

Euskararako analizatzaile sintaktiko-estatistikoa hobetzeko teknikak

*Kepa Bengoetxea**, *Koldo Gojenola***

Hizkuntza eta Sistema Informatikoak Saila (UPV/EHU)

* kepa.bengoetxea@ehu.eus

** koldo.gojenola@ehu.es

DOI: 10.1387/ekaia.14548

Jasoa: 2015-05-31

Onartua: 2015-07-09

Laburpena: honetan euskararako analizatzaile sintaktiko-estatistikoaren emaitzak hobetzeko helburuarekin egindako esperimentu-multzoa aurkezten da. Lan honetan teknika ez-berdinak aztertzen dira: i) zuhaitz-transformazioak, ii) analizatzaileen pilaketa, eta iii) analizatzaile-modelo desberdinen irteeren konbinazioa. Emaitza guztiak zuhaitz-bankutik zuzenean hartutako urre-patroiko ezaugarri morfosintaktikoak erabiliz eta analisi morfologiko eta desanbiguatzeko-moduluetatik hartutako ezaugarri morfosintaktiko automatikoak erabiliz egin dira.

Hitz gakoak: Dependentsietan oinarritutako analisia, Analisi morfologikoa eta desanbiguazioa, Analizatzaile sintaktikoaren konbinazioa.

Abstract: This paper presents a set of experiments to improve the results of the statistical syntactic analyzers for Basque. The present work has examined different techniques: i) tree transformations, ii) stacking, and iii) combinations of the output of several parsers. All the results have been obtained using gold morphosyntactic tags coming directly from the treebank and using automatic morphosyntactic tags coming from morphological analysis and disambiguation module.

Keywords: Dependency Parsing, Morphological Analysis and Disambiguation, Parser combination.

1. SARRERA

Egun, informazio gehiena ordenagailuetan gorde eta prozesatzen da. Ordenagailuan dagoen informazioa era egokian prozesatzeko, beharrezkoa da gizakiaren hizkuntza ulertzea. Hizkuntza ulertzearen ataza ondo gauzatzeko, hizkuntzaren maila ezberdinak (morfologia, sintaxia, semantika, eta abar) landu eta hizkuntzaren prozesamenduko (HP) teknikak eta

baliabideak garatu behar dira, adibidez: datu-base lexikala, zuhaitz-bankua (ingelesez, *treebank*), analizatzaile morfologikoak, analizatzaile sintaktikoak, eta abar. Ideia bat ulertzeko eta adierazteko esaldiak erabiltzen dira, eta esaldia ulertzeko lehen baldintza da esaldi horrek hizkuntzaren ordenaren arabera izatea, hau da, sintaxiaren arabera. Imajinatu, adibidez, «urdira eta edo delako» esaten dela. Esaldi horrek lau hitz ezagun ditu, baina horrela lotuta, hitzek ez dute inongo zentzurik. Esaldiak lehen baldintza hori betetzen duen jakiteko, analizatzaile sintaktikoa erabil daiteke. Baina, analizatzaile sintaktikoen erabilera arlo desberdinetara zabaltzen ari da: analisi semantikoa lortzeko, gramatika-zuzentzaile modura lan egiteko edo gaizki erabilitako egiturak detektatzeko, bai eta, besteak beste, hizkuntzaren modelizazioan, informazioaren berreskurapenean, itzulpen automatikoan, galdera-erantzunetan, itzulpen automatikorako baliabideak sortzean edota parafraasiaren eskuratzeko automatikoan ere. Analizatzaile sintaktiko bat garatzeko orduan, azken urteotan asko ugartu dira ikasketa automatikoan oinarritutako analizatzaile sintaktiko-estatistikoak. Dependentsia-sintaxian eta datuetan oinarritutako 2006ko eta 2007ko *CoNLL Shared Task* zereginen ondoren, grafoetan eta trantsizioetan oinarritutako hurbilpenak nagusitu ziren. Lan honetan, 2006ko *CoNLL* zereginen ondoren nagusitutako hurbilpen bietan (hots, trantsizio eta grafoetan) oinarritutako sistema onenen egokitzapena egin da, hau da, MaltParser [1] eta MSTParser [2] izenekoena, hurrenez hurren. Artikulu honetan, datuetan oinarritutako euskararako analizatzaile sintaktikoaren emaitzak hobetzeko helburuarekin egindako esperimentu-multzoa aurkezten da. Berton, teknika ezberdinak aztertzen dira: zuhaitz-transformazioak, analizatzaileen pilaketa eta analizatzaile-modelo desberdinen irteeren konbinazioa. Teknika horiek izan dezaketen eragina probatzeko, honako atal hauetan egituratu da lana. Baliabideen atalean, sintaxi konputazionalan erabili diren baliabideak aipatzen dira. Baliabideen artean, euskarako zuhaitz-bankuak, analizatzaile sortzaile nagusiak (MSTParser eta MaltParser), informazio morfosintaktikoa lortzeko, erabili diren baliabideak edo moduluak aurkezten dira. Tekniken atalean, erabilitako zuhaitz-transformazio, pilaketa eta bozketa bidezko konbinazio-tekniken deskribapena egiten da. Ondoren, emaitzen atalean, teknika ezberdinekin analisi sintaktikoan lortutako hobekuntzak aztertzen dira. Azkenik, ondorioen atalean, emaitze tatik at eratako zenbait ondorioen berri emango da.

2. BALIABIDEAK

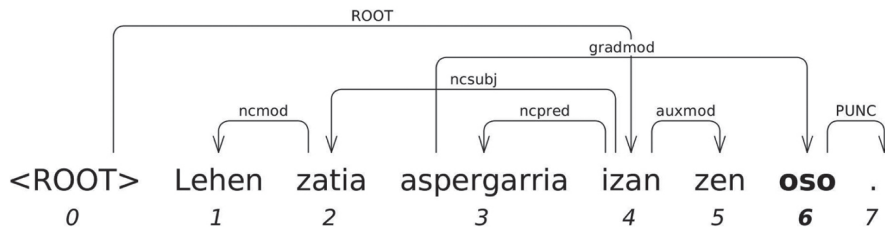
Atal honetan, datuetan oinarritutako euskararako analizatzaile sintaktiko-estatistiko bat izateko erabilitako baliabideak aurkezten dira: euskarako zuhaitz-bankua eta analizatzaile sintaktiko-sortzaileak. Lehenengo azpiatalean, euskarako zuhaitz-bankua eta etiketatze erabilitako ereduak aurkez-

tuko da; bigarren azpiatalean, teknika estatistikoetan oinarritutako bi analizatzaile sintaktiko nagusienak.

2.1. Euskarako zuhaitz-bankuak

Zuhaitz-bankua sintaktikoki etiketatutako corpus bat da. Corpuseko esaldi guztiak analizatu dira eta esaldiko hitz bakoitza etiketatu da, esaldiaren zuhaitz sintaktikoa lortuz. Esaldiaren egitura sintaktikoa adierazteko orduan, bi eredu nagusitzen dira: osagai-egitura eta dependentzia-egitura. Osagai-egituraren ereduan, esaldia osagai edo sintagma txikiagoetan banatzen da eta osagaiek esaldien egitura sintagmatikoa adierazten dute. Dependentzia-egituraren ereduan, aldiz, esaldia osatzen duten hitzen arteko dependentzia irudikatzen da, hitz bat beste baten mende dagoela adieraziz. Euskarako zuhaitz-bankua dependentzia-egituraren ereduan etiketatua izan da. Esaldiaren egitura sintaktikoak esaldia osatzen duten elementu lexikoen arteko binakako erlazioak dira. Binakako dependentzia-erlazio hori gobernatzailearen eta mendekoaren artean gertatuko da beti: i j -ren gobernatzailea (*gurasoa* edo *burua* terminoekin ezagutzen dena) edo j i -ren mendekoa (*umea* terminoarekin ezagutzen dena). Dependentzia-egitura etiketatutako eta zuzendutako grafo modura ikus daiteke, hau da, hitzez etiketatutako adabegiak eta euren arteko arku etiketatuak edo etiketatu gabeak. Dependentzia-etiketen multzo bat aukeratzekoan ohikoena da funtzio-informazioa erabiltzea (adibidez, subjektua, objektua eta abar). Hemendik aurrera dependentzia-egiturak (grafo zuzendua eta etiketatua) adierazteko, dependentzia-zuhaitza edo zuhaitza erabiliko da. Dependentzia-zuhaitzak proiektiboak (*projective*) eta ez-proiektiboak (*non-projective*) izan daitezke. Dependentzia-zuhaitzean, proiektibitatea gurutzatu gabeko arkuak daude gertatzen da. Horrela, esaldi osoko arkuak plano batean gurutzatu barietate murriztu ahal izanez gero esaldi proiektiboa dela esango da; eta, elkar gurutzatu behar izanez gero, aldiz, esaldia ez-proiektiboa dela. Adibidez, 1. irudian jasotzen den «Lehen zatia aspergarria izan zen oso.» esaldian, «oso» graduatzailea eta beraren gobernatzailea den «aspergarria» banatuta dago «izan zen» aditzarekin. Ondorioz, esaldi horretan arkuak gurutzatu egiten dira. Beraz, perpausa ez-proiektiboa da. Esaldi ez-proiektiboak maiz gertatzen dira euskara, txekiera, nederlandera eta turkieran, hots, hurrenkera libreko hizkuntzetan.

2006an izan zen X. *Computational Natural Language Learning (CoNLL)* lehiaketan (*CoNLL-X*), 13 hizkuntzatarako corpusak aurkeztu ziren dependentzietan oinarritutako analizatzaile sintaktiko-estatistikoak garatzeko. Lehiaketa horretarako erabili zen formatuari *CoNLL-X* formatua deitu zitzaion. Formatu horretan, esaldiak lerro huts batekin banatuta daude. Esaldiko hitz bakoitzaren informazioa lerro batean jartzen da,



1. irudia. Euskarako zuhaitz-bankutik hartutako esaldi baten dependentzien irudikapena.

