanpo-prozesu ala aplikazio independente modura izanik.

Berkeley Aligner

Berkeley Aligner Berkeley Unibertsitateko hizkuntzaren prozesamenduko taldeak garaturiko lerrokatzaile bat da, GNU General Public License lizentziapean eskaintzen dena. Ikerketa-arloan hitz-lerrokatzearen inguruan eman izan diren azken berrikuntzentzat eskainiriko euskarriak egin du ezagun Berkeley Aligner, besteak beste distortsio sintaktikoaren bidez 2.1.5 atalean azalduriko analisi sintaktikoa lerrokatze-prozesuan barneratzeko (DeNero and Klein, 2007) nahiz lerrokatze-simetrizaziora bidean bi zentzuetako entrenamendua modu bateratuan egiteko (Liang et al., 2006) aukera ematen duelarik. Proiektu

honetan sintaxiari eta lerrokatzeari loturiko murriztapenekin egingo denez lan, bigarrena lehena aintzakotzat hartuz egiteko aukera zinez interesgarria suertatzen da, emaitza trinkoagoak lortu eta murriztapenak gehiagotan betetzeko aukera eman lezakeela aurreikus baitaiteke. Hori gutxi balitz, Berkeley Aligner Javaz garaturik dago eta, berez aplikazio independente modura erabiltzeko diseinatua izan bazen ere, liburutegi modura txertatzeko aukera ematen du horrek, GIZA++ lerrokatzailearekin baino integrazio egokiagoa lortuz. Gauzak honela, aipatu berri den GIZA++ ez ezik Berkeley Aligner ere erabiltzeko aukera eskaintzearen alde egin da, aldagai esperimental interesgarri bat sartzeaz gain sistemaren erabilera erraz baitaiteke bera erabiliz gero.

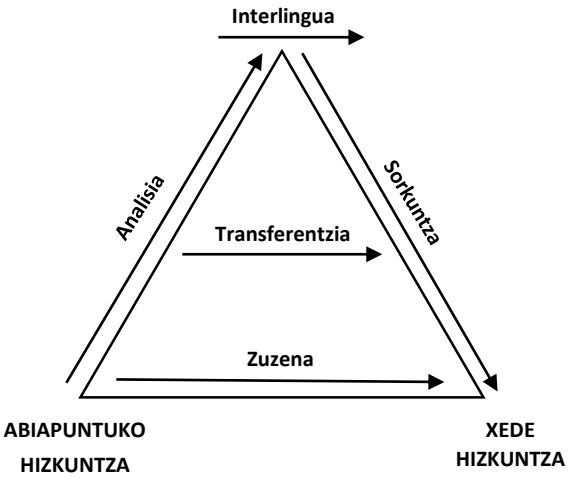
2.4 Itzulpen automatikoa

Itzulpengintza automatikoaren helburua konputagailu bidez hizkuntza batetik beste baterako itzulpena automatizatzea da. Hizkuntzaren prozesamenduaren aplikazio esanguratsuenetarikoa izan da hau bere hastapen-hastapenetatik, problemaren osotasun eta aberastasunak berez erakargarria egiteaz gain interes praktiko handikoa suertatzen baita geroz eta mundu globalago batean.

Itzulpen automatikoaren erronkari aurre egiteko estrategia oso ezberdinak jarraitu izan dira. Ikuspegi orokor batetik, baina, bitan bana daitezke hurbilpen hauek, arrazionalismotik joz eta giza ezagutza linguistikoa abiapuntutzat hartuz diharduen **erregeletan oinarrituriko itzulpengintza automatikoaren** eta enpirismotik joz eta jada eginiko itzulpenak abiapuntutzat hartuz aritzen den **corpusetan oinarrituriko itzulpengintza automatikoaren** artean bereiziz. Azken honen baitan, halaber, itzulpenak analogiaz egiten dituen **adibideetan oinarrituriko itzulpengintza automatikoa** nahiz eredu estatistikoekin diharduen **itzulpen automatiko estatistikoa** aurkitzen dira. Ondorengo lerroetan, bada, hurbilpen hauek guztiak aztertzen dira nor bere aldetik azpiatal banatan. Kapituluaren hasieran zehaztutako erreferentziez gain, Labaka (2010), Nirenburg (1993), Knight (1999) eta Somers (2003) ere erabili dira horretarako.

2.4.1 Erregeletan oinarrituriko itzulpen automatikoa (RBMT)

Erregeletan oinarrituriko itzulpen automatikoa (RBMT edo *Rule-based Machine Translation* modura ere ezaguna), jatorri eta xede hizkuntzen gaineko ezagutza linguistikoan funtsatzen da itzulpen automatikoaren erronkari aurre egiteko. Ezagutza honen izaera,



2.5 Irudia: Vauquoisen triangelua

abstrakzio maila eta berau aplikatzeko baliaturiko tarteko adierazpidearen arabera, hurbilpen ezberdinak jarraitu izan dira horretarako, sistema zuzenen, transferentzian oinarriturikoen eta interlinguan oinarriturikoen artean sailkatu izan direnak. Hiru estrategia hauek irudikatzeko, Vauquoisen triangelua delakoa erabili ohi da, 2.5 irudian ematen dena. Ikus daitekeenez, gailurretik geroz eta hurbilago egon orduan eta tarteko adierazpide konplexuagoa erabiltzen da, bertara iristeko prozesamendu sakonago bat eskatzen duena. Zehazki, sistema bakoitzaren funtzionamendua honakoa da:

- **Sistema zuzenek** itzulpena hitzez hitz eta urrats bakar batean egiten dute inolako tarteko adierazpiderik erabili gabe. Horretarako darabiltzaten baliabide nagusiak hiztegi elebidunak dira, baina hauetaz gain oinarrizko komunztadura eta berrantolaketa arauak ere erabil ditzakete.

- **Transferentzian oinarrituriko sistemek** jakintza kontrastiboa, hots, bi hizkuntzen arteko ezberdintasunen gaineko ezagutza, dute oinarri, eta ezberdintasun hauek gainditzeko erregelekin egiten dute lan. Bide horretan, tarteko bi adierazpidez baliatzen dira, bat jatorrizko hizkuntzarentzat eta beste bat helburukoarentzat, itzulpena ondorengo hiru fasetan burutuz:

  – **Analisi fasean** 2.1 atalean azalduriko bitartekoen bidez jatorrizko testua aztertu eta beroni dagokion tarteko adierazpide batera pasatzen da. Analisi honen nolakotasunaren arabera, sistema itzaleko transferentziakoa (*shallow-*

transfer) izan daiteke, azaleko analisi sintaktikoa egiten badu, ala transferentzia sakonekoa (*deep-transfer*), analisi sintaktiko sakona egiten badu.

  - **Transferentzia fasean** abiapuntuko hizkuntzaren tarteko adierazpidetik xede hizkuntzaren tarteko adierazpidera pasatzen da. Transferentzia sintaktikoko sistemek maila lexikoan eta estrukturalean egiten dute hau, hurrenez hurren hiztegi elebidunak eta transferentzia erregelak erabiliz. Transferentzia semantikoko sistemek, berriz, transferentzia lexikala eta estrukturala ez ezik semantikoa ere egiten dute, esanahiaren errepresentaziorako egiturak erabiliz.
  - **Sorkuntza fasean** xede hizkuntzaren tarteko adierazpidea testu itzulian bilakatzen da. 2.1.2 atalean azaldu bezala, hiztegi morfologikoak erabiltzen dira nagusiki horretarako.

- **Interlinguan oinarrituriko sistemek** tarteko adierazpide bakar bat erabiltzen dute, interlingua deitua eta edozein hizkuntzarekiko independentea dena. Era honetara, itzulpen-prozesua bi urratsetara murrizten da, analisi fasean jatorrizko testuaren esanahia interlinguaren bidez errepresentatuz eta sorkuntza fasean esanahi hau xede hizkuntzan adieraziz. Honela, transferentzian oinarrituriko sistemak hizkuntza bikote mailan aritzen baziren, interlinguan oinarriturikoek hurbilpen global bat jarraitzen dute, ikuspegi kontzeptual batetik lan eginez. Honen abantaila bat hizkuntza askoren artean itzultzeko eraginkortasuna da, haietako bakoitzeko interlinguan kodetu eta deskodetzeko modulu bat nahikoa baita, beste hizkuntzenekiko erabat independentea.

2.4.2 Corpusetan oinarrituriko itzulpen automatikoa

Corpusetan oinarrituriko itzulpen automatikoak jada eginiko itzulpenei atxikiriko ezagutza enpirikoa hartzen du itzulpen automatikorako abiapuntutzat. Informatikaren eta, bereziki, Interneten zabalkuntzak, testu bolumen handiekin lan egiteko aukera ekarri du azken urteotan, halako hurbilpenei indar eman diena. Horretarako baliabide nagusiak 2.2.3 atalean landuriko corpus paraleloak dira eta, zentzu honetan, hainbat hizkuntza ofizial dituzten herrialdeetako testu ofizialei dagozkienak baliatu izan dira batik bat, amaraunaren bidez edozeinen eskura direnak.

Corpus paraleloen abiapuntu honekin, bada, bi hurbilpen ezberdin jarraitu izan dira itzulpegintza automatikoan, adibideetan oinarriturikoa batetik eta estatistikoa bestetik, azpiatal banatan azaltzen direnak jarraian.

Adibideetan oinarrituriko itzulpen automatikoa (EBMT)

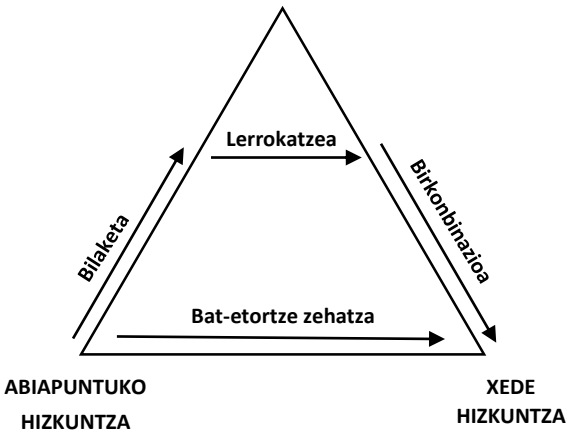
Adibideetan oinarrituriko itzulpen automatikoak (EBMT edo *Example-based Machine Translation* modura ere ezaguna) itzulpenak analogiaz egiten ditu, corpus paralelo bat erabiliz exekuzio-denboran ezagutza iturritzat. Ideia honen baitan, hurbilpen ezberdinak jarraitu izan dira, oro har ondorengo faseetan banatuak guztiak:

- ***Matching* edo bilaketa fasea**, itzuli beharreko testuaren bat-etortzeak topatzeaz arduratzen dena corpus paraleloan.
- ***Alignment* edo lerrokatze fasea**, bat-etortze bakoitzeko honi dagokion itzulpenaren testu-zatia identifikatzeaz arduratzen dena corpus paraleloan.
- ***Recombination* edo birkonbinazio fasea**, aurreko zatiak uztartuz irteerako testua sortzeaz arduratzen dena jatorrizkoaren itzulpen ahalik eta egokiena izan dadin saiatuz.

Oinarrizko planteamendua zeharo ezberdina izanagatik, urrats hauek analisi, transferentzia eta sorkuntza faseekin antzekotasun handiak dituzte, hurrenez hurren jatorrizko hizkuntzako testua tratatu, helburukora pasatu, eta azken itzulpena sortzeko hau moldatzeaz arduratzen baitira. Hori dela eta, adibideetan oinarrituriko itzulpengintza automatikoa transferentzian oinarriturikoarekin alderatu izan da. Honen harira, 2.6 irudiak EBMT sistemei egokituriko Vauquoisen triangelua erakusten du. Ikus daitekeenez, transferentzian oinarrituriko sistemekin ez ezik sistema zuzenekin ere parekatzen da kasu tribial batean, bat-etortze zehatzei dagokienean hain zuzen ere, lerrokatzearen beharrik gabe zuzenean corpuseko itzulpena berrerabiliko litzatekeelarik halakoetan.

Itzulpen automatiko estatistikoa (SMT)

Itzulpen automatiko estatistikoak (SMT edo *Statistical Machine Translation* modura ere ezaguna) corpus paraleloen azterketatik erauziriko eredu estatistikoak baliatzen ditu itzulpenak sortzeko. Era honetara, orain arte jorraturiko metodoek itzulpen-prozesua bera bazuten ardatz, itzulpen automatiko estatistikoak emaitzari erreparatzen dio abiapuntutzat. F abiapuntuko hizkuntzako testu bat emanda, bada, sistemaren egitekoa $\hat{E}$ xede hizkuntzako testu bat aurkitzea izango da, abiapuntukoaren itzulpen ahalik eta probableena izan beharko dena. Honenbestez, $P(E|F)$ espresioak F abiapuntuko testua finkatuta E bere itzulpena izateko probabilitatea adierazten badu, sistemaren helburua probabilitate hau



2.6 Irudia: Vauquoisen triangelua EBMT sistemei egokiturik

maximizatzen duen $\hat{E}$ xede hizkuntzako testua bilatzea izango da. Bayesen teorema aplikatuz, honela berridatz daiteke hau:

$$\hat{E} = \arg\max_E P(E|F) = \arg\max_E \frac{P(F|E)P(E)}{P(F)} = \arg\max_E P(F|E)P(E)$$

Ikus daitekeenez, azken berdintza honen arabera $\hat{E}$ itzulpenaren hautaketa bi faktoreren menpe dago: alde batetik abiapuntuko hizkuntzako testua xede hizkuntzako testuaren itzulpena izateko $P(F|E)$ probabilitatea eta, bestetik, xede hizkuntzako testua errealitatean topatzeko (edo, zentzu batean, xede hizkuntzan bertan testu zuzen eta egokia izateko) $P(E)$ probabilitatea. Honek problema bitan banatzeko aukera ematen du, lehenaz itzulpen-eredua arduratuko delarik eta bigarrenaz, berriz, hizkuntza-eredua:

$$\hat{E} = \arg\max_E \underbrace{P(F|E)}_{\substack{\text{itzulpen}\\ \text{eredua}}} \underbrace{P(E)}_{\substack{\text{hizkuntza}\\ \text{eredua}}}$$

Nola itzulpen hala hizkuntza-ereduak eredu estatistikoak dira, eta euren parametroak corpus erraldoiak aztertuz estimatzen dira. Hauetaz gain, itzultzaile automatiko estatistikoek hirugarren osagai bat ere badute: deskodetzailea, abiapuntuko hizkuntzako testua emanda xede hizkuntzako bere itzulpen probableena aurkitzeaz arduratzen dena. Banan-banan aztertuz, bada, SMT sistemen osagai bakoitzaren funtzionamendua honakoa da:

- **Itzulpen-eredua** abiapuntuko hizkuntzako esaldi bat eta xede hizkuntzako beste bat

emanda, azkenak lehena sortzeko probabilitatea esleitzeaz arduratzen da. Horretarako eredu ezberdinak jarraitu izan dira, hitzez hitz diharduten oinarri-oinarrizko hurbilpenetatik hasi eta ikerketa-arloan geroz eta protagonismo handiagoa duten egitura sintaktiko hierarkikoak darabiltzateneraino. Nolanahi ere, paradigma nagusia sintagmetan oinarriturikoa da (*phrase-based translation* delakoa), hitz-multzo jarraiak hartzen dituena oinarrizko itzulpen unitatetzat (kontuan izan behar da testuinguru honetan *phrase* edo sintagma terminoak hitz-segida bat adierazten duela, ez dena zertan unitate sintaktiko bat izan behar ikuspuntu linguistiko batetik). Halakoetan, itzulpen-eredua ondorengo adierazpenari dagokio:

$$P(F|E) = \prod_{i=1}^{I} \phi(\bar{f}_i, \bar{e}_i) d(a_i - b_{i-1})$$

Ikus daitekeenez, itzulpen-eredu hau bi faktoretan bereizten da: $\phi(\bar{f}_i, \bar{e}_i)$ itzulpen-probabilitatea eta $d(a_i - b_{i-1})$ distortsio-probabilitatea. Lehenak $\bar{e}_i$ xede hizkuntzako sintagmak $\bar{f}_i$ abiapuntu hizkuntzako sintagma sortzeko probabilitatea adierazten du. Bigarrena, berriz, bi hizkuntzetako sintagmen posizioen arteko distantziari dagokio. Zehazki, a_i xede hizkuntzako $i$garren sintagmak abiapuntuko hizkuntzan sorturiko sintagmaren hasierako posizioa da, eta b_{i-1} xede hizkuntzako $i-1$garren sintagmak abiapuntuko hizkuntzan sorturiko sintagmaren bukaerako posizioa. Honen arabera neurri ezberdinak har daitezke, hala nola distortsioa α konstante txiki bati berretuz:

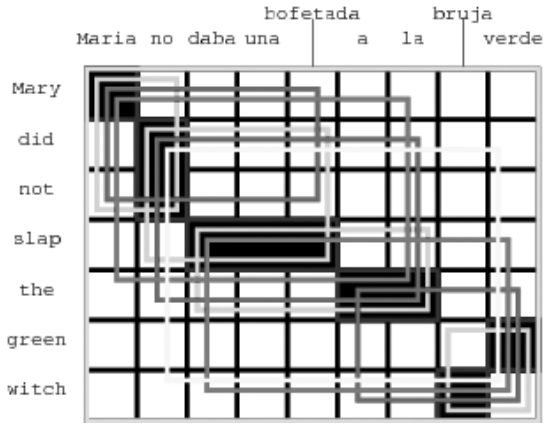
$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|}$$

$\phi(\bar{f}_i, \bar{e}_i)$ itzulpen-probabilitateak estimatzeko, berriz, corpus paralelo bateko sintagma bakoitza beste hizkuntzan dagokionarekin lotzen da, sintagma-lerrokatze deritzona. Behin hau eginda, $(\bar{f}, \bar{e})$ sintagma bikote bakoitza zenbatu eta balio hau normalizatzen da, probabilitateak eskuratuz:

$$\phi(\bar{f}, \bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f}, \bar{e})}$$

Sintagmak lerrokatu ahal izateko, bestalde, 2.3 atalean landuriko hitz-lerrokatzea hartzen da abiapuntutzat, azken hauekin kontsistenteak diren parekatze guztiak onartuz. 2.7 irudiak honen adibide bat erakusten du.

- **Hizkuntza-eredua** $w_1, \ldots, w_m$ moduko hitz-segida edo testu bat emanda, haren agerpen-probabilitatea esleitzeaz arduratzen da. Horretarako eredu ezberdinak pro-



2.7 Irudia: Sintagma lerrokatzea hitz-lerrokatzetik abiatuta

posatu badira ere, gehien-gehienek n-gramak dituzte oinarri. Halakoetan, testu baten agerpen-probabilitatea kalkulatzeko testu hori osatzen duen hitz bakoitza bere aurreko n hitzen ostean agertzeko probabilitateen biderkadura hartzen da, n parametro aldagarri bat izanik:

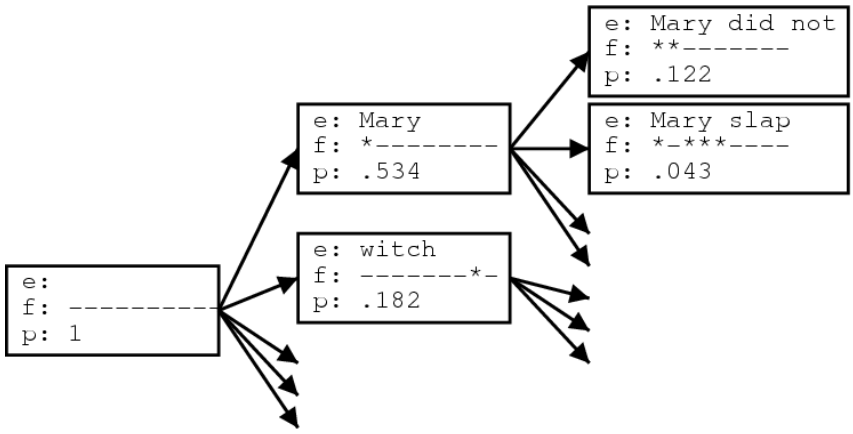
$$P(w_1,\ldots,w_m) \approx \prod_{i=1}^{m} P(w_i|w_{i-(n-1)},\ldots,w_{i-1})$$

Era honetara, n parametroa egokituz unigrama ($n=1$, hitz bakoitzaren probabilitatea modu independentean hartu eta biderkatzen duena), bigrama ($n=2$), trigrama ($n=3$) nahiz lau-grama ($n=4$) bezalako ereduak eraiki daitezke. Probabilitate baldintzatuak kalkulatzeko, berriz, funtsezko ideia corpus elebakar handi bat hartu eta n-grama bakoitzaren maiztasuna zenbatzean datza:

$$P(w_i|w_{i-(n-1)},\ldots,w_{i-1}) = \frac{\text{count}(w_{i-(n-1)},\ldots,w_{i-1},w_i)}{\text{count}(w_{i-(n-1)},\ldots,w_{i-1})}$$

Nolanahi ere, eredu honek zeroko probabilitatea esleituko lioke corpusean ageri ez den hitz edo n-grama bati eta, hori dela eta, *smoothing* edo leuntze teknika ezberdinak aplikatu ohi dira arestiko berdintzaren gainean.

- **Deskodetzailea** abiapuntuko hizkuntzako testu bat emanda, itzulpen- eta hizkuntza-ereduen arabera xede hizkuntzan honen itzulpen probableena litzatekeena sortzeaz arduratzen da. Algoritmiaren ikuspegitik, bilaketa problema bat da hau, baina egoera-espazioa hain da handia (luzera maximoa finkatu ezean, infinitua eta, bestela ere, hitz posible guztien konbinazio posible guztiak gogoan hartuz, erraldoia) ezen



2.8 Irudia: *Maria no daba una bofetada a la bruja verde* esaldiaren deskodetzea

soluzio zehatza bilatzea ez baita bideragarria inondik inora. Hori dela eta, bilaketa heuristiko bat burutu ohi da, xede hizkuntzako testua sintagmaz sintagma eraikitzen duena abiapuntuko hizkuntzako testu osoa barne hartu arte. Aztertu beharreko hurrengo hautagaia hautatzeko, jada eraikiriko itzulpenaren probabilitatea eta falta denaren estimazio bat uztartzen dituen funtzio bat erabiltzen da. Zehazki, algoritmorik erabiliena sorta-bilaketa edo *beam search* delakoa da, egoera espazioa mailaz maila aztertzen duena maila hauetako bakoitzean adabegi onenen kopuru aurrezarri bat hedatuz. 2.8 irudiak halako bilaketa heuristiko baten adibide bat erakusten du.

2.4.3 Baliatutako softwarea

Jada aipatu bezala, proiektu honen jomuga adibideetan oinarrituriko itzulpen automatikoko teknikak erabiliz itzulpen partzialak sortu eta, itzultzaile nagusiaren aurreprozesu gisara txertatuz, honen portaera hobetuko duten itzulpen-sistema hibridoak sortzea da. Erregeletan oinarrituriko itzulpen automatikoaren kasuan, hurbilpen arrazional eta enpirikoaren arteko uztardura lekarke honek, lehenaren zurruntasunarekin hautsi eta jada eginiko itzulpenak berrerabiltzeko aukera emanez hauen konfiantza maila handia denean. Itzulpen automatiko estatistikoaren kasuan, berriz, corpusen erabilera ez litzateke berria, baina orokortasuna lortzearren bere ereduen eraikuntzak dakarren doitasun galera dela medio, testu edo testu-zati errepikakorren kasuan berrerabilpen zuzen eta egokiago bat ekar lezake proposaturiko hurbilpenak.

Gauzak honela, interesgarria suertatzen da integrazioa bi motatako sistemekin burutzea eta, hori dela eta, bai itzulpen automatiko estatistikoko Moses softwarearekin bai eta erre-

geletan oinarrituriko itzulpen automatikoko OpenTrad proiektuko Apertium eta Matxin sistemekin lan egitearen alde egin da, atal banatan labur-labur azaltzen direnak jarraian.

Moses

Moses (Koehn et al., 2007) itzulpen automatiko estatistikoko sistema bat da, EuroMatrix proiektuaren baitan garatua nagusiki Europako Batzordearen babespean, eta GNU Lesser General Public License lizentziapean eskainia. Corpus paraleloen abiapuntu hutsarekin itzulpen-ereduak entrenatu eta erabiltzeko aukera ematen du, bai sintagmetan oinarrituriko ohiko hurbilpenaren bidez, bai eta zuhaitzetan oinarriturikoen nahiz eredu faktorizatuen bidez ere. Era berean, konfusio- nahiz hitz-sareekin lan egiteko aukera ematen du, hizketa-ezagutzaileak edota analizatzaile morfologikoak bezalako bitarteko anbiguoak erraztasunez txertatzea posible eginez.

OpenTrad proiektua: Apertium eta Matxin

OpenTrad proiektua (Loinaz et al., 2006) 2004. urtean sortu zen espainiar estatuko hizkuntzentzako itzulpen-sistema bat sortzeko asmoz bertako Industria, Turismo eta Merkataritza Ministerioaren babespean, eta beronen garapenean Euskal Herriko Unibertsitateko Ixa taldeak berak, Universitat d'Alacanteko Transducens taldeak, Universitat Politècnica de Catalunyako TALP taldeak eta Eleka Ingeniaritza Linguistikoak hartu zuten parte besteak beste.

OpenTrad proiektuak erregeletan oinarrituriko itzulpen automatikoaren hurbilpena jarraitzen du eta, estatuko hizkuntzen ezaugarri linguistikoak medio batzuen ala besteen artean itzultzearen arteko aldea dela eta, bi sistema nagusitan bereizten da:

- **Apertium** (Forcada et al., 2011) *shallow-transfer* edo itzaleko transferentziako itzulpen automatikoko sistema bat da, hau da, azaleko analisi sintaktikoarekin lan egiten duena. Izan ere, estatuko hizkuntza erromantzeekin lan egiteko izan zen diseinatua, elkarren arteko hurbiltasun linguistikoa handia izaki metodo sinple eta eraginkorrena suertatzen dena. Nolanahi ere, urteetan zehar hain hurbilekoak ez diren hizkuntzekin lan egiteko hedatua izan da, besteak beste ingelesa, bretoiera, suediera edota daniera bezalako hizkuntzentzako euskarria erantsiz. Honela, bere bertsio egonkorrean 37 hizkuntza bikoterekin lan egiteko aukera eskaintzen du egun, eta gehiago ere bidean dira.

- **Matxin** (Mayor et al., 2011) *deep-transfer* edo transferentzia sakoneko itzulpen automatikoko sistema bat da, hau da, analisi sintaktiko sakonarekin lan egiten duena. Honela, elkarren artean ezberdintasun linguistiko sakonak dituzten hizkuntza bikoteekin jarduteko diseinatua izan zen, azaleko analisi sintaktikoarekin nahikoa ez dutenak eta, bereziki, gaztelania-euskara bikotearen inguruan garatu eta erabili izan da.

Bai Apertium bai eta Matxin ere software librea dira, GNU General Public License lizentziapean eskainiak, eta hainbat osagai partekatzen dituzte elkarren artean. Osagai haueta-ko bat, lttoolbox, da, hain zuzen ere, 2.1.2 atalean azalduriko sorkuntza egiteko proiektu honetan baliatuko dena honen Java *port* lttoolbox-javaren bidez.

3. KAPITULUA

Analisia

Kapitulu honetan proiektuan ebatzi nahi den problemaren analisia egiten da. Honela, egungo itzulpen-sistemen berrikuspen batekin hasten da, EBMT bidezko hibridazioaren ideia azaldu ondoren eta, amaitzeko, hibridazio-estrategiatzat itzulpen partzialak erabiltzeko ideia garatu.

3.1 Egungo itzulpen-sistemen berrikuspena

2.4 atalean ikusirikoa gogora ekarriz, itzulpen automatikoko sistemak hiru multzo nagusitan sailkatu izan dira: erregeletan oinarrituak, adibideetan oinarrituak eta estatistikoak. Ikuspegi orokor batetik, bakoitzaren lorpen eta mugak ondorengoan laburbil daitezke:

- **Erregeletan oinarrituriko itzulpengintza automatikoa (RBMT)** itzuli asmo diren hizkuntzen gaineko ezagutza linguistikoan funtsatzen da, nolabaiteko erregelen bidez formalizatua dena. Horri esker, sistemaren portaeraren gaineko ulermen eta kontrola erabatekoa izaten da garatzaileen aldetik, erregelak moldatuz edota berriak gehituz portaera hau findu eta hautemaniko akatsak konpon ditzaketelarik. Edozelan ere, hizkuntzen anbiguotasuna eta berauek ulertzeko beharrezkoa den jakintza zabala erregela-sistemaren zurruntasunari kontrajartzen zaio hurbilpen honen baitan eta, honela, sistemaren estaldura eta zehaztasunean aurrera egin ahala erabili beharreko errepresentazioa nabarmenki konplexuagoa egiten da, beronen mantenua nahiz hobekuntza berriak egiteko aukera geroz eta zailagoa eginez. Hori dela eta,

egitura sinple eta argitasun handiko testuentzako emaitza onak eskuratu izan dira sistema hauen bidez, baina hitzez hitzeko esanahitik haratago doazen esamoldeekin nahiz testu korapilatsuagoekin buruhauste handiak erakutsi izan dituzte naturaltasun gutxiko itzulpenak sortuz, sarri maiz ulergaitzak, literalegiak, edota joskera baldarrekoak.

- **Adibideetan oinarrituriko itzulpengintza automatikoak (EBMT)** itzulpenak analogiaz egiten ditu, hau da, aldez aurretik eginak direnak eta corpus paralelo baten baitan ematen zaizkionak nolabait berrerabili eta elkarren artean uztartuz. Honela, itzuli beharreko testuaren eta corpusaren arteko bat-etortze maila altua denean, adibideetan oinarrituriko itzultzaile automatikoek emaitza oso onak eman ohi dituzte, berrerabilpen hau zentzuz egiteko gai baitira jatorrizko itzulpenen doitasun eta naturaltasunari eutsiz. Honekin batera, baina, sistema hauen estaldura eta orokortze-ahalmena oso eskasa izan ohi da eta, ondorioz, bat-etortze maila txikiagoa denean emaitza benetan kaskarrak eman ohi dituzte, corpuseko itzulpenak behar bezala berrerabiltzea ezinezkoa baitzaie.

- **Itzulpengintza automatiko estatistikoak (SMT)** corpus paralelo bateko aldez aurretiko itzulpenak ditu abiapuntutzat adibideetan oinarriturikoaren antzera baina, azken hau ez bezala, hauen baitan automatikoki ikasiriko eredu batean funtsatzen da itzulpenak egiteko, hurbilpen erabat estatistikoa jarraituz bide horretan. Era honetara, sistema estatistikoek corpuseko itzulpenak adibideetan oinarriturikoek baino modu malguagoan baliatzen dituzte, orokortze-ahalmen handiago bat eskuratuz. Horrekin batera, baina, corpuseko testua bera eredu-estatistikoen eraikuntzan erabat lausotzen da, jatorrizko itzulpen hauek doitasunez berrerabiltzeko aukera galduz. Erregeletan oinarrituriko itzultzaile automatikoekin alderatuz, berriz, itzultzaile estatistikoak hauen zurruntasunari aurre egin eta itzulpen naturalagoak emateko gai izan ohi dira, eraikiriko eredua eraginkortasunez aplikatzea posible bada beti ere. Izan ere, entrenamenduan erabiliriko corpus paraleloa itzuli beharreko testuari behar bezala egokitzen ez bazaio (beste domeinu batekoa, orokorregia ala zehatzegia delako, kasu), trinkotasun eta fidagarritasun gutxiko emaitzak eman ditzake, jatorrizkoaren esanahia erabat desitxura dezaketenak.

3.1 taulan RBMT eta SMT sistemen portaera akastunaren muturreko adibide batzuk ematen dira. Orain arte esanikoaren ildotik, RBMT sistemak *abrir mañana y tarde* nahiz *dejar en paz* bezalako esamoldeekin arazoak dituela ikus daiteke, itzulpen literalegiak proposatzen baititu. SMT sistemak, berriz, hitzez hitzeko itzulpenetik haratago joateko gaitasuna

Jatorrizkoa	**Lucy (RBMT)**	**Google translate (SMT)**
La tienda está abierta mañana y tarde.	Denda ireki dago bihar eta berandu.	Denda goiz eta arratsaldez irekita dago.
No me dejes aquí plantado.	Ez nazazu hemen landatuta utz.	Ez utzi ni hemen zutik.
El ciruelo de mi huerta.	Nire baratzeko aranondoa.	Inhar nire lorategian.
No te dejaré en paz.	Ez zaitut bakean utziko.	Ez dut bakarrik utzi.

3.1 Taula: RBMT eta SMT itzultzaileek eginiko akatsen muturreko adibideak

erakusten du, esamolde hauentzako itzulpen natural eta egokiagoak eginez. Aldi berean, baina, RBMT sistemak arazorik gabe itzul ditzakeen esaldi sinple bezain argiekin akats larri eta itxuraz ulertezinak egiten ditu jatorrizkoen esanahia erabat aldatuz (*Inhar nire lorategian* eta *ez dut bakarrik utzi*), eraikiriko eredu-estatistikoak huts egiten baitu kasu zehatz hauekin. Balizko EBMT sistema baten emaitzak, berriz, esaldi hauek corpus paraleloan izango luketen bat-etortzearen menpekoak lirateke erabat.

Gauzak honela, metodo bakoitzak bere sendotasun eta ahuleziak dituela esan daiteke, gainerakoek baino portaera hobe zein okerragoa erakutsi ohi duen erabilera-kasuekin. Ikuspegi historiko batetik begiratuz (Hutchins, 2007), itzulpen automatikorako hurbilpen klasikoa RBMT bidezkoa izan da, bere hastapenak 70. hamarkadan eman zituena Adimen Artifizialeko ezagutzan oinarrituriko sistemen harira. EBMT sistemen gaineko lehen ideiak, berriz, 80. hamarkada hasierakoak dira baina, euren orokortze-ahalmen kaskarraren kariaz, hurbilpen horrek ez du gaur-gaurkoz hedapen zabalegirik izan. SMT sistemek, berriz, geroz eta protagonismo handiagoa izan dute 90. hamarkadatik aurrera eta egun nagusi izatera iritsi dira, corpus paralelo erraldoiak eskuragarri egin ahala RBMT sistemen ageriko mugei aurre egiteko aukeratzat ikusi izan baitira. Oraindainoko emaitzarik onenak euren bidez eskuratu izan badira ere, baina, puntu batetik aurrera aurrerapauso esanguratsuak ematea oso zaila suertatzen ari da corpus handiagoak erabiliagatik, planteaturiko eredu-estatistikoak premisa sobera sinplifikatuetatik abiatzen baitira azken finean.

Metodo bakoitza bere muga propioekin topatu izan dela ikusirik, **hibridazio** ala metodo ezberdinen arteko nolabaiteko uztardurak dirudi egun etorkizuneko bidea. Ildo honetatik, ondorengo saiakerak egin izan dira azkenaldian (Lu and Xue, 2010):

- **Baliabideen aberasketa edo konbinazio hibridoak** sistema nagusi bat du oinarri, eta haren baliabideak bestelako metodoen bidez aberasten ditu. Planteamendu hau jarraituz, SMT sistemen baliabideak RBMT sistemen erregelen bidez egokitzeko ahaleginak egin dira, bai eta SMT bidezko teknikak erabiliz RBMT sistemak corpus paraleloen bidez aberastekoak ere.

- ***Multi-engine* edo konbinazio paraleloak** itzulpenak sistema ezberdinen bidez egi-

ten ditu modu independentean, eta modulu gehigarri bat horien irteera konbinatzeaz arduratzen da ahalik eta itzulpen egokiena sortzeko.

- ***Multi-pass* edo serieko konbinazioak** itzulpen-sistema baten irteera beste baten sarreratzat erabiltzen du. Hurbilpen honen erabilerarik ohikoena post-edizio automatikoaren bidezkoa da, non sistema bat beste baten irteera hobetzen ahalegintzen baita. Planteamendu hau jarraituz, SMT bidezko RBMT sistemen post-edizioak izan du gaur-gaurkoz oihartzunik handiena.

3.2 EBMT bidezko hibridazioaren ideia

Orain arte ikusi bezala, EBMT sistemen portaera itzuli beharreko testuaren eta baliaturiko corpus paraleloaren arteko bat-etortze mailak baldintzatzen du erabat:

- Itzuli beharreko testuaren eta baliaturiko corpus paraleloaren arteko bat-etortze maila handia denean, emaitzak oso onak dira (RBMT eta SMT sistemen gainetik kokatuz), corpuseko itzulpenak modu egokian berrerabil baitaitezke euren zehaztasun eta naturaltasunari eutsiz.

