EUSKAL HERRIKO UNIBERTSITATEA
Lengoaia eta Sistema Informatikoak Saila

Doktorego-tesia

Estaldura zabaleko euskararako analizatzaile sintaktiko estatistikoa

Kepa Xabier Bengoetxea Kortazar

2014

EUSKAL HERRIKO UNIBERTSITATEA
Lengoaia eta Sistema Informatikoak Saila

Estaldura zabaleko euskararako analizatzaile sintaktiko estatistikoa

Kepa Xabier Bengoetxeak Koldo Gojenolaren zuzendaritzapean egindako tesiaren txostena, Euskal Herriko Unibertsitatean Informatikan Doktore titulua eskuratzeko aurkeztua.

Bilbo, 2014ko azaroa.

Eskerrak

Tesi-lan hau bukatutzat jotzeko ordua iritsi da, atzera begira jarri eta errepaso azkar bat eginez, azken urte hauetan modu batera edo bestera lagundu didazuenei eskerrak emateko garaia iritsi zait:

- IXA taldeko diren eta izan diren lagun guztiei, tesi hau neurri handi batean zuena ere badelako.
- Zuzendariari, Koldori, bide luze honetan arlo guztietan gidari aparta izan zarelako.
- Tesi-txostenaren idatzitakoaren eskuzko zuzenketa fina egin duten guztiei, bereziki, Maxux, Koldo eta Iñigori.
- Artikuluetan lagun izandako guztiei: Koldo, Eneko, Arantza, Itziar, Joakim, Yue, Nerea, Iakes eta Maxux, langile apartak izan zaretelako.
- Maxuxi, nire duda linguistiko guztiak pazientzia handiz argitu dituzulako.
- Kikeri eta Nereari, esperimentuak egiteko makinak beti prest izan ditudalako.
- Latex-en tripak erakutsi dizkidazuen horiei (Maite, Aitziber eta Alicia)
- Etxekoei, bereziki nire emazte Estiri, nire alabei (Ainhize eta Itxaso), eta nire aita eta amari, dagokizuen denbora kendu dizuedalako.
- Zure izena hemen ikustea espero zenuen horri. Berdin eskertzen dizut egindakoa, baina une honetan buruak ez dit gehiagorako ematen!!!!

Zuei denei, berriz ere, eskerrik asko!

Gaien aurkibidea

1 Proiektuaren nondik norakoak

1.1 Sarrera

Egun, informazio gehiena ordenagailuetan gorde eta prozesatzen da. Ordenagailuan dagoen informazioa era egokian prozesatzeko, gizakiaren hizkuntza ulertzea beharrezkoa da. 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 araberakoa izan behar duela, hau da, sintaxiaren araberakoa. Imajinatu, adibidez, "urdina eta edo delako" esaten dela. Esaldi honek lau hitz ezagun ditu, baina horrela lotuta, hitzak ez dute inongo zentzurik. Esaldiak lehen baldintza hori betetzen duen jakiteko, analizatzaile sintaktikoa erabil daiteke. Baina hori ez da nahikoa. Izan ere, "karratua zirkulua da" esaldia aztertuz gero, zuzen josia dagoela, baina zentzugabea dela antzeman daiteke. Horrela, esaldiaren zentzua jasotzeko, semantika erabil daiteke. Hizkuntza ulertu ahal izateko, analisi sintaktikoaren ataza, testuko esaldi bakoitzari egitura sintaktiko bat esleitzea, izan da historikoki gehien landu den ataza.

Baina, analizatzaile sintaktikoen erabilera arlo desberdinetara zabaltzen ari da: analisi semantikoa lortzeko (Gildea eta Palmer, 2002), gramatika-zuzentzaile modura lan egiteko edo gaizki erabilitako egiturak detektatzeko (Oronoz, 2008), bai eta, besteak beste, hizkuntza modelizazioan (Chel-

ba *et al.*, 1997), informazioaren berreskurapenean (Culotta eta Sorensen, 2004), itzulpen automatikoan (Ding eta Palmer, 2004), galdera-erantzunetan (Wang *et al.*, 2007), itzulpen automatikorako baliabideak sortzean (Tiedemann, 2012) edota parafrasi eskuratze automatikoan (Shinyama *et al.*, 2002) ere.

Tesi-lan honen helburua estaldura zabala izango duen analizatzaile sintaktikoa garatzea izango da, bereziki euskararentzat, baina ingelesa ere erabili da, kanpoko baliabideen ekarpena probatzeko. Horretarako, 1.2 atalean, tesi-lan hau IXA taldeko lanen artean kokatuko da. 1.3 atalean, helburuak eta helburuak lortzeko ezarri diren zeregin zehatzak azalduko dira. Kapitulu honetako 1.4 atalean, tesiaren eskemak eta egindako argitalpenek alor horiek aberas ditzaketela erakutsiko da.

1.2 Lanaren kokapena

Tesi-lan hau hizkuntzaren prozesamenduan kokatzen da, Euskal Herriko Unibertsitateko Informatika Fakultateko IXA ikerketa-taldearen jardunean. IXA taldeak hogei urte baino gehiago darama hizkuntzaren tratamendu automatikoa egiten, eta urte horietan, batez ere, euskara landu da. Denbora tarte horretan, hizkuntzalarien eta informatikarien elkarlanari esker, euskararako sortutako baliabideak eta tresnak ugariak izan dira, esaterako, euskararen Datu-Base Lexikala (EDBL) (Aldezabal *et al.*, 2001), MORFEUS analizatzaile morfologikoa (Aduriz *et al.*, 1998) eta euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe, 2008). Azken urteotan, IXA ikerketa-taldean lan handia egin da sintaxian, eta sintaxi osoaren bide horretan, besteak beste, honako lanak egin dira: murriztapen-gramatika (ingeleseko *Constraint Grammar*) formalismoa erabiliz sintaxi partziala -esaldia osorik ulertzera iritsi gabe- landu da (Aduriz *et al.*, 2000; Arriola, 2000), euskararen sintaxia lantzeko oinarrizko baliabideen garapena (Gojenola, 2000), aditzaren azpi-kategorizazioaren azterketa (Aldezabal, 2004), eta azkenik, dependentzia-gramatiken formalismoa jarraituz garatutako analizatzaile sintaktiko osoa (Aranzabe, 2008). Horrela, hasierako helburu hori - estaldura zabala izango duen euskararako analizatzaile sintaktiko estatistikoa - IXA taldean analisi sintaktikoan landutako beste bide bat besterik ez da. Tesi-lan hau 2006an hasi zen, hain zuzen ere, dependentzietan oinarritutako analizatzaile sintaktiko estatistikoak berpiztu ziren garaian. 2006 eta 2007ko *CoNLL (Computational Natural Language Learning) Shared Task* konferen-

tziak hizkuntza askotarako zuhaitz-bankuak eta dependentzietan oinarritutako analizatzaileak garatzeko antolatu ziren (Buchholz eta Marsi, 2006; Hall *et al.*, 2007). 2006 eta 2007ko *CoNLL Shared Task* konferentzien ostean, ikasketa automatikoan oinarritutako hurbilpen bi nagusitu ziren: grafoetan oinarritutako modeloak (Eisner, 1996, 2000; McDonald eta Pereira, 2006) eta trantsizioetan oinarritutako modeloak (Nivre, 2003; Yamada eta Matsumoto, 2003; Titov eta Henderson, 2007). 2006ko *CoNLL Multilingual Dependency Parsing* lehiaketan[1], 13 hizkuntza eta 18 sistema lehiatu ziren. MSTParser (McDonald *et al.*, 2005; McDonald eta Pereira, 2006) grafoetan oinarritutako analizatzaile sintaktiko estatistikoa lehenengo postuan geratu zen. 2007ko *CoNLL*eko lehiaketan, euskarako zuhaitz-bankuak parte hartzeko esfortzu handia egin genuen euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe, 2008) *CoNLL-X* formatura bihurtzen. Horrela sortu zen, *CoNLL-X* formatuan, 55.469 hitzeko eta 3.700 esaldiko euskarako zuhaitz-bankua (hemendik aurrera, EZB I zuhaitz-bankua deituko zaio). 2007ko *CoNLL Multilingual Dependency Parsing* lehiaketan[2], beste 18 hizkuntzarekin batera eta 20 sistemaren artean ebaluatu zuten. MaltParser (Nivre *et al.*, 2006a) sistemak lortu zuen euskara hizkuntzarekin emaitzarik onena.

Tesi-lan honen helburu nagusia da euskararako estaldura zabala izango duen dependentzietan oinarritutako analizatzaile sintaktiko estatistiko sendo bat lortzea, eta bide horretan, grafoetan (Eisner, 1996; Eisner, 2000; McDonald eta Pereira, 2006) eta trantsizioetan (Nivre, 2003; Yamada eta Matsumoto, 2003; Titov eta Henderson, 2007) artearen egoera diren MSTParser eta MaltParser sistemak euskarara egokitzea erabaki genuen.

1.3 Helburuak

Tesi-lan honen helburu nagusia da euskararako estaldura zabala izango duen dependentzietan oinarritutako analizatzaile sintaktiko estatistiko sendo bat lortzea, eta, xede horretarako, hurrengo eginkizunak egin dira:

1. Artearen egoera diren analizatzaile sintaktiko estatistiko sortzaileen jarduteko modua ikasi eta egokienak aukeratzea.

2. Aukeratutako analizatzaile sintaktiko estatistiko sortzaileak euskarara

[1] http://ilk.uvt.nl/conll/
[2] http://www.cs.jhu.edu/EMNLP-CoNLL-2007

egokitzea. Egokitzeko behar diren oinarrizko elementuak (beltzez edo 1 zenbakiarekin adierazita) ikus daitezke 1.1 irudian:

- Zuhaitz-bankua: hizkuntzalari talde batek eskuz sintaktikoki etiketatutako corpusa da.
- Analizatzaile sintaktiko estatistiko sortzailea: zuhaitz-bankuan dauden erlazio sintaktiko zuzenetatik entrenatu eta ikasi, eta analizatzaile modelo bat sortzen duen sistema da.
- Analizatzaile modeloa: ikasitako analizatzaile modeloa esaldi berriei erlazio sintaktikoa esleitzeko gai izango da.
- Ezaugarrien ingeniaritza: analizatzaile sintaktiko estatistiko sortzailearen konfigurazioan, modelo bat ikasteko zein ezaugarri-modelo edo informazio erabili behar duen definitu behar da. Euskara morfologia aldetik aberatsa izanik, ezaugarri-modelo bat definitzerakoan zuhaitz-bankuan dauden ezaugarri morfosintaktikoen hautaketa ez da lan erraza.

3. Lehenengo probak egitea zuhaitz-bankuan dauden urre-patroiko ezaugarri morfosintaktikoekin (hizkuntzalari talde batek gainbegiratukoekin). Zuhaitz-bankuko esaldiak analisi morfologiko eta desanbiguatze moduluetatik (Ezeiza *et al.*, 1998) pasa ostean lortutako ezaugarri morfosintaktiko automatikoekin analizatzaile sintaktikoa benetako egoera batean probatuko da. Esan beharra dago, euskaraz, hitz-forma batek, batez beste, 2,65 interpretazio ezberdin izan ditzakeela, eta, horregatik, bada, ezaugarri morfosintaktikoek izan dezaketen eragina aztertzeko eta desanbiguatze morfologikoaren kalitatea neurtzeko balioko dute.

4. Sistemak egokitu ostean, eta sistema hauen zehaztasuna hobetze aldera, beste hizkuntzekin arrakastatsuak izan diren hainbat teknika euskarara moldatzea. Moldatutako teknikak dira:
   - Zuhaitz-transformazio teknikak (ikusi 1.1 irudian 2 zenbakia edo berdez marraztutako elementuak): nahiz eta buru-osagarri eta buru-modifikatzaile egitura gehienek analisi berdintsua izan dependentzia-gramatikan, badaude eztabaidagarriak diren egitura asko, besteak beste: aditz-laguntzailea aditz nagusiaren gobernatzailea izatea edo ez; determinatzaile-sintagman, determinatzailea burua izatea edo ez; postposizio-sintagman, azken hitza burua izatea edo ez; koordinazioetan, juntagailu edo koordinazioaren lehenengo edo azken osagaia buru izatea edo ez. Erabakitzeko unean

teoria ezberdinak aurki daitezke. Etiketatze-teoria desberdinen eragina aztertzeko, zuhaitz-bankuari aplikatutako aldez aurretiko eta ondorengo prozesaketa ezberdinak azalduko dira: proiektibizazio transformazioa, sintagmen transformazioa, mendeko perpausen transformazioa eta koordinazioaren transformazioa.

- Pilaketa edo *stacking* teknika (ikusi 1.1 irudian 3 zenbakia edo granatez marraztutako elementuak) ikasketa denboran, analizatzaile bi bateratzeko, lehenengo analizatzailearen irteeran lortutako egitura ezaugarriak bigarren analizatzailearen sarrera aberasteko erabiliko dira (Martins *et al.*, 2008; Nivre eta McDonald, 2008; McDonald eta Nivre, 2011). Euskara buru-azkeneko hurrenkera duen hizkuntza izanik, lehenengo analizatzailearen irteera ematen duten ezaugarri morfosintaktikoek (numeroa, kasua eta mendeko perpausa bezalakoak) bigarren analizatzailea aberas dezaten, printzipio linguistikoak hartuko dira oinarritzat (Bengoetxea eta Gojenola, 2009a, 2010).
- Bozketa bidezko konbinaketa teknika (ikusi 1.1 irudian 3 zenbakia edo granatez marraztutako elementuak): analizatzaile modelo desberdinen irteerak kontuan hartuko dira, irteera bateratu eta egoki bat lortzeko asmoarekin. 2007ko *CoNLL Multilingual Dependency Parsing* lehiaketan, Hall *et al.*-ek (2007) trantsizioetan oinarritutako 6 modeloren irteerak konbinatuz lehenengo postua lortu zuten 19 hizkuntzarekin eta 20 sistemarekin lehiatu ostean. Ildo beretik, aztergai dauden esperimentuak egite aldera, eta dependentzietan oinarritutako analizatzaileen irteerak bateratzeko, bozketaren bidezko konbinaketa erabili da. Sortutako oinarrizko sistemen (sistemen egokitzapena gauzatu ostean) eta sistema hedatuen (pilaketa eta zuhaitz-transformazio teknika osagarriak gauzatu ostean) irteerak konbinatu dira aniztasun faktoreak analisian izan dezakeen eragina probatzeko.

5. Informazio semantikoa aberastea (ikusi 1.1 irudian 4 zenbakia edo urdinez marraztutako elementuak): aspalditik erabiltzen da informazio semantikoa hizkuntzaren prozesamenduan egitura sintaktikoak desanbiguatzeko (hitzen adierek desanbiguatzeko, eta bide batez, analizatzaile sintaktikoaren lana hobetzeko). Hizkuntzaren prozesamenduan semantika jorratu ahal izateko ezinbestekoa da ezagutza-base lexiko-semantikoak (EBLS) garatzea. EBLSak hitzei eta adierei buruzko in-

formazioa duten baliabide lexikal egituratuak dira. IXA taldean, euskararako EBLSa garatzen den bitartean, ingeleserako garatuta dagoen EBLSa (*WordNet*a) erabili da. Horrela informazio semantikoak analisi sintaktikoan izan dezakeen eragina aztertzeko, *WordNet*eko klase semantikoak eta *corpus*etik ateratako hitz-multzoak probatu dira.

1.4 Tesi-lanaren eskema eta argitalpenak

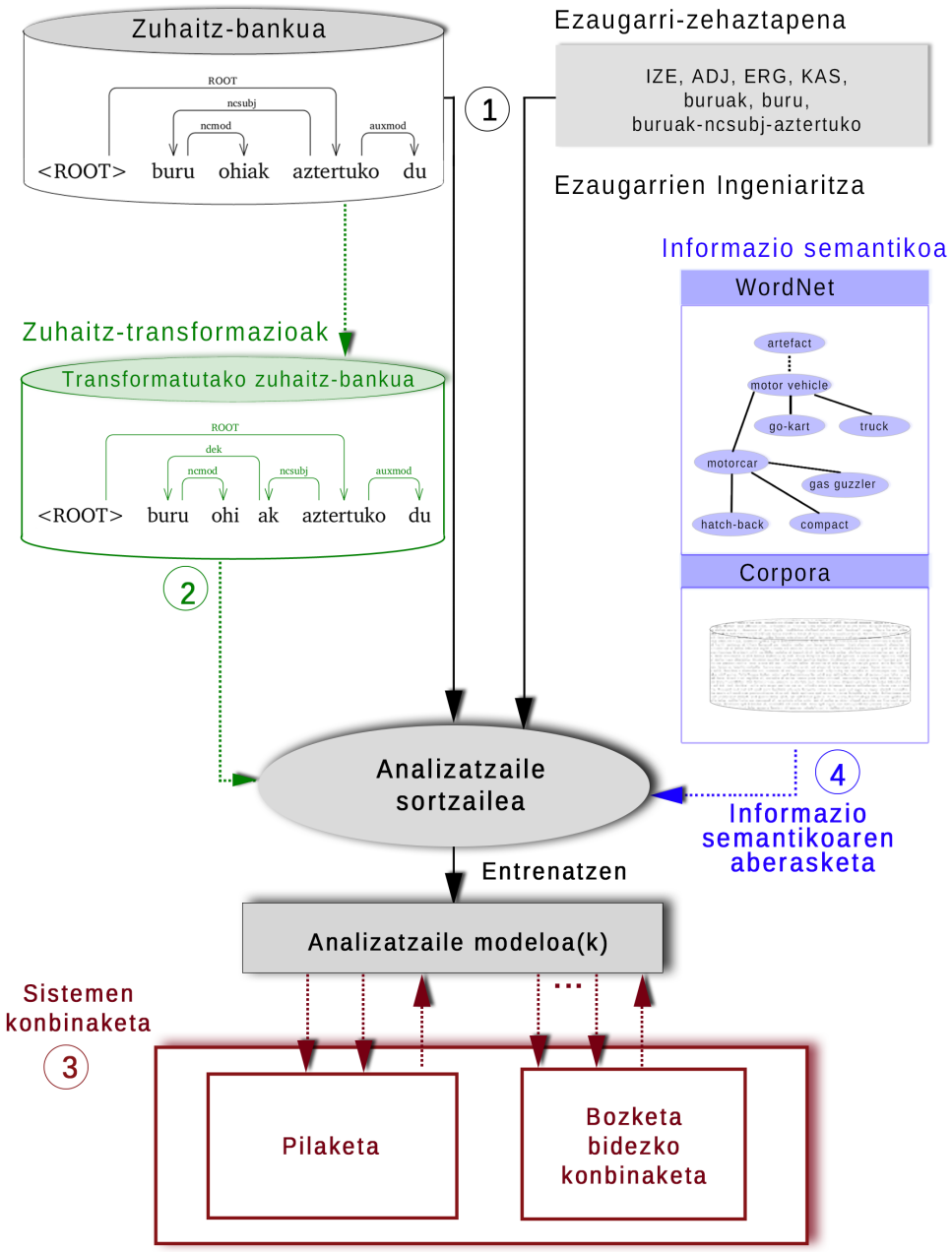
Tesi-lana lau ataletan banatu da, eta jarraian datozen 6 kapituluak 4 atal horietan bildu dira. 1.1 taulan agertzen da tesi-lanaren eskema (taularen ezkerraldean atalak eta eskuinaldean kapituluak zehaztu dira). Multzokatze honen nondik norakoak eta horietan agertzen diren kapituluetako bakoitzean landu diren gaiak labur-labur deskribatuko dira ondorengo lerroetan.

Atalak	**Kapituluak**
Motibazioa eta sarrera	1. Proiektuaren nondik norakoak
Datuetan oinarritutako analisia eta teknikak	2. Sintaxi konputazionala
	3. Euskararako analizatzaile sintaktikoa
	4. Zuhaitz-transformazioak, pilaketa eta bozketa analisi sintaktikoan
Semantika analisi sintaktikoan	5. Informazio semantikoa eta analisi sintaktikoa
Ondorioak eta zabaldutako ikerlerroak	6. Ondorioak eta etorkizuneko lanak

1.1 taula – Tesi-lanaren antolaketa.

Lehen atala esku artean dugun kapituluak osatzen du. Tesian egindako lanaren helburuak zehazten dira, lana IXA taldean egiten denaren barruan kokatzen da, eta tesi-lanean topatuko den informazioaren berri ematen da.

Bigarren atalean, "Datuetan oinarritutako analisia eta teknikak" izeneko atalean, hiru kapitulu bildu dira. Bigarren kapituluan, "Sintaxi konputazionala" deritzonean, sintaxi konputazionalean erabiltzen ari diren teknika eta baliabideak aipatuko dira. Baliabideen artean, baliatuko diren euskarako zuhaitz-bankuak eta ingeleseko zuhaitz-bankuak aztertuko dira, bai eta zuhaitz-bankuak etiketatzeko erabiltzen diren ereduak ere. Teknikan, berriz, analizatzaile sintaktikoen joera nagusiak aipatuko dira, baina bereziki teknika estatistikoetan oinarritutako sintaxian murgilduz. Hirugarren kapituluan, "Euskararako analizatzaile sintaktikoa" izenekoan, euskararako analizatzaile sintaktiko estatistikoa erabiltzeko eta informazio morfosintaktikoa lortzeko, gaur egun IXA taldean erabiltzen diren baliabideak edo moduluak aurkeztuko dira. Dependentzietan eta datuetan oinarritutako analizatzaile



1.1 irudia – Tesi-lana

sortzaile nagusiak (MSTParser eta MaltParser) euskara bezalako hurrenkera libreko hizkuntza eranskarira egokitzeko emandako urratsak azalduko dira. Zuhaitz-bankuan dauden ezaugarriak hizkuntzalari talde batek eskuz gainbegiratutako ezaugarriak dira, baina era automatikoan lortutako ezaugarriak erabiliz gero, benetako egoera batean analizatzaile sintaktikoak izan dezakeen zehaztasuna neurtuko da. Laugarren kapituluan, "Zuhaitz-transformazioak, pilaketa eta bozketa analisi sintaktikoan" izenekoan, teknika horien eragina analisian aztertuko da.

Hirugarren atala, "Semantika analisi sintaktikoan" izeneko atala, bosgarren kapituluarekin, "Informazio semantikoa eta analisi sintaktikoa" izenekoarekin osatuta dago. Kapitulu horretan, *WordNet*eko klase semantikoek eta corpuseko hitz-multzoek analisi sintaktikoan izan dezaketen eragina aztertuko da.

Tesi-lanari amaiera emateko, ateratako zenbait ondorio eta zabaldu diren ikerkuntzarako zenbait bideren berri emango da seigarren kapituluan.

Argitalpenak

Tesi-lan honek hainbat artikulu argitaratzeko bidea eman digu. Hona hemen argitalpen horien zerrenda, kapituluen arabera sailkatuta[3]:

- 3. kapitulua – Euskararako analizatzaile sintaktikoa:
  - Bengoetxea, K., eta Gojenola, K. **Desarrollo de un analizador sintáctico estadístico basado en dependencias para el euskera.** *Procesamiento del Lenguaje Natural*, 1(39), 5–12. ISSN idatzitako edizioan: 1135-5948 eta edizio digitalean: 1989-7553. Sevilla. 2007.

- 4. kapitulua – Zuhaitz-transformazioak, pilaketa eta bozketa analisi sintaktikoan:
  - Bengoetxea, K., eta Gojenola, K. **Exploring Treebank Transformations in Dependency Parsing.** *Recent Advances in Natural Language Processing, RANLP 2009.* ISSN 1313-8502. 33–38. Borovets. Bulgaria. 2009.

[3]Oharra: Argitalpen hauetan autoreen ordena alfabetikoa da, 5. kapituluari dagokionean izan ezik.

  - Bengoetxea, K., eta Gojenola, K. **Application of feature propagation to dependency parsing.** *In Proceedings of the 11th International Conference on Parsing Technologies (IWPT).* 142–145. Association for Computational Linguistics. Paris. 2009.
  - Bengoetxea, K., eta Gojenola, K. **Application of different techniques to dependency parsing of Basque.** *In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages.* 31-39. NAACL Workshop, Los Angeles. 2010.
  - Bengoetxea, K., Gojenola, K., eta Casillas, A. **Testing the effect of morphological disambiguation in dependency parsing of basque.** *International Conference on Parsing Technologies (IWPT). 2nd Workshop on Statistical Parsing Morphologically Rich Languages (SPMRL)* 28–33. Association for Computational Linguistics. 2011.

- 5. kapitulua – Informazio semantikoa eta dependentzia ereduko analisi sintaktikoa:

  - Agirre, E., Bengoetxea, K., Gojenola, K., eta Nivre, J. **Improving dependency parsing with semantic classes.** *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* short papers-Volume 2, 699–703. Association for Computational Linguistics. Short Paper. Portland, Oregon. 2011.
  - Bengoetxea K., Agirre E., Nivre J., Zhang Y. eta Gojenola K. **On WordNet Semantic Classes and Dependency Parsing** *Proceedings of the 52th Annual Meeting of the Association of Computational Linguistics, ACL 2014.* Short Paper. Baltimore. 2014.

- Beste hauek ere, kapitulu zehatz batekin lotura ez badute ere, tesiarekin zerikusia dute:

  - Aduriz I., Esparza, I., Bengoetxea, K., Gojenola, K. **Migración de una gramática sintáctica parcial entre dos formalismos de unificación**. *Procesamiento del lenguaje natural*, 83–90, zk. 37. ISSN idatzitako edizioan: 1135-5948 eta edizio digitalean: 1989-7553. Zaragoza. 2006.

– Aranzabe, M. J., De Ilarraza, A. D., Ezeiza, N., Bengoetxea, K., Goenaga, I., eta Gojenola, K. **Combining Rule-Based and Statistical Syntactic Analyzers** *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages (SP-Sem-MRL 2012).* 48–54, Association for Computational Linguistics (ACL). USA, ISBN: 978-1-937284-30-5, July 12, 2012, Jeju Island, Republic of Korea.

2 Sintaxi konputazionala

2.1 Sarrera

Lan honen helburua estaldura zabala izango duen analizatzaile sintaktikoa (ingelesez, *parser*) lortzea da, diren baliabideak oinarrian hartuta. Datuetan oinarritutako euskararako analizatzaile sintaktiko estatistiko bat izateko, hainbat baliabide behar dira: informazio morfologikoa lortzeko baliabide eta tresnak, zuhaitz-bankua, analizatzaile sintaktiko sortzaile bat, eta abar. Kapitulu honetan, sintaxi konputazionalean erabiltzen ari diren teknika eta baliabideak aipatuko dira. Lehenengo atalean, zuhaitz-bankuak eta hauek etiketatzeko erabiltzen diren ereduak aztertuko dira. Halaber, euskarako zuhaitz-bankua eta ingeleseko zuhaitz-bankuak aurkeztuko dira. Bigarren atalean, analizatzaile sintaktikoen joera nagusiak aipatuko dira, baina bereziki teknika estatistikoetan oinarritutako sintaxian oinarrituz.

2.2 Zuhaitz-bankuak

Zuhaitz-bankua sintaktikoki etiketatutako corpusa da. Corpuseko esaldi guztiak analizatu dira eta esaldiko hitz bakoitza etiketatu da esaldiaren zuhaitz sintaktikoa lortuz. Gaur egun, bi modutara etiketatutako zuhaitz-bankuak aurki ditzakegu: hizkuntzalari talde batek eskuz etiketatutako zuhaitz-bankuak eta teknika erdi-automatikoak erabiliz sortutako zuhaitz-bankuak, azken horiek analizatzaile batek etiketatutako esaldiak, hizkuntzalari batek gainbegiratu ostean lortzen direnak dira.

2.2.1 Esaldiak sintaktikoki adierazteko ereduak

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.

2.2.1.1 Osagai-egituraren eredua

Osagai-egituraren ereduan osagaia da oinarria. Osagaia unitate edo sintagma bakarra osatzen duen hitz multzoa da. Esaldia osagai edo sintagma txikiagoetan banatzen da, eta ohikoena da izen-sintagma, aditz-sintagma eta antzerako barne-egitura osagaietan banatzea. Ezagupen sintaktikoa osagaietan irudikatzerakoan, formalismo guztiak Chomsky-ren Testuingururik Gabeko Gramatika (TGG) ereduan oinarritzen dira (Chomsky 1956, 1965, 1981).

TGG erregela multzoa da. Erregelek hizkuntzaren sinboloak euren artean nola multzokatu eta ordenatzen diren esango digute. Adibidez 2.1 taulan dagoen *"a flight"* sintagmaren erregela sortan, 3. eta 4. lerroko erregelek sinbolo bakoitza bere kategoria gramatikalarekin lotzen dute, hau da, *a* determinatzailea izango da eta *flight* izena izango da. Erregeletan bi motatako sinboloak daude: terminalak (hizkuntzan erabiltzen diren hitzak) eta ez-terminalak (abstrakzio edo multzokatze-maila adierazteko erabiltzen direnak).

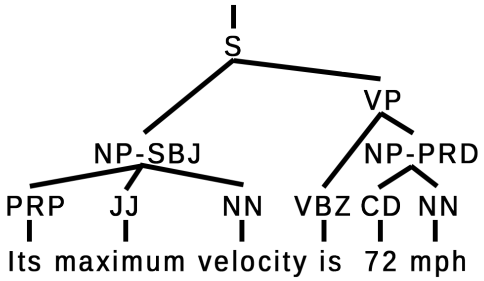
1.	$NP \rightarrow Det\ Nominal$
2.	$Nominal \rightarrow Noun \mid Noun\ Nominal$
3.	$Det \rightarrow a$
4.	$Noun \rightarrow flight$

2.1 taula – Ingeleseko izen-sintagma baten erregela sorta

TGGren deskribapen formala egiteko, G=(N, Σ, P, S) laukotea erabiliko da:

- N: sinbolo ez-terminalen multzoa
- Σ: sinbolo terminalen multzoa (N-rekiko disjuntua)
- P: $A \rightarrow \alpha$ bezalako produkzioen edo erregelen multzoa, non $A \epsilon N$ eta $\alpha \epsilon (\Sigma\ U\ N)^*$ multzoaren sinboloen katea den
- S: hasierako sinboloa

2.1 irudian, osagai-egituran oinarritutako esaldi baten zuhaitz sintaktikoa ikus genezake:



2.1 irudia – Ingeleseko *Penn Treebank*eko esaldi baten zuhaitz sintaktikoa osagai-egituran oinarrituta.

Osagai-egituraren analisian hiru motako adabegiak agertzen dira: erroa (gurasorik gabeko adabegia), hostoak (umerik gabeko adabegiak) eta barne-adabegiak (gurasoak eta umea(k) dituzten adabegiak). Adibidez, 2.1 irudian, erroa S[1] litzateke; barne-adabegiak, berriz: NP-SBJ, PRP, JJ, NN, VP, VBZ, NP-PRD eta CD lirateke; eta, azkenik, hostoak esaldiko hitzak lirateke: *its, maximum, velocity, is, 72 eta mph.* Adabegi horiek multzokatze sintaktikoak adierazten dituzte.

Chomsky-ren TGGko formalismo linguistikoez gain, badira sortutako beste formalismo ugari ere, besteak beste: *Lexical-Functional Grammar* (LFG) (Kaplan eta Bresnan, 1982; Bresnan, 2000), *Generalized-Phrase Structure Grammar* (GPSG) (Gazdar *et al.*, 1985), *Tree Adjoining Grammar* (TAG) (Joshi *et al.*, 1985), *Head-driven Phrase Structure Grammar* (HPSG) (Pollard, 1994) eta *Word Grammar* (WG) (Hudson, 1998). Aipatutakoez gain, badaude, esan bezala, beste asko ere.

2.2.1.2 Dependentzia-egituraren eredua

Aranzabe-ren (2008) tesi-lanean esaten den bezala, nahiz eta deskripzio linguistikoetan tradizio handia izan, dependentzia-gramatika nahiko ahaztua egon da, bai teoria linguistikoetan, bai hizkuntzaren tratamendu konputazionalean Mel'čuk-en (1988:3-4) hitzek jasotzen duten moduan:

[1]Ingeleseko laburdura hauen itzulpena edo esanahia, euskaraz: S: perpausa; NP-SBJ: izen-sintagma subjektua; PRP: izenordain posesiboa; JJ: adjektiboa; NN: izena, singularra; VP: aditz-sintagma; VBZ: aditza, orainaldian; NP-PRD: atributua; CD: zenbakia

> *"Phrase structure representation in syntax was strongly promoted by the Structuralist school during the thirties, forties and fifties (...). It became the only syntactic representation ever seriously discussed in the work of N. Chomsky and the Transformational-Generative School he founded in the late fifties. As a result of the triumphal offensive of the trasformational-generative approach throughout the world, phrase-structure syntax forced dependency syntax into relative obscurity"*

Tesnière-ren (1959) lanean oinarrituz gero, 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) da edo j i-ren mendekoa (umea terminoarekin ezagutzen dena) da. Dependentzia-egitura etiketatutako eta zuzendutako grafo bat bezala ikus daiteke, hau da, hitzez etiketatutako adabegiak eta euren arteko arku etiketatuak ala etiketatu gabeak. Dependentzia-etiketa multzo bat aukeratzerakoan ohikoena da funtzio-informazioa erabiltzea (adibidez: subjektua, objektua, eta abar).

Har dezagun, $E = w_1, ..., w_n$, n hitzak dituen esaldi bat eta R arkuak etiketatzeko erabiliko diren dependentzia-etiketen multzoa; E esaldiaren grafo zuzendu etiketadunaren definizio formala egiteko, $G = (V, A, L)$ hirukotea erabiliko da:

- $V = \{1, ..., n\}$, esaldiaren hitz-formen identifikadoreen multzoa
- $A \subseteq V$ x V, esaldiaren dependentzia-arkuen multzoa
- $L : A \rightarrow R$, esaldiaren dependentzia-arkuen etiketen multzoa

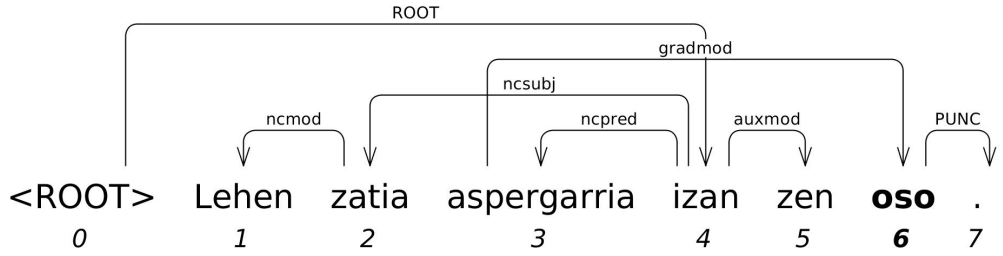
A multzoko arkuak hitz burutik hitz mendekora doaz (adibidez, $i \rightarrow j$ arkuak, gurasoa w_i eta umea w_j izango ditu) eta arkuek L dependentzia-egituraren etiketa izango dute.

Grafoek jarraian azalduko diren mugak bete behar dituzte:

- Grafoa aziklikoa eta konexua izango da.
- Ume adabegi batek guraso bakarra izan dezake, baina guraso adabegi batek ume ugari izan ditzake.

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 daudenean gertatzen da. Horrela, esaldi osoko arkuak plano batean gurutza-

tu barik marraztuz gero esaldi proiektiboa dela esango da, eta, elkar gurutzatuz gero, aldiz, esaldia ez-proiektiboa dela. Adibidez, 2.2 irudian jasotzen den "Lehen zatia aspergarria izan zen oso." esaldian, "oso" graduatzailea eta bere gobernatzailea den "aspergarria", "izan zen" aditzarekin banatuta datoz. Ondorioz, esaldi honetan arkuak gurutzatu egiten dira. Beraz, perpausa ez-proiektiboa da. Esaldi ez-proiektiboak maiz gertatzen dira euskara, txekiera, nederlandera eta turkiera bezalako hurrenkera libreko hizkuntzetan.



2.2 irudia – Euskarako zuhaitz-bankutik hartutako esaldi baten dependentzien irudikapena.

2.2 irudian, "ROOT" hitz artifiziala sartu da. Era honetan, esaldiko hitz guztiek gobernatzaile edo buru bat izango dute. Formalismo honetan grafo guztiek esaldi osoaren buru "ROOT" erro nodoa izango dute eta esaldiaren hitz guztiek guraso bakarra izango dute, zuhaitz bat osatuz.

Dependentzia-erlazioak ezartzeko eta erlazio horietan gobernatzailea eta mendekoa bereizteko irizpideek garrantzi handia izan dute dependentzia-gramatikan. Hona hemen Hudson-ek (1990) proposatzen dituen irizpideak, egitura batean gobernatzailearen eta mendekoaren arteko erlazio sintaktikoa ezagutzeko:

1. Gobernatzaileak egituraren kategoria sintaktikoa zehaztuko du eta egitura ordezka dezake.

2. Gobernatzaileak egituraren kategoria semantikoa zehaztuko du; mendekoak zehaztasun semantikoa emango du.

3. Gobernatzailea nahitaezkoa da; mendekoa ez da nahitaezkoa.

4. Gobernatzaileak mendekoa aukeratuko du eta mendekoa nahitaezkoa ala aukerakoa den zehaztuko du.

5. Mendekoaren forma gobernatzailearen araberakoa da.

6. Mendekoaren hurrenkera lineala gobernatzailearen arabera dago zehaztuta.

Ikus daitekeen bezala, zerrenda horretan irizpide desberdinak daude, batzuk sintaktikoak dira eta beste batzuk semantikoak. Halaber, badaude dependentzia-erlazio desberdinak bereizteko beharra azpimarratu duten beste egile batzuk ere. Hala, Mel'čuk-ek (1988) dio esaldiko hitzak hiru dependentzia moten bitartez lot daitezkeela: dependentzia morfologikoa, sintaktikoa eta semantikoa. Era berean, bada dependentzia sintaktikoen artean egitura endozentrikoak eta exozentrikoak bereizten dituenik ere (Nivre, 2005).

Egitura endozentrikoetan gobernatzaileak mendekoa ordezkatu ahal du, egitura sintaktikoan eraginik izan gabe. Adibidez, 2.2 irudian, "zatia" eta "lehen" hitzen arteko egitura endozentrikoa dela esan daiteke. Egitura exozentrikoek ez dute goian aipatutako lehenengo araua betetzen, ezin baitira bere gobernatzailearekin ordezkatu, hau da, "zatia" eta "izan" hitzen artean egitura exozentrikoa dago, "zatia" ezin daitekeelako ordezkatu.

Egitura endozentrikoen eta exozentrikoen arteko bereizketa buru-osagarri eta buru-modifikatzaile (edo buru-adjuntu) erlazioen bereizketarekin erlazionatu izan ohi da. Osagarrien eta modifikatzaileen arteko bereizketa hau balentzia terminoaz ere ezagutu izan da. Balentzia aditz bati dagozkion nahitaezko osagarrien multzoa da. Aditz desberdinek mota bateko osagarriak edo bestelakoak hartuko dituzte. Horietako osagarri batzuk nahitaezkoak dira aditzarentzat, subjektua, esaterako. Beste osagarri batzuk, aldiz, aukerakoak edo zirkunstantzialak dira, hala nola, lekua, denbora edo moduaren berri ematen duten adjuntuak.

Nahiz eta buru-osagarri eta buru-modifikatzaile egitura gehienek dependentzia-gramatikan analisi berdintsua izan, badaude eztabaidagarriak diren egitura asko, besteak beste: aditz-laguntzailea, aditz nagusien gobernatzailea izatea edo ez; determinatzaile-sintagman determinatzailea burua izatea edo ez; postposizio-sintagman azken hitzak burua izatea edo ez; koordinazioetan, juntagailu edo koordinazioaren lehenengo edo azken osagaiak buru izatea edo ez. Erabakitzeko unean teoria ezberdinak aurki ditzakegu.

Osagaietan oinarritutako ereduan gertatzen den bezala, dependentzietan oinarritutako esaldi-egitura adierazteko ere formalismo ugari daude. Horrela, Tesnière-k garatutako egitura sintaktikoaren teoriaz gain, dependentzia-gramatika teoria ezagunenen artean hurrengo hauek daude: *Word Grammar* (WG) (Hudson, 1998), *Functional Generative Description* (FGD) (Sgall *et al.*, 1986), *Lexicase* (Starosta, 1988), *Meaning-Text Theory* (MTT) (Mel'čuk,

1988) eta *Dependency Unification Grammar* (DUG) (Hellwig *et al.*, 2003). Horiek ez ezik, murriztapenetan oinarritutako dependentzia-gramatikek ere tradizio handia izan dute, adibidez, *Constraint Dependency Grammar* (CDG) (Maruyama, 1990), *Functional Dependency Grammar* (FDG) (Tapanainen eta Järvinen, 1997) eta *Topological Dependency Grammar* (TDG) (Duchier eta Debusmann, 2001). Dependentzia-gramatikei buruzko sintesia *Dependency Grammar Logic* (DGL) (Kruijff, 2001) lanean aurki daiteke.

CoNLL-X formatua

Urterik urte, *Computational Natural Language Learning (CoNLL)* konferentzian, sistema ezberdinek lehiatu egiten dute eginkizun partekatu batean eta partaideek datu multzo bera erabiltzen dute euren sistemak entrenatu eta ebaluatzeko, sistema onena zein den jakite aldera. 2006an izan zen X. *CoNLL* lehiaketan *(CoNLL-X)*, dependentzietan oinarritutako analizatzaile sintaktiko estatistikoak garatzeko 13 hizkuntzatarako corpusak aurkeztu ziren. Lehiaketa horretarako erabili zen formatuari *CoNLL-X* formatua deitu zitzaion. Formatu horretan, esaldiak lerro huts batekin daude banatuta. Esaldiko hitz bakoitzaren informazioa lerro batean jartzen da 2.2 taulan ikusten den bezala. Hitz-forma bakoitzeko, tabuladore batekin banatutako 10 zutabetan hurrengo 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, mugatasuna (mugatua

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

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

ala mugagabea den), aspektua, eta abar. Adibidez, 2.2 taulan "Lehen zatia aspergarria izan zen oso." esaldiaren irudikapena *CoNLL-X* formatuan agertzen da.

2.2 taulan "Lehen" hitzaren identifikatzailea 1 zenbakia da, eta "zatia" hitzaren identifikatzailea 2 zenbakia da. "Lehen" hitzaren HEAD zutabean 2 bat agertzeak "zatia" hitzaren mende dagoela esan nahi du. Modu berean, "zatia" hitza "izan" laugarren hitzaren mende dago. Azkenik, "izan" aditzak (erroa denez) ez du gobernatzailerik.