1. taulan ikusten den modura. Hitz-forma bakoitzeko, ondoko informazioa aurki daiteke:

1. ID: esaldian hitzak duen ordena-zenbakia.
2. WORD: esaldian duen hitz-forma.
3. LEM: hitzaren lema.
4. CPOS: hitzaren kategoria.
5. POS: hitzaren azpikategoria.
6. FEATS: hitzaren ezaugarri morfosintaktikoak.
7. HEAD: hitzaren gobernatzailea.
8. DEP: gobernatzailearekiko duen dependentzia-etiketa.

Euskararen kasuan, ezaugarri morfosintaktikoak dituen zutabeak (FEATS) informazio ugari bil dezake: kasu-marka, numeroa, mugatua (mugatua ala mugagabea den), aspektua, eta abar. Adibidez, 1. taulan «Lehen zatia aspergarria izan zen oso» esaldiaren irudikapena *CoNLL-X* formatuan agertzen da.

1. taula. Euskarako zuhaitz-bankuko esaldi bat *CoNLL-X* formatuan (H. = HEAD).

ID	WORD	LEM	CPOS	POS	FEATS	H	DEP		
1	Lehen	lehen	DET	ORD		2	ncmod		
2	zatia	zati	IZE	ARR	BIZ:- KAS:ABS NUM:S	4	ncsubj		
3	aspergarria	aspergarri	ADJ	ARR	IZAUR:- KAS:ABS NUM:S	4	ncpred		
4	izan	izan	ADI	SIN	ADM:PART ASP:BURU	0	ROOT		
5	zen	izan	ADL	ADL	MDN:B1 NOR:HURA	4	auxmod		
6	oso	oso	ADB	ARR	—	3	gradmod		
7	.	.	PUNT	PUNT	—	6	PUNC		

«Lehen» hitzaren identifikatzailea 1 zenbakia da 1. taulan, eta «zattia» hitzaren identifikatzailea 2 zenbakia da. «Lehen» hitzaren HEAD zutabearen 2 bat agertzeak «zattia» hitzaren mende dagoela esan nahi du. Modu berean, «zattia» hitza «izan» laugarren hitzaren mende dago. Azkenik, «izan» aditzak (erroa denez) ez du gobernatzailearik.

Lehen euskarazko corpusa [3] *CoNLL-X* formatura egokitu zen, dependentzietan oinarritutako analizatzaile sintaktiko-sortzaileek erabili ahal izateko. Egokitzapenaren ondoren, 55.469 hitz eta 3.700 esaldikoa den *CoNLL-X* formatuko lehen euskarako zuhaitz-bankua lortu zen (hemendik aurrera EZB I zuhaitz-bankua izenaz ezagutuko dena). Lehen bertsio horrekin, *CoNLL 2007 Multilingual Dependency Parsing* lehiaketan, beste 9 hizkuntzarekin batera, eta 20 sistemaren artean ebaluatu zuten. 2010ean, dependentzia-ereduan etiketatuta dagoen euskarazko EPEC-DEP zuhaitz-bankua [4] *CoNLL-X* formatura egokitu zen, dependentzietan oinarritutako analizatzaile sintaktiko-sortzaileek erabili ahal izateko. *CoNLL-X* formatura moldatu ondoren, 150.000 hitz eta 11.225 esaldikoa den euskarazko zuhaitz-bankua lortu zen (hemendik aurrera EZB II zuhaitz-bankua izenaz ezagutuko dena). Esaldiak xx. mendeko euskararen corpus estatistikotik eta Euskaldunon Egunkaritik lortu dira. Corpusa hitz-mailan dago etiketatuta, eta dependentzia-erlazioak hitzen artean baino ez dira gertatzen. Gainera, esanahi semantikoa duen hitza ezarri da sintagmen buru. Corpusa sintaktikoki etiketatzeko, dependentzia-gramatikaren eredia erabili da. Dependentzia-erlazioak oinarri dituen etiketatze-eskema definitzeko proposamena erabili da. Testuko esaldi bakoitza erlazio gramatikal batzuekin markatzen da, mende-koaren eta bere gobernatzailearen arteko dependentzia sintaktikoa zehaztuz.

2.1.1. *Urre-patroiko ezaugarriak versus ezaugarri automatikoak*

Normalean, esperimentuak egiteko, zuhaitz-bankuan dauden ezaugarriak erabiltzen dira, hau da, hizkuntzalari-talde batek eskuz gainbegiratu-tako ezaugarriak (hemendik aurrera zuhaitz-bankuko urre-patroiko ezaugarriak deituko ditugunak). Baina euskaraz ezaugarriak era automatikokan lortzeko aukera dago, testua analizatzaile morfologikotik [5] eta desanbiguatzaile morfologikotik [6] pasatu ostean. Desanbiguazio-moduluen irteera zuzena % 97ra iristen da. Modulu horien irteerak, batez beste, 1,3 aukera eskaintzen ditu hitz-forma bakoitzeko. Erabiltzen diren analizatzaile sintaktikoen aukera bakarra behar dutenez, desanbiguazio-moduluak ematen dituen aukeretatik lehenengo aukera (sarriena) hartu da. EZB II zuhaitz-bankuko esaldiak analizatzaile morfologikotik pasatu ostean, EZB II zuhaitz-bankuko esaldiak ezaugarri automatikoen osatu dira (hemendik aurrera ezaugarri automatikoak dituen zuhaitz-bankua izenaz ezagutuko dena). Horrela, urre-patroiko ezaugarriekin eta ezaugarri automatikoen dagoen aldea konparatzeko balio izango digu, eta benetako egoeretan analizatzaile sintaktikoak izan dezakeen zehaztasuna neurtuko da.

2.2. Analizatzaile sintaktikoak

Dependentzietan oinarritutako sistemak eta zuhaitz-bankuak garatzeko, *CoNLL Shared Task* lehiaketak egin ziren 2006an eta 2007an. Hain zuzen, 2007ko lehiaketan erabili ziren zuhaitz-bankuetatik, EZB I zuhaitz-bankua aukeratu zen. Lehiaketa horietan parte hartu zuten sistema onenak aintzat hartuta esperimentuak egiteko, trantsizioetan oinarritutako sistemetatik MaltParser aukeratu da eta grafoetan oinarritutako sistemetatik, aldiz, MST-Parser. Hurrengo azpiataletan, MaltParser eta MSTParser sistemen deskribapen zabala egiten da.

2.2.1. MaltParser. Trantsizioetan oinarritutako sistema

MaltParser trantsizioetan oinarritutako analizatzaile sintaktiko-sortzaillea da. MaltParser sistemarekin sortutako modeloak, dependentzia-zuhaitza sortzeko gai dira trantsizio-segida amaitu ostean. MaltParser modeloek dependentzia-arkuak ikasi beharrean, makina abstraktu baten trantsizioak ikasten dituzte. Horiek ikasteko, beharrezkoa da dependentzietan etiketatutako eta *CoNLL-X* formatuan dagoen zuhaitz-banku bat izatea. Malt-Parser sistemak hitzaren analisi morfosintaktiko bakarra behar du; hau da, hitz-sarrerak guztiz desanbiguatuta egon behar du. MaltParser sistemarekin analizatzaile sintaktiko bat sortzeko, hiru mailatan hartu behar dira erabakiak: analisi sintaktikorako algoritmoan, ezaugarri-modeloan eta ikasketa automatikorako erabiliko den sailkatzailean.

Analisi sintaktikorako algoritmoak

MaltParser sistemak trantsizioetan oinarritutako hainbat aldaera eskaintzen ditu:

1. Nivre algoritmoen familian, denbora linealean exekututzen diren bi algoritmo proiektibo daude: Nivre Arku Azkarra (ingelesez, *arc-eager*) edo Nivre Arku Estandarra (ingelesez, *arc-standard*).
2. Pila algoritmoen familian, algoritmo proiektibo bat: Pila Proiektiboa (ingelesez, *Stackprojective*), eta algoritmo ez-proiektibo bi: Pila Azkarra (ingelesez, *Stackeager*) eta Pila Nagia (ingelesez, *Stacklazy*).
3. Covington algoritmoen familian, denbora koadratikoan exekututzen diren algoritmo ez-proiektiboen edo proiektiboen artean aukera daiteke.

Zuhaitz-banku ez-proiektiboekin, hiru aukera daude:

1. Algoritmo ez-proiektiboak erabiltzea; hau da, arku proiektiboekin eta ez-proiektiboekin lan egiten dutenak. Oro har, algoritmo

ez-proiektiboak konplexuagoak eta neketsuagoak dira konputazionalki. Adibidez, Covington algoritmo ez-proiektiboa koadratikoa da, eta Nivre-ren algoritmo proiektiboa, berriz, lineala.

2. Zuhaitz-banku ez-proiektiboa proiektibotzat jotzea eta algoritmo proiektiboak aplikatzea. Adibidez, trantsizioetan oinarritutako Nivre-ren algoritmo bakun, efiziente eta zehatza.
3. Algoritmo proiektiboak aplikatzea, baina teknika osagarriak aplikatuz. Teknika osagarri ugari daude. Adibidez: transformazio pseudo-proiektiboa (ingelesez, *pseudo-projective*) [7], aldez aurretiko eta atzeko prozesu bat erabiltzen duena.

Ezaugarri-modeloa

Atal honetan, MaltParser sistemak hurrengo trantsizioa zein den jakiteko erabiliko duen konfigurazioa edo ezaugarri-modeloa definituko da. Ezaugarri-modeloa definitzeko, analizatzailearen historian eta uneko sarreraren dauden elementuen ezaugarriak erabiliko dira. Ezaugarri-modeloa dimentsio handiko ezaugarri-bektore bat izango da. Bektorea ezaugarri sinplez osatuta dago. Adibidez, kategoria, forma, lema, kasua, numeroa, dependentzia-arkuen etiketa eta abar.