- Itzuli beharreko testuaren eta baliaturiko corpus paraleloaren arteko bat-etortze maila eskasa denean, berriz, emaitzak oso kaskarrak dira (RBMT eta SMT sistemenak baino nabarmenki okerragoak), corpuseko itzulpenak tentuz berrerabiltzea ezinezkoa suertatzen baita.

Bigarren puntu hau dela eta, EBMT sistemek ez dute oraindaino RBMT eta SMT sistemek izan duten hedapen eta indarrik izan. Izan ere, hurbilpen honek arrakastaz funtzionatzeko beharrezkoa den errepikakortasun-maila domeinu oso zehatzetako testuetan baino ez da ematen errealitatean, eta halakoek ere corpuseko itzulpenak zuzenean berrerabiliz modu egokian itzul ezin daitezken zatiak izan ohi dituzte. Hori dela eta, adibideetan oinarrituriko itzulpen automatikoaren erabilgarritasuna oso mugatua da bere baitan, RBMT eta SMT bidezko metodoak modu argian gailendu izan zaizkiolarik.

Hurbilpen hibridoen etorrerarekin, baina, EBMT bidezko metodoek oso bestelako zeresana izan dezakete, euren sendotasunei eutsi eta ahultasunak bestelako metodoen bidez estaltzea posible egiten baita. Honela, orain arteko hibridazio-saiakera gehienak RBMT

eta SMT sistemen artekoak izan badira ere, proiektu honetan EBMT tekniken bidezko hibridazioa izango da jorratuko dena. Planteamendu hau ondorengo behaketetatik abiatzen da:

- Bat-etortze indizea handia denean, EBMT sistema batek fidagarritasun handiko itzulpenak sortuko lituzke, RBMT nahiz SMT sistemek eman litzaketenak baino hobeak printzipioz. Honela, kasu hauetan EBMT bidez sorturiko itzulpenak erabiltzea komeniko litzateke.
- Bat-etortze indizea txikia denean, EBMT sistema batek fidagarritasun eskaseko itzulpenak sortuko lituzke, RBMT nahiz SMT sistemek eman litzaketenak baino okerragoak printzipioz. Halakoetan, bada, azken hauek sorturiko itzulpenak lirateke hobetsi beharrekoak.

Behaketa hauek ageriko bezain hutsalak badirudite ere, EBMT bidezko itzulpen-sistema hibridoak eraikitzeko oinarritzat har daitezke. Planteamendu honen baitan, corpusarekiko bat-etortze maila altua duten testu-zatiak EBMT sistemak itzuliko lituzke, eta gainerakoak, berriz, beste itzulpen-sistema baten esku geratuko lirateke. Bestela esanda, EBMT eta beste itzulpen-sistemaren arteko hautua lehenaren ziurtasun-mailaren menpe legoke erabat, testu-zati bakoitzean ebaluatu beharko litzatekeena corpusarekiko bat-etortze mailaren arabera.

Ikus daitekeenez, hibridazio-mekanismo hau edozein itzulpen-sistemaren gainean aplika liteke, kasuan kasu honako ekarpena eginez:

- Erregeletan oinarrituriko itzultzaile automatikoei dagokienez, hauen hurbilpen hertsiki arrazionala itzulpen errealak berrerabiliz osatzeko aukera lekarke. Puntu hau bereziki erabilgarria izan liteke RBMT sistemen ohiko hainbat arazori aurre egiteko. Batetik, sistema hauek modu trakets ala literalegian itzultzen dituzten esamoldeak behar bezala itzultzea posible egin lezake corpus paraleloan ageri diren bitartean, irteera egoki eta naturalagoa emanez. Bestetik, corpusetan oinarrituriko itzulpengintza automatikoak entrenamendu-corpusaren domeinu bereko testuekin lan egitean duen abantaila berdin lezake. Honela, itzulpen-sistema espezializatuak eraiki litezke helburu orokorreko RBMT sistemetatik abiatu eta espezializazio domeinuko corpus paraleloen bidez modu honetara hibridatuz.
- Itzultzaile automatiko estatistikoei dagokienez, euren ereduen eraikuntzan lausoturiko itzulpenak eraginkortasunez berrerabiltzeko aukera lekarke. Honen inpak-

tu erreala eskasa izan liteke esamolde laburren kasuan, eredu-estatistikoetako n-gramen baitan agertu eta behar bezala tratatuak izango direla espero baitaiteke. Zati errepikakor luzeagoei dagokienean, baina, hobekuntza aipagarria izango litzateke, eredu-estatistikoek ez baitituzte halakoak aintzakotzat hartzen maneiukortasunaren alde. Hau bereziki erabilgarria izan zitekeen administrazio-testuetan bezala esaldi ala esapide berberak behin eta berriz errepikaturik ageri diren kasuetarako.

Gauzak honela, RBMT sistemekiko hibridazioak SMT sistemekikoak baino hobekuntza argiagoak ematea espero litekeela ondoriozta daiteke. Izan ere, SMT eta EBMT oso gertuko hurbilpenak dira azken finean, corpusetan oinarrituriko itzulpengintza automatiko delakoaren baitan multzokatu ohi direnak. Honela, ulertzekoa da entrenamendu-corpus berbera erabiltzen duten bitartean biak hibridatuz lor litekeen hobekuntza-marjina planteamendu zeharo ezberdinetatik abiatzen diren RBMT sistemekiko espero litekeena bezainbestekoa ez izatea.

Amaitzeko, aipatzekoa da hibridazio-mekanismo hau bestelako sistema hibridoei aplikatzea ere posible izan litekeela, nagusi diren RBMT-SMT sistema konbinatuei kasu, erabiltzen duten metodoa erabiltzen dutela ere. Nolanahi ere, gaur-gaurkoz ikerketa-fase hutsean daudenez eta, neurri handi batean, antzeko asmoekin eginiko hibridazio-saiakerak izaki, proiektu honetan ez da aukera hori aztertuko.

3.3 Itzulpen partzialen erabilera hibridazio-estrategiatzat

Aurreko atalean EBMT bidezko itzulpen-sistema hibridoak eraikitzeko oinarriak eztabaidatu dira, erabakiak EBMT sistemaren beraren ziurtasun-mailaren esku uzten zituen arrazonamendu bat jarraituz. Atal honetan, planteamendu hori gauzatzeko EBMT bidezko itzulpen partzialen sorkuntzaz baliatuko litzatekeen hibridazio-estrategia bat aztertzen da.

Proposaturiko estrategia oso ideia sinpletik abiatzen da: erabili beharreko itzulpen-sistemaren hautua EBMT sistemaren beraren menpekoa denez, aintzakotzat hartu beharreko testu-zatiak soilik itzuliko ditu honek, ziurtasun-maila handiz trata ditzakeenak alegia, gainerako testua bere horretan utziz. Beste itzulpen-sistema, berriz, era honetara sorturiko itzulpen partziala hartu eta berau osatzeaz arduratuko da. Honenbestez, proiektuaren mamia testu-zati hauek ahalik eta modu egokienean identifikatu eta itzultzean egonen da. Bide horretan, honako puntuak hartu beharko dira gogoan:

- Proposaturiko sistemak benetako hobekuntza bat ekar dezan, EBMT bidez sorturiko itzulpen partzialak ahalik eta kalitate handienekoak izan beharko dira beste itzulpen-sistemak egingo lituzkeenen aldean. Izan ere, itzulpen partzial hauek bestela egingo liratekeenen parekoak ala okerragoak badira, hibridazioak ez du oinarri duen itzulpen-sistemaren portaera hobetzerik lortuko.

- Proposaturiko sistemak benetako inpaktu bat izan dezan, ahalik eta itzulpen partzial gehien sortu beharko dira. Izan ere, hibridazioaren erabilgarritasun erreala EBMT bidez itzuli ahalko den testu-portzentajearen araberakoa izango da, eta hau oso eskasa bada ez du bukaerako itzulpenetan eragin aipagarririk izango, oinarri duen itzulpen-sistemaren ia pareko emaitzak emanez.

Bi puntu hauek ez dira uztartzen errazak, itzulpen partzialen indizeak gora egin ahala haien kalitateak behera egitea espero baitaiteke oro har. Honenbestez, proiektuaren erronka nagusia bien arteko oreka bat aurkitzea izango da, sistema osoaren portaera optimizatzen saiatuz. Bide horretan, ondorengo urratsak aztertuko dira:

1. **Corpus paraleloko sarreren berrerabilpen zuzena.** Honen baitan, itzuli beharreko esaldi bat corpus paraleloan bere horretan ageri bada, bertako itzulpena emango da irteeratzat.

2. **Corpus paraleloko sarreren orokortzea entitateen bidez.** Honen baitan, corpus paraleloan entitateak ezagutu, jatorri eta helburu esalditakoen arteko loturak finkatu, eta halako beste entitateez modu askean ordezka zitezen onartuko litzateke. Hori behar bezala egin dadin, kasu batzuetan entitateak eurak ere itzuli egin beharko lirateke (herrialde-izenen kasuan hala nola), baina beste batzuetan ez legoke halako beharrik (pertsona-izen gehienen kasuan adibidez, pertsonaia historiko batzuen salbuespenaz). Entitateen inguruko xehetasun gehiagorako, ikus 2.1.4 atala.

3. **Corpus paraleloko sarreren orokortzea segmentuen azpiko unitate sintaktikoen bidez.** Honen baitan, esaldien azpiko itzulpen-erlazioak identifikatu eta berrerabiltzea posible izango litzateke.

Ikus daitekeenez, urrats hauek parekotasun handia dute hurrenez hurren lehen, bigarren eta hirugarren itzulpen-memorietan jarraitzen den hurbilpenarekin, 2.2.3 atalean hausnartzen zenarekin lotura estuan. Itzulpen-memoriekin bezala, halaber, urrats horietako bakoitzak orokortze handiago bat lekarke, itzulpen partzialen indizea handituz baina, era

berean, hauen kalitatea galtzeko arriskuarekin. Proiektu honetan, bada, hauetako bakoitza aurrera eramateko tekniken inguruan ikertu eta euren portaera esperimentalki neurtuko da, era honetara diseinaturiko metodoarentzat inplementazio funtzional eta eraginkor bat emanez.

4. KAPITULUA

Diseinua

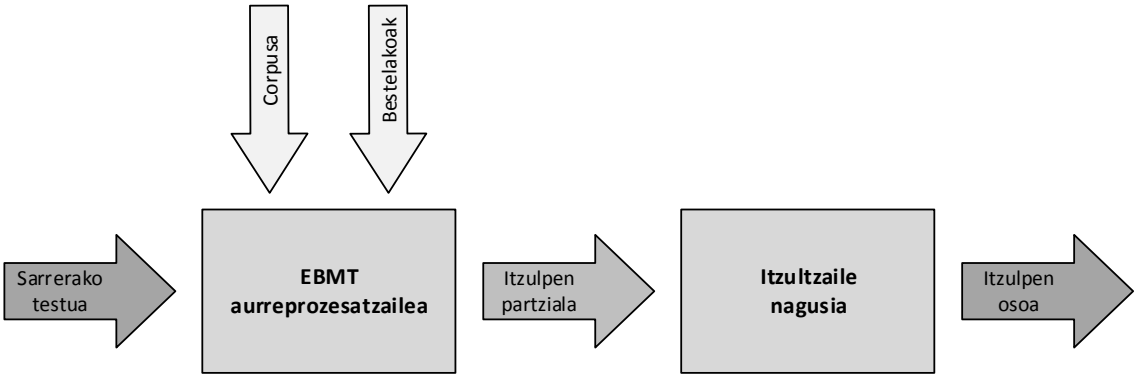
Kapitulu honetan proposatutako sistemaren diseinua azaltzen da. Horretarako, lehen atal batean bere arkitektura orokorra azaltzen da. Bigarren atalean, bere deskribapen funtzionala ematen da, ikuspegi funtzional batetik zer egiten duen zehazten duena. Amaitzeko, azken atal batean sistemaren deskribapen operatiboa egiten da, aurrekoa modu eraginkorrean gauzatzeko diseinaturiko datu-egitura eta algoritmoa azalduz.

4.1 Arkitektura orokorra

Analisiaren kapituluan hausnartu bezala, proiektu honetan EBMT bidezko itzulpen partzialen erabilera aztertzen da hibridazio-mekanismotzat. Hori aurrera eramateko, *multi-pass* edo konbinazio serialaren arkitektura jarraitzea deliberatu da, itzulpen partzialen sorkuntza aurreprozesu modura txertatzen duena. 4.1 irudiak arkitektura orokor hori erakusten du, ondorengo bi moduluek osatzen dutena bertan ikus daitekeen bezala:

- **EBMT aurreprozesatzailea**, jatorrizko testua hartu eta bere itzulpen partzial bat sortzeaz arduratzen dena corpus paraleloa eta bestelako baliabideak erabiliz.
- **Itzultzaile nagusia**, EBMT aurreprozesatzailearen itzulpen partziala hartu, osatu, eta irteerako testua sortzeaz arduratzen dena.

Bien arteko lotura burutzeko moduaren arabera, EBMT aurreprozesatzailearen bi erabilera-kasu posible aurreikusten dira:



4.1 Irudia: Proposatutako sistemaren arkitektura orokorra

- **Aplikazio independente modura**. Aplikazio bezala erabiltzean, EBMT aurreprozesatzaileak itzulpen partzialak XML formatu berezi batean emango lituzke, jatorrizko testuaz gain haren zati ezberdintzat sortutako itzulpenak zehaztuko lituzkeena. Itzultzaile nagusiak XML hori hartuko luke sarrera modura, eta bertako testua itzuli emandako itzulpen partzialak aintzakotzat hartuz.

- **Liburutegi modura**. Liburutegi bezala erabilita aurreprozesuaren kontrol handiago bat eskuratuko litzateke eta, horrekin batera, bertan erabiltzen diren baliabide eta abstrakzioak berrerabiltzeko aukera emango luke. Horri esker, EBMT aurreprozesua itzultzaile nagusiarekin modu estuagoan integra zitekeen eta, horretaz gain, oinarri beraren gainean bestelako aplikazioak garatzeko aukera ematen du. Era honetara, adibidez, proposaturiko sisteman oinarritutako itzulpen-memorien kudeatzaile aurreratu bat eraiki zitekeen, bigarren eta hirugarren belaunaldikoen lerro berean.

Arkitektura honek analisiaren atalean planteaturiko lerro nagusiak arazorik gabe gauzatzeko aukera ematen du, eta sinplea, argia eta erabat modularra suertatzen da. Modulartasun horri esker, proposaturiko sistema oso hedagarria egiten da, modu errazean integra daitekeelarik itzulpen-sistema ezberdinetan, bai eta aplikazio berriak sortzeko oinarri bezala erabili ere.

4.2 Sistemaren deskribapen funtzionala

Ikuspegi funtzional batetik, proposaturiko sistema ondoko hiru faseetan banatzen da:

- **Entrenamendua**, esaldi-mailan lerrokaturiko corpus paralelo bat hartu eta EBMT

bidez itzulpen partzialak sortzeko erabilgarri izan dadin prozesatzeaz arduratzen dena.

- **EBMT aurreprozesua**, itzuli beharreko testua hartu eta entrenamendu-fasean sorturiko baliabideen bidez haren itzulpen partzial bat sortzeaz arduratzen dena, ezagunak zaizkion zatien itzulpenak eman eta ezezagunak zaizkionak bere horretan utziz.
- **Integrazioa**, EBMT aurreprozesuan sorturiko testu partzialki itzulia hartu eta itzultzaile nagusiaren bidez hura osatzeaz arduratzen dena.

Bistakoa denez, fase horiek modu sekuentzialean egikaritu behar dira, euretako bakoitza aurrekoen menpekoa baita. Honela, entrenamendu-faseak ondorengoetarako beharrezkoak diren baliabideak prestatzen ditu, eta behin baino ez da burutu behar corpus elebidun bakoitzeko. EBMT aurreprozesua eta integrazioa, berriz, sistemaren arkitektura orokorrean azalduriko oinarrizko bi moduluei dagozkie.

Ondorengo orrialdeetan, bada, fase hauetako bakoitzaren funtzionamendua zehazten da azpiatal banatan.

4.2.1 Entrenamendua

Entrenamendu-fasearen eginkizuna esaldi-mailan lerrokaturiko corpus paralelo bat hartu eta ondorengo faseetan erabilgarri izan dadin prozesatzea da. Bi urratsetan egiten da hau:

- **Corpus elebakarren prozesaketa**, hizkuntza bakoitzeko sarrerak modu independentean analizatzeaz arduratzen dena.
- **Lerrokatzea**, aurrez azterturiko corpus paraleloko sarrera bakoitzeko abiapuntuko hizkuntzako eta xede hizkuntzako esaldien arteko loturak finkatzeaz arduratzen dena.

Jarraian, bi urrats horiek azaltzen dira bakoitza bere aldetik.

Corpus elebakarren prozesaketa

Corpus elebakarren prozesaketan hizkuntza bakoitzeko sarrerak banan-banan aztertzen dira 2.1 atalean azalduriko tresnak erabiliz. Eginiko analisia ondorengo puntuetara mugatzen da:

- **Tokenizazioa**, 2.1.1 atalean azaldu bezala esaldia tokenetan banatuz. Proposaturiko sisteman entitate-izenak, hainbat hitzez osaturikoak barne, token bakartzat hartzen dira beti (hau da, *Barack Obama* edo *Etxe Zuria* bezalakoek, adibidez, token bat osatuko lukete). Gainontzean, baina, ez da *multiword* adierazpenen trataerarik egiten (hau da, *lehen ministro* edo *adarra jo* bezalakoak, hala nola, bina token izango lirateke).

- **Entitate-izenen ezagutzea eta sailkapena**, 2.1.4 atalean azaldurikoaren baitan pertsona-izen bereziak, erakunde-izen bereziak, leku-izen bereziak, bestelako izen bereziak eta zenbaki dezimalak hauteman eta dagokien kategoria esleituz. Kasuan kasu baliaturiko analizatzaileak bestelako entitate motaren bat ezagutuz gero (datak, orduak, monetak...), ez lirateke aintzakotzat hartuko, token arrunt gisara utziz. Aurreko puntuan zehazten zenaren harira erabiliriko analizatzaileak entitate-izen jakin bat hainbat tokenez osatua dagoela ebazten duenetan, berriz, haiek guztiek token bakar bat osa dezaten moldatuko litzateke analisia. Adibidez, analizatzaileak *Etxe Zuria* entitate-izena *Etxe* eta *Zuria* token bikoteak osatzen duela erabakiagatik, *Etxe Zuria* bere horretan token bat bakarra izan dadin egokituko litzateke analisia.

- **Entitate-izenen analisi morfologikoa eta etiketatzea**, 2.1.2 eta 2.1.3 ataletan azaldurikoaren arabera aurreko puntuaren baitan erauziriko entitate-izenen azaleko formatik kasuan kasu dagokien forma lexikala eskuratuz. Forma lexikaltzat entitate-izenaren lema eta hari atxikiriko informazio gramatikala kodetzen duten etiketak hartzen dira, eta eurak ateratzeko moduaren arabera ondoko bi kasuen artean bereizi da:

  - **Morfologiaren tratamendu berezi bat eskatzen duten hizkuntzetan** nola lema hala informazio gramatikala analizatzaileak ematen dituenak dira. Analizatzaileak entitate-izena token bakar batez osaturik dagoela erabakiz gero, zuzen-zuzenean har daitezke hauek. Aurreko puntuan aipatzen zenaren harira entitate-izena hainbat tokenez osaturik dagoela ebatzi eta analisia guztia token bakar bat izan dadin eskuz egokitzen denetan, baina, hurbilpen honek ez du balio, analizatzaileak jatorrian hark ezaguturiko tokenen forma lexikala bakoitza bere aldetik emango bailuke. Halakoetan azkenak emaniko informazio gramatikala izango da bere horretan ontzat emango dena. Lema osatzeko, berriz, azkenarentzat emaniko lema eta gainerakoen azaleko forma kateatuko dira '_' azpimarra-ikur batez bereizirik. Proiektu honetan euskara bakarrik da, bere morfologia aberatsa medio, multzo honetan sartu dena.

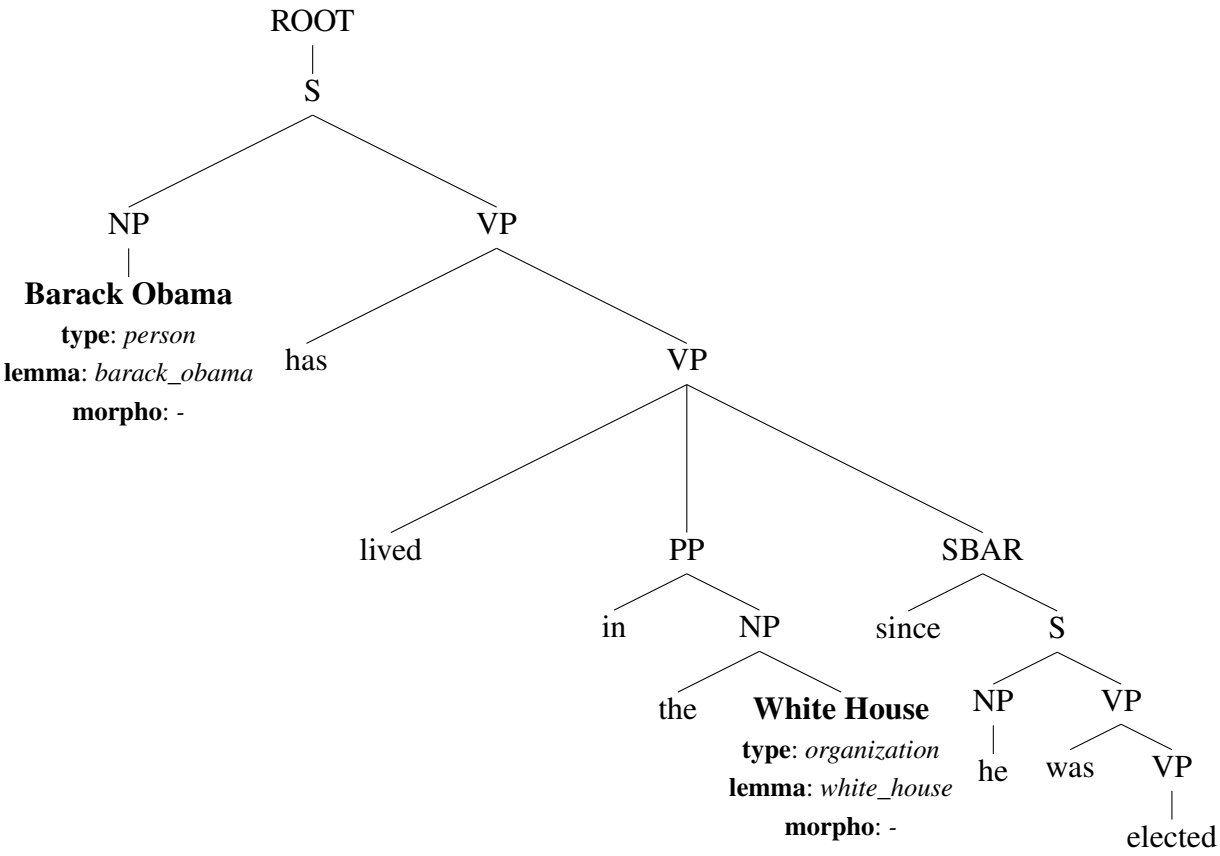
  - **Morfologiaren tratamendu berezirik behar ez duten hizkuntzetan** lema modura azaleko forma hartzen da zuriuneak '_' azpimarra-ikurrez ordezkatuta, eta informazio gramatikala adierazteko ez da etiketa bat bera ere erabiltzen. Bistakoa denez, inolako analisi morfologikorik ez egitearen parekoa da hau, lema bakoitzaren kasu posible ezberdinak (singularra eta plurala, adibidez), entitate ezberdin modura hartuko lituzkeena. Diseinu ikuspegi batetik, baina, sistemaren uniformetasuna eskuratzen da honela, morfologiaren tratamendu berezi bat eskatu ala ez ondorengo urratsetan hizkuntza guztiekin modu berean jokatzeko aukera emanez. Eta, hurbilpen sobera sinplea izanagatik, morfologia murritzeko hizkuntzekin lan egiteko aukera eraginkorra suertatzen da hau, aurrekoa baino egokiagoa izan daitekeena bere arintasunarengatik. Proiektu honetan, hain zuzen ere, gaztelania eta ingelesarekin hala jokatzea erabaki da.

- **Analisi sintaktikoa**, 2.1.5 atalean azaldurikoaren arabera esaldiaren zuhaitz sintaktikoa eraikiz. Bertan zehaztu bezala, azaleko analisi sintaktikoa baino egiteko gai ez diren analizatzaileen kasuan, haien emaitza zuhaitz sintaktiko modura interpretatzen da, non *chunk*ak maila bakarreko barne-adabegiak izango bailirateke. Horretaz gain, zentzuzkoa den bezala tokenizazioa eta analisi sintaktikoa bateragarriak izan behar dira beti ere, hau da, zuhaitz sintaktikoko hostoak tokenak behar dira izan. Entitateen aferaren kariaz hau hala ez denetan, tokenizazioarekin egiten ziren antzeko egokitzapenak egiten dira, zuhaitz sintaktikoa behar den puntuan moztuz entitatea hosto modura atera dadin. Bukatzeko, barne-nodoak unitate sintaktiko esanguratsuak izan daitezen, analizatzaileek token bakoitzaren gainean euren kategoria gramatikala edo *part of speech* delakoa adieraziz sortzen dituzten nodoak zuhaitz sintaktikotik ezabatu egiten dira.

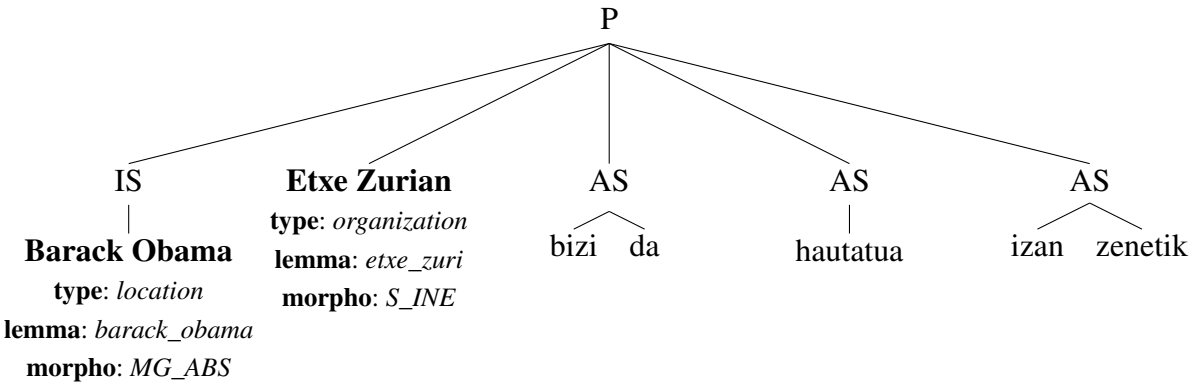
4.2 irudian corpus elebakarren prozesaketaren adibide bat erakusten da ingelesa-euskara bikoteko *Barack Obama has lived in the White House since he was elected - Barack Obama Etxe Zurian bizi da hautatua izan zenetik* balizko sarrerarentzat. Ingeleserako Stanford CoreNLP analizatzailea izan da erabili dena eta euskararako, berriz, Eustagger, kasuan kasu azaldu berri diren egokitzapenak eginez haien analisitik abiatuta esaldi prozesatua eskuratzeko.

Lerrokatzea

Behin corpus elebakarren prozesaketa burututa, entrenamenduaren bigarren eta azken urratsa lerrokatzeari dagokio. Lerrokatze-prozesuaren bidez, corpus elebiduneko sarre-



(a) Ingelesezko esaldi prozesatua Stanford CoreNLPren analisitik abiatuta



(b) Euskarazko esaldi prozesatua Eustaggerren analisitik abiatuta (analisiak akatsak ditu)

4.2 Irudia: Corpus elebakarren prozesaketaren adibide bat ingelesa-euskara bikoteko balizko sarrera batentzat

ra bakoitzeko abiapuntu eta xede hizkuntzako esaldien arteko loturak ezartzen dira, bakoitzaren zein zati zehazki zeinen itzulpena den identifikatuz. Lerrokatze hau, halaber, ondorengo bi pausutan burutzen da:

- **Hitz-lerrokatzea**, 2.3 atalean azaldurikoaren harira corpus elebiduneko sarrera bakoitzeko hitzez hitzeko itzulpen-erlazioak zehazten dituena. Zehazki, hitz-lerrokatzaileari corpus elebakarren prozesaketaren baitan tokenizaturiko testua ematen zaio sarreratzat, hizki xehez (esaldi hasieretako hizki larriek zaratarik sor ez dezaten) eta entitateen lekuan euren lema jarriz *enti_* aurrizkiarekin (entitate bakoitza token berezi bat bezala trata dadin). Irteera gisa, berriz, honako bi emaitzak hartzen dira:

  - **Hitzen arteko loturak**, hau da, hitzen arteko lerrokatzeari berari dagokion grafo bipartigarriaren ertzak. Bestela esanda, abiapuntuko hizkuntzako hitz bat eta xede hizkuntzako beste bat lotuta egongo dira baldin eta soilik baldin lerrokatzaileak bien arteko itzulpen-erlazio bat dagoela ebatzi badu. Honela, corpus elebiduneko sarrera bakoitzeko hitzen arteko loturak A multzo bat izango dira, non $(i, j) \in A$ izango baita baldin eta soilik baldin abiapuntuko hizkuntzako i. tokenaren eta xede hizkuntzako j. tokenaren arteko itzulpen-erlazio bat badago sarrera jakin horretan.
  - **Pisu lexikalak** edo tokenen itzulpen-probabilitateak bi zentzuetan (abiapuntuko hizkuntzatik xede hizkuntzara eta xede hizkuntzatik abiapuntuko hizkuntzara). Honela, f abiapuntuko hizkuntzako token bat eta e xede hizkuntzako beste bat emanda, pisu lexikala $p(e|f)$ e tokena f tokenaren itzulpena izan dadin probabilitateari dagokio zentzu batean, eta $p(f|e)$ f tokena e tokenaren itzulpena izan dadin probabilitateari aurkako zentzuan. Beste hizkuntzan baliokide zuzenik ez izateko probabilitateak adierazi ahal izateko, berriz, corpusean ageri diren berezko tokenez gain *null* sasi-tokena erabiltzen da. Modu honetara, $p(null|f)$ abiapuntuko hizkuntzako f tokena itzulpen batean baliokide zuzenik gabe geratzeko (edo, nolabait esatearren, itzulpenean *desagertzeko*) probabilitateari dagokio, eta $p(f|null)$, berriz, itzultzean baliokide zuzenik gabe geratzen diren abiapuntuko hizkuntzako tokenen artean f agertzeko probabilitateari. Antzeko eran, $p(null|e)$ eta $p(e|null)$ ere definitzen dira xede hizkuntzako tokenentzako.

- **Entitate-lerrokatzea**, entitate-izenen arteko itzulpen-erlazioak zehazten dituena. Ikusi berri denez, hitz-lerrokatzeak token guzti-guztien arteko itzulpen-erlazioak

finkatzen ditu, entitate-izenenak barne. Nolanahi ere, entitateak corpus elebiduneko itzulpenak orokortzeko mekanismotzat erabiliko direnez, haiek lerrokatzeko ziurtasun-maila altu bat izatea komeniko da. Hori dela eta, entitateen lerrokatze espezifiko bat egiten da hiztegi elebidunak erabiliz hitz-lerrokatzetik modu erabat independentean. Honela, bi entitate lerroka daitezen euren lemak elkarren itzulpentzat jasota egon beharko dira ondorengoren batean:

- **Zuzenean emandako hiztegi elebidunak**. 2.2.1 atalean azaldu bezala, abiapuntu eta xede hizkuntzetako hitzen baliokidetzak bilduz aurrez sorturiko hiztegiak erabili ahalko dira, erregeletan oinarrituriko itzultzaile automatikoek darabiltzatenetatik erauziak hala nola.
- **Hitz-lerrokatzetik erauzitako hiztegiak**. Hitz-lerrokatzearen irteeratzat jasoriko pisu lexikalak abiapuntutzat hartuz eta gutxieneko eskakizun batzuk finkatuz entitateen hiztegi bat eraikiko da. Zehazki, hiztegi honi $f - e$ entitate bikotea erantsiko zaio baldin eta soilik baldin $\frac{p(e|f)+p(f|e)}{2} < \theta$ bada eta f nahiz e tokenek k agerpen badituzte gutxienez corpusean. Proiektu honetan $\theta = 0.5$ eta $k = 10$ hartu dira.
- **Wikipedia**. 2.2.2 atalean azaldurikoaren harira, Wikipedia hiztegi eleanitz bat bailitzan erabiltzen da bere *dump*etatik abiatuta, bertako artikuluek hizkuntza ezberdinetan dituzten izenburuak elkarren itzulpentzat hartuz eta berbideraketa posible guztiak jarraituz. Honela, *United States* izenburudun ingelesezko artikuluari *Ameriketako Estatu Batuak* deituriko euskarazkoa badagokio, ingelesezko *USA* sarrerak *United States* artikulura berbideratzen badu eta euskarazko *AEB* sarrerak *Ameriketako Estatu Batuak* artikulura berbideratzen badu, adibidez, *united_states* - *ameriketako_estatu_batuak*, *united_states* - *aeb*, *us* - *ameriketako_estatu_batuak* eta *us* - *aeb* bikoteak elkarren itzulpentzat onartuko lirateke.
- **Zenbakien lerrokatzea**. Erabiliriko adierazpide dezimala edozein dela ere, zenbaki berbera adierazten duten entitateak lerrokatu egiten dira.
- **Hizkuntza bietan berdinak diren entitateak**. Lema berbera duten entitateak elkarren itzulpentzat onartzen dira orain arteko hiztegiren batean agertu ala ez. Pertsona- eta leku-izen berezi gehienak eta, batez ere, bereziki garrantzitsu edo ezagunak ez direnak (hain justu ere, aurreko hiztegiren batean agertzeko aukera gutxien dutenak) tratatzeko nahikoa suertatzen da hau, oro har ez baitira aldatzen hizkuntza batetik bestera itzultzean. Era berean, neurri honek sor

lezakeen zarata erabat mespretxagarria da, esaldi bikote berean elkarren itzulpena izan gabe hizkuntza bietan lema berbera duten entitateekin topo egitea kasik pentsaezina baita. Honela, *Urdaneta* edo *Aitzol Arruti* izen bereziak, adibidez, bere horretan mantenduko lirateke euskaratik ingelesera itzultzean eta, orain arte ikusiriko hiztegietan sarrera hauek topatzeko aukerak benetan txikiak izanik, irizpide honek biak lerroka daitezen ahalbidetuko luke. Bistakoa da, halaber, halako lerrokatzeak nekez izan litezkeela okerrak.

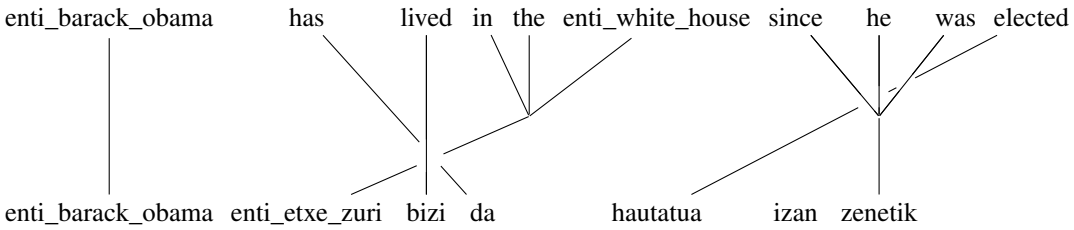
Entitate bakoitza gehienez ere beste batekin bakarrik lerrokatzea onartzen da. Honela, abiapuntuko hizkuntzako entitateak euren agerpen-ordenaren arabera prozesatzen dira, lerrokaturik ez dagoen eta azaldu berri den hiztegi edo irizpideren batek jasoriko xede hizkuntzako lehen entitatearekin lerrokatuz. Arestian aipatu bezala, entitateen lemari baino ez zaio erreparatzen lerrokatze hau egiteko. Kategoria, honenbestez, ez da aintzakotzat hartzen zenbakien kasuan salbu, entitateen sailkapenean akats asko egiten direla ikusi izan baita (4.2b irudiko analisia bera, kasu, halako errore baten adibide da). Prozedura honen bidez baliokiderik topatu ezin zaien entitateak, berriz, lerrokatu gabe uzten dira, aurrerantzean token arrunt bezala tratatuz. Horri esker, elkarren itzulpentzat hartzeko nahikoa berme eskaintzen duten entitateak baino ez dira lerrokatzen, entitateen bidezko orokortzeak zarata sortzeko arriskua minimizatuz.