Ebaluazioa

Sailkapen prozesuetan, askotan ez da erraza izaten sistemen arteko konparaketa garbiak egitea. Beraz, zein da sailkatzaile edo analizatzaile hobea? Galdera honi erantzuteko jakin beharko genuke zer den lortu nahi dena, eta horren arabera erabaki. Dependentzietan oinarritutako sailkapen sistemaren zehaztasuna ebaluatzeko zenbait neurri definitu izan dira, eta gure helburua lortzeko erabilitakoak azalduko dira. Gure ebaluazioa egiteko zuhaitz-bankuak erabiliko dira, zuhaitz-bankuaren zati handi bat (% 80 inguru) entrenatzeko erabiliko da eta bertatik ikasi beharko du sistemak; beraz, garrantzitsua da entrenatzeko zatian ahalik eta kasu gehien agertzea, eta, ahal izanez gero, denak agertzea. Entrenatzeko erabiliko ez dena sistema ebaluatzeko edo probatzeko erabiliko da. Ebaluatzeko corpus zati honetan, adibide edo instantzia bakoitzak informazio sintaktiko osoa definituta izango du, baina sailkatzaileak ez du informazio sintaktiko hori kontuan izango iragarpena egiteko; haatik, ikasitakoan oinarrituz emango du bere ustez instantzia bakoitzari dagokion informazio sintaktikoa. Gero, sailkatzaileak iragarritako informazio sintaktikoak zuhaitz-bankuko etiketa errealekin konparatuko dira eta emaitzak aterako dira, asmatze-tasak eta bestelako neurriak kalkulatuz. Tesi-lan honetan erabili diren neurriak hauexek izan dira:

1. *Labeled Attachment Score (LAS)*: Sistemaren erabakia zuzena izateko probabilitatea neurtzen du, baina erabaki zuzena izateko bai dependentzia-etiketa bai gobernatzaile esleipena, zuzenak izan behar dira.

2. *Unlabeled Attachment Score (UAS)*: Sistemaren erabakia zuzena izateko probabilitatea neurtzen du, baina erabaki zuzena izateko soilik gobernatzaile esleipenak izan behar du zuzena.

3. Doitasuna (ingelesez, *precision*): Sistemak hautemandako elementu guztietan zuzen hautemandako elementuen proportzioa da, edo sistemaren erabakia zuzena izateko probabilitatea.

4. Estaldura (ingelesez, *recall*): Sistemak hauteman beharreko elementu guztietatik hautemandako elementuen proportzioa da.

5. Esangura-test (ingelesez, *significance test*): Bi analizatzaile sintaktikoen ebaluazioen artean dauden diferentziak estatistikoki esanguratsuak diren jakiteko (beti ere huts-egite probabilitate bat kontuan izanik) Bikelen ausazko analizatzaile sintaktikoen ebaluazioen konparadorea[2](ingelesez, *Randomized Parsing Evaluation Comparator*) erabili da. Bikelen konparadoreak McNemar-en testa inplementatzen du. McNemar-en testeko p balioa 0 eta 1 bitartekoa da, eta $p$ren balio baxuek "ez dela hobekuntzarik gertatu esaten duen hipotesia" baztertzen dute. Horrela, esperimentuen tauletan, hobekuntza oso esanguratsua dela adierazteko p <0,005 eta ‡ikurra eta hobekuntza esanguratsua dela adierazteko, p <0,05 eta †ikurra erabili ditugu.

2.2.2 Zuhaitz-bankuen adibideak

Aranzabe-ren (2008) tesi-lanean azaltzen denez, zuhaitz-bankua osatzerakoan erabili behar den eredua aukeratzeko orduan arrazoi anitz daude:

- Ingelesa bezalako hizkuntzetan, osagaien hurrenkera finkoa duten hizkuntzetan osagai-eredua erabiltzea
- Euskara bezalako hizkuntzetan, osagaien hurrenkera librea duten hizkuntzetan dependentzia-eredua erabiltzea
- Erabileraren araberako aukera egitea
- Tradizio linguistikoa jarraitzea

[2]http://ilk.uvt.nl/conll/software.html

Hurrengo ataletan, tesi-lan honetan erabilitako bi zuhaitz-bankuak deskribatuko dira: osagai-ereduan etiketatuta dagoen *Penn Treebank*a eta dependentzia-ereduan etiketatuta dagoen euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe, 2008).

2.2.2.1 *Penn Treebank*a (PTB)

Ingeleseko *Penn Treebank*[3] zuhaitz-bankuan (Marcus *et al.*, 1993) *Wall Street Journal* eta *Brown Corpus*eko testuak baliatu dira (Taylor *et al.*, 2003). Etiketatze-lan hori erdi-automatikoki egin da, osagaietan oinarritutako ereduari eta X Marra teoriari (ingelesez, *X-bar theory*) jarraiki; hau da, lehendabizi automatikoki etiketatu eta, ondoren, eskuz egin da zuzenketa. Tesi-lan honetan, entrenatzeko *Wall Street Journal*eko 2-21. sekzioak (milioi bat hitz eta 40.000 esaldi ingurukoa) eta ebaluatzeko *Wall Street Journal*eko 23. sekzioa (56.684 hitz eta 2.416 esaldi ingurukoa) erabili dira. *Penn Treebank (PTB)* osoa osagai-ereduan etiketatuta dago. Adibidez, *Penn Treebank*eko *Wall Street Journal* sekziotik ateratako ingeleseko esaldia etiketatzeko, 2.3 irudiko etiketatze-modua segitu da; hau da, egiturei dagozkien etiketak parentesien hasieran baino ez dira jartzen. Baina tesi-lan honetan dependentzietan oinarritutako analizatzaileak soilik erabiliko diren legez, osagai-zuhaitzak dependentzia-zuhaitz bihurtu behar dira. Horiek dependentzia-zuhaitz bihurtzeko, hainbat tresna daude, eta gure esperimentuak egiteko *Pennconverter* (Johansson eta Nugues, 2007b) eta *Penn2Malt* (Yamada eta Matsumoto, 2003) bihurtzaileak probatu dira.

((S (NP-SBJ (PRP **Its**) (JJ **maximum**) (NN **velocity**)) (VP (VBZ **is**) (NP-PRD (CD **72**) (NN **mph**))) (. **.**)))

2.3 irudia – Ingeleseko *Penn Treebank*eko eta 2.1 irudiko esaldiaren etiketatze-modua (S: perpausa; NP-SBJ: izen-sintagma subjektua; PRP: izenordain posesiboa; JJ: adjektiboa; NN: izena, singularra; VP: aditz-sintagma; VBZ: aditza, orainaldian; NP-PRD: atributua; CD: zenbakia)

[3] *Penn Treebank*aren informazioa `http://www.cis.upenn.edu/treebank/home.html` dago

2.2.2.2 Euskarako zuhaitz-bankuak

Lehenengo atalean euskararen ezaugarri nagusiak aztertuko dira. Euskarak perpauseko osagaien hurrenkera librea du; hori dela-eta, dependentzia--ereduari jarraituz osatu da EPEC-DEP zuhaitz-bankua. Bigarren atalean, EPEC-DEP zuhaitz-bankua *CoNLL-X* formatura egokitutako euskarazko lehen zuhaitz-bankua (EZB I zuhaitz-bankua) eta gaur egun erabiltzen ari garen bigarren bertsioaren (EZB II zuhaitz-bankua) deskribapena egiten da.

Euskararen ezaugarriak

Euskara hurrenkera libreko hizkuntza da; modu batera baino gehiagotara antola daitezke perpaus bateko osagaiak. Adibidez, 2.4 irudiko "gizonak sagar bat jan du" perpausean, perpauseko osagaiek ordena-aldaketa ugari izan ditzaketela ikusten da, perpausaren esanahi nagusia aldatu barik.

gizonak sagar bat jan du
sagar bat gizonak jan du
gizonak jan du sagar bat
sagar bat jan du gizonak
jan du gizonak sagar bat

2.4 irudia – "gizonak sagar bat jan du." perpauseko osagaien ordena--aldaketa

Sareko Euskal Gramatikan[4] agertzen den bezala, ezaugarri morfologikoei dagokienez, euskara hizkuntza eranskaria da (Aduriz *et al.*, 2000); izan ere, testu-hitz baten lemak morfema asko har ditzake; adibidez, "gizon" bezalako izen arrunt batek 135 forma ezberdin har ditzake, kasua eta numeroa soilik konbinatuz (gizonak, gizonek, gizona, etab.). Gainera, hitzari errekurtsiboki gehitu ahal dizkiogu atzizkiak. Adibidez, "Etxekoak etorri dira" esaldian, "etxekoak" hitzak bi postposizio-atzizki edo kasu-marka ditu: "-ko" lokatibo adnominala eta "-ak" absolutiboa.

Euskarak dituen adizkiak konplexuak dira, horietan informazio handia jasotzen delako, hala nola, ekintzan parte hartzen duten elementuei buruz (subjektua, osagarri zuzena eta zeharkako osagarria) edo kasu-gramatikalari buruz (nor, nor-nori, nor-nork edo nor-nori-nork), aspektuari buruz (burutua, ez-burutua, geroa), eta denborari buruz (oraina, iragana, modua).

[4] http://www.ehu.es/seg/

Euskararen sintaxi-egituran, morfemek oso zeregin garrantzitsua betetzen dute. Fenomeno hau hizkuntza postpositiboetan gertatzen da. Hizkuntza postpositiboetan, sintagmaren azkeneko osagaian biltzen dira ezaugarri morfologikoak, besteak beste: kasua, numeroa eta mendeko perpausen informazioa. Adibidez, "Nork dantzatuko ote du soinutxo hori?" esaldian, "soinutxo hori" hitzek "zer" sintagma osatzen dute eta zer sintagmaren azken osagaia den "hori" erakuslearen bitartez adierazi dira ezaugarri morfologiko horiek, eta ez sintagma osatzen duten osagai guztien bitartez.

Euskaraz, gehienetan, sintagmaren azkeneko osagaian biltzen dira ezaugarri morfologikoak; hori dela-eta, hizkuntza buru-azkena dela esango da, japoniera edo turkiera bezala. Tesi-lan honetan euskara hizkuntza buru-azkena den ezaugarria kontuan izan da, euskarako zuhaitz-bankuan transformazio ugari egiteko (ikusi 4.1) analisia hobetzeko asmoarekin.

Euskarako zuhaitz-bankuak

Euskaraz sintaktikoki etiketatutako Eus3LB corpusa (Aduriz *et al.*, 2006a) 3LB proiektuaren barnean garatu zen. 3LB proiektuan, katalaneko, euskarako eta gaztelaniako maila morfosintaktikoan eta sintaktikoan eskuz etiketatutako 3 corpus sortu ziren (Palomar *et al.*, 2004). Katalaneko eta gaztelaniako corpusak osagai-eredua erabiliz sortu ziren eta euskarazko corpusa, berriz, dependentzia-eredua erabiliz. Corpus honetan ez dago inolako anbiguotasun sintaktikorik eta esaldi bakoitzari zuhaitz bakarra dagokio. Esaldiak XX. mendeko euskararen corpus estatistikotik (Aduriz *et al.*, 2006b) eta Euskaldunon Egunkaritik lortu dira. Corpusa hitz mailan etiketatuta dago, eta dependentzia-erlazioak hitzen artean baino ez dira ematen. Gainera, sintagmen buru esanahi semantikoa duen hitza ezarri da. Corpusa sintaktikoki etiketatzeko, dependentzia-gramatikaren eredua (Tesnière, 1959) jarraitu da. Dependentzia-erlazioak oinarri dituen etiketatze-eskema definitzeko Carroll *et al.*-en (1998) proposamena jarraitu da. Testuko esaldi bakoitza erlazio gramatikal batzuekin markatzen da, mendekoaren eta bere gobernatzailearen arteko dependentzia sintaktikoa zehaztuz. Lehen euskarazko corpusa (Aduriz *et al.*, 2006a) *CoNLL-X* formatura (ikusi 2.2 taula) egokitu zen (Bengoetxea eta Gojenola, 2007) dependentzietan oinarritutako analizatzaile sintaktikoen sortzaileek erabil zezaten. 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 bezala ezagutuko dena). EZB I zuhaitz-bankuak badu beste berezitasun bat ere: arku ez-proiektiboak. Eus-

karako zuhaitz-bankuak % 2,9 arku ez-proiektibo ditu, eta esaldien % 26,2k arku proiektiboren bat du. Lehen bertsio honekin, *CoNLL* 2007 *Multilingual Dependency Parsing*[5] lehiaketan, beste 9 hizkuntzekin batera, eta 20 sistemen artean ebaluatu zuten. 2010ean, dependentzia-ereduan etiketatuta dagoen euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe, 2008) *CoNLL-X* formatura egokitu zen dependentzietan oinarritutako analizatzaile sintaktikoen sortzaileek erabil zezaten. *CoNLL-X* formatura moldatu ondoren, 150.000 hitz eta 11.225 esaldikoa den bigarren euskarako zuhaitz-bankua lortu zen (hemendik aurrera EZB II zuhaitz-bankua bezala ezagutuko dena). EZB II zuhaitz-bankuak diseinu eta ezaugarri berriak ditu:

1. EZB I zuhaitz-bankua baino hiru aldiz handiagoa da.

2. Euskarako zuhaitz-bankuaren bigarren bertsio honek % 1,3 arku ez-proiektibo ditu. Agerikoa denez, arku ez-proiektiboen kopurua jaitsi egin da; izan ere, EZB I zuhaitz-bankuan elipsia markatzen zen. Hori dela-eta, *CoNLL-X* formatura bihurtzerakoan elipsiak *ROOT* errora lotzen ziren, erlazio ez-proiektiboak sortuz. Baina EZB II zuhaitz-bankuan elipsia ez da existitzen, esaldian esplizitu agertzen diren hitzak etiketatu direlako.

3. *CoNLL-X* formatuan lortzeko, TEI-P4 (Artola *et al.*, 2005) anotazio-amaraunean dauden *XML*ko fitxategietatik informazioa jasotzeko, C++ programazio-lengoaia eta LibiXaML-ko[6] liburutegia erabili dira.

2.3 Analizatzaile sintaktikoak

Atal honen hasieran, analizatzaile sintaktikoek esaldi baten egitura sintaktikoa bueltatzeko gainditu behar dituzten arazoak azalduko dira. Halaber, arazo hauek konpontzeko dauden ikuspegi desberdinak aipatuko dira. Tesi-lan honetan estatistikan eta datuetan oinarritutako analisia egin da. Hala, hurrengo ataletan arlo honetan izan diren joera nagusiak azalduko dira.

[5] *CoNLL* 2007 *Multilingual Dependency Parsing* lehiaketa http://www.cs.jhu.edu/EMNLP-CoNLL-2007

[6] LibiXaML liburutegia http://ixa2.si.ehu.es/docs/libixaml/html/hierarchy.html

2.3.1 Problemaren zehaztapena

Sintaxiaren tratamendu konputazionalari ekiteko, Gojenola-ren (2000) tesian zailtasun nagusi bi aipatzen dira:

1. **Osatu gabeko formalismo teorikoa**. Gaur egun, esaldi mailan formalizazio teorikoa ez da osoa, gauzatu gabeko lana baita. Testu errealetan agertzen diren hainbat esaldi luzetan (adibidez, 50 hitzetik gorako esaldietan), linguistikoki deskribatu gabeko egitura mota berriak agertzen dira.

2. **Anbiguotasuna**. Esaldiko hitz bakoitzak bi edo hiru analisi posible izateak edozein esaldi arrunten analisi sintaktikoan aukera asko sorraraziko ditu; gehienak zentzugabeak izan daitezke testuingurua aztertuz gero, baina sintaktikoki zuzenak izan daitezke. Adibidez, "Gizonak etxeak ikusi ditu" perpausak, gutako edozeinentzat arazorik izango ez lukeenak, arazoak eman ahal dizkio sistema automatiko bati, modu bitara uler baitaiteke: "Gizonak" subjektutzat eta "etxeak" objektutzat hartuz (interpretazio arrunta), baina baita "etxeak" subjektu bezala interpretatuz ere lor daitekeena. Beraz, esaldi mailako tratamenduaren ikergai nagusi bat anbiguotasuna ezabatzearena izango da. Horren zailtasunaren neurri bat emateko, esan dezagun euskaraz, hitz mailako anbiguotasuna 2,6 hitzekoa dela, eta horri gramatika sintaktiko batek lortutako egiturak gehitzen bazaizkio, esaldi normal batek milaka aukera izan ditzake. Helburua, beraz, esaldi bakoitzeko analisi bakarra lortzea izango da.

2.3.2 Hurbilpenak

Bi arazo hauek gainditzeko, hau da, esaldi osoaren interpretazio posible guztiak identifikatu eta beraien artean bat aukeratzeko, hurbilpen asko asmatu dira. Sintaxi konputazionala gauzatzeko dauden joera nagusiak sailkatzeko orduan, ikuspegi desberdinak erabil ditzakegu. Gojenola-ren (2000) tesian hiru aukera nagusi bereizten dira:

Ezagutza linguistikoan oinarritutako sintaxia
Ezagutza linguistikoa kodetzeko gramatikak erabili ohi dira perpausaren analisiak lortu eta zuzena aukeratzeko. Gramatika erregela multzo bat da, eta, gehienetan, pertsona batek egin ditu. Linguistikoki interesgarrienak diren

esaldiez arduratzen dira eta testu errealak kontuan hartuz. Ezagutza linguistikoan oinarritutako gramatika batek abantaila nabarmenak ditu, izan ere, gizaki batek egiteak zehaztasuna, zorroztasuna eta zuzentasuna lortzea dakar; aitzitik, gramatika bera garatzeak eragiten duen kostu handia eta hura mantentzeko zailtasunak aipatu behar dira. Sintaxiaren tratamendurako aukera desberdinak aztertu ondoren, euskararen sintaxia lantzeko ezagutza linguistikoan oinarritutako testuingururik gabeko gramatikak (TGG) eta egoera finituko mekanismoak (automatak eta transduktoreak) erabil daitezkeela ondorioztatu zuten Gojenola-k (2000) eta Aduriz-ek (2000) beren tesi-lanetan.

Teknika estatistikoetan oinarritutako sintaxia
Teknika estatistikoetan oinarritutako sistemek indar handia hartu dute 1990-eko hamarkadatik aurrera. 1980ko hamarkadan, batik bat, sintaxiaren tratamendua hizkuntzalariek eskuz idatzitako gramatiketan oinarritzen zen. Teknika estatistikoetan oinarritutako sistemek zuhaitz-bankuaren beharra dute. Zuhaitz-bankutik ateratako probabilitateak erabiltzen dituzte erlazio sintaktikoak erabakitzeko mementoan. Gramatikak garatzeko orduan, batzuetan, eskuzko lan gramatikal minimoa egiten da, ezagutza linguistikoa zuhaitz-bankuetan agertzen diren elementuetatik eta beren maiztasunetatik ateratzen baita. Horren adibide, sintaktikoki etiketatutako ingelesezko *Penn Treebank*a[7] ipini da. Zuhaitz-bankuak tamaina handia izan behar du, gertaera linguistikoen deskribapen zabala izateko. Gaur egun, hizkuntza askotarako *treebank*ak eskuragarri dira. Esan behar da corpus bat etiketatzea lan luzea eta zaila dela, baita informazio gramatikalaren zati bat automatikoki ateratzen denean ere.

Ezagutza linguistikoaren eta teknika estatistikoen konbinazioa
Estatistika hutsa erabiltzeak arazoak izan ditu testuinguru mugatuetan gertatzen ez diren fenomenoak tratatzeko. Adibidez, *trigram*etan oinarritutako sistema batean nekez azter daitezke hiru hitz baino gehiago hartzen dituzten gertaera linguistikoak, aditza eta osagarri nagusien artekoak kasu; ez, ordea, testuinguru hurbileko erlazioak aztertzeko, izenaren, adjektiboaren eta determinatzaileen artekoak, adibidez. Gainera, ikuspuntu estatistiko hutsean oinarritutako gramatikekin lortutako analisiek beste arazo bat dute, emaitza horiek linguistikoki interpretatzea ez baita erraza, eta horrek zailtasun

[7] http://www.cis.upenn.edu/~treebank

handiak jar diezazkieke ondorengo prozesuei, interpretazio semantikoa kasu. Hizkuntzalariek idatzitako gramatiketan, aldiz, maila altuko gertaera linguistikoak deskribatu ohi dira, sintagmak zein esaldi osoak konbinatzeko, baina arreta gutxiago eskaini zaio esaldi errealetan agertzen den zenbait fenomenori, egitura jakin baten maiztasuna esaterako. Horregatik, metodo probabilistikoak eta ezagutza linguistikoa lotzeko saioak egin dira, bakoitzaren abantailak biltzeko asmoz. Horren adibide dira LFG[8](Riezler *et al.*, 2002) edo HPSG[9](Müller, 1996) formalismo linguistikoen inplementazioak, zeinetan hasieran formalismo linguistiko hutsa zenari ezagutza kuantitatibo eta probabilistikoa gehitu zaion, horrela anbiguotasunaren arazoari aurre egin ahal izateko.

Gaur egun, estatistikan eta datuetan oinarritutako analizatzaileen etekina asko hobetu da. Tesi-lan honetan, teknika estatistikoetan eta datuetan (dependentzietan etiketatutako zuhaitz-bankuetan) oinarritutako sintaxia egin da. Hurrengo ataletan, arlo honetako joera nagusiak zeintzuk diren azalduko dira.

2.3.3 Teknika estatistikoetan oinarritutako sintaxia

Teknika estatistikoetan oinarritutako sintaxian bi joera nagusi bereiz daitezke.

- Osagai-egituretan oinarritutako analizatzaileak, erregeletan oinarritutako gramatika erabiltzen dutenak, adibidez, *Stanford*eko analizatzailea (Klein eta Manning, 2002) edo *Berkeley*ko analizatzailea (Petrov *et al.*, 2006) aipa daitezke.
- Dependentzia-egituretan oinarritutako analizatzaileak, adibidez, MaltParser (Nivre *et al.*, 2004), trantsizioetan oinarritutako modeloa, eta MSTParser (McDonald *et al.*, 2005), grafoetan oinarritutako modeloa aipa daitezke.

2.3.3.1 Osagai-egituretan oinarritutako analizatzaileak

Esaldia egitura hierarkikoa bailitzan ikusteko algoritmo gehienak TGGn oinarritzen dira (ikusi 2.2.1.1). Erregela bakoitzak probabilitate bat esleituta

[8] http://www.ehu.es/seg/hizk/1/4#lexical-functional_grammar_lfg

[9] http://www.ehu.es/seg/hizk/1/4#head-driven_phrase_structure_grammar_hpsg

dauka. Erregela bakoitzaren probabilitatea kalkulatzeko ikasketa automatikoa eta hizkuntzalariek eskuz kodifikatutako zuhaitz-bankua erabiltzen da. Zuhaitz onargarrien basoan, probabilitate handiena izango duen zuhaitza bilatzeko algoritmo ugari daude. *Probabilistic context-free grammar*en (PCFG) zuhaitz baten probabilitatea kalkulatzeko, zuhaitza osatzeko erabilitako erregelen probabilitatearen biderkaketa erabiltzen da. Gaur egun, teknika estatistikoetan eta osagai-egituretan oinarritutako analizatzaile garrantzitsuenak *PCFG*etan oinarritzen dira, besteak beste, *Stanford*ekoa (Klein eta Manning, 2002), Collins-ena (Collins, 2009; Bikel, 2004), Charniak eta Johnson-ena (Charniak eta Johnson, 2005) eta *Berkeley*koa (Petrov eta Klein, 2007).

2.3.3.2 Dependentzia-egituretan oinarritutako analizatzaileak

Azken urteotan, hizkuntzaren prozesamenduan areagotu egin da dependentzietan oinarritutako analisien interesa. Arrakasta horren arrazoi batzuk aipatzearren:

- Tresna bakar batekin, tipologia ezberdineko hizkuntzak analizatu ahal dira.
- Hurrenkera libreko hizkuntzei osagai-eredua baino dependentzia-eredua egokitzen zaie hobeto.
- Dependentzia-ereduko irudikapena, hainbat aplikaziotan (itzulpen automatikoan, informazio-berreskuratzean, eta abar) oso mesedegarria da, oso ondo adierazita baitator predikatu-argumentu egitura.
- Hizkuntza askotarako balio duten analizatzaile sintaktikoak garatzea ekarri du.

Metodo gainbegiratuek ikasketa fasean egitura zuzena behar dute. Zuhaitz-banku batetik ikasi ostean, esaldi berriak analizatzeko gai izango dira. Metodo gainbegiratuek erabiltzen dituzten analizatzaile sintaktiko automatikoek bi arazo konpondu behar dituzte:

- Ikasteko arazoa: dependentzia-egitura zuzenekin emandako D zuhaitz-bankua erabiliz, M analizatzaile sintaktiko bat ikasi behar du.
- Esaldi berriak analizatzeko arazoa: ikasitako M analizatzaile sintaktikoarekin, esaldi berriei dependentzia-egitura esleitzeko gai izatea.

2006 eta 2007ko *CoNLL Shared Task* konferentzietan hizkuntza askotarako zuhaitz-bankuak eta dependentzietan oinarritutako analizatzaileak garatzeko izan ziren (Buchholz eta Marsi, 2006; Hall *et al.*, 2007). Konferentzia horien ostean ikasketa automatikoan oinarritutako hurbilpen bi nagusitu ziren:

- Grafoetan oinarritutako modeloak (Eisner, 1996, 2000; McDonald eta Pereira, 2006)
- Trantsizioetan oinarritutako modeloak (Nivre, 2003; Yamada eta Matsumoto, 2003; Titov eta Henderson, 2007)

Grafoetan oinarritutako analizatzaileak

Grafoetan oinarritutako analisi sintaktikoaren helburu nagusia da esaldia osatzen duten zuhaitz-arku posible guztien batasunetik sortutako zuhaitz-bildumatik edo -basotik, probabilitate handiena duen dependentzia-zuhaitza lortzea. Zuhaitza konektatutako grafo zuzendua izango da. Erpinak esaldiko hitzak izango dira eta erro erpina edozein zuhaitzen buru izango da. Puntuazioak (ingelesez, *score*) zuhaitz posible baten probabilitatea adierazten du. Sistema ezberdinek puntuazioa era ezberdinetan tratatzen dute: sailkatzaile lineala, baldintzazkoa edo elkarrekiko probabilitateak bezala. Baina denek azpiegituren puntuazioen faktorea erabiltzen dute. Grafoetan oinarritutako analizatzaile sintaktikoaren ezaugarri nagusiak honako hauek dira:

- Arkitektura: gehienetan, analisi sintaktikoa bi urratsetan gauzatzen da. Lehen urratsean, buru egokia zein den aukeratzen da eta, bigarren urratsean, dependentzia-etiketak ezartzen dira zuhaitz-arkuetan.
- Modeloak: puntuazio funtzioaren arabera, bi modelo bereizten dira:
  - Lehen mailako modeloak (ingelesez, *first-order models*): zuhaitza osatzen duten arkuen puntuazioen faktorizazioa erabiltzen denean, lehen mailako algoritmoa erabiltzen ari dela esango da. Baina arkuak era independentean tratatzean, hizkuntzetan dauden egitura konplexuen pisua ordezkatuta etortzeko aukera txikiagoa da.
  - Bigarren mailako modeloak (ingelesez, *second-order models*): bigarren mailako algoritmoek, ordea, elkarren ondoko arku bikoteen puntuazioa hartzen dute kontuan. Ondoko dependentzia-egituren informazio guztia konputazionalki kontuan hartzea ezinezkoa denez, Markov-en hurbilpena erabiltzen da.
- Inferentzia algoritmoa: zuhaitz baten puntuaziorik altuena bilatzen duen inferentzia algoritmo aukeraketa, lehen edo bigarren mailako modeloa erabiliko den eta arku proiektibo edo ez-proiektiboak tratatu nahi diren araberakoa da. Adibidez, lehen-mailako modeloa eta arku ez-proiektiboekin lan eginez gero, gehienek MST (*Maximum Spanning Tree*) algoritmoa erabiltzen dute. Analisi proiektiboan, ordea, gehien

erabiltzen den algoritmoa Eisner-en (1996) algoritmoa da, algoritmo honek zuhaitz proiektibo guztien gaineko bilaketa lehen edo bigarren mailako ezaugarriak erabiliz, denbora kubikoan gauzatzen baitu.

- Ikasketa metodoa: gehienek sailkatzaile bat entrenatzeko lineako inferentzian oinarritutako (ingelesez, *online inference-based*) metodoak erabiltzen dituzte, adibidez: *Perceptron* algoritmoa, *Probabilistic Loglinear*, *Averaged Perceptron* (Freund *et al.*, 1999; Collins, 2002), *Large-Margin MIRA (Margin Infused Relaxed Algorithm)* (Crammer eta Singer, 2003; Shalev-Shwartz *et al.*, 2003), eta abar.

Mota honetako analizatzailerik adierazgarriena MSTParser (McDonald *et al.*, 2005, 2006) da, baina beste sistema batzuk (Dreyer *et al.*, 2006) edo (Carreras, 2007) gutxietsi gabe.

Trantsizioetan oinarritutako modeloak

Trantsizioetan oinarritutako modeloek (Nivre *et al.*, 2004; Yamada eta Matsumoto, 2003) dependentzia-arkuak ikasi beharrean, makina abstraktu baten trantsizioak ikasten dituzte. Trantsizio-segida amaitu ostean, dependentzia-zuhaitza sortzeko gai dira. Une bakoitzean trantsizio egokiena zein den erabakitzeko, sailkatzaile bat erabiltzen dute. Sailkatzailearen sarrera konfigurazioak (ezaugarri-bektoreak) izango dira eta sailkatzailearen irteera trantsizio zuzena izango da. Sailkatzaile bat sortzeko, emaitza zuzenak dituen zuhaitz-banku bat erabiliko da, hau da, ikasketa automatiko gainbegiratua erabiliko da. Trantsizioetan oinarritutako analizatzaile sintaktikoaren ezaugarri nagusiak ondoren aipatzen dira:

- Arkitektura: gehienetan analisi sintaktikoa urrats batean gauzatzen da, baina sistema gehienek analisi proiektiboa gauzatzen dute. Zuhaitz ez-proiektiboekin lan eginez gero, teknika osagarriak erabiltzeko aukera dago.
- Modeloak: erabilitako trantsizio algoritmoaren arabera gehien erabiltzen diren modeloak *shift-reduce* algoritmoaren aldaerak dira. Analizatzailearen uneko egoera gordetzeko, hurrengo egiturak erabiltzen ditu:
  - Pila: pilan partzialki tratatutako hitzak daude.
  - *Buffer*: *buffer*ean oraindik tratatu gabe dauden esaldiaren hitzak daude.
  - Dependentzia-arkuen multzoa: dependentzia-arkuen multzoan uneko egoerara heldu baino lehen analizatzaileak hartutako erabakiekin eratutako zuhaitz partzialak daude.

Modelo hauetan, pila eta *buffer*en artean ematen dira trantsizioak. Baina badaude beste modelo batzuk: *Covington*, *Stack* eta *Multiplanar* bezalakoak, esaterako.

- Inferentzia algoritmoa: algoritmo gehienek, esaldiak ezkerretik eskuinera prozesatzeko, algoritmo jale eta determinista bat erabiltzen dute. Algoritmo hauek, hurrengo trantsizio egokia zein den jakiteko, analizatzailearen historia (uneko egoerara heldu baino lehen analizatzaileak hartutako erabakiekin eratutako zuhaitz partziala) eta uneko sarreraren informazioa (sarrerako esaldia) kontuan hartzen dituzte.
- Ikasketa metodoa: sailkatzailea entrenatzeko analizatzailearen historia eta uneko sarreraren informazioa erabiliko da. Sailkatzaile bat entrenatzeko hainbat hurbilpen erabili dira: *SVM, Modified Finite Newton SVMs, Maximum Entropy Models, Multiclass Averaged Perceptron, Maximum Likelihood Estimation*, eta abar.

Trantsizioetan oinarritutako sistema ugari daude (Nivre *et al.*, 2006b; Johansson eta Nugues, 2006; Wu *et al.*, 2006.

Sistema hibridoak

Gaur egun, grafoetan eta trantsizioetan oinarritutako modeloen onurak sistema batean bateratzen dituzten sistemak aurki ditzakegu, hala nola, Zpar[10] (Zhang eta Clark, 2008; Zhang eta Nivre, 2011). Horretaz gainera, badira sistema biak integratzen dituztenak, eta horietan zein erabili nahi den aukera ematen digutenak, adibidez, Mate (Bohnet eta Nivre, 2012). Zpar (Zhang eta Clark, 2008; Zhang eta Nivre, 2011) trantsizioetan oinarritutako sistema bat da, baina bilaketa lokala, determinista eta jalea erabili beharrean, Zhang *et al.*-ek (2011) *beam-search*[11] eta algoritmo globala erabiliko dute. *Beam-search* algoritmoak une oro puntuazio funtzioa[12] kalkulatzeko ezaugarri-bektore globala[13] eta ikasitako parametro bektorea erabiliko ditu. Parametro bektorea ikasteko, ikasketa automatiko gainbegiratua egiten duen *Averaged Perceptron* sailkatzailea erabiltzen du. Sailkatzailea entrenatzeko ezaugarri-bektore globala erabiliko da. Zpar-ek MaltParser sistemako Ar-

[10] http://www.sourceforge.net/projects/zpar

[11] *Parsing* urrats inkremental bakoitzean puntuazioak batu ostean, puntuazio maximoa lortzen dituzten B zuhaitz-egiturekin geratzen da.

[12] $Puntuazioa(y) = \Phi(y) * \vec{w}$, non y esaldiaren irteera egitura, $\Phi(y)$ ezaugarri-bektore globala eta $\vec{w}$ parametro bektorea da.

[13] Esaldi-egitura, ezaugarri-bektore global batera bideratu da eta ezaugarri-txantiloi batean definitutako ezaugarri guztietatik, zenbat bete diren adierazten da.

ku Azkarra (ingelesez *arc-eager*) algoritmoaren trantsizio eta datu-egitura berdinak erabiltzen ditu. Esaldiaren dependentzia-zuhaitza lortzeko, denbora lineala erabiltzen du. Mate-k (Bohnet eta Nivre, 2012; Carreras, 2007; Johansson eta Nugues, 2008) lanetan deskribatutako algoritmoa garatzen du. MST (*Maximum Spanning Tree*) algoritmo bat da, hau da, zuhaitz-arkuen parametroen batuketa maximizatzen duen algoritmoa. Bigarren mailako faktorizazioa erabiltzen du, hau da, guraso bereko ondoko arku-bikoteen puntuazioa kontuan hartzen du zuhaitz baten puntuazioa kalkulatzeko orduan. Analizatzaile honek (Bohnet, 2010) zehaztasuna eta abiadura hobetzen ditu, paraleloan analizatu eta ezaugarrien erauzketa egiten duen *hash kernel* algoritmo berri bat ustiatzen du.

Sistema konbinatuak

Sistema bakun batek egiten dituen erroreak zuzentzeko, dependentzietan oinarritutako analizatzaileak konbinatzeko ahalegin ugari egon dira:

- Ikasketa denboran analizatzaile sendoak integratzen dituzten modeloak. Adibidez, *classifier stacking* edo analizatzaile-pilaketan, lehenengo analizatzailearen irteeran lortutako ezaugarriak bigarren analizatzaile baten sarrera aberasteko erabiliko dira (Martins *et al.*, 2008; Nivre eta McDonald, 2008; McDonald eta Nivre, 2011).
- Analisi garaian, era independentean entrenatutako modeloak konbinatuko dira, adibidez bozketa bidezko analizatzaile ezberdinen konbinaketa (Sagae eta Lavie, 2006; Hall *et al.*, 2007; Attardi eta Dell'Orletta, 2009). Analisi garaian, dependentzia-zuhaitzaren zuzentasuna ziurtatzeko irtenbide hauek nagusitu dira:
  - Oinarri diren modeloen zuhaitz osoa aukeratzen duen algoritmoa erabili da (Henderson eta Brill, 1999).
  - *Re-parsing* egiteko konplexutasun kubikoa duen eta programazio dinamikoan oinarritutako Eisner-en (1996) algoritmoa erabili da (Sagae eta Lavie, 2006; Hall *et al.*, 2007).
  - Attardi eta Dell'Orletta-ren (2009) lanean, *re-parsing* egiteko *top-down* estrategia jale eta konplexutasun lineala duen algoritmoa erabili da.

2.4 Laburbilduz

Kapitulu honetan sintaxi konputazionalaren bilakaera aztertu da. Zuhaitz--bankuak etiketatzeko dauden eredu nagusiak eta tesi-lan honetan erabiliko diren zuhaitz-bankuen deskribapena egin da. Gero, analizatzaile sintaktikoak aztertu dira, ezagutza linguistikoa, teknika estatistikoak eta euren arteko konbinazioa aintzat hartuta. Azkenean, tesi-lan honetan erabiliko diren teknika estatistikoetan oinarritutako sintaxiaren joera nagusiak aurkeztu dira. Hurrengo kapituluan, alde batetik, teknika estatistikoetan oinarritutako analizatzaileen sortzaileak erabiltzeko behar den informazio morfosintaktikoa lortzeko IXA taldean dauden baliabideak eta tresnak aurkeztuko dira; beste aldetik, teknika estatistikoetan oinarritutako analizatzaileen sortzaileak zuhaitz-bankuetara egokitzeko jorratutako urratsak azalduko dira.

3

Euskararako analizatzaile sintaktikoa

Kapitulu honetan, IXA taldean informazio morfosintaktikoa eta sintaktikoa lortzeko gaur egun erabiltzen diren baliabideen deskribapena egiten da. 3.1 atalean, IXA taldean esaldi baten analisi sintaktikoa lortzeko erabiltzen diren baliabideak edo moduluak aurkeztuko dira; ondoren, 3.2 atalean IXA taldean analisi sintaktikoa egiteko erabiltzen ari garen tresnak aztertuko dira eta, gero, 3.3 eta 3.4 ataletan, MaltParser eta MSTParser sistemak egokitzeko emandako urratsak azalduko dira. Hurrengo 3.5 atalean, ezaugarri morfosintaktikoek analisian duten eragina aztertuko da, eta 3.6 atalean, gure egokitzapena egun MaltParser sistema era automatikoan konfiguratzeko erabiltzen den MaltOptimizer tresnarekin lortutakoarekin konparatuko da. Azkenik, 3.7 atalean, ezaugarri automatikoek analisi sintaktikoan duten eragina aztertuko da. Zuhaitz-bankuan dauden ezaugarriak hizkuntzalari talde batek eskuz gainbegiratutako ezaugarriak dira. Era automatikoan lortutako ezaugarriak erabilita, benetako egoera batean analizatzaile sintaktikoak izan dezakeen zehaztasuna neurtuko da.

3.1 Oinarrizko baliabideak

Euskararako analizatzaile sintaktiko estatistikoa erabiltzeko, ezinbestekoa da informazio morfosintaktikoa lortzea. IXA taldean garatutako analisi-kateko modulu bakoitzak aurreko moduluak eskaintzen dion informazioa erabiltzen du sarrera moduan, eta jasotako analisia informazio linguistiko berriarekin aberasten du. Hizkuntza-ezagutzari dagokionez, egitura modular honek sin-

taxi-analisia sakontasun ezberdinarekin egitea ahalbidetzen du geruza bakoitzean. Gainera, modulartasunak hizkuntza-datuak mantentzen laguntzen du eta sistema erabilgarria eta erraz moldatzeko modukoa egiten du. Sintaxi-analisia urratsez urrats egiten da eta erabiltzailearen esku geratzen da erabili nahi duen hizkuntza-ezagutza mailaren aukeraketa. Ezeiza-ren (2002) eta Oronoz-en (2008) tesi-lanetan gai hauei buruzko informazio zabala aurki daiteke. Hori dela-eta, 3.1 irudian agertzen diren analisi-kateko moduluak eta haien analisi-geruzak gainetik ikusiko dira orain.

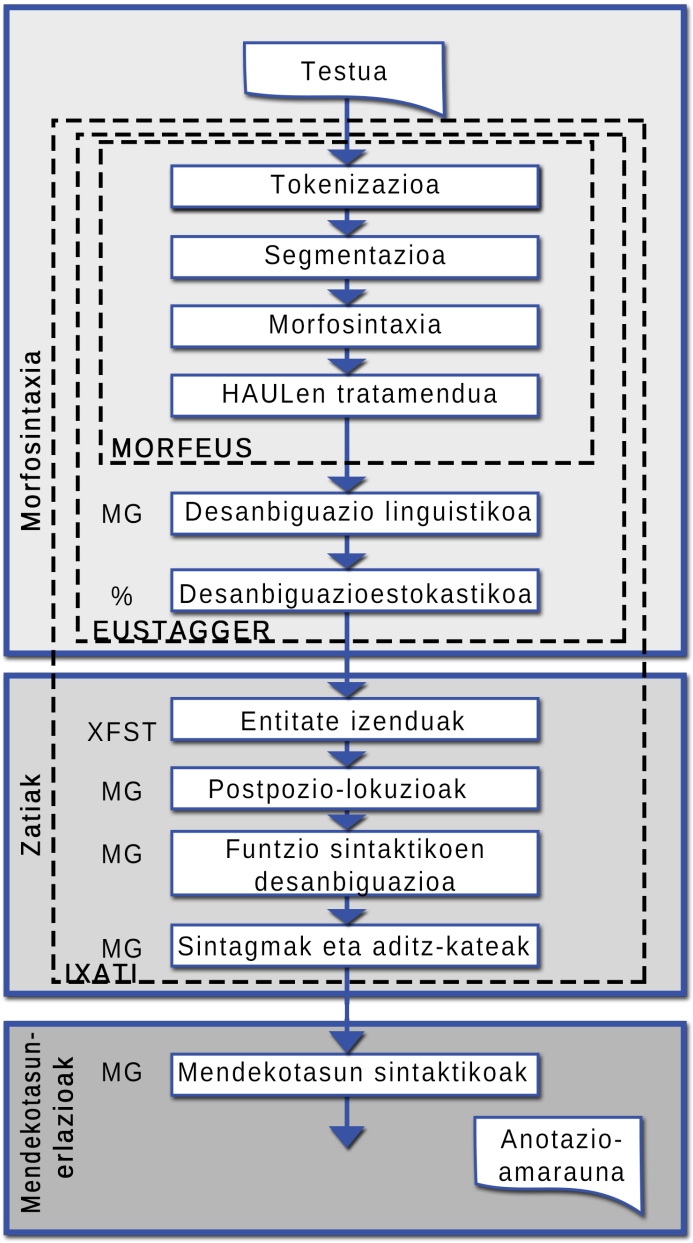
3.1.1 MORFEUS analizatzaile morfosintaktikoa

Analisi-prozesua MORFEUS (Aduriz *et al.*, 1998) analizatzaile morfosintaktikoarekin hasten da. MORFEUSek sarrerako testua jaso eta tokenizazioa, analisi morfologikoa, morfosintaxia eta hitz anitzeko unitate lexikalaren atazak gauzatzen ditu.

Tokenizazioa
Tokenizatzaileak sarreran testu gordina jaso eta testua hitzetan banatzen du. Banatutako token edo unitate bakoitzak, hitza, zenbaki arrunta, zenbaki erromatarra, deklinatu gabea, deklinatua, laburdura, sigla, zuriunea edo puntuazio-marka den detektatuko du, eta dagozkion ezaugarriak gehituko dizkio hitz bakoitzari.

Analisi morfologikoa
Segmentatzaileak testu-hitz bakoitza lema eta morfemetan banatzen du, interpretazio posible guztiak identifikatuz. Adibidez, izen eta adjektiboen kasuan, hitz-formaren kategoria, azpikategoria, kasu-marka, numeroa eta mugatasun ezaugarriak antzematen dira eta, aditzaren kasuan, aldiz, modua, denbora eta aspektua detektatzen dira. Hizkuntza-informazioa Euskararen Datu Base Lexikaletik (EDBL) (Aldezabal *et al.*, 2001) jasotzen du.

Morfosintaxia
Ezeiza-ren (2002) hitzen arabera, "segmentazioa gainditu eta formaren barruko informazioa elaboratu behar da, hitzaren egituraren berri emateko eta horixe da analisi morfosintaktikoaren zeregina". Hitzaren egitura deskribatzeko, testuingururik gabeko gramatika bat erabili da (Gojenola, 2000; Aduriz, 2000), PATR formalismoaren inplementazio bat aukeratuta (Shieber,



3.1 irudia – Oronoz-en (2008) tesi-lanetik ateratako irudia, euskararako geruza anitzeko sintaxi partzialeko analizatzailea.