Ikasketa automatikorako sailkatzailea

Sailkatzailearen lana x sarrerako datuari dagokion y klasea iragaritzea da. Trantsizioetan oinarritutako algoritmoak sailkatzaileari galde-tuko dio pauso bakoitzean, aukera anitzen artean trantsizio onena zein den erabakitzeko orduan. Sailkatzailearen x sarrera analizatzailearen historia edo ezaugarri-modeloa izango da; eta y irteera-klasea sailkatzaileak erabakitako trantsizioa izango da. Sailkatzaileak puntuazio bat ematen dio y klase bakoitzari x ezaugarri-modelo bakoitzarekin, eta puntuazio altuena duen y klasea izango da sailkatzailearen erabakia edo irteera. Ikasteko emaitza zuzenak dituen zuhaitz-banku bat erabiliko du sailkatzaileak. MaltParser 1.4 bertsioako sistemak bi pakete ezberdin eskaintzen ditu *SVM* sailkatzailea aukeratzeko orduan, preseski, LibSVM eta LibLINEAR paketeak:

1. LibSVM paketeak bi kernel ezberdin hauek eskaintzen ditu: *RBF*, polinomiala eta lineala.
2. LibLINEAR paketeak sailkatzaile lineala erabiltzeko aukera baino ez du ematen. Baina ebazpen-aukera ezberdinak (ingelesez, *solver type*) eskaintzen ditu: *SVM* lineala, erregresio logistikoa eta klase anitzeko sailkatzailea.

2.2.2. MSTParser. Grafoetan oinarritutako sistema

Grafoetan oinarritutako analisi sintaktikoaren helburu nagusia da esaldia osatzen duten zuhaitz-arku posible guztien batasunik sortutako zuhaitz-bildumatik probabilitate handiena duen dependentzia-zuhaitza lortzea. Konektatutako grafo zuzendua izango da zuhaitza. Erpinak esaldiko hitzak izango dira eta erro-erpina edozein zuhaitzen buru izango da. Puntuazioak (ingelesez, *score*) zuhaitz posible baten probabilitatea adierazten du. Sistema ezberdinek era ezberdinetan tratatzen dute puntuazioa: sailkatzaile lineal modura, baldintzazko eran edo elkarrekiko probabilitateak gisa. Baina denek erabiltzen dute azpiegituren puntuazioen faktorea. MSTParser erabiltzeko, beharrezkoa da *CoNLL-X* formatuan dependentzietan etiketatutako zuhaitz-banku bat izatea. MSTParser sistemak hitzaren analisi morfosintaktiko bakarra behar du; hau da, sarrera guztiz desanbiguatua egon behar da. MSTParser sistemaren deskribapena egiterakoan, honako hiru ezaugarri nagusi hauek erabiliko dira: analisi sintaktikorako algoritmoa, ezaugarri-modeloa eta ikasketa automatikoa.

Analisi sintaktikorako algoritmoak

Analisi sintaktikorako algoritmoa aukeratzeko orduan, bigarren edo lehen mailako modeloen arteko aukera eta arku proiektiboekin edo ez-proiektiboekin lan egiteko aukera eskaintzen dira. Zuhaitza osatzen duten arkuen puntuazioen faktORIZAZIOA erabiltzen denean, lehen mailako modeloa edo puntuazio-funtzioa erabiltzen da. Baina arkuak independente modura hartuz gero, arriskua dago hizkuntzetan dauden egitura konplexuen pisua era egokian ordezkatzuta ez agertzeko. MSTParser sistemaren bigarren mailako algoritmoa erabiliz gero, kontuan har daiteke guraso bereko bi elkarren ondoko umeen arkuen puntuazioa. Ondoko dependentzia-egituren informazio guztia kontuan hartzea konputazionalki ezinezkoa denez, MSTParser sistemako bigarren mailako algoritmoek modu horizontalean aplikatzen dute Markov-en hurbilpena. Dependentzia-egituraren kalkuluan, ume baten arku bakarreko informazioari ondoko ezkerreko senidearen arku-informazioa gehitzen zaio. Etiketa asmatu behar izanez gero, aurreko informazioari dependentzia-etiketaren informazioa gehituko zaio. Analisi proiektiboan, Eisner algoritmoa erabiltzen da. Algoritmo honek zuhaitz proiektibo guztien gaineko bilaketa $O(n^3)$ denboran gauzatzen du, lehen edo bigarren mailako ezaugarriak erabiliz. Analisi ez-proiektiboan, aldiz, Chu-Liu-Edmonds algoritmoa erabiltzen da. Algoritmo honek zuhaitz guztien gaineko bilaketa $O(n^2)$ denboran gauzatzen du, lehen edo bigarren mailako ezaugarriak erabiliz.

Ezaugarri-modeloa

MSTParser sistemak erabiltzen du, zehazki, esaldia osatzen duten zuhaitz-arkuen parametroen batuketan maximizatzen duen *MST* (*Maximum*

Spanning Tree) algoritmo globala. MSTParser sistemarekin sortutako analizatzaile sintaktikoak puntuazio gehien dituzten arkuekin geratzen dira. Arkuen puntuazioa kalkulatzeko, arkuen mailan definitutako ezaugarrien multzo bat erabiltzen da. Adibidez, arku bakarra osatzen duten gurasoen eta umearen hitz-forma, kategoria, azpikategoria, informazio morfologikoa eta abar.

Ikasketa automatikorako sailkatzailea

Ezaugarrien pisua ikasteko MSTParser sistemaren sailkatzaileak lineako inferentzian oinarritutako (*online inference-based*) *MIRA* (*Margin Infused Relaxed Algorithm*) metodoa erabiltzen du. *MIRA* izeneko ikasteko algoritmo iteratibo bat da, eta entrenamenduko esaldiekin N iterazio egiten ditu, W bektoreak pisuaren balioa finkatu eta entrenamenduko datuak linealki banatzeko gai izan arte. Nahiz eta N ren balio lehenetsia 10 izan, beraren balioa aldatu egin daiteke. Iterazio bakoitzean entrenamenduko instantzia (esaldi) guztiak tratatzen ditu. Pisu-bektorean gutxieneko aldaketak egiten ditu, gutxieneko puntuazioen marjina mantentzen den bitartean.

3. TEKNIKAK

Atal honetan, analizatzaile sintaktikoen emaitzak hobetze aldera aplikatu ditugun teknikak azaltzen dira. Lehenengo azpiatalean, euskararen izaera kontuan hartuta, euskararako egokiak diren zuhaitz-transformazioak azaltzen dira. Bigarren azpiatalean, lehenengo analizatzaile baten irteeran lortutako egiturazko ezaugarriak bigarren analizatzailearen sarrera aberasteko erabiltzen den pilaketa-teknika azaltzen da. Bukatzeko, sortutako oinarritzko sistemen eta sistema hedatuen (zuhaitz-transformazio eta pilaketa sistemen) irteerak konbinatzeko bozketa, bidezko teknika azaltzen da.

3.1. Zuhaitz-transformazioak

Nahiz eta buru-osagarri eta buru-modifikatzaile egitura gehienek analisi berdintsua izan dependentzia-gramatikan, badaude eztabaidagarriak diren egitura asko; besteak beste honako hauek: aditz-laguntzailea aditz nagusien gobernatzailea izatea edo ez; determinatzaile-sintagman determinatzailea burua izatea edo ez; postposizio-sintagman azken hitza burua izatea edo ez; koordinazioetan, juntagailua edo koordinazioaren lehenengo edo azken osagaia buru izatea edo ez. Erabakitzeke unean teoria ezberdinak daude. Etiketatze teoria desberdinen eragina aztertzeke, sekzio honetan euskarako zuhaitz-bankuari aplikatutako aurretiko eta ondorengo prozesaketa ezberdinak azalduko dira: sintagmen transformazioa, mendeko

perpauzen transformazioa eta koordinazioaren transformazioa. Dependentsia-zuhaitzetan aplikatzen diren transformazioek kutxa beltzaren metodoa erabiltzen dute. Euskarako zuhaitz-bankuko transformazioak egiteko, oro har, ondoko bost urratsak emango dira:

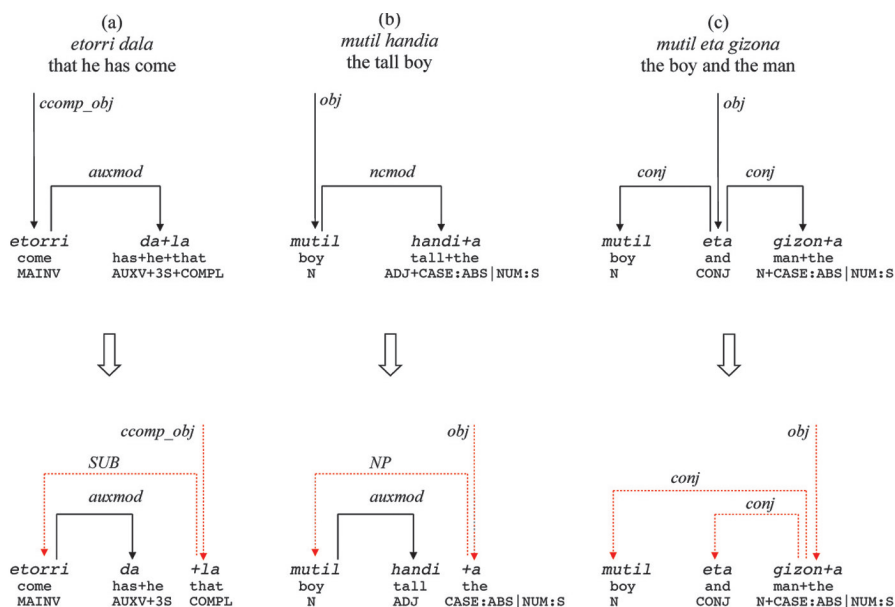
1. Entrenamendu multzoko esaldiak transformatzea.
2. Transformatutako multzoarekin ikasiz, analizatzaileen sortzaileak analizatzaile sintaktiko baten modeloa sortzea.
3. Test multzoko esaldiak sortutako analizatzaile sintaktikoarekin analizatzea.
4. Analizatzaile sintaktikoaren irteerari alderantzizko transformazioa aplikatzea.
5. Bukatzeko, alderantzizko transformazioaren irteera eta test multzoko esaldiak alderatzea.

Atalez atal, euskarako zuhaitz-bankuetan aplikatutako hiru transformazio-mota azalduko dira: sintagmen transformazioa, mendeko perpauzen transformazioa eta koordinazioaren transformazioa. Euskara hizkuntza buru-azkena izateak eta ezaugarri morfologiko garrantzitsuenak (kasua, numeroa, mendeko perpauzen informazioa, eta abar) sintagmaren azkeneko hitzean dituen hizkuntza izateak transformazioak gauzatzea ahalbidetu du.