4.3 irudiak lerrokatzearen adibide bat erakusten du 4.2 irudiko sarrera berberarentzat. Ikus daitekeenez, eta orain arte azaldurikoa jarraituz, lerrokatzearen irteera lautan banatzen da: hitzen arteko loturak eta entitateen lerrokatzea esaldi bikote mailan (hau da, corpus paraleloko sarrera bakoitzeko halako bana izango litzateke), eta bi zentzuetako pisu lexikalak corpus mailan (hau da, corpus paralelo osoarentzat taula bakar bat izango litzateke zentzu bakoitzean).

4.2.2 EBMT aurreprozesua

EBMT aurreprozesuaren eginkizuna entrenamendu-fasean prozesaturiko corpus elebidunaz baliatuz sarrerako testua partzialki itzultzea da, ziurtasun-maila altuz egin daitezkeen zatien itzulpenak eman eta gainerakoa itzultzaile nagusiak osa dezan utziz. Honako lau urratsetan egiten da hori:

- **Analisia**, sarrerako testua entrenamendu-fasean corpus elebakarrekin egiten zen bezala aztertzen duena.



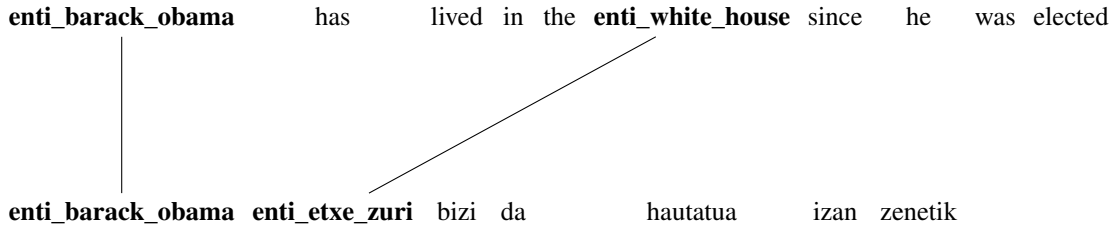
(a) Hitzen arteko loturak, non $A = \{(1,1),(2,4),(3,3),(4,2),(5,2),(6,2),(7,7),(8,7),(9,7),(10,5)\}$

$p(enti\_barack\_obama \mid enti\_barack\_obama)$	$= 1.000$
$p(da \mid has)$	$= 0.243$
$p(bizi \mid lived)$	$= 0.541$
$p(enti\_etxe\_zuri \mid in)$	$= 0.002$
$p(enti\_etxe\_zuri \mid the)$	$= 0.001$
$p(enti\_etxe\_zuri \mid enti\_white\_house)$	$= 1.000$
$p(izan \mid null)$	$= 0.024$
$\vdots$	

(b) Ingelesetik euskararako pisu lexikalak (corpus mailan)

$p(enti\_barack\_obama \mid enti\_barack\_obama)$	$= 1.000$
$p(in \mid enti\_etxe\_zuri)$	$= 0.087$
$p(the \mid enti\_etxe\_zuri)$	$= 0.421$
$p(enti\_white\_house \mid enti\_etxe\_zuri)$	$= 0.421$
$p(has \mid da)$	$= 0.217$
$p(lived \mid bizi)$	$= 0.453$
$p(null \mid izan)$	$= 0.128$
$\vdots$	

(c) Euskaratik ingeleserako pisu lexikalak (corpus mailan)



(d) Entitateen lerrokatzea, hiztegiren batean *white_house* - *etxe_zuri* sarrera ageri dela suposatuz

4.3 Irudia: Lerrokatzearen adibide bat ingelesa-euskara bikoteko balizko sarrera batentzat

- **Bilaketa**, corpus elebidunean itzuli beharreko testu analizatuaren bat-etortzeak bilatzen dituena.
- **Itzulpena eta iragazketa**, topaturiko bat-etortze bakoitzari dagokion itzulpena identifikatu eta balekoak ez direnak baztertzen dituena.
- **Hautaketa**, bat-etortze bakoitzarentzat sorturiko itzulpen posible guztien artean egokiena aukeratzen duena.

Jarraian, urrats hauek puntuz puntu azaltzen dira.

Analisia

Analisian sarrerako testuko esaldi bakoitza 2.1 atalean azalduriko tresnen bidez aztertzen da. Jarraituriko prozesua entrenamendu-fasean corpus elebakarren prozesaketarentzat azaldurikoaren berbera da, eta tokenizazioak, entitate-izenen ezagutze eta sailkapenak, entitate-izenen analisi morfologiko eta etiketatzeak, eta analisi sintaktikoak osatzen dute.

Horien aurretik, baina, urrats gehigarri bat ematea beharrezkoa izan daiteke, 2.1.1 atalean azalduriko esaldi-segmentazioa hain zuzen ere. Izan ere, entrenamendu-fasean corpus elebiduna esaldi-mailan lerrokaturik dago, eta hori dela eta esaldi-segmentazioa ez da beharrezkoa (alde batetik, corpusa berez esalditan banatuta dagoelako eta, bestetik, urrats gehigarri gisa bi hizkuntzatan esaldi-segmentazioa modu independentean burutuz gero sor litezkeen aldeek elkarren arteko lerrokatzea hauts lezaketelako). Itzuli beharreko testua esaldiz esaldi ematen ez denetan, baina, analisiaren baitan esaldi-segmentazioa ere egin behar da.

Bilaketa

Behin sarrerako testuaren analisia burututa, hurrengo urratsa corpus elebidunean haren bat-etortzeak bilatzean datza. Bilaketa hori honako irizpideak jarraituz burutzen da:

1. **Esaldiaren egitura sintaktikoa errespetatzea**. Irizpide honen arabera, bilaketa esaldiaren egitura sintaktikoa hautsi gabe beharko da egin, hau da, bilatu beharreko testu-zatiak unitate sintaktikoak beharko dira izan. Zehazki, zuhaitz sintaktikoko barne-adabegi solteak nahiz anai-arrebak izanik elkarren segidan dauden barne-adabegien multzoak baino ez dira onartuko.

2. **Gutxieneko token kopurua**. Bistakoa denez, testu-zati laburren itzulpen tribialak ematea mesedegarri beharrean kaltegarria izan liteke. Izan ere, itzultzaile nagusia, berez, zati hauek behar bezala tratatzeko gai izan beharko litzateke (ingelesezko *house* euskarazko *etxe* dela eta antzerakoak arazorik gabe jakin beharko lituzke), eta itzulpen partzialak iradokitzeak eragin negatiboa izan lezake honenbestez balizko integrazio gorabeherak medio (euskarazko deklinabideari lotuta, adibidez). Hori gutxi balitz, testu-zati laburrekin lerrokatze-arazoak eman ohi dira hizkuntzen arteko desberdintasunek eraginda, haientzako itzulpen partzial desegokiak sortzea eragin lezakeena.[1] Hori dela eta, irizpide honen arabera bilaketa-prozesua gutxieneko token kopuru bat duten testu-zatietara mugatzen da. Zehazki, proiektu honetan lau tokeneko mugarekin egin da lan. Salbuespen modura, murriztapen hau ez zaio aplikatzen erro-adabegiari, hau da, esaldi osoa beti bilatuko da gutxieneko token kopurua izan ala ez, esaldi osoen kasuan ez baita hizpide zen integrazio arazorik ematen.

3. **Bilaketa-prozesu hierarkikoa**. Orain arte azalduriko bi irizpideek zein testu-zati bilatzea onartzen den eta zein ez zehazten zuten. Aurrerago sakonduko den bezala, hurrengo urratsetan modu honetara bilatu eta bat-etortzeak dituzten testu-zatiak bakoitza bere aldetik itzultzen funtzionatzen du sistemak. Testu-zati bakoitzaren itzulpena besteekiko modu independentean egiten denez, euren arteko gainjartzeen arazoari nolabaiteko irtenbide bat eman behar zaio, eta bilaketa-prozesuan bertan lehentasunak ezarriz egiten da hori hizpide den irizpide honen bidez. Zehatzagoak izanez, lehentasunak mailaz maila ezartzen dira, zuhaitz sintaktikoan maila altuagoan (hau da, errotik gertuago) dauden adabegi eta adabegi-segidak lehenetsiz. Maila berean dauden adabegi eta adabegi-segiden artean, berriz, lehentasunak token kopuruaren arabera ezartzen dira, token gehiago dituztenak lehenetsiz. Maila berean egonik token kopuru berbera dutenen artean, azkenik, ez da inolako lehentasunik ezartzen, aukeraketa modu arbitrarioan eginez. Behin lehentasunak ezarrita, testu-zati bat lehentasun handiagoko beste batekin gainjartzen delarik azken horrek bat-etortzeak baditu eta haien bidez itzulia izan bada, lehen testu-zatia erabat baztertzen da bilaketa-prozesuan. Aipatzekoa da, beti ere, lehentasunak ez direla zertan bilaketa ordenarekin bat etorri behar. Are gehiago, gainjartzen ez diren zatien artean ziurra da ez dela gatazkarik egongo eta, honenbestez, edozein hurrenkeran sakon li-

[1] Adibide modura, ingelesezko *we* euskarazko *dugu* hitzarekin lotzen zela ikusi zen. Honek badu bere zentzua, ingelesez pertsonaren informazioa subjektuak ematen baitu eta ez aditzak, eta euskaraz berriz aditz laguntzaileak pertsona ere adierazten duenez subjektua eliditzeko joera baitago. Erabat okerra dela ezin esan badaiteke ere, baina, portaera hau ez da, inola ere, lan honetan bilatzen dena.

teke euretan lehentasunari erreparatu ere egin gabe.

4. **Egitura sintaktikoaren erabateko bat-etortzearen eskakizuna**. Irizpide honen baitan, bilaturiko testu-zati bat corpuseko beste batekin bat datorrela onartzeko ez da nahikoa izango token-segida berberaz osatuta egotea. Izan ere, beharrezkoa izango da zuhaitz sintaktikoan eurei dagokien azpi-zuhaitza ere (lehen puntuan azaldu den abiapuntuko barne-adabegi ala barne-adabegien segida eta haien azpiko guztiguztia) berdina izatea.

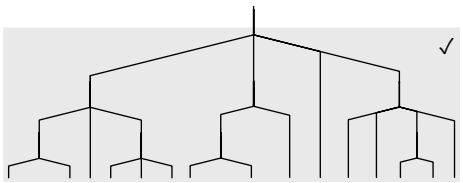
5. **Corpus elebiduneko entitate lerrokatuen orokortzea**. Irizpide honen arabera, corpus elebiduneko entitate lerrokatuak sarrerako testuan ezaguturiko edozein entitaterekin bat datozela onartuko da. Lerrokatu gabeko entitateak, berriz, token arrunten modura tratatuko dira, eta sarrerako testuko entitateren batekin bat datozela onar dadin euren testua bere horretan berdina beharko da izan.

Ikus daitekeenez, lehengo bi irizpideek bilatu beharreko testu-zatien gaineko murriztapenak zehazten dituzte, hirugarrenak murriztapenak betetzen dituztenen arteko lehentasunak finkatu, eta azken biek bat-etortzeek bete beharreko baldintzak ezarri. 1 algoritmoan irizpide hauek guztiak jarraitzen dituen bilaketa-funtzio bat aurkezten da. Bi adabegi-segida bat ote datozen egiaztatzeko 2 algoritmoko funtzioaz baliatzen da bertan, era berean bi adabegi bat ote datozen egiaztatzeko 3 algoritmoko funtzioan oinarritzen dena.

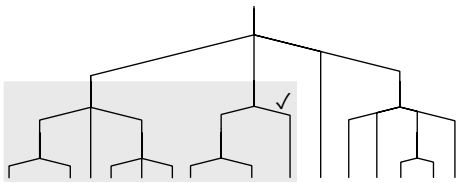
4.4 irudian, azkenik, lehen bi baldintzen arabera bilaketa-prozesuan onargarri eta baztergarriak liratekeen testu-zatien adibideak biltzen dira. Lehentasunak, berriz, testu-zatien abiapuntu-adabegien mailaren eta, berdinketa kasuan, dagozkien token kopuruaren arabera ezartzen dira hirugarren irizpideak zehaztu bezala. Honenbestez, 4.4a kasuan arrakastarik izanez gero, hala nola, 4.4b eta 4.4c multzoak zuzen-zuzenean baztertuko lirateke, lehena maila altuagoan baitago (izan ere, erro-adabegia bera da). 4.4b eta 4.4c multzoak maila berdinean egonik, berriz, lehentasuna lehenak izango luke, token gehiago hartzen baititu beregain. Haren bilaketa arrakastatsua izanez gero, honenbestez, bigarrena zuzenean baztertuko litzateke.

Itzulpena eta iragazketa

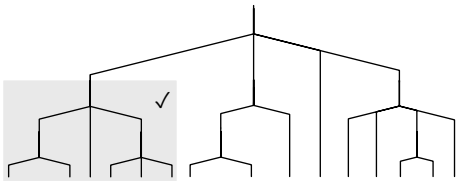
Bilaketaren baitan testu-zati jakin baten bat-etortze guztiak eskuratuta, hurrengo urratsa bat-etortze hauetako bakoitzari dagokion itzulpena sortzean datza, baleko itzulpenik ez dutenak baztertuz. Ondorengo bi pausutan egiten da hori:



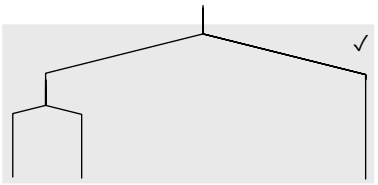
(a) Erro-adabegia



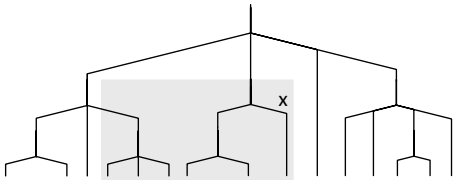
(b) Barne-adabegien segida



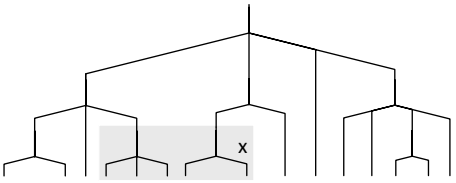
(c) Barne-adabegi soltea



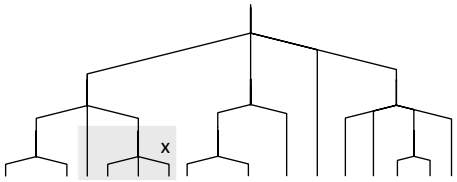
(d) Token kopuru minimorik ez erroarentzat



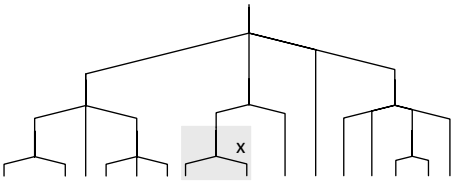
(e) Maila ezberdineko adabegiak



(f) Adabegiak ez dira anai-arrebak



(g) Token solte bat hartu da



(h) Token kopuru minimoa ez da betetzen

4.4 Irudia: Bilaketa-prozesuan onargarri nahiz baztergarriak liratekeen zatien adibideak

Algoritmo 1 Bilaketa-funtzioa (lehenengo deia erro-adabegiarekin beharko da egin)

function BILAKETA($adabegia, corpus$)
  $semeak \leftarrow$ SEMEAK($adabegia$)
  $zatiak \leftarrow adabegia \cup \{z :$ AZPIZERRENDA_DA$(z, semeak)\}$
  while $zatiak \neq \emptyset$ **do**
    $zatia \leftarrow z \in zatiak : \forall z' \in zatiak,$ ZENBAT_TOKEN$(z) \geq$ ZENBAT_TOKEN(z')
    $erroa\_da \leftarrow |zatia| = 1 \wedge$ ERROA_DA$(zatia[1])$
    $tokenak\_ditu \leftarrow \exists x \in zatia :$ TOKENA_DA(x)
    $laburregia \leftarrow$ ZENBAT_HOSTO_TOKEN$(zatia) <$ token_kopuru_minimoa
    $itzulpenekin\_teilakatua \leftarrow \exists adabegi \in zatia :$ ITZULIA_IZAN_DA$(adabegi)$
    if $(erroa\_da \vee \neg laburregia) \wedge \neg tokenak\_ditu \wedge \neg itzulpenekin\_teilakatua$ **then**
      $match \leftarrow \{adabegi\_segida \in corpus :$ BAT_DATOZ$(zatia, adabegi_segida)\}$
      if $match \neq \emptyset$ **then**
        ITZULI$(zatia, match)$
      end if
    end if
  end while
  for all $semea \in semeak$ **do**
    if BARNE_ADABEGIA_DA$(semea) \wedge \neg$ITZULIA_IZAN_DA$(semea)$ **then**
      BILAKETA$(semea, corpus)$
    end if
  end for
end function

1. **Corpus elebidunean itzulpenari dagokion token-segida identifikatzea**. Corpuseko n. sarreran abiapuntuko hizkuntzako i. tokenetik j. tokenera doan bat-etortze bati dagokion itzulpentzat sarrera berean xede hizkuntzako i'. tokenetik j'. tokenera doan eta honako baldintzak betetzen dituen token-segida laburrena hartzen da:

   - **Gutxienez token eta lotura bat izatea**. Baldintza honen bidez itzulpen hutsak nahiz jatorrizko testu-zatiarekin inolako erlaziorik ez dutenak ematea saihesten da. Hori hala izan dadin, honakoa bete beharko da:

     $$\exists (a, a') \in A_n : i \leq a \leq j \wedge i' \leq a' \leq j'$$

   - **Bat-etortzea eta dagokion itzulpena erabat lerrokatuta egotea**, hau da, zati bakoitzetik ateratzen den lotura oro beste zatira joatea. Hau hala izan dadin, honako baldintza bete beharko da:

     $$\forall (a, a') \in A_n : (i \leq a \leq j \wedge i' \leq a' \leq j') \vee ((a < i \vee a > j) \wedge (a' < i' \vee a' > j'))$$

Algoritmo 2 Adabegi-segidak bat datozen egiaztatzeko funtzioa

```
function BAT_DATOZ(adabegi_segida1[], adabegi_segida2[])
    n ← TAMAINA(adabegi_segida1)
    m ← TAMAINA(adabegi_segida2)
    if n ≠ m then
        return False
    end if
    for i ← 1..n do
        if ¬ADABEGIAK_BAT_DATOZ(adabegi_segida1[i], adabegi_segida2[i]) then
            return False
        end if
    end for
    return True
end function
```

Algoritmo 3 Adabegi solteak bat datozen egiaztatzeko funtzioa

```
function ADABEGIAK_BAT_DATOZ(adabegi1, adabegi2)
    if BARNE_ADABEGIA_DA(adabegi1) ∧ BARNE_ADABEGIA_DA(adabegi2) then
        etiketa1 ← ETIKETA(adabegi1)
        etiketa2 ← ETIKETA(adabegi2)
        semeak1 ← SEMEAK(adabegi1)
        semeak2 ← SEMEAK(adabegi2)
        return etiketa1 = etiketa2 ∧ BAT _DATOZ(semeak1, semeak2)
    else if HOSTOA_DA(adabegi1) ∧ HOSTOA_DA(adabegi2) then
        berdinak ← TESTU_BERDINA(adabegi1, adabegi2)
        entitateak ← ENTITATEA_DA(adabegi1) ∧ ENTITATE_LERROKATUA_DA(adabegi2)
        return berdinak ∨ entitateak
    else
        return False
    end if
end function
```

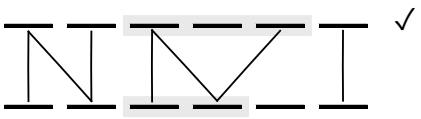
- **Esaldiaren egitura sintaktikoa errespetatzea**, hau da, zuhaitz sintaktikoan anai-arrebak diren adabegietatik datozen hostoak izatea. Bestela esanda, anai-arrebak diren alboz alboko adabegi-segida bat existitu behar da zuhaitz sintaktikoan zeinetatik datozen token guztien segida i'tik j'ra doan berbera baita. Hau bilaketa-prozesuan ezartzen zen murriztapenarekin bat dator betebetean ezberdintasun bakar batekin: oraingoan ez da beharrezkoa adabegi guzti-guztiak barne-adabegiak izatea, hau da, hostoak ere izan ahalko dira gainerako baldintzak betetzen diren bitartean. Bilaketa-prozesuan baztertu beharreko 4.4g irudiko moduko kasuak, honenbestez, erabat onargarriak lirateke itzulpen modura. Horretaz gain, aurreko puntuan zehaztu bezala gutxieneko token kopurua batekoa da itzulpenen kasuan eta, hori dela eta, 4.4h modukoak ere onartu egingo lirateke itzulpen bezala. 4.4 irudiko gainerako multzo baztergarriak, berriz, itzulpen modura ere ez lirateke onartuko.

Hiru baldintza hauek betetzen dituen token-segidarik existitzen ez bada, berriz, batetortzea erabat baztertzen da, eta ez da harentzat inolako itzulpenik sortzen. Zorroztasun honen bidez, bukaerako emaitzan eragin negatiboa izan lezaketen ziurtasun-maila baxuko itzulpenak iragaztea da asmoa.

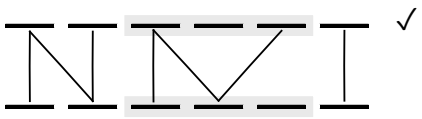
4.5 irudian lehen bi baldintzen arabera onargarri nahiz baztergarriak liratekeen segiden adibideak ematen dira. Aipatzekoa da sinpletasunaren alde token-segida laburrak erabili direla, ez dutenak beti bilaketa-prozesuan ezarritako token kopuru minimoaren murriztapena betetzen, baina helburu honetarako erabat baliagarriak direnak. Azken baldintzari dagokionez, xede hizkuntzako esaldiaren zuhaitz sintaktikoari erreparatu beharko litzaioke, eta murriztapena betetzen duen segida laburrenaren alde egin halakorik bada. Honela, 4.5a eta 4.5b kasuek biek ere baldintza hura betez gero lehentasuna aurrenekoak izango luke, token gutxiago baititu. Murriztapena bigarrenak soilik betez gero, berriz, hura izango litzateke hautatua.

2. **Entitateen itzulpena**. Bilaketa-prozesua entitateak orokortuz egiten denez, batetortzeko entitate lerrokatuak jatorrizko esaldikoen ezberdinak izan daitezke, eta beste horrenbeste gertatzen da, ondorioz, aurreko pausuan itzulpentzat identifikaturiko token-segidarekin. Hurrengo eginkizuna, beraz, token-segida hartako entitate lerrokatu bakoitza itzultzean datza, eta honela egiten da hori:

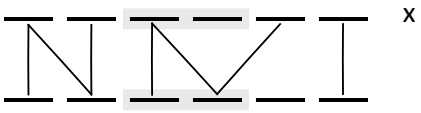
   (a) **Entitate lerrokatuari jatorrizko testuan dagokiona identifikatu**. Lehendabizi, entrenamendu-faseko entitateen lerrokatzearen arabera xede hizkuntzako entitate lerrokatua abiapuntuko hizkuntzako zeini lotuta dagoen identifikatu



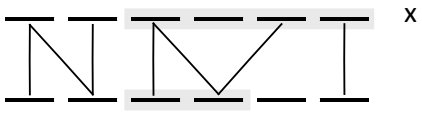
(a) Zuzena (lehentasun handiena)



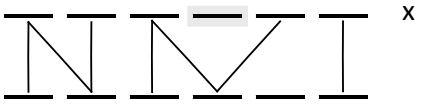
(b) Zuzena (**a** baino lehentasun txikiagoa)



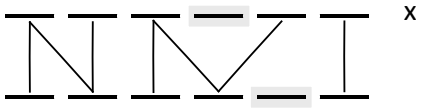
(c) $(5,4)$ lotura arazotsua



(d) $(6,6)$ lotura arazotsua



(e) Itzulpenak tokenik ez



(f) Bi segiden artean loturarik ez

4.5 Irudia: Lerrokatzearen arabera itzulpen prozesuan onargarri nahiz baztergarriak liratekeen zatien adibideak

behar da eta azken hau, berriz, jatorrizko testuan itzuli beharreko zein entitateren bat-etortzea den.

(b) **Jatorrizko entitatearen lema itzuli**. Behin jatorrizko testuko entitatea identifikatuta, haren lema itzuli behar da jarraian. Entrenamendu-fasean entitateak lerrokatzeko baliabide berdin-berdinak erabiltzen dira horretarako, bai zuzenean emandako hiztegi elebidunak, bai hitz-lerrokatzetik erauzitakoak, bai Wikipedia, bai eta zenbakien tratamendu berezia ere. Itzuli beharreko entitatea haietan topatzen ez bada, berriz, lema bere horretan uzten da. Entitateen lerrokatzean hizkuntza bietan berdinak ziren entitateekin egiten zenaren lerroan kokatzen da irizpide hau, eta hizkuntzatik hizkuntzara aldatzen ez diren pertsona- nahiz leku-izen berezi gehienak tratatzeko neurri egokia da.

(c) **Lema itzuliaren sorkuntza egin**. Amaitzeko, entitatearen lema itzulia eta xede hizkuntzako entitatearen informazio gramatikala kodetzeko etiketak uztartuz entitatearen azaleko forma sortu behar da 2.1.2 atalean azaldu bezala. Entrenamendu-faseko corpus elebakarren prozesaketan bi jokabide aurreikusten dira entitate-izenen analisi morfologiko eta etiketatzeari dagokionez, hizkuntza bakoitzaren ezaugarri morfologikoen arabera morfologiaren tratamendu berezi bat egin nahiz azaleko forma eta lema berdin mantendu eta informazio gramatikala adierazteko etiketa bat bera ere ez erabiliz. Sorkuntza prozesua, bada, horren araberakoa izango da, azken kasuan azaleko formatzat jatorrizko entitatearen lema itzulia bera utziz. Beti ere, baina, lemen '_' azpimarra-ikurrak zuriuneez ordezkatu beharko dira.

4.2 eta 4.3 irudietako adibidearekin jarraituz, demagun *François Hollande has lived in the Élisée Palace since he was elected* esaldia dela itzuli beharrekoa. Entitateen orokortzearen baitan *Barack Obama has lived in the White House since he was elected* haren bat-etortze bat izango litzateke, zeinaren itzulpentzat *enti_barack_obama enti_etxe_zuri bizi da hautatua izan zenetik* token-segida hartuko bailitzateke. Azken honek bi entitate lerrokatu ditu eta lehen egitekoa, honenbestez, jatorrizko esaldian euretako bakoitzari dagokiona identifikatzea izango da, kasu honetan jatorrizko esaldiko *François Hollande* entitatea itzulpeneko *Barack Obama* entitatearekin eta jatorrizko *Élisée Palace* entitatea itzulpeneko *Etxe Zurian* entitatearekin lotuz. Hurrengo egitekoa jatorrizko entitateen lemak itzultzea da, kasu honetan *françois_hollande* lema bere horretan mantenduko litzatekeelarik (izan nolabaiteko hiztegiren batean agertu delako ala izan portaera lehenetsiaren ondorioz), eta *élisée_palace*, berriz, *eliseo_jauregi* bilakatu hiztegiren batean halako sarrera-

rik ageri bada bederen. Azkenik, lema itzuli hauetako bakoitzaren sorkuntza egingo litzateke, *françois_hollande* lemak eta absolutibo mugagabearen adierazle diren *MG_ABS* etiketek *françois hollande* ematen dutelarik eta *eliseo_jauregi* lemak eta inesibo singularraren adierazle diren *S_INE* etiketek, berriz, *eliseo jauregian*. Era honetara, *françois hollande eliseo jauregian bizi da hautatua izan zenetik* itzulpena eskuratuko litzateke bat-etortze hartatik abiatuta.

Hautaketa

Ikusi berri den bezala, aurreko urratsean bat-etortze bakoitzeko itzulpen posible bat sortzen da, balekoak ez direnak iragaziz behar izanez gero. Honenbestez, laugarren eta azken urrats honen sarrera testu-zati bakoitzeko haren itzulpen posibleen hautagai-multzo bat da. Bertan egin beharrekoa, bada, hautagai-multzo honen araberakoa da, ondorengo hiru kasuen artean bereiziz:

- **Hautagai bat bera ere ez izatea**. Aurreko urratseko iragazketaren baitan bat-etortze guzti-guztiak baztertuak izan badira, ezin izango da eskuartean den testu-zatiarentzat inolako itzulpenik sortu. Halakoetan, bilaketa-prozesuak testu-zati horrek bat-etortzerik izan ez balu bezala jarraitzen du aurrera.

- **Hautagai ezberdin bakar bat izatea**. Itzulpen posible guzti-guztiak berdinak direnean hautaketa ongi erraza da, eta aukera bakar hori iradokitzen da uneko testu-zatiaren itzulpentzat.

- **Hainbat hautagai ezberdin izatea**. Halako kasuetan itzulpen posibleen artean bat aukeratzen da honako irizpideak jarraituz:

  1. **Gehien errepikatzen dena**. Lehen irizpide honen baitan itzulpen posible ezberdin bakoitza zenbat bat-etortzeren bidez sortu den zenbatzen da, eta errepikapen gehien dituena aukeratu.
  2. **Pisu lexikal altuena duena**. Lehen irizpidea aplikatu ostean errepikapen kopuru bera duten hautagai bat baino gehiago geratuz gero, bigarren irizpide honek pisu lexikal altuena duenaren alde egiten du. Corpus elebiduneko n. sarreran abiapuntuko hizkuntzako i. tokenetik j. tokenera doan testu-zatiaren itzulpentzat xede hizkuntzako i'. tokenetik j'. tokenera doana hartu bada, s_k abiapuntuko hizkuntzako k. tokena izanik eta s'_k xede hizkuntzako k. tokena,

hari dagokion pisu lexikala honako adierazpenaren bidez kalkulatzen da:

$$\sqrt[j-i+1]{\prod_{a=i}^{j} \max\left(\{p(null|s_a)\} \cup \{p(s'_{a'}|s_a) : (a,a') \in A_n\}\right)} \times \sqrt[j'-i'+1]{\prod_{a'=i'}^{j'} \max\left(\{p(null|s'_{a'})\} \cup \{p(s_a|s'_{a'}) : (a,a') \in A_n\}\right)}$$

Ikus daitekeenez, pisu lexikalak hizkuntza bietako token bakoitzaren loturek emandako itzulpen-probabilitate handienak biderkatuz kalkulatzen dira. Probabilitate hauek zero eta baten arteko balioak izaki sekuentzia luze eta laburren artean dagoen desoreka berdintzeko, berriz, biderketen erroa hartzen da hizkuntza bakoitzeko token kopuruaren arabera.

Esan beharrik ere ez da hautagai ezberdin bakoitzeko nola jatorrizko testuzatia hala dagokion itzulpena, definizioz, berdina izango dela bat-etortze guztientzat. Nolanahi ere, arraroa izanagatik berez posible da elkarren arteko lerrokatzean ezberdintasunak egotea bat-etortze haien artean eta, honenbestez, euren bidez kalkulaturiko pisu lexikalak ez dute zertan berdinak izan behar. Gauzak honela, hautagai batek kalkuluan erabiliriko bat-etortzearen arabera pisu lexikal ezberdinak har ditzakeenean guztien artean handiena izango da ontzat emango dena.

3. **Hautaketa arbitrarioa**. Aurreko irizpideak aplikatu ostean hautagai posible bat baino gehiago geratzen bada, euretako edozeinen alde egiten da modu arbitrarioan. Aipatzekoa da, hala ere, berez posible izanagatik oso arraroa litzatekeela egoera hau.

4.2.3 Integrazioa

Behin EBMT aurreprozesuaren bidez testu partzialki itzulia eskuratu ostean, hirugarren eta azken fasea itzultzaile nagusiaren bidez hura osatzea ahalbidetuko duen integrazioari dagokio. Esan beharrik ere ez da integrazioa itzultzaile nagusiaren menpe dagoela erabat eta, zentzu honetan, bi kasu nagusi bereiz daitezke:

- **Itzultzaile nagusiak itzulpen partzialak iradokitzeko berezko euskarria eskaintzea**. Bere funtzionalitatearen baitan itzultzaile nagusiak itzulpen partzialak

emateko aukera eskaintzen badu, EBMT aurreprozesuaren irteera horretarako darabilen formatura egokitu eta zuzen-zuzenean berari pasatu baino ez da egin beharko. Era honetara, aurreko fasearen baitan iradokitako itzulpen partzialak bukaerako itzulpenean txertatzearen ardura itzultzaile nagusiaren beraren esku geratzen da. Bistakoa denez, posible denetan hauxe da aukerarik onena ezbairik gabe, itzultzaile nagusiak egokien deritzen irizpideak jarraitu ahalko baititu kasuan kasu iradokiriko itzulpen partziala baztertu, behar den posizioan txertatu, edota gainerako itzulpena haren arabera moldatzeko.

- **Itzultzaile nagusiak itzulpen partzialak iradokitzeko aukerarik ez eskaintzea**. Aurreko puntuan azalduriko hurbilpena posible ez denean, itzultzaile nagusiari jatorrizko testua oso-osorik itzularazten zaio (partzialki itzuliak izan diren zatiak barne), EBMT aurreprozesuaren baitan jada itzuliak izan diren zatiei dagozkien itzulpen atalak identifikatu ondoren, eta aurrez sorturiko itzulpen haiekin ordezkatu azkenik. Horretarako, itzultzaile nagusiaren sarrera eta irteeran itzulpen partzialen hasiera eta bukaera markatzeko trikimailuren bat erabili behar da, itzulpenean dagokien posizioan mantenduko diren markatzaile modukoak txertatuz jatorrizko testuan. Proiektu honetan jarraituriko hurbilpena, zehazki, testu osoarekin HTML dokumentu bat sortu, itzulpen partzialen mugak etiketa berezi batzuekin markatu, eta postprozesu baten bidez testu itzulian etiketa haien artean aurkitzen dena EBMT aurreprozesuaren baitan eskuraturiko itzulpenekin ordezkatzea izan da.

4.3 Sistemaren deskribapen operatiboa

Aurreko atalean proposaturiko sistemak ikuspegi funtzional batetik zer egiten zuen azaltzen bazen, oraingo honetan hori bera modu eraginkor eta eskalagarrian gauzatzeko bidea lantzen da. Horretarako, lehen azpiatal batean azpikate-bilaketaren problema eta atzizki-taulak aurkezten dira, proposaturiko soluzioa horietan oinarritzen baita. Bigarren azpiatal batean, hain justu ere EBMT aurreprozesuan atzizki-taulak erabiltzeko aukeraren inguruan hausnartzen da. Honen ostean, horretarako diseinaturiko datu-egitura zehazten da. Amaitzeko, azken azpiatal batean aurreko guztian oinarrituz garatutako itzulpen-algoritmoa azaltzen da.

4.3.1 Azpikate-bilaketaren problema eta atzizki-taulak

Azpikate-bilaketa delakoa algoritmiako problema klasikoa bat da, n luzerako kate bat eta m luzerako honen balizko azpikate bat emanda, bigarrenak aurrenekoan dituen agerpenak aurkitzean datzana. Problema hau ebazteko proposatu diren algoritmoak ondorengo lau multzoetan sailka daitezke (Melichar et al., 2005):

- **Inolako aurreprozesaketarik egiten ez dutenak**. Multzo honetan problemaren soluzio tribialei dagozkien oinarri-oinarrizko algoritmoak aurkitzen dira, ulertu eta inplementatzen errazak baina eraginkortasun kaskarrekoak. Adibide agerikoena *naïve search* edo indar gorriaren bidezkoa da, azpikatearen bilaketa elementuz elementu burutzen duena katearen indize guztietatik hasita. Hurbilpen zehatz honek, baina $O(nm)$ denbora hartzen du kasurik okerrenean.

- **Azpikatea aurreprozesatzen dutenak**. Multzo honetan bilatu beharreko azpikatea aurreprozesatuz nolabaiteko datu-egitura bat eraikitzen duten algoritmoak aurkitzen dira, oro har automataren bat errepresentatzen duena eta kate nagusiko elementu bakoitza behin eta birritan trata ez dadin baliatzen dena. Honen adibide ezagun bat Knuth-Morris-Pratt ala KMP algoritmoa da (Knuth et al., 1977), azpikatea $O(m)$ denboran aurreprozesatuz bilaketak $O(n)$ denboran burutzeko gai dena.