1986), hitzen barruko analisi morfosintaktikoa egin da; hau da, morfemetatik lortutako informazioa konbinatu eta hitz-formaren interpretazio bakoitzeko ezaugarri-egitura bat lortzen da.

Hitz Anitzeko Unitate Lexikalen (HAUL) tratamendua
Testuan Hitz Anitzeko Unitate Lexikal (HAUL) asko daude, "hala eta guztiz ere" bezalakoak, esaterako. HAUL bilatzaileak (HABIL) (Urizar, 2012) hitz-konbinazio eta hitz anitzeko unitate lexikal horiek identifikatzen eta tratatzen ditu.

3.1.2 EUSTAGGER lematizatzailea/etiketatzailea

Sarreran MORFEUSek emandako analisi morfosintaktiko posible guztiak izanik, EUSTAGGERek, aukera guztien artean eta testuinguruko informazioari begiratuta, onartezinak diren interpretazioak baztertu behar ditu, eta egokia baino ez dena mantenduko du.

Ezeiza-ren (2002) lanean, hiru anbiguotasun morfosintaktiko mota bereizten dira: kategoriarena, morfema ez-askearena eta sintaxiarena. Kategoria soilik kontuan hartuz gero, hitz-forma bakoitzeko, batez beste, 1,55 aukera izan ditzakegu. Baina informazio morfosintaktiko guztia kontuan hartuz gero, batezbestekoa 2,65era iristen da. Hitz-forma baten desanbiguazio morfologikoaren ataza konplexua gauzatzeko, IXA taldean desanbiguazio morfologikoaren modulua (Ezeiza *et al.*, 1998) bi mailatan konbinatu da:

- Lehenengo mailan, hizkuntza-ezagutzan oinarritutako desanbiguazioa egiten da. Murriztapen Gramatika formalismoa (Karlsson *et al.*, 1995) erabiltzen da ezagutza jasotzeko. Desanbiguazio linguistikoaren berri Aduriz-en (2000) tesi-lanean ematen da.
- Bigarren mailan, estatistika (corpusetan oinarritutako teknika) erabiltzen da, lehen mailako Markov-en eredu (HMM) ezkutua. Moduluan (Ezeiza, 2002) corpus handi batetik automatikoki erauzitako estatistikak erabiltzen dira, hitzari dagozkion interpretazioen artean desanbiguazioa gauzatzeko.

Hainbat esperimenturen emaitzak aztertuta, desanbiguaziorako hurbilpenik egokiena metodoen konbinazioa dela ikusi da. Horrela, metodo konbinatua erabiliz % 97 baino gehiagotan aukera zuzena itzultzen du. Baina modulu honen irteerak hitz-forma bakoitzeko 1,3 aukera ematen ditu. Erabiltzen diren analizatzaile sintaktikoek aukera bakarra behar dutenez, desanbiguazio moduluak ematen dituen aukeretatik, lehenengo aukera (sarriena) hartzea

erabaki da. Anbiguotasun morfologikoa oso altua da euskaraz ingelesarekin alderatuz gero, eta analizatzaile sintaktikoari (Bengoetxea *et al.*, 2011) kaltegarri gerta dakioke, analisi zuzena aukeratzen ez denean.

3.1.3 IXATI zatitzailea edo *chunker*ra

IXATI zatitzaileak esaldiak *chunk* edo kateetan banatzea du helburu (Aduriz *et al.*, 2004). Katea sintagma kategoriako zatia da eta sintaktikoki erlazionatutako hitzez osatuta dago. Kate bezala, sintagmak eta aditz-kateak, postposizio-lokuzioak[1] eta entitate izendunak[2] banatuko ditu.

3.2 Sintaxiaren tratamendu automatikoa

IXA taldearen helburua sintaxirako tresna erabilgarriak lortzea da, testu errealetarako balio behar dutenak. Tresna horiek garatzeko hurrengo bi hurbilpen nagusi bereiz ditzakegu:

- Hizkuntza-ezagutzan oinarritutako hurbilpenak: Gojenola-ren (2000), Aduriz-en (2000) eta Arriola-ren (2000) tesi-lanetan aukera guztiak aztertu ostean, testuingururik gabeko gramatika eta egoera finituko mekanismoak erabiltzeari ekin zioten. Horrenbestez, baterakuntzan oinarritutako testuingururik gabeko gramatika erabiltzen duen PATR-IXA analizatzaile partziala sortu da eta Murriztapen Gramatikan oinarritzen den Euskararako Dependentzia Gramatika Konputazionala (EDGK) gramatika (Aranzabe, 2008) sortu da, eta gramatika hori baliatuta garatu da analizatzaile sintaktikoa.
- Datuetan oinarritutako hurbilpenak: azken urteotan, ikasketa automatikoan oinarritutako analizatzaile sintaktiko estatistikoak asko ugaritu dira. Dependentzia-sintaxian eta datuetan oinarritutako 2006ko eta 2007ko *CoNLL Shared Task* zereginen ondoren, grafoetan (Eisner, 1996, 2000; McDonald eta Pereira, 2006) eta trantsizioetan (Nivre, 2003; Yamada eta Matsumoto, 2003; Titov eta Henderson, 2007) oinarritutako hurbilpenak nagusitu ziren. Tesi-lan honetan, 2006ko *CoNLL* zereginen ondoren nagusitu ziren hurbilpen bietan (hots, trantsizio eta

[1] Perpausaren sintagmen arteko erlazio gramatikalak adierazten dituzten forma askeak dira.

[2] Entitate-izenak (pertsonak, tokiak eta erakundeak), denbora-adierazpenak (datak eta orduak) eta zenbakizko zenbait espresio (ehunekoak, diru-balioak, eta abar.) dira.

grafoetan) oinarritutako sistema onenen egokitzapena egin da, hau da, MaltParser (Nivre *et al.*, 2006b) eta MSTParser (McDonald *et al.*, 2005; McDonald eta Pereira, 2006) izenekoena, hurrenez hurren.

Ondorengo ataletan, IXA taldean euskal sintaxiaren tratamendu automatikoan egindako lan desberdinak aztertuko dira.

3.2.1 Hizkuntza-ezagutzan oinarritutako hurbilpenak

IXA taldean, hizkuntza-ezagutzan oinarritutako hurbilpena jarraituz, euskal sintaxiaren tratamendu automatikorako ondorengo bideak landu dira:

Testuingururik Gabeko Gramatika ereduan oinarritutako analizatzaileak

Baterakuntzan oinarritutako gramatikak (Shieber, 1986), funtsean, testuingururik gabeko gramatikak dira. Alabaina, formalismo sintaktiko horri gehitzen zaizkio, batetik, osagai sintaktikoen informazioa biltzeko aukera ezaugarri-egituren bidez, eta bestetik, erregela bakoitzari zenbait ekuazio definitu ahal izatea komunztadurak edo beste edozein murriztapen edo erlazio sintaktiko egiaztatzeko.

IXA taldean, PATR formalismoaren inplementazio bat aukeratu zen (Shieber, 1986). PATR formalismoa aukeratu genuen euskararen deskribapena egiteko, arrazoi nagusi bi kontuan izanda: lehenengoa, analizatzaile konputazionala egiteko oinarrian dagoen EDBL datu-base lexikala erabili behar da, eta datu-base honek ez du LFG edo HPSG moduko formalismo batek beharko lukeen informazio konplexu osoa. Bigarrena, PATR formalismoa malgua eta sinplea da, eta horregatik errazagoa izan da euskararen lehen tratamendua egiteko, ondorengo lanetan HPSG edo LFGren inplementazioak baztertu gabe. Esan denez, testu errealak tratatzeko analizatzaile bat eraikitzea izan da tesi-lan honen helburua eta, beraz, Abaitua-ren (1988) bezalako lanetan landu den gramatika konputazional sakona baino, nahiago izan da maizen gertatzen diren osagai sintaktikoak ezagutzea; horretarako, bada, PATR formalismoa da egokiena.

PATR-IXArekin garatutako gramatikarekin analisi partziala lortu da. Testu errealeko izen-sintagmen eta adizlagunen analisia lor daiteke. Zer analiza dezake gramatika honek? Zein da bere estaldura? Maila lexikoan oso estaldura handia dagoenez (EDBL datu-base lexikaleko 70.000 sarrerak erabiltzen dira analisian), esan daiteke testu errealetako ia izen-sintagma eta

adizlagun guztiak analiza daitezkeela. Berdina esan dezakegu esaldi sinpleen kasuan, hau da, esaldian puntuazio-markarik gabe honelako elementuen sekuentzia agertzen bada: aditza, mendeko perpaus erlatiboak, mendeko perpaus konpletiboak, mendeko moduzko perpausak, mendeko denborazko perpausak eta zehar-galderak. Gramatika honen deskribapena Gojenola-ren (2000) eta Aldezabal *et al.*-en (2003) lanetan egin da.

Egoera finituko teknikak (Karttunen *et al.*, 1997) azaleko sintaxia egiteko erabiltzen dira, hau da, perpausetako osagaiak bereizteko erabiltzen dira. Espresio erregularrak baliatzen dituzte osagaien eraketa definitzeko eta oso denbora txikian lortzen dituzte emaitzak. Euskararen tratamendua egiteko, formalismo bi aztertu dira: Murriztapen Gramatika (MG) eta *Xerox Finite State Tool* (XFST). *Xerox Finite State Tool* (XFST) tresnaren xehetasunak Gojenola-ren (2000) tesi-lanean daude eta MGrenak Aduriz-en (2000) eta Arriola-ren (2000) tesi-lanetan.

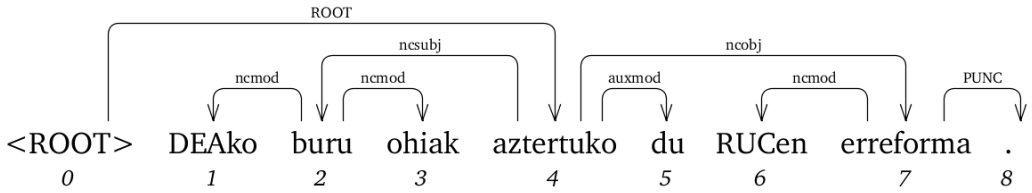
EDGK sintaxi-analizatzaileak Murriztapen Gramatika (MG) formalismoa jarraitzen duenez, egoera finituko tekniken barruan kokatu beharko litzateke, baina bere ezaugarri bereziak direla-eta, aipamen berezia egin zaio. EDGKak IXATI zatitzailearen irteera jasotzen du eta dependentzia-erlazioak bi urratsetan ezartzen ditu, Aranzabe-ren (2008) tesi-lanean azaltzen denez:

- EDGK-I gramatikarekin egindako *Constraint Grammar Parser CG-2* (Tapanainen, 1996) analizatzaileak hitz bakoitzari dependentzia-etiketa bat esleitzen dio. Sintaxi-erlazioaren izena jartzeaz gain, "$>$" edo "$<$" ikurraren bidez, gobernatzailea zein noranzkotan dagoen adierazten da.
- EDGK-II gramatika aplikatuta, eta, BURUBIL programarekin, buruko eta mendeko loturak egin dira.

Adibide baten bidez ikusiko da dependentzia-erlazioak nola ezar daitezkeen. Har bedi "DEAko buru ohiak aztertuko du RUCen erreforma." esaldia. Esaldi honi 3.1 taulan adierazitako dependentzia-etiketak ipiniko lizkioke EDGK-I gramatikak.

Hitzak	DEAko	buru	ohiak	aztertuko	du	RUCen	erreforma
Dependentzia-etik.	NCMOD>	NCOBJ>	<NCMOD	ADITZ_NAGUSI	<AUXMOD	NCMOD>	<NCOBJ

3.1 taula – Euskarako zuhaitz-bankuko esaldi bat, EDGK-I dependentzia--etiketak ipinita.



3.2 irudia – "DEAko buru ohiak aztertuko du RUCen erreforma." esaldiaren dependentzia-zuhaitza

3.1 taulan adierazten den informaziotik abiatuta, 3.2 irudiko dependentzia-zuhaitza lortu beharko litzateke. Adibidean, "ohiak" hitzari "`<NCMOD`" etiketa ipinita, hitz horrek gobernatzaile duen "buru" hitzarekiko, "ncmod" perpausa ez den modifikatzailea dependentzia-erlazioa duela adieraziko litzateke. Hori beharrean, erlazioa eta gobernatzailearen noranzkoa soilik adierazten da, eta ez da adierazten gobernatzailea zein den. Hori EDGK-II gramatikak egiten du elementuen kategoria gramatikalak erabiliz. Normalean, teknika hauek doitasun ona eskaintzen dute, baina badaude muga nagusi batzuk ere, bakan batzuk aipatzearren:

- Lan handia eskatzen du erregelak idatzi eta mantentzeak.
- Etiketak aldatuz gero, erregelak aldatu egin behar dira.
- Estaldura baxua izaten da testu errealak tratatzerakoan.

3.2.2 Datuetan oinarritutako hurbilpenak

Tesi-lan honen helburu nagusia estaldura zabala izango duen euskararako analizatzaile sintaktiko bat lortzea izango denez, euskarako zuhaitz-bankuan (Aranzabe, 2008) oinarritutako euskararako analizatzaile sintaktiko estatistikoak erabiliko dira.

Dependentzia-datuetan oinarritutako analizatzaile sintaktikoen sortzaileak

2006an eta 2007an, hain zuzen ere, tesi-lan honen hasieran, dependentzietan oinarritutako sistemak eta zuhaitz-bankuak garatzeko *CoNLL Shared Task* lehiaketak egin ziren. 2007ko lehiaketan erabili ziren zuhaitz-bankuetatik, EZB I zuhaitz-bankua aukeratu zen. Lehiaketa hauetan parte hartu zuten sistema onenak aintzat hartuta, trantsizioetan oinarritutako sistemetatik MaltParser (Nivre *et al.*, 2004) izenekoa aukeratu genuen eta grafoetan

oinarritutako sistemetatik, aldiz, MSTParser (McDonald *et al.*, 2005) aukeratu genuen esperimentuak egiteko. Bi sistema hauek aukeratzeko arrazoi nagusiak hurrengo hauek izan ziren:

- *Open-source* banaketa librea izatea
- 2006ko *CoNLL-X Shared Task Multilingual Dependency Parsing* lehiaketan (Buchholz eta Marsi, 2006) 13 hizkuntza eta 18 sistemen artean onenak izatea
- Analizatzaileak zuhaitz proiektiboak eta ez-proiektiboak sortzeko aukera izatea
- Hizkuntza ezberdinetan probatzeko aukera eskaintzea

Hurrengo ataletan, MaltParser eta MSTParser sistemen deskribapen zabala egiten da.

3.2.2.1 MaltParser. Trantsizioetan oinarritutako sistema

MaltParser[3] (Nivre, 2006; Nivre *et al.*, 2007b; Hall *et al.*, 2007) trantsizioetan oinarritutako analizatzaile sintaktikoen sortzailea da (ikusi 2.3.3.2 azpiatala). MaltParser sistemarekin sortutako modeloak trantsizio-segida amaitu ostean, dependentzia-zuhaitza sortzeko gai dira. MaltParser modeloek dependentzia-arkuak ikasi beharrean, makina abstraktu baten trantsizioak ikasten dituzte. Horiek ikasteko, dependentzietan etiketatutako eta *CoNLL-X* formatuan dagoen zuhaitz-banku bat izatea beharrezkoa da. MaltParser sistemak hitzaren analisi morfosintaktiko bakarra behar du, hau da, hitz sarrerak guztiz desanbiguatuta egon behar du. MaltParser sistemarekin analizatzaile sintaktiko bat sortzeko, erabakiak hiru mailatan hartu behar dira: algoritmo, ezaugarri-modelo eta ikasketa automatikorako erabiliko den sailkatzaile mailan.

Analisi sintaktikorako algoritmoak

Trantsizioetan oinarritutako sistemek algoritmo ugari erabiltzeko aukera eskaintzen dute. Gehien erabiltzen diren algoritmoak *shift-reduce* algoritmoaren aldaerak dira. *Shift-reduce* algoritmoek analizatzailearen uneko egoera gordetzeko hurrengo egiturak erabiltzen dituzte:

- Pila: partzialki tratatutako hitzak
- *Buffer*a: oraindik tratatu gabe dauden esaldiko hitzak
- Dependentzia-arkuen multzoa: uneko egoerara heldu baino lehen analizatzaileak hartutako erabakiekin eratutako zuhaitz partziala

[3] http://maltparser.org

Analisi sintaktikoa urrats batean gauzatzen da; trantsizio egokia zein den jakiteko kontuan hartzen ditu analizatzailearen historia (uneko egoerara heldu baino lehen, analizatzaileak hartutako erabakiekin eraikitako dependentzia-arkuen multzoa eta pila) eta uneko sarreraren informazioa (*buffer*era). Trantsizioak pilaren eta *buffer*aren artean ematen dira. MaltParser sistemak trantsizioetan oinarritutako *shift-reduce* algoritmoaren hainbat aldaera eskaintzen ditu, Nivre algoritmoaren izenarekin ezagutzen direnak. Baina horrez gain, bestelako hiru familia nagusitan banatzen diren algoritmoak ere eskaintzen ditu: Covington algoritmoak, Pila (*Stack*) algoritmoak eta Multiplanar algoritmoak. Familia bakoitzean, bi algoritmo edo hiru algoritmo aurki ditzakegu:

- Nivre algoritmoen familian (Nivre, 2003; Nivre *et al.*, 2004), denbora linealean exekutatzen diren bi algoritmo proiektibo daude: Nivre Arku Azkarra (ingelesez, *arc-eager*) edo Nivre Arku Estandarra (ingelesez, *arc-standard*). Bien arteko alde nagusia hurrengoa da: Nivre Arku Azkarra algoritmoak ezkerreko arkua (ingelesez, *left-arc*) ahal duen bezain pronto aukeratzen du. Adibidez, 3.2 taulan Nivre Arku Azkarra algoritmoa aukeratu ostean "Kepa etorri da" esaldiaren trantsizio-sekuentzia ikus daiteke. Nivre familiako algoritmoek esaldiak ezkerretik eskuinera prozesatzeko, algoritmo jale eta determinista bat erabiltzen dute. Nivre algoritmoek analisi proiektiboa gauzatzen dute eta zuhaitz ez-proiektiboekin lan eginez gero, teknika osagarriak erabiltzeko aukera dago.
- Pila algoritmoen familian (Nivre, 2009; Nivre *et al.*, 2009), algoritmo proiektibo bat: Pila Proiektiboa (ingelesez, *Stackprojective*), eta algoritmo ez-proiektibo bi: Pila Azkarra (ingelesez, *Stackeager*) eta Pila Nagia (ingelesez, *Stacklazy*) aurki daiteke. Algoritmo hauen analisi ordena Nivre Arku Estandarraren berdina da.
- Covington algoritmoen familian (Covington, 2001), denbora koadratikoan exekutatzen diren algoritmo ez-proiektiboa edo proiektiboen artean aukera daiteke.
- Multiplanar algoritmoen familian (Gómez-Rodríguez eta Nivre, 2010) denbora linealean exekutatzen diren Planar algoritmo proiektiboa eta 2-Planar algoritmo ez-proiektiboen artean aukera daiteke.

Zuhaitz-banku ez-proiektiboekin, hiru aukera dira:

- Algoritmo ez-proiektiboak erabiltzea; hau da, arku proiektiboa eta ez-proiektiboekin lan egiten dutenak. Oro har, algoritmo ez-proiektiboak konplexuagoak eta konputazionalki neketsuagoak dira. Adibidez, Co-

vington algoritmo ez-proiektiboa koadratikoa da, eta Nivre-ren algoritmo proiektiboa, berriz, lineala da.
- Zuhaitz-banku ez-proiektiboa proiektibotzat jotzea eta algoritmo proiektiboak (adibidez, trantsizioetan oinarritutako Nivre-ren algoritmo bakun, efiziente eta zehatza) aplikatzea.
- Algoritmo proiektiboak aplikatzea, baina teknika osagarriak aplikatuz. Teknika osagarri ugari daude. Adibidez: transformazio pseudo-proiektiboa (ingelesez, *pseudo-projective*) (Nivre eta Nilsson, 2005), 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 sarreran dauden elementuen ezaugarriak erabiliko dira. Ezaugarri-modeloa dimentsio handiko ezaugarri bektore bat izango da. Bektorea ezaugarri sinplez osatuta dago. Ezaugarri sinple bakoitza definitzeko bi funtzio erabiltzen ditu, helbide-funtzioa eta egozpen-funtzioa:

- Helbide funtzioa algoritmoetan erabiltzen diren egituretatik edozein hitzetara heltzeko erabiltzen da, hau da, algoritmo gehienetan, pilan, *buffer*ean eta orain arte analizatzaileak hartutako erabakiekin eratutako zuhaitz partzialean dauden hitzetara heltzeko. Adibide batzuk aipatzearren: pilako edo *buffer*eko n. hitza, pilako edo *buffer*eko hitz baten gurasoa, pilako edo *buffer*eko ezkerreko edo eskuineko senidea, eta abar.
- Egozpen funtzioa hitzaren informazio morfosintaktikoa lortzeko erabiltzen da, *CoNLL-X* formatuan dauden ezaugarriak lortzeko, adibidez: kategoria, forma, lema, kasua, numeroa, dependentzia-arkuen etiketa, eta abar.

Bi funtzio hauekin pilaren gailurrean dagoen hitzaren kategoria bezalako ezaugarri sinpleak definituko dira. MaltParser sistemaren konfigurazioan[4] hizkuntza batek dituen ezaugarri garrantzitsuenak ezaugarri-modeloan gehitzea da lan neketsuena eta garrantzitsuena, sailkatzaileak hurrengo trantsizioa zein den asmatzeko erabiliko baitu.

MaltParser sistemarekin datorren ezaugarri-modeloaren balio lehenetsiak hurrengo ezaugarri-taldeak biltzen ditu:

[4] Ikus http://maltparser.org/userguide.html

- Pila eta *buffer*ean dauden hitzen (ohikoena 6 hitzeko luzera duen leiho zabala) kategoria gramatikala.
- Pila eta *buffer*ean dauden hitzen (ohikoena 3 hitzeko luzera duen leiho zabala) forma.
- Pila eta *buffer*ean dauden hitzen (ohikoena 4 hitzeko luzera duen leiho zabala) gurasoarekin daukan dependentzia-etiketa.
- Ezaugarrien arteko ohiko konbinaketak: kategoria gramatikalen n-grama eta kategoria gramatikala eta hitz-formen arteko bikoteak.

Ikasketa automatikorako sailkatzailea

Sailkatzailearen lana x sarrerako datuari dagokion y klasea iragartzea da. Trantsizioetan oinarritutako algoritmoak pauso bakoitzean, aukera anitzen artean trantsizio onena zein den erabakitzeko orduan, sailkatzaileari galdetuko dio. Sailkatzailearen x sarrera analizatzailearen historia edo ezaugarri--modeloa izango da; eta y irteera-klasea sailkatzaileak erabakitako trantsizioa izango da. Sailkatzaileak x ezaugarri-modelo bakoitzarekin puntuazio bat ematen dio y klase bakoitzari, eta puntuazio altuena duen y klasea izango da sailkatzailearen erabakia edo irteera. Sailkatzaileak ikasteko emaitza zuzenak dituen zuhaitz-banku bat erabiliko du. MaltParser 1.4 bertsio sistemak ikasketa automatikoa gauzatzeko, euskarri bektoredun makinak (ingelesez, *Support Vector Machine, SVM*)(Cortes eta Vapnik, 1995) paketea erabiltzen du. Ikasketa gainbegiratua egiteko, Vapnik-ek bitarra eta lineala bezala erabiltzeko sortu zuen *SVM*. Sarrerako datuak bi klase eremutan banatzeko marjina handiena duen hiperplanoa aukeratzen du. Geroxeago, *SVM*ak arazo ez-linealak ebazteko, espazioaren egokitzapena egiten duten kernel trikimailu-funtzio batzuk asmatu ziren. Gaur egun, kernel funtzio erabilienak honako hauek dira:

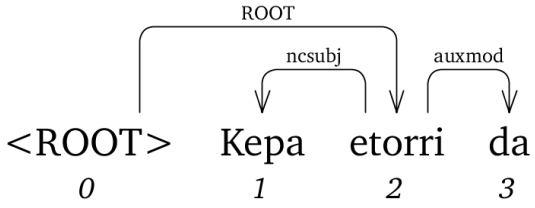
- Lineala
- Polinomiala
- *Radial Basis Function (RBF)*
- Sigmoidea

*SVM*aren eraginkortasuna kernel funtzioaren parametroen eta C penalizazio parametroaren aukeraketan dago. MaltParser 1.4 bertsio sistemak *SVM* sailkatzailea aukeratzeko orduan, bi pakete ezberdin eskaintzen ditu, LibSVM (Chang eta Lin, 2001) eta LibLINEAR (Fan *et al.*, 2008) paketeak:

- LibSVM[5] paketeak hurrengo kernel ezberdinak eskaintzen ditu: *RBF*,

[5] http://www.csie.ntu.edu.tw/~cjlin/LibSVM/

polinomiala eta lineala.
– LibLINEAR[6] paketeak sailkatzaile lineala erabiltzeko aukera baino ez du ematen. Baina ebazpen aukera ezberdinak (ingelesez, *solver type*) (Fan *et al.*, 2008) eskaintzen ditu: *SVM* lineala, erregresio logistikoa eta klase anitzeko sailkatzailea (Crammer eta Singer, 2002).



3.3 irudia – "Kepa etorri da" esaldiaren dependentzia-zuhaitza.

Tran.	Pila	*Buffer*a	Dependentzia-burua	Dependentzia-etiketa
-	hutsik	root Kepa etorri da		
Shift	root	Kepa etorri da		
Shift	root Kepa	etorri da		
L-arc	root	etorri da	[Kepa ← etorri]	ncsubj
R-arc	root	etorri da	[etorri ← root]	ROOT
Shift	root etorri	da		
R-arc	root etorri	da	[da ← etorri]	auxmod
Shift	root etorri da	hutsik		

3.2 taula – 3.3 esaldiaren Nivre-ren Arku Azkarra algoritmoaren trantsizio-segida (Tran. = Trantsizioak, L-arc = Left-arc eta R-arc = Right-arc)

Adibidez, *"Kepa etorri da"* esaldiaren dependentzia-zuhaitza 3.3 irudian ikus daiteke. Esaldi honen zuhaitza eraikitzeko *shift-reduce* algoritmoarekin trantsizio-segida nola gauzatzen den 3.2 taulan ikusten da.

1. zutabean gauzatzen diren trantsizioak agertzen dira, EZB II zuhaitz-bankuaren kasuan 30 dependentzia-etiketa desberdin daude eta Nivreen Arku Azkarra algoritmoa erabilita, algoritmoak une oro 61 trantsizioren artean erabaki beharko du:
   – *Shift*: *buffer*eko hitza pilaren gailurrean jartzea
   – *Left-arc* (30 dependentzia-etiketa posible): dependentzia-arkuen multzoan, *buffer*aren uneko eta pila gailurreko hitza lotuko dira, eta pila buruan dagoen hitza pilatik atera egiten da.

[6] http://www.csie.ntu.edu.tw/~cjlin/LibLINEAR/

- *Right-arc* (30 dependentzia-etiketa posible): dependentzia-arkuen multzoan, pila gailurreko hitza, *buffer* uneko hitzarekin lotuko da.

2. zutabean, pila (partzialki tratatutako hitzekin)

3. zutabean, *buffer*a (oraindik tratatu gabe dauden esaldiko hitzak).

4. zutabean, dependentzia-arkuen multzoan, pilaren gailurrean eta *buffer*aren unean dauden hitzen artean analizatzaileak erabaki duen buru-mende erlazioa agertzen da. Zutabe horretan hartutako erabakiekin eratutako zuhaitz partziala egongo da.

5. zutabean, dependentzia-arkuen multzoan, analizatzaileak 30 dependentzia-etiketa posibleetatik zein aukeratu duen esaten da.

3.2.2.2 MSTParser. Grafoetan oinarritutako sistema.

MSTParser grafoetan (ikusi 2.3.3.2 atala) oinarritutako analizatzaile sintaktikoen sortzailea da. Tesi-lan honetan MSTParser 0.5[7] bertsioa erabili da. MSTParser sistemak analisi sintaktikoa bi urratsetan gauzatzen du. Lehen urratsean, buru egokia zein den aukeratzen du eta, bigarren urratsean, dependentzia-etiketak ezartzen ditu zuhaitz-arkuetan.

MSTParser erabiltzeko, *CoNLL-X* formatuan dependentzietan etiketatutako zuhaitz-banku bat izatea beharrezkoa da. MSTParser sistemak hitzaren analisi morfosintaktiko bakarra behar du, hau da, sarrera guztiz desanbiguatua egon behar da.

MSTParser sistemaren deskribapena egiterakoan, hurrengo hiru ezaugarri nagusi erabiliko dira: algoritmoa, ezaugarri-modeloa eta ikasketa automatikoa.

Algoritmoak

MSTParser 0.5 bertsioan, bigarren edo lehen mailako modeloen artean eta arku proiektiboekin edo ez-proiektiboekin lan egiteko aukera eskaintzen da. Zuhaitza osatzen duten arkuen puntuazioen faktorizazioa erabiltzen denean, lehen mailako modeloa edo puntuazio-funtzioa erabiltzen dela esango da. Baina arkuak independente bezala hartuz gero, hizkuntzetan dauden egitura konplexuen pisua era egokian ordezkatuta ez agertzeko arriskua dago. MSTParser sistemaren bigarren mailako algoritmoa erabiliz gero, guraso bereko bi

[7] http://sourceforge.net/projects/mstparser

elkarren ondoko umeen arkuen puntuazioa kontuan har daiteke. Ondoko dependentzia-egituren informazio guztia kontuan hartzea konputazionalki ezinezkoa denez, MSTParser sistemako bigarren mailako algoritmoek Markoven hurbilpena modu horizontalean aplikatzen dute. 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-en (1996) 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 (Chu eta Liu, 1965; Edmonds, 1967) algoritmoa eta Tarjan-en (1977) inplementazioa erabiltzen du. Algoritmo honek zuhaitz guztien gaineko bilaketa $O(n^2)$ denboran gauzatzen du, lehen edo bigarren mailako ezaugarriak erabiliz.

Ezaugarri-modeloa

MSTParser sistemak esaldia osatzen duten zuhaitz-arkuen parametroen batuketa maximizatzen duen *MST (Maximum Spanning Tree)* algoritmo globala erabiltzen du. MSTParser sistemarekin sortutako analizatzaile sintaktikoak puntuazio gehien dituzten arkuekin geratzen dira. Arkuen puntuazioa kalkulatzeko, arku mailan definitutako ezaugarrien multzo bat erabiltzen da. MSTParser 0.5 bertsioarekin lortutako arku-ezaugarri multzoen balio lehenetsien txantiloia ikus daiteke 3.3 taulan. MSTParser sistemarekin lehen mailako algoritmoarekin datozen arku bakarreko ezaugarriak (ikusi 3.3 taulako 1, 2 eta 3 ataletan dauden ezaugarriak) arkuaren norabidearekin eta distantziarekin konbinatuko dira. Lehenengo atalean, arku bakarra osatzen duten gurasoen eta umearen hitz-forma, kategoria, azpikategoria, informazio morfologikoa, eta abar agertzen dira. Bigarren atalean, guraso eta umeen artean dauden hitz guztien kategoria eta azpikategoria agertzen dira. Eta, hirugarren atalean, guraso eta umearen ondoko eskuineko eta ezkerreko hitzen kategoria eta azpikategoria agertzen dira. Laugarren atalean dauden ezaugarriak bigarren mailakoak dira, guraso bereko ondoko ezkerreko senide bat edukiz gero, arku horren informazioari gehituko zaio. Bertan, guraso eta guraso bereko ondoko senideen hitza eta azpikategoria senideen artean dagoen distantziarekin konbinatuko dira. Distantzia hitzen indizeen artean dagoen diferentzia izango da.

Atala	Erpina	Arku-ezaugarri multzoen balio lehenetsiak
1	Guraso hitza (P) Ume hitza (C) P eta C	Px, Py, Cx, Cy, Pxy, Cxy PxCx, PyCy, PxCy, PyCx PxyCy, PxCxy, PyCxy, PxyCxy non (x, y) ∈{(word, pos), (word, cpos), (lemma, pos) (lemma, cpos), (word, f_i)} (lemma, f_i)}
2	P eta C-ren bitartean dauden hitzak (B)	PxBxCx non x ∈ {pos, cpos}
3	P eta C-ren auzokoak, eskuinekoak (PR/CR) eta ezkerrekoak (PL/CL)	PLxPxCxCRx, PLxPxCx, PLxCxCRx, PLxPxCRx, PxCxCRx, PxPRxCLxCx, PxPRxCLx, PxPRxCx, PxCLxCx, PRxCLxCx, PLxPxCLxCx, PxPRxCxCRx non x ∈ {pos, cpos}
4	C-ren senideak (S)	CxSy, PzSzCz non (x, y) ∈ {(word, word),(pos, pos), (word, pos), (pos, word) } eta z ∈ {pos} Ezaugarri horiek ere norabide eta distantzia ezaugarriekin batera doaz

3.3 taula – Oinarrizko MSTParser sistemarekin datorren ezaugarrien txantiloia (*CPOS* = kategoria, *POS* = azpikategoria, F_i = *CoNLL-X* formatuan *FEATS* zutabean agertzen diren ezaugarri morfosintaktikoak).

Ikasketa automatikorako sailkatzailea

MSTParser sistemaren sailkatzaileak ezaugarrien pisua ikasteko, lineako inferentzian oinarritutako (*online inference-based*) *MIRA (Margin Infused Relaxed Algorithm)* (Crammer eta Singer, 2003; McDonald *et al.*, 2005) metodoa erabiltzen du. *MIRA* ikasteko algoritmo iteratibo bat da, entrenamenduko esaldiekin N iterazio egiten du, W bektoreak pisuaren balioa finkatu eta entrenamenduko datuak linealki banatzeko gai izan arte. Nahiz eta Nren balio lehenetsia 10 izan, bere balioa aldatu egin daiteke. Iterazio bakoitzean entrenamenduko instantzia (esaldi) guztiak tratatzen ditu. Pisu bektorean gutxiengo aldaketak egiten ditu, gutxiengo puntuazioen marjina mantentzen den bitartean.

3.3 MaltParser sistemaren egokitzapena

3.3.1 Problemaren zehaztapena

MaltParser sistemak dependentzia-arkuak ikasi beharrean, makina abstraktu baten trantsizioak ikasten ditu. EZB II zuhaitz-bankuan, guztira, 30

dependentzia-etiketa desberdin daude eta *shift-reduce* algoritmo sinple bat erabilita, algoritmoak, une oro, 62 trantsizioren artean erabaki beharko du:

- *Shift*: *buffer*eko hitza pilaren gailurrean jartzea.
- *Reduce*: pilaren gailurrean dagoen hitza pilatik ateratzea.
- *Left-arc* (30 dependentzia-etiketa posible): dependentzia-arkuen multzoan, *buffer*eko hitza eta pila gailurreko hitza lotzea.
- *Right-arc* (30 dependentzia-etiketa posible): dependentzia-arkuen multzoan, pila gailurreko hitza *buffer*eko hitzarekin lotzea.

Shift-reduce algoritmoak trantsizio askoren artean erabaki behar duenean, sailkatzaileak ezaugarri-modeloa erabiliko du trantsizio posibleen artean bat aukeratzeko. MaltParser sistemako balio lehenetsiak aukeratuz gero, emaitzak asko jaisten dira; adibidez, 2007ko *CoNLL Shared Task* lehiaketan ikusi zen MaltParser sistemarekin, LAS neurrian, balio lehenetsi eta doituen artean balio absolutuan ehuneko hiruko aldea zegoela (Hall *et al.*, 2007). MaltParser euskarara egokitzea ez da lan tribiala, erabakiak hiru mailatan hartu behar baitira, hain zuzen ere, algoritmoaren, ikasketa automatikoaren eta ezaugarri-modeloaren mailan.

3.3.2 Antzeko lanak

Azken urteotan, MaltParser sistemaren ezaugarrien bilaketa automatikoa gauzatzeko lan ugari egin dira (Nilsson eta Nugues, 2010). Gaur egun, MaltParserren optimizazioa egiteko, MaltOptimizer (Ballesteros eta Nivre, 2012) erabil daiteke, ezaugarrien aurreragoko aukeraketan oinarritutako algoritmo jale bat erabiliz ezaugarri-modelo egokia bilatzeko gai den tresna. MaltOptimizer entrenamenduko datuak aztertu ostean, MaltParser sistemaren parametroak egokitzen saiatzen da. MaltOptimizer sistemak hiru fasetan egiten du analisia:

- Lehenengo fasean, datuen analisia egiten du oinarrizko parametroak aukeratzeko.
- Bigarren fasean, algoritmo egokiena aurkitzen ahalegintzen da.
- Hirugarren fasean, modelo-ezaugarrien aukeraketa eta *LibLINEAR*eko hiperparametroen aukeraketa egiten du.

EZB I eta EZB II zuhaitz-bankuen egokitzapena egin genuenean, ez zegoen MaltOptimizer bezalako tresnarik. Hurrengo ataletan, Maltixa, MaltParser sistemaren euskararako egokitzapena (Bengoetxea eta Gojenola, 2007), nola gauzatu den azalduko da. Egokitzapena hiru fasetan egin da: lehenengo fasean, MaltParser sistemaren konfigurazio sendo baten bilaketa gauzatzen da;

bigarren fasean, ezaugarri morfologikoen antolaketa eta, bukatzeko, hirugarren fasean, sistema konbinatuak aplikatzeko asmoz (Hall *et al.*, 2007), gure ezaugarri-modeloa beste algoritmo familiekin probatu ostean, LibLINEAR paketeko funtzio linealetik LibSVM paketean datorren bigarren graduko *kernel* funtziorako esportazioa egin da.

3.3.3 Konfigurazio sendo baten bilaketa

Problemaren zehaztapena

MaltParser sistemaren oinarria (ingelesez, *baseline*) bilatzeko hurrengo urratsak jarraitu dira:

1. Ezaugarri-modeloa eta ikasketa automatikoaren parametroak konstante jarrita, algoritmo ezberdinekin probatu da.

2. Algoritmo onena finkatuta eta ikasketa automatikoaren parametroak konstante daudela, ezaugarri-modeloaren balioak aldatu dira sistema doitu arte. Urrats honetan, hobekuntza nabaria lortzeko aukera dago, eta denbora gehien daraman urratsa da. Urrats honetan, Hall eta Nivre-k (2005) aurkitutako ezaugarri-modelotik abiatu gara (ikusi 3.4 taulan). Hall eta Nivre-ren (2005) ezaugarri-modeloa hizkuntza ezberdinetan baliagarria dela probatu da. 3.4 taulan, ezaugarri bakoitza egozpen-funtzio eta helbide-funtzio bikote batekin adierazita dator. Bikotearen lehenengo elementua egozpen-funtzioa da, informazio morfosintaktikoa lortzeko erabiliko dena; ezaugarri-modelo honetan, hitzaren

Zk.	Egozpen, Helbide funtzioak
1	POSTAG, Stack[0]
2	POSTAG, Input[0]
3	POSTAG, Input[1]
4	POSTAG, Input[2]
5	POSTAG, Input[3]
6	POSTAG, Stack[1]
7	DEPREL, Stack[0]
8	DEPREL, ldep(Stack[0])
9	DEPREL, rdep(Stack[0])
10	DEPREL, ldep(Input[0])
11	FORM, Stack[0]
12	FORM, Input[0]
13	FORM, Input[1]
14	FORM, head(Stack[0])

3.4 taula – Hall eta Nivre-k (2005) aurkitutako ezaugarri-modeloa, hizkuntza ezberdinetan baliagarria dena.

forma (FORM), azpikategoria (POSTAG) eta dependentzia-arkuen etiketa (DEPREL) soilik erabiltzen dira. Bikotearen bigarren elementuan, gehien erabiltzen diren helbide-funtzioak daude:

- Input[0]-rekin *buffer*eko lehenengo posizioan dagoen hitza lortuko da.
- Input[1]-ekin *buffer*eko bigarren posizioan dagoen hitza lortuko da.
- Input[n]-rekin *buffer*eko $n+1$. posizioan dagoen hitza lortuko da.
- Stack[0]-rekin pilako gailurrean dagoen hitza lortuko da.
- Stack[1]-ekin pilako gailurraren (Stack[0]) azpian dagoen hitza lortuko da.
- Stack[n]-rekin pilako gailurraren (Stack[0]) azpian dagoen n. hitza lortuko da.

Algoritmoak orain arte hartutako erabakiekin sortutako zuhaitz partzialetatik informazioa ateratzeko, gehien erabiltzen diren helbide-funtzioak hurrengo hauek dira:

- Head(Stack[0]), pilako gailurrean dagoen hitzaren gobernatzailea (zuhaitz partzialean) lortuko da.
- Rdep(Stack[0]), pilako gailurrean dagoen eta hitzetik eskuinen (*rightmost*) dagoen dependentzia-etiketatik mendeko hitza lortuko da.
- Ldep(Stack[0]), pilako gailurrean dagoen eta hitzetik ezkerren (*leftmost*) dagoen dependentzia-etiketatik mendeko hitza lortuko da.

3.4 taularen 11. lerroan dagoen "*FORM*, Stack[0]" funtzio bikotea erabiliz, pilako gailurrean dagoen hitzaren hitz-forma lortzen da. Hizkuntza gehienetan, *FORM*, *POSTAG* eta *DEPREL* ezaugarriekin lan egiten da, baina kanpoan utziko lirateke euskaran oso garrantzitsuak diren *FEATS* (ezaugarri morfosintaktikoak) eta *LEMMA* (lema). Oinarri honekin bi estrategia landuko dira:

(a) Atzerako aukeraketa. Ezaugarri interesgarrienak dituen eredu batetik hasiko gara. Ezaugarri batek laguntzen duen jakiteko, proba bakoitza ezaugarri bat kenduz egiten da. Sortutako eredu murriztu berri guztietatik portaera hobea duenarekin gelditu gara, eta hau izango da aldi baterako eredu berria. Aldi baterako ereduari gainerako ezaugarriak berriro kenduz gero, portaera onena duenarekin gelditu gara. Prozesu hau bukatuko da sortutako eredu

murriztu guztietan jaitsiera esanguratsua gertatzen denean.

(b) Aurrerako aukeraketa. Prozesu honetan ezaugarri multzo batekin proba egiten da. Proba bakoitza ereduari ezaugarri berri bat gehituz egiten da. Multzoko ezaugarri guztiekin probatu ostean, hobekuntza gehien ematen duen ezaugarria ezaugarri-modelora gehituko da, eta sortutako eredu berriari gainerako ezaugarriak banan-banan gehituko dizkiogu. Gainerako ezaugarriekin probatu ostean ez denean hobekuntza esanguratsurik gertatzen izango da prozesuaren bukaera.

Euskarazko ezaugarri-modeloaren bilaketan, oinarri sendotik abiatuta (Hall eta Nivre, 2005), lehenengo atzerako aukeraketa estrategia aplikatu da eta, azkenik, aurrerako aukeraketa estrategia gauzatu da (konparaketa egiteko LAS neurria erabili da).