3.1.1. *Sintagmen transformazioa*

Sintagmen transformazioa egiteko, bi arrazoi nagusi hartu dira kontuan:

1. Euskarako zuhaitz-bankuak hitz mailan etiketatuta daude; hau da, dependentzia-erlazio sintaktikoak hitzen artean etiketatu dira (ikus goiko *b* adibidea 2. irudian). Euskara hizkuntza eranskaria eta morfologikoki aberatsa izanik, hitzaren azken morfema banatu ostean etiketatu dira dependentzia-erlazio sintaktikoak. Horrela, ikasketa automatikorako mesedegarria izan daitekeelakoan gaude (ikus beheko *b* adibidea 2. irudian).
2. Euskarako zuhaitz-bankuak etiketatzerakoan, esanahi semantikoa duen hitza jarri da sintagmen buru modura. Euskara hizkuntza buru-azkena izanik, hau da, sintagmaren ezaugarri morfosintaktiko garrantzitsuenak (kasu-marka, numero-marka, mendeko-marka, eta abar) hitz-hurrenkeran azkeneko kokagunea hartzen duen osagaian biltzen direla ikusirik, ondoko transformazioa burutuko da: hitzaren azken morfema banatu ostean, sintagmen burua, buru semantikoa izan beharrean, sintagmen azkeneko kokagunea duen hitzaren morfema printzipala izango da. Horrela, sintagmen buruak ezaugarri



2. irudia. Euskarako zuhaitz-bankuari aplikatutako mendeko perpausen (a), sintagmen (b), eta koordinazioaren (c) zuhaitz-transformazioak.

morfosintaktikoak dituenean, aditzarekin lotzerako mementuan, lagungarri izan daiteke dependentzia-erlazio zuzena asmatzeko (ikus beheko *b* adibidea 2. irudian).

3.1.2. Mendeko perpausen transformazioa

Oro har, mendeko perpausak aditz nagusiari edo aditz laguntzaileari morfema bat gehituz sortzen dira euskaraz. Mendeko morfemaren informazioa oso garrantzitsua da, alde batetik, esaldiaren perpaus nagusia eta mendeko perpausa banatzeko, eta, beste aldetik, mendeko perpausak bere buruarekin zein motatako dependentzia-erlazioa duen jakiteko. Euskarako zuhaitz-bankuetan, mendeko perpausak aditz nagusiaren (buru semantikoaren) inguruan daude antolatuta. Baina gehienetan mendeko perpausen morfema (buru sintaktikoa) aditz laguntzaileari lotuta etortzen da; hau da, ez dator bat buru semantikoarekin. Aurreko *a* adibidean (2. irudian) ikus daitekeenez, «etorri dela» mendeko perpausa konpletiboa «da» aditz laguntzaileari «la» mendeko konpletiboaren morfema gehituz sortu da. Eta «dela» hitzean mendeko konpletiboaren informazio guztia dagoela ikus daiteke: ezaugarrien artean mendeko konpletiboaren informazioa («COMPL») eta «la» mendeko morfema. Horrela, analizatzaileak

mendeko dependentzia-erlazioa zein den asmatzea zailago izango du. Alde batetik, «etorri» mendeko perpausaren aditz nagusiak perpaus nagusiko aditz nagusiarekin lotzeko orduan ez duelako mendekoa izatearen inolako informaziorik. Eta bestetik, «dela» zuhaitz-sintaktikoan «etorri» hitza baino maila bat beherago dagoelako. Analizatzailearen zehaztasuna hobetzeko asmoarekin, hurrengo zuhaitz-transformazioa burutu da. Horretan, «dela» hitzari «la» morfema konpletiboa banatu ostean, «etorri» aditz nagusiaren gainetik (dependentzia-zuhaitzean bi maila) igo da. Horretarako, «etorri» aditzaren buru «la» morfema eta euren arteko erlazio-etiketa SUB dependentzia-erlazio berria jarriko dira. Modu horretan, «la» morfemak lehen «etorri» aditz nagusiak zituen burua eta erlazio-mota hartuko ditu; hau da, aditz nagusia buru eta «ccomp obj» erlazio-mota hartuko ditu.

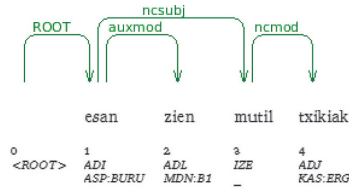
3.1.3. *Koordinazioaren transformazioa*

Koordinazioa maila bereko edo estatus gramatikal bereko perpausak nahiz sintagmak lotzen direnean gertatzen da. Koordinazioan elkartzen diren osagaiak juntagai deituko zaie, eta elkartze hori gauzatzeko erabiliko den tresna gramatikala juntagailua izango da. Hitzen artean erlazio bitarra argia denean, dependentzietan oinarritutako teoriak ondo baino hobeto egiten dute lan. Baina koordinazioaren kasuan, juntagailuaren eta juntagaien artean erlazio bitarrak sortzeko orduan, teoria edo estilo ezberdinak daude. Gureari dagokionez, EZB zuhaitz-bankuetako esaldi koordinatuak Praga estiloa jarraituz etiketatu dira. Praga estiloan (PE) juntagailua koordinazioa osatzen duten juntagaien burua da (ikus goiko *c* adibidea 2. irudian). Euskara hizkuntza buru-azkena izanik eta berezitasun morfologiko garrantzitsuenak sintagmaren azken juntagaian kokatuta daudela jakinda, Melcuk Estiloaren Simetrikoa aplikatu da (hemendik aurrera, MES izenarekin adieraziko da). MESaren koordinazioaren azken juntagaia koordinazioaren buru jarriko da eta gainerako osagaiak osteko osagaiarekin lotuko dira, kate bat osatuz (ikus beheko *c* adibidea 2. irudian).

Sintagmen, mendeko perpausen eta koordinazioaren transformazioak bateragarriak eta mesedegarriak diren jakiteko, bata bestearen ondoren aplikatu dira. Moldaketa horiek zuhaitz-bankuan izan duten eragina aztertzeko orduan, honako datuak hauteman daitezke: mendeko perpausen transformazioak arku guztien % 7,92 aldatzen du, koordinazioaren transformazioak arku guztien % 10,38 eta sintagmen transformazioak arku guztien % 38,01. Halaber, hiru transformazioek arku guztien % 56 aldatzen dute. Hori dela eta, alderantzizko konbinazioa ez da lan erraza izan. Azkenik, EZB II zuhaitz-bankuan transformazio guztiak aplikatu ostean, esaldi ez-proiektiboen kopurua % 13,28tik % 31,84ra igotzen dela ikusi da.

3.2. Analizatzaileen pilaketa edo *stacking*

Analizatzaileen pilaketan bi analizatzaile bateratzeko, lehenengo analizatzailearen irteeran lortutako informazioa bigarren analizatzailearen sarrera aberasteko erabili da [8] eta modelo bi elkarren osagarri izan ahal direla probatu da [9]. Gainera, horrek aukera eman du, lehenengo analizatzailearen irteeran lortutako dependentzia-zuhaitzetatik informazioa goratzeko eta irteerako datuak gehiago aberasteko.



(a) Lehenengo faseko analizatzailearen irteera

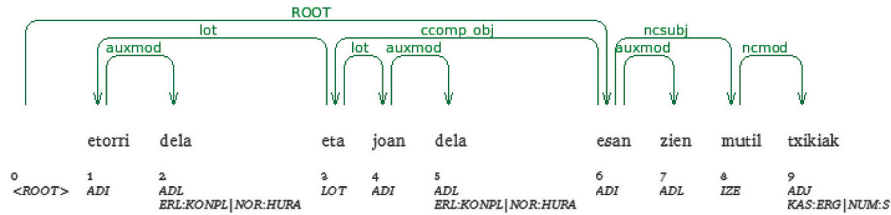


(b) Egitura-ezaugarriekin aberastutako lehenengo faseko analizatzailearen irteera

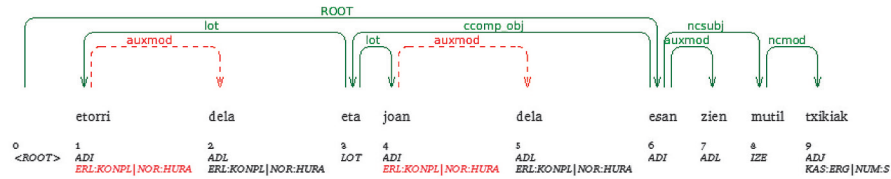
3. irudia. Dependentzia-zuhaitzetatik goratutako egitura-ezaugarriak (GKAT = Gurasoaren KATEGoria; GDEP = Gurasoaren DEPENDentzia; GNOR = Gurasoaren NORabidea; ES = ESKuma; EZ = EZkerra).

Bigarren analizatzailea aberasteko, era askotako informazioa erabil daiteke: hitzen kategoria, ezaugarri morfosintaktikoak, dependentzia-erlazioa eta abar. Erabiliko diren ezaugarriak bi multzotan sailkatuko dira: egitura-ezaugarriak eta ezaugarri linguistikoak.

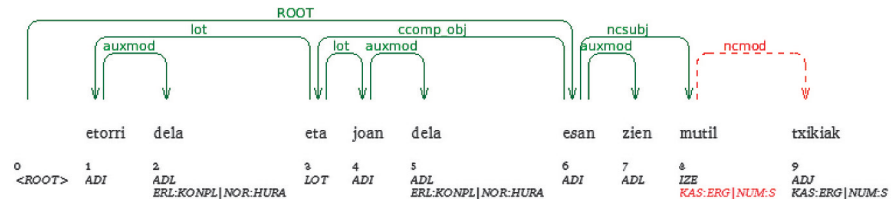
- Egitura-ezaugarriak. Uneko adabegia aberasteko, egitura hierarkikoa osatzen duten adabegien ezaugarriak erabili dira. Adibidez, uneko hitzaren ezaugarrietan gurasoen, umeen eta aitonen kategoria, norabidea eta dependentzia-etiketak gehitu dira. Kasurako, 3b irudian, «txikiak» adjektiboaren gurasoa den «mutil» izenaren informazio hau: Gurasoaren KATEGoria (GKAT: IZE), Gurasoaren DEPENDentzia (GDEP: nc-subj) eta Gurasoaren NORabidea (GNOR: EZkerra).