- **Katea aurreprozesatzen dutenak**. Multzo honetako algoritmoek aurrekoen antzeko ideia jarraitzen dute, baina azpikatea beharrean katea bera aurreprozesatuz. Bide horretan, bi hurbilpen jarraitu izan dira nagusiki: erabiltzeko oso eraginkorrak baina eraikitzeko konplexuak suertatu ohi diren automaten bidezkoak batetik, eta atzizki-zuhaitzen nahiz aurrerago hizpide izango diren atzizki-taulen moduko indize-metodoak bestetik.

- **Katea eta azpikatea aurreprozesatzen dituztenak**. Multzo honetan sinadura bidezko bilaketa-algoritmoak aurkitzen dira, bai eta aurreko bien ildo bereko automatak darabiltzatenak ere.

Aukera posible guztien artean, proiektu honetan atzizki-taulak edo *suffix array* direlakoak erabiltzea erabaki da (Manber and Myers, 1990). Atzizki-taulen aurkikuntza dezente berria da, eta Udi Manber eta Gene Myersek 1990ean aurkeztu zituztenetik gaurdaino aurrerapauso handiak ematen ari dira haien gaineko ezagutzan, eurekin lan egiteko algoritmo berri eta eraginkorragoak proposatu izan direlarik. Informatika teorikoaren ikuspegitik

gaurkotasun handiko zerbait izateaz gain, hizkuntzaren prozesamenduaren arloan ere euren gaineko interes eta erabilerak gora egin du nabarmen azkenaldian, testu-masa handiak modu eraginkorrean prozesatzeko ezin hobeak suertatzen baitira. Izan ere, atzizki-taulek honako abantailak dituzte bestelako aukeren aurrean:

- **Eraginkortasuna**. Aurrerago ikusiko den bezala, kasu arruntenetan atzizki-taulak $O(n)$ denboran eraiki daitezke katea aurreprozesatuz, eta behin hori eginda edozein azpikateren bilaketa $O(m)$ denboran egin daiteke. Bistakoa denez, kate berberaren gainean hainbat bilaketa egin behar direnean soluzio asintotikoki optimoa suertatzen da hau. Aztergai den problema, hizkuntzaren prozesamenduaren alorreko beste hainbat bezala, kasu honi dagokio hain justu ere, corpus jakin bat emanda testu-kate berriek bertan dituzten agerpenak bilatzea baitu helburu.

- **Trinkotasuna**. Atzizki-taulen memoria-eskakizunak oso baxuak suertatzen dira bestelako indize-metodoen aldean. Izan ere, atzizki-zuhaitzen moduko hurbilpenek datu-egitura konplexuagoak erabiltzen dituzte, memoria-espazio askoz handiagoa hartzen dutenak. Puntu honek berebiziko garrantzia du aztergai den kasuan eta, oro har, hizkuntzaren prozesamenduaren esparruan, testu-masa oso handiekin egiten baita lan.

- **Maneiukortasuna**. Arestian esaten zenaren ildotik, bestelako metodoek baino egitura sinpleagoak erabiltzen dituztela-eta atzizki-taulak haiek baino maneiagarriagoak ere badira. Izan ere, sakontasun handiko gaia suertatzen da informatika teorikoaren ikuspegitik, interes eta gaurkotasun nabarmenekoa eta sofistikazio maila altuko soluzioak jaso izan dituena, baina oinarri-oinarrizko kontzeptuak oso argiak egiten dira beti ere.

Gaiari helduz, bada, *suffix array* edo atzizki-taula bat kate bateko atzizki ordenatuen indizeak biltzen dituen datu-egitura bat da. Zehazki, $S = s_1, s_2, ..., s_n$ kate bat emanda eta $S[i, j]$ bere i eta j arteko azpikatea izanik, A atzizki-taula honako baldintza betetzen duen osokoen taula bat izango da:

$$1 < i \leq n : S[A[i-1], n] < S[A[i], n]$$

4.6 irudiak atzizki-taulen adibide bat erakusten du 4.6a taulako $S = patata$ karaktere-katearentzat. Ikus daitekeenez, 4.6c taulan karaktere-kate horren atzizki posible guztiak biltzen dira euren hasiera indizearen arabera ordenaturik, eta 4.6d taulak, berriz, atzizki

i	1	2	3	4	5	6
S[i]	p	a	t	a	t	a

(a) *patata* karaktere-katea

i	1	2	3	4	5	6
A[i]	6	4	2	1	5	3

(b) *patata* katearen atzizki-taula

Atzizkia	**i**
patata	1
atata	2
tata	3
ata	4
ta	5
a	6

(c) *patata* katearen atzizkiak

Atzizkia	**i**
a	6
ata	4
atata	2
patata	1
ta	5
tata	3

(d) *patata* katearen atzizki ordenatuak

4.6 Irudia: Atzizki-taulen adibide bat

berberak jasotzen ditu ordena lexikografikoan. Eta azken hauen hasiera indizeei dagozkio, hain zuzen ere, *patata* katearen 4.6b atzizki-taula.

n luzerako kate bat emanda, m luzerako bere azpikate bat $O(m \log n)$ denboran aurki daiteke lehenaren atzizki katearen bidez bilaketa bitar bat burutuz. Euren jatorrizko lanean bertan, Manber eta Myersek muga hau $O(m + \log n)$ denboraraino hobetzeko modua proposatu zuten LCP taula (*longest common prefix array* edo aurrizki komun luzeenaren taula) deritzon egitura gehigarria erabiliz. Era berean, Abouelhoda, Kurtz eta Ohlebuschek muga hau are gehiago hobetzen zuen metodo bat aurkeztu zuten 2004ean (Abouelhoda et al., 2004), taula gehigarriak erabiliz bilaketa $O(m)$ denboran burutzeko gai zena.

Bilaketarena ez ezik, atzizki-taulen eraikuntzaren problema ere ikerlan askoren helburua izan da, eta hau ebazteko soluzioen konplexutasun teorikoan nahiz portaera esperimentalean aurrerapauso nabarmenak eman dira urteetan zehar. Problema hau, halaber, erabilitako alfabetoaren menpe dago erabat, ondorengo kasuen artean bereiz daitekeelarik (Kim et al., 2003):

- **Alfabeto konstanteak**, hau da, $\Sigma = [0, c]$ modura $|\Sigma| \in O(1)$ dutenak. Hasierako soluzioek $O(n \log n)$ denbora hartzen bazuten ere, gerora $O(n)$ kostuko algoritmoak ere proposatu izan dira, bistakoa denez asintotikoki optimoak suertatzen direnak. Horietako batzuk zeharkako algoritmoak dira, lehendabizi atzizki-zuhaitza sortu eta honetan oinarrituz atzizki-taula eraikitzen dutenak, baina soluzio zuzenak ere ezagutzen dira. Ildo honetatik, bada, exekuzio-denborari ez ezik beharrezko memoria-espazio gehigarriari ere arreta handia eskaini izan zaio, eta bi alderdietan eraginkorrak diren soluzioak diseinatzeko ahaleginak egin izan dira.

- **Osoko alfabetoak**, hau da, $\Sigma = [0, n^c]$ modura $|\Sigma| \in n^{O(1)}$ dutenak. Kasu hau aurrekoaren orokortze bat da, zailtasun nabarmenki handiagokoa eta ikerketa-eremu oso aktiboa izan dena. Halakoentzat ere $O(n)$ kostuko algoritmoak aurkitu izan dira, zeharkakoak ez ezik zuzenak ere bai (Kärkkäinen and Sanders, 2003), bistakoa denez asintotikoki optimoak suertatzen direnak berriro.
- **Alfabeto orokorrak**, hau da, euren osagaiak denbora konstantean konpara daitezkeela baino onartzen ez dutenak. Halako kasuetan behe-bornea $O(n \log n)$ dela frogatuta dago, eta denbora hori eskuratzen duten algoritmoak ezagutzen dira.

4.3.2 Atzizki-taulak EBMT aurreprozesuan

Algoritmiaren ikuspegitik, 4.2.2 atalean azalduriko EBMT aurreprozesuaren muina bilaketa-prozesuan, itzulpenetako token-segiden identifikazioan, entitate lerrokatuak eskuratzean eta pisu lexikalen kalkuluan aurkitzen da. Horien artean, bilaketa jotzen da diseinu-elementu zentraltzat, corpus paraleloaren tamainak zuzen-zuzenean eragiten dion heinean eskalagarritasunaren ikuspegitik alderdi kritikoena izateaz gain, gainerakoak horretarako baliaturiko prozeduraren eta sorturiko irteeraren menpe egongo baitira. Hori dela eta, sistemaren diseinu operatiboaren erronka nagusia bilaketa ahalik eta modu eraginkorrenean gauzatzea da, hurrengo urratsak ere eraginkortasunez burutzea ahalbidetzen duen modu batean beti ere.

Algoritmiako problema klasikoen artean, proiektu honetako bilaketa-prozesutik hurbilen dagoena 4.3.1 atalean azalduriko azpikate-bilaketaren problemarena da, azken finean itzuli beharreko zatiek corpus paraleloan dituzten bat-etortzeak identifikatzean baitatza. Nolanahi ere, jarraian zehazten diren berezitasunak medio, problema hori ez da zuzenzuzenean aplikagarria kasu honetan:

- **Oinarrizko unitatea tokena da, eta ez karakterea.** Azpikate-bilaketaren aplikazio ohikoena karaktere-kateei dagokie. Proiektu honetan ere bilaketa testuaren eta, honenbestez, karaktere-kateen gainean egin behar da, baina oinarrizko unitatea ez da karakterea bera. Izan ere, hala balitz *hartz*, adibidez, *behartzen* katearen azpikate bat izango litzateke, baina hori ez da, inola ere, kasu honetan nahi dena. Horren ordez, egitura hierarkiko batekin egin behar da lan, oinarrizko unitate bezala tokenak dituena eta horien gainean sintagma eta esaldiak.
- **Bilaketa ezin daiteke literala izan.** Azpikateen bilaketaren problemak, berez, patroiarekin osagaiz osagai bat datozen sekuentziak bilatzen ditu katean. Eskuartean

den probleman, baina, bilaketa ezin daiteke testuarekiko erabat literala izan ondorengo arrazoiak direla eta:

  - **Entitateen orokortzea.** Corpus elebiduneko entitate lerrokatuek beste edozein entitaterekin bat etorri beharko lukete, baina lerrokatu gabeek, berriz, token modura berdinak direnekin baino ez. Patroiaren bilaketan, bada, bi aukerak aurreikusi behar dira.
  - **Murriztapenak.** Bilaketa-prozesuan bat-etortze guztiak ez dira onargarriak, hainbat murriztapen bete behar baitira. Besteak beste, behar-beharrezkoa da token-segida berdina izateaz gain haien egitura sintaktikoa ere bat etortzea.

- **Bat-etortzeen hasiera-indizeaz gain informazio gehigarria behar da.** Azpikate-bilaketaren problemaren irteera, berez, bat-etortzeen hasiera-indizeak baino ez dira. Kasu honetan, baina, bat-etortze bakoitzeko informazio gehiago behar da, bere hitz-lerrokatzea eta entitate-lerrokatzea adibidez.

Zailtasun horiei aurre egin eta atzizki-taulen bidez EBMT aurreprozesua modu eraginkorrean burutzeko, bada, datu-egitura berezi bat diseinatu da corpus elebidun prozesatuarentzat, 4.3.3 atalean azaltzen dena. 4.3.4 atalean, azkenik, datu-egitura hori darabilen itzulpen-algoritmoa azaltzen da, bilaketa eta ondorengo urratsak beregain hartzen dituena.

4.3.3 Erabilitako datu-egitura

Corpus elebiduna errepresentatzeko diseinatutako datu-egituran hainbat alderdi bereiz daitezke. Lehen tokian zuhaitz sintaktikoa kate modura errepresentatzeko hurbilpena aipatu behar da, azpikate-bilaketaren problema aplikagarria izan dadin ezinbestekoa dena. Hori oinarri hartuta, datu-egitura elkarloturiko bi osagaitan bereizten da, identitate-taula eta eduki-taula deitu zaiena. Ondorengo lerroetan, puntuz puntu alderdi horietako bakoitza jorratzen da.

Zuhaitz sintaktikoaren errepresentazioa kate modura

Zuhaitz sintaktikoa, zuhaitz oro bezala, egitura hierarkiko bat da, maila ezberdinetako osagaiz eratua. Azpikate-bilaketaren probleman, baina, kateekin egiten da lan, maila berean dauden osagaien segidei dagozkienak. Hori dela eta, corpus paraleloarentzako datu-egituraren diseinuan zuhaitz sintaktikoa kate modura errepresentatzea da lehen erronka.

Horretarako, zuhaitzeko elementuak ondoren zehazten diren osagaien bidez errepresentatuko dira kate gisa:

- **Tokenak.** Osagai-mota honetan elementu bakoitzari dagokion testua izango da gordeko dena.
- **Entitateak.** Tokenen kasu partikular bat izanagatik, beste osagai-mota bat bezala errepresentatuko dira. Dagokien testua, mota, lema eta informazio gramatikala adierazteko etiketak jasoko dira.
- **Zuriuneak.** Token eta entitateen arteko zuriuneak osagai-mota honen bidez errepresentatuko dira, kasuan kasu dagokien testua gordez. Hitzen arteko ohiko ' ' zuriuneez gain, tabulazioak edota antzerakoak ere izan ahalko dira, bai eta haien segidak ere.
- **Sintagma hasierak.** Sintagma hasierak osagai-mota honen bidez markatuko dira, kasuan kasu dagokien etiketa gordez.
- **Sintagma bukaerak.** Sintagma hasierekin batera sintagmen bukaerak sintagmen mugak markatzeko erabiliko dira, kasuan kasu dagokien etiketa jasoz. Sintagmen teilakatzea ez da onartuko, zuhaitzetan halakoak ez baitira posible, baina abiaratzea bai, eta horixe izango da, hain justu ere, zuhaitz sintaktikoen egitura hierarkikoa kate bezala errepresentatzeko modua.

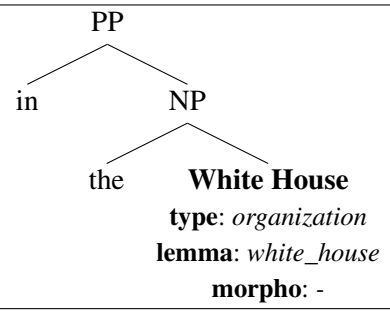
Era honetara, zuhaitz sintaktiko bat oso modu errazean errepresenta daiteke kate modura, zuhaitza eratzen duten elementuak azaldu berri diren osagaien segida bezala jarri baino ez baita egin behar haien ordena errespetatuz. 4.7 irudiak horren adibide bat erakusten du *in the White House* testu-zatiaren zuhaitz sintaktikoarekin.

Identitate-taula

Aurreko puntuan zuhaitz sintaktiko bat kate modura jartzeko hurbilpena azaldu bada ere, errepresentazio horrek ondorengo bi arazoak ditu atzizki-taulekin erabiltzeko:

- **Osagaien ordena-erlazio baten beharra.** 4.3.1 atalean ikusi bezala, atzizki-taulek, definizioz, kate bateko atzizki ordenatuen indizeak biltzen dituzte. Horretarako, noski, kateko oinarrizko osagaien arteko ordena-erlazio bat behar da. Aurreko atalean aurkezturiko osagaiak, baina, bost mota ezberdinetakoak izan daitezke, eta ez dago euren arteko ordena-erlazio argirik.

ZUHAITZ SINTAKTIKOA



KATEA									
Mota	sint. has.	tokena	zuriunea	sint. has.	tokena	zuriunea	entitatea	sint. buk.	sint. buk.
Edukia	PP	in	' '	NP	the	' '	**White House** **type**: *organization* **lemma**: *white_house* **morpho**: -	NP	PP

4.7 Irudia: Zuhaitz sintaktiko baten errepresentazioa kate modura

- **Osagaien tamaina aldakorra.** Atzizki-taulak erabiltzeko katearen oinarrizko osagaiek tamaina finkoa izatea komeni da, baina kasu honetan ez da hori gertatzen. Izan ere, token, entitate, zuriune nahiz sintagmen etiketak edozein tamainatakoak izan daitezke printzipioz.

Laburbilduz, tamaina finkoko eta ordena-erlazio argi bat duten osagaiak izatea komeni da, baina aurreko puntuan aurkeztutakoek ez dituzte baldintza horiek betetzen. Horri aurre egiteko, identitate-taulak erabiltzea erabaki da, osagai horien instantzia ezberdin bakoitza ID batekin lotuko dutenak. ID modura, beti ere, zenbaki arruntak erabiliko dira zentzuzko balio maximo batekin, hizpide diren baldintza biak betetzen dituztenak: tamaina finko bat izatea (balio maximoa adierazteko behar den bit kopurua) eta ordena-erlazio argi bat izatea (zenbaki arrunten ordena naturala). Era honetara, aurreko puntuko osagaiak zuzenzuzenean erabili beharrean kateak ID horien bidez osatuko dira, haien kodeketa eta deskodeketarako identitate-taula erabiliz. 4.8 irudiak ideia honen adibide bat erakusten du, 4.7 irudiko zatiaren identitate-taula eta haren bidez kodeturiko katea jasoz. Ikus daitekeenez, identitate-taulan osagai bikoiztuak onartzen ez direnez ' ' zuriunearen bi agerpenak ID bakar batekin kodetzen dira, kasu honetan 4 dena.

Planteamendu honek behar bezala funtziona dezan, baina, ezinbestekoa da identitate-taularen kontsulta modu eraginkorrean egin ahal izatea bi zentzuetan. Horretarako, taula bikoiztea erabaki da, norantza bakoitzean egitura ezberdin bat erabiliz. Zehazki, IDa emanda osagaia eskuratzeko IDak berak indexaturiko bektore bat baliatuko da, eta osa-

ID	OSAGAIA	
	Mota	**Edukia**
1	sint. has.	PP
2	sint. buk.	PP
3	tokena	in
4	zuriunea	' '
5	sint. has.	NP
6	sint. buk.	NP
7	tokena	the
8	entitatea	**White House** **type**: *organization* **lemma**: *white_house* **morpho**: -

KATEA								
1	3	4	5	7	4	8	6	2

4.8 Irudia: 4.7 irudiko zatiaren identitate-taula eta haren bidez kodeturiko katea

gaia emanda IDa eskuratzeko, berriz, hash-taula bat.

Eduki-taula

Aurreko puntuan identitate-taulak azaldu dira, osagai bakoitzari ID ezberdin bat esleituz zuhaitz sintaktikoak atzizki-taulekin erabiltzeko moduko kate gisa kodetzeko aukera ematen zutenak. 4.3.2 atalean hausnartzen zen modura, baina, aztergai den problema honetan datu gehiagorekin egiten da lan, hitz- eta entitate-lerrokatzeekin adibidez. Informazio gehigarri hori modu eraginkorrean tratatu ahal izateko, eduki-taula konposatu bat erabiltzea erabaki da corpus paraleloko hizkuntza bakoitzarentzat. Eduki-taula hizkuntza horretako osagaien segida modura antolatzen da, osagai bakoitzarentzat ondorengo eremuak jasoz:

- **ID (64 bit).** Osagaiaren identifikatzailea, identitate-taularen bidez dagokion edukiarekin lotzen duena. Entitateen kasuan, jatorrizko IDen ukapenak gordeko dira. 4.8 irudiko *White House* entitatearentzat, adibidez, eremu honek -8 balioa hartuko luke, eta *the* tokenarentzat, berriz, 7.

- **Lerrokatze-segmentuaren hasiera-indizea (64 bit).** Beste hizkuntzako eduki-taulan osagai honekin lerrokatuta daudenen indize minimoa. Osagaia lerrokatuta ez badago (izan lerrokatu ez den token edo entitate bat delako ala izan beste mota bateko osagaia delako), eremu honek 0 balioa hartuko du.

- **Lerrokatze-segmentuaren tamaina (32 bit).** Osagai honekin lerrokaturiko segmentuaren osagai kopurua (beste hizkuntzako eduki-taulan osagai honekin lerrokatuta daudenen indize maximo eta minimoaren arteko kendura gehi bat). Aurreko eremuarekin bezala, lerrokatu gabeko osagaientzat 0 balioa erabiliko da.

- **Entitatearen lerrokatze-identifikatzailea (32 bit).** Elkarren artean lerrokatutako entitate-bikoteak identifikatzeko zenbaki bat. Corpus paraleloko sarrera bakoitzean, elkarren artean lerrokatutako entitateek balio bera izango dute eremu honetan. Osagaia beste mota batekoa edo entitatea izanda lerrokatu gabea bada, eremu honek 0 balioa hartuko du.

- **Pisu lexikala (64 bit).** Osagai honi pisu lexikalaren kalkuluan dagokion biderkagaiaren logaritmoa. Corpus paraleloko n. sarrerako i. tokena emanda, eta s_k eta s'_k hurrenez hurren sarrera horretako abiapuntu eta xede hizkuntzetako k. tokenak izanik, honako adierazpenaren bidez kalkulatzen da hori:

$$\ln\left(\max\left(\{p(null|s_i)\} \cup \{p(s'_{i'}|s_i) : (i,i') \in A_n\}\right)\right)$$

  Eremu hau token eta entitateekin soilik erabiliko da, eta gainerako osagaientzat 0 balioa hartuko du.

- **Atzizki-taulako balioa (64 bit).** IDen katearen atzizki-taulan osagai honen indizeko balioa. Aurrerago azalduko den prozeduraren bidez entitateen orokortzea gauzatu ahal izateko, zenbaki ez-positibo guztiek berdintasun-erlazioa betetzen dutela suposatuko da atzizki-taula osatzean. Beste hitz batzuetan, $\ldots = -3 = -2 = -1 = 0 < 1 < 2 < 3 < \ldots$ ordena-erlazioa erabiliko da.

Ikus daitekeenez, eremuek tamaina finko bat dute, eta muga horiek ezartzeko oso irizpide kontserbadoreak jarraitu dira. Honela, 320 exabytera arteko (320 milioi terabyte) eduki-taulak onartzen dituzte, corpus paraleloko sarrera bakoitzeko tamaina maximoa 80 gigabytekoa izanik. Bistakoa denez, gaur egun erabat pentsaezina da muga horiek gainditzea[2], eta sekula hala beharko balitz, oso modu errazean egoki zitezkeen tamaina maximoak.

Eduki-taulen adibide modura, 4.9 irudian *" White House"* zatiari dagokiona erakusten da, ' ' zuriuneak eta *White House* entitateak osatzen dutena. Lehenari dagokionez, IDa eta atzizki-taulari posizio horretan dagokion balioa baino ez dira gordetzen, zuriune bat denez gainerako eremuei 0 balioa esleituz. Bigarrenaren kasuan, berriz, entitate nahiz hitz mailan lerrokatuta dagoela ikus daiteke eta, ondorioz, eremu guzti-guztiak bete dira.

[2] Argibide bezala, eduki-taulen tamaina jatorrizko testu lauena baino 20-25 bat aldiz handiagoa dela ikusi da esperimentalki, eta testu lau modura gehienez ere 16 exabyte inguruko tamaina eta 4 gigabyte inguruko esaldiak dituzten corpusekin aritzeko aukera ematen du horrek.

1. OSAGAIA						2. OSAGAIA					
ID	**ler. has.**	**ler. tam.**	**enti. ler.**	**pisu lex.**	**atz. taula**	**ID**	**ler. has.**	**ler. tam.**	**enti. ler.**	**pisu lex.**	**atz. taula**
4	0	0	0	0	1	-8	2	1	1	-0.865	2
64bit	*64bit*	*32bit*	*32bit*	*64bit*	*64bit*	*64bit*	*64bit*	*32bit*	*32bit*	*64bit*	*64bit*

4.9 Irudia: Eduki-taulen adibide bat

4.3.4 Itzulpen-algoritmoa

Aurreko azpiatalean azalduriko datu-egitura EBMT aurreprozesua modu eraginkorrean burutzeko diseinatu da, eta azpiatal honetan horretarako garatu den itzulpen-algoritmoa azaltzen da. Abiapuntu modura, noski, entrenamenduaren irteerarekin identitate-taula eta eduki-taulak osatu beharko dira eta, behin hori eginda, EBMT aurreprozesuko urratsak jarraian zehazten diren modura burutu:

- **Analisia.** Urrats honi lotuta, egiteko berezi bakarra irteerako zuhaitz sintaktikoa kate modura errepresentatzea da corpus elebidunaren identitate-taula erabiliz. Entitatea ez den osagairen bat identitate-taulan agertzen ez bada, hura beregain hartzen duten zati guztiak zuzen-zuzenean baztertuko dira, ezinezkoa izango baita corpusean bat-etortzerik izatea. Entitateentzat, berriz, 0 identifikatzailea erabiliko da.

- **Bilaketa.** Hemen sartzen dira atzizki-taulak jokoan, haiek erabiliz aurreko urratsean eraikitako katearen zatiak corpus paraleloari dagokion katean bilatu beharko baitira modu hierarkikoan. Horretarako eduki-taulako ID eta atzizki-taulen eremuak erabiliko dira, ID balio ez-positibo guztiek berdintasun-erlazioa betetzen dutela kontuan hartuz beti ere. Era honetara entitateen orokortzea gauzatzen da, baina hurbilpen honek badu arazo bat: lerrokatu gabeko entiate ezberdinen bat-etortzeak izatea. Hori konpontzeko, izandako bat-etortze guztietan lerrokatu gabeko entitateen edukia bat datorrela egiaztatu beharko da, baldintza hori betetzen ez dutenak baztertuz. Bistakoa denez, eduki-taulako entitateen lerrokatzaile-identifikatzailearen eremua erabiliz bat-etortzeen osagai kopuruarekiko denbora linealean egin daiteke hori.

  Sinpletasunaren alde egitura gehigarririk erabiltzen ez denez, bilaketa bitar baten bidez egingo da hau guztia, 4.3.1 atalean azaldu bezala $O(m \log n)$ kostua duena. Kasu honetan bilatu beharreko zatiaren m osagai kopurua txikia izatea espero daiteke, gehienez ere esaldi oso batenak izan ahalko baitira. Termino kritikoa, batez ere eskalagarritasunaren ikuspegitik, corpus paraleloaren n osagai kopurua da beraz eta, harekiko hazkuntza abiadura logaritmikoa denez, proposaturiko hurbilpena oso-oso eraginkorra suertatzen da.

- **Itzulpena eta iragazketa.** Urrats hau corpus elebidunean itzulpenari dagokion token-segida identifikatu eta entitateak itzultzeaz arduratzen da. Bigarrenerako nahikoa da eduki-taulan horretarako dagoen eremuaren bidez entitate bakoitzarekin lerrokatuta dagoena identifikatu eta arestian azalduriko itzulpen-prozedura aplikatzea. Aurrenekorako, berriz, eduki-taulako lerrokatze-segmentuaren hasiera-indizearen eta tamainaren eremuak erabiliko dira, osagai bakoitzarekin lerrokaturiko segmentua non hasi eta non bukatzen den jakiteko balio dutenak. Bat-etortze bakoitzeko, bertako osagaien hasiera-indize minimoak eta bukaera-indize maximoak mugatuko dute itzulpenari dagokion segmentua. Ondoren, segmentu horrekin prozedura berbera aplikatu eta haren hasiera-indize minimoa eta bukaera-indize maximoa jatorrizkoren barruan sartzen direla ziurtatuko da, aurkako kasuan bat-etortzea baztertuz. Amaitzeko, itzulpenari dagokion segmentuan itxi gabeko sintagma hasiera edota bukaerak badaude, segmentu hori zabaltzen joango da horiek itxi arte, era honetara erantsitako osagaien lerrokatze-segmentuaren hasiera eta bukaera indizeak jatorrizkoaren barruan sartzen direla ziurtatuz beti ere, eta bestela bat-etortzea baztertuz. Nahasgarri samarra egin badaiteke ere, eduki-taula aurrean izanda prozedura oso sinplea suertatzen da, eta jatorrizko eta itzulitako segmentuekiko denbora linealean egin dadin zailtasun handiegirik gabe inplementa daiteke, biak ala biak pasada bakar batean prozesatuz.

- **Hautaketa.** Hautaketarako hainbat irizpide ematen baziren ere, tribiala ez den bakarra pisu lexikalena da. Nolanahi ere, eduki-taulan horretarako dagoen eremuari esker modu errazean egin daiteke hori ere. Honela, nahikoa da zentzu bakoitzeko segmentuko osagaiek eremu horretan dituzten balioak batu eta segmentu horretako token kopuruarengatik zatitzea, eta horrela lortutako bi balioen batura handiena duen bat-etortzea izango da pisu lexikal handiena dagokiona ere.

Argiago gera dadin urrats hauek puntuz puntu azaldu badira ere, bistakoa denez inplementazioan dena batera egin ahalko da eraginkortasunaren alde, izandako bat-etortzeak pasada bakar batean prozesatuz.

5. KAPITULUA

Inplementazioa

Kapitulu honetan aurrekoan diseinatutako sistemarentzat garatu den inplementazioa aurkezten da. Honela, atalez atal jarraituriko hurbilpen teknikoa, kodearen egituraketa eta egindako eraginkortasun-optimizazioak azaltzen dira.

5.1 Hurbilpen teknikoa

Aurreko kapituluan diseinaturiko sistema inplementatzeko Java programazio-lengoaia erabili da. Erabaki honen atzean hainbat arrazoi aurkitzen dira, besteak beste lengoaiaren aldez aurretiko ezagutza, bere liburutegi estandar zabala, eta proiektuan erabili beharreko hirugarrenen software gehienak lengoaia horretarako API bat eskaintzea.

Inplementazioa esportatutako APIaren terminoetan pentsatuz egin da, osagai hedagarri eta berrerabilgarriak eraikitzen ahaleginduz, eta Bloch (2008) erabili da horretarako erreferentzia nagusitzat. Era berean, garatutako softwarea aldatu, hedatu edota liburutegi bezala erabili nahiko luketen programatzaileei zuzenduriko Javadoc dokumentazioa idatzi da, esportatutako APIa oso-osorik eta xehetasun guztiekin deskribatzen duena. Egindako lana mundu osoko garatzaileentzat baliagarria izan dadin, halaber, ingelesa erabili da bai Javadoc dokumentaziorako bai eta iturburu-koderako ere.

Aplikazioa ahalik eta modular eta hedagarriena izan dadin, Objektuei Orientatutako Programazio paradigma jarraituz interfazeen erabilera sakona egin da. Honela, aplikazioaren domeinu-eredurako hainbat interfaze sortu dira, nagusiki testuaren errepresentazio

hierarkikora bideratuak, eta haientzako inplementazioak eman. Modu honetara, interfaze horientzako errepresentazio ezberdinak erabiltzea ahalbidetzen da, aurrerago azalduko den bezala datuak memorian ala fitxategietan mantentzen dituztenak adibidez. Ondoren, domeinu-ereduko interfazeetan oinarrituz aplikazioaren funtsezko osagaientzako interfazeak definitu dira: analizatzailea, entitate-ezagutzailea, hiztegi elebiduna, sortzaile morfologikoa... Honela, eginkizun hauetarako metodo edota tresna berriak erabili nahi izanez gero, nahikoa da kasuan kasu dagokion interfazea modu egokian inplementatzea. Proiektu honetan, zehazki, baliabide eta aurrekarien kapituluan aurkezturiko tresnei loturik ondorengo liburutegiak erabili dira:

- **Stanford CoreNLP.** Analizatzaile hau Javaz idatzita dago oso-osorik, eta JAR liburutegi bezala erabili da honenbestez. Hori dela eta, erabiltzaileak ez dauka aparteko ezer instalatu beharrik berarekin lan egiteko.

- **Freeling.** Analizatzaile hau C++ lengoaian idatzita badago ere, Javarako API bat ere eskaintzen du. Hori erabili ahal izateko, baina, erabiltzaileak Freelingen instalazio lokal bat behar du, bai eta Swig erabiliz Java wrapper-aren liburutegi natiboa konpilatu ere. Freelingen iturburu kodearekin batera horretarako jarraibideak aurki daitezke.

- **Lttoolbox-java.** Apertiumen Java *port* hau sorkuntza morfologikoa egiteko erabiltzen da, eta sorkuntzarako hiztegi konpilatuaz gain, ez du aparteko ezer instalatzea eskatzen beraz.

- **Berkeley Aligner.** Lerrokatzaile hau Javaz idatzita badago ere, aplikazio independente modura erabiltzeko izan zen diseinatua. Honela, esportatzen duen APIa ez da batere egokia liburutegi bezala erabiltzeko, batez ere esparru eta ikusgaitasun arazo larriak baititu. Horren aurrean, Berkeley Alignerren *fork* arin bat sortzea izan zitekeen aukera bat, baina etorkizunari begira bere garapena eta mantenimendua garestiegia izango litzatekeela pentsatu da. Horren ordez, arazoaren jatorrian dauden diseinu-akats berberez baliatuz bere portaera aurreikusita ez zegoen modu batean manipulatu da herentzia-mekanismoaren bidez, eta era honetara bere JAR exekutagarria liburutegi modura erabiltzea eskuratu. Honenbestez, aparteko ezer instalatu gabe erabil daiteke Berkeley Aligner ere.

Bestalde, ondorengo tresnek Java APIrik eskaintzen ez dutenez kanpo-prozesu bezala erabili behar dira:

- **Eustagger.** Euskarazko testua analizatzeko Eustagger aplikazio independente modura exekutatu eta haren irteera-fitxategiak irakurri eta interpretatzen dira. Itzuli beharreko testuarekin, baina, hurbilpen hau ez da bideragarria denbora errealean aritu nahi bada eta, ondorioz, garatutako sistemak euskara xede hizkuntza bezala soilik onartzen du.
- **GIZA++.** Eustaggerren antzera, GIZA++ erabiltzeko aukera bakarra aplikazio independente modura da. Hori dela eta, lerrokatzaile hau erabili ahal izateko entrenamendua hainbat urratsetan egin behar da: lehendabizi corpus paraleloa analizatu, ondoren dagokion testu tokenizatua eskuratu, GIZA++ erabiliz lerrokatu eta, amaitzeko, bere irteera interpretatuz identitate- eta eduki-taulak osatu.

Bukatzeko, aplikazioaren oinarrizko funtzionalitatea inplementatzeko ondorengo kanpo liburutegiak ere erabili dira:

- **Apache Commons CLI.** Komando-lerroko parametroak prozesatzeko liburutegi bat da hau, bistakoa denez aplikazioaren komando-lerroko interfazea inplementatzeko erabili dena.
- **MapDB.** MapDB Javarako biltegiratze-motore edo *storage engine* bat da, datu-base bat propioki ez izanagatik haiek eraikitzeko euskarritzat erabil daitekeena. Proiektu honetan fitxategietan oinarritutako (*file-backed*) hash-taulak inplementatzeko erabili da, aurrerago azalduko den bezala identitate-taulekin eta Wikipediako *dump*ekin modu eraginkorrean lan egiteko baliatzen direnak.

5.2 Kodearen egituraketa

Garatutako softwarearen tamaina eta konplexutasuna dezente handia da eta, horren erakusgarri, 70 interfaze eta klasek osatzen dute klase habiaratuak kontuan hartu gabe. Hau guztia helburu ziren modulartasun, hedagarritasun eta berrerabilgarritasuna lortzera bideraturik dago eta, kodea ulerterraza eta maneiagarria izan dadin, modu egoki batean antolatu eta egituratu da, ondorengo paketeetan bereiziz:

- **es.ehu.si.ixa.prebmt.** Erro-paketea. Komando-lerroko interfazea inplementatzen duen klasea eta 4.2.3 atalean azaldu bezala itzulpen partzialak iradokitzeko berezko euskarririk eskaintzen ez duten itzultzaileekiko integrazioa gauzatzeko postprozesurako klasea biltzen ditu.