3. Algoritmoa eta ezaugarri-modelo onena finkatuta, ikasketa automatikoaren optimizazioa gauzatzeko, LibSVM paketeko bigarren graduko kernel polinomikoa eta LibLINEAR paketearekin datozen sailkatzaile linealak aztertu dira tesi-lan honetan. LibSVM paketeko bigarren graduko kernel polinomikoek, oro har, sailkatzaile linealek baino emaitza hobeak ematen dituzte. MaltParser sistemak, aldiz, EZB II zuhaitz-bankuarekin LibSVM paketeko bigarren graduko kernel polinomikoko sailkatzaile bat entrenatzeko 30 ordu behar ditu eta, LibLINEAR paketeko sailkatzaile lineal batek, berriz, 5 minutu[8] behar ditu. Hori dela-eta, esperimentu ugari egin behar direnean, LibSVM paketeko bigarren graduko kernel polinomikoen ikasketa fasea azkartzeko aukera bi erabili dira:

   – Ikasketa fasea azkartzeko, entrenamenduko datuak azpitalde txikiagoetan banatzeko teknika erabil daiteke (Yamada eta Matsumoto, 2003). Teknika honetan, entrenamenduko datuak azpitalde txikiagoetan banatzen dira, eta ikasketa denbora nabarmen gutxituz, zehaztasuna gutxitu gabe adibidez, pilako edo *buffer*eko azpikategorian balio berdina duten azpitaldeetako bakoitzak klase-anitzeko sailkatzaile batekin ikasten du. Azpitalde txiki ugari kentzeko helburuarekin, N zenbakitik beherako kategoria (POS)

[8] Linux sistema eragilea, Intel Core Duo (2,66 GHz-ko 2 prozesadore) eta 4 Gb-ko memoria duen makina batean.

berdineko taldeak multzo bakar batean biltzeko aukera ematen du adierazitako teknikak.
- Ezaugarri-modeloaren bilaketa egiteko, LibLINEAR erabil daiteke, probatuta baitago LibLINEAR paketearekin lortutako ezaugarri-modeloa, LibSVM paketeko bigarren graduko kernel polinomikora esportatu ostean hobekuntza mantentzen dela (Cassel, 2009).

2007an, hain zuzen, LibLINEAR paketea ez zegoenean, esperimentuak azkartzeko, azpitaldeetan banatzen ziren datuak. Alabaina, LibLINEAR paketea agertu denetik, esperimentuak egiteko pakete hau erabiltzen da. LibLINEAR paketearekin jarraitutako urratsak hauek dira:

- Balio lehenetsiak probatu dira.
- Ebazpen guztiak probatu dira.
- Sailkatzailearen emaitza hobetze aldera, ezaugarri-modeloko ezaugarri garrantzitsuenak binaka edo hirunaka elkartu dira.
- Sailkatzailearekin ezaugarri-modelo egokia bilatu ostean, LibSVM paketea bigarren graduko kernel funtziora esportatu da.

Esperimentuak

2007an antolatu zen *CoNLL Multilingual Dependency Parsing* lehiaketan, euskararako sortu zen EZB I zuhaitz-bankua izan zen erabili zen zuhaitz-bankuetako bat. Hori dela-eta, EZB I zuhaitz-bankuarekin egindako esperimentu guztietan *CoNLL* 2007 lehiaketan erabilitako hiru multzoak erabili dira: entrenamenduko azpimultzoa (45.000 hitzekoa) eta garapenerako eta ebaluatzeko azpimultzoak (5.000 hitzekoak). EZB I eta EZB II zuhaitz-bankuen egokitze-prozesuak garai desberdinetan egin dira, eta horien artean egondako aldeak deskribatuko dira ondoren:

- Bertsioa. EZB I zuhaitz-bankua erabilita egindako esperimentuak egiteko MaltParser 0.4 bertsioa erabili zen; EZB II zuhaitz-bankuko esperimentuak egiteko, berriz, MaltParser 1.4 bertsioa erabili da.
- Proba multzoak. EZB I zuhaitz-bankuko tamaina txiki samarra dela-eta, 10 geruzako balidazio gurutzatua (ingelesez, *10-fold cross-validation*) teknika erabili da. Balidazio gurutzatua teknika erabiliz gehiegi egokitzeko (ingelesez, *overfitting*) joera eragotzi da. EZB II zuhaitz-bankuko tamaina handiagoa dela-eta, euskarako zuhaitz-bankua hiru zati desberdinetan banatu da, entrenamendurako % 80, garapenerako % 10 eta ebaluaziorako (testa, ingelesez *test*) % 10. Esperimentu

guztiak garapenerako datuekin egin dira, eta sistematik onena ebaluaziorako azpimultzoarekin probatzeko utzi da.

- Ikasketa-fasea azkartzeko. EZB I zuhaitz-bankuko esperimentuetan ikasketa fasea azkartzeko entrenamenduko datuak azpitalde txikiagoetan banatzeko Yamada eta Matsumoto-ren (2003) teknika erabili da. EZB II zuhaitz-bankuko ezaugarri-modeloaren bilaketa egiteko aldiz LibLINEAR erabili da.
- Algoritmoak. EZB I zuhaitz-bankuan ezaugarri-modeloa eta ikasketa automatikoaren parametroak konstante jarrita, algoritmo ezberdinekin probatu ostean, Nivre Arku Azkarra algoritmoa aukeratu da. EZB II zuhaitz-bankuko emaitzarik onena aldiz, Nivre Arku Estandarra transformazio pseudo-proiektiboarekin lortu da.
- Ezaugarri-modeloa. EZB I zuhaitz-bankuko ezaugarri-modeloaren bilaketan, Hall eta Nivre-ren (2005) ezaugarri-modelotik (ikusi 3.5 taulan Φ_1 ezaugarri-modeloa) abiatuta, lehenengo atzerako aukeraketa estrategia aplikatu da eta, azkenik, aurrerako aukeraketa estrategian euskararentzat interesgarriak izan ahal diren beste 28 ezaugarriekin probatu ostean, 3.5 taulan dagoen Φ_2 ezaugarri-modeloa lortu da. Ildo beretik, EZB II zuhaitz-bankuko ezaugarri-modeloaren bilaketan, 3.5 taulan dagoen $\Phi 4$ ezaugarri-modeloa lortu da.
- Ikasketa automatikoa. EZB I zuhaitz-bankuko algoritmoa eta ezaugarri-modelo onena finkatuta, ikasketa automatikoaren optimizazioa gauzatzeko esperimentu honetan, LibSVMeko bigarren graduko kernel polinomiko eta penalizazio parametro desberdinekin probatu ostean, $C = 0,4$ penalizazioarekin emaitzarik onena lortu da. EZB II zuhaitz-bankuko ikasketa automatikoaren optimizazioa gauzatzeko esperimentu honetan, aldiz, LibLINEAR paketearekin datozen ebazpen guztiekin probatu da. Emaitzarik onena *Crammer eta Singer-en multiclass support vector classification* ebazpen moduarekin eta $C = 0,1$ penalizazioarekin lortu da.

Emaitzak

Analizatzailea ebaluatzeko, etiketa eta gobernatzaile zuzenaren esleipena neurtzen duen neurria, *Labeled Attachment Score, (LAS)* erabili da.

3.6 taulan Φ_1 ezaugarri-modelo sendoari 7 ezaugarri (informazio lexiko eta morfosintaktiko) berri gehitu ostean, EZB I zuhaitz-bankuko gure Φ_2 ezaugarri-modeloarekin, 10 geruzako balidazio gurutzatuko teknikarekin eta

Zk.	Egozpen, Helbide funtzioak	Φ_1	Φ_2	Φ_3	Φ_4
1	POSTAG,Stack[0]	+	+	+	+
2	POSTAG,Input[0]	+	+	+	+
3	POSTAG,Input[1]	+	+	+	+
4	POSTAG,Input[2]	+			+
5	POSTAG,Input[3]	+			+
6	POSTAG,Stack[1]	+	+	+	+
7	DEPREL,Stack[0]	+	+	+	+
8	DEPREL,ldep(Stack[0])	+	+		+
9	DEPREL,rdep(Stack[0])	+			+
10	DEPREL,ldep(Input[0])	+			+
11	FORM,Stack[0]	+	+	+	+
12	FORM,Input[0]	+		+	+
13	FORM,Input[1]	+			+
14	FORM,head(Stack[0])	+			+
15	POSTAG,Stack[2]				
16	LEMMA,Stack[0]		+	+	+
17	LEMMA,Input[0]		+	+	+
18	Split(FEATS,Stack[0]),|		+	+	+
19	Split(FEATS,Input[0]),|		+	+	+
20	POSTAG,succ(Stack[0])				
21	POSTAG,pred(Input[0])				
22	Split(FEATS, Input[2]),|				+
23	CPOSTAG,Stack[2]				
24	FORM,rdep(Stack[0])				
25	FORM,pred(Input[0])				
26	POSTAG,ldep(Stack[0])		+	+	+
27	POSTAG,ldep(Input[0])		+	+	+
28	LEMMA,Input[1]		+	+	
29	CPOSTAG,Stack[0]			+	+
30	CPOSTAG,ldep(Stack[0])			+	
31	CPOSTAG,Input[0]			+	+
32	CPOSTAG,Input[1]			+	
33	CPOSTAG,Input[2]			+	
34	DEPREL(lsib(rdep(Stack[0]))			+	
35	FEATS,rdep(Stack[0])),|			+	+
36	POSTAG,rdep(Stack[0])				+
37	POSTAG,rdep(Input[0])				+
38	Split(FEATS, Input[1]),|				+
39	POSTAG,prdesc(Stack[0])				+
40	POSTAG,prdesc(Input[0])				+
41	POSTAG,prdesc(Input[1])				+
42	POSTAG,prdesc(Input[2])				+

3.5 taula – Taulan lau ezaugarri-modelo agertzen dira: Φ_1 hizkuntza ezberdinetan erabilitako oinarri sendoa (Hall eta Nivre, 2005), Φ_2 EZB I zuhaitz-bankuarekin lortutako oinarria (Bengoetxea eta Gojenola, 2007), Φ_3 *CoNLL* 2007ko *SingleMalt*ekin (Hall *et al.*, 2007) lortutako oinarria eta Φ_4 EZB II zuhaitz-bankuarekin lortutako oinarria

LAS	Φ_1	Φ_2	Φ_3	Φ_4
10 geruzako balidazio gurutzatua	67,64	75,06 (+7,42)		
Testa	65,08	74,52 (+9,44)	74,99	78,46

3.6 taula – EZB I zuhaitz-bankuarekin erabilitako Φ_1, Φ_2 eta Φ_3 ezaugarri-modeloekin eta EZB II zuhaitz-bankuarekin erabilitako Φ_4 ezaugarri--modeloarekin lortutako LAS balioak

ebaluaziorako multzoan, LAS neurrian, 7,42 eta 9,44 puntuko hobekuntza lortu da, hurrenez hurren. MaltParser SingleMalt (Φ_3) sistemarekin alderatuta, ezaugarri-modeloan beste 7 ezaugarri gehiago dituela ikus daiteke, baina euren hobekuntza ez dator ezaugarri-modelotik, aplikatutako transformazio pseudo-proiektibotik baino. Gure ezaugarri-modeloari transformazio pseudo-proiektiboa aplikatuta, beste puntu bateko hobekuntza lortu da. EZB II zuhaitz-bankuko ebaluaziorako azpimultzoarekin lortutako zehaztasuna % 78,46koa da. Tamaina desberdineko zuhaitz-bankuen (EZB I eta EZB II zuhaitz-bankuak) emaitzak izateak tamainaren eragina aztertzeko aukera eman digu. Euskarako zuhaitz-bankuaren tamaina hirukoizterakoan, hau da, EZB I zuhaitz-bankutik (55.469 hitz eta 3.700 esaldiko zuhaitz-bankutik) EZB II zuhaitz-bankura (150.000 hitz eta 11.225 esaldiko zuhaitz-bankura) pasatzerakoan, ebaluaziorako azpimultzoan, ia lau puntu irabazten da (ikusi 3.6 taulan).

3.3.4 Ezaugarri morfologikoen antolaketa

Problemaren zehaztapena

Euskara hizkuntza eranskaria eta morfologikoki aberatsa da. *CoNLL* 2007ko lehiaketan parte hartu zuten hizkuntza guztien artean, euskara balio morfosintaktiko desberdin gehien zituena zen. Bigarren fase honetan, EZB I zuhaitz-bankuaren eta EZB II zuhaitz-bankuaren ezaugarri morfologikoen antolaketa egin da. EZB I zuhaitz-bankuan 359 eta EZB II zuhaitz-bankuan 350 balio morfosintaktiko desberdin daude. Balio morfosintaktiko gehienak *CoNLL-X* formatuko *FEATS* zutabean pilatuta agertzen dira, barra bertikal batez banaturik (ikusi 2.2.1.2 azpiatalean); adibidez, informazioa era honetan ager daiteke "BIZ-|INE|NUMS|MUGM"[9]. EZB I zuhaitz-bankuan eta EZB II zuhaitz-bankuan dauden ezaugarri linguistikoak gutxitzeak eta antolatzeak analizatzailearen zehaztasunean ezaugarri morfosintaktikoen eragina azter-

[9]BIZ-: ez biziduna, INE: inesiboa, NUMS: numero singularra eta MUGM: mugatua

tzeko aukera emango digu.

Esperimentuak

Esperimentu honetan, EZB I eta EZB II zuhaitz-bankurako hurrengo moldaketak proposatzen dira:

- Ezaugarri konplexuak balio sinpleagoak bihurtu. Adibidez, badaude postposizioak sortzen dituzten atzizki elkarketa batzuk. Postposizio hauek ezaugarri morfologikoa hartuz gero, kasu konplexua aurkituko da. Adibidez, "Corpusaren_bidez" postposizioan GEN_INS (genitibo atzizkiaren ondoren atzizki instrumentala) kasu konplexua aurkituko da. Kasu hauetan, azkeneko kasuarekin geratuko gara; GEN_INS kasua INS kasuagatik ordezkatuko da, analizatzailearentzat informazio garrantzitsuena azkenekoa izan daitekeelakoan.
- Balio morfosintaktiko desberdinak klaseetan multzokatu barik agertzen dira; hori dela-eta, balio morfosintaktiko bakoitzari aurrizki bat jarriko diogu. Adibidez: ABS (olutibo) kasuan, "KAS:" aurrizkia gehituz gero, "KAS:ABS" bihurtuko da.
- Ezaugarri batzuk ezabatuko dira, hitz osaketan garrantzia dutenak, baina analizatzaileari zarata sor diezaioketenak.
- Ezaugarri berri batzuk gehituko dira, beste ezaugarri batzuetatik inferi daitezkeenak. Helburua da maizago agertuko diren bestelako ezaugarri orokor bihurtzea. Adibidez: aditzen informazioan, subjektuaren, osagarri zuzenaren eta zeharkako osagarriaren pertsona agertu beharrean, aditz iragankor eta iragangaitz bezala bereiztea (edo nor, nor-nori, nor-nork edo nor-nori-nork ezaugarriekin bereiztea). Horrela APM (Aditz Pertsona Marka) marka berria sortu da APM:DA, APM:DU, APM:ZAIO eta APM:DIO balioekin; POS:+ marka berria postposizioa dela adierazteko, eta abar.
- Puntuazio markaren azpikategoria PUNT_MARKA hitz-formaren arabera, hurrengo azpikategorietan banatu da: PUNT_PUNT (.), PUNT_BI_PUNT (:), PUNT_ESKL (!), PUNT_GALD (?), PUNT_HIRU (...), PUNT_KOMA (,) eta PUNT_PUNT_KOMA (;).

Aldaketa hauek aplikatu ostean, EZB II zuhaitz-bankuaren ezaugarrien zutabean 136 balio desberdin dira, 26 klase morfosintaktikotan sailkaturik: mendeko perpausen informazioa, kasua, pertsona, numeroa, etab.

Emaitzak

Zehaztasuna neurtzeko LAS neurria (ikusi 2.2.1.2 atala) erabili da. 3.7 taulan ezaugarri morfologikoen moldaketarekin lortutako emaitzak *CONLL 2007 Multilingual Dependency Parsing* lehiaketan parte hartutako 20 sistemekin lortutako emaitzekin parekatu dira.

EZB I	**Sistemak**	**Testa (LAS)**
CoNLL 2007	Nilsson Malt konbinatua (Hall *et al.*, 2007)	% 76,94
	Carreras, 2007	% 75,75
	Titov and Henderson, 2007	% 75,49
	Johansson et al., 2007	% 75,08
	Hall Malt bakarra (Hall *et al.*, 2007)	% 74,99
Bengoetxea eta Gojenola, 2007	Informazio morfologikorik gabe	% 70,20
	Informazio morfologikoarekin (oinarria)	% 74,52
	Ezaugarrien moldaketarekin	% 75,01 (+0,49)
EZB II	Informazio morfologikoarekin (oinarria)	% 78,46
	Ezaugarrien moldaketarekin	% 78,89 (+0,43)

3.7 taula – EZB I zuhaitz-bankuko eta EZB II zuhaitz-bankuko ezaugarri morfologikoen moldaketa, *CoNLL* 2007ko lehiaketako sistema onenekin konparatuz

3.7 taularen lehenengo atalean ikus daitekeen legez, *CONLL 2007 Multilingual Dependency Parsing* lehiaketan EZB I zuhaitz-bankua erabiliz eskuratutako bost sistema onenak daude. Lehenengo lerroan, MaltParser sistemako aldaera ezberdineko 6 sistema konbinatuta (algoritmo proiektibo eta ez-proiektibo ezberdin eta ezkerretik eskuinera eta alderantziz aplikatutako sistema erabiliz) lortutako sistema dago. Bosgarren lerroan, Hall *et al.*-eko (2007) MaltParser sistema transformazio pseudo-proiektiboarekin konbinatu ostean lortutako emaitza ikus genezake.

3.7 taularen bigarren atalean, gure esperimentuetan lortutako emaitzak daude, MaltParser sistema bakarra (transformazio pseudo-proiektiboa gehitu barik) erabilita. Atal honetan, hiru emaitza dago: lehenengo lerroan, informazio morfosintaktikoarekin, erabilera urriak analisian (soilik hitza, lema, POS eta CPOS erabiliz) % 70,20ko zehaztasuna du. Bigarren lerroan, ezaugarrien moldaketa egin gabe, lortutako oinarriak % 74,52ko zehaztasuna du. Hirugarren lerroan, ezaugarriak moldatuta, zehaztasuna % 75,01era iristen da.

3.7 taularen hirugarren atalean, gure esperimentuetan EZB II baliatuta lortutako emaitzak daude. Atal honetan, emaitza bi dago: lehenengo lerroan, ezaugarrien moldaketa egin gabe, gure oinarriak % 78,46ko zehaztasuna du, eta, bigarren lerroan, ezaugarriak moldatuta, zehaztasuna % 78,89ra iristen

da.

Ondorioak
Ezaugarri morfologikoen antolaketa egin ostean, lehenengo ekarpenak plazaratuko dira:

- EZB I zuhaitz-bankuan, ezaugarri morfosintaktikorik gabe, zehaztasuna 4,5 puntu jaitsi da, eta horrek ezaugarri morfosintaktikoek euskaran duten garrantzia adierazten du.
- EZB I eta EZB II zuhaitz-bankuan dauden ezaugarri linguistikoak bakarrik antolatu ostean, analizatzailearen zehaztasuna hobetu da, oinarriarekiko 0,5 eta 0,43 puntuko igoera esanguratsua lortzen baita, hurrenez hurren.
- Ezaugarri morfosintaktiko gutxiago izateak entrenamenduko fasea 8 aldiz azkartzea ekarri du.
- EZB I zuhaitz-bankuan, izen kategoriako hitzen zehaztasuna % 66 da, eta zuhaitz-bankuko kategoriarik ugariena dela jakinda, ia errore guztien % 50 dela esan daiteke. Errore honen zergatia hau dela uste da: izenak berak kasu-markarik ez duenez (beste hitz batean dago), eta dependentzia-etiketa kasu gramatikalaren informazioarekin ezartzen denez, izena aditzarekin lotzerakoan dependentzia-etiketa okerrekoarekin egiten da batzuetan. Adibidez, “etxe handi horrekin” sintagman “etxe” aditzarekin lotzerakoan, ez dauka kasu-markarik, kasu-marka “horrekin” hitzan baitago (“-ekin” hitzari lotuta dago). Ondorioz, hitzak morfemetan banatzeak izan dezakeen eragina aztertzera garamatza (ikusi 4.1.2 atalean).
- EZB I zuhaitz-bankuan dauden esaldiak literatur mundutik eta euskaldunon egunkaritik lortu dira. Genero bereko esaldiekin lan eginez gero, zehaztasuna % 5 igotzen da.

3.3.5 Gainerako algoritmo familietara eta LibSVM-era esportatzea

Hirugarren fasean, algoritmo familia ezberdinen konbinaketaren zehaztasuna probatzeko helburuarekin, EZB II zuhaitz-bankurako onena den Nivre Arku Estandar algoritmoa eta transformazio pseudo-proiektiboa konbinaturik lortutako ezaugarri-modeloa beste familia batzuetako algoritmoen egiturara eta portaerara esportatu da.

Emaitzak
Esportatu ostean algoritmo familia bakoitzarekin lortutako zehaztasuna 3.6 taulan ikus daiteke.

Algoritmoa	**Testa (LAS)**
Nivre Arku Estandarra (*LibLINEAR*) + Pseudo	78,89
Stacklazy Ez-proiektiboa (*LibLINEAR*)	78,79
Covington Ez-proiektiboa (*LibLINEAR*)	78,71

3.8 taula – EZB II zuhaitz-bankuko eredurik onena gainerako algoritmo familietara esportatu osteko emaitzak (Pseudo = Pseudo-proiektiboa).

LibLINEAR ikasketa automatikoan, ezaugarri garrantzitsuenak binaka eta hirunaka konbinatu dira. LibLINEAR paketearekin lortutako ezaugarri-modeloa LibSVM paketeko bigarren graduko kernel polinomikora esportatzeko orduan, binaka eta hirunaka konbinatutako ezaugarriak banaka jarri dira eta EZB II zuhaitz-bankuarekin egindako esperimentuetan, LibSVM paketeko bigarren graduko kernel polinomikoan bilatutako parametro egokiak konstante mantendu dira, LibSVM paketeko bigarren graduko kernelaren parametroekin doikuntza prozesu bat egin ordez.

Esperimentua egiteko Stacklazy ez-proiektiboa eta Nivre Arku Estandar algoritmoa (transformazio pseudo-proiektibo barik) probatu dira. LibLINEAR ezaugarri-modeloari biko eta hiruko ezaugarrien konbinaketa guztiak kenduz, LibSVM paketeko bigarren graduko kernel funtzioarekin lortutako emaitzak 3.9 taulan agertzen dira.

Testa (LAS)	**LibLINEAR**	**LibSVM**
Nivre Arku Estandar + Pseudo	78,89	79,11 (+1,13)
Stacklazy ez-proiektiboa	78,79	79,81 (+1,02)

3.9 taula – LibLINEAR paketetik LibSVM paketera esportatuta lortutako LAS balioak (Pseudo = Pseudo-proiektiboa).

3.9 taulan LibSVM paketearentzat egokiak diren parametroak erabilita, hobekuntzaren proportzioa mantendu egiten dela ikusten da, eta ebaluaziorako laginean puntu bateko irabazia lortzen da LAS neurrian.

3.4 MSTParser sistemaren egokitzapena

Nahiz eta 2007ko *CoNLL* lehiaketan grafoetan oinarritutako lehen eta biga-

rren mailako sistemak aurkitu, MSTParser sistemak 2006ko *CoNLL* lehiaketan baino ez zuen parte hartu. EZB I zuhaitz-bankua 2007ko *CoNLL* lehiaketan baino ez zenez aurkeztu, ezin da euskararentzat erreferentziatzat hartu. MSTParsera instalatu ostean, urrats garrantzitsuena eta neketsuena euskara bezalako hizkuntza eranskarira moldatzea izan da, erabakiak hiru mailatan hartu behar baitira: ikasketa automatikoaren mailan, algoritmoaren mailan eta ezaugarri-modeloaren mailan. Ikasketa automatikoaren mailan, MSTParser sailkatzaileak ezaugarrien pisua ikasteko lineako inferentzian oinarritutako *MIRA* metodoa erabiltzen du. Algoritmoaren mailan, Eisner-en algoritmo proiektiboa eta Chu-Liu-Edmonds algoritmo ez-proiektiboak erabil daitezke. Halaber, zuhaitz guztien gaineko bilaketa lehen edo bigarren mailako ezaugarriak baliatuta egitea aukeratu behar da. Algoritmo onena finkatuta, ezaugarri-modeloaren balioak finkatzea da MSTParser sistemaren urratsik neketsuena eta zailena.

Antzerako lanak

MSTParser sistemari informazio morfologikoa gehitzeko ahalegin ugari egon dira. McDonald *et al.*-ek (2005) txekierak duen informazio morfologikoa MSTParser sistemari gehitzeko, *CoNLL-X* formatuan dagoen azpikategoriaren ezaugarriak (POS zutabeak) 13 karaktere izango ditu. Karaktere bakoitzak kategoria, azpikategoria, generoa, numeroa, kasua, edutezko genitiboaren generoa eta numeroa, pertsona eta denbora bezalako informazioa kodetuta eramango du. Collins *et al.*-ek (1999) datuen sakabanatze hori eta informazio berdineko datuen urritasuna saihesteko, azpikategoria zutabean, bi karaktereko kodeketa erabili dute: lehenengo karakterea kategoriarentzat eta, bigarrena, kasuarentzat edo, kasurik ezean, azpikategoriarentzat.

Esperimentuak

MSTParser sistemarekin lan egiteko erabakia hartu zenean, EZB II zuhaitz-bankuko bertsioa erabilgarri zegoen. MSTParser sistemaren oinarria bilatzeko hurrengo urratsak jarraitu dira:

1. Ezaugarri-modeloa eta ikasketa automatikoaren parametroak konstante jarrita, algoritmo ezberdinekin probatu ostean, bigarren mailako algoritmo ez-proiektiboarekin emaitzarik onena lortu da.

2. Algoritmo onena finkatuta eta ikasketa automatikoaren parametroak

konstante jarrita, ezaugarri-modeloan euskararen informazio morfosintaktikoa gehitzeko bi bide aztertu dira:

(a) EZB II zuhaitz-bankuan hurrengo aldaketak egin dira: alde batetik, azpikategoriari euskararen ezaugarri morfologiko garrantzitsuenak kateatuko dizkiogu (kasua eta mendeko perpausen informazioa, ikusi 3.5 atala), kategoria informazioarekin batera. Kategoriaren informazioa gehitzeko arrazoi nagusia izan da azpikategoriaren informazioan izen, adjektibo eta adberbio kategoriek "arrunta" balioa erabiltzen dutela adibidez: "Horko eredu famatua" sintagman agertzen diren hiru hitzek azpikategoria "arrunta" marka bera dute, nahiz eta adberbio, izen eta adjektibo kategoria ezberdinetakoak izan, hurrenez hurren.

(b) MSTParser sistemaren iturri kodea moldatuko da ezaugarri-modeloan, kasu-marka eta mendeko perpausen erlazio mota, eta azpikategoriaren maila berean jartzeak izan dezakeen eragina aztertuko da. MSTParser 0.5 bertsioarekin datozen ezaugarri lehenetsiei (ikusi 3.2.2.2 atala) 3.10 taulan agertzen direnak gehitu zaizkie. Oro har, azpikategoria ezaugarria agertzen den kasu guztietan, kasu-marka eta mendeko perpausen erlazio moten ezaugarriak gehitu dira. Horrez gain, EZB II zuhaitz-bankuan, azpikategoriaren zutabean, kategoria eta azpikategoria azpimarra batekin lotu dira.

Emaitzak

MSTParser bigarren mailako algoritmo ez proiektiboarekin egin dira 3.11 taulan agertzen diren esperimentu guztiak. Lehen lerroan, MSTParser sistemarekin datorren ezaugarri-modelo lehenetsia eta moldatu gabeko EZB II zuhaitz-bankuarekin lortutako emaitza agertzen da. Bigarren lerroan, MSTParser sistemarekin datorren ezaugarri-modelo lehenetsia dago, eta EZB II zuhaitz-bankuko azpikategorian, kategoria eta azpikategoriaren kateaketarekin, ebaluaziorako multzoan, 0,82ko hobekuntza esanguratsua lortu da. Hirugarren lerroan, MSTParser sistemarekin datorren ezaugarri-modelo lehenetsia agertzen da, eta EZB II zuhaitz-bankuko azpikategorian, kategoria, azpikategoria, kasua eta mendeko perpausen informazioaren kateaketarekin, ebaluaziorako multzoan, 6,67ko hobekuntza esanguratsua lortu da. Azkenik,

Atala	Erpina	Arku-ezaugarri multzoen balio berriak
1	Guraso hitza (P) Ume hitza (C) P eta C	Px, Py, Cx, Cy, Pxy, Cxy PxCx, PyCy, PxCy, PyCx PxyCy, PxCxy, PyCxy, PxyCxy non (x, y) ∈{(pos, kas), (pos, erl), (kas, erl) (erl, kas) }
2	P eta C-ren bitartean dauden hitzak (B)	PxBxCx non x ∈ {kas, erl}
3	P eta C-ren auzokoak eskuinekoak (PR/CR) eta ezkerrekoak (PL/CL)	PLxPxCxCRx, PLxPxCx, PLxCxCRx, PLxPxCRx, PxCxCRx, PxPRxCLxCx, PxPRxCLx, PxPRxCx, PxCLxCx, PRxCLxCx, PLxPxCLxCx, PxPRxCxCRx non x ∈ {kas, erl}
4	C-ren senideak (S)	CxSy, PzSzCz non (x, y) ∈ {(kas, pos), (erl, pos), (pos,kas), (pos,erl), (kas, word), (erl, word), (word,kas), (word,erl) } eta z ∈ {pos} Ezaugarri horiek ere norabide eta distantzia ezaugarriekin batera doaz

3.10 taula – MSTParser sistemari gehitutako ezaugarri berriak (CPOS = kategoria, POS = azpikategoria, KAS = kasu gramatikala, ERL = mendeko perpausen informazioa dira).

Zk.	EZB II	Ezaugarri-modelo	Testa (LAS)	Testa (UAS)
1	Moldatu barik	Lehenetsia	72,63	83,37
2	POSean CPOS gehituta	Lehenetsia	73,45 (+0,82)	83,61 (+0,24)
3	POSean CPOS, KAS eta ERL	Lehenetsia	79,30 (+6,67)	85,44 (+2,07)
4	Moldatu barik	Kode moldaketa	80,17 (+7,54)	86,01 (+2,64)

3.11 taula – MSTParser sistema lehenetsia eta doikuntza ezberdinekin EZB II zuhaitz-bankuarekin lortutako emaitzak (CPOS = kategoria, POS = azpikategoria, KAS = kasu gramatikala, ERL = mendeko perpausen informazioa dira)

laugarren lerroan, MSTParser sistemaren iturri kodea moldatu da, 3.10 taulako ezaugarri berriak gehitu dira ezaugarri-modeloan, hau da, kasu-marka eta mendeko perpausen erlazio mota azpikategoriaren maila berean jartzean, ebaluaziorako multzoan, 7,54ko hobekuntza esanguratsua lortu da. Hori dela-eta, hemendik aurrera, tesi-lan honetan, MSTParser sistemaren oinarri bezala kode moldaketa hau erabiliko da. Ondorioz, kasua eta mendeko erlazio marka azpikategoriaren maila berean jartzerakoan, MSTParser sistemarekin sortutako analizatzaileen zehaztasuna asko igotzen dela probatu da.

3.5 Ezaugarri morfosintaktikoen eragina analisian

Problemaren zehaztapena
MaltParser eta MSTParser sistemetan oinarri sendoak bilatu ostean, eta ezaugarriak klaseetan multzokatuta izateak hurrengo hipotesiak aztertzeko aukera eman du:

- Ezaugarri morfosintaktiko guztietatik, zeintzuk dira analizatzailearen zehaztasunean ia eraginik ez dutenak, eta baztertu ahal direnak?
- Zein da analizatzailearen zehaztasuna gehien igotzen duen konbinaketa murriztua?
- Erabili ahal da konbinaketa murrizturen bat 26 ezaugarriak ordezkatzeko?
- Lortutako konbinaketa murriztuek izaera ezberdineko sistemetan antzeko portaera dute? Galdera honi erantzuteko sistema ezberdin bi probatu dira: batetik, trantsizioetan oinarritutako eta izaera lokala duen MaltParser sistema, eta bestetik, grafoetan oinarritutako eta izaera globala duen MSTParser sistema.

Esperimentuak
Esperimentuak egiteko, MSTParser sistemako kode moldaketa eta MaltParser sistemarekin datorren MaltOptimizer sistema erabili da. Lehenengo eta behin, ezaugarri bakoitza bakarka gehitu da. Gero, bakarka lortutako ezaugarririk onena gainerako beste 25 ezaugarriekin konbinatu da. Biko konbinaketarik onena lortu ostean, gainerako 24 ezaugarriekin konbinatu da. Modu horretan jokatu da harik eta lau ezaugarri onenak konbinatzera heldu arte.

Emaitzak

3.12 taulan ikus daitekeenez, sistema bietan, EZB II zuhaitz-bankuko 5 ezaugarririk onenak (bakarka) agertzen dira, eta, azkeneko lerroetan, berriz, biko, hiruko eta lauko ezaugarrien konbinaketa onenak.

Ezaugarri morfosintaktikoak	**MaltParser (MaltOptimizer)**	**MSTParser**
Gabe	69,28	71,35
Guztiak	77,47 (+8,19)	80,17 (+8,82)
KAS	**75,53 (+6,25)**	**78,28 (+6,93)**
ERL	70,32 (+1,04)	73,17 (+1,82)
MUG	70,26 (+0,98)	72,58 (+1,23)
NUM	70,03 (+1,02)	72,34 (+0,99)
ASP	69,44 (-0,22)	70,89 (-0,46)
KAS+ERL	**77,29 (+8,01)**	**80,17 (+8,82)**
KAS+ERL+NUM	77,12 (+7,84)	79,89 (+8,54)
KAS+ERL+NUM+MUG	**77,29 (+8,01)**	-
KAS+ERL+NUM+NORK	-	**80,41 (+9,06)**

3.12 taula – EZB II zuhaitz-bankuko 5 ezaugarri onenak (bakarka) eta biko, hiruko eta lauko konbinaketa onenak, sistema bietan testarekin lortutako emaitzak LAS neurrian (NUM = Numeroa, MUG = mugatasuna, NORK = Pertsona, KAS = kasu gramatikala, ERL = mendeko perpausen informazioa dira)

Bi sistemetan gauza parekoak gertatzen dira:

- Informazio morfologiko barik, sistema bien galera oso altua da. Adibidez, ebaluaziorako multzoan, MaltParser eta MSTParser sistemekin, $-8,19$ eta $-8,82$ puntuko jaitsiera dago, hurrenez hurren. Informazio morfosintaktikoaren eragina analizatzaile bietan oso handia da.
- 26 ezaugarrietatik, kasua da sistema bietan zehaztasuna gehien igotzen duena, 6,25 eta 6,93 puntuko igoera lortuz. Oso kopuru nabaria da, gainerakoekin alderatuta.
- Sistema bietan, bakarkako bigarren ezaugarririk onena mendeko erlazioaren marka da, 1,04ko eta 1,82ko igoerekin.
- Kasua, mendeko perpausen informazioa, numeroa eta mugatasuna kenduta, gainerako ezaugarriek ez dute bakarka inolako hobekuntzarik ematen.
- Kasua eta mendeko perpausen informazioen ezaugarriak konbinatuta, igoera +8,01ekoa eta 8,82koa da.
- Kasua eta mendeko perpausen informazioa gainerako ezaugarriekin konbinatu ostean, hiruko konbinaketa onena numeroarekin lortu da sistema bietan, eta zehaztasunean 7,84 eta 8,54 puntuko hobekuntzak lortu dira.

– Kasua, mendeko perpausen informazioa eta numeroa gainerako ezaugarriekin konbinatu ostean, lauko konbinaketarik onena mugatasunak eman du MaltParser sistema baliatuta; zehaztasunak 8,01eko igoera izan du; MSTParser sistemarekin emaitzarik onena NORK pertsona ezaugarriaren konbinaketarekin lortu da, zehaztasunean 9,06ko igoera lortuz. Hortik aurrera, lauko konbinaketa onenari ezaugarri gehiago batu bazaizkio ere, ez da inolako hobekuntzarik lortu. Aipatzekoa da, lau ezaugarri hauekin, gainerako ezaugarri guztiekin lortutakoa edo gehiago lortu dela.

Ondorioak

Ezaugarri morfosintaktikoek, bakarka edo konbinatuta, eta sistema desberdinak baliatuta analisian duten eragina aztertu ostean, hurrengo ondorio nagusiak atera dira:

– Kasua eta mendeko perpausen informazioa izeneko ezaugarriak osagarriak dira, bakarka ematen duten hobekuntza batzerakoan, mantendu egiten baita. Ezaugarri gabeko sistemarekin alderatuta, 8 eta ia 9 puntuko hobekuntza lortzen da sistema bietan. KAS+ERL konbinaketari gehitutako gainerako ezaugarriek analisian eragin txikiagoa dute.
– KAS+ERL+NUM+MUG eta KAS+ERL+NUM+NORK konbinaketa murriztuak 26 ezaugarriak ordezkatzeko erabil ditzakegu, ezaugarri guztiak erabilita lortutako adina lortu baitugu.
– Trantsizioetan oinarritutako eta izaera lokala duen MaltParser eta grafoetan oinarritutako eta izaera globala duen MSTParser sistemekin probatu ostean, lortutako konbinaketa murriztuek izaera ezberdineko sistemetan antzeko portaera dutela esan daiteke.
– FEATS zutabean dauden ezaugarri morfosintaktiko guztiak kasua eta mendeko perpausen informazioa ezaugarrien konbinaketarekin ordezkatuz gero, entrenamenduko fasea azkartu da, ezaugarri klase gutxiago egon baitira.
– MSTParser sistemarekin MaltParser+MaltOptimizer sistemarekin baino emaitza hobeak lortu dira, oinarrian 2,7 puntuko aldea baitago. Hori dela-eta, hurrengo atalean euskararekiko MaltParser+MaltOptimizer optimizazioa gauzatzen ahaleginduko gara.

3.6 MaltOptimizer, MaltParser egokitzeko tresna

Problemaren zehaztapena

Aurreko atalean, MSTParser sistemarekin MaltOptimizer sistemarekin baino 2,7 puntu gehiago lortu dira oinarrian. Gauza bera gertatzen da MaltParser euskarara egokitu ondoren sortutako sistemarekin, hots, Maltixarekin, 2,44 puntu gehiago lortzen baitugu (ikusi 3.13 taulako bigarren lerroan). Atal honetan, MaltOptimizer sistemarekin, emaitza hobeak lortzeko asmoz, EZB II zuhaitz-bankuan egindako moldaketak aurkeztuko dira. MaltOptimizer sistemaren funtzionamendua aztertu ostean, ezaugarri-modeloa bilatzeko orduan, batez ere hitza eta azpikategoria ezaugarriekin lan egiten duela ikusi da. Euskara hizkuntza eranskaria izanik, hitzari baino lemari garrantzia gehiago emateko, hitza eta lema trukatu dira. Azkenik, FEATS zutabeko (ikusi 2.2.1.2 taula) ezaugarrietan hain garrantzitsuak diren kasua eta mendeko perpausen informazioa azpikategoriarekin batera kateatu dira, azpikategoria zutabean.

Esperimentua

Esperimentua egiteko EZB II zuhaitz-bankua eta MaltOptimizer sistemaren 1.0.2 bertsioa erabili dira. Zehaztasuna neurtzeko, LAS eta UAS neurriak erabili dira. *CoNLL-X* formatuko EZB II zuhaitz-bankuan hitza eta lema trukatu dira, eta azpikategoria zutabean, ezaugarri morfosintaktiko garrantzitsuen (ikusi 3.5) kateaketa egin da, hau da, azpikategoriaren zutabean kategoria, azpikategoria, kasua eta mendeko perpausen informazioa kateatu dira. Azkenik, aurreko kateaketari numeroa eta nork marka gehitu zaizkio.

Emaitzak

Zk.	Deskribapena	LAS (Testa)	UAS (Testa)
1	MaltOptimizer sistemaren oinarria LibLINEARekin	76,45	82,44
2	Maltixaren oinarria LibLINEARekin	**78,89 (+2,44)**	**83,63 (+1,19)**
3	MaltOptimizer non lema eta hitza trukatu eta POS=CPOS+POS+KAS+ERL	**79,55 (+3,1)**	**84,53 (+2,09)**
4	MaltOptimizer non lema eta hitza trukatu eta POS=CPOS+POS+KAS+ERL+NUM+NORK	74,14 (-2,31)	80,18 (-2,26)
5	3. lerroko konfigurazioa LibSVMra esportatuta	**79,80**	**84,78**
6	Maltixa-ren oinarria LibSVMra esportatuta	**79,81**	**84,41**

3.13 taula – MaltOptimizer sistemaren egokitzapena

Erabilitako sistemak eta lortutako emaitzak 3.13 taulan lerroz lerro deskribatu dira:

1. lerroan, MaltOptimizer sistemarekin lortutako oinarria dago, MaltOptimizerek LibLINEAR kernel funtzioak baino ez dituelako erabiltzen.

2. lerroan, Maltixak LibLINEARekin % 78,89ko zehaztasuna du, hau da, 2,44 puntuko igoera MaltOptimizer oinarriarekin alderatuta.

3. lerroan, EZB II zuhaitz-bankuan lema eta hitza zutabeak trukatu ondoren eta azpikategoriaren zutabean, kategoria, azpikategoria, kasua eta mendeko perpausen informazioa kateatu ostean, MaltOptimizer sistemarekin +3,1 puntuko hobekuntza lortu da.

4. lerroan, EZB II zuhaitz-bankuan lema eta hitza zutabeak trukatu ondoren eta azpikategoriaren zutabean, kategoria, azpikategoria, kasua, mendeko perpausen informazioa, numeroa eta nork marka kateatu ostean, MaltOptimizer sistemarekin 2,31 puntuko jaitsiera egon da, datu-sakabanaketaren eraginez.

5. lerroan, 3. lerroan lortutako konfiguraziorik onena LibSVM paketera esportatu ostean, hau da, LibLINEAR paketearekin lortutako ezaugarri-modeloari biko eta hiruko ezaugarri-konbinaketa guztiak kendu ostean, eta LibSVM paketearen parametroen bilaketa egin barik, Maltixako LibSVM paketerako parametroak erabilita, 0,25 puntuko hobekuntza lortu da, % 79,80ra heldu arte.

6. lerroan, Maltixa LibSVMekin lortutako gure oinarria da MaltOptimizer sistemarekin lortutako konfiguraziorik onena, LibSVMra esportatu osteko emaitzarekin konparatzeko.

Ondorioak

EZB II zuhaitz-bankuko moldaketekin eta moldaketarik gabe, MaltOptimizer sistemarekin lortutako ezaugarri-modeloak alderatuz gero (ikusi 3.14 taula), hurrengo berezitasunak agertu dira:

– Lemak bere protagonismoa bereganatzen duela ikusten da; ordezkaritza bikoiztu egiten du, 2tik 4ra pasatzen baita (ikusi 3.14 taula).