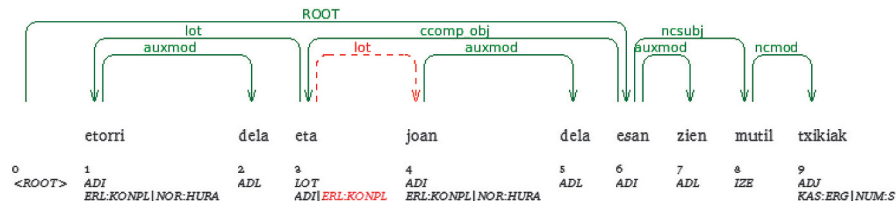
(a) Lehenengo faseko analizatzailearen irteera



(b) Aditz konposatuen aberasketa ezaugarri linguistikoekin



(c) Sintagmen aberasketa ezaugarri linguistikoekin



(d) Koordinazioaren aberasketa ezaugarri linguistikoekin

4. irudia. Dependentsia-zuhaitzetatik goratutako ezaugarri linguistikoak.

– Ezaugarri linguistikoak. Euskara buru-azkena izanik, lehenengo analizatzailearen irteera ematen duten ezaugarri morfosintaktikoek (numeroa, kasua eta mendeko perpausa besteak beste) bigarren analizatzailea aberasteko, ume-adabegititik guraso-adabegira goratzeko hiru printzipio linguistiko hauetan oinarritu gara:

- Aditz konposatuen aberasketa: euskarako zuhaitz-bankuetan, aditz konposatuaren burua, aditz laguntzailea (buru sintaktikoa)

izan beharrean, aditz nagusia (buru semantikoa) dela ikus daiteke. Aditz laguntzaileari lotuta datorren informazio oso garrantzitsua (mendeko perpausen morfema eta nor, nori eta nork kasuen pertsona eta numeroa) aditz nagusira goratu da, informazio hori buruarekiko hurbilago egotea analisisian mesedegarria izan daitekeelakoan. Esate baterako, 4a irudian, «dela» hitzean mendeko perpausa konpletiboaren morfema dago, eta suposa deza-kegu aditz nagusiaren behean agertzean, «eta» aditz nagusiarekin «ccomp obj» mendeko erlazio-motarekin batzeko zailtasunak izan ditzakeela. Lehenengo faseko analizatzailearen irteera aberasteko eta bidez batez bigarren faseko analizatzaileari laguntzeko, honako informazioa (ikus 4b irudian) goratuko da: «dela» hitzaren ezaugarri zutabearen mendeko konpletiboaren informazioa (ERL: KONPL) eta aditzaren pertsonaren eta numeroaren informazioa (NOR: HURA), «dela» aditz laguntzaileetatik «etorri» eta «joan» aditz nagusietara goratu dira, «auxmod» dependentzia-erlazioaren bitartez.

- Sintagmen aberasketa: euskarako zuhaitz-bankuak sintagmen buru modura etiketatzerakoan, esanahi semantikoa duen hitza jarri da. Sintagmaren azkeneko kokagunea hartzen duen hitzari lotuta datorren informazio oso garrantzitsua (kasu-marka eta numeroa) sintagmaren burura goratuko da. Informazioa goratzeko, «detmod» determinatzaile-erlazioa (ingelesez, *determinant modifier*) eta «ncmod» ez-perpausa erlazioa (ingelesez, *non-clausal modifier*) erabiliko dira. Informazio hori buruarekiko hurbilago egotea analisisian mesedegarria izan daitekeelakoan gaude. Adibidez, 4c irudian, «txikiak» izenondoaren ezaugarri-zutabearen dagoen ergatibo-informazioa (KAS: ERG) «txikiak» izenondotik «mutil» sintagmaren burura nola goratu den ikus daiteke, «ncmod» dependentzia-erlazioaren bitartez.
- Koordinazioaren aberasketa: euskarako zuhaitz-bankuko esaldi koordinatuak Praga estiloa jarraituz etiketatu dira; hau da, juntagailua koordinazioa osatzen duten juntagaien buru modura etiketatu da. Euskara hizkuntza buru-azkena izanik eta berezitasun morfologiko garrantzitsuenak (kasu-marka, mendeko erlazioa, mugatasuna, numeroa, eta abar) sintagmaren azkeneko osagaian kokatuta daudela jakinda, koordinazioaren azken juntagaitik juntagailura kategoria, kasu-marka eta mendeko erlazioa goratuz probatuko da. Adibidez, 4b iruditik 4d irudira gertatu den koordinazioaren aberasketa ikus daiteke; hain zuzen, «ADI» kategoria eta «ERL: KONPL» mendeko konpletiboaren informazioa «joan» koordinazioaren azken juntagaitik «eta» juntagailura, «lot» dependentzia-erlazioaren bitartez nola goratu diren.

Analizatzaileen pilaketa egiteko, aukera hauek probatu ditugu:

- Aberasteko, egitura-ezaugarriak edo linguistikoak erabiltzea.
- Erabilitako sistemak izaera berdinekoak ($M alt_{M alt}$ edo $M ST_{M ST}$) edo desberdinetakoak ($M alt_{M ST}$ edo $M ST_{M alt}$) izatea.
- Urre-patroiko ezaugarriekin edo ezaugarri automatikoekin probatzea.
- Tamaina desberdineko zuhaitz-bankuak erabiltzea.

3.3. Bozketa bidezko konbinazioa

Aurreko ataletan bi motatako analizatzaileak lortu dira: oinarrizko algoritmoak erabiltzen dituztenak, eta, zuhaitz-transformazioak eta pilaketa moduko teknika hedatuak erabiltzen dituztenak. Bozketa bidezko konbinazioan analizatzaile desberdinen irteerak kontuan hartzen dira, irteera bateratu eta egokia lortzeko asmoarekin. Hain zuzen, [10] lanean, etiketa gabeko analisi-dependenzian, trantsizioetan oinarritutako hiru modeloren eta grafoetan oinarritutako modelo baten irteerak konbinatu zituzten, eta oinarrizko algoritmo onenekin baino emaitza hobekak lortu zituzten. *CONLL 2007 Multilingual Dependency Parsing* lehiaketan [11], trantsizioetan oinarritutako 6 modeloren irteerak konbinatuz lehenengo postua lortu zuten, 19 hizkuntzarekin eta 20 sistemarekin lehiatu ostean.

Ildo beretik, gure esperimentuak gauzatzeko, dependentzietan oinarritutako analizatzaileen irteerak bateratzeko *Maximum Spanning Tree* ko Chu-Liu/Edmonds algoritmoa erabiltzen duen MaltBlender tresna erabili da. MaltBlender-ek, analizatzaile sintaktikoen irteera bakoitzari pisu bat ematerakoan, pisu-eskema ezberdinetan oinarritzeko aukera eskaintzen du: POS kategoriaren, dependentzia-erlazioaren edo analizatzaile bakoitzaren puntuazioaren araberakoa. Bestalde, [12] lanean, pisu-eskema ezberdinak aplikatu ostean, analizatzaileen aniztasuna faktore garrantzitsuenetako bat zela probatu zuten, pisu-eskemen aldean.

MaltBlender-ekin pisurik gabeko eta pisu-eskema ezberdinekin probatu ostean, pisurik gabeko konbinaketekin lortu dira emaitzarik onenak. Hori dela eta, pisurik gabeko eskema erabili da gainerako esperimentuetarako. Analizatzaileen aniztasunaren eragina aztertzeko, analizatzaile hauek aukeratu dira:

- Oinarrizko analizatzaileak: multzo honetan MaltParser sistemako Nivre, Stacklazy eta Covington algoritmoak eta MSTko algoritmo proiektibo eta ez-proiektiboak sartu dira.
- Analizatzaile hedatuak: oinarrizko algoritmoen zuhaitz-transformazioak eta pilaketa-teknikak gehitu zaizkie.

- Aniztasun handiagoa egoteko, oinarritzko algoritmoekin zuhaitz-bankuko esaldiak ezkerretik eskuinera eta eskuinetik ezkerreara tratatu dira, eta salbuespen bakarra MSTko algoritmo globala izan da; izan ere, MSTko algoritmo globalarekin emaitza berbera lortzen da, esaldiak ezkerretik eskuinera edo eskuinetik ezkerreara tratatzean.

Orain arte, bozketa-sistema konbinatzeko orduan, analizatzaile multzo mugatua erabili dituzten lanek [11] eta analizatzaile multzo zabala erabili dituzten lanek [13], beti n analizatzaile multzotik aukeratzen dituzte x analizatzaile onenak; hemendik aurrera, x onenen konbinazioa deituko diogu horri.

Baina konbinatzeko garaian, analizatzaileen aniztasunak (osagarritasuna) garrantzi handiagoa duen aztertzeke, puntakoak ez diren analizatzaileak konbinatuta probatu da. Adibidez, 3 analizatzailearen konbinazioak edo 3 onenen konbinazioak doitasun hobea emateko aukera bera duten probatzeko, ondoko esperimentuak gauzatu dira:

- x onenen konbinazioa eta ausazko onenen konbinazioa probatu dira.
- Analizatzaile hedatuak gehitu dira, zuhaitz-transformazioak eta pilaketak ekar dezaketen aniztasuna onuragarria den ala ez ikusteko. Horretarako, alde batetik, oinarritzko algoritmoak konbinatu dira, eta, beste aldetik, analizatzaile hedatuekin batera konbinatu dira.

4. EMAITZAK

Atal honetan, analizatzaile sintaktikoen emaitzak hobetze aldera erabiliko hiru tekniken emaitzak aurkeztuko dira. Lehenengo azpiatalean, sintagmen, mendeko perpausen eta koordinazioaren transformazioen emaitzak aztertuko ditugu. Bigarren azpiatalean, pilaketa-teknika ezberdinekin lortutako emaitzak agertzen dira. Hirugarren azpiatalean, sortutako oinarritzko sistemen eta sistema hedatuen (zuhaitz-transformazio eta pilaketa sistemen) irteerak bozketa bidezko MaltBlender tresnarekin konbinatu ostean lortutako emaitzak aurkeztuko dira.