- **es.ehu.si.ixa.prebmt.model.** Domeinu-eredua errepresentatzeko interfazeak: corpus elebakarra, corpus elebiduna, testua, sintagma, tokena, entitatea, entitate lerrokatua, zuriunea... Haien inplementazioak ondorengo bi azpipaketeetan banatzen dira:
  - **es.ehu.si.ixa.prebmt.model.inmemory.** Edukia oso-osorik memorian kargatzen duten inplementazioak.
  - **es.ehu.si.ixa.prebmt.model.filebacked.** Edukia fitxategi batean mantentzen duten inplementazioak, datuak behar diren neurrian bertatik irakurriz.
- **es.ehu.si.ixa.prebmt.analysis.** Testuaren analisiari lotutako funtzionalitatea biltzen duen paketea. Analizatzaileentzako interfaze bat zehazten du, bai eta Stanford CoreNLP eta Freelingentzako inplementazio bana eman ere.
- **es.ehu.si.ixa.prebmt.entityrecognizer.** Entitateak testuaren ohiko analisiarekiko modu independetean ezagutzeko klase eta interfazeak. Proiektu honetan zenbakien ezagutzaile bat baino ez da inplementatu, baina pakete honetako baliabideen bidez beste edozein aukera ere modu errazean txerta zitekeen.
- **es.ehu.si.ixa.prebmt.generation.** Sorkuntza morfologikoa egiteko interfazea, eta haren inplementazio bat lttoolbox-java (Apertiumen Java *port*a) erabiliz.
- **es.ehu.si.ixa.prebmt.dictionary.** Entitate-hiztegi elebidunentzako interfazea, eta haren inplementazioak zenbakientzat, testu lauz emandako hiztegientzat, Wikipediarentzat, nahiz aurrekoak konbinatuz sorturiko hiztegi konposatuentzat.
- **es.ehu.si.ixa.prebmt.parsing.** Corpus elebakar bat (normalean, corpus paralelo batean hizkuntza jakin bati dagokiona) Javako *Reader* eta *Writer*-en bidez irakurri eta idazteko interfaze eta klaseak. Ideia hau serializazioaren lerro berean kokatzen bada ere, ez da Javaren berezko serializazio APIa erabili. Izan ere, corpus elebakar analizatuak XML bezala irakurri eta idazteko klase batez gain itzulgarriak ez diren inplementazioak ere ematen dira, kanpo-prozesu bezala exekutatu behar diren tresnekiko lotura lanak egiten dituztenak. Horien artean Eustaggerren analisiak irakurtzeko klase bat aurkitzen da, bai eta corpus analizatu bat GIZA++ lerrokatzaileari pasatzeko testu tokenizatu bezala idazten duen beste bat ere.
- **es.ehu.si.ixa.prebmt.translation.** EBMT aurreprozesuaren bidez itzulpen partzialak sortzeko funtzionalitate orokorra biltzen duen paketea, diseinuaren kapituluan zehazturiko defektuzko inplementazioa ere jasotzen duena.

- **es.ehu.si.ixa.prebmt.util.** Aplikazioan zehar erabiltzen diren klase lagungarriak: sarrera/irteerari loturiko bat, atzizki-taulekin lan egiteko beste bat, konkurrentziarekin dihardutenak...

- **edu.berkeley.nlp.wordAlignment.** Aurreko atalean azaldu bezala Berkeley Alignerrekiko lotura lanak egiteko klaseak.

Pakete hauen nahiz, oro har, garatutako softwarearen API publikoaren inguruko informazio gehiagorako, Javadoc dokumentaziora jotzea gomendatzen da.

5.3 Eraginkortasun-optimizazioak

Era honetako aplikazioen eraginkortasunari begira erabilitako algoritmoa izan ohi da alderdirik funtsezkoena, 4.3 atalean sakontasunez landu dena. Diseinuaz gain, baina, inplementazioak ere badu bere zeresana eta, nola prozesamendu-denborari hala memoria-erabilerari dagokionean ahalik eta arinena izan dadin, ondorengo teknikak erabili dira proiektu honetan:

- **Paralelizazioa.** Duela hamarkada bat *frequency scaling* edo maiztasun-eskalatzeak konputagailuen arkitekturako paradigma nagusia izateari utzi zionetik, konputazio paraleloa geroz eta indar handiagoa hartzen ari da sistema eraginkor eta eskalagarrien eraikuntzan. Joera horrekin bat eginez, eskainitako inplementazioak dagozkion bi faseak paraleloan exekutatzeko aukera eskaintzen du:

  - **Entrenamenduan** corpus paraleloko sarrera bakoitza hari batek analizatzen du, eta Berkeley Aligner erabiliz ere lerrokatzea paraleloan burutzen da.
  - **EBMT aurreprozesuan** sarrerako testua lerroka banatzen da, haietako bakoitza hari bereizi batean prozesatuz.

  Bi kasuetan programak erabili beharreko hari kopurua zehazteko aukera ematen du, sarrerako edukia horren arabera kontsumitu eta irteera ordena egokian idatziz. Hari bat bere lana amaituta lehenago zihoan ataza astunago batek bere irteera idatzi zain gera ez dadin, tamaina maximo bateko buffer bat erabiltzen da. Era honetara, buffererrarentzako leiho egoki bat erabiltzen den bitartean eskuragarri diren baliabideak erabat aprobetxatzea eskuratzen da. Horri esker, eginiko neurketetan sistemaren

eskalagarritasuna oso ona dela ikusi da, exekuzio denbora prozesadore kopuruarekiko modu zuzenean murriztuz ia-ia (prozesadore kopurua bikoizteak, adibidez, exekuzio-denbora erdira ingurura jaitsiko luke).

- ***Flyweight* patroia.** Diseinu-patroi hau datu berberak errepresentatzen dituzten osagaiak objektu ezberdinen artean partekatzean datza. Modu honetara, informazioaren bikoizketa saihesten da, memoria-erabilera murriztuz. Testu bateko elementuak oso errepikakorrak direnez (hitz gehienek, adibidez, agerpen ugari izan ohi dituzte), memoria asko aurrez daiteke modu honetara eta, horretarako, domeinu-ereduko osagai gehienak patroi hau erabiliz inplementatu dira. Adibide modura, token edo entitate batek hainbat agerpen izanagatik bere edukia behin bakarrik gordeko litzateke memorian, eta instantzia guztiek hari egingo liokete erreferentzia.

- **Diskoaren erabilera eta *memory-mapping* teknika.** Memoria-eskakizunak ahalik eta baxuenak izan daitezen, datu-egitura astunenak memorian oso-osorik kargatu beharrean diskoan mantendu eta behar den neurrian irakurtzen dira. Honela, eduki-taulak fitxategi bitarretan gordetzen dira, eta *memory-mapping* teknikaren bidez prozesatu. Teknika honek memoria birtualeko segmentu bat fitxategi batekin mapeatzen du bytez byte, fitxategi handien ausazko atzipena modu eraginkorrean egitea ahalbidetzen duena. Identitate-taulekin lan egiteko, berriz, 5.1 atalean aurkeztutako MapDB liburutegia erabiltzen da. Zehatzagoak izanez, identitate-taula hash-taula baten bidez errepresentatzen da zentzu bakoitzean, liburutegi honi esker fitxategi batean oinarritzen direnak datu-base arin bat balitz bezala. Wikipediaren *dump*ekin ere hurbilpen bera jarraitzen da, jatorrizko XMLak fitxategietan oinarritutako hash-tauletara bihurtu eta horiek erabiltzeko aukera eskainiz.

6. KAPITULUA

Esperimentua eta emaitzak

Kapitulu honetan proposaturiko sistemaren portaera hobeto ezagutzeko buruturiko esperimentua eta bertan eskuraturiko emaitzak aurkezten dira. Honela, lehen atal batean esperimentuaren diseinuan sakontzen da, egin beharreko probak eta haietan aztertu beharreko aldagai eta aldaerak zehaztuz. Bigarren atalean, modu horretara eskuraturiko emaitzak aurkeztu eta aztertzen dira. Amaitzeko, azken atal batean urrats bat haratago joan eta emaitza hauen interpretazioa egiten da.

6.1 Esperimentuaren diseinua

Esperimentuaren helburua ongi argia da: proposaturiko sistemaren portaera baldintza ezberdinetan ebaluatu eta portaera honen zergatia ulertu ahal izatea, eginiko lanaren benetako ekarpena zein den hobekiago ezagutu eta etorkizuneko lan-lerroak markatzeko asmoz. Hori dela eta, esperimentuaren diseinuan hainbat aldagai kontsideratu dira sistemaren alderdi ezberdinek haren portaeran duten eragina neurtu eta ulertu ahal izateko, ondorengo azpiataletan xehe-xehe aurkezten direnak.

6.1.1 Landuriko hizkuntza eta corpusak

Orain arte ikusi bezala, proposaturiko sistema edozein hizkuntza eta domeinurekin lan egiteko diseinatua izan da, analizatzaile nahiz lerrokatzaile ezberdinak erabiltzeko modulartasuna eskainiz. Malgutasun horri esker testu oro erabilgarria izanagatik, baina, bista-

koa da jorraturiko hizkuntza bikoteak zein entrenamendu-corpusaren eta itzuli beharreko testuaren hurbiltasun eta ezaugarriek eskuraturiko emaitzetan eragin nabarmena izan dezaketela. Eragin hori doitasunez neurtzeko, ondorengo bi corpusekin esperimentatu da:

- **IVAP corpusa (gaztelania-euskara)**. IVAP corpusa deitu zaiona Herri Arduralaritzaren Euskal Erakundeak gaztelaniatik euskarara itzuliriko 91 lan-hitzarmenek osatzen dute, Euskal Herriko Agintaritzaren Aldizkarian argitaratuak guztiak. Lanhitzarmen hauetako 81 (50.824 esaldi guztira) entrenamendurako corpus modura hartu dira ausaz, 5 (2.366 esaldi) garapenerako, eta beste 5 (1.928 esaldi) testerako, gaztelaniarako Freeling eta euskararako Eustagger analizatzaileak erabiliz. Nolanahi ere, entrenamendu-fasean hitz-lerrokatzea behar bezala burutzeko esaldi gutxiegi dira horiek eta, hori dela eta, Elhuyarren corpus administratibo bateko beste 4.747.332 esaldirekin aberastu da entrenamenduko zatia, Elhuyarren itzulpen zerbitzuek erakunde publikoetarako (Jaurlaritza, udaletxeak...) eta enpresetarako (Euskaltel, Naturgas...) itzulitako web-orri eta dokumentuek osatzen dutena, lerrokatzeaz haratago ez luketenak emaitzetan eragin handiegirik izan beharko. Tamainaz txikia izanagatik, baina, IVAP corpusaren errepikakortasun-maila oso altua da, zehatz-mehatz errepikatzen diren formula eta esaldiz josita baitago. 3.2 ataleko analisiari jarraiki, baldintza hauek ezin mesedegarriagoak suertatzen dira EBMT bidezko itzulpen-sistema hibridoentzat eta, hori dela eta, proposaturiko sistemak emaitza onak ematea espero daiteke kasu honetan. Era berean, kontuan hartu behar da gaztelania eta euskararen arteko itzulpen-dibergentziek eta, bereziki, morfologia aberatsa nahiz ordena librea bezalako euskararen ezaugarriek hizkuntza bikote hau bereziki zaila egiten dutela eta, hori gutxi balitz, ez dela gehiegi landua izan gaur-gaurkoz.

- **Europarl corpusa (gaztelania-ingelesa)**. Europarl corpusa Europako Parlamentuko aktekin osatuta dago, eta europako 21 hizkuntzatan aurkitzen da. Aldiro-aldiro corpusaren bertsio berriak argitaratzen badira ere, kasu honetan ACL 2007an itzulpen automatiko estatistikoaren inguruko *workshop*eko ataza partekaturako gaztelania-ingelesa bikoterako luzatu zena erabili da[1], entrenamendu eta testerako banaketa berbera jarraituz eskuraturiko emaitzak ekitaldi hartakoekin alderagarriak izan daitezen. Honela, entrenamendu-corpusean 1.254.414 esaldi daude, eta testekoa, berriz, bitan banatzen da: bata domeinukoa, Europarletik hartutako eta entrenamendutik kendutako 2.000 esaldiz osatua, eta bestea domeinuz kanpokoa,

[1] Informazio gehiagorako nahiz corpusa jaisteko ikus http://www.statmt.org/wmt07/shared-task.html.

albistetako 2.007 esaldiz osatua eta News Commentary deritzona. Analizatzaile modura, berriz, Freeling erabili da gaztelaniarako eta Stanford CoreNLP ingeleserako. IVAP corpusarekin alderatuta, Europarlen errepikakortasun-maila dezente txikiagoa da, eta zer esanik ez domeinuz kanpoko test-esaldiei dagokienean. Honekin batera, aipatzekoa da gaztelania-ingelesa sakonki landuriko hizkuntza bikote bat dela, kalitate handiko emaitzak ematen dituzten sistemak garatu izan direlarik norabide honetarako. Esperimentuaren ikuspegitik, bada, interesgarritzat jo da puntu hauek sistemaren portaeran duten eragina neurtu eta aztertu ahal izatea.

6.1.2 Sistemaren osagaiak

4.2 atalean azaldu bezala, entrenamendua alde batera utzita proposaturiko sistemak bi fasetan lan egiten du: EBMT aurreprozesua batetik, sarrerako testuaren itzulpen partzialak sortzeaz arduratzen dena, eta integrazioa bestetik, itzulpen partzial horiek itzultzaile nagusiaren bidez osatu eta irteerako testua ematen duena. Ikusi denez, azken hau itzultzaile nagusiaren menpe dago erabat eta, neurri handi batean itzultzaile nagusia nahiz bestelako osagaiekiko desio zen independentziak baldintzaturik, ezin izan da gehiegi landu proiektu honetan. Proposaturiko sistemaren mamia, honenbestez, EBMT aurreprozesuan aurkitzen da ezbairik gabe eta, hori dela eta, sistema osoaren portaera ez ezik aurreprozesuarena ere bere baitan ebaluatzea erabaki da. Gauzak honela, eginiko esperimentuan sistemaren ondorengo osagaiak ebaluatzen dira bakoitza bere aldetik:

- **EBMT aurreprozesua**, hau da, sarrerako testuarentzat proposaturiko itzulpen partzialak euren baitan. Horretarako itzulpen partzial horien azterketa kuantitatibo eta kualitatibo bat burutzearen alde egin da, kopuru gordina, luzera eta antzerako datuak atera eta interpretatzeaz gain euren kalitatea baloratzeko saiakera bat ere eginez. Asmo horrekin, hizkuntza bikote bakoitzeko test-multzotik 100 esaldi hartu dira ausaz[2], eta bakoitzerako bost ebaluatzaileren laguntzaz eskuzko ebaluazio bat burutu. Honela, ebaluatzaileei itzulpen partzial bakoitza 1 (itzulpen okerra) eta 4 (itzulpen zuzena) arteko eskala batean baloratzeko eskatu zaie, eta era horretara eskuratutako emaitzak aztertu ondoren.

- **Integrazioa**, hau da, sistema osoaren portaera behin EBMT aurreprozesuko itzulpen partzialak itzultzaile nagusiaren bidez osatu ostean. Alderdi hau ebaluatzeko BLEU (*Bilingual Evaluation Understudy*) metrika automatikoa erabili da (Papineni

[2]Gaztelania-ingelesa bikotearen kasuan, domeinuko test-multzoa soilik erabili da.

et al., 2002), aztergai den sistemaren itzulpenari puntuazio bat esleitzen diona gizaki baten erreferentziazko itzulpenarekiko duen korrespondentzia neurtuz ondorengo adierazpenaren arabera:

$$BLEU = BP \times \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n\right)$$

non BP honela definitzen den laburtasun-penalizazioa baita:

$$BP = \begin{cases} 1 & c > r \\ e^{(1-r/c)} & c \leq r \end{cases}$$

non c sistemaren itzulpenaren luzera, r erreferentziazko itzulpenaren luzera, eta p_n jarraian erakusten den bezala definituriko n-grama doitasun egokitua baita:

$$p_n = \frac{\sum_{C \in \text{Candidates}} \sum_{\text{n-gram} \in C} \text{count}_{clip}\left(\text{n-gram}\right)}{\sum_{C' \in \text{Candidates}} \sum_{\text{n-gram}' \in C'} \text{count}\left(\text{n-gram}'\right)}$$

BLEU metrika automatiko ezagun eta erabiliena da, eta sistema ezberdinak modu errazean alderatzeko aukera ematen du. Eskuzko ebaluazioa, berriz, askoz garestiagoa suertatzen da, bereziki kasu honetan bezala hainbat aldagairen eragina neurtu nahi denean, eta emaitzak modu egokian kontrastatu eta estrapolatzea oso zaila egiten du. Horregatik guztiagatik hartu da, hain justu ere, sistema osoaren portaera ebaluatzeko BLEU erabiltzearen erabakia. Zehazki, aztertu beharreko ezarpen bakoitzeko proposaturiko sistemaren BLEU puntuazioa kalkulatu da test-multzo osoa erabiliz, bai eta sistema hartako itzultzaile nagusiarena bere baitan ere *baseline* modura. Biak alderatuz, proposaturiko sistemaren ekarpena neur daiteke, eta emaitzak behar bezala interpretatzeko zerbait gehiago behar denean hainbat esaldi hartu eta eskuz aztertu dira diferentziak.

Bestalde, esan beharrik ere ez da emaitzak itzultzaile nagusiaren menpe egongo direla erabat eta, zentzu horretan, interesgarria suertatzen da 3.2 atalean azaldurikoaren harira RBMT eta SMT sistemekiko hibridazioa aztertzea, bertan esaten zen bezala kasu bakoitzean proposaturiko sistemaren ekarpena ezberdina izatea espero baitaiteke. Hori dela eta, esperimentuak RBMT eta SMT sistema banarekin egin dira landuriko hizkuntza bikote bietan jarraian zehazten den bezala:

– **RBMT** sistema modura Matxin erabili da gaztelania-euskara bikotearentzat eta Apertium, berriz, gaztelania-ingelesaren kasuan, biak ala biak 2.4.3 atalean azalduriko OpenTrad proiektuaren parte direnak. 4.2.3 atalean landurikoaren harira, sistema hauek ez dute itzulpen partzialak iradokitzeko berezko euskarririk eskaintzen eta, hori dela eta, bertan azalduriko HTML etiketen erabilera jarraitu da integrazio-estrategiatzat.

– **SMT** sistema modura Moses erabili da landuriko bi hizkuntza bikoteentzat, 6.1.1 atalean azalduriko corpus berberak erabiliz bai hizkuntza-eredua bai eta sintagmetan oinarrituriko (*phrase-based*) itzulpen-eredua ere entrenatzeko, kasu bietan ezarpen lehenetsiekin. Matxin eta Apertiumek ez bezala, Mosesek itzulpen partzialak iradokitzeko berezko euskarria eskaintzen du ondorengo hiru estrategien bidez:

  * ***Exclusive***, iradokiriko itzulpen partziala soilik erabiltzen duena zehazturiko zatiarentzat, harekin teilakatzen diren sintagma-taulako sarrerak erabat baztertuz.
  * ***Constraint***, iradokiriko itzulpen partziala berau barne hartzen duten sintagma-taulako sarrerekin lehiarazten duena zehazturiko zatiarentzat.
  * ***Inclusive***, iradokiriko itzulpen partziala sintagma-taulako gainerako sarrerekin lehiarazten duena zehazturiko zatiarentzat.

  Ikus daitekeenez, jarraituriko hurbilpenaren arabera iradokiriko itzulpen partzialak beti errespetatzera behartuko litzateke, ala sistemak berak beste itzulpenen baten alde egiteko aukera onartu. Lehena proposaturiko sistemaren portaera gordina ikusteko egokiagoa izan daiteke, eta RBMT sistemekin jarraituriko integrazio-estrategiaren antzekoa da. Nolanahi ere, itzultzaile nagusiari berak komenigarriak deritzen itzulpen partzialak soilik berrerabiltzeko aukera ematea zentzuzkoa bezain komenigarria izan daiteke horretarako mekanismoa ondo badabil. Hori dela eta, hiru hurbilpenak erabiliz errepikatu dira esperimentu guztiak, kasuan kasu bakoitzaren eragina zein den aztertu eta estrategia egokiena identifikatzeko asmoz.

6.1.3 Orokortze-urrats bakoitzaren ekarpena

3.3 atalean planteatu eta aurrerago garatu bezala, itzulpen partzialen sorkuntzan ondorengo hiru urratsak eman dira:

1. Corpus paraleloko sarreren berrerabilpen zuzena
2. Corpus paraleloko sarreren orokortzea entitateen bidez
3. Corpus paraleloko sarreren orokortzea segmentua baino txikiagoak diren unitate sintaktikoen bidez

Esan bezala, urrats horietako bakoitzari esker orokortze handiago bat eskuratzen da baina, era berean, itzulpen partzialen kalitatean galera bat emango dela aurreikus daiteke. Hori modu zehatzagoan neurtu eta aztertzea biziki interesgarria suertatzen da, proposaturiko sistemaren ekarpen nagusia nondik datorren ulertu eta etorkizunera begira ze alderdi jorratzea komeni den jakiteko lagungarria izan baitaiteke. Hori dela eta, esperimentu-saio ezberdinetako emaitzak modu globalean baloratzeaz gain aipatu berri diren urratsen arabera ere bereiztea erabaki da komenigarri ikusi denean.

6.1.4 Hitz-lerrokatzea

4.2 atalean zehar ikusi bezala, hitz-lerrokatzeak berebiziko garrantzia du proposatutako sistemaren funtzionamenduan, itzulpen partzialak haren arabera identifikatzen baitira azken finean. Hori dela eta, geroz eta kalitate altuagoko hitz-lerrokatzea izan, orduan eta itzulpen partzial gehiago eta egokiagoak sortu ahalko direla espero daiteke. Gauzak honela, 2.3.1 atalean planteatu bezala esperimentuan lerrokatzaile eta lerrokatze-teknika ezberdinak probatzea erabaki da jarraian zehazten diren aukeren bidez alderdi honen eragina neurtu eta ezarpen egokienak identifikatzeko asmoz:

- **GIZA++** bere ezarpen lehenetsiekin, zalantzarik gabe aukerarik erabiliena dena itzulpen automatikoaren arloan eta, besteak beste, proposaturiko sistema bestelakoekin modu zuzen eta errazagoan alderatzea ahalbidetzen duena horri esker.
- **Berkeley Aligner (HMM)**, hau da, Berkeley Unibertsitatean garaturiko lerrokatzaile honen ezarpen lehenetsiak Markoven Eredu Ezkutua erabiliz. Modu honetara, proposaturiko sistema antzerako hurbilpena jarraitzen duten lerrokatzaile biren aurrean zenbaterainoko sentikorra den neurtu ahalko da eta, horrekin batera, arestian azalduriko arrazoiak medio sisteman hobekiago integratzen den Berkeley Aligner erabiltzearen komenigarritasuna ikusi.
- **Berkeley Aligner (distortsio sintaktikoa)**, hau da, aurreko lerrokatzaile bera lerrokatze-prozesuan analisi sintaktikoa barneratzeko ezarpenekin. Gogoan izan

behar da, beti ere, aukera hau fase esperimentalean aurkitzen dela gaur-gaurkoz, lerrokatzaileen inguruko ikerketa-esparrutik haratago ez duelarik arreta handiegirik bereganatu. Nolanahi ere, lehen aipatzen zen bezala proiektu honetan sintaxiari eta lerrokatzeari loturiko murriztapenekin egiten da lan, eta bigarrena lehena aintzakotzat hartuz egiteak emaitza trinkoagoak lortu eta murriztapenak gehiagotan betetzea ahalbide dezakeela espero daiteke honenbestez. Hori dela eta, interesgarri ikusi da aldagai esperimental modura aukera honen benetako eragina aztertzea.

6.2 Emaitzen azterketa

Aurreko atalean esperimentuan aztertu asmo diren aldagai eta aldaera guztiak zehazten baziren ere, funtsean hiru baino ez ziren horretarako diseinaturiko probak 6.1.2 azpiatalean ikusi bezala: EBMT aurreprozesuaren azterketa kuantitatiboa, EBMT aurreprozesuaren eskuzko ebaluazioa, eta sistema osoaren BLEU bidezko ebaluazio automatikoa. Honela, sistemaren portaeran alderdi ezberdinek duten eragina aztertzeko proba hauexek berak baliatuko dira corpusa, itzultzaile nagusia, lerrokatzailea edota orokortze-urratsentzat ezarpen ezberdinak erabiliz. Atal honetan, bada, proba horietako bakoitzerako eskuratutako emaitzak aurkeztu eta aztertzen dira azpiatal banatan.

6.2.1 EBMT aurreprozesuaren azterketa kuantitatiboa

Esperimentu-saiakera honen helburua EBMT aurreprozesuaren bidez itzul daitekeen testu proportzioa neurtu eta aldagai ezberdinen arabera dituen gorabeherak aztertzea da, proposaturiko sistemak baldintza ezberdinetan izan dezakeen inpaktua zein den eta nondik datorren ikusteko baliagarri izan daitekeena. Horretarako bi alderdiri erreparatu zaie: itzulpen partzial (hots, EBMT aurreprozesuaren bidez itzuli diren esaldi ala testu-zati) kopuruari batetik, eta haien token kopuruari bestetik (hau da, jatorrizko testutik zenbat token itzuli diren EBMT aurreprozesuaren bidez).

Halako analisi batek zentzurik izan dezan, baina, ezinbestekoa da test-multzoaren beraren esaldi eta token kopurua gogoan hartzea. Datu hauek, bada, 6.1 taulan ematen dira. Ikus daitekeenez, lehenago aipatu bezala multzo guztiek dituzte 2000 esaldi inguru, baina token kopuruan alde handiagoak daude. Zehatzagoak izanez, Europarlerako biek luzera bertsua dute esaldiko 30 token ingururekin, baina IVAPekoak apenas ditu 20 token esaldiko. Test-multzoei gertuagotik erreparatuz, IVAP corpusaren esaldi-luzeraren aldakortasunean

	Esaldi kopurua	Token kopurua	Tokenak esaldiko
IVAP (domeinuan)	1928	39625	20,55
Europarl (domeinuan)	2000	56213	28,01
Europarl (domeinuz kanpo)	2007	61341	30,67

6.1 Taula: Test-multzoen esaldi eta token kopurua

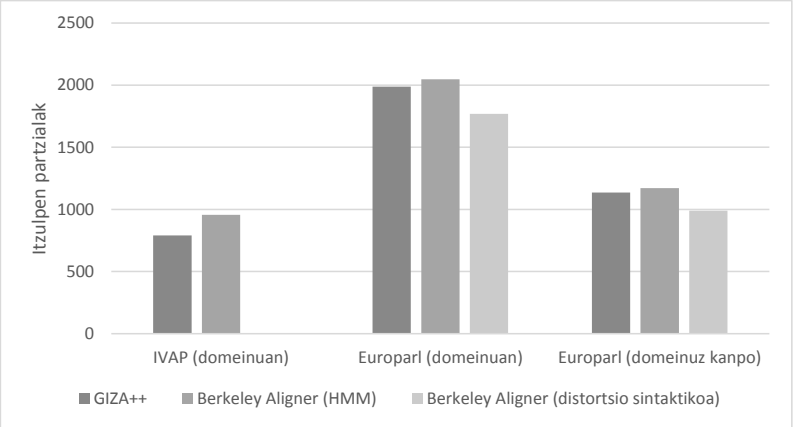
	Esaldi osoak	Entitateak	Zatiak		
			GIZA++	Berkeley (HMM)	Berkeley (sint.)
IVAP (domeinuan)	1211	1247 (+36)	2038 (+791)	2204 (+957)	-
Europarl (domeinuan)	54	72 (+18)	2060 (+1988)	2120 (+2048)	1841 (+1769)
Europarl (domeinuz kanpo)	2	2 (+0)	1138 (+1136)	1174 (+1172)	991 (+989)

6.2 Taula: Itzulpen partzial kopurua eta urrats bakoitzaren ekarpena

aurki daiteke honen zergatia. Izan ere, bizpahiru tokeneko hainbat eta hainbat sarrera ditu baina, horiek alde batera utzita, bertako esaldiak beste test-multzoetakoen antzekoak dira luzera aldetik. Gauzak honela, ez da uste alderdi honek emaitzetan eragin aipagarririk izan lezakeenik.

Puntu hau argituta, 6.2 taulan EBMT aurreprozesuaren bidez lortutako itzulpen partzial kopurua ematen da aldagai ezberdinen arabera. Zehazki, errenkada bakoitza test-multzo bati dagokio, eta zutabeka hiru orokortze-urratsak ematen dira, azkenaren kasuan hiru lerrokatze-ezarpenentzat. Parentesi artean, berriz, urrats horietako bakoitzaren ekarpena ikus daiteke. Adibidez, IVAPen kasuan esaldi osoak berrerabiliz 1211 esaldi itzultzen dira, eta entitateen bidezko orokortzearen bidez, berriz, 1247. Entitateen bidezko orokortzearen ekarpena, honenbestez, 36koa da, eta hori da hain justu ere parentesi artean ematen den balioa.

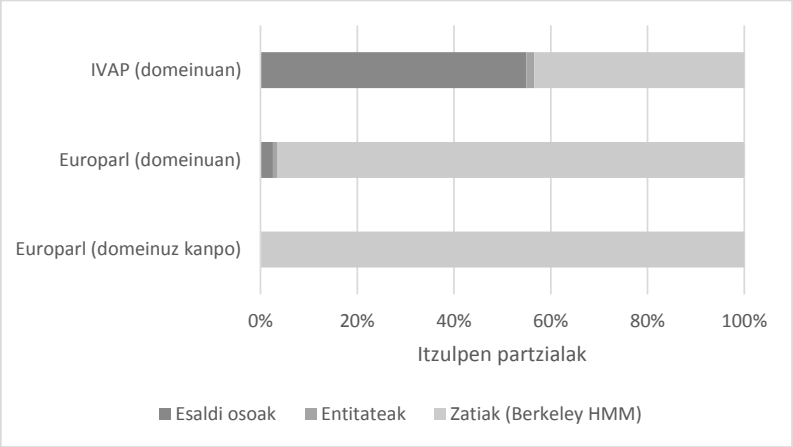
Emaitzei erreparatuz, erabilitako lerrokatze-ezarpenen arabera zatien bidezko orokortzearen inpaktua modu argi eta kontsistentean aldatzen dela ikus daiteke. Honela, 6.1 irudian modu grafikoan erakusten da bakoitzaren ekarpena kasuan kasu eta, bertan ikusten den bezala, Berkeley Aligner da Markoven Eredu Ezkutua darabilen ezarpenarekin itzulpen partzial gehien sortzeko gai dena test-multzo guztietan. Haren ostean GIZA++ dator, bien arteko aldea bereziki aipagarria izanik gaztelania-euskara bikotearen kasuan. Berkeley Aligner distortsio sintaktikoarekin erabilita, azkenik, emaitzarik okerrenak eskuratzen dira. Itzulpen partzial kopurua guztietan txikiena izateaz haratago, lehen begirada batean itzulpen haien kalitatea ere besteena baino nabarmenki okerragoa zela antzeman zitekeen eta, hori dela eta, IVAP corpusaren kasuan ezarpen hau zuzen-zuzenean baztertzea erabaki zen, entrenamendua bereziki garestia izatea espero baitzitekeen baldintza hauetan eta, azken finean, proiektu honetarako eskuragarri ziren baliabideak mugatuak baitziren.



6.1 Irudia: Zatien bidezko orokortzearen ekarpena itzulpen partzial kopuruarekiko lerrokatze-ezarpenen arabera

Bestalde, ikusi berri den bezala orokortze-gaitasun handiena duen Berkeley Alignerren HMM modua erabiliz, bi milana itzulpen partzial baino gehixeago lortzen dira domeinuko bi test-multzoentzat 6.2 taulak erakusten duen bezala. Honenbestez, batezbeste esaldiko itzulpen partzial bat baino gehiago sortzen dela ondoriozta daiteke, proposaturiko sistemaren inpaktua handia dela iradokitzen duena. Europarlen domeinuz kanpoko test-multzoan, berriz, itzulpen partzialen kopurua erdira jaisten da, baina hala eta guztiz ere kopuru altua izaten jarraitzen du, EBMT aurreprozesuak inpaktu erreal bat izan dezan aukera ematen duena.

Edozelan ere, itzulpen partzialen kopuru totalean antzeko emaitzak eskuratuagatik, orokortze-urrats bakoitzak horretarako egiten duen ekarpenean alde handiak ematen dira test-multzo ezberdinen artean. Argiago ikus dadin, 6.2 irudiak grafikoki jasotzen du itzulpen partzial kopuruarekiko urratsen arteko banaketa zein den kasu bakoitzean. Ikus daitekeenez, IVAPen kasuan, non esan bezala testuaren errepikakortasun-maila oso altua baita, itzulpenen erdia baino gehiago esaldi osoak bere horretan berrerabiliz lortzen dira. Europarlen domeinuko test-multzoan, berriz, modu honetara apenas sor daitezke itzulpen partzialen %2,5a, gainerako ia guztiak zatien bidezko orokortzeari esker lortzen direlarik. Europarlen domeinuz kanpoko test-multzoan, azkenik, itzulpen partzialen %100a da kasik zatien bidez eraikitzen dena. Gauzak honela, espero zitekeen bezala testuaren errepikakortasun-maila geroz eta altuagoa izan, orduan eta esaldi oso gehiago berrerabili ahalko direla ondoriozta daiteke, kopuru hau oso altua izatera irits daitekeelarik (IVAPen kasuan, hain justu ere, test-multzoko sarreren erdia baino gehiago errepikatzen da bere horretan corpusean). Bestalde, ikus daitekeen bezala entitateen bidezko orokortzearen



6.2 Irudia: Orokortze-urrats bakoitzaren ekarpena itzulpen partzial kopuruan

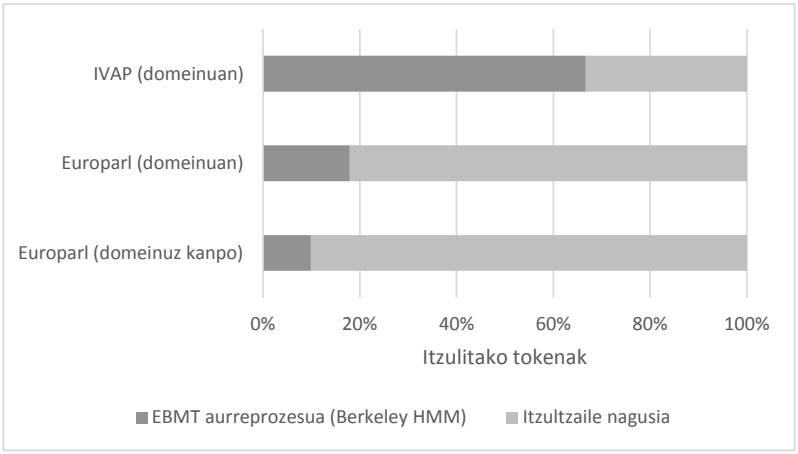
ekarpena nahiko eskasa da kasu guztietan beste biekin alderatuz gero. Hala eta guztiz ere, kontuan izan behar da datu hauek esaldi-mailan ematen direla, eta gogoan hartu behar da zatien bidezko orokortzearen bidez topatutako bat-etortzeek entitateak ere izan ditzaketela.

Aurrera eginez, 6.3 taulan EBMT aurreprozesuaren bidez itzulitako token kopurua biltzen da itzulpen partzial kopuruarekin egiten zen bezalaxe, parentesi artean token kopuru totalarekiko bakoitzari dagokion ehunekoa jasoz. Portzentaje horiek proposaturiko sistemaren inpaktu erreala neurtzeko ezin egokiagoak suertatzen dira eta, argiago ikus daitezen, 6.3 irudiak Berkeley Alignerren HMM modua erabiliz EBMT aurreprozesuaren eta itzultzaile nagusiaren bidez itzultzen diren tokenen arteko banaketa erakusten du test-multzo ezberdinentzat. Ikus daitekeenez, IVAPen kasuan EBMT aurreprozesuaren esku-hartzea oso handia da, tokenen bi heren haren bidez itzultzen baitira. Europarlen kasuan kopuru hau dezente txikiagoa da, baina esanguratsua izaten jarraitzen du hala ere: ia bosten bat domeinuan eta hamarren bat inguru domeinuz kanpo. Datu hauek oso positiboak dira, ingurune mesedegarrienean (testuaren errepikakortasun-maila oso altua denean alegia) proposaturiko sistemaren inpaktua izugarri altua izan daitekeela erakusten baitute, bai eta baldintza okerrenetan ere (domeinuz kanpo eta, honenbestez, testuaren errepikakortasun-maila oso baxua dela) eragin nabarmena izan dezakeela, tokenen %10 inguru EBMT aurreprozesuaren bidez itzultzera iritsiz proba hauetan.