	Egozpen, Helbide funtzioak		
	Hall eta Nivre, 2005	MaltOptimizer	
Zk.	Hizkuntza guztietan	EZB II	EZB IIko moldaketa onena
1	POSTAG, Stack[0]	POSTAG, Stack[0]	POSTAG, Stack[0]
2	POSTAG, Input[0]	POSTAG, Input[0]	POSTAG, Input[0]
3	POSTAG, Input[1]	POSTAG, Input[1]	POSTAG, Input[1]
4	POSTAG, Input[2]	POSTAG, Input[2]	POSTAG, Input[2]
5	POSTAG, Input[3]	POSTAG, Input[3]	POSTAG, Input[3]
6	POSTAG, Stack[1]	POSTAG, Stack[1]	POSTAG, Stack[1]
7	DEPREL, Stack[0]	DEPREL, Stack[0]	DEPREL, rdep(Input[0])
8	DEPREL, ldep(Stack[0])	DEPREL, ldep(Stack[0])	DEPREL, ldep(Stack[0])
9	DEPREL, rdep(Stack[0])	DEPREL, rdep(Stack[0])	DEPREL, rdep(Stack[0])
10	DEPREL, ldep(Input[0])	DEPREL, ldep(Input[0])	DEPREL, ldep(Input[0])
11	FORM, Stack[0]	FORM, Stack[0]	-
12	FORM, Input[0]	FORM, Input[0]	-
13	FORM, Input[1]	FORM, Input[1]	LEMMA, Input[1]
14	FORM, head(Stack[0])	FORM, head(Stack[0])	LEMMA, head(Stack[0])
15		FORM, Stack[1]	FORM, Input[0]
16		CPOSTAG, Input[0]	POSTAG, ldep(Stack[0])
17		CPOSTAG, Stack[0]	POSTAG, rdep(Stack[0])
18		LEMMA, Input[0]	LEMMA, Input[0]
19		LEMMA, Stack[0])	LEMMA, Stack[0]
20		FEATS, Input[0]	-
21		FEATS, Input[1]	-
22		FEATS, Stack[0]	FEATS, Stack[0]
23		FEATS, Stack[1]	-
24		FEATS, Input[0]	FEATS, Input[0]
25		FEATS, Input[1]	-
26		FEATS, Input[2]	-

3.14 taula – Ezaugarri-modelo ezberdinak zutabeka: Hall eta Nivre-k (2005) aurkitutako ezaugarri-modeloa, hizkuntza ezberdinetan baliagarria dena, MaltOptimizer-ek EZB II zuhaitz-bankurako optimizatutako ezaugarri-modeloa eta MaltOptimizer-ek EZB II zuhaitz-bankuko moldaketa onenarako optimizatutako ezaugarri-modeloa

- Kategoria, kasua eta mendeko perpausaren informazioa azpikategorian kateatu izanak bai kategoriaren bai FEATS zutabearen protagonismoaren galera dakar; izan ere, ezaugarrien-modeloan FEATS ezaugarri morfosintaktikoen ordezkaritza 7tik 2ra eta kategoriaren ordezkaritza 2tik 0ra pasatzen dira.
- Informazio kontzentrazio honek algoritmoa aukeratzerakoan ere asko laguntzen duela ikusi da. Moldaketarik gabeko EZB II zuhaitz-bankuan *Covington* algoritmo ez-proiektiboa aukeratzen badu, moldatutako EZB II zuhaitz-bankuan, berriz, Nivre Arku Estandar algoritmoa aukeratzen du, eta probatuta dago (ikusi taulan) EZB II zuhaitz-bankurako egokiago dela Nivre Arku Estandar algoritmoa.

3.7 Ezaugarri automatikoen eragina

Problemaren zehaztapena

Orain arte egindako esperimentu guztietan zuhaitz-bankuan dauden ezaugarriak, hau da, hizkuntzalari talde batek eskuz gainbegiratutako ezaugarriak (hemendik aurrera zuhaitz-bankuko *urre-patroiko* ezaugarriak bezala ezagutuko direnak) erabili dira. Atal honetan, era automatikoan lortutako ezaugarriak erabiltzerakoan, benetako egoeretan analizatzaile sintaktikoak izan dezakeen zehaztasuna neurtuko da. Euskara bezalako hizkuntza eranskariak oso anbiguoak direnez, euskararako analizatzaile morfologikoaren irteeran kategoria baino ez bada kontuan hartzen, hitz-forma bakoitzeko, batez beste, 1,55 aukera izan ditzakegu. Baina informazio morfosintaktiko guztia (kasu-marka, numeroa edo aspektua bezalakoak) kontuan hartuz gero, batez bestekoa 2,65era iristen da. Analizatzaile morfologikoaren irteera desanbiguatzea arazo garrantzitsua bilakatu da. Aukera egokia hartzeko, askotan, hitzaren inguruko informazio lokala izatearekin nahikoa da, baina, beste askotan, izen-sintagmak aditzarekin izan dezakeen komunztadura (subjektua, objektu zuzena eta zehar-objektua) zein den aukeratzeko orduan, urruneko informazioa behar da. Gainera, euskararen kasuan, hurrenkera askeko hizkuntzen arazoa gehitu beharra dago. Hori dela-eta, kategoria eta edozein ezaugarri morfosintaktiko ez egokik analizatzaile sintaktikoan izan dezakeen eragina aztertuko da.

Antzerako lanak

Analizatzaile sintaktikoa analizatzaile morfologikoarekin batzeko ahalegin

ugari egin dira: hebreeraz (Goldberg eta Tsarfaty, 2008); latinez, grekoz, txekieraz eta hungarieraz (Lee *et al.*, 2011); eta, orain dela gutxi, Bohnet eta Nivre-ek (2012) aurkeztutako trantsizioetan oinarritutako sistema, hau da, kategoria-ezarpena (*POS tagging*) eta dependentzietan oinarritutako analizatzaile etiketaduna (*Labelled Dependency Parsing*) batuz emaitza onak lortu dituzte. Morfologikoki aberatsak diren hizkuntzetan (euskara, grekoa eta turkiera bezalakoetan) analizatzaile morfologikoaren emaitzak apalagoak dira (Nivre, 2007). EZB I zuhaitz-bankuan eta EZB II zuhaitz-bankuan datozen ezaugarri morfosintaktikoak erabili beharrean, analizatzaile morfologikoak emandako ezaugarri morfosintaktikoak erabiltzerakoan zenbaterainoko galera gertatzen den jakiteko, hurrengo esperimentuak gauzatu dira.

Esperimentuak

Gure esperimentuetan, analizatzaile sintaktikoa analizatzaile morfologikoarekin batzeko, tutu edo kanalizazio sistema erabili da. Testua analizatzaile morfologikotik (Aduriz, 2000) pasa ostean, desanbiguatzaile morfologikotik (Ezeiza *et al.*, 1998) pasa da (ikusi 3.1 atala). Desanbiguazio morfologikoa egiteko modulu bi konbinatzen (ikusi 3.1.2 atala) eta gehien desanbiguatzen dituen aukera erabili da. Aukera honekin, desanbiguazio moduluen irteera zuzena % 97ra iristen da. Modulu honen irteerak, batez beste, 1,3 aukera eskaintzen ditu hitz-forma bakoitzeko. Erabiltzen diren analizatzaile sintaktikoek aukera bakarra behar dutenez, desanbiguazio moduluak ematen dituen aukeretatik, lehenengo aukera (sarriena) hartu da.

EZB II zuhaitz-bankuko esaldiak analizatzaile morfologikotik pasa ostean, EZB II zuhaitz-bankuko esaldiak ezaugarri automatikoekin osatu dira (hemendik aurrera ezaugarri automatikoak dituen zuhaitz-bankua bezala ezagutuko dena). Horrela, urre-patroiko ezaugarriekin eta ezaugarri automatikoekin dagoen aldea konparatzeko balio izango digu. EZB II zuhaitz-bankuko esaldiak analizatzaile morfologikotik pasa ostean, hitzak parekatzeko arazoa gertatu da. Arazoa, gehienbat, hitz konposatuetan (entitate, aditz konposatu eta postposizio konplexu, eta abarretan) gertatu da, eskuz etiketatzerakoan tratamendu ezberdina jarraitu baitzen. Hori dela-eta, parekatze prozesu bat egin da.

Analizatzaile sintaktikoak sortzerakoan, ezaugarri automatikoekin edo urre-patroiko ezaugarriekin entrena ditzakegu. Bengoetxea *et al.*-en (2011) lanean, ezaugarri automatikoekin entrenatuz gero analizatzaile sintaktikoaren emaitzak hobeak zirela probatu genuen. Hori dela-eta, esperimentuetan,

entrenatzeko ezaugarri automatikoak erabili dira.

Esperimentuak egiteko, MSTParser eta MaltParser sistemetako algoritmo familia ezberdinekin probatu da, portaera berdintsua lortzen den jakiteko. Anbiguotasun morfologikoak izan dezakeen eragina ebaluatzeko, euskarazko zuhaitz-bankua hiru zati desberdinetan banatu da: entrenamendurako % 80, garapenerako % 10 eta ebaluaziorako % 10. Esperimentu guztiak garapenerako datuekin egin dira, eta, azkenik, konfigurazio onena ebaluatzeko datuekin egin da.

Emaitzak

Testa	**Ezaugarriak**	**Urre-patroikoak**		**Automatikoak**	
Zk.	**Deskribapena**	**LAS**	**UAS**	**LAS**	**UAS**
1	MST ez-proiektiboa	80,17	86,01	77,88 (-2,29)	84,08 (-1,93)
2	Malt Nivre Arku Estandar + Pseudo	78,89	83,63	77,31 (-1,58)	82,62 (-1,01)
3	Malt Stacklazy	78,79	83,43	76,83 (-1,96)	82,18 (-1,25)
4	Malt Covington	78,71	83,50	76,98 (-1,73)	82,17 (-1,33)

3.15 taula – MaltParser (Malt) eta MSTParser (MST) sistemekin, eta EZB II zuhaitz-bankuko urre-patroiko ezaugarriak eta ezaugarri automatikoak erabilita lortutako emaitzak (Pseudo = Pseudo-proiektiboa).

3.15 taulan erabilitako sistemak eta lortutako emaitzak lerroz lerro deskribatuko dira:

1. lerroan, bigarren mailako MSTParser algoritmo ez-proiektiboa eta kode-moldaketarekin lortutako emaitzak daude. Ezaugarri automatikoekin LAS neurriko zehaztasunean -2,29 puntuko galera dago.

2. lerroan, MaltParser sistemako LibLINEAR paketea eta Nivre Arku Estandar algoritmo proiektiboarekin batera transformazio pseudo-proiektiboa aplikatu ostean lortutako emaitzak daude. Ezaugarri automatikoekin LAS neurriko zehaztasunean -1,58 puntuko galera dago.

3. lerroan, MaltParser sistemako LibLINEAR paketea eta Stacklazy algoritmo ez-proiektiboa aplikatu ostean lortutako emaitzak daude. Ezaugarri automatikoekin LAS neurriko zehaztasunean -1,96 puntuko galera dago.

4. lerroan, MaltParser sistemako LibLINEAR paketea eta Covington algoritmo ez-proiektiboa aplikatu ostean lortutako emaitzak daude. Ezaugarri automatikoekin LAS neurriko zehaztasunean -1,73 puntuko galera dago.

Ondorioak
EZB II zuhaitz-bankuan ezaugarri automatikoak erabilita, MSTParser sistemarekin -2,29 puntuko galera gertatzen da LAS neurrian ebaluaziorako azpimultzoan, eta MaltParser sistema onenarekin -1,58 puntuko galera gertatzen da. Euskaraz dagoen anbiguotasuna kontuan harturik, emaitzak espero baino hobeak dira.

3.8 Laburpena eta ondorengo pausoak

Labur-zurrean, tesi kapitulu honetan euskararako analizatzaile sintaktiko estatistikoa erabiltzeko ezinbestekoa den informazio morfosintaktikoa nola lortzen den ikusi da. IXA taldean esaldi baten analisi sintaktikoa lortzeko, gaur egun erabiltzen diren baliabideak edo moduluak aurkeztu dira. Ikusten denez, euskara bezalako hurrenkera libreko hizkuntza eranskariek, dependentzietan eta datuetan oinarritutako analizatzaileen sortzaile nagusietara (MSTParser eta MaltParser) egokitzeko jorratzen duten prozesua ez da lan tribiala. Bukatzeko, ezaugarri automatikoek analisi sintaktikoan duten eragina aztertu da. Euskaran dagoen anbiguotasuna kontuan harturik, emaitzak espero baino hobeak izan dira. Euskararentzako oinarriak finkaturik, hurrengo kapituluan analisi sintaktikoaren emaitzak hobetze aldera aplikatutako teknika ezberdinak deskribatuko dira.

4

Zuhaitz-transformazioak, pilaketa eta bozketa analisi sintaktikoan

Aurreko kapituluan euskararako analizatzaile sintaktiko estatistikoa erabiltzeko, ezinbestekoa den informazio morfosintaktikoa nola lortzen den ikusi da. IXA taldean esaldi baten analisi sintaktikoa lortzeko gaur egun erabiltzen diren baliabideak edo moduluak aurkeztu dira. Ikusi denez, euskara bezalako hurrenkera libreko hizkuntza eranskarietan, dependentzietan eta datuetan oinarritutako analizatzaileen sortzaile nagusietara (MSTParser eta MaltParser) egokitzeko prozesua ez da lan tribiala izan. Horretaz gainera, ezaugarri automatikoek analisi sintaktikoan duten eragina aztertu da. Kapitulu honetan, analizatzaile sintaktikoen emaitzak hobetze aldera, hainbat hizkuntzatan egokiak izan diren teknikak (Collins, 2009; Nilsson *et al.*, 2008a) euskaran ere egokiak direla ikusiko da. Baina gure esperimentuak haratago doaz; izan ere, euskararen izaera kontuan hartuta, euskararako egokiak diren sintagmen transformazioa, mendeko perpausen transformazioa eta koordinazioaren transformazio berriak probatu dira (ikusi 4.1 atalean). Hainbat lanetan, analizatzailearen zehaztasuna hobetzeko pilaketa teknika erabili da, lehenengo analizatzaile baten irteeran lortutako egiturazko ezaugarriak bigarren analizatzailearen sarrera aberasteko erabili dira (Martins *et al.*, 2008; Nivre eta McDonald, 2008; McDonald eta Nivre, 2011). Hemen ere gure esperimentuak haratago doaz; izan ere, pilaketan egiturazko ezaugarriak goratzeaz gain, informazioa irizpide linguistikoen bidez goratuko da. Bukatzeko, sortutako oinarrizko sistemen eta sistema hedatuen (zuhaitz-transformazio eta pilaketa sistemen) irteerak bozketa bidezko MaltBlender tresnarekin konbinatu dira.

Gainera, esperimentuak egiteko, urre-patroiko etiketak eta etiketa automatikoak erabili dira, egoera errealean teknika hauek izan dezaketen eragina probatzeko. Azkenik, aplikatutako teknikekin lortutako emaitzak aurkeztu eta ondorio batzuk aterako dira.

4.1 Zuhaitz-transformazioak

Nahiz eta buru-osagarri eta buru-modifikatzaile egitura gehienek analisi berdintsua izan dependentzia-gramatikan, badaude eztabaidagarriak diren egitura asko, besteak beste: 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. Erabakitzeko unean teoria ezberdinak aurki daitezke. Sekzio honetan, etiketatze teoria desberdinen eragina aztertzeko euskarako zuhaitz-bankuari aplikatutako aurretiko eta ondorengo prozesaketa ezberdinak azalduko dira: proiektibizazio-transformazioa (arku ez-proiektiboak proiektibo bihurtzeko transformazioa, hemendik aurrera proiektibizazioa deituko dena) sintagmen transformazioa, mendeko perpausen transformazioa eta koordinazioaren transformazioa.

Collins-ek (2009) zuhaitzen aurretiko prozesaketa erabili zuen analizatzailearen emaitzak hobetzeko helburuarekin. Dependentzietan oinarritutako analizatzaileei dagokienez, Nilsson *et al.*-ek (2008a) proiektibizazioa, koordinazio-transformazioak eta aditz-taldeko transformazioak hizkuntz anitzetan onuragarriak zirela probatu zuten. Turkieran (Eryiğit *et al.*, 2008), unitate azpi-lexikalak edo talde inflexionalak erabiliz emaitza hobeak lortu zituzten.

Dependentzia-zuhaitzetan aplikatzen diren transformazioek kutxa beltzaren metodoa erabiltzen dute. Euskarako zuhaitz-bankuko transformazioak egiteko, oro har, hurrengo bost urratsak jarraitu dira:

1. Entrenamendu multzoko esaldiak transformatu

2. Transformatutako multzoarekin ikasiz, analizatzaileen sortzaileak analizatzaile sintaktiko baten modeloa sortu

3. Test multzoko esaldiak sortutako analizatzaile sintaktikoarekin analizatu

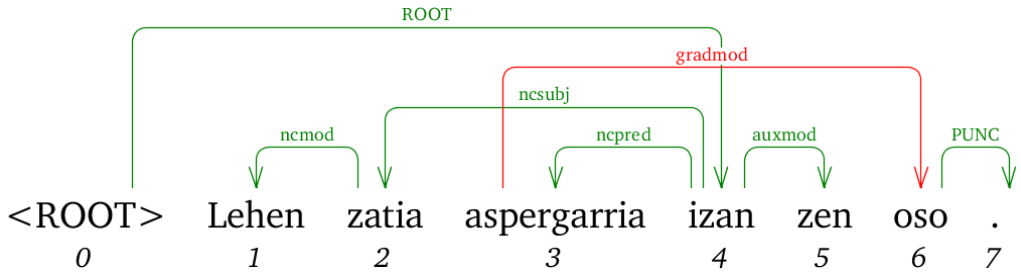
4. Analizatzaile sintaktikoaren irteerari alderantzizko transformazioa aplikatu

5. Bukatzeko, alderantzizko transformazioaren irteera eta test multzoko esaldiak alderatu

Atalez atal, euskarako zuhaitz-bankuetan aplikatutako lau transformazio mota azalduko dira: proiektibizazioa, sintagmen transformazioa, mendeko perpausen transformazioa eta koordinazioaren transformazioa. Lehenengo atalean, proiektibizazioa azalduko da, hizkuntzarekiko independentea den transformazioa. Gainerako hiru ataletan, hizkuntzarekiko mendekoagoak diren sintagmen, mendeko perpausen eta koordinazioaren transformazioak azalduko dira. Euskara hizkuntza buru-azkena eta ezaugarri morfologiko garrantzitsuenak (kasua, numeroa, mendeko perpausen informazioa, eta abar) sintagmaren azkeneko hitzean dituen hizkuntza izateak, transformazioak gauzatzea ahalbidetu du.

4.1.1 Proiektibizazioa

Proiektibitate printzipioa (ingelesez, *projectivity* edo *adjacency*) hitzen arteko arkuak elkarrekin gurutzatu gabe irudikatzen direnean gertatzen da hitzak hurrenkera linealean jarriz gero. Adibidez, 4.1 irudiko esaldian "oso" graduatzailea eta modifikatzen duen "aspergarria" hitza, "izan zen" aditzak banatzen ditu. Ondorioz, esaldi honen arkuak elkarrekin gurutzatzen dira. Beraz, esaldi honetan proiektibitate printzipioa ez dela betetzen eta esaldi ez-proiektiboa dela esango da.



4.1 irudia – EZB II zuhaitz-bankuko esaldi ez-proiektiboa

Trantsizioetan oinarritutako algoritmo gehienek dependentzia-zuhaitz proiektiboak baino ez dituzte sortzen, hau da, elkarrekin gurutzatzen ez diren

arkuak. 4.1 taulan, 2006ko *CoNLL-X Shared Task Multilingual Dependency Parsing* lehiaketan (Buchholz eta Marsi, 2006) 13 hizkuntzetatik dependentzia ez-proiektibo gehien izan zutenak ikus daitezke.

Hizkuntza	**Dependentzia ez-proiektiboak (%)**	**Esaldi ez-proiektiboak (%)**
Nederlandera	5,4	36,4
Alemana	2,3	27,8
Txekiera	1,9	23,2
Esloveniera	1,9	22,2
Portugalera	1,3	18,9
Daniera	1,0	15,6

4.1 taula – 13 hizkuntzetatik dependentzia ez-proiektibo gehien izan zuten hizkuntzak 2006ko *CoNLL-X Shared Task Multilingual Dependency Parsing* lehiaketan (Buchholz eta Marsi, 2006).

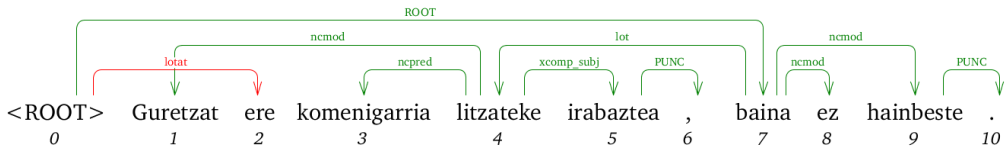
Euskara hizkuntza ez-proiektiboen artean koka daiteke, kontuan hartuta, EZB I zuhaitz-bankuak % 2,9 arku ez-proiektibo eta % 26,2 esaldi ez-proiektibo[1] dituela eta, EZB II zuhaitz-bankuak, aldiz, % 1,18 arku ez-proiektibo eta % 13,28 esaldi ez-proiektibo. Gure aburuz, EZB I zuhaitz-bankuan arku ez-proiektibo gehiago izateko arrazoi bat hurrengoa da: EZB I zuhaitz-bankua lortzeko, 3LBko corpusa *CoNLL-X* formatura egokitu zenean, 3LBko corpusean etiketatuta zetorren elipsia kendu eta elipsira zihoazen arku guztiak *ROOT*era bideratu ziren. EZB II zuhaitz-bankua lortzeko, erabili zen corpusa, aldiz, esaldian agertzen ziren hitzetan baino ez dago oinarritua. EZB II zuhaitz-bankuko arku ez-proiektiboak zeintzuk diren aztertzeko orduan, % 35 "lotat" erlazioagatik dela ikusi da. Erlazio hau kanpoko esaldi baten jarraipena edo lotura adierazteko erabiltzen da, eta erlazio honek *ROOT* buruarekin lotzerakoan erlazio ez-proiektiboa sortzen du (ikusi 4.2 irudia).

Nahiz eta EZB II zuhaitz-bankuan dependentzia ez-proiektiboen kopurua jaitsi, sintagmen transformazioa eta mendeko perpausen transformazioa aplikatu ostean, esaldi ez-proiektiboen kopurua % 13,28tik % 31,84ra igo dela ikusi da.

EZB zuhaitz-bankuetako arku ez-proiektiboekin, hiru eratara joka daiteke:

– Algoritmo ez-proiektiboak erabiltzea, oro har, konplexuagoak eta konputazionalki neketsuagoak direnak. Adibidez, Covington algoritmo ez-proiektiboa koadratikoa da eta Nivre-ren algoritmo proiektiboak, berriz, linealak.

[1] http://nextens.uvt.nl/depparse-wiki/DataOverview



4.2 irudia – EZB II zuhaitz-bankuko "lotat" erlazioa ez-proiektiboa duen esaldia.

- Arku ez-proiektiboak galdutzat ematea eta algoritmo proiektiboak soilik aplikatzea; adibidez, trantsizioetan oinarritutako Nivre-ren familiako algoritmo bakunak, efizienteak eta zehatzak.
- Algoritmo proiektiboak, baina teknika osagarriekin aplikatzea; adibidez, transformazio pseudo-proiektiboa[2] (Nivre eta Nilsson, 2005) edo *corrective modeling* (Hall eta Novák, 2005).

2007ko *CoNLL Shared Task* lehiaketan, EZB I zuhaitz-bankuan transformazio pseudo-proiektiboarekin puntu bateko hobekuntza lortu zen LAS neurrian. Transformazio pseudo-proiektiboa analisi sintaktikoan aplikatzeko gauzatu behar diren urratsak hauexek dira:

1. Entrenamenduko zuhaitz-bankuan dauden arku ez-proiektibo guztiak arku proiektibo bihurtzen dira, eta bihurketan egindako mugimenduak arku-etiketa konplexu batzuetan gordetzen dira.

2. Ikasketa fasean, ikasi behar diren arku-etiketak konplexuagoak eta ugariagoak dira.

3. Analisi fasean, ebaluatzeko esaldiei ikasitako arku proiektibo konplexuak ezarriko zaizkie.

4. Ebaluatzeko esaldiak sintaktikoki analizatu ostean agertzen diren arku proiektibo konplexuak arku ez-proiektibo bihurtuko dira, arku proiektibo konplexuetan kodifikatutako mugimenduek gidatuko dituztela.

Hurrengo ataletan, EZB II zuhaitz-bankuan, transformazio pseudo-proiektiboaren eragina aztertuko da.

[2] http://w3.msi.vxu.se/%7ENivre/research/proj/0.2/doc/Proj.html

Esperimentuak

EZB zuhaitz-bankuetako arku ez-proiektiboekin, hiru eratara joka daiteke. Atal honetan, EZB II zuhaitz-bankuan, algoritmo proiektiboak transformazio pseudo-proiektiboarekin, algoritmo proiektiboak bakar-bakarrik eta algoritmo ez-proiektiboak erabiliz lortutako emaitzak aztertuko dira. Esperimentuak egiteko, 4.2 taulan agertzen diren MaltParser (LibLINEAR paketearekin) eta MSTParser sistemako hurrengo konfigurazioak erabili dira:

1. Algoritmo proiektiboak soilik: MaltParser sistemako Nivre Arku Estandarra eta MSTParser sistemako bigarren mailako algoritmoak
2. Algoritmo proiektiboak eta teknika osagarriak: MaltParser sistemako Nivre Arku Estandarra algoritmo proiektiboa transformazio pseudo-proiektiboarekin konbinatuta
3. Algoritmo ez-proiektiboak: MaltParser sistemako Stacklazy eta Covington, eta MSTParser sistemako bigarren mailako algoritmo ez-proiektiboak

Ebaluazioa

Jokaera ezberdinak	**Konfigurazioak**	**LAS**	**UAS**
Alg. proiektiboak soilik	Malt Nivre Arku Estandarra	77,98	82,84
	MST bigarren mailakoa	**79,98**	**85,96**
Alg. proiektiboa + teknika osagarria	Malt Nivre Arku Estandarra + Pseudo	**78,89**	**83,63**
Alg. ez-proiektiboak	MST bigarren mailakoa	**80,17**	**86,01**
	Malt Stacklazy	78,79	83,43
	Malt Covington	78,71	83,50

4.2 taula – EZB II zuhaitz-bankuko ebaluazio multzoarekin, algoritmo ez-proiektiboak, algoritmo proiektiboak soilik eta algoritmo proiektiboak teknika osagarriekin konbinatu ostean lortutako emaitzak (Pseudo = Pseudo-proiektiboa).

4.2 taulan, emaitzarik onena MSTParser sistemako bigarren mailako algoritmo ez-proiektiboa erabiliz lortzen da. MSTParser eta MaltParser sistemetako algoritmo ez-proiektiboek algoritmo proiektiboak baino emaitza hobeak ematen dituzte. Baina MaltParser sistemako Nivre Arku Estandarra algoritmo proiektiboa transformazio pseudo-proiektiboarekin batera konbinatzerakoan LAS neurrian, algoritmo proiektiboarekiko ia puntu bateko hobekuntza

lortzeaz gain, MaltParser sistemako algoritmo ez-proiektiboekin baino emaitza hobea lortzen da. Horrela, transformazio pseudo-proiektiboak analisian dependentzia-etiketa konplexuak gehitu eta analisia neketsuagoa egiten badu ere, EZB II zuhaitz-bankuan hobekuntza esanguratsua lortzen dela probatu da.

4.3 taulan, sistema desberdinekin arku proiektiboetan eta ez-proiektiboetan gertatzen den estaldura eta doitasuna ikus daiteke. Estaldura handiena MSTParser sistemako MST bigarren mailako algoritmo ez-proiektiboa erabiliz lortzen da, zuhaitz-bankuan dauden arku ez-proiektibo guztietatik, sistemak % 39,9 ez-proiektibo zuzen eman baititu. Bigarrenik, Nivre Arku Estandar algoritmoa transformazio pseudo-proiektiboarekin konbinatuta geratu da, urre-patroiko zuhaitz-bankuan dauden arku ez-proiektibo guztietatik, sistemak % 37,6 ez-proiektibo zuzen eman ditu. Doitasun handiena MaltParser sistemako Stacklazy erabiliz lortu da, sistemak emandako arku ez-proiektiboen % 46,4 zuzenak baitira. Bigarrenik, Nivre Arku Estandar algoritmoa transformazio pseudo-proiektiboarekin konbinatuta geratu da, sistemak emandako arku ez-proiektiboen % 45 zuzenak baitira. Gainerako sistemak erabiliz doitasuna nabarmen jaisten da. Estalduraren jaitsiera ez da hain nabarmena. Nahiz eta transformazio pseudo-proiektiboak analisian dependentzia-etiketa konplexuak gehitu, estaldura eta doitasuna hobetzen duela probatu da.

Testa	**Doitasuna**		**Estaldura**	
Konfigurazioa	**Ez-proiek.**	**Proiek.**	**Ez-proiek.**	**Proiek.**
Malt Nivre Arku Estandarra + Pseudo	**0,45**	**0,79**	0,37	0,78
Malt Nivre Arku Estandarra		0,78		0,78
MST bigarren mailako ez-proiek.	**0,37**	**0,80**	**0,39**	**0,80**
MST bigarren mailako proiek.		0,80		0,80
Malt Covington ez-proiek.	0,30	0,79	**0,36**	**0,79**
Malt Stacklazy ez-proiek.	**0,46**	**0,79**	0,35	0,79

4.3 taula – Sistema desberdinekin arku ez-proiektiboetan eta proiektiboetan lortutako estaldura eta doitasuna (Pseudo = Pseudo-proiektiboa).

4.1.2 Sintagmen transformazioa

Sintagmen transformazioa egiteko bi arrazoi nagusi kontuan hartu dira:

– Euskarako zuhaitz-bankuak (EZB I eta II zuhaitz-bankuak) hitz mailan etiketatuta daude, hau da, dependentzia-erlazio sintaktikoak hitzen

artean etiketatu dira (ikusi 4.3a irudia). Euskara hizkuntza eranskaria eta morfologikoki aberatsa izanik (ikusi 4.4 taulan ezaugarriak zutabea), dependentzia-erlazio sintaktikoak hitzaren azken morfema banatu ostean etiketatu dira, ikasketa automatikoan mesedegarria izan daitekeelakoan. Adibidez, euskara bezala eranskaria eta morfologikoki aberatsa den turkieran, Oflazer-ek (1996) turkierako zuhaitz-bankuan dependentzia-erlazioak hitz mailan etiketatu beharrean, unitate morfosintaktikoak diren hitz-zatietan (ingelesez, *inflectional groups (IGs)*) etiketatu ostean hobekuntza esanguratsuak lortu zituen.

– Euskarako zuhaitz-bankuak (EZB I eta II zuhaitz-bankuak) etiketatzerakoan sintagmen buru bezala esanahi semantikoa duen hitza jarri da. 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, hitzaren azken morfema banatu ostean, sintagmen burua, buru semantikoa izan beharrean, sintagmen azkeneko kokagunea duen hitzaren morfema printzipala izango da. Horrela, sintagmen buruak ezaugarri morfosintaktikoak dituenean, aditzarekin lotzerako mementuan dependentzia-erlazio zuzena asmatzeko lagungarri izan daitekeelakoan gaude.

Zk.	Hitza	Lema	Kat.	Azpikat.	Ezaugarriak	Buru	Erlazioa
1	buru	buru	IZE	ARR	_	3	ncsubj
2	ohiak	ohi	ADJ	ARR	KAS:ERG|NUM:S	1	ncmod
3	aztertuko	aztertu	ADI	ADI_SIN	ASP:GERO	0	ROOT
4	du	*edun	ADL	ADL	MDN:A1|NOR:HURA|NORK:HARK	3	auxmod

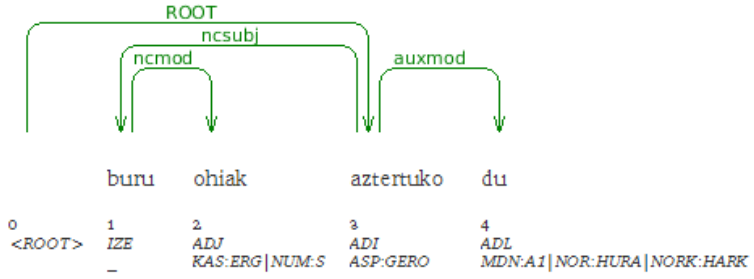
4.4 taula – EZB II zuhaitz-bankuko esaldi zatia *CoNLL-X* formatuan.

Sintagmen transformazioa gauzatzerakoan, hitz-markak dituzten hitzak bere lema eta morfema printzipalean banatu ostean, morfema printzipalak sintagmaren barruan hartzen duen kokagunearen arabera maila bat edo maila bi lemaren gainetik igotzeko transformazioa aplikatu da.

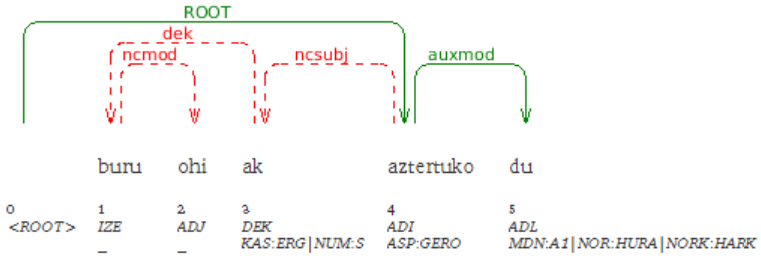
Maila biko igoera

Oro har, maila biko igoera aplikatzeko morfema printzipalak sintagmaren barruan azkeneko kokagunea hartzen duen morfema izan behar du (ikusi salbuespenak 4.1.2 atalean). Sintagmaren azkeneko kokagunean dauden morfema printzipalak sintagmaren buru jarriko dira. Adibidez, 4.3b irudian, "buru ohi ak" sintagmaren azkeneko kokagunean dagoen "ak", esanahi semantikoa

duen “buru” hitzaren gainetik jartzeko, hurrengoa ikus daiteke: alde batetik, “ohi” lemak “ohiak” hitzaren erlazioa (“ncmod”) eta gobernatzailea (“buru”) hartzen ditu; beste alde batetik, “ak” morfema printzipalak “ohiak” hitzaren gobernatzailearen erlazioa (“ncsubj”) eta gobernatzailea (“aztertuko”) hartzen ditu; eta, azkenik, “buru” sintagmaren gobernatzaile semantikoak “dek” erlazioa eta “ak” gobernatzailea hartzen ditu.



(a) hitz mailan etiketatua



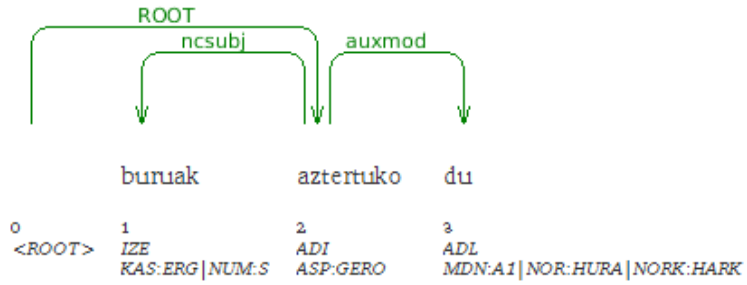
(b) morfema printzipala sintagmaren buru

4.3 irudia – 4.4 esaldiaren maila biko sintagmen transformazioa, 4.3a irudian EZB II zuhaitz-bankuko esaldia moldatu baino lehen, 4.3b irudian maila biko sintagmen transformazioa egin ostean.

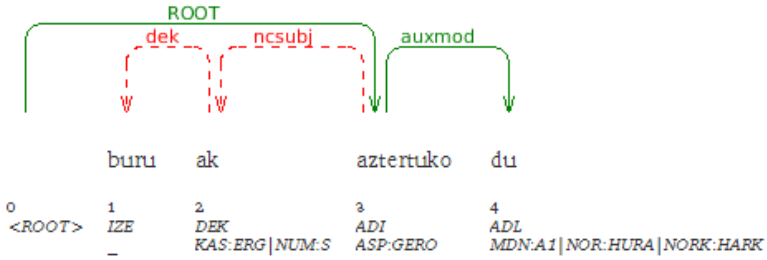
Maila bateko igoera

Morfema maila bi igotzen ez denean maila bat igoko da. Oro har, morfema printzipala buru-semantikoaren morfema denean edo sintagmaren barruan morfema honen eskuinaldean beste morfemaren bat dagoenean, maila bat igoko da. Maila bateko igoeran, kasu-marka duen hitza bere lema eta morfema printzipalean banatu ostean, morfema lemaren gobernatzaile jarriko da.

Adibidez, 4.4b irudian, "buruak" sintagmaren osagai bakarra izanik eta buru sintaktikoa eta semantikoa bat datozela ikusirik, ez du zentzurik "ak" bi maila igotzea ("aztertuko" aditzaren gainetik). Horrela, "ak" nahiz eta sintagmaren azkeneko kokagunean egon, maila bateko transformazioa gauzatuko da. Maila bateko igoeraren osteko 4.4b irudian ikus daiteke, alde batetik, "ak" morfemak "buruak" hitzaren erlazioa ("ncsubj") eta gobernatzailea ("aztertuko") hartzen dituela, eta, beste alde batetik, "buru" lemak "dek" erlazioa eta "ak" gobernatzaile hartzen dituela.



(a) hitz mailan etiketatua



(b) morfema printzipala lemaren gainetik maila bat igotzea

4.4 irudia – 4.4 esaldiaren maila bateko igoera; 4.4a irudian EZB II zuhaitz-bankuko esaldia moldatu baino lehen eta 4.4b irudian maila bateko igoera egin ostean.

Maila biko eta maila bateko igoerak gauzatu ostean, lortu diren 4.3b eta 4.4b irudiak oso antzekoak direla ikus daiteke. Orokortze honen ondorioz antzeman daitekeenez, sistemak "ncsubj" bezalako erlazioak errazago ikasteko eta hobeto iragartzeko balioko du.

Salbuespenak

Sintagmen transformazioan, morfema printzipala zuhaitz-sintaktikoan lemaren gainetik maila bat edo bi igotzea hitzak bere buruarekin duen erlazioaren eta morfemaren kokagunearen arabera egin da. Hartutako erabakiak hauexek izan dira:

- Jatorrizko hitzak "ncmod" edo "detmod" dependentzia-erlazioa izanez gero, morfema printzipala bi maila igotzea erabaki da, hurrengo salbuespenak kontuan hartuz:
  - Zuhaitz-sintaktikoan tratatzen ari garen hitza, kasua duen osteko senideren bat izanez gero, eta gurasoarekin "detmod" edo "ncmod" erlazioren bat izanez gero.
  - Zuhaitz-sintaktikoan tratatzen ari garen hitzaren gurasoa ROOT izanez gero.
  - Zuhaitz-sintaktikoan tratatzen ari garen hitzaren gurasoa kasua izanez gero.
  - Zuhaitz-sintaktikoan tratatzen ari garen hitzaren aitonak morfema izanez gero.
  - Zuhaitz-sintaktikoan tratatzen ari garen hitzaren gurasoa aditza izanez gero.
- Jatorrizko hitzak izan dezakeen gainerako erlazioekin morfema printzipala beti maila bat igotzeko erabakia hartu da.

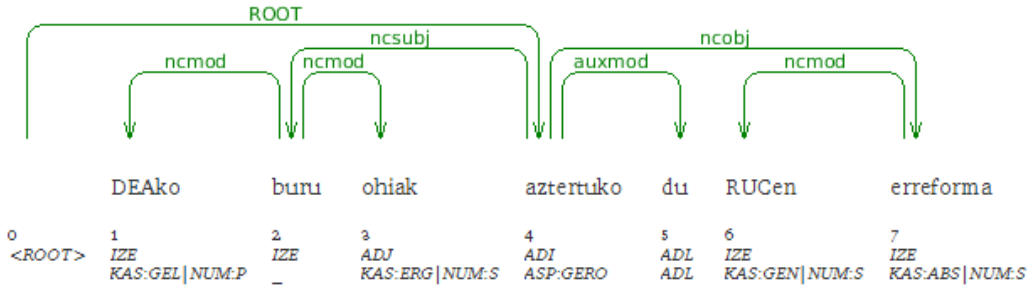
Salbuespenak aztertzeko, 4.5 taulan dagoen EZB II zuhaitz-bankuko esaldia aztertuko da. Adibidez, 4.5a eta 4.5b irudian, sintagmen transformazioa egin

Zk.	Hitza	Lema	Kat.	Azpikat.	Ezaugarriak	Buru	Erlazioa
1	DEAko	DEA	IZE	LIB	ENT:Er|KAS:GEL|NUM:P	2	ncmod
2	buru	buru	IZE	ARR	_	4	ncsubj
3	ohiak	ohi	ADJ	ARR	KAS:ERG|NUM:S	2	ncmod
4	aztertuko	aztertu	ADI	ADI_SIN	ASP:GERO	0	ROOT
5	du	*edun	ADL	ADL	MDN:A1|NOR:HURA|NORK:HARK	4	auxmod
6	RUCen	Ruc	IZE	LIB	ENT:Er|KAS:GEN|NUM:S	7	ncmod
7	erreforma	erreforma	IZE	ARR	KAS:ABS|NUM:S	4	ncobj

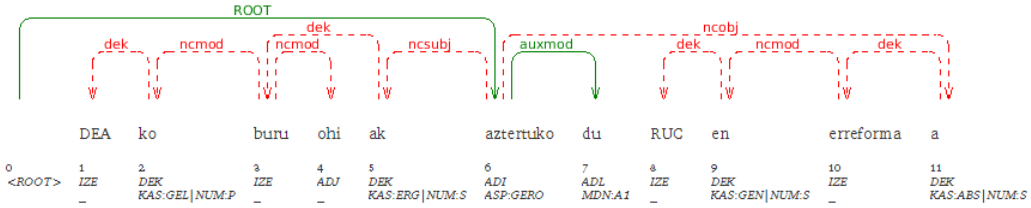
4.5 taula – EZB II zuhaitz-bankuko esaldia *CoNLL-X* formatuan.

aurretik eta egin ostean nola geratu den ikus daiteke.

4.5b irudian, kasu-marka dituzten "DEAko", "ohiak", "RUCen" eta "erreforma" hitzen lemak eta morfema printzipalak banatuta agertzen dira. Morfema printzipalak eta lemak banatu ostean, bakoitzak berari dagokion ezaugarriak hartuko ditu, adibidez, "ak" morfemak "KAS:ERG|NUM:S" ezaugarriak hartuko ditu. Baina gobernatzailea eta dependentzia-erlazioaren ba-



(a) hitz mailan etiketatua



(b) transformazioaren ostean

4.5 irudia – 4.4 esaldiaren sintagmen transformazioaren ostean, 4.5a irudian EZB II zuhaitz-bankuko esaldia moldatu baino lehen, 4.5b irudian maila bateko eta biko igoerak aplikatu ostean.

lioen esleipena adibidearen arabera egin da. Adibidez, "DEAko buru ohiak" sintagman, "DEAko" eta "ohiak" hitzak "ncmod" erlazio berbera daukate, baina egoera ezberdinak ematen dira: batetik, "DEAko" hitzaren kasuan sintagmaren barruan kasu-marka duen osteko senideren bat izateagatik, "ko" morfema "DEA" lemaren gainetik maila bat igotzen dela ikus daiteke; bestetik, "ohiak" hitzaren kasuan, salbuespen guztiak ez betetzeagatik, "ak" morfema "buru" esanahi semantikoa duen hitzaren gainetik bi maila igotzen dela ikus daiteke. Bukatzeko, "RUCen erreforma" sintagmaren kasuan, bai "RUCen" eta bai "erreforma" hitzen morfemak maila bat igoko dira. "RUCen" kasuan, kasu-marka duen osteko senideren bat izateagatik, eta "erreforma" hitzaren kasuan "ncobj" maila bateko erlazio edo buru semantikoa izategatik, morfema printzipalak maila bat igo direla ikus daiteke.