Esperimentuak egiteko, urre-patroiko ezaugarriak eta ezaugarri automatikoak erabiltzeaz gain, zuhaitz-bankuaren tamaina bikoizteak (bai oinarriarekin, bai aplikatutako transformazioekin) analisisan duen eragina aztertuko da. Horretarako, EZB II zuhaitz-bankuko % 80 den entrenamendu-multzoa 500, 1.000, 2.000, 4.000 eta 9.100 esaldiko azpi-multzoetan banatu da, eta garapenerako EZB II zuhaitz-bankuko % 10 eta ebaluaziorako EZB II zuhaitz-bankuko % 10 utzi da. Esperimentu guztiak garapenerako datuekin egin dira, eta sistema onena ebaluaziorako azpimultzoarekin probatzeko utzi da.

Analizatzaileak iragarritako informazio sintaktikoak zuhaitz-bankuko etiketa errealekin konparatu dira. Lan honetan erabili den neurria *Labeled Attachment Score (LAS)* izan da. LAS neurriak analizatzailearen erabakia zuzena izateko probabilitatea neurtzen du, baina erabaki zuzena izateko bai dependentzia-etiketa bai gobernatzaile-esleipena, biak zuzenak izan behar dira.

4.1. Zuhaitz-transformazioak

Ondoko 2. eta 3. tauletan emaitza hauek agertzen dira: EZB II zuhaitz-bankuko esaldiak transformatu barik sortutako oinarritzko sistemak (Oin.) eta EZB II zuhaitz-bankuko esaldiak sintagmen, mendeko perpausen eta koordinazioen zuhaitz-transformazioak (ZT) burututa, urre-patroiko ezaugarriekin (UE) eta ezaugarri automatikoekin (EA), hurrenez hurren. Tauletako datuak aztertuz, honako emaitza hauek aipa daitezke:

2. taula. Oinarritzko sistemak (Oin.) eta hiru zuhaitz-transformazioak (ZT) urre-patroiko ezaugarriekin ebaluatzeko multzoan (Niv. Arku Est. + Pse. = Niv. Arku Estandarra + Pseudoproiektiboarekin).

UE	Stacklazy		Covington		Niv. Arku Est. + Pse.		MST	
	Oin.	ZT	Oin.	ZT	Oin.	ZT	Oin.	ZT
500	70,54	73,31 (+2,77)	70,03	73,35 (+3,32)	70,97	74,20 (+3,23)	70,67	72,40 (+1,73)
1.000	72,43	74,94 (+2,51)	72,16	75,77 (+3,61)	72,40	75,36 (+2,96)	73,35	74,56 (+1,21)
2.000	74,91	77,12 (+2,21)	75,25	77,16 (+1,91)	74,94	77,64 (+2,70)	75,66	77,45 (+1,79)
4.000	77,49	79,32 (+1,83)	77,45	78,76 (+1,31)	77,53	79,36 (+1,83)	78,39	79,57 (+1,18)
9.100	78,79	80,55 (+1,76)	78,71	80,23 (+1,52)	78,89	80,80 (+1,91)	80,17	80,87 (+0,70)

3. taula. Oinarria (Oin.) eta hiru zuhaitz-transformazioak (ZT) ezaugarri automatikoekin ebaluatzeko multzoan (Niv. Arku Est. + Pse.: Niv. Arku Estandarra + Pseudo-proiektiboarekin).

EA	Stacklazy		Covington		Niv. Arku Est. + Pse.		MST	
	Oin.	ZT	Oin.	ZT	Oin.	ZT	Oin.	ZT
500	68,11	71,05 (+2,94)	68,72	71,79 (+3,07)	68,67	71,87 (+3,20)	69,02	70,58 (+1,56)
1.000	70,45	73,06 (+2,61)	70,60	73,75 (+3,15)	70,87	73,84 (+2,97)	71,22	72,66 (+1,44)
2.000	73,17	74,80 (+1,63)	73,18	75,48 (+2,30)	73,24	75,38 (+2,14)	73,44	74,93 (+1,49)
4.000	75,06	76,71 (+1,65)	75,40	76,85 (+1,45)	75,42	77,47 (+2,05)	76,22	76,97 (+0,75)
9.100	76,83	78,26 (+1,43)	76,98	78,44 (+1,46)	77,31	79,09 (+1,78)	77,88	78,58 (+0,70)

- **Ezaugarri automatikoak vs urre-patroiko ezaugarriak.** Aurreko, 2. eta 3. tauletan agertzen diren emaitzak alderatuz gero, analizatzaile morfologikoak automatikoki lortutako ezaugarri morfosintaktikoak erabiltzerakoan ia 2 puntuko galera dagoela ikus daiteke sistema guztietan. MSTParser sistemarekin, urre-patroiko ezaugarriak erabiltzetik ezaugarri automatikoak erabiltzera igarotzen denean, 2,3 puntuko galera dago; aitzitik, MaltParser sistema guztiarekin 1,5 eta 2 puntu bitarteko galera baino ez dago. Horrela esan daiteke, MSTParser sisteman informazio morfologikoak eragin handiagoa duela. Hala ere, MSTParser sistemarekin lortu da oinarrikerik onena, bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin.
- **Transformazioen konbinazioa.** Transformazioen konbinazioen emaitza hobeak lortzen dira MaltParser sistemarekin, MSTParser sistemarekin baino, eta sistema bien arteko aldea desagertu egiten da. Esperimentu guztietan, transformazioen konbinazioarekin hobekuntza handiagoa lortu da entrenamendu-azpimultzoaren tamaina txikiagoa denean, handiagoa denean baino. Algoritmo gehienetan, transformazioen konbinazioaren eragina zuhaitz-bankua bikoiztearen adinakoa dela esan daiteke. Hiru transformazioen konbinazio guztiak esanguratsuak dira estatistikoki; alabaina, salbuespena ezaugarri automatikoak erabiltzean gertatzen da MSTParser 4.000 eta 9.100 tamainarekin. Emaitza bat aipatzearen, MaltParser sistemako Nivre Arku Estandar algoritmoa transformazio pseudoproiektiboarekin konbinatuz lortutakoa da; izan ere, ezaugarri automatikoekin, 79,09ko zehaztasuna lortu da LAS neurrian, urre-patroiko ezaugarriekin lortutako emaitza gaindituz (LAS = 78,89).
- **Zuhaitz-bankuaren tamaina bikoizten denean.** Bai oinarriarekin, bai aplikatutako transformazioekin, algoritmo gehienetan, transformazioen konbinazioaren eragina zuhaitz-bankua bikoiztearen adinakoa dela esan daiteke. Transformazioen konbinazioarekin emaitza hobeak lortzen dira zuhaitz-banku txikietan.

4.2. Analizatzaileen pilaketa edo *stacking*

Zuhaitz-bankuaren tamainak pilaketan izan dezakeen eragina aztertzeko esperimentu honetan entrenatzeko erabili diren zuhaitz-bankuak 500, 1.000, 2.000, 4.000 eta 9.100 esaldikoak (zuhaitz-banku osoa) izan dira. MaltParser algoritmo ezberdinekin probatu ostean, bigarren fasean Stacklazy algoritmoarekin lortu dira emaitzarik onenak. Horrela, 4. taulan, bigarren fasean erabiliko den Stacklazy algoritmoa aberasteko, lehenengo fasean lortutako MSTko ezaugarriak ($Stacklazy_{MST}$), Covington-eko ezaugarriak ($Stacklazy_{Covington}$) edo Stacklazy-ko ezaugarriak ($Stacklazy_{Stacklazy}$) erabili dira. Pilaketa-teknikarekin oinarriarekiko hobekuntza esanguratsuak lortu dira. Gure esperimentuak harago doaz, eta kasu errealetan aplikaga-

rritasuna duten jakiteko, ezaugarri automatikoak erabiltzeak zehaztasunean izan dezakeen eragina aztertuko da. Esaterako, 5. taulan, ezaugarri automatikoko zuhaitz-bankuekin erabilitako pilaketetan, ezaugarri automatikoekin eta tamaina ezberdineko zuhaitz-bankuekin lortutako emaitzak ikus daitezke.

4. taula. Urre-patroiko ezaugarriak erabiliz, pilaketan bigarren analizatzailea aberasteko ezaugarri linguistikoak eta egitura-ezaugarriak erabili ostean lortutako emaitzak (Oin. = Oinarria, Ezaug. ling.= Ezaugarri linguistikoak, Egit. ezaug. = Egitura-ezaugarriak).

Testa (LAS)	Stacklazy	<i>Stacklazy</i> _{Stacklazy}	<i>Stacklazy</i> _{MS}	<i>Stacklazy</i> _{C ovington}
Kop.	Oin.	Ezaug. Ling.	Egit. Ezaug.	Egit. ezaug.
500	70,54	71,82 (+1,28)	71,04 (+0,50)	70,23 (-0,31)
1.000	72,43	73,30 (+0,87)	73,23 (+0,80)	72,61 (+0,18)
2.000	74,91	75,66 (+0,75)	76,52 (+1,61)	75,83 (+0,92)
4.000	77,49	78,09 (+0,60)	79,27 (+1,78)	78,16 (+0,67)
9.100	78,79	79,36 (+0,57)	80,77 (+1,98)	80,02 (+1,23)

5. taula. Ezaugarri automatikoak erabiliz, pilaketan bigarren analizatzailea aberasteko ezaugarri linguistikoak eta egitura-ezaugarriak erabili ostean lortutako emaitzak (Oin. = Oinarria, Ezaug. Ling. = Ezaugarri Linguistikoak, Egit. Ezaug. = Egitura Ezaugarriak).

Testa (LAS)	Stacklazy	<i>Stacklazy</i> _{Stacklazy}	<i>Stacklazy</i> _{MS}	<i>Stacklazy</i> _{C ovington}
Kop.	Oin.	Ezaug. Ling.	Egit. Ezaug.	Egit. Ezaug.
500	68,11	69,02 (+0,91)	68,58 (+0,47)	67,49 (-0,62)
1.000	70,45	71,68 (+1,23)	71,30 (+0,85)	70,58 (+0,13)
2.000	73,17	73,74 (+0,57)	73,89 (+0,72)	73,56 (+0,39)
4.000	75,06	75,68 (+0,62)	76,59 (+1,53)	76,08 (+1,02)
9.100	76,83	77,22 (+0,39)	78,55 (+1,72)	77,52 (+0,69)

Tauletako datuak aztertuz, emaitza hauek aipa daitezke:

- **Zein da onena pilaketan? Izaera bereko sistemekin ala izaera desberdineko sistemekin eraikitako pilaketa? Izaera bereko baina algoritmo desberdineko sistemekin eraikitako pilaketa?** Oinarriarekiko hobekuntzarik onena pilaketaren lehen eta bigarren fasean erabilitako sistemak izaera desberdinekoak direnean gertatzen da, [14] lanean probatu zuten modura. Oinarriarekiko bigarren

hobekuntzarik onena, bi faseetan, izaera bereko sistema baina algoritmo ezberdinak erabiltzean lortu dira. Hobekuntza txikienak bi faseetan izaera bereko sistema eta algoritmo berbera erabiltzerakoan lortu dira. Horrela, pilaketan sistema eta algoritmo ezberdinen osagarritasuna mesedegarria dela esan dezakegu.