Lerrokatze-ezarpenek itzulitako token kopuruan duten eraginari dagokionez, 6.4 irudian haietako bakoitzaren ekarpena erakusten da test-multzo ezberdinetan. Ikus daitekeenez, emaitzak 6.1 irudian jasotako itzulpen partzial kopuruekin lortzen zirenen oso antzekoak

	Esaldi osoak	Entitateak	Zatiak		
			GIZA++	Berkeley (HMM)	Berkeley (sint.)
IVAP (domeinuan)	18284 (%46,14)	18691 (%47,17)	23962 (%60,47)	26436 (%66.72)	-
Europarl (domeinuan)	379 (%0,62)	548 (%0,89)	10565 (%17,22)	10986 (%17,91)	9653 (%15,74)
Europarl (domeinuz kanpo)	12 (%0,02)	12 (%0,02)	5365 (%9,54)	5566 (%9,90)	4674 (%8,31)

6.3 Taula: EBMT aurreprozesuaren bidez itzulitako token kopurua eta totalarekiko ehunekoa

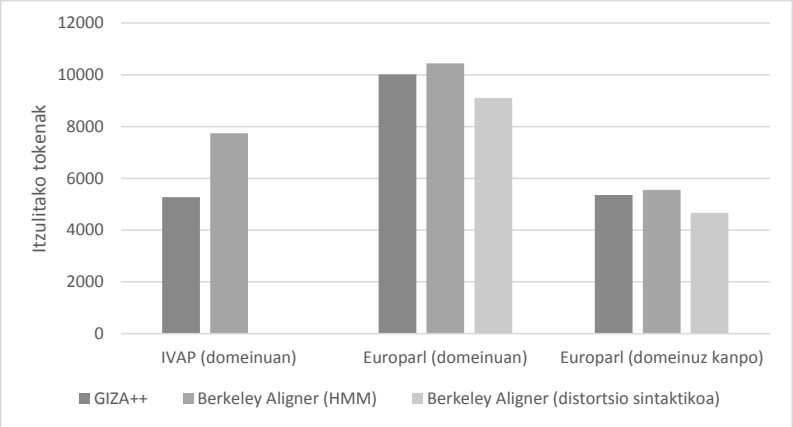


6.3 Irudia: Sistemaren osagai bakoitzaren ekarpena itzulitako token kopuruarekiko

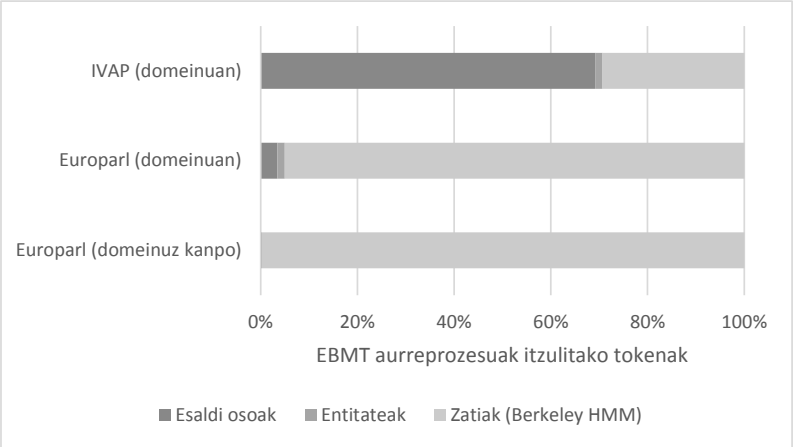
dira, ezberdintasun aipagarri bakarra IVAPen kasuan ematen delarik. Izan ere, test-multzo horretan Berkeley Alignerren nagusitasuna are nabarmenagoa egiten da token kopuruari erreparatuz gero, GIZA++ lerrokatzailearekin baino itzulpen partzial gehiago lortzeaz gain haien luzera handiagoa ere badela erakusten duena. Horretaz gain, IVAPen kasuan lortzen diren itzulpen partzialak beste bi test-multzoetakoak baino nabarmenki luzeagoak direla orokorrean antzeman daiteke, itzulpen partzial kopuruari zegokionean azken lekuan egonagatik token kopurua aintzakotzat hartuz Europarlen bi test-multzoen artean kokatzen delarik.

6.5 irudiak, bere aldetik, orokortze-urrats bakoitzak itzulitako token kopuruan duen ekarpena islatzen du. Espero zitekeen bezala, emaitza hauek 6.2 irudian itzulpen partzial kopuruarentzat lortzen zirenen lerro berean kokatzen dira, baina kasu honetan esaldi osoek eta, neurri txikiagoan, entitateen bidezko orokortzeak, tarte handiagoa hartzen dute, batez ere IVAPen kasuan. Esaldi osoen nahiz entitateen bidez lortzen diren itzulpenak zatiak erabiliz eskuratzen direnak baino luzeagoak direla erakusten du honek, erabat zentzuzkoa dena.

Itzulpenen luzeraren azterketan sakondu asmoz, 6.4 taulan orokortze-urrats bakoitzak emaniko itzulpen partzialen batezbesteko token kopurua jaso da orain arte ikusiriko da-



6.4 Irudia: Zatien bidezko orokortzearen ekarpena itzulitako token kopuruarekiko lerrokatze-ezarpenen arabera



6.5 Irudia: Orokortze-urrats bakoitzaren ekarpena itzulitako token kopuruan

tuak uztartuz. Arestian esaten zenaren harira eta esperotzekoa zenez, esaldi osoak berrerabiliz itzulitako esaldiak zatien bidez itzulitakoak baino nabarmenki luzeagoak direla ikus daiteke bertan, batez ere euskararen kasuan, non lehenengoen batezbesteko token kopurua bikoitza inguru izatera iristen baita. Entitateen bidezko orokortzearen bidez, berriz, bat-etortze zehatzekin baino esaldi dezente laburragoak itzultzen dira IVAPen kasuan, baina luzeagoak, berriz, Europarli dagokionean. Zentzu honetan, bada, ezin daiteke inolako joera argirik antzeman, eta corpus bakoitzaren ezaugarri zehatzen menpeko zerbait dela ondoriozta daiteke.

Hizpide den taulako daturik aipagarriena, ordea, test-multzoen arteko aldea da, batez ere IVAP eta Europarleko bien artekoa. Honela, lehenengoan itzulitako esaldien luzera bikoi-

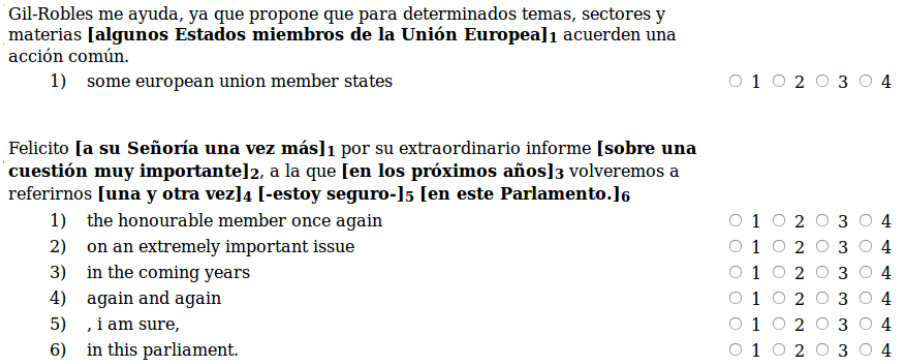
	Esaldi osoak	Entitateak	Zatiak		
			GIZA++	Berkeley (HMM)	Berkeley (sint.)
IVAP (domeinuan)	15,10	11,31	6,66	8,09	-
Europarl (domeinuan)	7,02	9,39	5,04	5,10	5,15
Europarl (domeinuz kanpo)	6,00	-	4,71	4,74	4,71

6.4 Taula: Itzulpen partzialen batezbesteko token kopurua

tza baino handiagoa da beste biekin alderatuz gero (15 token 6-7 tokenen aurrean), eta zatien artean ere alde nabarmenak daude (8 token 5 bat tokenen aurrean). Europarleko bi test-multzoen artean, berriz, domeinukoa domeinuz kanpokoaren gainetik aurkitzen da kasu guztietan, baina aldeak txikiagoak dira kasu honetan. Aipatzekoa da, halaber, 6.1 taulan jaso bezala test-multzoko esaldien batezbesteko luzerari dagokionean hain justu ere aurkakoa gertatzen dela, IVAPekoak direlarik laburrenak (20-21 token esaldiko) Europarleko domeinukoen (28 token esaldiko) eta Europarleko domeinuz kanpokoen (30-31 token esaldiko) ondoren, aldeak are esanguratsuagoak egiten dituena. Gauzak honela, testua geroz eta errepikakorragoa izan orduan eta esaldi gehiago berrerabili ahal izateaz gain haien luzera ere askoz handiagoa suertatzen dela ondoriozta daiteke, proposaturiko sistemaren inpaktua areagotuz baldintza horietan. Bestalde, esaldi osoen kasuan ez ezik EBMT aurreprozesuaren bidez itzulitako zatien luzeran ere testuaren errepikakortasun-mailaren arabera alde handiak daudela ikusten da, testua geroz eta errepikakorragoa izan orduan eta zati luzeagoak itzul daitezkeelarik, proposaturiko itzulpen partzialak erabilgarriagoak ere izan daitezkeela pentsarazten duena.

6.2.2 EBMT aurreprozesuaren eskuzko ebaluazioa

Behin EBMT aurreprozesuaren azterketa kuantitatiboa eginda, azpiatal honetan aurkezten den esperimentu-saiakeraren helburua haren bidez lortutako itzulpen partzialen kalitatea neurtzea da. 6.1.2 atalean ikusi bezala, eskuzko ebaluazio bat burutu da horretarako, IVAP eta Europarl corpusetako domeinuko test-multzoetatik 100 esaldi ausaz hartu eta bosna laguni bertako itzulpen partzialen egokitasuna 1 (itzulpen okerra) eta 4 (itzulpen zuzena) arteko eskala batean baloratzeko eskatuz. Bide horretan, orokortze-urrats guztiak aplikatu dira, lerrokatzaile modura GIZA++ erabiliz itzulpen automatikoaren arloan aukerarik ohikoena denez gero. Ebaluazio-irizpideen artean balorazio hau testuinguruan egiteko eskatu da, esperimentuaren helburua proposaturiko itzulpen partziala kasu jakin bakoitzean egokia den ala ez jakitea baita. Horretarako, jatorrizko esaldiak oso-osorik eman dira, hainbat zati markatuz eta zati horietako bakoitzarentzat proposaturiko itzulpen partziala



6.6 Irudia: Eskuzko ebaluaziorako formularioaren itxura

	1	2	3	4	Batezbestekoa
1. ebaluatzailea	2 (%1,56)	5 (%3,91)	19 (%14,84)	102 (%79,69)	3,73
2. ebaluatzailea	5 (%3,91)	4 (%3,13)	18 (%14,06)	101 (%78,91)	3,68
3. ebaluatzailea	11 (%8,59)	8 (%6,25)	9 (%7,03)	100 (%78,13)	3,55
4. ebaluatzailea	13 (%10,16)	14 (%10,94)	25 (%19,53)	76 (%59,38)	3,28
5. ebaluatzailea	19 (%14,96)	23 (%18,11)	21 (%16,54)	64 (%50,39)	3,02
Batezbestekoa	10 (%7,82)	10,8 (%8,45)	18,4 (%14,4)	88,6 (%69,33)	3,45

6.5 Taula: Eskuzko ebaluazioaren emaitzak IVAP corpusean (domeinuan)

zein den zehaztuz. Adibide modura, 6.6 irudiak horretarako erabili den formularioaren zati bat erakusten du.

IVAPi dagozkion emaitzak 6.5 taulan jasotzen dira eta Europarli dagozkionak, berriz 6.6 taulan. Ikus daitekeenez, errenkada bakoitzean ebaluatzaile baten balorazioak biltzen dira, zutabez zutabe 1, 2, 3 eta 4 puntuko zenbat balorazio eman dituen jasoz eta parentesi artean totalarekiko dagokion ehunekoa adieraziz. Azken zutabean, berriz, era horretara eskuraturiko batezbesteko puntuazioa ematen da eta azken errenkadan, bukatzeko, ebaluatzaile guztien arteko batezbestekoa. Aipatzekoa da, halaber, ebaluatzaileak ez datozela bat bi test-multzoetan eta, honenbestez, IVAPeko eta Europarleko 1. ebaluatzailea, adibidez, ez direla zertan pertsona bera izan behar.

Emaitzak aztertuz, orokorrean puntuazioak oso altuak direla ikus daiteke bi test-multzoetan, IVAPen kasuan 3,45eko batezbestekoa eskuratuz eta Europarlen kasuan, berriz, 3,39koa. Batezbestekoak elkarrengandik oso hurbil egonagatik, baina, balorazioen aldakortasunean alde handiagoak daude, Europarli dagokionean emaitza oso trinkoak eskuratzen direlarik (batezbesteko guztiak 3,30 eta 3,49 artean daude) eta IVAPen kasuan, berriz, gorabehera aipagarriak ematen direlarik (1. ebaluatzailearen batezbesteko puntuazioa 3,73koa da 5. ebaluatzailearen 3,02koaren aurrean). Fenomeno hau argiago ikus daiteke 6.7 eta 6.8 irudietan, non hurrenez hurren IVAP eta Europarl corpusentzako jasoriko balorazioak gra-

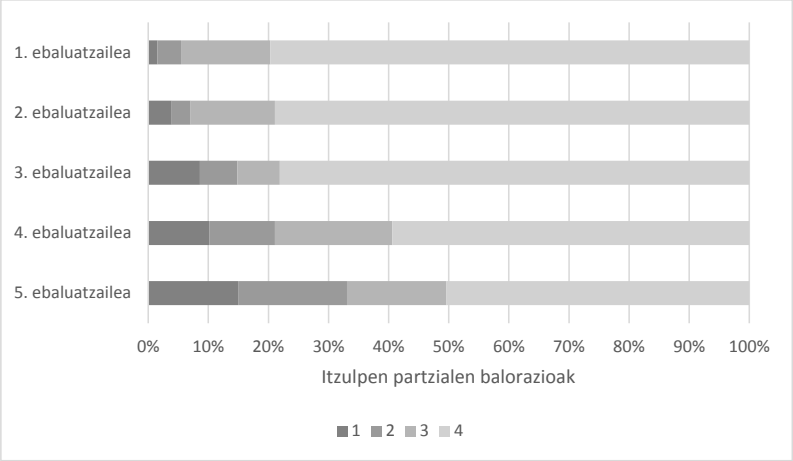
	1	2	3	4	Batezbestekoa
1. ebaluatzailea	8 (%4,79)	11 (%6,59)	40 (%23,95)	108 (%64,67)	3,49
2. ebaluatzailea	14 (%8,38)	11 (%6,59)	28 (%16,77)	114 (%68,26)	3,45
3. ebaluatzailea	11 (%6,71)	20 (%12,2)	25 (%15,25)	108 (%65,85)	3,40
4. ebaluatzailea	16 (%9,58)	14 (%8,38)	38 (%22,75)	99 (%59,28)	3,32
5. ebaluatzailea	17 (%10,24)	20 (%12,05)	25 (%15,06)	104 (%62,65)	3,30
Batezbestekoa	13,2 (%7,94)	15,2 (%9,15)	31,2 (%18,77)	106,6 (%64,14)	3,39

6.6 Taula: Eskuzko ebaluazioaren emaitzak Europarl corpusean (domeinuan)

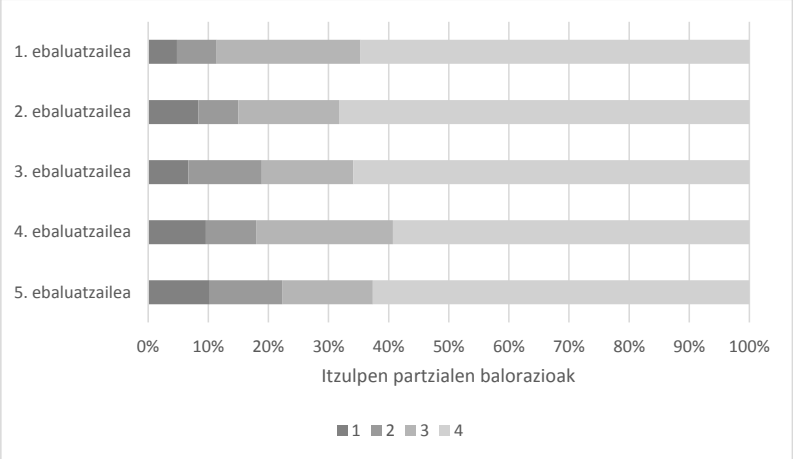
fikoki erakusten baitira. Ikus daitekeenez, bigarren kasuan barra guztiek antzeko itxura dute, baina lehenari erreparatuz gero ezberdintasun handiak antzeman daitezke, batez ere lehen hiru ebaluatzaileen eta azken bien artean. Honen zergatia ulertu nahirik, balorazioetan muturreko aldeak zituzten itzulpen partzialei erreparatu zaie, eta ezberdintasunen atzean ebaluatzaile batzuen gehiegizko zorroztasuna aurkitzen dela ondorioztatu. Izan ere, 1 ala 2 puntu jaso zituzten itzulpen partzial asko eta asko esaldi osoei dagozkie, gehien-gehienetan proposaturiko itzulpena erreferentziazkoaren berdin-berdina izanik. Bistakoa denez, proposaturiko itzulpena erreferentziazkoaren berdina denean puntuazio altuena jasotzea espero zitekeen, eta hizpide diren kasuetan hori hala ez izatea ondorengo bi arrazoiei dagokiela uste da:

- **Lan-hitzarmenetako hizkuntza tekniko eta korapilatsua.** Hainbat ebaluatzailek testua ulertzeko zailtasunak izan zituztela aitortu zuten, ez baitzeuden ohituta halakoak euskaraz irakurtzera. Gauzak honela, kasu batzuetan berez zuzenak ziren itzulpenak okerrak zirela edo, gutxienez, hobeto egin zitezkeela uste izango zutela pentsatzen da.
- **Itzulpen automatikoaren inguruko ebaluatzaile batzuen ezjakintasuna.** Guztien artean baloraziorik txarrenak eman dituen 5. ebaluatzaileak, adibidez, Euskal Filologiako ikasketak ditu eta euskara oso ondo menderatzen du, baina ez da sekula arlo honetan lanean aritu. Hori dela eta, gehiegizko zorroztasunarekin jokatuko zuela uste da, topatzen zituen akatsak itzulpen automatikoaren esparruan espero zitekeena baino gogorrago zigortuz.

Gauzak honela, benetako emaitzak datu gordinek islaturikoak baino zertxobait hobeak direla uste da, IVAPen kasuan bereziki esan bezala. Honen erakusgarri, aipatzekoa da bi test-multzoetako 2. ebaluatzailea itzulpen automatikoaren arloan aritzen den itzultzaile profesional bat dela, arlo honetan esperientzia handikoa. Ikus daitekeenez, bere puntuazioak batezbestekoak baino hobexeagoak dira, eta itzulpen partzialen benetako kalitatearen adierazle egokiagoak izan daitezkeela uste da.



6.7 Irudia: Eskuzko ebaluazioaren emaitzak IVAP corpusean (domeinuan)



6.8 Irudia: Eskuzko ebaluazioaren emaitzak Europarl corpusean (domeinuan)

Kontuak kontu, arestian esaten zenaren harira emaitzak oso positiboak dira batezbestekoak hartuta ere. Honela, puntuazio orokorrak altuak izateaz gain (ia-ia 3,5 lauren gainean bi kasuetan) balorazioen banaketatik ere ondorio positiboak ateratzen dira. Zehatzagoak izanez, itzulpen partzialen bi herenen inguru guztiz zuzenak direla ikus daiteke batezbestean, erabat okerrak direnen portzentajea %8aren azpitik kokatzen delarik. 3 eta 4ko balorazioak bilduz, berriz, itzulpen partzialen %80 baino gehiago zuzena ala ia zuzena dela ikusten da bi test-multzoetan. Honenbestez, EBMT aurreprozesuaren bidez emandako itzulpen partzialak kalitate altukoak direla ondoriozta daiteke, 3.3 atalean planteatzen zenaren harira proposaturiko sistemak benetako hobekuntza bat ekar dezan ezinbestekoa dena. Modu honetara, entitate eta zatien bidezko orokortzea itzulpenen kalitatean galera

handiegirik eman gabe gauzatzen da, proposaturiko sistemaren inpaktua modu positiboan areagotzea posible eginez.

Bestalde, IVAPen emaitzak Europarlenak baino zertxobait hobeak badira ere (batez ere, ebaluatzaile jakin batzuekin aipatzen ziren gorabeherak kontuan hartuz), orokorrean nahiko parekoak direla ikus daiteke. Itxura guztien arabera, ezberdintasunen arrazoi nagusia 6.2.1 atalean ikusi bezala IVAPen kasuan esaldi oso askoz gehiago berrerabiltzen direla da, haietan akatsak egotea kasik ezinezkoa izanik. Hala eta guztiz ere, errepikakortasunmaila txikiagoko test-multzo batean lortutako emaitzak hainbeste hurbiltzea datu positiboa da, aipatzen zen bezala orokortze-mekanismoak ondo dabiltzala erakusten baitu.

Amaitzeko, zuzenak ez diren itzulpenetan zerk egiten duen huts jakin nahirik puntuazio baxuak eskuratu dituzten esaldiak banan-banan aztertu dira, egindako akats gehienak ondorengo motatakoak direla ondorioztatuz:

- **Lerrokatzean edo itzulpenen identifikazioan akatsak.** Izan ere, hitz-lerrokatzaileak ez dira tresna perfektuak, eta haien hutsegiteen ondorioz itzulpen-zati okerrak identifika daitezke batzuetan. Beste batzuetan, berriz, corpuseko itzulpenak ez dira erabat literalak, eta horrek guztiz egokiak ez diren lerrokatzeak eragiten ditu halabeharrez antzeko ondorioekin. Mota honetako akatsen adibide modura, *"de las decisiones adoptadas"* adierazpena *"decisions"* bezala itzultzen da, edota "*en materia de competencia"* esamoldea *"competition"* bezala.

- **Hartutako zatiak unitate osoak ez izatea.** Hitz-lerrokatzaileen antzera, analizatzaile sintaktikoek ere akatsak egiten dituzte, EBMT aurreprozesuaren bidez sortutako itzulpen partzialetara heda daitezkeenak. Izan ere, analisi sintaktiko okerren ondorioz bilaketa-prozesuan unitate sintaktikoak ez diren zatiak har daitezke, eta haientzako sortutako itzulpen partzialak desegokiak izatea eragin dezake horrek.

  Landu diren bi corpusetan, gainera, gaztelania zen jatorrizko hizkuntza, eta Freeling harentzat erabili den analizatzailea. Bada, gaztelaniako kontrakzioak (*de*+*el*=*del* edo *a*+*el*=*al* modukoak, adibidez) arazotsuak suertatzen dira analisi sintaktikoa egiterakoan, eta Freelingek bi aukera eskaintzen ditu haiek tratatzeko:

  – **Kontrakzioak bitan banatzea.** Hau da aukerarik zentzuzkoena sintaxiaren ikuspegitik, baina hitz-mailan banaketa desegokiak eragiten ditu. Adibidez, Freelingek *de* token bat eta *l* beste bat dela ebazten du *del* kontrakzioarentzat, eta horren ondorioz bigarren zatia bakarrik hartzen duten itzulpen partzialetan *l* letra soltea ageriko litzateke *el* artikulua beharrean. Kasu hauek behar bezala

itzultzeko, bada, kontrakzioen tratamendu berezi bat egin beharko litzateke, baina baliabide faltagatik aukera hau baztertu egin da proiektu honetan eta etorkizuneko lan modura utzi.

– **Kontrakzioak token bakartzat hartu eta analisi sintaktikoa egokitzea.** Aurreko aukera baztertuta, hau da Freelingek eskaintzen duen alternatiba bakarra, kontrakzioak bere horretan mantendu eta analisi sintaktikoa horretara egokitzen duena. Hurbilpen honen baitan, baina, kontrakzioa eta, honenbestez, haren parte diren bi partikulak, sintagma bakar batean egotera behartuta daude, hori komeni ez den kasuetan zuhaitz sintaktiko akastun bat eraikiarazten duena halabeharrez.

Ikusi ahal izan denez, analisi sintaktiko oker askoren atzean aipatu berri den arazo hau aurkitzen da. Jatorria edozein izanik ere, analisi sintaktiko okerrek eragindako akatsen adibide modura aipa daitezke *"a la expiración del"* unitate osotzat hartu izana, bere itzulpentzat *"bukatu"* proposatuz, bai eta *"que al fin y al"* zatia ere, haren itzulpen modura *"who, in the end,"* emanez.

- **Preposizioen tratamenduan akatsak.** Aurreko bi puntuen kasu partikulartzat har badaiteke ere, aipatzekoa da preposizioekin izandako katramila, ebaluatzaileei *feedback*a eskatzean ia guztiek lehen tokian aipatu izan duten puntu bat delarik. Izan ere, hainbat kasutan itzulitako zatian preposizioa ageri da baina ez, ordea, harentzat proposaturiko itzulpenean. Adibide modura, *"a los departamentos franceses de ultramar"* zatiaren itzulpen modura *"the french overseas departments"* ematen da, edota *"de la política medioambiental"* zatiarentzat *"environmental policy"*. Aipatzekoa da, hala ere, esperotako portaera beste bat izanagatik kasu batzuetan posible dela itzulpena preposiziorik gabe eraikitzea, proposaturiko itzulpen partzialaren egokitasuna itzultzaile nagusiaren eta, batez ere, harekiko integrazioaren menpe geratzen delarik.

- **Entitateen tratamenduan akatsak.** Entitateen bidezko orokortzea itzulpen partzial gehiago sortu eta, honenbestez, proposaturiko sistemaren inpaktua areagotzeko mekanismo modura planteatzen zen baina, ikusi ahal izan denez, ez du beti espero bezain ongi funtzionatzen. Topatu diren akatsak, bada, ondorengo motatakoak izan dira nagusiki:

  – **Mugen identifikazioan akatsak.** Izan ere, entitateen mugak ez dira beti modu kontsistentean definitzen bi hizkuntzetan, berez eman zitezkeen ezberdintasunez gain aipatzekoa delarik aztergai diren kasuetan entitate-ezagutzaileak

eurak zeharo ezberdinak direla. Ikusi ahal izan denez, entitate konposatuak eta zenbakizko kodeak bereziki arazotsuak suertatzen dira, hitz-lerrokatzeko gorabeherekin batera itzulpen okerrak sortzera eramaten duena batzuetan. Adibide modura, *"(A5-0368/2000)"* testu-zatiaren itzulpen modura *"(a4a5-0368/2000/97)"* proposatzen da, bistakoa denez corpuseko sarreraren batean entitateen mugak modu inkontsistentean identifikatu izanaren ondorio dena.

- **Itzulpenean akatsak.** 4.2.2 atalean ikusi bezala, entitateen lemak itzultzeko hainbat baliabide erabiltzen dira eta, haien bidez arrakastarik izan ezean, lema bere horretan uztearen alde egiten da, hiztegietan aterako ez liratekeen pertsona- eta leku-izen berezi ia guztiak itzultzeko neurri egokia dena. Ikusi ahal izan denez, baina, modu horretara hainbat entitate ez dira zuzen itzultzen edo, hobeto esanda, ez dira itzuli ere egiten, berez hala eskatzen badute ere ez baitira hiztegi elebidun edo bestelako baliabideetan ageri. Honen adibide modura, *"Organización de la Pesca en el Atlántico Noroccidental"* testu-zatiaren itzulpen bezala *"the organización de la pesca in the atlántico noroccidental"* proposatzen da, derrigor itzuli beharreko hainbat hitz gaztelaniaz uzten dituena.
- **Sorkuntzan akatsak.** 4.2.2 atalean bertan ikusi bezala, entitateen lemak itzuli ostean haien sorkuntza egin behar da, euskararen kasuan lema horiek deklinatzea eskatzen duena. Horretarako, baina, lema izen arrunt gisa (oro har, mugatzailearekin) ala izen berezi gisa (oro har, mugatzailerik gabe) deklinatu behar den jakin behar da, eta hori ez da batere erraza. Izan ere, kasuan kasu modu batera ala bestera egitea da zuzena, hala nola *Gurutze Gorri* izen arrunt modura deklinatzen delarik (*Gurutze Gorri**an***, mugatzailearekin), eta *Aizkorri*, berriz, izen berezi modura (*Aizkorri**n***, mugatzailerik gabe). Kontuan izan behar da, beti ere, corpuseko jatorrizko formari erreparatzeak ez duela balio, adibide berberei eutsiz posible baita *Gurutze Gorri* entitateak *Aizkorri* ordezkatzea ala alderantziz. Hori dela eta, proiektu honetan irizpide global bat jarraitzearen alde egin da, anbiguotasun edo zalantzarik den kasu guztietan izen arrunt bezala deklinatuz beste aukera baino sarriago agertzen delakoan. Erabaki honen ondorioz, baina, hainbat akats ere egiten dira, hala nola *"con el País Vasco"* segida *"Euskadiarekin"* bezala itzultzen delarik *"Euskadirekin"* beharko lukenean.

- **Zatien berrerabilera okerra.** Proposaturiko sistemaren oinarri-oinarrizko planteamendutik datorren arazo bat da hau, testuinguru jakin batean zuzena den itzulpen

bat ez baita zertan egokia izan behar, printzipioz, beste testuinguru batean. Hala eta guztiz ere, emaitzak aztertuz era honetako akatsen presentzia oso eskasa dela ikusi ahal izan da, gainerako arazoen alboan erabat mespretxagarri egiteraino. Edozelan ere, haren zantzuak aurki daitezke hainbat itzulpenetan eta, adibide modura, *"el registro y publicación"* zatiaren itzulpentzat *"erregistratzeko eta argitaratzeko"* proposatzen da, esaldiaren egituraren arabera desegokia izan daitekeena, eta beste horrenbeste esan daiteke *"y en Francia también"* eta haren itzulpentzat proposatzen den *"and they exist in france too"* zatiari buruz ere.

6.2.3 Sistema osoaren BLEU bidezko ebaluazio automatikoa

EBMT aurreprozesuaren azterketa kuantitatiboa eta eskuzko ebaluazioa burututa, hirugarren eta azken esperimentu-saiakera honen helburua sistema osoaren portaera aztertzea da, EBMT aurreprozesua ez ezik integrazioa ere aintzakotzat hartuz. Esan bezala, ebaluazio automatiko bat burutzea erabaki da horretarako, test-multzoak proposaturiko sistemaren bidez osoki itzuli eta dagokien BLEU puntuazioa kalkulatuz. Esperimentuaren diseinuan zehaztu bezala, bide horretan corpus, orokortze-urrats, lerrokatze-ezarpen eta itzultzaile nagusi ezberdinekin aritu da bakoitzaren eragina zehazki zein den ezagutzeko asmoz.

IVAPerako eskuratutako emaitzak 6.7 taulan bildu dira eta Europarlerako lortutakoak berriz, 6.8 eta 6.9 tauletan hurrenez hurren domeinuko eta domeinuz kanpoko test-multzoentzat. Kasu guztietan errenkadaz errenkada itzultzaile nagusi ezberdinentzako balioak ematen dira, lehendabizi RBMT sistema batekin lortutakoak (Matxin gaztelania-euskararentzat eta Apertium gaztelania-ingelesarentzat) eta ondoren Moses SMT sistemak emandakoak integrazio-estrategia ezberdinak erabiliz. Zutabeka, berriz, hiru orokortze-urratsak zehazten dira, zatien kasuan lerrokatze-ezarpen ezberdinen arabera bereiziz, eta lehen errenkadan *baseline*aren puntuazioa ematen da. *Baseline* hori itzultzaile nagusiak bere baitan ematen dituen itzulpenei dagozkio kasuan kasu, inolako EBMT aurreprozesurik gabe. Balio horiek gainerakoekin alderatuz, bada, proposaturiko sistemaren benetako ekarpena zein den ikus daiteke.

Horretan hasi aurretik, baina, lerrokatze-ezarpen eta Mosesen integrazio-estrategia optimoak identifikatzea komeni da lehendabizi ondorengo azterketa haien konfigurazio egokienera mugatu ahal izateko. Honela, lehenari dagokionez lerrokatze-ezarpen ezberdinek test-multzo bakoitzean duten eragina argiago ikusteko 6.9 eta 6.10 irudiak sortu dira, hurrenez hurren RBMT (Matxin/Apertium) eta SMT (Moses inclusive integrazio-estrategiarekin) eskuratutako BLEU puntuazioak grafikoki erakusten dituztenak. Ikus dai-

		Baseline	Esaldi osoak	Entitateak	Zatiak	
					GIZA++	Berkeley (HMM)
Matxin		0.0498	0.3350	0.3330	0.2977	0.3168
Moses	Exclusive	0.3368	0.4478	0.4460	0.4395	0.4441
	Constraint	0.3368	0.4479	0.4460	0.4407	0.4463
	Inclusive	0.3368	0.4483	0.4472	0.4528	0.4593

6.7 Taula: BLEU puntuazioak IVAP corpusean (domeinuan)

		Baseline	Esaldi osoak	Entitateak	Zatiak		
					GIZA++	Berkeley (HMM)	Berkeley (sint.)
Apertium		0.1755	0.1786	0.1790	0.1987	0.1983	0.1796
Moses	Exclusive	0.3307	0.3307	0.3300	0.3123	0.3098	0.2711
	Constraint	0.3307	0.3307	0.3300	0.3141	0.3119	0.2839
	Inclusive	0.3307	0.3307	0.3304	0.3259	0.3251	0.2989

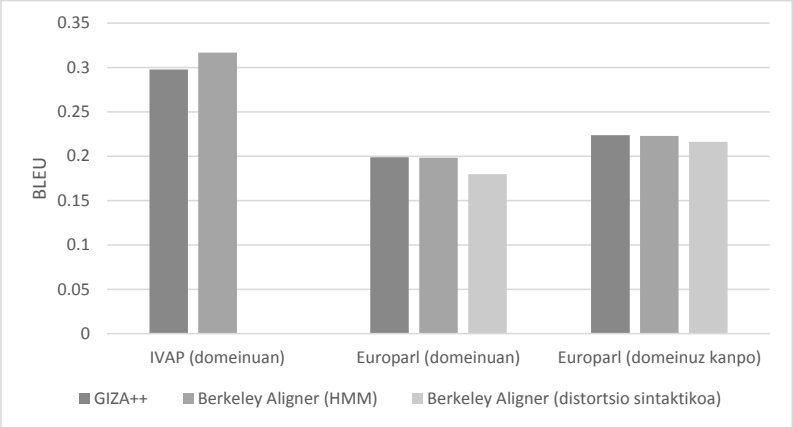
6.8 Taula: BLEU puntuazioak Europarl corpusean (domeinuan)

tekeenez, bai SMT bai eta, batez ere, RBMT sistemak erabiliz, Berkeley Alignerren HMM modua modu argian gailentzen zaio GIZA++ lerrokatzaileari IVAPen kasuan. Europarleko bi test-multzoetan, berriz, bien arteko aldea askoz txikiagoa da, baina GIZA++ lerrokatzailea nagusituz oraingoan. Berkeley Alignerren distortsio sintaktikoa, azkenik, argi eta garbi emaitzarik okerrenak ematen dituena da. Gauzak honela, Berkeley Alignerren HMM modua da, orotara, lerrokatze-ezarpen egokiena dirudiena.