Ebaluazioa

Aurreko ataletan sintagmen transformazioa azaldu da eta 4.6 taulan, sintagmen transformazioa algoritmo desberdinekin probatu ostean lortutako emai-

tzak daude.

Konfigurazioak	**Oinarria**		**Sintagmen transf.**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	77,98	82,84	77,94 (-0,04)	82,57 (-0,27)
Malt Nivre Arku Estandarra + Pseudo	78,89	83,63	**80,35 (+1,46)**‡	85,14 (+1,51)
Malt Stacklazy ez-proiek.	78,79	83,43	**80,26 (+1,47)**‡	85,25 (+1,82)
Malt Covington ez-proiek.	78,71	83,50	**79,24 (+0,53)**‡	83,99 (+0,49)
MST bigarren mailakoa ez-proiek.	80,17	86,01	79,98 (-0,19)	85,93 (-0,08)

4.6 taula – Sintagmen transformazioa EZB II zuhaitz-bankuko urre-patroiko ezaugarriekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

4.6 taulan, emaitzarik onena Nivre Arku Estandar algoritmoa transformazio pseudo-proiektiboarekin konbinatuta erabiliz lortzen dela ikus daiteke, 80,35 (oinarriarekiko +1,46ko hobekuntza esanguratsua) LAS neurrian. Bigarren postuan, 80,26ko emaitzarekin *Stacklazy* algoritmo ez-proiektiboa dago, oinarriarekiko +1,47ko hobekuntza esanguratsuarekin. Emaitzarik txarrena, aldiz, Nivre Arku Estandarra algoritmo proiektiboarekin gertatzen da, 77,94ko emaitzarekin. Kasu honetan, sintagmen transformazioarekin esaldi ez-proiektiboen kopurua % 13,28tik % 26,97ra igotzea da jaitsiera honen arrazoi nagusia. Azkenik aipatu behar da MST sistemarekin ez dela hobekuntzarik lortzen.

MaltParser sistemako algoritmo ez-proiektiboekin urre-patroiko ezaugarriak erabiltzerakoan, hobekuntza esanguratsuak lortzen direla ikusi da, eta urre-patroiko ezaugarriak erabili beharrean, ezaugarri automatikoak erabiltzerakoan, ebaluaziorako multzoan zer gertatzen den aztertuko da.

Konfigurazioak	**Oinarria**		**Sintagmen transf.**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	76,14	81,60	76,22 (+0,08)	80,92 (-0,68)
Malt Nivre Arku Estandarra + Pseudo	77,31	82,62	**78,26 (+0,95)**‡	83,46 (+0,84)
Malt Stacklazy ez-proiek.	76,83	82,18	**77,96 (+1,13)**‡	83,20 (+1,02)
Malt Covington ez-proiek.	76,98	82,17	**77,75 (+0,77)**‡	82,63 (+0,46)
MST bigarren mailako ez-proiek.	77,88	84,08	78,26 (+0,38)†	84,62 (+0,54)

4.7 taula – Sintagmen transformazioa EZB II zuhaitz-bankuko ezaugarri automatikoekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

4.7 taulan, emaitzarik onenak bigarren mailako MST ez-proiektibo eta Nivre Arku Estandar algoritmoa transformazio pseudo-proiektiboarekin konbinatuta erabiliz, *LAS* neurrian 78,26 lortu dela ikus daiteke. Bigarren postuan, 77,96ko emaitzarekin *Stacklazy* algoritmo ez-proiektiboa dago, oina-

rriarekiko +1,13ko hobekuntza esanguratsuarekin. Emaitzarik txarrena, aldiz, Nivre Arku Estandarra algoritmo proiektiboarekin gertatzen da, 76,22ko emaitzarekin. Nahiz eta ezaugarri automatikoekin lortutako emaitzak urre-patroiko ezaugarriekin lortutakoak baino apalagoak izan, algoritmo ez-proiektiboekin lortutako hobekuntzak esanguratsuak dira.

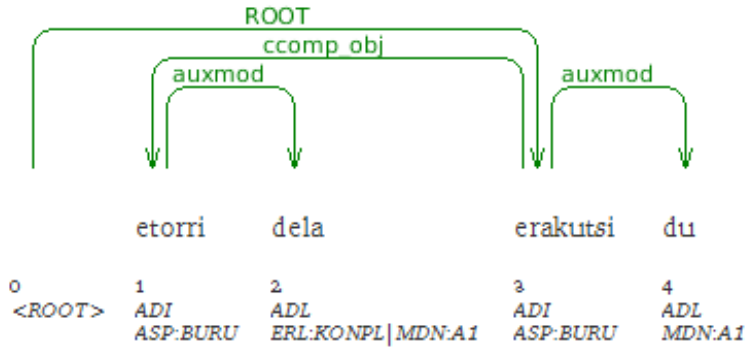
4.1.3 Mendeko perpausen transformazioa

Oro har, euskaraz, mendeko perpausak aditz nagusi edo aditz laguntzailearri morfema bat gehituz sortzen dira. 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.

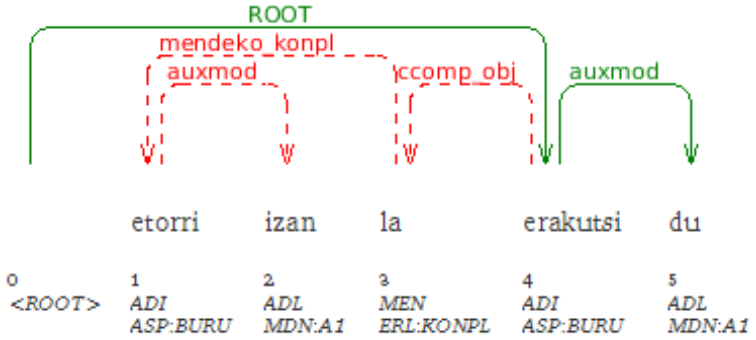
Euskarako zuhaitz-bankuetan (EZB I eta II zuhaitz-bankuetan), mendeko perpausak aditz nagusiaren (buru semantikoaren) inguruan antolatuta daude. Baina gehienetan mendeko perpausen morfema (buru sintaktikoa) aditz laguntzaileari lotuta etortzen da; hau da, ez dator bat buru semantikoarekin.

4.6a irudian, "etorri dela" mendeko perpausa konpletiboa "da" aditz laguntzaileari "la" mendeko konpletiboaren morfema gehituz sortu dela ikus daiteke. "dela" hitzean mendeko konpletiboaren informazio guztia dagoela ikus daiteke: ezaugarrien artean mendeko konpletiboaren informazioa ("ERL:KONPL") eta "la" mendeko morfema. Horrela, analizatzaileak mendeko dependentzia-erlazioa zein den asmatzea zailago izango du, alde batetik, "etorri" mendeko perpausaren aditz nagusiak "erakutsi" perpaus nagusiko aditz nagusiarekin lotzeko orduan ez duelako mendekoa denaren inolako informaziorik, eta bestetik, "dela" zuhaitz-sintaktikoan "etorri" hitza baino maila bat beherago dagoelako.

Sintagma bakunak eta mendeko perpausak hobeto bereizteko eta mendeko morfemaren informazio sintaktikoa bere burutik hurbilago egoteko, mendeko morfema bere aditz laguntzailetik banatu eta mendeko perpausaren buru jarriko da. Collins *et al.*-ek (1999) antzeko proposamena gauzatu zuten, *Prague Dependency Treebank*ean (Hajic, 1998), aditz-sintagma bakunen eta mendeko erlatibozko perpausak "VP" kategoria berdinarekin adierazi beharrean, mendeko erlatibozko perpausentzat "SBAR" kategoria berria erabili zutelako. Collins *et al.*-en (1999), moldaketak egitura-osagaietan eta hitzen zerrenda aldatu barik egin dira; haatik, gure moldaketak mendeko dependentzia-erlazio guztiekin (erlatibo, denbora, konpletibo, eta abarrekin) eta morfema hitz errotik banatuz egin dira.



(a) hitz mailan etiketatua



(b) mendeko morfema maila bi igotzen duen transformazioa

4.6 irudia – 4.6b irudian, 4.6a esaldiaren mendekoen transformazioa.

Mendeko perpausen transformazioa gauzatzerakoan, mendeko informazioa "ERL:" duten aditzak, beren lema eta mendeko morfema banatu ostean, mendeko morfema aditz nagusiari (buru semantikoari) edo aditz laguntzaileari (buru sintaktikoari) lotuta etorriz gero, lemaren gainetik maila bat edo maila bi igotzeko transformazioa aplikatu da.

Maila biko igoera

Oro har, maila biko igoera aplikatzeko mendeko morfema aditz laguntzaileari (buru sintaktikoari) lotuta egonez gero, mendeko morfema banatu ostean, morfema bi maila igoko da. 4.6b irudian, mendeko perpausen transformazioa ikus daiteke. Esan bezala, mendeko perpausaren burua den "etorri" aditzak ez darama inolako mendeko informaziorik. Horrela, analizatzaileak esaldi

nagusiarekin nola lotu behar den jakiteko, maila bat beherago dagoen "dela" aditz laguntzailean dagoen mendeko informazioa behar du. Analizatzailearen zehaztasuna hobetzeko asmoarekin, "dela" hitzari "la" morfema konpletiboa banatu ostean, "etorri" aditz nagusiaren gainetik dependentzia-zuhaitzean bi maila igoko da. Horretarako, "etorri" aditzaren buru "la" morfema eta euren arteko erlazio-etiketa mendeko_konpl dependentzia-erlazio berria jarriko da. "la" morfemak lehen "etorri" aditz nagusiak zituen burua eta erlazio-mota hartuko ditu, hau da, "erakutsi" buru eta "ccomp_obj" erlazio-mota hartuko ditu.

Maila bateko igoera

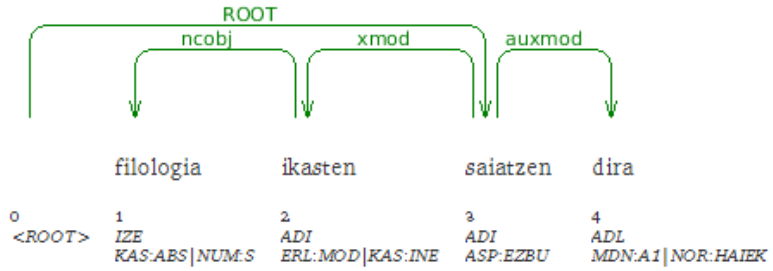
Baina aditz nagusiak mendeko morfema izanez gero, mendeko morfema banatu eta zuhaitzean mendeko morfema maila bat baino ez da igoko. Adibidez, 4.7a irudian, mendeko perpausa modala den esaldian, mendeko morfema eta mendeko informazioa aditz nagusian kokatuta daudela ikus daiteke. Horrela "ikasten" aditz nagusiak, mendeko perpausaren buruak, mendeko informazioa eraman eta esaldi nagusiarekin nola lotu behar den jakiteko morfema banatu ostean maila bat igoko da. Hori dela eta, "ikasten" hitzari "ten" modala den morfema banatu eta "ikasi" aditz nagusiaren gainetik ipiniko da, dependentzia-zuhaitzean morfema maila bat igoz. "ikasi" aditzaren buruari "ten" morfema jarriko zaio eta euren arteko erlazio-etiketa "mend_mod" dependentzia-erlazio berria jarriko da. "ten" morfemak "ikasi" aditz nagusiak duen burua eta erlazio-mota hartuko ditu, hau da, "saiatzen" burua eta "xmod" erlazio-mota hartuko ditu (ikusi 4.7b irudian).

Azkenik, analizatzaile sintaktikoaren irteerari alderantzizko moldaketa aplikatuko zaio; hau da, banatutako morfema bakoitzari aplikatuko zaio:

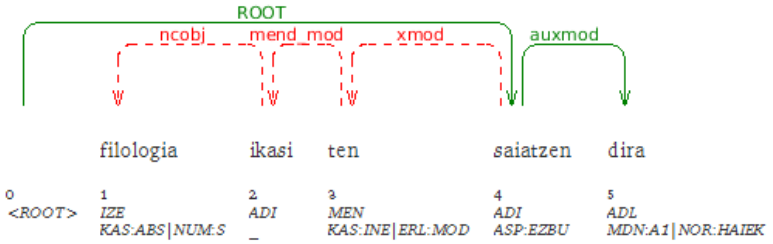
1. Hitza morfemarekin batu eta morfemaren berezitasunak jatorrizko hitzean berriro sartzea
2. Morfemaren erlazioa jatorrizko hitzari pasatzea
3. Morfemari sintaktikoki lotuta dauden hitzak jatorrizko hitzari lotzea

Ebaluazioa

Esperimentuak urre-patroiko ezaugarriekin eta ezaugarri automatikoekin probatu dira. 4.8 taulan, urre-patroiko ezaugarriekin, mendeko sintagmen transformazioa algoritmo ezberdinekin probatu ostean lortutako zehaztasuna ikus daiteke.



(a) hitz mailan etiketatua



(b) mendeko morfema maila bat igotzen duen transformazioa

4.7 irudia – 4.7b irudian, 4.7a esaldiaren mendekoen transformazioa.

Konfigurazioak	**Oinarria**		**Mendeko transf.**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	77,98	82,84	78,16 (+0,18)	82,95 (+0,11)
Malt Nivre Arku Estandarra + Pseudo	78,89	83,63	**79,12 (+0,23)**	**83,95 (+0,32)**
Malt Stacklazy ez-proiek.	78,79	83,43	78,94 (+0,15)	83,69 (+0,26)
Malt Covington ez-proiek.	78,71	83,50	**79,06 (+0,35)**	**83,86 (+0,36)**
MST bigarren mailako ez-proiek.	80,17	86,01	80,17 (+0,00)	86,02 (+0,01)

4.8 taula – Mendeko perpausen transformazioa EZB II zuhaitz-bankuko urre-patroiko ezaugarriekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

4.8 taulan ikusten denez, mendekoen transformazioaren eragina oso ahula da. MSTParser sistemako bigarren mailako algoritmoarekin izan ezik, gainerako kasu guztietan LAS hobetzen dela ikus daiteke. Mendeko perpausen transformazioa gauzatu ostean, esaldi ez-proiektiboen kopurua % 13,28tik % 20,02ra eta arku ez-proiektiboen kopurua %1,18tik 1,83ra igotzen dela ikusi da.

4.9 taulan, ezaugarri automatikoekin mendeko perpausen transformazioak algoritmo ezberdinekin duen eragina ikus daiteke. Transformazio ho-

Konfigurazioak	**Oinarria**		**Mendeko transf.**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	76,14	81,60	76,42 (+0,28)	81,52 (-0,08)
Malt Nivre Arku Estandarra + Pseudo	77,31	82,62	77,49 (+0,18)	82,74 (+0,12)
Malt Stacklazy ez-proiek.	76,83	82,18	77,05 (+0,22)	82,13 (-0,05)
Malt Covington ez-proiek.	76,98	82,17	**77,56 (+0,58)‡**	**82,64 (+0,47)**
MST bigarren mailako ez-proiek.	77,88	84,08	**78,22 (+0,34)**	**84,30 (+0,22)**

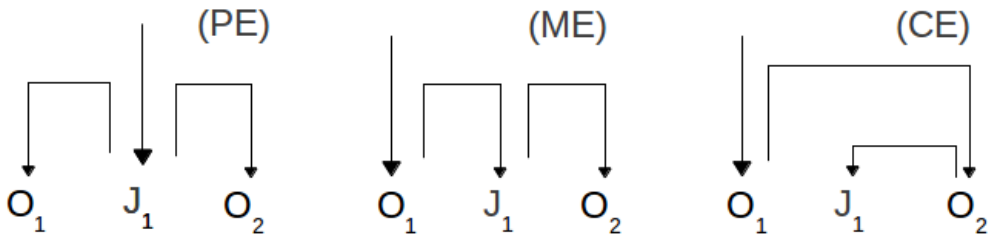
4.9 taula – Mendeko perpausen transformazioa EZB II zuhaitz-bankuko ezaugarri automatikoekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

nek kasu errealetan aplikagarritasuna duen jakiteko, ezaugarri automatikoak erabiltzerakoan mendeko perpausen transformazioak zehaztasunean izan dezakeen eragina aztertuko da. 4.9 taulan, algoritmo guztiekin hobetzen dela ikus dezakegu. Covington algoritmo ez-proiektiboarekin lortzen da emaitzarik onena, +0,58 hobekuntza esanguratsuarekin. Aipatu beharra dago, oro har, ezaugarri automatikoekin urre-patroiko ezaugarriekin baino hobekuntza hobeak lortu direla.

4.1.4 Koordinazioaren transformazioa

Koordinazioa edo juntadura maila bereko edo estatus gramatikal bereko perpausak nahiz sintagmak lotzen direnean gertatzen da. Koordinazioan elkartzen diren osagaiei juntagai deituko zaie, eta elkartze hori gauzatzeko erabiliko den tresna gramatikala juntagailua izango da. Hitzen artean erlazio bitarra argia denean, dependentzietan oinarritutako teoriek ondo baino hobeto lan egiten dute. Baina koordinazioaren kasuan, juntagailuaren eta juntagaien artean erlazio bitarrak sortzeko orduan, teoria edo estilo ezberdinak sortu dira (ikusi 4.8 irudia):

- Praga estiloa (PE)(Böhmová *et al.*, 2003): juntagailua koordinazioa osatzen duten juntagaien burua da.
- Mel'cuk estiloa (ME) (Mel'čuk, 1988): koordinazioaren lehenengo juntagaia koordinazioaren buru jartzen da eta gainerako osagaiak aurreko osagaiarekin lotzen dira, kate bat osatuz.
- Collins estiloa (CE)(Collins *et al.*, 1999): juntagailua kenduta, juntagaien artean erlazio zuzena dago, eta juntagailua eskuinen dagoen juntagaiarekin lotzen da. Ezker alderago dagoen juntagaia koordinazioaren buru izango da.

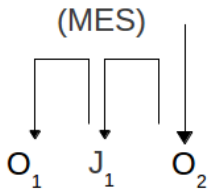


4.8 irudia – Koordinazio estiloak. J_1 juntagailua eta O_1 eta O_2 juntagaiak.

Teoria batzuk analisi semantikoan daude oinarrituta eta, beste batzuk, berriz, analisi sintaktikoan. Tesi-lan honetan ez da teoria ezberdinen eztabaidan sartu nahi, EZB zuhaitz-bankuetan gauzatutako zuhaitz-transformazioekin lortutako emaitzak plazaratzea baizik, euskararen analisi sintaktikoaren zehaztasuna hobetzeko helburuarekin.

Antzeko lan ugari egin dira, adibidez, Nilsson *et al.*-ek (2006) probatu zuten txekierako PDT zuhaitz-bankuan (*Prague Dependency Treebank*) dauden koordinazio eta aditz-taldeak Praga estilotik Mel'cuk estilora transformatu ondoren, MaltParser sistemek hobeto ikasteko balio zuela. Orokortze aldera, (Nilsson *et al.*, 2008b) lanean, koordinazio, aditz-talde eta proiektibizazio-transformazioak zuhaitz-banku eta analizatzaile desberdinekin probatu zituzten. Zehazki, 2006ko *CoNLL-X* lehiaketan parte hartutako txekiar, nederlandera, esloveniar eta arabiera hizkuntzatako zuhaitz-bankuekin eta MaltParser eta MSTParser sistemak erabilita probatu zituzten. Horrela, txekiar, arabiera eta esloveniar hizkuntzetako zuhaitz-bankuak Praga estilotik Mel'cuk estilora eta nederlandera hizkuntzakoa Praga estilotik Collins estilora transformatu ostean, MaltParser sistemak zuhaitz-banku guztietan igoera esanguratsua lortu zuten UAS neurrian transformazio pseudo-proiektiboarekin konbinatuta. Salbuespen bakarra txekiera hizkuntzakoa izan zen, hain zuzen, koordinazioaren transformazioarekin eta MSTParser sistemako algoritmo ez-proiektiboarekin UAS neurrian puntu bateko jaitsiera izan baitzuten. Gureari dagokionez, EZB zuhaitz-bankuetako esaldi koordinatuak Praga estiloa jarraituz etiketatu dira. Euskara hizkuntza buru-azkena izanik eta berezitasun morfologiko garrantzitsuenak sintagmaren azken juntagaian kokatuta daudela jakinda, Mel'cuk Estiloaren Simetrikoa (hemendik aurrera, MES izenarekin adieraziko da) aplikatu da. MESaren koordinazioaren az-

ken juntagaia koordinazioaren buru jarriko da eta gainerako osagaiak osteko osagaiarekin lotuko dira, kate bat osatuz (ikusi 4.9 irudia).



4.9 irudia – Mel'cuk Estiloaren Simetrikoa (MES). J_1 juntagailua eta O_1 eta O_2 juntagaiak.

MES estiloa euskararako intuitiboki egokiagoa dela ikus daiteke 4.10a eta 4.10b irudietan. "Luze eta gogorra" sintagma koordinatuan, kasu-marka eta numeroa azkeneko "gogorra" juntagaian baino ez daudela ikus daiteke; hori dela-eta, "gogorra" juntagaia koordinazioaren buru jartzerakoan "izan" aditz nagusiarekiko hurbilago egongo da eta analizatzaileak "gogorra" eta "izan" hitzaren eta aditzaren arteko "ncpred" erlazioa asmatzeko behar duen informazio morfologiko guztia eskura izango du. Hurrengo atalean, hobekuntza handienak MES estiloarekin lortu direla ikusiko da, eta estiloa bera deskribatuko da.

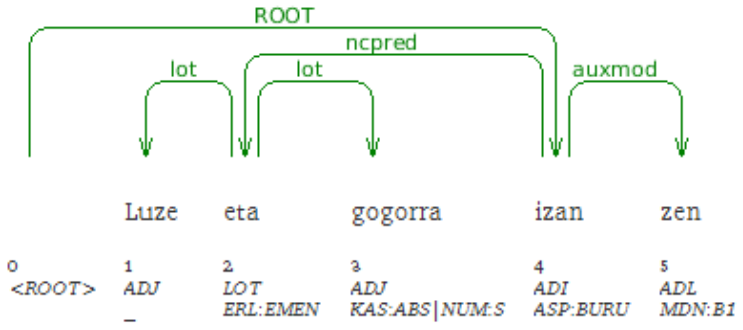
Mel'cuk Estiloaren Simetrikoa (MES)

Atal honetan, PE estilotik MES estilora pasatzeko egindako transformazioa nola gauzatu den azalduko da. Horretarako, 4.10a irudian, PE estiloan agertzen den esaldia erabiliko da. Hurrengo pausoak jarraitu dira:

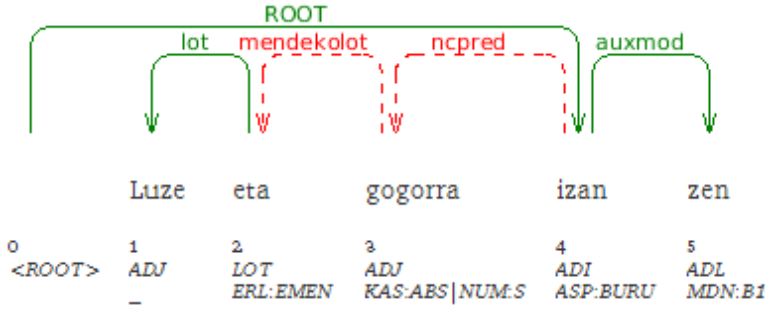
1. Koordinazioa osatzen dituzten osagaiak, juntagailuak eta juntagaiak identifikatu dira. Adibidez, 4.10a irudian, "eta" juntagailua eta gainerako "Luze" eta "gogorra" juntagai umeak identifikatu dira.

2. Tratatuko diren juntagailuak identifikatu dira. Adibidez: *eta*, *edo*, *ala*, eta abar.

3. Azkenik, azkeneko juntagai umeari koordinazio buruaren dependentzia-erlazioa eta burua jartzen zaizkio, eta juntagailua azkeneko juntagai umearekin lotzen da, "mendekolot" dependentzia-erlazio berri batekin. Adibidez, 4.10b irudian ikus daitekeenez, "gogorra" hitzari (azkeneko



(a) Koordinazioa Praga estiloan etiketatua



(b) Koordinazioa MES estilora transformatu ostean

4.10 irudia – koordinazio-transformazioa, PE estilotik MES estilora.

juntagai umeari), "ncpred" koordinazio buruaren dependentzia-erlazioa eta "izan" burua jartzen zaizkio, eta "eta" juntagailua "gogorra" hitzarekin lotzen da, "mendekolot" dependentzia-erlazioarekin.

Analisi sintaktikoaren ostean eta ebaluatu aurretik, alderantzizko transformazioa egin da. Alderantzizko transformazio egiteko hurrengo urratsak jarraitu dira:

1. Hitz batean "mendekolot" erlazioa aurkitu eta, hitz horren gurasoaren burua eta dependentzia-erlazioa lortu ostean, "mendekolot" erlazioa duen hitzari jarriko zaizkio. 4.10b irudian ikus daitekeenez, "mendekolot" erlazioa duen "eta" aurkituta, "eta" hitzaren gurasoa den "gogorra" hitzaren "izan" burua eta "ncpred" dependentzia-erlazioa lortu ostean, "eta" hitzari jarriko zaizkio aurreko 4.10b irudira bueltatzeko.

2. “mendekolot” burua duen “gogorra” hitzari “lot” dependentzia-erlazioa jarriko zaio eta burutzat “mendekolot” erlazioa duen “eta” hitza ezarriko zaio.

Esperimentuak

Koordinazio-transformazioek euskarako zuhaitz-bankuetan izan dezaketen eragina aztertzeko lan ezberdinak gauzatu dira. EZB I zuhaitz-bankua erabiliz, (Bengoetxea eta Gojenola, 2009b) lanean, Praga eta Mel'cuk estiloaz gain, Mel'cuk Estiloaren Simetrikoa (MES) aplikatzea probatu zen. EZB I zuhaitz-bankuko esperimentuak egiteko, MaltParser 0.4 bertsioarekin zetorren Nivre Arku Azkarra algoritmoa eta *SVM*ko sailkatzailea[3] baliatu ziren. EZB I zuhaitz-bankuko tamaina txiki samarra dela-eta, 10 geruzako balidazio gurutzatua (ingelesez, *10-fold cross-validation*) teknika erabili zen, eta azkeneko proba test multzoarekin egin zen. Tesi-lan honetan, EZB II zuhaitz-bankuan MES transformaziorik onenak algoritmo eta sistema desberdinetan duen eragina jakiteko, MaltParser eta MSTParser sistemak erabili dira. Probak egiteko, EZB II zuhaitz-bankuaren tamaina handiagoa izanik, euskarako zuhaitz-bankua hiru zati desberdinetan banatu da: entrenamendurako % 80, garapenerako % 10 eta ebaluaziorako % 10. Esperimentu guztiak garapenerako datuekin egin dira, eta sistemarik onena ebaluaziorako azpimultzoarekin probatzeko utzi da.

Ebaluazioa

EZB I zuhaitz-bankuan, ME eta MES transformazioak gauzatu ostean, 10 geruzako balidazio gurutzatuarekin eta proba multzoarekin lortutako emaitzak ikus daitezke 4.10 taulan.

Koordinazioa Estiloak	**10-fold cross-validation (LAS)**	**Testa (LAS)**
PE	76,15	74,52
ME	72,05 (-4,10)	69,99 (-4,53)
MES	76,43 (+0,28)	**75,25 (+0,73)**‡

4.10 taula – koordinazio-transformazioak EZB I zuhaitz-bankuko urre-patroiko ezaugarriekin ebaluatzeko multzoan.

Emaitzarik onena Mel'cuk Estiloaren Simetrikoarekin lortu zen, Praga estiloaren aldean 0,73ko gehikuntza erdietsi baitzen LAS neurrian. Hobe-

[3] *SVM*ko bigarren graduko *kernel* polinomikoaren parametroak: -s 0 -t 1 -d 2 -g 0,2 -c 0,4 -r 0 -e 0,1 -S 0

kuntza honen arrazoien artean, besteak beste, dependentzia-erlazioen batez besteko distantziak gehien gutxitzen duen aukera izatea dago.

Konfigurazioak	**PE**		**MES**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	77,98	82,84	**78,52 (+0,54)‡**	83,27 (+0,43)
Malt Nivre Arku Estandarra + Pseudo	78,89	83,63	79,19 (+0,30)	83,94 (+0,31)
Malt Stacklazy ez-proiek.	78,79	83,43	79,10 (+0,31)	83,83 (+0,40)
Malt Covington ez-proiek.	78,71	83,50	**79,23 (+0,51)‡**	83,98 (+0,48)
MST bigarren mailako ez-proiek.	80,17	86,01	80,50 (+0,33)	86,11 (+0,10)

4.11 taula – koordinazio-transformazioak EZB II zuhaitz-bankuko urre-patroiko ezaugarriekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

EZB II zuhaitz-bankuan MES transformaziorik onenak algoritmo eta sistema desberdinetan duen eragina ikus daiteke 4.11 taulan. Algoritmo eta sistema guztietan, MES transformazioak 0,30etik gorako gehikuntzak lortu ditu. Nahiz eta emaitzarik onena MSTParser sistemako bigarren mailako algoritmoarekin lortu % 80,50 LAS neurrian, hobekuntza handiena lortu duen algoritmoa Nivre Arku Estandarra algoritmo proiektiboa izan da, 0,54ko hobekuntza esanguratsuarekin. Izan ere, EZB II zuhaitz-bankuan arku ez-proiektiboen kopurua ez da igotzen MES koordinazioaren transformazioa gauzatu ostean.

Transformazio honek kasu errealetan aplikagarritasuna duen jakiteko, ezaugarri automatikoak erabili dira. 4.12 taulan ikusten denez, ezaugarri automatikoak erabiltzerakoan urre-patroiko ezaugarriekin baino emaitza hobeak lortzen dira. Hemen ere, emaitzarik onena MSTParser sistemako bigarren mailako algoritmoarekin lortzen da, LAS neurrian % 78,27rekin, hau da, 0,39ko gehikuntza. Baina hobekuntza handiena MaltParser sistemako Nivre Arku Estandarra algoritmo proiektiboan lortu da, 1,01eko igoerarekin. MES koordinazioaren transformazioarekin algoritmo eta sistema guztietan gutxienez, 0,39ko gehikuntza lortu da LAS neurrian.

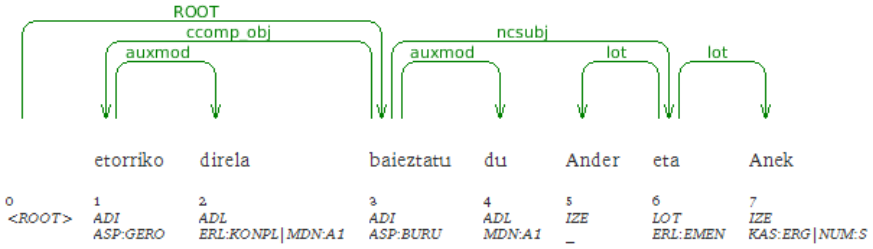
4.1.5 Transformazioen konbinaketa

Sintagmen, mendeko perpausen eta koordinazioaren transformazioak bateragarriak eta mesedegarriak diren jakiteko, bata bestearen ondoren aplikatuko dira transformazioak. 4.11a eta 4.11b irudietan, transformaziorik gabeko esaldiak eta aipatutako hiru transformazioak aplikatu osteko esaldiak ikus daitezke, hurrenez hurren. Moldaketa hauek EZB II zuhaitz-bankuan izan

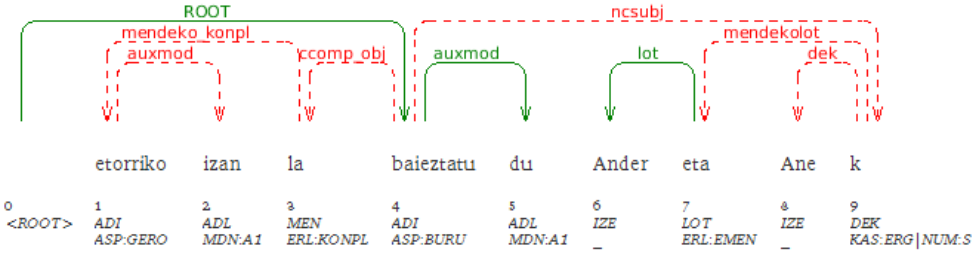
Konfigurazioak	**PE**		**MES**	
Testa	**LAS**	**UAS**	**LAS**	**UAS**
Malt Nivre Arku Estandarra	76,14	81,60	**77,15 (+1,01)‡**	82,48 (+0,88)
Malt Nivre Arku Estandarra + Pseudo	77,31	82,62	**78,01 (+0,70)‡**	83,15 (+0,53)
Malt Stacklazy ez-proiek.	76,83	82,18	**77,68 (+0,85)‡**	82,86 (+0,68)
Malt Covington ez-proiek.	76,98	82,17	**77,77 (+0,79)‡**	82,83 (+0,66)
MST bigarren mailako ez-proiek.	77,88	84,08	78,27 (+0,39)	84,37 (+0,29)

4.12 taula – koordinazio-transformazioak EZB II zuhaitz-bankuko ezaugarri automatikoekin ebaluatzeko multzoan (Pseudo = Pseudo-proiektiboa).

duten eragina aztertzeko orduan, hurrengo 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 konbinaketa 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.



(a) Transformazio gabeko esaldia



(b) Transformazio guztiak egin osteko esaldia

4.11 irudia – Transformazioen konbinaketa.

Esperimentuak
Transformazioen konbinaketak sistema eta algoritmo desberdinekin duen eragina aztertzeko, MaltParser eta MSTParser sistemekin probatu da EZB II zuhaitz-bankuan. Transformazioak aplikatzerakoan jarraitutako ordena hurrengoa izan da: lehenengo, sintagmen transformazioa gauzatu da, gero, mendekoen transformazioa eta, azkenik, koordinazioaren transformazioa gauzatu da. Transformatutako multzoarekin entrenatu ostean, sistemak analizatzaile sintaktiko bat sortuko du, eta proba multzoko esaldien morfemak banatu ostean eta ikasitako analizatzaile sintaktikoarekin analizatu ostean, analizatzailearen irteerari alderantzizko transformazioak aplikatuko zaizkio, alderantzizko ordena jarraituz; hala, bada, lehenengo, koordinazioaren transformazioa aplikatuko da, gero, mendekoen transformazioa eta, azkenik, sintagmen transformazioa. Bukatzeko, alderantzizko transformazioen irteera ebaluatuko da. Esperimentuak egiteko urre-patroiko ezaugarriak eta ezaugarri automatikoak erabiltzeaz gain, zuhaitz-bankuaren tamaina bikoizten denean, bai oinarriarekin, bai aplikatutako transformazioekin, analisian duen eragina aztertuko da. Horretarako, EZB II zuhaitz-bankuko % 80 den entrenamendu multzoa 500, 1.000, 2.000, 4.000 eta 9.100 esaldiko azpimultzoetan 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.

Ebaluazioa
EZB II zuhaitz-bankuko esaldiak transformatu barik sortutako oinarrizko 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) lortutako emaitzak 4.13 eta 4.14 tauletan agertzen dira, hurrenez hurren.

Laburbilduz, 4.15 tauletan, EZB II zuhaitz-bankuko transformazio bakoitza banan-banan konbinatuta edo guztiak konbinatuta eta bai urre-patroiko ezaugarriak, bai ezaugarri automatikoak erabiltzerakoan lortutako gehikuntzak konparatuko dira.

Tauletako datuak aztertzerakoan, hurrengo emaitzak aipa daitezke:

– **Ezaugarri automatikoak vs urre-patroiko ezaugarriak**. Analizatzaile morfologikoak automatikoki lortutako ezaugarri morfosintaktikoak erabiltzerakoan, ia 2 puntuko galera dagoela ikus daiteke sistema

UE	Stacklazy		Covington		Niv. Arku Est. + Pse.		MST	
Kop.	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)**

4.13 taula – Oinarrizko sistemak (Oin.) eta hiru zuhaitz-transformazioak (ZT), urre-patroiko ezaugarriekin ebaluatzeko multzoan. Emaitza guztiak estatistikoki esanguratsuak dira; salbuespena MST 9.100 da (Niv. Arku Est. + Pse.= Nivre Arku Estandarra + Pseudo-proiektiboarekin).

EA	Stacklazy		Covington		Niv. Arku Est. + Pse.		MST	
Kop.	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)**

4.14 taula – Oinarria (Oin.) eta hiru zuhaitz-transformazioak (ZT), ezaugarri automatikoekin ebaluatzeko multzoan. Emaitza guztiak estatistikoki esanguratsuak dira; salbuespenak MST 4.000 eta 9.100 dira (Niv. Arku Est. + Pse.: Nivre Arku Estandarra + Pseudo-proiektiboarekin).

Transformazioak	Stacklazy	Covington	Niv. Arku Est. + Pse.	MSTParser
Oin.	78,79	78,71	78,89	80,17
ST	**80,26 (+1,47)**	79,24 (+0,53)	**80,35 (+1,46)**	79,98 (-0,19)
MPT	78,94 (+0,15)	**79,06 (+0,35)**	79,12 (+0,23)	80,17 (+0,00)
KT	79,10 (+0,31)	**79,22 (+0,51)**	79,19 (+0,30)	**79,84 (+0,33)**
ST + MPT + KT	**80,55 (+1,76)**	80,23 (+1,52)	**80,80 (+1,91)**	80,87 (+0,70)

(a) Urre-patroiko ezaugarriak.

Transformazioak	Stacklazy	Covington	Niv. Arku Est. + Pse.	MSTParser
Oin.	76,83	76,98	77,31	77,88
ST	**77,96 (+1,13)**	77,75 (+0,77)	**78,26 (+0,95)**	78,26 (+0,38)
MPT	77,05 (+0,22)	**77,56 (+0,58)**	77,49 (+0,18)	**78,22 (+0,34)**
KT	**77,68 (+0,85)**	**77,77 (+0,79)**	78,01 (+0,70)	78,27 (+0,39)
ST + MPT + KT	78,26 (+1,43)	**78,44 (+1,46)**	**79,09 (+1,78)**	78,58 (+0,70)

(b) Ezaugarri automatikoak.

4.15 taula – Sistema eta algoritmo ezberdinekin zuhaitz-transformazioak banan-banan edo elkarrekin konbinatzerakoan LAS neurrian lortutako emaitzak (Oin.: Oinarria, ST: Sintagmen Transformazioa, MPT: Mendeko Perpausen Transformazioa, KT: Koordinazioaren Transformazioa, Niv. Arku Est. + Pse.: Nivre Arku Estandarra + Pseudo-proiektiboarekin).

guztietan. MSTParser sistemarekin, urre-patroiko ezaugarriak erabiltzetik ezaugarri automatikoak erabiltzera igarotzen denean, 2,3 puntuko galera dago; aitzitik, MaltParser sistema guztiekin 1,5 eta 2 puntu bitarteko galera baino ez dago. Horrela, MSTParser sisteman informazio morfologikoak eragin handiagoa duela esan daiteke. Hala ere, MSTParser sistemarekin lortu da oinarririk onena, bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin.

- **Hiru transformazioen konbinaketa**. Transformazioen konbinaketan emaitza hobeak lortzen dira MaltParser sisteman, MSTParser sistemarekin baino, eta bien arteko aldea desagertu egiten da. Esperimentu guztietan, transformazioen konbinaketarekin hobekuntza handiagoa lortu da entrenamendu azpimultzoaren tamaina txikiagoa denean, handiagoa denean baino. Algoritmo gehienetan, transformazioen konbinaketaren eragina zuhaitz-bankua bikoiztearen adinakoa dela esan daiteke. Hiru transformazioen konbinaketa guztiak estatistikoki esanguratsuak dira; alabaina, salbuespena ezaugarri automatikoak erabiltzerakoan gertatzen da MSTParser 4.000 eta 9.100 tamainarekin. Emaitza bat aipatzearren, MaltParser sistemako Nivre Arku Estandar algoritmoa transformazio pseudo-proiektiboarekin konbinatuz lortu dena izan da, izan ere, ezaugarri automatikoekin, 79,09ko zehaztasuna lortu da LAS neurrian, urre-patroiko ezaugarriekin lortutako emaitza gaindituz (LAS = 78,89).
- **Transformazio bakoitza banan-banan edo guztiak elkarrekin konbinatuta**. Bakarkako transformaziorik onena sintagmen transformazioa da, bai ezaugarri automatikoekin bai urre-patroiko ezaugarriekin. Oro har, osagai sintaktikoaren azkeneko morfema azpi-zuhaitzaren buru jartzeak analizatzailearentzat mesedegarria dela esan daiteke. Lan hau bat dator Goenagak azkeneko morfema buru ezartzeko gauzatutako lanarekin (Goenaga, 1980). Teknika hauek esaldi berriekin lan egiteko prestatuta daude, hau da, hitza bere lema eta morfema nagusietan banatu eta analizatu ostean, hitz-forma osatu eta itzultzen dute. 4.15 taulan, oro har, transformazioen eragina elkarrekiko independentea dela, hau da, transformazio desberdinen eragina batu egiten dela ikus daiteke. Azkenik, esan beharra dago aukerarik onena algoritmo guztiekin transformazio guztiak konbinatzea dela, nahiz eta transformazioek arku ez-proiektibo ugari sortu, itzulerako prozesuan zenbait erlazio galdu eta dependentzia-arkuen luzeraren batez bestekoa handitu.

4.1.6 Ondorioak

Collins-ek (Collins, 2009) zuhaitzen aurre-prozesaketa erabili zuen analizatzailearen emaitzak hobetzeko helburuarekin, eta euskararako zuhaitz-bankuaren etiketatze moduak eta euskara hizkuntza buru-azkena den ezaugarria kontuan izan dira euskarako zuhaitz-bankuan transformazio ugari egiteko (ikusi 4.1 atala). Horrela, analisia hobetzeko euskararako egokiagoak izan daitezkeen transformazio pseudo-proiektiboaren, sintagmen transformazioaren, mendeko perpausaren transformazioaren eta koordinazio-transformazioaren emaitzak aztertuta, hurrengo ondorioak atera dira:

- **Transformazio pseudo-proiektiboak:** EZB II zuhaitz-bankua, % 1,18 arku ez-proiektibo eta % 13,28 esaldi ez-proiektiboekin, zuhaitz-banku ez-proiektiboen artean koka dezakegu. Algoritmo ez-proiektiboek algoritmo proiektiboak baino emaitza hobeak ematen dituzte. Transformazio pseudo-proiektiboak analisian dependentzia-etiketa konplexuak gehitu eta analisia neketsuago egiten badu ere, transformazio pseudo-proiektiboarekin hobekuntza esanguratsua lortzen dela probatu da.
- **Sintagmen transformazioa:** EZB I zuhaitz-bankuan, izen kategoriako hitzen zehaztasuna % 66koa da, eta zuhaitz-bankuko kategoriarik ugariena dela jakinda, ia errore guztien % 50 dela esan daiteke. Errore honen zergatia hau dela uste da: izenak berak kasu-markarik ez duenez (beste hitz batean dago), eta dependentzia-etiketa kasu gramatikalaren informazioarekin ezartzen denez, izena aditzarekin lotzerakoan dependentzia-etiketa okerrekoarekin egiten da batzuetan. Ondorioz, hitzak morfemetan banatzeak izan dezakeen eragina aztertzera eraman gaitu. Euskarako zuhaitz-bankuak (EZB I eta II zuhaitz-bankuak) etiketatzerakoan sintagmen buru bezala esanahi semantikoa duen hitza jarri da. 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, hitzaren azken morfema banatu ostean, sintagmen burua, buru semantikoa izan beharrean, sintagmen buru sintagmen azkeneko kokagunea hartzen duen hitzaren morfema printzipala izanez gero, analisian hobekuntza esanguratsuak lortzen direla probatu da. Sintagmen transformazioak arku ez-proiektiboak sortzen ditu eta transformazio pseudo-proiektiboarekin batera emaitzak hobetzen dira.