- **Tamainaren eragina pilaketan.** *Stacklazy*_{Stacklazy} pilaketarekin zuhaitz-banku txikiekin lortzen da hobekuntza handiena; *Stacklazy*_{MST} eta *Stacklazy*_{Covington} pilaketekin, aldiz, zuhaitz-banku handiekin lortzen da hobekuntza. Emaizta ez-esanguratsuak *Stacklazy*_{Covington} pilaketan baino ez dira gertatu, urre-patroiko ezaugarriekin, 500 eta 1.000 esalditako zuhaitz-bankuetan.
- **Ezaugarri automatikoak pilaketan.** Ezaugarri automatikoekin lortutako emaitzak esanguratsuak izan arren, urre-patroiko ezaugarriekin lortutakoak baino apalagoak direla ikus daiteke. Analizatzaile morfologikoak emandako ezaugarri okerrak sistema aberasteko erabiltzean, analisisian eragina dutela suposa daiteke. Urre-patroiko ezaugarriekin modura, ezaugarri automatikoak erabiltzean *Stacklazy*_{MST} pilaketarekin lortzen da emaitzarik onena, 1,72 puntuko gehikuntzarekin, eta, txarrena, berriz, *Stacklazy*_{Stacklazy} pilaketarekin, 0,39 puntuko gehikuntzarekin.

4.3. Bozketa bidezko konbinazioa

Aurreko ataletan analizatzaile ezberdinak lortu dira: oinarritzko algoritmoak erabiltzen dituztenak eta zuhaitz-transformazioak eta pilaketa-teknika hedatuak erabiltzen dituztenak. Analizatzaileen aniztasunaren eragina aztertzeko, analizatzaile hauek aukeratu dira:

- Oinarritzko analizatzaileak. Multzo honetan sartu dira MaltParser sistemako Nivre, Stacklazy eta Covington algoritmoak eta MSTko algoritmo proiektibo eta ez-proiektiboak.
- Analizatzaile hedatuak. Oinarritzko algoritmoen zuhaitz-transformazioak eta pilaketa-teknikak gehitu zaizkie.

Alde batetik, ausazko onenen konbinazioa probatzeko, lehenengo eta behin, entrenamenduko datuekin indar basatia erabili da, hau da, n analizatzailearen irteera-multzo posible guztiak konbinatu dira. Azkenik, test-multzoari aplikatu zaio lortutako konbinazio onena. Beste aldetik, x onenen konbinazio-sistema erabili da. Aurreko bi teknika horiek analizatzaile-multzo ezberdinekin egin dira.

4.3.1. Oinarritzko algoritmoen konbinazioa

Lehenengo esperimentuan, oinarritzko algoritmoak erabili dira soilik. Horretarako, 12 analizatzaile ezberdinen multzoa osatu da, MaltParser eta MSTParser-ekin erabilgarri dauden oinarritzko algoritmoekin osatuta: MSTParser sistemako bi aldaera (lehenengo eta bigarren mailako algoritmo ez-proiektiboarekin) MaltParserreko 10 aldaera ezberdin:

- Ezkerretik eskuinera Stacklazy, Covington, Nivre Arku Estandarra eta Nivre Arku Estandarra transformazio pseudoproiektiboarekin LibLINEAR ikasketa automatikoarekin.
- Ezkerretik eskuinera Stacklazy SVM ikasketa automatikoarekin.
- Aurreko guztiak eskuinetik ezkerreara.

Urre-patroiko ezaugarriekin eta ezaugarri automatikoekin x onenen konbinazioa erabiltzerakoan, oinarritzko 12 analizatzaileen irteerak multzo posible guztiak (3, ..., 12) konbinatu ostean lortutako emaitzak agertzen dira 6. taulan.

6. taula. Urre-patroiko ezaugarriekin eta ezaugarri automatikoekin lorturiko ausazko onena eta x onenen konbinazio onenak, oinarritzko sistema onenarekin alderatuta.

Kop. (n)	Urre-patroiko ezaugarriekin		Ezaugarri automatikoekin	
	Ausazko onena	x onenen komb.	Ausazko onena	x onenen komb.
Oin. onena	80,17	80,17	78,35	78,35
3	81,97 (+1,80)	81,58 (+1,41)	80,22 (+1,87)	78,89 (+0,54)
4	82,34 (+2,17)	81,54 (+1,37)	80,69 (+2,34)	79,01 (+0,66)
5	82,03 (+1,86)	81,05 (+0,88)	80,63 (+2,28)	80,47 (+2,12)
6	82,83 (+2,66)	82,33 (+2,16)	80,84 (+2,49)	80,62 (+2,27)
7	82,99 (+2,82)	82,69 (+2,52)	81,17 (+2,82)	80,99 (+2,64)
8	82,72 (+2,55)	82,87 (+2,70)	81,15 (+2,80)	80,90 (+2,55)
9	82,88 (+2,71)	82,68 (+2,51)	81,16 (+2,81)	80,91 (+2,56)
10	82,97 (+2,80)	82,75 (+2,58)	81,15 (+2,80)	80,98 (+2,63)
11	82,78 (+2,61)	82,70 (+2,53)	81,04 (+2,69)	81,01 (+2,66)
12	82,73 (+2,56)	82,73 (+2,56)	81,12 (+2,77)	81,12 (+2,77)

Goiko 6. taulatik emaitza hauek nabarmendu daitezke:

- Urre-patroiko ezaugarri morfologikoak erabilita, ausazko onena metodoarekin 82,99 lortzen da LAS neurrian, hots, oinarritzko sistema onenarekin alderatuz gero (LAS = 80,17), ia 3 puntuko hobekuntza lortzen da.

- Bestetik, n balioa 3tik 12ra arte izanda, bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin egindako konbinazioetan, ausazko onena konbinazioarekin x onenen konbinazioarekin baino emaitza hobeak lortzen dira; salbuespen bakarra urre-patroiko ezaugarriekin eta sistemen kopurua 8 denean gertatzen da (ikus 6. taula).
- Halaber, 7 sistemarekin lortutako ausazko onena konbinazioan lortutako emaitza gainerako n sistemekin lortutakoaren pare gelditzen dela ikus daiteke, bai ezaugarri automatikoak, bai urre-patroiko ezaugarriak erabilia. Aldiz, x onenen konbinazio-metodoarekin, urre-patroiko ezaugarriekin 8 sistema eta ezaugarri automatikoekin 12 sistema konbinatu behar dira konbinazio onenaren parera heltzeko (ikus 6. taula).
- Ezaugarri automatikoak erabilia, ausazko onena metodoarekin 81,17 lortzen da LAS neurrian, eta oinarritzko sistema onenarekin alderatuz gero (LAS = 78,35), 2,82 puntuko hobekuntza lortzen da. Ezaugarri automatikoekin zehaztasunik ez galtzea oso emaitza interresgarria da, aplikazio zuzena baitu.
- Azkenik, x onenen konbinazioan eta ausazko onena konbinazioan, bai urre-patroiko ezaugarriekin bai ezaugarri automatikoekin lortutako hobekuntzak 2,75 ingurukoak izan dira, nahiz eta ausazko onena metodoarekin zerbait hobetu.

4.3.2. *Analizatzaile hedatuen konbinazioa*

Bigarren esperimentuan, aurreko oinarritzko analizatzaileez gain, analizatzaile hedatuak (zuhaitz-transformazioak eta pilaketa) gehitu dira, teknika hedatuak konbinazioa egitean lagungarriak diren ala ez jakiteko, edo oinarritzko analizatzaileen konbinazioa egin ondoren nahikoa den jakiteko. Printzipioz, analizatzaile hedatuak gehitzeak aniztasuna gehitu beharko luke, baina gerta liteke, analizatzaile hedatuekin lortzen den hobekuntza desagertzea konbinatzean. Esperimentuak egiteko 26 analizatzaile ezberdin lortu dira (oinarritzkoak + hedatuak). Baina 26 sistemen azpimultzo posible guztiak kalkulatzeko mementuan, esperimentuan kopurua izugarri igotzen zenez, bigarren esperimendua egiteko, 14 sistema onenak erabiltzea erabaki da.

Sistema hedatuen irteerak konbinazioan sartzeak zenbaterainoko eragina du zehaztasunean? Galdera horri erantzuteko, 7. taulan urre-patroiko ezaugarri morfologikoekin eta ezaugarri automatikoekin x onenen konbinazioa edo ausazko onena metodoa erabiltzean, 14 sistemen (oinarritzko eta hedatuen) irteerak multzo posible guztietan (3, ..., 14) konbinatu ostean lortutako emaitzak agertzen dira. Horretarako, 14 sistema onenen aniztasuna aztertzean, urre-patroiko ezaugarriekin 5 zuhaitz-transformazio, 7 pilaketa eta 2 oinarritzko sistemen irteerak bildu dira, eta, ezaugarri automatikoekin, 5 zuhaitz-transformazio, 6 pilaketa eta 3 oinarritzko sistemen irteerak.

7. taula. 14 sistema onenak.