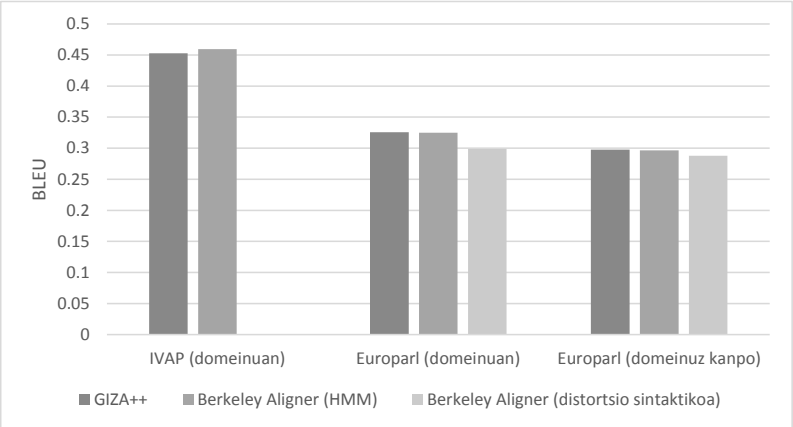
Mosesen integrazio-estrategiei dagokienez, ezarpen bakoitzak Berkeley Alignerren HMM modua erabiliz ematen dituen BLEU puntuazioak grafikoki islatuz 6.11 irudia sortu da. Grafikoak erakutsi bezala, exclusive eta constraint aukeren artean apenas dago alderik, kasu guztietan lehena nagusitzen bada ere oso tarte txikiarekin egiten baitu. Haien aurrean, ordea, inclusive moduak emaitza nabarmenki hobeak ematen ditu, domeinuko bi test-multzoen kasuan puntu eta erdi inguruko aldea eskuratuz eta domeinuz kanpokoan, berriz, ia puntu erdikoa. Honenbestez, proposaturiko itzulpen partzialak beti erabiltzera behartu beharrean sintagma-taulako gainerako sarrerekin lehiarazi eta baztertuak izateko aukera onartzeak emaitza nabarmenki hobeak ematen dituela ondoriozta daiteke, eta hau izan da, honenbestez, lehenetsi den aukera. Zergatien inguruan hausnartuz, honen atzean ondorengo bi arrazoiak egon daitezkeela uste da:

		Baseline	Esaldi osoak	Entitateak	Zatiak		
					GIZA++	Berkeley (HMM)	Berkeley (sint.)
Apertium		0.2173	0.2173	0.2173	0.2235	0.2227	0.2161
Moses	Exclusive	0.2984	0.2982	0.2982	0.2944	0.2929	0.2825
	Constraint	0.2984	0.2982	0.2982	0.2951	0.2932	0.2851
	Inclusive	0.2984	0.2982	0.2982	0.2979	0.2967	0.2878

6.9 Taula: BLEU puntuazioak Europarl corpusean (domeinuz kanpo)



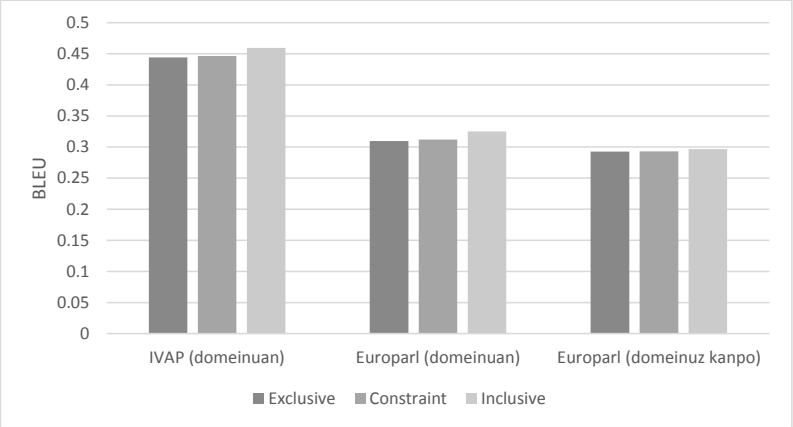
6.9 Irudia: BLEU puntuazioak lerrokatze-ezarpenen arabera Matxin/Apertium erabiliz



6.10 Irudia: BLEU puntuazioak lerrokatze-ezarpenen arabera Mosesen inclusive modua erabiliz

- **Itzulpen partzialetako akats larrienak hauteman eta baztertzeko gai izatea.** Izan ere, hizkuntza-ereduak probabilitate oso baxua esleituko lieke, hala nola, arestian aipatzen ziren *"the organización de la pesca in the atlántico noroccidental"* moduko itzulpen akastunei, non hitz gehienak jatorrizko hizkuntzan uzten baitira. Horren aurrean, bada, SMT sistema bera itzulpen egokiagoak sortzeko gai izan beharko litzateke, probabilitate handiago bat jasoko luketenak.

- **Integrazio arazoak daudenean itzulpen partzialak baztertzeko gai izatea.** Izan ere, bistako da kasu batzuetan saihestezina izango dela itzulpen-zati bat testu luzeago batean txertatzean galera bat ematea. Hori dela eta, proposaturiko itzulpen partziala zuzena izanagatik litekeena da berau berrerabiltzea komeni ez izatea, izan itzultzaile nagusia bera ere zati hori ondo itzultzeko gai delako ala izan, besterik



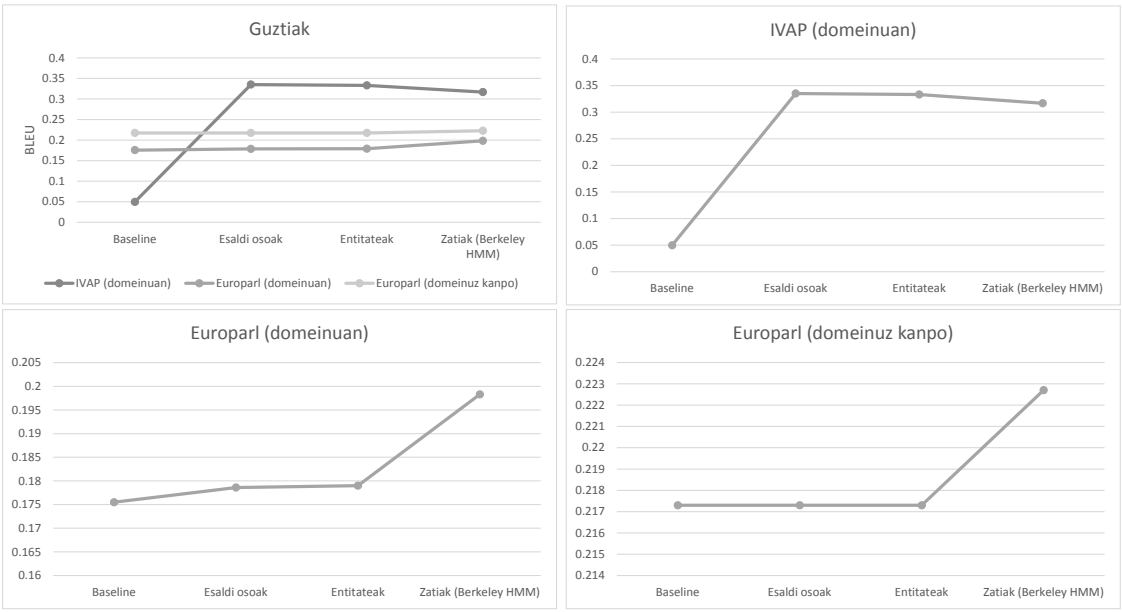
6.11 Irudia: BLEU puntuazioak Mosesen integrazio-estrategiaren arabera Berkeley Aligner (HMM) erabiliz

gabe, integrazioan emango litzatekeen galera medio mesede beharrean kalte egingo lukeelako.

Behin lerrokatze-ezarpen eta Mosesen integrazio-estrategia optimoak finkatuta, azter ditzagun, orain, orokortze-urratsen araberako BLEU puntuazioak Matxin eta Apertium RBMT sistemekin hasita. Lan hau errazteko, 6.12 irudiko grafikoa sortu da, test-multzo bakoitzean BLEU puntuazioaren bilakaera erakusten duena. IVAPekin ematen diren aldeak beste bienak baino nabarmenki handiagoak izanik, test-multzo guztiekin gertatzen dena argiago ikusteko banakako grafikoak ere ematen dira bertan guztiak batera daudenaz landa.

Grafikoei erreparatuz, RBMT itzultzaileekin batera erabiltzean proposaturiko sistemak eragin positiboa duela beti ikus daiteke. IVAPen kasuan, hobekuntza hau izugarrizkoa da, BLEU puntuazioa seikoitza baino are handiagoa izatera pasatzen baita, balio absolututan 0,267ko aldea eskuratuz. Europarleko domeinuko test-multzoan ere aldea benetan aipagarria da, bi puntu BLEUko muga gainditzera iritsiz. Europarleko domeinuz kanpoko test-multzoan, azkenik, ezberdintasunak xumeagoak dira, baina aipagarriak izaten jarraitzen dute, puntu erdi BLEU baino gehixeagoko hobekuntza eskuratzen baita. Gauzak honela, espero zitekeen bezala testuaren errepikakortasuna geroz eta handiagoa izan orduan eta hobekuntza handiagoa eskuratzen dela ondoriozta daiteke, emaitzak, beti ere, oso-oso positiboak direlarik.

Orokortze-urrats bakoitzaren ekarpen zehatza aztertuz, berriz, daturik harrigarriena IVAPen kasuan entitateak eta, batez ere, zatiak erabiliz ematen den galerarena da, ia bi puntu BLEUtara iristen dena orotara. Honenbestez, benetako hobekuntza esaldi osoak bere



6.12 Irudia: BLEU puntuazioak orokortze-urratsen arabera Matxin/Apertium erabiliz

horretan berrerabiliz, hau da, ohiko itzulpen-memoria batek lukeen portaera berberaren bidez eskuratzen dela esan daiteke, orokortze-urratsek mesede beharrean kalte egiten dutelarik test-multzo jakin horretan. Itzulpenak gertuagotik aztertuz, ondorengo arrazoiengatik gertatzen dela hori ondorioztatu da:

- **Lan-hitzarmenenen domeinuan Matxinek duen portaera eskasa.** Izan ere, *baseline*an bertan Matxinen itzulpenak oso txarrak direla ikus daiteke, bost puntu BLEU baino gutxiago eskuratuz eta, lehen begirada batean ageriko egiten denez, erabat ulertezina den irteera bat emanez. Honen arrazoia itzuli beharreko testuaren beraren izaerari dagokiola uste da, lan-hitzarmenetakoa bezalako hizkuntza korapilatsua era honetako itzultzaileentzat bereziki zaila suertatzen baita. Hori dela eta, jatorrizko itzulpenak zeharo aldrebesak dira gehienetan, eta hainbat zati aldatzen hasita gauzak are gehiago nahastea baino ez da lortzen.

- **Integrazio-arazoak.** Izan ere, euskararen eta gaztelaniaren arteko itzulpen-dibergentziak hain handiak izanik, itzulpen partzialak bukaerako testuan txertatzea bereziki zaila suertatzen da eta, gehiegi landu ez den alderdi bat izanik, hainbat arazorekin topatu da zentzu honetan. Izandako buruhausteen adibide modura, Matxinek itzulpen partzialak emateko baliatzen ziren HTML etiketak bikoiztea erabakitzen du batzuetan, testuaren itxurari lotutako alderdiekin (letra mota, estekak...) zentzua izan balezake ere, kasu honetan inola ere komeni ez dena. Arazo horri aurre egiteko,

etiketa bikoiztuak hauteman, ordezkapena lehenengoan burutu, eta gainerakoen testua erabat baztertzeko mekanismo bat inplementatu da baina, agerikoa denez, era honetako gorabeheren ondorioz azken itzulpenek kalitate-galera nabarmena paira lezakete. Matxinentzako integrazio-mekanismo egokiago bat garatzea, bada, etorkizuneko lan bezala geratzen da.

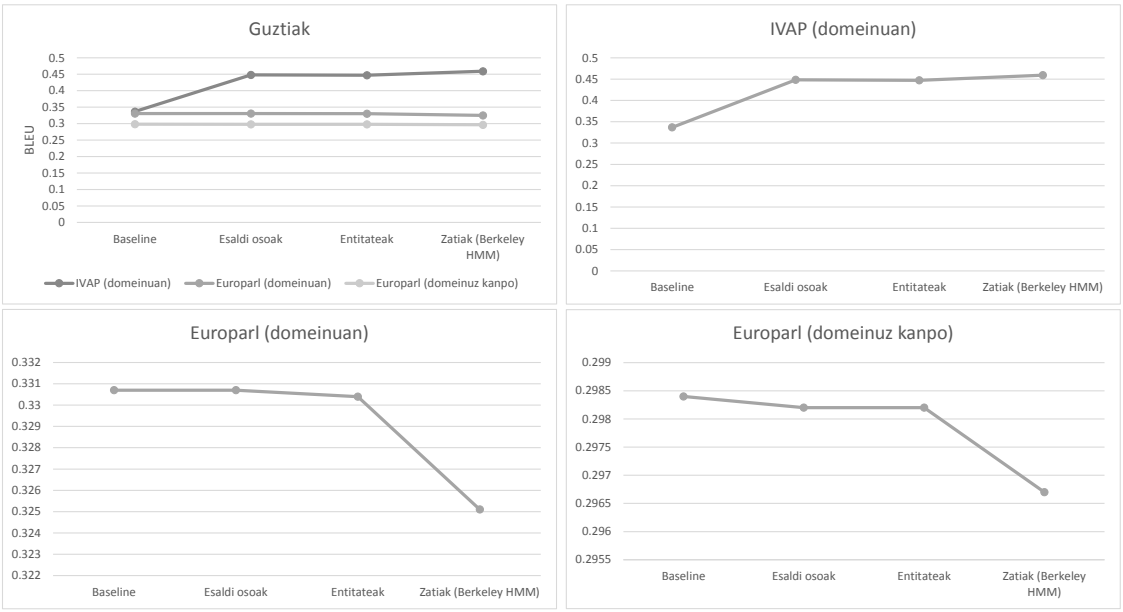
Gauzak honela, arazoa Matxin eta harekiko integrazioa dela uste da eta ez, edozein kasutan, EBMT aurreprozesua bera, esperimentuan zehar eskuratutako gainerako emaitzek (gainontzeko BLEU puntuazioek edo itzulpen partzialen eskuzko ebaluazioak, adibidez), hala pentsarazten baitute.

Europarleko bi test-multzoetan, berriz, orokortze-urratsekiko BLEUren bilakaera espero zitekeena da, irabazirik handiena zatien bidezko orokortzearen bidez ematen delarik. Zehatzagoak izanez, domeinukoan esaldiak bere horretan berrerabiliz apenas hobetzen dira hiru hamarren BLEU, eta entitateen bidezko orokortzearen bidez lau ehunen baino ez. Ia bi puntutara iristen den gainerako hobekuntza, bada, zatien bidezko orokortzeari esker eskuratzen da, eta domeinuz kanpoko test-multzoan hobekuntza guzti-guztia (puntu erdi BLEU pasatxo) modu honetara dator. Gauzak honela, espero zitekeen bezala testuaren errepikakortasuna geroz eta txikiagoa izan orduan eta esaldi oso gutxiago berrerabil daitezkeela ondoriozta daiteke, zatien bidezko orokortzearen protagonismoa areagotuz.

SMT sistemari eta, zehatzagoak izanez, Moses eta bere inclusive integrazio-estrategiari dagokionez, berriz, eskuratutako emaitzak argiago ikusteko 6.13 irudiko grafikoak sortu dira RBMT sistemenen antzera. Bertan ikus daitekeenez, oraingoan ere alderik handienak IVAPekin ematen dira, 0,1225 BLEUko hobekuntza izugarria eskuratuz orotara. Matxinekin ez bezala, gainera, zatien bidezko orokortzearen ekarpena positiboa da kasu honetan, BLEU puntu oso baten gainetik kokatuz. Honek orokortze-mekanismoak behar bezala funtzionatzen ari direnaren ideia berresten du, integrazio-estrategia egoki batekin emaitzak hobetzeko gaitasuna erakutsiz eta arestian azterturiko arazoak Matxini berari eta harekiko integrazio-gorabeherei zegozkiela ageriko eginez.

Europarleko bi test-multzoei dagokienez, berriz, emaitzak hobetu beharrean okertu egiten direla ikus daiteke. Galerak, hala ere, ez dira handiegiak, bi hamarrenen azpitik kokatuz domeinuz kanpokoan eta puntu erdi inguruan domeinukoan. Orokortze-urratsei erreparatuz, guztiek ere eragin negatiboa dutela ikusten da, baina beherakadarik nabarmenena zatiekin dator hala ere. Egindako itzulpenak sakonago aztertuz, honako bi arrazoiengatik gertatzen dela hori uste da:

- ***Baseline*aren maila altua.** Ikusi ahal izan denez, Mosesek oso emaitza onak ema-



6.13 Irudia: BLEU puntuazioak orokortze-urratsen arabera Mosesen inclusive modua erabiliz

ten ditu gaztelania-ingelesa bikoteko bi test-multzoetan, Apertiumenak baino askozaz hobeak hala nola. Abiapuntua hain ona izanda, bada, emaitza positiboak eskuratzea oso zaila da, hobekuntza-marjina beste kasuetan baino askoz txikiagoa baita.

- **Topatutako zatiak laburregiak izatea.** 6.2.1 atalean aztertu bezala eta 6.4 taulak jasotzen duenez, Europarleko bi test-multzoetan topatzen diren itzulpen partzialak zati laburrentzat dira oro har, 5 token ingurukoak batezbeste bi kasuetan. Esaldi osoen kasuan ere nahiko motzak dira, 6-7 tokenetako batezbestekoa emanez. Bada, itxura guztien arabera, Moses bezalako sintagmetan oinarrituriko SMT sistemak tamaina horretako n-gramak modu egokian maneiatzeko gai dira eta, honenbestez, EBMT aurreprozesuaren bidez iradokitako itzulpen partzialek ez dute benetako hobekuntzarik suposatzen kasurik gehienetan, integrazio gorabeherekin batera mesede beharrean kalte egitea eragiten duena.

Benetako arazoa, noski, bigarren puntua da, aurrenekoarekin elkartzean emaitzak abiapuntukoak baino okerragoak izatea eragiten duena. Hori konpontzeko neurri zuzenena, halaber, token kopuru minimoa handitzea da argi eta garbi. Zentzu honetan egindako aurretiazko esperimentuek emaitza positiboak eman zituzten, murriztapen zorrotzagoekin BLEU puntuazioa ehuneko gutxi batzuetan hobetzera iritsiz. Nolanahi ere, baldintza horietan proposaturiko sistemaren inpaktua hain zen baxua, ezen bide hori erabat baztertzea erabaki baitzen momentu hartan. Gauzak honela, halako inguruneetan proposaturiko sis-

tema ez dela ekarpen handiegirik egiteko gai ondoriozta daiteke eta, izatekotan, murriztapen oso zorrotzekin batera erabili beharko litzatekeela. Baliabide faltagatik, baina, horren komenigarritasuna aztertu eta murriztapen egokienen inguruan ikertzea etorkizuneko lan bezala uzten da.

6.3 Emaitzen interpretazioa

Aurreko atalean esperimentuaren emaitzak aurkeztu eta aztertzen baziren, oraingo honen helburua urrats bat haratago joan eta emaitza horien atzean zer aurkitzen den hausnartzea da. Horretarako, hiru esperimentu-saiakerak eta haien bidez lortutako datu guztiak modu globalean aztertu dira, ondorengo ondorioak ateraz:

- Oro har, proposaturiko sistemak oso emaitza onak ematen ditu: bere inpaktua handia da, eta kasurik gehienetan itzultzaile nagusiak bere baitan egiten duena nabarmenki hobetzea lortzen du. Gauzak honela, hobetu beharreko hainbat alderdi identifikatu badira ere itzulpen partzialen kopuru eta kalitatearen balorazio oso positiboa egiten da, eta esan bezala integrazioaren ostean ere sistemaren portaera oso ona da. Hala eta guztiz ere, ikusi den bezala emaitza hauek hainbat baldintzaren arabera oso aldakorrak dira, proposaturiko sistemaren erabilera-kasu jakin batzuentzat beste batzuentzat baino egokiagoa eginez jarraian zehazten den bezala:
  - Geroz eta errepikakorragoa izan itzuli beharreko testua, orduan eta hobea da proposaturiko sistemaren portaera, bai inpaktuaren aldetik bai eta bukaerako itzulpenen kalitatearen aldetik ere. Hori dela eta, domeinuan domeinuz kanpo baino emaitza hobeak eskuratzen dira, batez ere lan-hitzarmenen antzera testuak berezko izaera errepikakorra baldin badu. Edozein kasutan, baldintza txarrenetan ere, hau da, domeinuz kanpo, proposaturiko sistemak aintzakotzat hartzeko moduko ekarpen positiboa egin dezakeela ikusi da.
  - Orokorrean, sistemaren portaera hobea da RBMT sistemekin SMT sistemekin baino, batez ere testuaren errepikakortasun-maila baxua denean. Izan ere, RBMT sistemen kasuan proposaturiko itzulpen partzial gehienak itzultzaile nagusiak berak egingo lituzkeenak baino hobeak dira eta, honenbestez, eragin positiboa dute. SMT sistemak, berriz, entrenamendu-corpusetik ikasitako n-gramak oso ondo tratatzeko gai dira tamaina-jakin batera arte eta, ondorioz, proposaturiko itzulpen partzial labur gehienek ez dute benetako hobekuntzarik

suposatzen, aurkakoa baizik. Datuek erakusten dutenez, testua geroz eta errepikakorragoa izan orduan eta zati luzeagoak itzul ditzake EBMT aurreprozesatzaileak, RBMT eta SMT sistemen arteko aldea faktore horren menpe egotea eragiten duena. Ikusi ahal izan denez, gainera, errepikakortasun-maila txikia denean proposaturiko sistema kaltegarria izatera irits daiteke SMT sistemekin. Baldintza horietan, bada, eginiko lana ez da lagungarriegia suertatzen eta, izatekotan, murriztapen zorrotzagoak ezartzeko aukera aztertu beharko litzateke. RBMT eta SMT sistemen inguruko gogoetarekin amaitzeko, aipatzekoa da eginiko esperimentuan bigarrenen integrazio-mekanismoa lehenena baino askoz egokiagoa zela, RBMT sistementzat integrazio-estrategia landuagoren bat erabiliz gero bien arteko aldea are eta handiagoa izatea espero zitekeelarik.

- Lerrokatze-ezarpenei dagokienez, aukerarik onena Berkeley Alignerren HMM modua dela ikusi da argi eta garbi. Izan ere, aukera hori da kasu guztietan inpakturik handiena duena, bai eta orotara emaitza hoberenak ematen dituena ere Europarlen kasuan oso gutxigatik GIZA++ gailentzen bazaio ere. Hori gutxi balitz, aipatu bezala garaturiko sisteman bigarren tokian geratzen den GIZA++ baino askoz hobeto integratzen da, entrenamendu-prozesua dezente erraztuz. Distortsio sintaktikoaren aukerari dagokionez, berriz, paperaren gainean ezin aproposagoa bazirudien ere praktikan emaitza oso txarrak eman ditu. Amaitzeko, aipatzekoa da proposaturiko sistemak lerrokatze-ezarpenekiko dezente sentikorra dela erakutsi duela, batez ere gaztelania-euskara bikotearen kasuan.

- Itzultzaile nagusiarekiko integrazioari dagokionez, alderdi honek garrantzi handia duela ikusi ahal izan da, proposaturiko itzulpen partzialak baztertzeko aukera ematearen komenigarritasunaz jabetuz. Honela, Mosesen kasuan proposaturiko itzulpen partziala sintagma-taulako gainerako sarrerekin lehiarazten duen inclusive moduak ematen ditu emaitzarik onenak, neurri batean garaturiko sistemaren gabeziak estali eta EBMT aurreprozesatzaileak sorturiko itzulpenak erabiltzea mesedegarri ez deneko kasu nabarmenenak hautemateko gai baita. Matxin eta Apertiumen kasuan, berriz, hurbilpen traketsago bat jarraitu behar izan da, besteak beste proposaturiko itzulpen partzialak beti ontzat ematen zituena, eta bukaerako emaitzetan eragin negatiboa izan duela usten dena, batez ere lehenari dagokionean.

7. KAPITULUA

Ondorioak eta etorkizuneko lana

Behin egin beharrekoak eginda, kapitulu honetan proiektuan zehar egindako lanari buruz hausnartzen da. Honela, lehen atal batean ateratako ondorioak azaltzen dira, bai proiektu-mailakoak bai eta maila pertsonalekoak ere. Bigarren atal batean, berriz, etorkizunean aztertzeko interesgarriak liratekeen lan-lerroak aurkezten dira.

7.1 Ondorioak

Egindako lanaz hausnartuz ateratako ondorioak bitan banatu dira: lehen azpiatalean proiektu-mailakoak azaltzen dira eta, bigarrenean, maila pertsonalekoak.

7.1.1 Proiektuko ondorioak

Proiektu honetan EBMT tekniken bidez itzulpen partzialak sortzen dituen aurreprozesu batean oinarritutako itzulpen automatikorako hibridazio-mekanismo bat garatu da. Corpus paraleloko sarrerak bere horretan berrerabiltzeaz gain, proposaturiko sistema entitateen eta esaldia baino txikiagoak diren unitate sintaktikoen bidez haiek orokortzeko gai da, itzulpen partzialen indizea handituz hain kalitatean eragin minimoarekin. Era berean, oso arina eta eskalagarria izan dadin diseinatua izan da, eta inplementazio modular, hedagarri eta eraginkor bat eskaini zaio.

Eraikitako sistema Stanford CoreNLP, Freeling eta Eustagger analizatzaileekin integratu

da, hainbat hizkuntzekin lan egiteko aukera ematen duena, besteak beste proiektuan bertan jorratu diren euskara, gaztelania eta ingelesarekin. Lerrokatzaileei dagokienez, GIZA++ eta Berkeley Alignerrekin bateragarria da, azkenaren kasuan ohiko HMM hurbilpenaz gain distortsio sintaktikoaren erabilera ere aztertu delarik. Itzultzaile nagusi bezala, berriz, Matxin eta Apertium RBMT sistemekin nahiz Moses SMT sistemarekin txertatu da. Sistemaren diseinuari esker, gainera, osagai hauek nahiz aipatu ez diren beste batzuk oso modu errazean heda daitezke, etorkizunean baliabide eta tresna are gehiagorekin bateragarri egiteko aukera ematen duena.

Behin sistema eraikita, bere portaera hobekiago ezagutu eta ebaluatzeko esperimentu bat diseinatu eta gauzatu da. Eskuratutako emaitzak oso positiboak izan dira, eta proposaturiko sistemak hobekuntza esanguratsuak ekar ditzakeela erakusten dute. Espero zitekeen bezala, testuaren errepikakortasuna geroz eta handiagoa izan sistemaren ekarpena orduan eta handiagoa dela ikusi da, baina domeinuz kanpo eta, honenbestez, baldintza zailenetan ere hobekuntza aipagarriak eskuratu dira. Bestalde, RBMT sistemekiko hibridazioak SMT sistemekikoak baino emaitza hobeak ematen dituela ikusi da oro har baina, testua nahikoa errepikakorra bada eta, horrela, topatutako bat-etortzeak itzultzaile nagusiak modu egokian trata ditzakeen n-gramak baino luzeagoak badira, hobekuntza handiak lortu daitezkeela erakutsi da.

Gauzak honela, hasiera batean zehaztutako helburuak arrakasta handiz bete direla esan daiteke, alderdi askotan bere garaian markatutakoa baino askoz urrunago joanez gainera. Bereziki aipagarria da proiektua ikerketa-lan bat bezala planteatu bazen ere eta egindako ekarpenik handiena lerro horretan kokatuagatik, software ingeniaritzaren ikuspegitik ere balio handiko lana egin dela. Hau guztia esanda, bada, proiektuaren balorazio oso positiboa egiten da.

7.1.2 Ondorio pertsonalak

Maila pertsonalean proiektua oso aberasgarria egin zait. Izan ere, hizkuntzaren prozesamendua eta, bereziki, itzulpen automatikoa, oso arlo erakargarriak iruditu zaizkit betidanik, eta proiektu honi esker haiei buruz gehiago ikasi ahal izan dut. Horrekin batera, graduko ikasketetan zehar ikasitako kontzeptu ugari praktikan jarri ahal izan ditut, bai konputazioaren adarrari lotutakoak bai eta software ingeniaritzatik hurbilago daudenak ere. Horien guztien artean algoritmiak aipamen berezi bat merezi duela iruditzen zait, oso gustuko dudan alor bat izanda proiektu honetan hari lotutako erronka benetan interesgarriei aurre egin zaiela uste baitut. Batez ere, kateen prozesamenduaren eta atzizki-taulen

inguruan interes handia nuen lehendik ere, eta proiektu honen bidez hori guztia praktikan jarri ahal izan dut problema erreal bat ebazteko.

Bestalde, proiektu honen harira ikerketa talde batean lanean aritu naiz lehen aldiz. Bi urtetako esperientzia hau oso aberasgarria izan da niretzako, eta etorkizunean ere ikerketaren munduan egin nahiko nukeela lan ikusarazi dit. IXA taldean, gainera, lan-giro oso ona topatu dut. Malgutasun eta autonomia handiz aritzeko aukera eman didate eta, horri esker, proiektua benetan eramanerraza egin zait. Era berean, lankide apartak izan ditut, eta haien laguntzari esker pila bat ikasteaz gain oso momentu onak biziarazi dizkidate.

Laburbilduz, proiektu hau oso aberasgarria suertatu zait maila pertsonalean. Gauza mordoa ikasi ditut hari esker eta, momentu gogorrak pasatuagatik, pila bat disfrutatu dut bidean.

7.2 Etorkizuneko lana

Ondorioen atalean hausnartu bezala proiektuaren hasierako helburuak arrakastaz bete badira ere, gehiago lantzea komeniko liratekeen hainbat puntu identifikatu dira bidean, guztiak amaiera bat behar duenez alde batera utzi behar izan direnak. Jarraian, bada, etorkizunerako lan-lerro horiek zerrendatzen dira:

- **Esperimentuak hizkuntza eta corpus gehiagorekin egitea.** Izan ere, hainbeste aldagai esperimental zirenez hiru test-multzo eta bi corpus bakarrik probatu dira proiektu honetan, bakoitza hizkuntza bikote batekoa. Aurrera begira, interesgarria litzateke hizkuntza bikote jakin batzuetan zentratu eta haietako bakoitzarentzat ingurune ezberdinetako corpusak eta test-multzoak probatzea. Era honetara, corpusaren eta hizkuntzaren eragina bakoitza bere aldetik aztertu ahalko litzateke.
- **Sistema osoaren eskuzko ebaluazio bat egitea.** Izan ere, sistema osoaren ebaluazioa BLEU metrika automatikoan oinarritu da, aldagai esperimentalen kopurua hain handia izanda konfigurazio guztiekin eskuzko ebaluazio bat egitea ez baitzen bideragarria. Behin aldagai hauen balio optimoak finkatuta, ordea, interesgarria litzateke eskuzko ebaluazio bat burutzea, esaldi bakoitzeko *baseline*aren eta proposatutako sistemaren itzulpenak eman eta zenbatetan hobetu eta zenbatetan okertzen den ikusiz.
- **Web bidezko itzulpen-sistemen domeinura egokitzea.** Erabiltzaile arrunt gehienek web bidez erabiltzen dute itzulpen automatikoa, eta interesgarria litzateke ga-

raturiko sistema ingurune horretara egokitzeko aukera aztertzea. Izan ere, domeinu bat propioki ez izanagatik hainbat eskaera askotan errepikatuko direla pentsa daiteke, sistema probatzeko asmoz egiten direnak adibidez. Era honetara, *"Esto es una prueba."* edo *"Yo me llamo Mikel."* bezalakoak behin eta berriz agertuko dira ziurrenik, bai eta *"Autobuses mañana y tarde."* edo *"Estoy en la segunda planta."* bezala harrapatzera doazen esaldi anbiguoak ere. Era berean, corpus gehienetan sekula agertuko ez liratekeen madarikazio, iseka edota esamolde herrikoiak ere oso ohikoak izatea espero daiteke. Horiek guztiak gaizki itzultzeak oso inpresio txarra sortzen du erabiltzailearengan, baina itzultzaile automatikoa horiek ondo egin ditzan moldatzea oso zaila izan ohi da. Gauzak honela, interesgarria litzateke web bidezko itzultzaile batek jasotzen dituen eskaeren azterketa bat egitea, gehien errepikatzen direnak eskuz itzuli eta proiektu honetan garatutako sistema haiekin entrenatuz. Honekin hibridatu ostean, bada, web bidezko itzulpen-sistemaren erabilera-kasu tipikoenetan hobekuntza handi bat lortzea espero daiteke.

- **Apertium eta Matxinekiko integrazioa hobetzea.** Izan ere, HTML dokumentuak itzultzeko eskaintzen duten aukeraren bidez egin da hau, bistakoa denez ez zena horretarako diseinatua izan. Esperimentuan ikusi denez, emaitzetan eragin negatiboa izan du horrek, batez ere Matxinen kasuan. Etorkizunera begira, bada, integrazio-mekanismo egokiagoak aztertzea interesgarria izango litzateke.

- **Entitateen tratamendua hobetzea.** Esperimentuak erakutsi bezala, entitateen tratamenduan akats sistematikoak egiten dira, bai mugen identifikazioan, bai itzulpenean, bai eta sorkuntzan ere. Etorkizunera begira alderdi hau hobetzen saiatu beharko litzateke beraz, alde batetik akatsak ahal diren neurrian konponduz eta, bestetik, arazoak hautemateko mekanismo bat inplementatu (adibidez, izen arruntak dituen eta, honenbestez, itzulpen bat behar duen entitateren bat hiztegian ez agertzea) eta halakoetan itzulpen-partzialak baztertuz.

- **Kontrakzioen tratamendua hobetzea.** Esperimentuan ikusi denez, kontrakzioen tratamendu desegoki baten ondorioz zuhaitz sintaktiko okerrak eraikitzen dira batzuetan, emaitzetan eragin negatiboa duena. Etorkizunean, bada, arazo hau konpontzen saiatu beharko litzateke, kontrakzioen birtokenizazioa onartu eta postprozesu baten bidez haien jatorrizko forma berreskuratuz.

- **Lerrokatze-akatsak hautemateko heuristikoak erabiltzea.** Esperimentuan behatutako beste arazo bat lerrokatze-akatsei dagokie. Bistakoa denez, lerrokatze-teknika zehatzagoen garapena proiektu honen esparrutik erabat kanpo dago, baina

posible litzateke, gutxienez, heuristikoak erabiliz lerrokatze-akats nabarmenenak hautematea. Hurbilpenik errazena bat-etortze bakoitzeko lerrokatutako segmentuen luzerari erreparatzea izango litzateke, bien arteko ezberdintasuna handia izanez gero bat-etortzea baztertuz.

- **Maiuskula/minuskulen tratamendu bat egitea.** Esaldi hasierako maiuskulei lotutako arazoak saihesteko, EBMT aurreprozesatzaileak hizki xehez ematen ditu bere itzulpen partzialak, eta itzultzaile nagusiaren esku geratzen da behar diren maiuskulak jartzea. Komenigarria litzateke, baina, EBMT aurreprozesatzaileak berak maiuskula eta minuskulen tratamendu egoki bat egitea.

- **Garatutako sistemarentzako aplikazio berriak aztertzea.** Proiektu hau itzulpen automatikoan soilik zentratu bada ere, egindako lanak beste aplikazio batzuk ere izan ditzakeela uste da. Bereziki interesgarria litzateke itzulpen-memorien kudeatzaile batekin txertatzea, haien inguruan ematen ari diren azken berrikuntzekin eta, bereziki, bigarren eta hirugarren belaunaldiko itzulpen-memoria bezala ezagutzen direnekin bat egiten baitu zentzu askotan.

Eranskinak

A. ERANSKINA

Proiektuko plangintza

Eranskin honetan proiektuko plangintza azaltzen da, azpiatal banatan bertan egin beharreko lanaren antolaketa eta komunikazio-, kalitate- eta arrisku-planak zehaztuz.

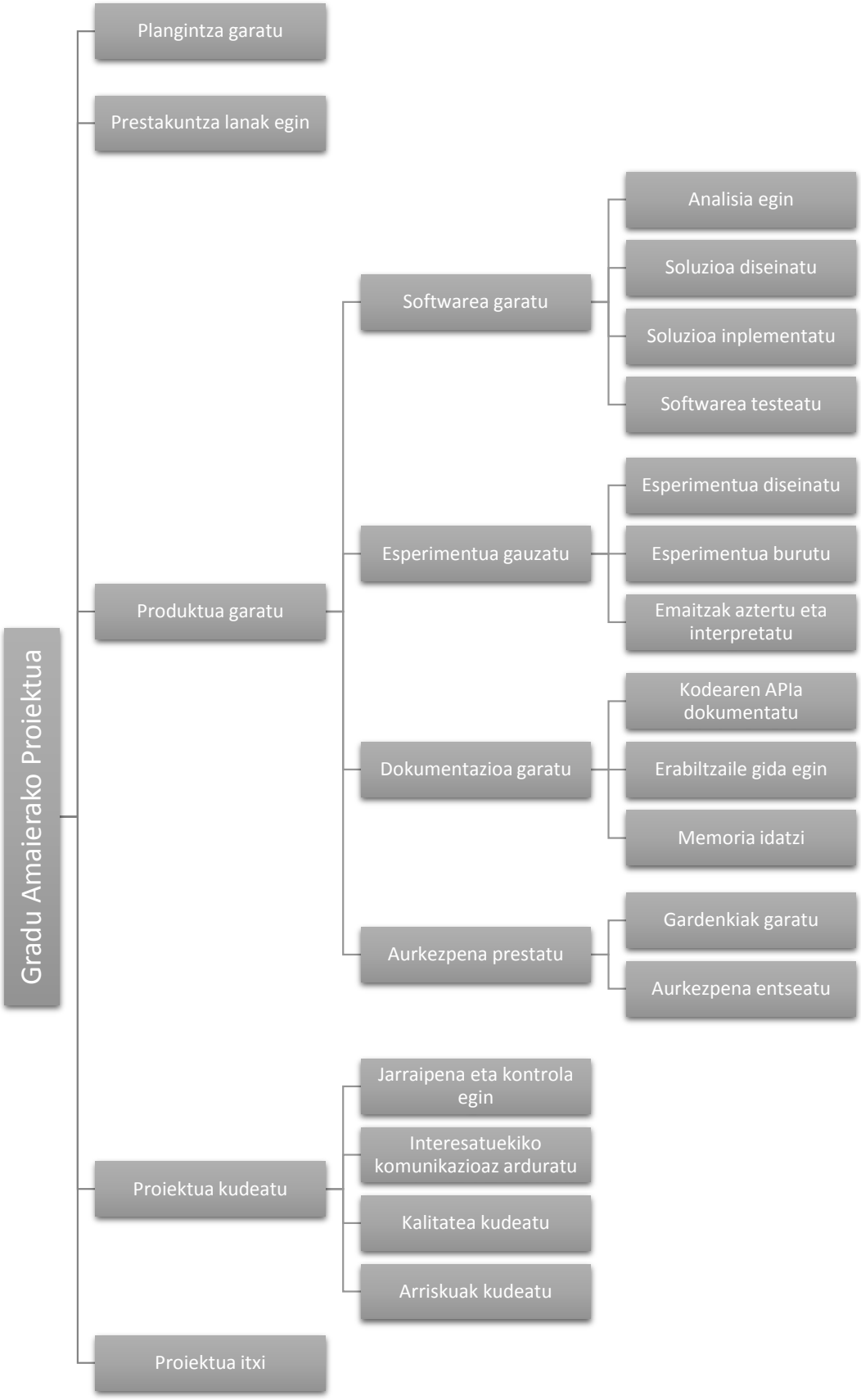
A.1 Lanaren antolaketa

Atal honetan proiektua aurrera ateratzeko burutu beharreko atazak eta denboraren araberako haien antolaketa azaltzen dira. Horretarako, lehen azpiatal batean lanaren deskonposaketa egituraren baitan identifikatutako atazen zehaztapena ematen da, eta ondoren haiek biltzen dituen kronograma aurkezten da.