- **Mendeko perpausen transformazioa:** Oro har, euskaraz, mendeko perpausak aditz nagusiari edo aditz laguntzaileari morfema bat gehituz sortzen dira. 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 (EZB I eta II zuhaitz-bankuetan), mendeko perpausak aditz nagusiaren (buru semantikoaren) inguruan antolatuta daude. Baina gehienetan mendeko perpausen morfema (buru sintaktikoa) aditz laguntzaileari lotuta etortzen da; hau da, ez dator bat buru semantikoarekin. Horrela, sintagma bakunak eta mendeko perpausak hobeto bereizteko eta mendeko morfemaren informazio sintaktikoa bere burutik hurbilago egoteko, mendeko morfema bere aditz laguntzailetik banatu eta mendeko perpausaren buru jarri ostean, hobekuntza xumeak gertatu dira. Esan beharra dago, transformazio honek arku ez-proiektiboak sortzen dituela eta transformazio pseudo-proiektiboarekin batera emaitzak hobetzen direla.
- **Koordinazioaren transformazioa:** Hitzen artean erlazio bitarra argia denean, dependentzietan oinarritutako teoriek ondo baino hobeto lan egiten dute. Baina koordinazioaren kasuan, juntagailuaren eta juntagaien artean erlazio bitarrak sortzeko orduan, teoria edo estilo ezberdinak sortu dira. Koordinazio estilo ezberdinekin (Praga, Melcuk eta Mel'cuk Estiloaren Simetrikoarekin) probatu ostean, emaitzarik onenak Mel'cuk Estiloaren Simetrikoarekin lortu dira. Horrela, euskara buru-azkena izanik koordinazioaren buru, koordinazioaren azken osagaia jartzeak analizatzailearen lana zehatzagoa dela probatu du. Oro har, koordinazioa hizkuntzaren ezaugarrietara egokituz gero, analisian emaitza hobeak lortzen direla esan dezakegu.
- **Transformazioen konbinaketa eta tamaina:** Sintagmen transformazioa, mendeko perpausen transformazioa, koordinazio transformazioa eta transformazio pseudo-proiektiboa bata bestearen atzetik aplikatu behar dira. Konbinaketan transformazioen ordenak garrantzia du, izan ere, transformazio pseudo-proiektiboa sintagmen transformazioa eta mendeko perpausen transformazioaren ostean joan behar da, sintagmen eta mendeko perpausen transformazioek hainbat arku ez-proiektibo sortzen baitute. Esperimentu guztietan, transformazioen konbinaketa bateragarria dela probatu da, emaitza esanguratsuak lortu baitira. Algoritmo gehienetan, transformazioen konbinaketaren eragina

zuhaitz-bankua bikoiztearen heinekoa dela esan daiteke.

4.2 Analizatzaileen pilaketa edo *stacking*

Aurreko ataletan, analizatzailearen zehaztasuna hobetzeko zuhaitz-transformazioak erabili dira. Atal honetan, analizatzaileen pilaketa edo, ingelesez, *classifier stacking* izenarekin ezagutzen den teknika erabiliko da.

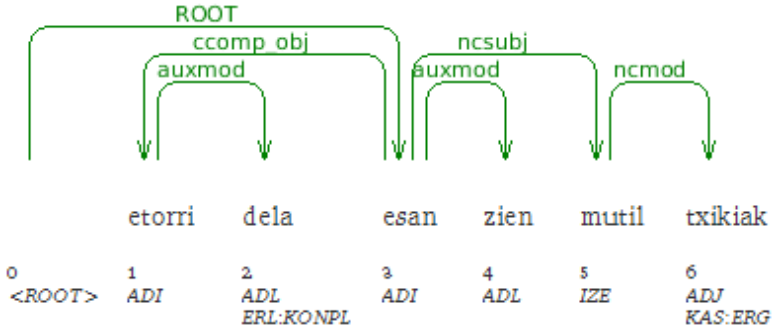
4.2.1 Sarrera

Analizatzaileen pilaketan bi analizatzaile bateratzeko, lehenengo analizatzailearen irteeran lortutako informazioa bigarren analizatzailearen sarrera aberasteko erabili da (Martins *et al.*, 2008; Nivre eta McDonald, 2008; McDonald eta Nivre, 2011) eta modelo bi elkarren osagarri izan ahal direla probatu da (Nivre eta McDonald, 2008). Datuetan oinarritutako sistemak erabiltzerakoan, ikasketa denboran ere bateratzeko aukera dago, izan ere, sistema batek beste sistema baten irteerako datuetatik ikas dezake. Horrela, lehenengo analizatzailearen irteeran lortutako informazioa bigarren analizatzailearen sarrera aberasteko erabili da (ikusi 4.12a irudian). Gainera, lehenengo analizatzailearen irteeran lortutako dependentzia-zuhaitzetatik informazioa goratzeko eta irteerako datuak gehiago aberasteko aukera eman du.

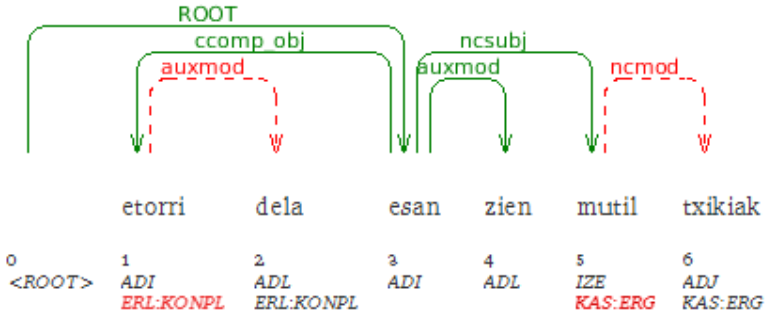
Adibidez, euskarako zuhaitz-bankuetan aditz konposatuaren burua aditz laguntzailea (buru sintaktikoa) izan beharrean, aditz nagusia (buru semantikoa) dela ikusi da. Baina, gehienetan, aditz laguntzaileari lotuta etortzen dira mendeko perpausen morfema eta nor, nori eta nork kasuen pertsona eta numeroa; hau da, ez dator bat buru semantikoarekin. Hori dela-eta, buruarekiko oso garrantzitsua den informazioa urrun egoteak analisian izan dezakeen eragina gutxitzeko, aditz laguntzaileetatik aditz nagusira goratuko den informazioa hau izango da: mendeko perpaus mota eta nor, nori eta nork kasuen pertsona eta numeroa.

Adibidez, aztertu dezagun lehenengo faseko analizatzailearen irteeran agertzen den esaldiaren analisia (ikusi 4.12b irudia), analizatzaileak "ccomp_obj" mendeko erlazio-motarekin batzeko zailtasunak izan dezakeela suposa daiteke, alde batetik, "etorri" mendeko perpausaren aditz nagusiak "esan" perpaus nagusiko aditz nagusiarekin lotzeko orduan ez duelako mendekoa denaren inolako informaziorik, eta bestetik, "dela" zuhaitz-sintaktikoan "etorri" hitza baino maila bat beherago dagoelako. Lehenengo faseko analizatzailearen

(a) lehenengo analizatzailearen irteeran lortutako informazioa, bigarren analizatzailearen sarrera aberasteko erabiltzen da



(b) lehenengo faseko analizatzailearen irteera (T.irteera)



(c) lehenengo faseko analizatzailearen irteeraren aberasketa (T.irteera.aberastua)

4.12 irudia – Dependentzia-zuhaitzetatik goratutako ezaugarriekin aberastutako zuhaitza.

irteera aberasteko eta, bidez batez, bigarren faseko analizatzaileari laguntzeko (ikusi 4.12c irudia), "dela" hitzaren ezaugarri zutabean dagoen mendeko konpletiboaren informazioa (ERL: KONPL) goratu da "auxmod" dependentzia-erlazioaren bitartez eta "dela" aditz laguntzailetik "etorri" aditz nagusira (betiere, lehenengo faseko sistemak "auxmod" erlazioa era egokian asmatzeko duen probabilitatea % 98,14koa dela jakinda). Gauza bera egin da sintagmekin, eta buru bezala esanahi semantikoa duen hitza jarri da. Baina sin-

tagmaren ezaugarri morfologiko garrantzitsuenak (kasu-marka eta numeroa) sintagmaren azkeneko kokagunea hartzen duen hitzean biltzen direla ikusirik, kasu-marka eta numeroa ezaugarriak sintagmaren azken osagaitik sintagmaren burura, "detmod" determinatzaile erlazioetatik (ingelesez, *determinant modifier*) (betiere, lehenengo faseko sistemak "detmod" erlazioa era egokian asmatzeko duen probabilitatea % 92,11koa dela jakinda) eta "ncmod" ez-perpausa erlazioetatik (ingelesez, *non-clausal modifier*) (betiere, lehenengo faseko sistemak "ncmod" erlazioa era egokian asmatzeko duen probabilitatea % 79,48koa dela jakinda) goratzerakoan analisian izan dezakeen eragina aztertu nahi da. Adibidez, 4.12c irudian, "txikiak" izenondoaren ezaugarri zutabean dagoen ergatibozko informazioa (KAS: ERG), "ncmod" dependentzia-erlazioaren bitartez, "txikiak" izenondotik "mutil" sintagmaren burura nola goratu den ikus daiteke.

Atal honen helburu nagusia esaldi berriak analizatzerakoan, informazio morfosintaktikoa lehenengo analizatzailearen irteeraren arkuetatik goratu ostean, bigarren analizatzailearen zehaztasuna hobetzea izango da.

Horretarako grafoetan eta trantsizioetan oinarritutako izaera berdin eta ezberdineko sistemak pilatuz euskarako analizatzailearen zehaztasuna hobetzea eta bidean hurrengo galderei erantzuteko ahaleginak egin dira:

- Zein ezaugarrirekin lortzen dira emaitzarik onenak pilaketan? Ezaugarri linguistikoekin edo egitura ezaugarriekin? Bakarkako zein ezaugarrirekin edo konbinaketa ezaugarrirekin lortzen dira emaitza hoberenak?
- Zein pilaketa teknika da hobea? Bengoetxea eta Gojenola-ren (2009a) proposamena, arku zuzenetatik ikasten duena, ala Martins *et al.*-en (2008) proposamena, analizatzailearen irteeraren arkuetatik ikasten duena?
- Zein sistemarekin lortzen dira emaitzarik onenak pilaketan? Izaera berdineko edo izaera desberdineko sistemekin? Edo izaera berdineko baina algoritmo desberdineko sistemekin?
- Zuhaitz-bankuaren tamainak eraginik ote du pilaketan?
- Urre-patroiko ezaugarriekin lortu ahal diren emaitzak, ezaugarri automatikoekin ere lor daitezke?
- Pilaketa teknika euskararako analizatzaile sintaktikoa hobetzeko baliagarria da? Zenbateraino?

Hurrengo ataletan, zerrendatutako aurreko galdera horiei erantzuna emango zaie.

4.2.2 Aukerak: zuhaitz zuzenetatik edo analizatzailearen irteeraren zuhaitzetatik aberastea

Bigarren faseko sistemarekin analizatzaile aberastu bat sortzeko, sistemak ikasteko erabiliko dituen datu aberastuak zuhaitz zuzenetatik (Bengoetxea eta Gojenola, 2009a) goratuak edo lehenengo faseko analizatzailearen irteerako zuhaitzetatik goratuak (Martins *et al.*, 2008) probatu dira.

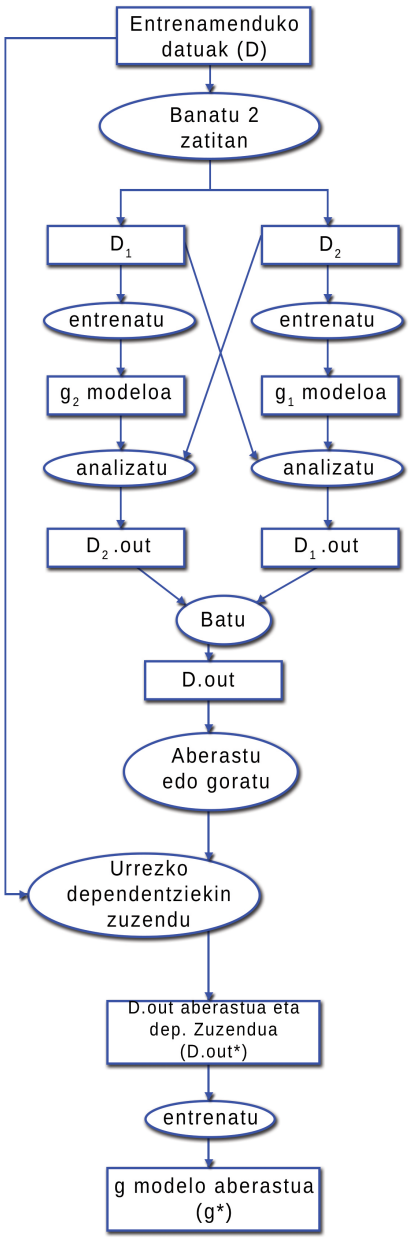
- **Martins *et al.*-en (2008) proposamena** da entrenamenduko datuak (D) L zatitan banatzea ($D_1, ..., D_L$). Modelo bakoitza $L-1$ zatirekin entrenatzen da, hau da, kanpoan zati bat utzita beti. $D_{-1} = D - D_1$ izanik, lehenengo g_1 modeloa D_{-1} zatiarekin entrenatu da. Urrats honen bukaeran, sortutako g_i modelo bakoitzarekin entrenamendutik at geratu den D_i zatia analizatu ostean, $D_i.out$ irteerak lortu dira. Adibidez, $L = 2$ erabiliz gero, entrenamenduko datuak (D) D_1 eta D_2 zatietan banatu dira, g_1 modeloa D_2 zatiarekin entrenatu ondoren lortutako modeloa eta g_2 modeloa D_1 zatiarekin entrenatu ondoren lortutako modeloa izan dira; eta azkenik, $D_1.out$ irteera D_1 g_1 modeloarekin analizatu ostean, eta, $D_2.out$ irteera D_2 g_2 modeloarekin analizatu osteko irteerak izan dira. Amaitzeko, lortutako $D_1.out$ eta $D_2.out$ irteerak batu egin dira, $D.out$ lortuz. $D.out$ irteera aberasteko 4.2.3 atalean azaldutako ezaugarriak erabili dira. $D.out$ irteera aberastu ostean, $D.out*$ datu-aberastua lortuko da, non dependentzia-arku zuzenak ikasteko analizatzailearen irteeraren arkuetatik goratutako ezaugarrietatik ikasteko gai izango da (ikusi 4.13 irudia).
- **Bengoetxea eta Gojenola-ren (2009a) proposamena** da zuhaitz zuzenak dituen urre-patroiko entrenamenduko datuak (D) aberastea, zuhaitzetatik informazio morfosintaktikoa goratuz eta irteera bezala aberastutako entrenamenduko datuak ($D*$) lortuz (ikusi 4.14 irudia).

4.2.3 Analizatzailea aberasteko informazioa

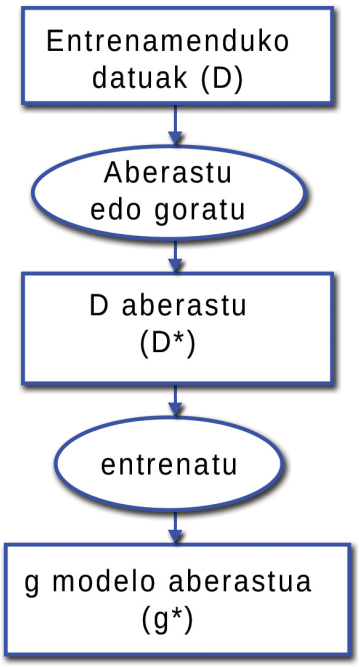
Bigarren analizatzailea aberasteko, era askotako informazioa erabil daiteke: kategoria, ezaugarri morfosintaktikoak, dependentzia-erlazioa, eta abar. Erabiliko diren ezaugarriak bi multzotan sailkatuko dira: egitura ezaugarriak eta ezaugarri linguistikoak.

Egitura ezaugarriak

Egitura ezaugarriak dependentzia-zuhaitzeko konfigurazio ezberdinetan oina-



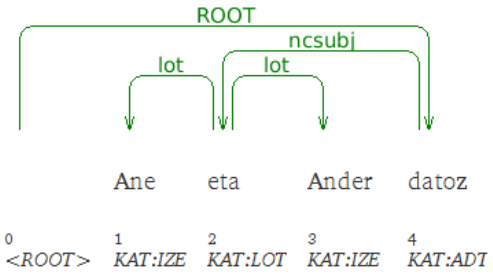
4.13 irudia – sistemak ikasteko erabiliko duen entrenamenduko datu aberastua sortzeko Martins *et al.*-ek (2008) egindako proposamena



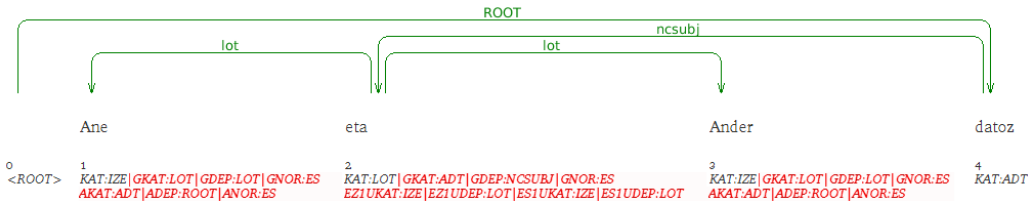
4.14 irudia – sistemak ikasteko erabiliko duen entrenamenduko datu aberastua sortzeko Bengoetxea eta Gojenola-k (2009a) egindako proposamena

rritutako ezaugarriak dira (Martins *et al.*, 2008; Nivre eta McDonald, 2008; McDonald eta Nivre, 2011). Tesi-lan honetan, bigarren analizatzailea aberasteko *CoNLL-X* formatuko *FEATS* zutabean gehitu diren egitura ezaugarriak hiru multzotan banatu dira:

- **Gurasoen informazioa**: uneko hitzaren ezaugarrietan gurasoaren kategoria, dependentzia-erlazioa eta dependentzia-erlazioaren norabidea gehitu dira. Adibidez, 4.15b irudian, "eta" juntagailuaren gurasoa den "datoz" aditzaren hurrengo informazioa gehitu da: Gurasoaren KATegoria (GKAT: ADT), Gurasoaren DEPendentzia (GDEP: ncsubj) eta Gurasoaren NORabidea (GNOR: ESkuina).
- **Umeen informazioa**: uneko hitzaren ezaugarrietan ume guztien kategoria eta dependentzia-erlazioa gehitu dira. Adibidez, 4.15b irudian, "eta" juntagailuaren EZkerreko 1. Umearen KATategoria (EZ1UKAT: IZE), EZkerreko 1. Umearen DEPendentzia (EZ1UDEP: lot), ESkuineko 1. Umearen KATategoria (ES1UKAT: IZE) eta ESkuineko 1.



(a) Lehenengo faseko analizatzailearen irteera



(b) Egitura ezaugarriekin aberastutako lehenengo faseko analizatzailearen irteera

4.15 irudia – Dependentzia-zuhaitzetatik goratutako egitura ezaugarriak (GKAT = Gurasoaren KATegoria; GDEP = Gurasoaren DEPendentzia; GNOR = Gurasoaren NORabidea; ES = ESkuma; EZ = EZkerra; EZ1UKAT = EZkerreko 1. Umearen KATategoria; EZ1UDEP = EZkerreko 1. Umearen DEPendentzia; ES1UKAT = ESkumako 1. Umearen KATategoria; ES1UDEP = ESkumako 1. Umearen DEPendentzia; AKAT = Aitonaren KATegoria; ADEP = Aitonaren DEPendentzia; ANOR = Aitonaren NORabidea).

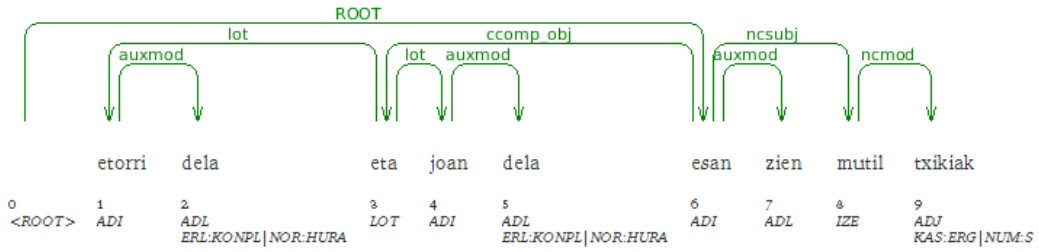
Umearen DEPendentzia (ES1UDEP: lot) ezaugarriak gehitu dira.

– **Aitonen informazioa**: Uneko hitzaren ezaugarrietan aitonaren kategoria, dependentzia-erlazioa eta dependentzia-erlazioaren norabidea gehitu dira. Adibidez, 4.15b irudian, “Ane” eta “Ander” hitzetan “datoz” Aitonaren KATegoria (AKAT: ADT), Aitonaren DEPendentzia (ADEP: ROOT) eta Aitonaren NORabidea (ANOR: ESkuma) gehitu dira.

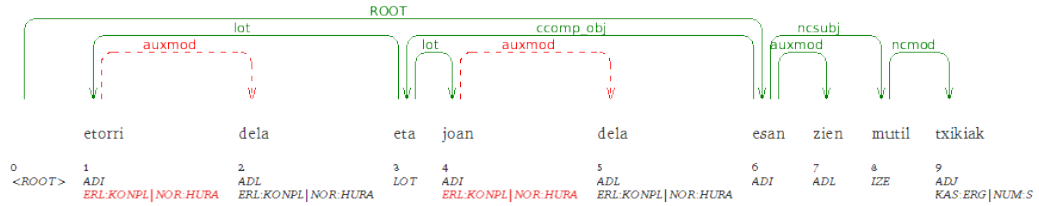
Ezaugarri linguistikoak

Euskara buru azkena izanik, lehenengo analizatzailearen irteera ematen duten ezaugarri morfosintaktikoak (numeroa, kasua eta mendeko perpausa be-

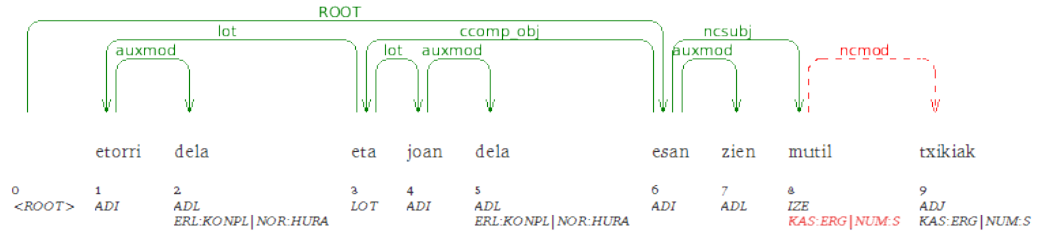
zalakoak) bigarren analizatzailea aberasteko printzipio linguistikoetan oinarritu gara (Bengoetxea eta Gojenola, 2009a; Bengoetxea eta Gojenola, 2010). Erabilitako ezaugarri linguistikoak hiru multzotan banatu dira:



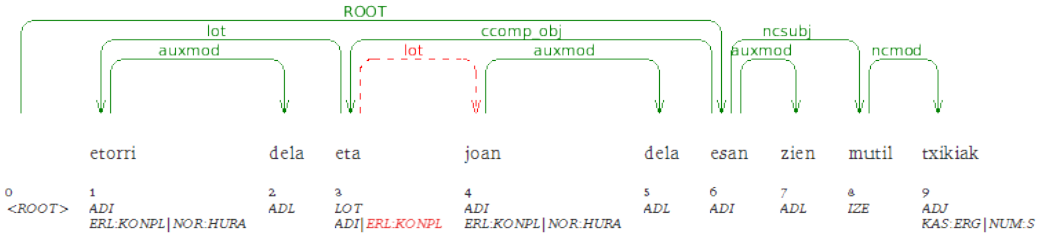
(a) Lehenengo faseko analizatzailearen irteera



(b) Aditz konposatuen aberasketa ezaugarri linguistikoekin



(c) Sintagmen aberasketa ezaugarri linguistikoekin



(d) Koordinazioaren aberasketa ezaugarri linguistikoekin

4.16 irudia – Dependentzia-zuhaitzetatik goratutako ezaugarri linguistikoak

- 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 analisian mesedegarria izan daitekeelakoan. 4.16a irudian, "dela" hitzean mendeko perpausa konpletiboaren morfema dago eta aditz nagusiaren behean agertzean, "eta" aditz nagusiarekin "ccomp_obj" mendeko erlazio-motarekin batzeko zailtasunak izan dezakeela suposa dezakegu. Lehenengo faseko analizatzailearen irteera aberasteko eta bidez batez bigarren faseko analizatzaileari laguntzeko (ikusi 4.16b irudian), "dela" hitzaren ezaugarri zutabean dagoen mendeko konpletiboaren informazioa (ERL: KONPL) eta aditzaren pertsonaren eta numeroaren informazioa (NOR: HURA) goratuko dira, "auxmod" dependentzia-erlazioaren bitartez eta "dela" aditz laguntzaileetatik "etorri" eta "joan" aditz nagusietara.
- Sintagmen aberasketa: euskarako zuhaitz-bankuak sintagmen buru bezala etiketatzerakoan esanahi semantikoa duen hitza jarri da. Sintagmaren azkeneko kokagunea hartzen duen hitzari lotuta datorren informazio oso garrantzitsua (kasu-marka eta numeroa) sintagmaren burura goratu da, "detmod" determinatzaile erlazioetatik (ingelesez, *determinant modifier*) eta "ncmod" ez-perpausa erlazioetatik (ingelesez, *non-clausal modifier*). Informazio hori buruarekiko hurbilago egotea analisian mesedegarria izan daitekeelakoan. Adibidez, 4.16c irudian, "txikiak" izenondoaren ezaugarri zutabean 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 bezala 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, 4.16b iruditik 4.16d irudira gertatu den koordinazioaren aberasketa ikus daiteke, hain zuzen, "joan" koordinazioaren azken juntagaitik "ADI" kategoria eta "ERL: KONPL" mendeko konpletiboaren informazioa "eta" juntagailura "lot"

dependentzia-erlazioaren bitartez nola goratu den.

Esperimentuak

Analizatzaileen pilaketa egiteko orduan aukera anitz daude:

- Bigarren analizatzailea entrenatzeko, Martins *et al.*-en (2008) proposamena edo Bengoetxea eta Gojenola-ren (2009a) proposamena.
- Aberasteko, egitura ezaugarriak edo linguistikoak erabiltzea.
- Erabilitako sistemak izaera berdinekoak ($Malt_{Malt}$ edo MST_{MST}) edo desberdinetakoak ($Malt_{MST}$ edo MST_{Malt}) izatea.
- Urre-patroiko ezaugarriekin edo ezaugarri automatikoekin probatzea.
- Tamaina desberdineko zuhaitz-bankuak erabiltzea.

Egin diren esperimentuekin, 4.2.1 ataleko galderei erantzuna aurkitzeko ahaleginak egin dira. Esperimentuak egiteko euskarako zuhaitz-bankua hiru multzotan banatu da. % 80 entrenatzeko, % 10 garapeneko eta gainerako % 10 azkeneko proba egiteko erabili da. Esperimentu guztiak garapeneko datuekin egin dira, sistema onenak azkeneko proba multzoarekin ebaluatzeko utzi dela. Gure esperimentuak egiteko orduan, emaitzarik onena $L = 2$rekin lortu da.

Ebaluazioa

Aurreko ataleko hipotesiak argitzeko, ebaluazio ezberdinak egin dira: ezaugarri linguistikoak banaka eta konbinatuta, egitura ezaugarriak banaka eta konbinatuta, izaera berdineko edo izaera desberdineko sistemak erabilita, tamaina ezberdineko zuhaitz-bankuak erabilita eta ezaugarri automatikoak erabilita egin dira. Atalez atal, lehenengo, egitura linguistiko eta egitura ezaugarrien pilaketaren emaitzak aurkeztuko dira; ondoren, izaera berdineko edo izaera desberdineko sistemen pilaketaren emaitzak, eta azkenik, tamainaren araberako pilaketaren emaitzak emango dira ezagutzera.

Ezaugarri linguistikoen pilaketaren emaitzak

Ondoren agertzen den 4.16 taulan, ezaugarri linguistikoak banaka eta konbinatuta goratzeak zehaztasunean duen eragina aztertzeko, Bengoetxea eta Gojenola-ren (2009a) proposamena erabili da, eta bigarren analizatzailera ezaugarri linguistikoak goratzeko, hurrengo aukerak: Aditz Konposatuen Aberasketa (AKA deituko dena), Sintagmaren Aberasketa (SA deituko dena), eta, azkenik, Koordinazioaren Aberasketa (KA deituko dena) eta euren arteko konbinaketak (AKA + SA + KA). Esperimentu hau egiteko, urre-pa-

troiko ezaugarriak dituen EZB II zuhaitz-bankua erabili da, eta MaltParser Stacklazy algoritmoko sistema bi pilatu dira.

Testa	**LAS**	**UAS**
Oinarria	78,79	83,43
AKA	78,96 (+0,17)	83,53 (+0,10)
SA	78,78 (-0,01)	83,27 (-0,16)
KA	78,83 (+0,04)	83,43 (+0,00)
SA + AKA + KA	79,36 (+0,57)‡	83,61 (+0,18)

4.16 taula – Ezaugarri linguistikoen pilaketa, Stacklazy algoritmoa urre-patroiko ezaugarri morfosintaktikoekin .

Adierazitako 4.16 taulan, ezaugarri linguistikoak banaka igotzeak ez diola pilaketari mesede handirik egiten ikus daiteke. Banakako ezaugarri linguistikorik onena AKA da, +0,17ko hobekuntzarekin. Aldiz, ezaugarri linguistiko guztiak (SA + AKA + KA) konbinatzerakoan +0,57ko hobekuntza esanguratsua lortu da.

Egitura ezaugarrien pilaketaren emaitzak

Egitura ezaugarriak banaka eta konbinatuta goratzeak zehaztasunean duen eragina aztertzeko, Martins *et al.*-en (2008) proposamena erabili da, eta bigarren analizatzailea aberasteko, gurasoen, umeen eta aitonen egiturazko ezaugarrien aberasketak banaka edo konbinatuta egin dira. Esperimentu hau egiteko, urre-patroiko ezaugarriak dituen EZB II zuhaitz-bankua erabili da, eta MaltParser Stacklazy sistema bi pilatu dira.

Testa	**LAS**	**UAS**
Oinarria	78,79	83,43
Gurasoen	79,33 (+0,54)‡	83,95 (+0,52)
Umeen	79,19 (+0,40)	83,95 (+0,52)
Aitonen	78,69 (-0,10)	83,50 (+0,07)
Gurasoen + Umeen + Aitonen	79,34 (+0,55)‡	84,06 (+0,63)

4.17 taula – Egitura ezaugarrien pilaketa, Stacklazy algoritmoa eta urre-patroiko ezaugarri morfosintaktikoekin.

Ondoren datorren 4.18 taulan, batetik, gurasoen eta umeen egiturazko ezaugarriek 0,54 eta 0,40 puntuko hobekuntza eman dute LAS neurrian, hurrenez hurren. Aitonen informazioak banaka ez duela laguntzen ikusten da, baina guztiak batera erabiltzerakoan emaitzarik onena lortu da, hots, 0,55eko hobekuntza esanguratsuarekin. Nahiz eta, LAS neurrian guztiak batera eta

gurasoen emaitzak berdinak izan, UAS neurrian guztiak batera hobea dela ikusirik, esperimentuak egiteko azken hau erabili da.

Izaera berdineko edo izaera desberdineko sistemen pilaketa
Hurrengo esperimentuan, izaera desberdineko MSTParser sistema eta MaltParser sistema pilatuz, sistema bakoitzak dituen onurak EZB II zuhaitz-bankuan bateragarriak diren aztertuko da. Jakina da MSTParser sistemak luzera handiko erlazioetan, aditz, izenlagun eta adberbioetan doitasun handiagoa duela, eta, aldiz, MaltPaser sistemak doitasun hobea duela erlazio laburretan, izen eta izenordainetan. McDonald eta Nivre-en (2011) lanean, izaera desberdineko $Malt_{MST}$ (MSTParser sistemarekin aberastutako MaltParser) eta MST_{Malt} (MaltParser sistemarekin aberastutako MSTParser) pilatu ziren, 2006ko CoNLL lehiaketan parte hartutako 13 hizkuntzetan. Horietan guztietan, hobekuntza esanguratsuak lortu zituzten. Emaitzarik onena esloveniarraren zuhaitz-bankuarekin lortu zuten MST_{Malt} eta $Malt_{MST}$ sistemekin eta 2,50 eta 3,94ko hobekuntzak lortu zituzten, hurrenez hurren; aldiz, emaitzarik txarrenak Japoniako zuhaitz-bankuarekin lortu ziren, 0,72 eta 0,55eko hobekuntzak baino ez baitzuten lortu, hurrenez hurren. Hizkuntza guztiak kontuan hartuta, batez beste, 1,70 eta 1,27ko hobekuntzak lortu zituzten.

MST_{Malt} pilaketan, EZB II zuhaitz-bankuaren egiturazko ezaugarri guztiak MSTParser sistemako *FEATS* zutabean gehitzerakoan, ezaugarri kopurua 156.880.393 ingurukoa izan da eta egikaritzeko orduan prozesuak 40 gigatik gorako memoria behar izan du; ondorioz, esperimentuak $Malt_{MST}$ pilaketarekin baino ez dira egin.

Proba ezberdinak egin dira, Bengoetxea eta Gojenola-ren (2009a) proposamenaren arabera (pilaketa arku-zuzenetatik ikasten duena) edo Martins *et al.*-en (2008) proposamenaren arabera (pilaketa arku ez-zuzenetatik ikasten duena). Oro har, Martins-en pilaketarekin emaitza hobeak lortu dira eta, ondorioz, esperimentuak egiteko Martins *et al.*-en (2008) proposamena erabili da.

MaltParser sistemarekin datozen Covington eta Stacklazy algoritmo ez-proiektiboak, pilaketaren bigarren fasean erabili ostean, emaitzarik onenak Stacklazy algoritmoarekin lortu dira. Horrela, gainerako sistemen irteerak erabiliko dira Stacklazy sistema aberasteko. 4.18 taulan, $Stacklazy_{MST}$ (MST-rekin aberastutako Stacklazy), $Stacklazy_{Stacklazy}$ (Stacklazy-rekin aberastutako Stacklazy) eta $Stacklazy_{Covington}$ (Covingtonekin aberastutako Stack-

lazy), bai ezaugarri linguistiko bai egiturazko ezaugarri guztiekin aberastu ostean lortutako emaitzak ikus daitezke.

Testa	**LAS**	**UAS**
Stacklazy oinarria	78,79	83,43
$Stacklazy_{Stacklazy}$ (egitura)	79,34 (+0,55)‡	84.06 (+0,63)
$Stacklazy_{Stacklazy}$ (linguistiko)	79,24 (+0,45)†	83,53 (+0,10)
$Stacklazy_{MST}$ (egitura)	**80,77 (+1,98)**‡	85,34 (+1,91)
$Stacklazy_{MST}$ (linguistiko)	**79,52 (+0,73)**‡	83,77 (+0,34)
$Stacklazy_{Covington}$ (egitura)	**80.02 (+1,23)**‡	84.78 (+1,35)
$Stacklazy_{Covington}$ (linguistiko)	79.31 (+0,52)‡	83.66 (+0,23)

4.18 taula – Izaera berdineko edo izaera desberdineko sistemen pilaketa, urre-patroiko ezaugarri morfosintaktikoekin

4.18 taulan, hurrengo emaitzak aipatuko dira:

– Egitura ezaugarriekin, ezaugarri linguistikoekin baino emaitza hobeak lortzen dira pilaketa guztietan.

– Pilaketaren bi faseetan erabiltzen diren algoritmoak bere garrantzia dutela ikus daiteke. Emaitzarik onenak izaera desberdineko sistemak (grafoetan oinarritutako MSTParser sistema eta trantsizioetan oinarritutako MaltParser sistema) pilatuz lortzen direla ikus daiteke. Adibidez, $Stacklazy_{MST}$ (egitura) pilaketarekin LAS neurrian, 1,98ko gehikuntza lortu da. Bigarren onenak, berriz, bi faseetan izaera berdineko sistema baina algoritmo ezberdinak erabiltzerakoan lortu dira. Adibidez, $Stacklazy_{Covington}$ (egitura) pilaketarekin 1,23ko gehikuntza lortu da, eta azkenik, bi faseetan izaera berdineko sistema eta algoritmo berdina erabiltzerakoan lortu dira. Adibidez, $Stacklazy_{Stacklazy}$ (egitura) pilaketarekin, 0,55eko gehikuntza lortu da.

– $Stacklazy_{MST}$ (egitura) pilaketarekin 1,98ko hobekuntza lortu da, hain zuzen, McDonald eta Nivre-ren (2011) lanean $Malt_{MST}$ pilaketarekin eta hizkuntza guztiak kontuan hartuta lortutako 1,27ko batez bestekoaren gainetik.

Tamainaren eragina pilaketan

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 esalditakoak (zuhaitz-banku osoa) izan dira. MaltParser algoritmo ezberdinekin probatu ostean, bigarren fasean Stacklazy algoritmoarekin lortu dira emaitzarik onenak. Horrela, 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.

Testa (LAS)	Stacklazy	$Stacklazy_{Stacklazy}$	$Stacklazy_{MST}$	$Stacklazy_{Covington}$
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)**‡

4.19 taula – Urre-patroiko ezaugarriak erabiliz, ezaugarri linguistikoak eta egitura ezaugarriak algoritmo eta sistema ezberdinetan goratu ostean lortutako emaitzak (Oin. = Oinarria, Ezaug. ling.= Ezaugarri linguistikoak, Egit. Ezaug. = Egitura ezaugarriak).

Pilaketa teknikarekin oinarriarekiko hobekuntza esanguratsuak lortu dira. $Stacklazy_{Stacklazy}$ pilaketarekin zuhaitz-banku txikiekin hobekuntza handiena lortzen da; $Stacklazy_{MST}$ eta $Stacklazy_{Covington}$ pilaketekin, aldiz, zuhaitz-banku handiekin lortzen da hobekuntza. Emaitza ez esanguratsuak $Stacklazy_{Covington}$ pilaketan baino ez dira gertatu, urre-patroiko ezaugarriekin, 500 eta 1.000 esalditako zuhaitz-bankuetan.

Ezaugarri automatikoak pilaketan

Gure esperimentuak harago doaz eta kasu errealetan aplikagarritasuna duen jakiteko, ezaugarri automatikoak erabiltzeak zehaztasunean izan dezakeen eragina aztertuko da. 4.20 taulan, ezaugarri automatikoko zuhaitz-bankuekin erabilitako pilaketak, ezaugarri automatikoekin eta tamaina ezberdineko zuhaitz-bankuekin lortutako emaitzak ikus daitezke.

Testa (LAS)	Stacklazy	$Stacklazy_{Stacklazy}$	$Stacklazy_{MST}$	$Stacklazy_{Covington}$
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)‡

4.20 taula – Ezaugarri automatikoak erabiliz, ezaugarri linguistikoak eta egitura ezaugarriak algoritmo eta sistema ezberdinetan goratu ostean lortutako emaitzak (Oin. = Oinarria, Ezaug. Ling.= Ezaugarri Linguistikoak, Egit. Ezaug. = Egitura Ezaugarriak).

Ezaugarri automatikoekin lortutako emaitzak esanguratsuak izan arren, urre-patroiko ezaugarriekin lortutakoak baino apalagoak direla ikus daiteke. Analizatzaile morfologikoak emandako ezaugarri okerrak sistema aberasteko erabiltzerakoan, analisian eragina dutela suposa daiteke. $Stacklazy_{Stacklazy}$ pilaketaren bitartez eta zuhaitz-banku txikiekin emaitzarik onenak lortzen dira; $Stacklazy_{MST}$ eta $Stacklazy_{Covington}$ pilaketekin, aldiz, zuhaitz-banku handiekin lortzen dira emaitzarik onenak. Emaitza ez esanguratsuak $Stacklazy_{Covington}$ pilaketan 500, 1.000 eta 2.000 esalditako zuhaitz-bankuekin baino ez dira gertatzen. Urre-patroiko ezaugarriekin bezala, ezaugarri automatikoak erabiltzerakoan emaitzarik onena $Stacklazy_{MST}$ pilaketarekin lortzen da, 1,72 puntuko gehikuntzarekin, eta, txarrena, berriz, $Stacklazy_{Stacklazy}$ pilaketarekin lortzen da, 0,39 puntuko gehikuntzarekin.

4.2.4 Ondorioak

Pilaketan, analizatzaile bi bateratzeko, lehenengo analizatzailearen irteeran lortutako informazioa bigarren analizatzailearen sarrera aberasteko erabili da. Gainera, lehenengo analizatzailearen irteeran lortutako dependentzia-zuhaitzetatik informazioa goratzeko eta irteerako datuak gehiago aberasteko egitura ezaugarriak erabiltzeaz gain ezaugarri linguistikoak erabili dira. Emaitzak aztertuta, sarreran egindako galderak erantzuten ahaleginduko gara:

- **Zein ezaugarrirekin lortzen dira emaitzarik onenak pilaketan? Ezaugarri linguistikoekin edo egitura ezaugarriekin? Bakarkako zein ezaugarrirekin edo konbinaketa ezaugarrirekin lortzen dira emaitza hobeak?** Ezaugarri linguistikoak eta egitura ezaugarriak banaka eta konbinatuta goratzeak zehaztasunean duen eragina aztertu ostean, guztien konbinaketarekin emaitzarik onenak lortu dira. Oro har, egitura ezaugarriekin, ezaugarri linguistikoekin baino emaitza hobeak lortu dira.
- **Zein da hobea, Bengoetxea eta Gojenola-ren (2009a) proposamena, arku zuzenetatik ikasten duen pilaketa, ala Martins *et al.*-en (2008) proposamena, arku ez-zuzenetatik ikasten duena?** Oro har, Martins *et al.*-en (2008) proposamenak Bengoetxea eta Gojenola-ren (2009a) proposamenak baino emaitza hobeak eman ditu. Esaldi berriak analizatzerakoan, informazio morfosintaktikoa lehenengo analizatzailearen irteeraren arkuetatik goratu ostean, bigarren analizatzailearen zehaztasuna hobetzen dela probatu da.