Kop. (n)	Urre-patroiko ezaugarriekin		Ezaugarri automatikoekin	
	Ausazko onena	x onenen konb.	Ausazko onena	x onenen konb.
Oin. onena	80,83	80,83	78,26	78,26
3	82,55 (+1,72)	82,11 (+1,28)	80,96 (+2,70)	80,53 (+2,27)
4	83,22 (+2,39)	82,55 (+1,72)	80,92 (+2,66)	80,90 (+2,64)
5	83,53 (+2,70)	83,43 (+2,60)	81,59 (+3,33)	81,87 (+3,61)
6	83,20 (+2,37)	83,46 (+2,63)	81,91 (+3,65)	81,66 (+3,40)
7	83,62 (+2,79)	83,74 (+2,91)	82,22 (+3,96)	81,86 (+3,60)
8	83,70 (+2,87)	83,76 (+2,93)	82,08 (+3,82)	82,11 (+3,85)
9	83,77 (+2,94)	83,77 (+2,94)	82,40 (+4,14)	82,31 (+4,05)
10	83,69 (+2,86)	84,02 (+3,19)	82,32 (+4,06)	82,11 (+3,85)
11	83,90 (+3,07)	83,95 (+3,12)	82,24 (+3,98)	82,17 (+3,91)
12	83,72 (+2,89)	83,83 (+3,00)	81,97 (+3,71)	82,03 (+3,77)
13	83,63 (+2,80)	83,63 (+2,80)	82,18 (+3,92)	81,81 (+3,55)
14	83,64 (+2,81)	83,64 (+2,81)	82,04 (+3,78)	82,04 (+3,78)

Taulan ikus daitekeenez, sistema hedatuaren irteerak konbinazioan sartzeak emaitzak hobetzen ditu. Adibidez, sistema hedatuak gehituz egindako konbinazioetan, urre-patroiko ezaugarriak eta ezaugarri automatikoekin eta ausazko onenaren konbinazioarekin +3,07 eta +4,14ko irabaziak lortu dira; oinarritzko sistemak konbinatu direnean, aldiz, +2,82ko irabaziak baino ez dira lortu.

5. ONDORIOAK

Lan honen helburu nagusia da euskararako estaldura zabala izango duen dependentzietan oinarritutako analizatzaile sintaktiko-estatistiko sendo bat lortzea, eta, xede horretarako, transformazio, pilaketa eta bozketateknikak erabili dira. Atal honetan, aurreko atalean lortutako emaitzen laburpen-taula (8. taula) eta lortutako emaitzetatik atera diren ondorio nagusiak azaldu dira:

- **Transformazioen konbinazioa eta tamaina.** Sintagmen transformazioa, mendeko perpausen transformazioa, koordinazio-transformazioa eta transformazio pseudoproiektiboa bata bestearen atzetik aplikatu behar dira. Transformazioen ordenak garrantzia du konbinazioan; izan ere, transformazio pseudoproiektiboa sintagmen transformazioa eta mendeko perpausen transformazioaren ostean joan behar da, sintagmen eta mendeko perpausen transformazioek hain-

8. taula. Teknika ezberdinekin lortutako emaitzen laburpen-taula.

Parsers	Urre-patroiko ezaugarriekin	Ezaugarri automatikoekin
Oinarrizko onena	80,17	77,88
Zuhaitz-transformazioak	80,87	79,09
Pilaketa	80,77	78,55
Bozketa (oinarriekin)	82,99	81,17
Bozketa (oinarri + hedatuekin)	84,02	82,40

bat arku ez-proiektibo sortzen baitituzte. EZB zuhaitz-bankuei egin-dako zuhaitz-transformazioekin hobekuntza esanguratsuak lortu dira, bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin. Algoritmo gehienetan, transformazioen konbinazioaren eragina zuhaitz-bankua bikoiztearen heinekoa dela esan daiteke. Ondorioz, teknika horiek erabilgarriak direla egiaztatu da.

- **Pilaketa-teknika euskararako analizatzaile sintaktikoa hobetzeko baliagarria ote da? Zenbateraino?** Oro har, zuhaitz-banku osoari erreparatuz gero, pilaketak emaitza esanguratsuak eman ditu, baina zuhaitz-transformazioan lortutakoak baino apalagoak dira. Salbuespen bakarra $Stacklazy_{M,ST}$ pilaketa izan da, kasu horretan zuhaitz-transformazioan lortutakoak baino hobeak lortu baitira. Bestalde, urre-patroiko ezaugarriekin lortutako hobekuntzak ezaugarri automatikoekin ere gertatzen direla probatu da. Ondorioz, pilaketa ezaugarri automatikoekin ere erabil daitekeela probatu da.
- **Bozketan, analizatzaileen LAS puntuazioaren araberako ordenak edo aniztasunak (osagarritasunak) garrantzi handiagoa ote du?** Ikerketan zehar, x onenen konbinazioa eta ausazko onenen konbinazioa probatu dira. Oinarrizko algoritmoekin 12 aldaera ezberdinetako analizatzaileen multzoa osatu da eta irteera-multzo posible guztiak (3, ... 12) konbinatu ostean, ondorio hauek atera ditugu: ausazko onena metodoarekin x onenen konbinazioarekin baino emaitza hobeak lortu dira; ausazko onena metodoarekin, 6 analizatzailearen irteerekin nahikoa da; aldiz, x onenen konbinazioarekin, 10 analizatzailearen irteerak behar izaten dira; eta, azkenik, ausazko onena metodoarekin lortutako konbinazio onena osatzen duten analizatzaileen multzoan, argitu da inoiz ez direla x onenen konbinazio-multzoa agertzen. Horrela, bozketa-sistemetan LAS neurria soilik kontuan hartu beharra baztertuko da, sistema konbinatu onenen artean beti aurkitzen baitugu LAS sailkapenaren bukaeran daudenak; horrez gain, beste faktore garrantzitsu batzuk aipa daitezke: bateragarritasuna, aldaera eta kalitatea ([12]).

- **Bozketan, oinarritzko sistemekin.** Kasu honetan, 12 oinarritzko sistemen irteerak ausazko onena konbinazioarekin, urre-patroiko ezaugarriekin eta ezaugarri automatikoekin 82,99 eta 81,17 lortu dira LAS neurrian, hurrenez hurren; eta bakarkako oinarritzko sistema onenarekin alderatuz gero, ia 3 puntuko hobekuntza lortu da.
- **Bozketan, sistema hedatuekin.** Esperimentuak egiteko 26 analizatzaile ezberdin lortu dira (oinarritzkoak + hedatuak), baina 26 sistemen azpimultzo posible guztiak kalkulatzeko mementuan esperimentuen kopurua izugarri igotzen zenez, 14 sistema onenak erabiltzea erabaki da. Horrela eginda, 14 sistema onenen artean aniztasuna mantentzen dela probatu da, urre-patroiko ezaugarriekin 5 zuhaitz-transformazio, 7 pilaketa eta oinarritzko 2 sistemaren irteerak bildu dira, eta, ezaugarri automatikoekin, 5 zuhaitz-transformazio, 6 pilaketa eta oinarritzko 3 sistemaren irteerak bildu dira. Bestalde, 7. taulan ezaugarri automatikoekin ausazko onena metodoa erabiltzean, 14 sistemen (oinarritzko eta hedatuak) irteerak multzo posible guztietan (3, ..., 14) konbinatu ostean, emaitzarik onena 9 sistemaren konbinaziotik (4 zuhaitz-transformazio, 2 pilaketa eta oinarritzko 3 sistemaren irteeren multzotik) lortu da; hau da, azken buruan, aniztasuna mantentzen dela esan daiteke. Horrez gain, urre-patroiko etiketak erabili beharrean, etiketa automatikoak erabiliz gero, hobekuntza esanguratsuak lortu dira. Emaitza bat aipatzearren, ausazko onena metodoarekin 82,40 balioa (+4,14 oinarri onenarekiko) lortu da.

6. ERREFERENTZIAK

- [1] NIVRE, J., HALL, J. eta NILSSON, J. 2006. «Maltparser: A data-driven parser-generator for dependency parsing». *Proceedings of LREC*, 6, 2216-2219.
- [2] MCDONALD, R. eta PEREIRA, F. 2006. «Online Learning of Approximate Dependency Parsing Algorithms». *EACL*.
- [3] ADURIZ, I., ARANZABE, M.J., ARRIOLA, J.M., ATUTXA, A., DE ILARRAMENDI, A., EZEIZA, N., GOJENOLA, K., ORONOZ, M., SOROA, A. eta URIZAR, R. 2006. «Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for automatic processing». *Language and Computers*, 56, 1, 1-15.
- [4] ALDEZABAL, I., ARANZABE, M.J., ARRIOLA, J.M. eta DE ILARRAMENDI, A. 2009. «Syntactic annotation in the Reference Corpus for the Processing of Basque (EPEC): Theoretical and practical issues». *Corpus Linguistics and Linguistic Theory*, 5, 2, 241-269.
- [5] ADURIZ, I. 2000. «EUSMG: morfologiatik syntaxira murriztapen gramatika erabiliz». *Doktoretza-tesia*.

- [6] EZEIZA, N., ALEGRIA, I. eta ARRIOLA, J.M. 1998. «Combining stochastic and rule-based methods for disambiguation in agglutinative languages». *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* 380-384.
- [7] NIVRE, J. eta NILSSON, J. 2005. «Pseudo-projective dependency parsing». *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 99-106.
- [8] MARTINS, A., DAS, D., NOAH, A. and XING, E. 2008. «Stacking dependency parsers». *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 157-166.
- [9] NIVRE, J. eta MCDONALD, R. 2008. «Integrating graph-based and transition-based dependency parsers». *Proceedings of ACL*.
- [10] SAGAE, K. eta LAVIE, A. 2006. «Parser combination by reparsing». *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*.
- [11] HALL, J., NILSSON, J., NIVRE, J., ERYIGIT, G., MEGYESI, B., NILSSON, M. eta SAERS, M. 2007. «Single Malt or Blended? A Study in Multilingual Parser Optimization». *Proceedings of the CoNLL Shared Task EMNLP-CoNLL*.
- [12] SURDEANU, M. eta MANNING, C. 2010. «Ensemble Models for Dependency Parsing: Cheap and Good?». *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference (NAACL-2010)*.
- [13] NIVRE, J., HALL, J., NILSSON, J., CHANEV, A., ERYIGIT, G., KÜBLER, S., MARINOV, S. eta MARSÌ, E. 2007. «Maltparser: A language-independent system for data-driven dependency parsing». *Natural Language Engineering*.
- [14] MCDONALD, R. eta NIVRE, J. 2011. «Analyzing and Integrating Dependency Parsers». *Computational Linguistics*.