A.1.1 Lanaren deskonposaketa egitura (LDE)

Proiektuan egin beharreko lana ondorengo ataza eta azpiatazetan deskonposatu da modu hierarkikoan A.1 irudiko LDE diagramak erakusten duen bezala:

- **Plangintza garatu.** Ataza honen bidez proiektua aurrera eramateko erreferentziatzat erabiliko den plangintza bat garatuko da, bere helburu eta betekizunak zehaztu, lana atazetan banatu eta denboran antolatu, eta komunikazio-, kalitate- eta arrisku-planak osatuz.
- **Prestakuntza lanak egin.** Ataza hau proiektua aurrera eramateko beharrezkoa den ezagutza espezifikoa eskuratzera bideraturik dago. Zehatzagoak izanez, jorratu



A.1 Irudia: LDE diagrama

beharreko alderdiak bitan banatzen dira: inplementazioan erabiliko diren teknologien inguruko prestaketa batetik, eta hizkuntzaren prozesamenduaren eta itzulpen automatikoaren inguruko oinarrizko trebakuntza bestetik, proiektuaren garapenerako beharrezkoak diren arlo jakinetan bereziki sakonduz.

- **Produktua garatu.** Ataza hau proiektuaren azken helburua den produktua garatzeaz arduratzen da, eta ondorengo azpiatazek osatzen dute:
  - **Softwarea garatu.** Azpiataza honen eginkizuna proiektuaren ardatza den itzulpen-sistema bera garatzea da eta, halakoetan ohikoa denez, ondorengo urratsek osatzen dute:
    * ***Analisia egin***
    * ***Soluzioa diseinatu***
    * ***Soluzioa inplementatu***
    * ***Softwarea testeatu***
  - **Esperimentua gauzatu.** Azpiataza honen bidez garaturiko itzulpen-sistemaren portaera esperimentalki probatuko da, emaitzen kalitatea ebaluatu eta aldagai ezberdinen eragina aztertuz. Horretarako, ondorengo urratsak jarraituko dira:
    * ***Esperimentua diseinatu***
    * ***Esperimentua burutu***
    * ***Emaitzak aztertu eta interpretatu***
  - **Dokumentazioa garatu.** Azpiataza hau proiektuan zehar eginiko lana dokumentatzean datza, bai garaturiko softwarea erabil dezaketen programatzaileei, bai azken erabiltzaileei, bai eta gradu amaierako proiektuaren eskakizunei ere zuzendurik. Horietako bakoitzari erantzuteko, hurrenez hurren ondorengo eginkizunak aurreikusten dira:
    * ***Kodearen APIa dokumentatu***
    * ***Erabiltzaile gida egin***
    * ***Memoria idatzi***
  - **Aurkezpena prestatu.** Azpiataza honen helburua proiektuaren defentsarako aurkezpena prestatzea da, eta honako bi urratsek osatzen dute:
    * ***Gardenkiak garatu***
    * ***Aurkezpena entseatu***

- **Proiektua kudeatu.** Ataza honen helburua proiektua bide onetik doala bermatzea izango da, plangintzan zehazturikoa jarraitzen dela ziurtatuz eta, beharrezkoa izanez gero, haren egokitzapenak planteatu eta onartuz. Bide horretan, proiektuko plangintzak markaturiko prozedurak jarraituko dira, komunikazio-, kalitate- eta arrisku-planetan zehazturikoei izaera propio bat emanez. Hori dela eta, ataza hau ondorengo azpiatazetan banatu da:
  - **Jarraipena eta kontrola egin.**
  - **Interesatuekiko komunikazioaz arduratu.**
  - **Kalitatea kudeatu.**
  - **Arriskuak kudeatu.**
- **Proiektua itxi.** Ataza honen egiteko bakarra emangarri guztiak entregatu eta egindako lana etorkizunean erabilgarri izan dadin gordetzea da. Era berean, proiektuko zuzendariarekin azken bilera bat egingo da egindako lanaren inguruan hausnartu eta, komenigarri ikusiz gero, haren harira plantea litezken proiektu edota lan-lerro berriak eztabaidatzeko.

A.1.2 Kronograma

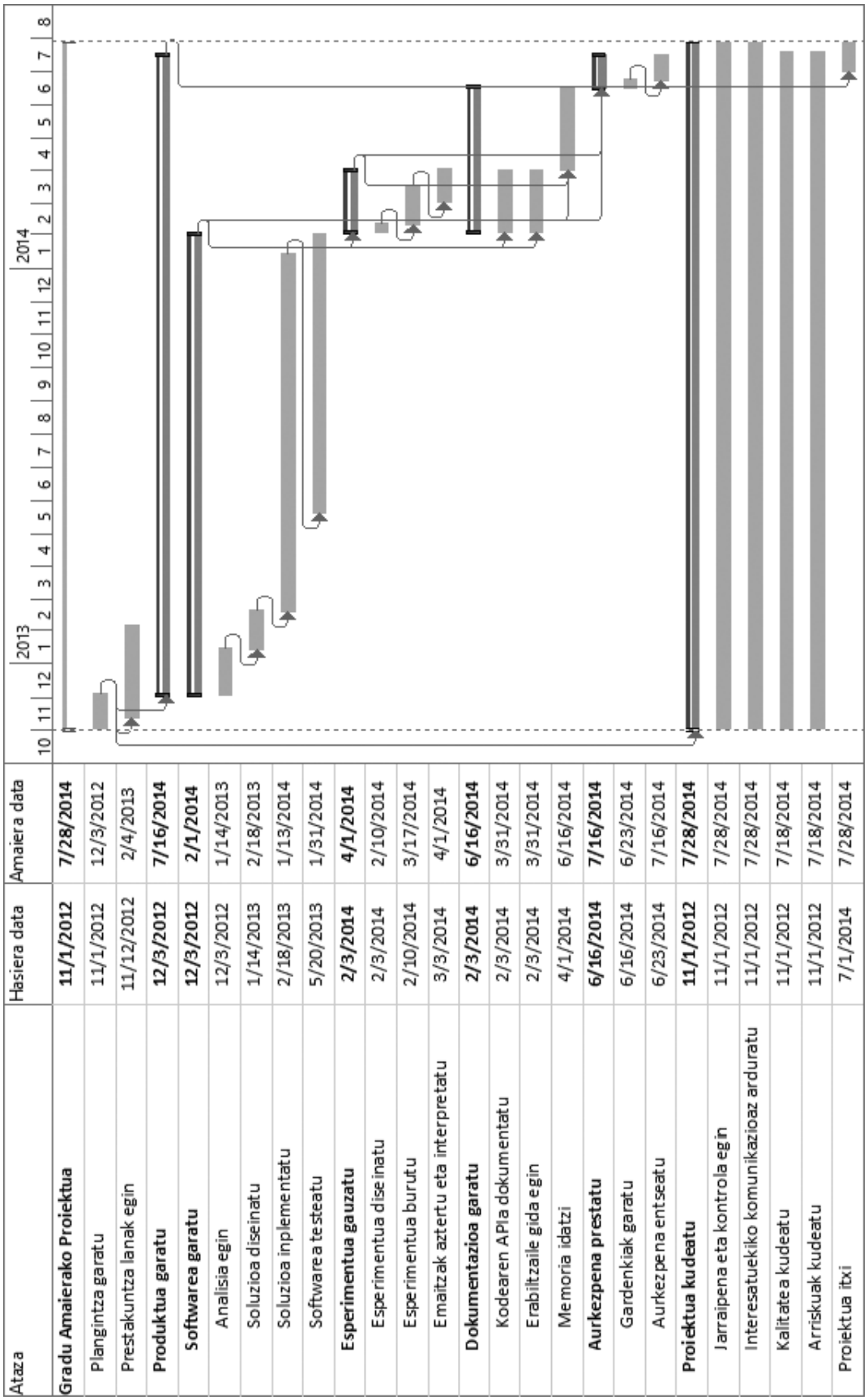
Aurreko azpiatalean azalduriko atazak denboran zehar antolatu eta haien arteko menpekotasunak jasoz A.2 irudiko kronograma osatu da.

A.2 Komunikazio-plana

Atal honetan proiektuko komunikazio-plana aurkezten da, azpiatalez azpiatal bertako interesatuak, haiekiko aurreikusiriko komunikazio-kanalak eta lan-metodologia, eta proiektuko informazio-sistema zehaztuz.

A.2.1 Interesatuen identifikazioa

Proiektuaren zehaztapena eta izaera orokorra aintzakotzat hartuz, ondoko interesatuak identifikatu dira:



A.2 Irudia: Kronograma

- **Mikel Artetxe:** Lehen interesatua proiektua aurrera eraman behar duen Mikel Artetxe bera da, Informatika Ingeniaritzako Graduko ikaslea bera. Abiapuntu bezala, kontuan hartu behar da ikasketa-planaren arabera gradu amaierako proiektua derrigorrezko 12 kredituri dagozkiola, graduan zehar ikasitako edukiak eta lortutako ahalmen eta trebetasunak praktikan jartzeko helburua dutenak. Horrekin batera, ikasleari interes handikoak zaizkion hizkuntzalaritza konputazionalari eta itzulpengintza automatikoari buruz gehiago ikasteko parada ezin hobetzat ere ikusten da.

- **Kepa Sarasola:** Kepa Sarasola izango da proiektuko zuzendaria, Informatikan Doktorea, EHUko Informatika Fakultateko irakaslea eta IXA taldeko kidea bera, itzulpen automatikoaren arloan aritzen dena. Proiektuaren ikuskapen orokorraz arduratuko da, goi-mailako jarraipena egin eta erabaki estrategikoetan parte hartuz.

- **Gorka Labaka:** Gorka Labaka Informatikan doktorea eta IXA taldeko ikerlaria da, itzulpen automatikoaren arloan diharduena. Proiektuaren alde teknikoaren gidaritzaz arduratuko da bera, proiektuaren garapena gainbegiratu, egunerokotasunean sortzen diren zalantzak ebatzi eta hartu beharreko erabakietan lagunduz.

- **IXA taldea:** Proiektua IXA taldearen barnean garatuko da, modu honetara ikerketalerro berri bat zabaldu eta eman litezkeen aurrerapenak bere itzulpen-sistemetan txertatzeko interesa duena. Gauzak honela, proiektuan zehar bertako taldekide gehiagorekin ere elkarlanean arituko da, eginiko lanaren berri emateko batetik eta jorratu beharreko bideen inguruan eztabaidatzeko bestetik. Era berean, proiektua aurrera eramateko ikerketa talde honen hainbat bitarteko eta baliabide ere erabiliko dira, zerbitzariak eta softwarea adibidez, eta haien inguruan sor litezkeen zalantza eta arazoak ebazteko ere bertako taldekideen laguntza izango da.

A.2.2 Lan-metodologia eta komunikazio-kanalak

Lan-poltsa eta lankidetza-bekaren harira Joxe Mari Korta eraikinean IXA taldeak duen bulegoan mahai bat esleitu zaio Mikel Artetxeri IXAren zerbitzarietarako atzipena duen mahaigaineko ordenagailu batekin batera. Gorka Labakak bulego berean egiten du lan eta, hori dela eta, proiektuaren oinarrizko jarraipena egunerokotasunaren baitan egin ahalko dela aurreikusten da aurrez aurreko harremanaren bidez. Nolanahi ere, lanaren zati bat etxetik ere egitea da asmoa, batez ere autonomia handiago batez egiteko modukoa bada eta IXA taldearen zerbitzariak erabiltzea eskatzen ez badu (adibidez, dokumentazio lana).

Proiektuaren goi-mailako ikuskapena egin eta erabaki garrantzitsuenak hartzeko, berriz, proiektuko zuzendariarekin bilduko da aldiro-aldiro.

Gauzak honela, honakoak dira proiektuko interesatuekin komunikatzeko erabiliko diren kanalak:

- **Bulegoko aurrez aurreko harremana.** Esan bezala, Mikel Artetxe eta Gorka Labaka bulego berean arituko dira lanean. Horri esker, proiektuaren egunerokotasunean sortzen diren zalantza, erabaki eta aurrerapausoak bien arteko aurrez aurreko harremanaren bidez eztabaidatuko dira.
- **Posta elektronikoa.** Posta elektronikoa izango da proiektuko zuzendariarekin komunikatzeko bitarteko nagusia, zer edo zer sakonago landu behar denetan bileraren bat hitzartuz. Era berean, Gorka Labakarekin zerbait komentatu behar eta bulegoan elkar topatu ezean (bidaiak, bilerak, bestelako konpromisoak...), posta elektronikoa erabiliko da horretarako. Amaitzeko, IXA taldeko gainerako kideekin komunikatzeko ere bitarteko hau baliatuko da, taldekide guzti-guztiei zuzendu nahi izanez gero IXAren posta-zerrenda erabiliz.
- **Bilerak.** Erabaki estrategikoak hartu nahiz proiektuaren goi-mailako jarraipena egiteko bilerak egingo dira proiektuko zuzendari den Kepa Sarasolarekin. Alderdi teknikoak eztabaidatzeko asmoa denean, Gorka Labaka ere bilera horietan izango da. Bilerak proiektuaren garapenaren baitan ematen diren gertaeren arabera deitu ahalko ditu edozein aldek eta, honenbestez, ez dira aldizkakotasun jakin batekin burutuko printzipioz. Bilera hauetaz gain, itzulpen automatikoaren inguruan aritzen diren IXA taldeko gainerako kideekin ere noiz edo noiz biltzeko asmoa da eginiko lanaren berri eman eta haien ekarpenak jasotzeko.

A.2.3 Proiektuko informazio-sistemaren deskribapena

Proiektuko informazio-sistema erabat digitala izango da. Aurreko atalean aipatu bezala, bulegoan ez ezik etxean ere egingo da lan eta, horrekin batera, informazio-galera ekiditeko segurtasun-politika egoki bat jarraitu nahi da. Hori dela eta, informazio-sistema banatu bat diseinatu da, datu berberak hainbat lekutan mantentzen dituena eta ondorengo elementuek osatzen dutena:

- **SVN errepositorioa.** Iturburu-kodea mantentzeko IXA taldearen SVN errepositorioa erabiliko da. Bertsio-kontrolerako, gainerako interesatuek egindako lana uneo-

ro eskuragarri izateko eta leku ezberdinetatik modu errazagoan lan egiteko aukera ematen du honek.

- **IXA taldearen zerbitzariak.** Kostu konputazional handiko lanak egiteko eta, bereziki, esperimentaziorako, IXA taldearen zerbitzariak erabiliko dira. Zerbitzari hauek guztiek disko partekatuak erabiltzen dituzte, eta bertan gordeko dira, honenbestez, exekutagarriak, corpusen moduko baliabideak, nahiz garaturiko softwareak sorturiko fitxategi prozesatuak.

- **Bulegoko ordenagailua.** Bulegoko ordenagailuan uneoro bertan lan egiteko beharrezkoa dena baino ez da gordeko: iturburu-kodearen kopia lokal bat, dokumentazio-fitxategiak, nahiz garaturiko softwarea testeatzeko beharrezkoak diren proba-fitxategiak.

- **Etxeko ordenagailua.** Etxeko ordenagailuak abiadura oso handiko baina edukiera txikiko SSD unitate bat du. Hori dela eta, bertatik lan egiteko aurrerago azaltzen den NAS unitatea erabiliko da nagusiki, era berean babes-kopiak modu errazean sortu eta kudeatzeko aukera ematen duena. SSD unitatean, berriz, atzipen abiadura azkar bat eskatzen duten aldi baterako proba-fitxategiak baino ez dira gordeko.

- **USB memoria.** Elkarri zuzenean konektaturik ez dauden gailuen artean datuak transmititzeko 32GBko edukiera duen USB memoria bat erabiliko da. Era berean, hilabetean behin egindako lanaren (iturburu-kodea eta dokumentazio-fitxategien) babes-kopia bat gordeko da bertan fitxategi trinkotu bezala.

- **Etxeko NAS unitatea.** Etxeko NAS unitateak bi disko ditu, bata datuen euskarri nagusi modura erabiltzeko eta bestea egunero-egunero automatikoki egiten diren babes-kopia inkrementalak gordetzeko. Etxeko ordenagailutik lan egiten denean bera erabiliko da informazio-euskarri nagusi modura, iturburu-kodearen kopia lokal bat eta dokumentazio-fitxategiak jasoz behinik behin, eta horri esker babes-kopiak automatikoki sortu eta kudeatuko dira haientzat. Era berean, etxetik lan egin ez den asteetan eskuz kopiatuko da astean zehar egindako lana haren babes-kopiak ere izan daitezen. Amaitzeko, IXA taldearen zerbitzarietan datu berriak prozesatzen diren aldiro, haien babes-kopiak gordeko dira NAS unitatean.

A.3 Kalitate-plana

Atal honetan proiektuko kalitate-plana aurkezten da, oinarrizko helburu eta betekizunetatik haratago joan eta garatu beharreko produktuaren bikaintasuna bilatzen duena. Honela, lehen azpiatal batean produktuaren kalitate-dimentsioak finkatzen dira, proiektuaren arrakastarako derrigorrezkoak izan ez arren maila altuko emaitza baten bila betetzen saiatuko direnak eskuragarri diren baliabideen arabera. Bigarren azpiatal batean, berriz, hori betetzeko asmoz garatu den kalitatearen ziurtapen eta kontrolerako prozedura azaltzen da.

A.3.1 Produktuaren kalitate-dimentsioak

Garatu beharreko produktuari loturik ondorengo kalitate-dimentsioak finkatu dira, posible den neurrian betetzen saiatu beharrekoak:

- **Hizkuntzaren analisirako tresnekiko modulartasuna.** Garaturiko itzulpen-sistemak erabiltzen dituen hizkuntzaren analisirako tresnekiko modularra izan beharko luke, hizkuntza edota tresna berriak modu errazean integratzeko aukera eskainiz.
- **Itzultzaile nagusiarekiko independentzia.** Itzulpen partzialak sortuko dituen EBMT aurreprozesatzailea itzultzaile nagusiarekiko erabat independentea izan beharko litzateke, gutxieneko lotura lanak eginda edozein itzulpen-sistemarekin erabiltzeko aukera emanez.
- **Iturburu-kodearen ulergarritasun eta hedagarritasuna.** Iturburu-kodea edozein programatzailek erraztasunez uler eta alda dezan garatuta egon beharko litzateke, ondorengo irizpideak jarraituz:
  - **Erabilitako programazio lengoaiaren konbentzio eta praktika onak errespetatzea.** Programazio lengoaia bakoitzaren inguruko komunitatean berezko sintaxi arauez haratago doazen konbentzio eta praktika onak sortu ohi dira (estiloari, aldagaien izendapenari edota kodearen egituraketari dagozkionak, adibidez), eta iturburu-kodea haiek errespetatuz idatzi beharko litzateke.
  - **Objektuei Orientatutako Programazioaren printzipioak jarraitzea.** Objektuei Orientatutako Programazioa software modular, hedagarri eta berrerabilgarria eraikitzeko programazio-paradigma erabiliena da gaur egun, eta proiektuaren ardatza den itzulpen-sistema ere haren printzipioak jarraituz garatu beharko litzateke.

  - **Kodea autoesplikatiboa (*self-explanatory*) izatea.** Irizpide honen arabera kodeak ahalik eta ulergarriena izan beharko luke bere baitan, ondo egituratuta egonez, identifikatzaile egokiak erabiliz eta abar.
  - **Kodearen ulermenerako beharrezko iruzkinak egotea.** Aurreko irizpidea jarraituz kodea ahalik eta autoesplikatiboena izan beharko balitzateke ere, batzuetan azalpen gehigarriak ematea komenigarria izan liteke, eta horretarako egoki diren iruzkinak jarri beharko lirateke.
  - **Kodearen APIa dokumentatzea.** Kodearen APIa behar bezala dokumentatuta egon beharko litzateke, atributu eta metodo publikoen funtzio eta hitzarmenak deskribatuz.

- **Sistemaren abiadura eta arintasuna.** Garaturiko sistema ingurune erreal batean aritzeko moduko eraginkorra izan beharko litzateke, bai memoria-erabilerari dagokionez, bai eta prozesamendu-denborari dagokionez ere. Zentzu honetan, garrantzi berezia izango du sistemaren eskalagarritasunak, baldintzak gogortu eta, bereziki, itzulpen-memoria luzeagoak erabili ahala, kostua arrazoizko modu batean handitu beharko litzatekeelarik.

- **Esperimentuaren sakontasuna.** Esperimentuak garatutako sistemaren portaera ebaluatzeaz gain aldagai ezberdinek bertan duten eragina neurtu eta bere sendotasun eta ahuleziak ezagutzeko balio beharko luke, modu honetara egindako lanaren benetako ekarpena zein den argitasunez ikusi eta etorkizuneko lan-lerroak markatu ahal izateko.

- **Dokumentazioaren argitasuna eta zehaztasuna.** Garaturiko dokumentazioak idazkera formal, zuzen eta zehatza jarraitu beharko luke, eta egindako lanaren berri zuzenik ez duen irakurleari zuzendurik egon beharko litzateke, gauzak oinarri-oinarritik hasita eta argitasunez azalduz.

- **Aurkezpenaren atsegintasuna eta sintesi-gaitasuna.** Proiektuaren defentsarako aurkezpena publikoarentzat atsegina eta jarraitzeko erraza izan beharko litzateke, eta proiektuan zehar egindako lan guztia argitasunez sintetizatu beharko luke.

A.3.2 Kalitatearen ziurtapen eta kontrolerako prozedura

Aurreko azpiatalean jasoriko kalitate-dimentsioak bete daitezen funtsezkotzat jo da garapenean zehar kalitatearen jarraipen bat egitea, zehazturiko puntuak bete direla edo bete

daitezen neurri egokiak hartzen ari direla egiaztatuz. Asmo horrekin, produktuaren kalitatea ebaluatzeko bi mekanismo nagusi finkatu dira:

- **Barne-ebaluazioa.** Garapenean zehar kalitate-dimentsioak betetzen direla edo betetzeko bidean direla bermatzeko ahalegin berezi eta jarraitu bat egingo da. Horretarako, eguneroko jardunean gogoan izateaz gain kalitate-dimentsioen banakako berrikuspen periodikoak ere egingo dira, haien betetze-maila neurtu eta beharrezko erabaki zuzentzaileak hartuz.

- **Kanpo-ebaluazioa.** Garatu beharreko softwarearen prototipo esanguratsuak edota dokumentazioaren zirriborroak eskuratu ahala proiektuko gainerako interesatuei haien berrikuspen bat egiteko eskatuko zaie. Zehatzagoak izanez, Gorka Labaka izango da alde teknikoari loturiko alderdien kanpo-berrikuspenaren arduradun nagusia, eta Kepa Sarasola proiektuko zuzendaria gainerakoena. Horietaz gain, baina, egindako lanaren inguruan interesa duten IXA taldeko beste kide batzuen parte hartzea ere aurreikusten da.

Bi mekanismoetakoren baten baitan aurreikusirikoarekiko inolako desbiderapenik hautemango balitz, albait azkarren informazioa kontrastatu eta desbiderapena baieztatu ala ezeztatzeko saiakera egingo litzateke. Baieztatuz gero, arazoaren jatorria identifikatzeko ezohiko bilera bat deituko litzateke proiektuko zuzendariarekin, bai eta, egoki iritziz gero, zeresana izan lezaketen gainerako interesatuekin ere. Era berean, bilera horretan neurri zuzentzaile egokiak eztabaidatu eta adostuko lirateke. Behin hori eginda, aurrerantzeko ebaluazio-saiakeretan lehenago huts egin duen atalari arreta berezia eskainiko litzaioke, bide onetik jarraitzen dela bermatuz.

A.4 Arrisku-plana

Atal honetan proiektuko arrisku-plana azaltzen da. Horretarako, lehen azpiatal batean hasiera batean hautemandako hiru arrisku nagusiak zehazten dira, euren deskribapena, probabilitatea, inpaktua, mitigazio-neurriak eta kontingentzia-plana emanez. Horren ostean, arrisku horien nahiz proiektuaren garapenean zehar sor litezkeen berrien jarraipen eta kontrolerako prozedura azaltzen da.

A.4.1 Arrisku nagusien zehaztapena

Hiru dira hasiera batean identifikaturiko arrisku nagusiak: antolakuntza-arazoak izatea, informazio-galera, eta ekipamendu teknikoa matxuratzea. Jarraian, banan-banan euren xehetasunak zehazten dira.

Antolakuntza-arazoak

- **Deskribapena**: Baliteke antolakuntza-arazoak medio plangintzan egindako aurreikuspenak ezin betetzea. Proiektuaren ezohiko izaerak eta horri aurre egiteko planteaturiko malgutasunak, gainera, arrisku hau areagotzen du.

- **Probabilitatea**: Ertaina (~%20).

- **Inpaktua**: Handia (plangintzarekiko izandako desbideraketen arabera). Desbideraketak handiak badira, atzerapenak edota proiektuaren betekizunak nahiz kalitate-helburuak ezin betetzea eragin lezakete. Hala eta guztiz ere, arrisku hau areagotzen duen malgutasunari berari esker arazoei plangintza zurrun batekin baino errazago egin daiteke aurre, bere balizko ondorioak neurri handi batean arinduz.

- **Mitigazio-neurriak**: Arrisku hau minimizatzeko proiektuaren jarraipen zorrotz bat egingo da, Gorka Labakaren ikuskapenarekin egunerokotasunean eta alderdi teknikoenetan eta Kepa Sarasola proiektuko zuzendariarenarekin ikuspegi orokorrago batetik. Horretarako, A.2 ataleko komunikazio-planean zehazturiko bitarteko eta prozedurak jarraituko dira.

- **Kontingentzia-plana**: Proiektuaren garapenean zehar interesatuetako edonork antolakuntza-arazoak izaten ari direla edota plangintzarekiko desbideraketak eman daitezkeela sumatzen badu ezohiko bilera bat deituko du lehenbailehen. Bilera horretan Mikel Artetxe, Kepa Sarasola eta Gorka Labakaren artean egoeraren diagnostiko bat egin eta hartu beharreko neurriak adostuko dira, izan proiektuaren antolakuntzan zuzenketak egitea edota plangintza bera birmoldatzea. Hartutako erabakiek proiektuko zuzendari Kepa Sarasolaren eta Mikel Artetxeren onespena beharko dute gutxien-gutxienez, baina interesatu guzti-guztien oniritzia lortzen ahaleginduko da.

Informazio-galera

- **Deskribapena**: Matxura, eraso informatiko edo giza akats baten ondorioz posible da egindako lana edo haren parte bat galtzea.
- **Probabilitatea**: Oso txikia (~%5).
- **Inpaktua**: Oso handia. Galduriko informazio kantitatearen arabera, hainbat egun, aste edo hilabetetako lana berregitea beharrezkoa izan daiteke, epe zehatzen arabera proiektuan atzerapenak eragin ditzakeena.
- **Mitigazio-neurriak**: Mitigazio-neurri bezala A.2.3 atalean azalduriko gidalerroak jarraituko dira proiektuko informazio-sistema kudeatzeko. Bere izaera banatuari esker, informazio-galera bat jasateko probabilitatea oso txikia da honela, eta babeskopien politikari esker izatekotan ere bere inpaktua minimoa izango litzateke kasurik gehienetan. Horretaz gain, hilabetean behin informazio-sistema osoaren berrikuspen periodikoak egingo dira guztia behar bezala dagoela bermatzeko.
- **Kontingentzia-plana**: Informazio-galeraren bat pairatu dela sumatzen bada, lehen urratsa informazio-sistemaren erabateko berrikuspen bat egitea izango da, galerarik izan denetz egiaztatu eta, hala balitz, jasaniko kalteak kuantifikatuz. Honela, izandako arazoek hainbat egunetako lanari eragiten diotela ikusten bada proiektuko zuzendariarekin ezohiko bilera bat deituko da, eta plangintzan beharrezko egokitzapenak adostu.

Ekipamendu teknikoa matxuratzea

- **Deskribapena**: Litekeena da proiektuaren garapenean zehar bertan erabiltzen den gailuren bat matxuratzea: bulegoko nahiz etxeko ordenagailuak, IXA taldearen zerbitzariak, USB memoria...
- **Probabilitatea**: Altua (~%50).
- **Inpaktua**: Baxua/Ertaina (matxuratutako gailuaren eta konpontzeko epeen arabera). Izandako matxuraren arabera, proiektua erabat blokeatuta gera daiteke, baina kasurik gehienetan epe motzean arazoak konpondu edo ekipamendu berria lortu ahalko litzatekeela aurreikusten da. Bereziki kezkagarritzat jotzen dira IXA taldearen zerbitzariekin izan litezken arazoak, kostu konputazional handiko atazak burutu

beharko direla aurreikusten baita, halakoek esperimentuaren garapena nabarmenki motel lezaketelarik.

- **Mitigazio-neurriak**: Aurreikusten den mitigazio-neurri posible bakarra ekipamendu teknikoaren erabilera egoki bat egitea da, ohiko mantentze-lanak burutuz eta zentzuzko segurtasun-politika bat jarraituz.

- **Kontingentzia-plana**: Matxuratutako gailua IXA taldearena bada bertako teknikariekin jarriko da harremanetan lehenbailehen. Honen ostean arazoa larria dela eta proiektuaren garapena serioski oztopa dezakeela ikusten bada, ezohiko bilera bat deituko da proiektuko zuzendariarekin hartu beharreko neurriak eztabaidatzeko, beste irtenbiderik ezean bestelako ekipamenduren bat eskuratzeko aukera aztertuz. Matxuraturiko gailua propioa bada, berriz, arazo zehatza zein den identifikatzen saiatuko da, teknikari baten laguntzaz behar izatekotan, eta epe motzean konpon ezin daitekeela ikusten bada, gailu berri bat erosi.

A.4.2 Arriskuen kudeaketa eta jarraipen eta kontrolerako prozedura

Aurreko azpiatalean proiektuko arrisku nagusiak identifikatu, zehaztu eta eurentzako mitigazio- eta kontingentzia-planak garatu dira. Horretaz gain, baina, beharrezkotzat jo da arriskuen kudeaketa eta jarraipen eta kontrolerako prozedura bat finkatzea, proiektuaren garapenean zehar euren egoera eguneratu eta, beharrezkoa balitz, egoki diren neurriak har daitezen, bai eta arrisku berriak identifikatu eta tratatzeko mekanismo bezala balio dezan ere.

Horri guztiari aurre egiteko, arrisku-erregistro bat erabiltzea erabaki da. Arrisku-erregistroa proiektuko informazio-sisteman dokumentazioarekin batera gordeko den taula bat izango da, une bakoitzean indarrean dauden arriskuak bilduko dituena. Deskribapen hutsaz gain, aurreko atalean jasoriko puntuen laburpen bat ere jasoko du arrisku bakoitzeko. Proiektuaren garapenean zehar arrisku-erregistroaren berrikuspen periodikoak egingo dira, behar izanez gero bertako arriskuak eguneratu, ezabatu ala berriak gehituz, eta modu honetara arriskuen jarraipen eta kontrola gauzatuz. Era berean, A.2.2 atalean zehazturikoaren arabera proiektuko zuzendariarekin egiten diren bileretan arriskuen azterketa orokor bat burutuko da hizpide den erregistroan oinarrituz.

Bibliografia

Abouelhoda, M. I., Kurtz, S., and Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2(1):53–86.

Aduriz, I., Agirre, E., Aldezabal, I., Arregi, X., Arriola, J., Artola, X., Gojenola, K., Maritxalar, A., Sarasola, K., and Urkia, M. (1999). MORFEUS: Euskararako analizatzaile morfosintaktikoa.

Aduriz, I., Aranzabe, M. J., Arriola, J. M., de Ilarraza, A. D., Gojenola, K., Oronoz, M., and Uria, L. (2004). A cascaded syntactic analyser for Basque. In *Computational Linguistics and Intelligent Text Processing*, pages 124–134. Springer.

Aduriz, I. and Díaz de Ilarraza, A. (2004). Morphosyntactic disambiguation and shallow parsing in computational processing of Basque. In *Inquiries into the lexicon-syntax relations in Basque*, pages 1–23.

Aldezabal, I., Arriola, J. M., Diaz de Ilarraza, A., and Sarasola, K. (2005). *Hizkuntzalaritza konputazionala*. Udako Euskal Unibertsitatea.

Alegrıa, I., Arregi, O., Ezeiza, N., and Fernandez, I. (2006). Lessons from the development of a named entity recognizer for Basque. *Procesamiento del lenguaje natural*, (36):25–37.

Bloch, J. (2008). *Effective Java*. Addison-Wesley, Upper Saddle River, NJ.

DeNero, J. and Klein, D. (2007). Tailoring word alignments to syntactic machine translation. In *Annual meeting-association for computational linguistics*, volume 45, page 17. Citeseer.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the*

43rd Annual Meeting on Association for Computational Linguistics, pages 363–370. Association for Computational Linguistics.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Gotti, F., Langlais, P., Macklovitch, E., Bourigault, D., Robichaud, B., and Coulombe, C. (2005). 3GTM: A third-generation translation memory. In *Proceedings of the 3rd Computational Linguistics in the North-East Workshop*, pages 8–15.

Hutchins, J. (2007). Machine translation: A concise history. *Computer aided translation: Theory and practice*.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall.

Kärkkäinen, J. and Sanders, P. (2003). Simple linear work suffix array construction. In *Automata, Languages and Programming*, pages 943–955. Springer.

Kim, D. K., Sim, J. S., Park, H., and Park, K. (2003). Linear-time construction of suffix arrays. In *Combinatorial Pattern Matching*, pages 186–199. Springer.

Knight, K. (1999). A statistical MT tutorial workbook. In *Prepared for the 1999 JHU Summer Workshop*.

Knuth, D. E., Morris, Jr, J. H., and Pratt, V. R. (1977). Fast pattern matching in strings. *SIAM journal on computing*, 6(2):323–350.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Labaka, G. (2010). EUSMT: incorporating linguistic information into SMT for a morphologically rich language. Its use in SMT-RBMT-EBMT hybridation. *Lengoaia eta Sistema Informatikoak Saila (UPV-EHU). Donostia. 2010ko martxoaren 29a*.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Loinaz, I. A., Arantzabal, I., Forcada, M. L., Guinovart, X. G., Padró, L., Campos, J. R. P., and Waliño, J. (2006). OpenTrad: Traducción automática de código abierto para las lenguas del Estado español. *Procesamiento del Lenguaje Natural*, 27:357–360.

Lu, W. and Xue, R. (2010). Comparative study on multi-systems combination in machine translation. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 8, pages 308–312. IEEE.

Manber, U. and Myers, G. (1990). Suffix Arrays: A New Method for On-line String Searches. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, pages 319–327, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Mayor, A., Alegria, I., De Ilarraza, A. D., Labaka, G., Lersundi, M., and Sarasola, K. (2011). Matxin, an open-source rule-based machine translation system for Basque. *Machine translation*, 25(1):53–82.

Melichar, B., Holub, J., and Polcar, J. (2005). Text searching algorithms. *Available on: http://stringology. org/athens*.

Nirenburg, S. (1993). *Progress in machine translation*. IOS Press.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Padró, L. and Stanilovsky, E. (2012). FreeLing 3.0: Towards Wider Multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey. ELRA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013). Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer.

Somers, H. (2003). An overview of EBMT. In *Recent advances in example-based machine translation*, pages 3–57. Springer.

Somers, H. and Fernández Díaz, G. (2004). Translation Memory vs. Example-based MT: What is the difference. *International Journal of Translation*, 16(2):5–33.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.