- **Zein da onena pilaketan? Izaera berdineko sistemekin ala izaera desberdineko sistemekin eraikitako pilaketa? Izaera berdineko baina algoritmo desberdineko sistemekin eraikitako pilaketa?** Oinarriarekiko hobekuntzarik onena pilaketaren lehen eta bigarren fasean erabilitako sistemak izaera desberdinekoak direnean gertatzen da, (McDonald eta Nivre, 2011) lanean probatu zuten bezala. Oinarriarekiko bigarren hobekuntzarik onena, bi faseetan, izaera berdineko sistema baina algoritmo ezberdinak erabiltzerakoan lortu dira. Hobekuntza txikienak bi faseetan izaera berdineko sistema eta algoritmo berdina erabiltzerakoan lortu dira. Horrela, sistema eta algoritmo ezberdinen osagarritasuna pilaketan mesedegarria dela esan dezakegu.
- **Zuhaitz-bankuaren tamainak eraginik ote du pilaketan?** Zuhaitz-bankuaren tamainak pilaketan duen garrantzia aztertu da, $Stacklazy_{Stacklazy}$ (sistema-algoritmo berdina, bi faseetan) pilaketaren bitartez eta zuhaitz-banku txikiekin hobekuntza handiena lortu da; $Stacklazy_{MST}$ eta $Stacklazy_{Covington}$ (sistema-algoritmo ezberdinak, bi faseetan) pilaketekin, aldiz, hobekuntzarik handienak zuhaitz-banku handiekin lortu dira.
- **Urre-patroiko ezaugarriekin lortutako hobekuntzak ezaugarri automatikoekin ere gertatzen dira?** Bai, pilaketa ezaugarri automatikoekin ere erabil daitekeela probatu da; emaitza bat aipatzearren, $Stacklazy_{MST}$ pilaketa eginez +1,72 puntuko hobekuntza lortu da LAS neurrian.
- **Pilaketa teknika euskararako analizatzaile sintaktikoa hobetzeko baliagarria da? Zenbateraino?** Oro har, zuhaitz-banku osoari erreparatuz gero, pilaketak emaitza esanguratsuak eman ditu, baina zuhaitz-transformazioan lortutakoak baino apalagoak dira. Salbuespen bakarra $Stacklazy_{MST}$ pilaketa izan da, zuhaitz-transformazioan lortutakoak baino hobeak lortu baitira.

4.3 Bozketa bidezko konbinaketa

4.3.1 Problemaren zehaztapena

Aurreko ataletan analizatzaile ezberdinak lortu dira: oinarrizko algoritmoak erabiltzen dituztenak, eta, zuhaitz-transformazioak eta pilaketa bezalako teknika hedatuak erabiltzen dituztenak. Pilaketan, bi analizatzaile ikasketa den-

boran bateratzen dira, hau da, lehenengo analizatzailearen irteeran lortutako ezaugarriak bigarren analizatzailearen sarrera aberasteko erabiltzen dira. Bozketa bidezko konbinaketan, aldiz, analizatzaile desberdinen irteerak kontuan hartzen dira, irteera bateratu eta egoki bat lortzeko asmoarekin. (Sagae eta Lavie, 2006) lanean, etiketa gabeko analisi dependentzian, trantsizioetan oinarritutako hiru modeloren eta grafoetan oinarritutako modelo baten irteerak konbinatu zituzten, eta oinarrizko algoritmo onenak baino emaitza hobeak lortu zituzten. *CONLL 2007 Multilingual Dependency Parsing* lehiaketan Hall *et al.*-ek (2007) trantsizioetan oinarritutako 6 modeloren irteerak konbinatuz lehenengo postua lortu zuten 19 hizkuntzekin eta 20 sistemekin lehiatu ostean.

Ildo beretik, gure esperimentuak gauzatzeko, dependentzietan oinarritutako analizatzaileen irteerak bateratzeko *Maximum Spanning Tree*ko Chu-Liu/Edmonds algoritmoa erabiltzen duen MaltBlender[4] tresna erabili da. MaltBlender-ek analizatzaile sintaktikoen irteera bakoitzari pisu bat ematerakoan, pisu-eskema ezberdinetan oinarritzeko aukera eskaintzen du: POS kategoria, dependentzia-erlazioa edo analizatzaile bakoitzaren puntuazioaren araberakoa. (Surdeanu eta Manning, 2010) lanean, pisu-eskema ezberdinak aplikatu ostean, analizatzaileen aniztasuna faktore garrantzitsuenetako bat zela probatu zuten, pisu-eskemen aldean.

Atal honen helburu nagusia aurreko ataletan lortutako oinarrizko sistemak eta sistema hedatuak (zuhaitz-transformazioak eta pilaketa) konbinatuz euskarako analizatzailearen zehaztasuna hobetzea eta bide horretan hurrengo galderei erantzuteko ahaleginak egin dira:

- Konbinatzeko garaian, analizatzaileen LAS puntuazioaren araberako ordenak edo aniztasunak (osagarritasunak) garrantzi handiagoa du?
- Oinarrizko sistemekin, zein da hobekuntza? Zein sistemaren artean ematen dira emaitzarik onenak?
- Sistema hedatuen irteerak konbinaketan sartzean, zein da hobekuntza? Zein sistemaren artean ematen dira emaitzarik onenak?
- Egoera erreal batean, urre-patroiko etiketak erabili beharrean etiketa automatikoak erabiltzerakoan, zein da hobekuntza?

Hurrengo ataletan zerrendatutako aurreko galdera horiei erantzuna emango zaie.

[4] http://w3.msi.vxu.se/users/jni/blend

Esperimentuak

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, hurrengo analizatzaileak 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 algoritmoei zuhaitz-transformazioak eta pilaketa teknikak gehitu zaizkie.
- Aniztasun handiagoa egoteko, (Hall *et al.*, 2007) lanean egindakoaren ildotik, oinarrizko algoritmoekin zuhaitz-bankuko esaldiak ezkerretik eskuinera eta eskuinetik ezkerrera tratatu dira, eta salbuespen bakarra MSTko algoritmo globala izan da; izan ere, MSTko algoritmo globalarekin, esaldiak ezkerretik eskuinera edo eskuinetik ezkerrera tratatzerakoan emaitza berdina lortzen da.

Orain arte, analizatzaile multzo mugatua erabili dituzten lanek (Hall *et al.*, 2007; Sagae eta Lavie, 2006), eta analizatzaile multzo zabala erabili dituzten lanek (Nivre *et al.*, 2007b), bozketa sistema konbinatzeko orduan, beti x analizatzaile onenak aukeratzen dituzte n analizatzaile multzotik; hemendik aurrera, x onenen konbinaketa deituko da. Konbinatzeko garaian, analizatzaileen puntuazioak edo aniztasunak (osagarritasuna) garrantzi handiagoa duen aztertzeko, hau da, puntakoak ez diren 3 analizatzaileren konbinaketak edo 3 onenen konbinaketak doitasun hobea emateko aukera bera duten probatzeko, hurrengo esperimentuak gauzatu dira:

- x onenen konbinaketa eta ausazko onenen konbinaketa probatu dira.
- Analizatzaile hedatuak gehitu dira, zuhaitz-transformazioak eta pilaketak ekar dezakeen aniztasuna onuragarria den ala ez ikusteko. Horretarako, alde batetik, oinarrizko algoritmoak konbinatuko dira, eta, beste aldetik, analizatzaile hedatuekin batera konbinatuko dira.

Baina gure esperimentuak harago doaz, izan ere, orain arte, hainbat lanetan urre-patroiko zuhaitz-bankuak ebaluatzeko bozketa bidezko teknika erabili da (Hall *et al.*, 2007; Sagae eta Lavie, 2006; Nivre *et al.*, 2007b). Tesi-lan honetan, urre-patroiko etiketak erabiltzeaz gain, etiketa automatikoak erabiltzerakoan, egoera erreal batean lor daitekeena aztertu da bozketa bidezko teknikarekin.

Ebaluazioa
Alde batetik, ausazko onenen konbinaketa probatzeko, lehenengo eta behin, entrenamenduko datuekin indar basatia erabili da, hau da, n analizatzaileren irteera multzo posible guztiak[5] konbinatu dira. Azkenik, lortutako konbinaketa onena test multzoari aplikatu zaio. Beste aldetik, x onenen konbinaketa sistema erabili da. Aurreko bi teknika hauek analizatzaile multzo ezberdinekin egin dira.

4.3.2 Oinarrizko algoritmoen konbinaketa

Lehenengo esperimentuan, oinarrizko algoritmoak soilik erabili dira. 12 analizatzaile ezberdinen multzoa osatu da, MaltParser eta MSTParser-ekin erabilgarri dauden oinarrizko algoritmoekin osatuta: MSTParser sistemako bi aldaera (lehenengo eta bigarren mailako algoritmo ez-proiektiboarekin eta 3.4 atalean azaldu den kode moldaketarekin) eta MaltParserreko 10 aldaera ezberdin:

- Ezkerretik eskuinera Stacklazy, Covington, Nivre Arku Estandarra eta Nivre Arku Estandarra transformazio pseudo-proiektiboarekin LibLINEAR ikasketa automatikoarekin
- Ezkerretik eskuinera Stacklazy SVM ikasketa automatikoarekin
- Aurreko guztiak eskuinetik ezkerrera

4.21 eta 4.22 tauletan, urre-patroiko ezaugarriak eta ezaugarri automatikoak erabiliz lortutako emaitzak onenetik txarrenera ordenatuta agertzen dira.

4.23 taulan, urre-patroiko ezaugarriekin eta ezaugarri automatikoekin x onenen konbinaketa erabiltzerakoan, oinarrizko 12 analizatzaileen irteerak multzo posible guztiak (3, ...,12) konbinatu ostean lortutako emaitzak agertzen dira. 4.23 taulatik hurrengo emaitzak aipatuko dira:

- Urre-patroiko ezaugarri morfologikoak erabilita, ausazko onena metodoarekin 82,99 lortzen da LAS neurrian, oinarrizko sistema onenarekin alderatuz gero (LAS = 80,17), eta ia 3 puntuko hobekuntza lortzen da.
- n balioa 3tik 12ra arte izanda, bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin egindako konbinaketetan, ausazko onena konbinaketarekin x onenen konbinaketarekin baino emaitza hobeak lortzen dira; salbuespen bakarra urre-patroiko ezaugarriekin eta sistemen kopurua 8 denean gertatzen da (ikusi 4.23 irudia).
- 7 sistemekin lortutako ausazko onena konbinaketan lortutako emaitza

[5] MaltBlender-en bozketa-sistemak gutxienez hiru sistemen irteerak behar ditu

Ordena	Deskribapena	Garapena		Testa	
		LAS	UAS	LAS	UAS
1	Bigarren Mailako MST Ez-proiektiboa	82,53	87,77	80,17	86,01
2	Stacklazy (LibSVM)	82,39	86,59	79,81	84,41
3	Nivre Arku Estandarra + Pseudo	81,50	86,06	78,89	83,63
4	Stacklazy (LibLINEAR)	81,50	86,04	78,79	83,43
5	Lehen Mailako MST Ez-proiektiboa	81,32	86,99	79,01	85,08
6	Stacklazy RL (LibSVM)	80,94	85,61	78,23	83,60
7	Covington	80,61	85,05	78,71	83,50
8	Stacklazy RL (LibLINEAR)	80,58	85,38	77,66	83,46
9	Nivre Arku Estandarra	80,57	85,24	77,98	82,84
10	Nivre Arku Estandarra + Pseudo RL	80,43	85,39	78,41	84,15
11	Nivre Arku Estandarra RL	79,22	84,50	77,02	82,85
12	Covington RL (LibLINEAR)	78,34	83,43	76,10	81,46

4.21 taula – Urre-patroiko ezaugarriak erabilita, oinarrizko 12 sistema onenak ordenatuta garapeneko LAS balioaren arabera ordenatuta (RL = Eskuinetik eskerrera; Pseudo = Pseudo-proiektiboa).

Ordena	Deskribapena	Garapena		Testa	
		LAS	UAS	LAS	UAS
1	Stacklazy (LibSVM)	79,56	84,33	78,35	83,35
2	Bigarren Mailako MST Ez-proiektiboa	79,30	85,28	77,88	84,08
3	Lehen Mailako MST Ez-proiektiboa	78,91	84,83	77,24	83,24
4	Nivre Arku Estandarra + Pseudo	78,21	83,09	77,31	82,62
5	Stacklazy (LibLINEAR)	77,90	82,84	76,83	82,18
6	Stacklazy RL (LibLINEAR)	77,85	83,32	76,54	82,86
7	Stacklazy RL (LibSVM)	77,84	83,30	77,09	82,92
8	Nivre Arku Estandarra	77,63	82,60	76,14	81,60
9	Nivre Arku Estandarra + Pseudo RL	77,49	83,07	76,40	82,77
10	Covington	77,33	82,33	76,98	82,17
11	Nivre Arku Estandarra RL	76,30	82,18	75,38	81,92
12	Covington RL	75,35	81,31	74,61	80,73

4.22 taula – Ezaugarri automatikoak erabilita, oinarrizko 12 sistema onenak ordenatuta garapeneko LAS balioaren arabera ordenatuta (RL = Eskuinetik eskerrera; Pseudo = Pseudo-proiektiboa).

	Urre-patroiko ezaugarriekin		Ezaugarri automatikoekin	
Kop. (n)	**Ausazko onena**	**x onenen konb.**	**Ausazko onena**	**x onenen konb.**
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)**

4.23 taula – Urre-patroiko ezaugarriekin eta ezaugarri automatikoekin ausazko onena eta x onenen konbinaketa onenak, oinarrizko sistema onenarekin alderatuta.

gainerako n sistemekin lortutakoaren pare gelditzen dela ikus daiteke, bai ezaugarri automatikoak, bai urre-patroiko ezaugarriak erabilita. x onenen konbinaketa metodoarekin, aldiz, urre-patroiko ezaugarriekin 8 sistema eta ezaugarri automatikoekin 12 sistema konbinatu behar dira konbinaketa onenaren parera heltzeko (ikusi 4.23 taulan).

– Ezaugarri automatikoak erabilita, ausazko onena metodoarekin 81,17 lortzen da LAS neurrian, oinarrizko sistema onenarekin alderatuz gero (LAS = 78,35), eta 2,82 puntuko hobekuntza lortzen da. Ezaugarri automatikoekin zehaztasunik ez galtzea oso emaitza interesgarria da, aplikazio zuzena baitu.
– x onenen konbinaketan eta ausazko onena konbinaketan, bai urre-patroiko ezaugarriekin bai ezaugarri automatikoekin lortutako hobekuntzak 2,75 ingurukoak izan dira, nahiz eta ausazko onena metodoarekin zerbait hobetu.

Baina, zein sistemaren artean ematen dira emaitzarik onenak? Izaera edo algoritmo desberdineko sistemak, eskuinetik ezkerrera edo ezkerretik eskuinera sortutako modeloek eragina dute konbinaketan? Ordenak badu garrantzirik? Galdera sorta honi erantzuteko, 4.24 taulan, ezaugarri automatikoekin ausazko onena konbinaketarekin 3tik 7ra arte lortutako sistemen zerrenda gehitu da. Sistema kopurua 7 denean, konbinaketa onena 1, 2, 3, 5, 7, 10 eta 12. sistemak konbinatuz lortzen da. Nahiz eta beste sistemak ere agertu, 1, 3 eta 7. aldaerak konbinaketa guztietan agertzen dira: Stacklazy ezker-eskuin, lehen mailako MST algoritmo ez-proiektiboa eta Stacklazy eskuin-ezker. Alde batetik, esan dezakegu MST eta Stacklazy izaera desberdineko (globala

eta lokala) sistemak direla (eta konbinaketak sistema dibergenteak bilatzen dituela); beste aldetik, esaldiak ezkerretik eskuinera edo alderantziz entrenatutako analizatzaileen irteerak konbinatzea aniztasun aberasgarria dela ematen du, nahiz eta eskuinetik ezkerrera entrenatutako sistemen irteeren emaitzak askoz ere baxuagoak izan LAS neurrian. Azkenik, esan behar da, x onenen konbinaketak ez direla inoiz taulan agertzen. Hori dela-eta, analizatzaile sintaktikoen konbinaketan, dibergentziak zenbaterainoko eragina izan dezakeen ebatzi gabe gelditzen da.

Sistema kopurua (n)	**x onenen konb.**	**Ausazko onena**	**Zerrenda (Ausazko onena)**
3	78,89 (+0,54)	80,22 (+1,87)	1, 3, 7
4	79,01 (+0,66)	80,69 (+2,34)	1, 3, 7, 9
5	80,47 (+2,12)	80,63 (+2,28)	1, 2, 3, 7, 10
6	80,62 (+2,27)	80,84 (+2,49)	1, 2, 3, 7, 9, 10
7	80,99 (+2,64)	81,17 (+2,82)	1, 2, 3, 5, 7, 10, 12

4.24 taula – Ezaugarri automatikoak erabiliz, oinarrizko sistemen, n azpimultzoak 3tik 7ra arte konbinatu ostean lortutako emaitzak LAS neurrian.

4.3.3 Analizatzaile hedatuen konbinaketa

Bigarren esperimentuan, aurreko oinarrizko analizatzaileez gain, analizatzaile hedatuak (zuhaitz-transformazioak eta pilaketa) gehitu dira, teknika hedatuak konbinaketa egiterakoan lagungarriak diren jakiteko, edo oinarrizko analizatzaileen konbinaketa egin ondoren nahikoa den jakiteko. Printzipioz, analizatzaile hedatuak gehitzeak aniztasuna gehitu beharko luke, baina gerta liteke, analizatzaile hedatuekin lortzen den hobekuntza desagertzea konbinatzerakoan. Esperimentuak egiteko 26 analizatzaile ezberdin lortu dira (oinarrizkoak + hedatuak). 4.26 eta 4.27 tauletan, urre-patroiko ezaugarriak eta ezaugarri automatikoak erabiliz lortutako emaitzak onenetik txarrenera ordenatuta agertzen dira. Baina 26 sistemen azpimultzo posible guztiak kalkulatzeko mementuan, esperimentuen kopurua izugarri igotzen zenez eta (ikusi 4.25 taula), bigarren esperimentua egiteko, 14 sistema onenak erabiltzea erabaki da. 4.25 taulan, $m = 12, 14, 19$ eta 26 elementuko multzoetan, egin daitezkeen n elementuko azpimultzoen kopurua agertzen da ($n < m$), azpimultzo bakoitza gutxienez elementu batean ezberdindu behar dela jakinda. Horrela, 4.25 taulan m eta n kopuruak aldatzerakoan konbinaketa kopurua nola igotzen den ikus daiteke. MaltBlender-ek, gutxienez, 3 sistema behar ditu egoki funtzionatzeko, hori dela-eta, konbinaketa-kopurua kalkulatzeko orduan $n = 3$tik hasi gara.

$C_{m,n}(n \leq m) = m!/(n! * (m-n)!)$	m=12	m=14	m=19	m=26
$n = 3$	220	364	969	2.600
$n = 4$	495	1.001	3.876	14.950
$n = 5$	792	2.002	11.628	65.780
$n = 6$	924	3.003	27.132	230.230
$n = 7$	792	3.432	50.388	657.800
$n = 8$	495	3.003	75.582	1.562.275
$n = 9$	220	2.002	92.378	3.124.550
$n = 10$	66	1.001	92.378	5.311.735
$n = 11$	12	364	75.582	7.726.160
$n = 12$	1	91	50.388	9.657.700
$n = 13$		14	27.132	10.400.600
$n = 14$		1	11.628	9.657.700

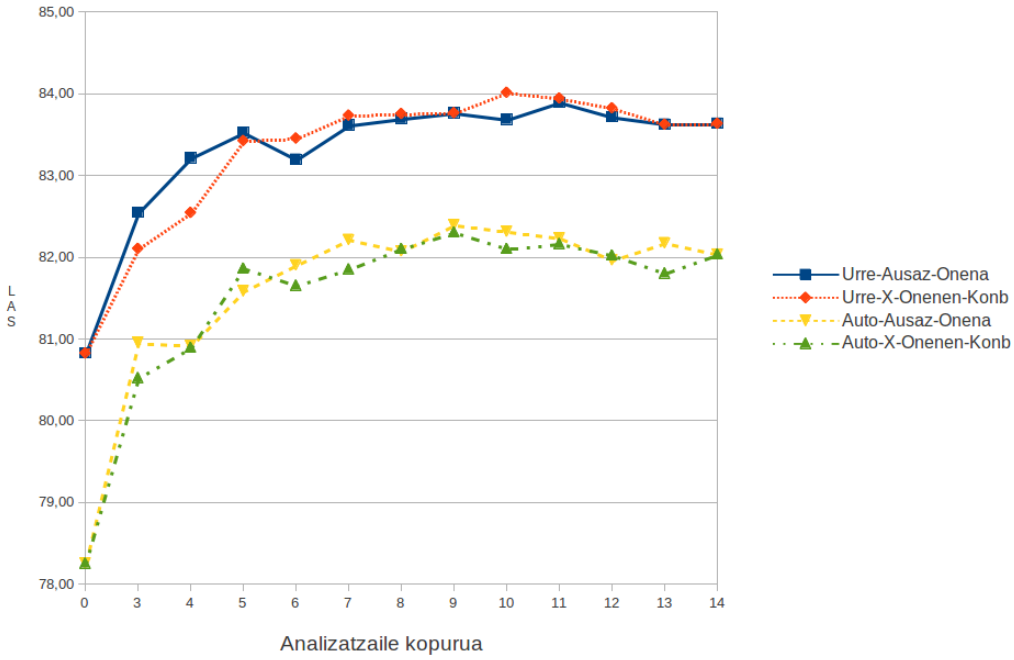
4.25 taula – konbinaketa-kopurua m eta n kopuruak aldatzerakoan

Ord.	**Algoritmoa**	**Aldaera**	**Ikasketa**	**Nor.**	**Garapena**		**Testa**	
					LAS	UAS	LAS	UAS
1	Nivre Arku Estandarra + Pseudo	ZT	LibLINEAR	LR	83,33	87,43	80,83	85,41
2	Stacklazy	ZT	LibLINEAR	LR	82,91	87,21	80,65	85,30
3	Bigarren Mailako MST Ez-proiektiboa	ZT	MIRA	LR	82,87	88,08	80,92	86,37
4	$MST_{Stacklazy}$	Martins Pil. (ling.)		LR	82,82	87,90	79,92	85,51
5	$Stacklazy_{MST}$	Martins Pil. (egit.)		LR	82,77	87,34	80,77	85,34
6	Stacklazy	ZT	LibSVM	LR	82,59	87,22	80,66	85,38
7	Bigarren Mailako MST Ez-proiektiboa		MIRA	LR	82,53	87,77	80,17	86,01
8	Stacklazy		LibSVM	LR	82,39	86,59	79,81	84,41
9	Covington	ZT	LibLINEAR	LR	82,24	86,39	80,25	84,95
10	$Stacklazy_{Covington}$	Martins Pil. (egit.)	LibLINEAR	LR	82,17	86,62	80,02	84,78
11	$Stacklazy_{MST}$	Martins Pil. (ling.)		LR	82,05	86,16	79,52	83,77
12	$Stacklazy_{Stacklazy}$	Martins Pil. (egit.)	LibLINEAR	LR	81,97	86,45	79,34	84,06
13	$Stacklazy_{Stacklazy}$	Bengoetxea Pil. (ling.)	LibLINEAR	LR	81,95	86,08	79,36	83,61
14	$Stacklazy_{Stacklazy}$	Martins Pil. (ling.)	LibLINEAR	LR	81,94	86,02	79,24	83,53
15	$Stacklazy_{Covington}$	Martins Pil. (ling.)	LibLINEAR	LR	81,89	86,25	79,31	83,66
16	Nivre Arku Estandarra + Pseudo		LibLINEAR	LR	81,50	86,06	78,89	83,63
17	Stacklazy		LibLINEAR	LR	81,50	86,04	78,79	83,43
18	Lehen Mailako MST Ez-proiektiboa		MIRA	LR	81,32	86,99	79,01	85,08
19	Stacklazy		LibSVM	RL	80,94	85,61	78,23	83,60
20	Nivre Arku Estandarra	ZT	LibLINEAR	LR	80,64	84,50	78,23	82,68
21	Covington		LibLINEAR	LR	80,61	85,05	78,71	83,50
22	Stacklazy		LibLINEAR	RL	80,58	85,38	77,66	83,46
23	Nivre Arku Estandarra		LibLINEAR	LR	80,57	85,24	77,98	82,84
24	Nivre Arku Estandarra + Pseudo		LibLINEAR	RL	80,43	85,39	78,41	84,15
25	Nivre Arku Estandarra		LibLINEAR	RL	79,22	84,50	77,02	82,85
26	Covington		LibLINEAR	RL	78,34	83,43	76,10	81,46

4.26 taula – Urre-patroiko ezaugarriak erabilita, 26 oinarrizko sistema eta sistema hedatu onenak ordenatuta garapen datuekin, LAS neurriaren arabera (RL = Eskuinetik eskerrera; LR = Ezkerretik eskuinera; Pseudo = Pseudo-proiektiboa; ZT = Zuhaitz-Transformazioak; egit. = egitura ezaugarriak; ling. = ezaugarri linguistikoak; Bengoetxea Pil. = Bengoetxea eta Gojenola-ren (2009a) Pilaketa).

Ord.	Algoritmoa	Aldaera	Ikasketa	Nor.	Garapena		Testa	
					LAS	UAS	LAS	UAS
1	$MST_{Stacklazy}$	Martins Pil. (ling.)	LibLINEAR	LR	79,90	85,73	78,26	84,28
2	$Stacklazy_{MST}$	Martins Pil. (egit.)		LR	79,71	85,27	78,55	83,86
3	Bigarren Mailako MST Ez-proiektiboa	ZT	LibLINEAR	LR	79,67	85,22	78,68	84,43
4	Nivre Arku Estandarra + Pseudo	ZT	LibLINEAR	LR	79,64	84,48	79,13	84,16
5	Stacklazy		LibSVM	LR	79,56	84,33	78,35	83,35
6	Bigarren Mailako MST Ez-proiektiboa		MIRA	LR	79,30	85,28	77,88	84,08
7	Stacklazy	ZT	LibSVM	LR	79,00	83,74	78,56	83,49
8	Covington	ZT	LibLINEAR	LR	78,99	83,59	78,45	83,32
9	$Stacklazy_{Covington}$	Martins Pil. (egit.)	LibLINEAR	LR	78,95	84,13	77,52	82,92
10	Lehen Mailako MST Ez-proiektiboa		MIRA	LR	78,91	84,83	77,24	83,24
11	Stacklazy	ZT	LibLINEAR	LR	78,90	83,89	78,41	83,59
12	$Stacklazy_{MST}$	Martins Pil. (ling.)		LR	78,89	83,81	77,20	82,34
13	$Stacklazy_{Stacklazy}$	Martins Pil. (ling.)	LibLINEAR	LR	78,69	83,49	77,22	82,23
14	$Stacklazy_{Covington}$	Martins Pil. (ling.)	LibLINEAR	LR	78,57	83,41	77,45	82,34
15	$Stacklazy_{Stacklazy}$	Martins Pil. (egit.)	LibLINEAR	LR	78,30	83,49	77,18	82,50
16	$Stacklazy_{Stacklazy}$	Bengoetxea Pil. (ling.)	LibLINEAR	LR	78,23	82,97	77,34	82,24
17	Nivre Arku Estandarra + Pseudo		LibLINEAR	LR	78,21	83,09	77,31	82,62
18	Stacklazy		LibLINEAR	LR	77,90	82,84	76,83	82,18
19	Stacklazy		LibLINEAR	RL	77,85	83,32	76,54	82,86
20	Stacklazy		LibSVM	RL	77,84	83,30	77,09	82,92
21	Nivre Arku Estandarra		LibLINEAR	LR	77,63	82,60	76,14	81,60
22	Nivre Arku Estandarra		LibLINEAR	RL	77,49	83,07	76,40	82,77
23	Covington		LibLINEAR	LR	77,33	82,33	76,98	82,17
24	Nivre Arku Estandarra	ZT	LibLINEAR	LR	77,06	81,61	76,47	81,16
25	Nivre Arku Estandarra		LibLINEAR	RL	76,30	82,18	75,38	81,92
26	Covington		LibLINEAR	RL	75,35	81,31	74,61	80,73

4.27 taula – Ezaugarri automatikoak erabilita, 26 oinarrizko sistema eta sistema hedatu onenak ordenatuta garapen datuekin, LAS neurriaren arabera (RL = Eskuinetik eskerrera; LR = Ezkerretik eskuinera; Pseudo = Pseudo-proiektiboa; ZT = Zuhaitz-Transformazioak; egit. = egitura ezaugarriak; ling. = ezaugarri linguistikoak; Bengoetxea Pil. = Bengoetxea eta Gojenola-ren (2009a) Pilaketa).



4.17 irudia – Oinarrizko sistema eta sistema hedatuekin osatutako 14 sistema onenen emaitzak LAS neurrian. Urre-Ausaz-Onena = Urre-patroiko ezaugarri morfologikoekin Ausazko Onena; Auto-Ausaz-Onena = ezaugarri morfologiko Automatikoekin Ausazko Onena; Urre-x-Onenen-Konb = Urre-patroiko ezaugarri morfologikoekin x Onenen Konbinaketa; Auto-x-Onenen-Konb = ezaugarri morfologiko Automatikoekin x Onenen Konbinaketa

	Urre-patroiko ezaugarriekin		Ezaugarri automatikoekin	
Kop. (n)	**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)

4.28 taula – 14 Sistema onenak.

Sistema hedatuen irteerak konbinaketan sartzeak zenbaterainoko eragina du zehaztasunean? 4.28 taulan, urre-patroiko ezaugarri morfologikoekin eta ezaugarri automatikoekin x onenen konbinaketa edo ausazko onena metodoa erabiltzerakoan, 14 sistemen (oinarrizko eta hedatuen) irteerak multzo posible guztietan (3, ...,14) konbinatu ostean lortutako emaitzak agertzen dira. 14 sistema onenen aniztasuna aztertzerakoan, urre-patroiko ezaugarriekin 5 zuhaitz-transformazio, 7 pilaketa eta 2 oinarrizko sistemen irteerak bildu dira, eta, ezaugarri automatikoekin, 5 zuhaitz-transformazio, 6 pilaketa eta 3 oinarrizko sistemen irteerak. 4.28 taulan, sistema hedatuen irteerak konbinaketan sartzeak emaitzak hobetzen dituela ikus daiteke. Adibidez, sistema hedatuak gehituz egindako konbinaketetan, urre-patroiko ezaugarriak eta ezaugarri automatikoekin eta ausazko onenaren konbinaketarekin +3,07 eta +4,14ko irabaziak lortu dira; oinarrizko sistemak konbinatu direnean, aldiz, +2,82ko irabaziak baino ez dira lortu. 4.17 irudian, sistema hedatuak sartzerakoan x onenen konbinaketaren eta ausazko onena konbinaketaren artean dagoen aldea murriztu egiten dela ikus daiteke, eta oso nabarmen sistema gutxi konbinatzen direnean, MaltBlender sistema orekatuago bat lortuz.

Sistema hedatuekin, zein sistemaren artean ematen dira emaitzarik onenak? Izaera edo algoritmo desberdineko sistemetan, eskuinetik ezkerrera edo ezkerretik eskuinera sortutako modeloek eragina dute konbinaketan? Ordenak badu garrantzirik? Oraingo honetan, aurreko oinarrizko analizatzaileez gain, analizatzaile hedatuak gehitutako 14 sistema onenekin, 4.29 taulan,

ezaugarri automatikoekin ausazko onenaren konbinaketarekin 3tik 9ra arte lortutako sistemen zerrenda gehitu da. 4.29 taulan, konbinaketa onena 1, 3, 4, 5, 7, 8, 10, 12 eta 20 sistemak konbinatuz lortzen da. Ia konbinaketa guztietan agertzen diren sistemak 3, 4, 5, 8 eta 20 dira:

- Bigarren mailako MST ez-proiektibo algoritmoa eta zuhaitz-transformazio aldaera
- Nivre Arku Estandar algoritmoa, pseudo-proiektibo eta zuhaitz-transformazio aldaerekin
- Stacklazy algoritmoa, LibSVM ikasketa eta ezker-eskuin norabidea
- Covington algoritmoa eta zuhaitz-transformazio aldaera
- Stacklazy algoritmoa, LibSVM ikasketa eta eskuin-ezker norabidea

Horrela, konbinaketak hurrengo sistema bilatzen ditu:

- **Sistema dibergenteak**, MSTParser eta MaltParser izaera desberdineko (globala eta lokala) sistemak. Adibidez, konbinaketa guztietan agertzen dira 3 edo 10 MSTParser sistemako sistemak.
- **Norabide desberdineko sistemak**, esaldiak ezkerretik eskuinera edo alderantziz analizatzea aberasgarria da konbinaketan, nahiz eta emaitza nabarmen jaitsi eskuinetik ezkerrera egiten denean. Adibidez, 5 eta 20 azkeneko 4 konbinaketa onenen artean agertzen dira.

Datuei begiratu sakon bat emanez gero, 14 sistemen (oinarrizko eta hedatuen) konbinaketa kontuan hartuta, 9 sistemen konbinaketatik onena lortutako emaitza, gainerako n sistemen konbinaketatik onenaren pare gelditzen dela ikusi da, eta 5 zuhaitz-transformazio, 6 pilaketa eta 3 oinarrizko sistemen irteeren konbinaketa multzotik, konbinaketa onena 4 zuhaitz-transformazio, 2 pilaketa eta 3 oinarrizko sistemen irteeren multzoarekin lortu da; hau da, azken buruan, aniztasuna mantentzen dela esan daiteke.

Sistema kopurua (n)	**x onenen konb.**	**Ausazko onena**	**Zerrenda (Ausazko onena)**
3	80,53 (+2,27)	80,96 (+2,70)	3, 4, 5
4	80,90 (+2,64)	80,92 (+2,66)	5, 8, 10, 12
5	81,87 (+3,61)	81,59 (+3,33)	1, 3, 4, 8, 20
6	81,66 (+3,40)	81,91 (+3,65)	2, 3, 4, 5, 7, 20
7	81,86 (+3,60)	82,22 (+3,96)	1, 3, 4, 5, 7, 8, 20
8	82,11 (+3,85)	82,08 (+3,82)	3, 4, 5, 7, 8, 9, 10, 20
9	82,31 (+4,05)	82,40 (+4,14)	1, 3, 4, 5, 7, 8, 10, 12, 20

4.29 taula – Ezaugarri automatikoak erabiliz, oinarrizko sistemen, n azpimultzoak 3tik 9ra arte konbinatu ostean lortutako emaitzak LAS neurrian.

4.3.4 Ondorioak

Analizatzaile sintaktikoaren zehaztasuna hobetzeko bozketa bidezko konbinaketarekin probatu da. Sistema ezberdinen irteerak konbinatzeko, MaltBlender boto sistema erabili da. Tesi-lan honetan x onenen konbinaketa eta ausazko onena konbinaketa probatu dira, bai oinarrizko sistemekin, bai sistema hedatuekin. Emaitzak aztertuta, sarreran egindako galderak erantzuten ahaleginduko gara:

- **Konbinatzeko garaian, analizatzaileen LAS puntuazioaren araberako ordenak edo aniztasunak (osagarritasunak) garrantzi handiagoa du?** x onenen konbinaketa eta ausazko onenen konbinaketa probatu dira. Oinarrizko algoritmoekin 12 aldaera ezberdineta-ko analizatzaileen multzoa osatu da eta irteera multzo posible guztiak (3, ...,12) konbinatu ostean, hurrengo ondorioak atera ditugu: ausazko onena metodoarekin, x onenen konbinaketarekin baino emaitza hobeak lortu dira; ausazko onena metodoarekin 6 analizatzaileren irteerekin nahiko da; aldiz, x onenen konbinaketarekin 10 analizatzaileren irteerak behar izaten dira; eta, azkenik, ausazko onena metodoarekin lortutako konbinaketa onena osatzen duten analizatzaileen multzoan, inoiz ez dela x onenen konbinaketa multzoa agertzen gertatu da. Horrela, bozketa sistemetan LAS neurria soilik kontuan hartu beharra baztertuko da, sistema konbinatu onenen artean, beti aurkitzen baitugu LAS sailkapenaren bukaeran daudenak; horretaz gainera, beste faktore garrantzitsu batzuk aurki daitezke: bateragarritasuna, aldaera eta kalitatea (Surdeanu eta Manning, 2010).
- **Oinarrizko sistemekin, zein da hobekuntza? Zein sistemaren artean ematen dira emaitzarik onenak?** 12 oinarrizko sistemen irteerak ausazko onena konbinaketarekin, urre-patroiko ezaugarriekin eta ezaugarri automatikoekin 82,99 eta 81,17 lortu dira LAS neurrian, hurrenez hurren, eta bakarkako oinarrizko sistema onenarekin alderatuz gero ia 3 puntuko hobekuntza lortu da. Konbinaketa onena osatzen duten sistemei erreparatuz, alde batetik esan daiteke MSTParser eta MaltParser izaera desberdineko (globala eta lokala) sistemak agertzen direla beti; beste aldetik, esaldiak ezkerretik eskuinera edo alderantziz entrenatutako analizatzaileen irteerak konbinatutako sistemak agertzen dira, nahiz eta eskuinetik ezkerrera entrenatutako sistemen irteeren

emaitzak LAS neurrian askoz ere baxuagoak izan.

- **Zein da hobekuntza, sistema hedatuen irteerak konbinaketan sartzean? Zein sistemaren artean ematen dira emaitzarik onenak?** Esperimentuak egiteko 26 analizatzaile ezberdin lortu dira (oinarrizkoak + hedatuak), baina 26 sistemen azpimultzo posible guztiak kalkulatzeko mementuan, esperimentuen kopurua izugarri igotzen zenez, 14 sistema onenak erabiltzea erabaki da. 14 sistema onenen artean aniztasuna mantentzen dela probatu da, urre-patroiko ezaugarriekin 5 zuhaitz-transformazio, 7 pilaketa eta oinarrizko 2 sistemen irteerak bildu dira, eta, ezaugarri automatikoekin, 5 zuhaitz-transformazio, 6 pilaketa eta oinarrizko 3 sistemen irteerak bildu dira. 14 sistemen (oinarrizko eta hedatuen) konbinaketa kontuan hartuta, 9 sistemen konbinaketatik onena lortutako emaitza aztertzerakoan, 5 zuhaitz-transformazio, 6 pilaketa eta oinarrizko 3 sistemen irteeren konbinaketa multzotik, konbinaketa onena 4 zuhaitz-transformazio, 2 pilaketa eta oinarrizko 3 sistemen irteeren multzoarekin lortu da; hau da, azken buruan, aniztasuna mantentzen dela esan daiteke. Hemen ere, oinarrizko analizatzaileen konbinaketan bezala MSTParser eta MaltParser izaera desberdineko (globala eta lokala) sistemak eta norabide desberdineko sistemak aurkituko ditugu. Sistema hedatuak gehituz egindako konbinaketetan, urre-patroiko ezaugarriak eta ezaugarri automatikoekin eta ausazko onenaren konbinaketarekin +3,07 eta +4,14ko irabaziak lortu dira; oinarrizko sistemak konbinatu direnean, aldiz, +2,82ko irabaziak baino ez dira lortu.
- **Konbinaketa egoera erreal batean erabili ahal da?** Urre-patroiko etiketak erabili beharrean, etiketa automatikoak erabiliz gero hobekuntza esanguratsuak lortu dira. Emaitza bat aipatzearren, ausazko onena metodoarekin 82,40 (+4,14 oinarri onenarekiko) lortu da.

Euskarako analisi sintaktikoa hobetzeko datuetan eta informazio linguistikoan oinarritutako teknikak (zuhaitz transformazioak, pilaketa eta bozketa sistema) erabiliz zehaztasuna handitzea lortu da.

4.4 Laburpena eta ondorengo pausoak

Labur-zurrean, tesi kapitulu honetan euskararako analizatzaile sintaktikoaren emaitzak hobetzeko hurrengo ekarpenak egin dira:

- 4.1 atalean, euskararekin bat datozen sintagmen transformazioa, men-

deko perpausen transformazioa eta MES koordinazio-transformazioa proposatu dira.

- 4.2 atalean, egiturazko ezaugarriak goratzeaz gain (Martins *et al.*, 2008), ezaugarri linguistikoak goratzeko ekarpena egin da.
- 4.3 atalean, analizatzaile ezberdinak konbinatzeko bozketa bidezko teknika erabili da. Atal honetan, oinarrizko sistemen eta sistema hedatuen (pilaketa eta zuhaitz-transformazio) irteerak integratzeko baliatu da teknika hau.
- Azkenik, aipatu behar da hemen erabilitako teknika guztiak ezaugarri morfologiko automatikoekin probatu direla.

Hurrengo kapituluan, analizatzailearen emaitzak hobetzeko informazio semantikoa erabiliko da; horretarako, ezinbestekoak diren kanpoko baliabide semantikoak (ezagutza-base lexiko-semantikoak eta corpusak) baliatuko dira.

5

Informazio semantikoa eta analisi sintaktikoa

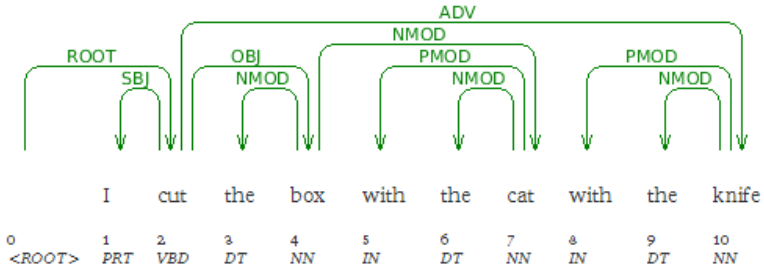
Aurreko kapituluetan, besteak beste, dependentzien analisia hobetzeko hainbat bide jarraitu dira, ezaugarrien hautaketa, zuhaitz-transformazioak, pilaketa eta bozketa teknikak probatu dira. Teknika hauek guztiek, zuhaitzbankutik eta analizatzaile morfosintaktikoen datuak erabili dituzte, eta informazio horren mugak gainditzen egin da indarra. Hortik haratago mugitzeko, kanpoko baliabideen ekarpena aztertuko da kapitulu honetan. Orain arte, analisirako ezaugarri morfosintaktikoen informazioa erabili da; alabaina, hizkuntzaren prozesamenduan, aspalditik, egitura sintaktikoen desanbiguazioan, hitzen adieren desanbiguazioan, analizatzaile sintaktikoaren lana hobetzeko informazio semantikoa erabiltzea pentsatu da. Hizkuntzaren prozesamenduan semantika jorratu ahal izateko ezinbestekoa da baliabide semantikoak garatzea (adibidez, ezagutza-base lexiko-semantikoak, hau da, hitzei eta adierei buruzko informazioa duten baliabide lexikal egituratuak, eta abar). IXA taldean euskararako baliabide semantikoak garatzen diren bitartean, lehenengo eta behin ingeleserako eskuragarri dauden baliabide semantikoak probatzea pentsatu da. Horrela, kapitulu honetako lehenengo atalean, informazio semantikoak analisian nola lagundu dezakeen ikusiko da, eta erabilitako baliabideen zergatia eta esperimentuen nondik norakoak zeintzuk izan diren azalduko dira. Bigarren atalean, analizatzaile sintaktikoekin egindako antzerako lanak aipatuko dira. Hirugarren atalean, berriz, erabilitako baliabideak aurkeztuko dira, eta, azkeneko ataletan, egindako esperimentuekin lortutako emaitzak eta ateratako ondorioak azalduko dira.

5.1 Sarrera

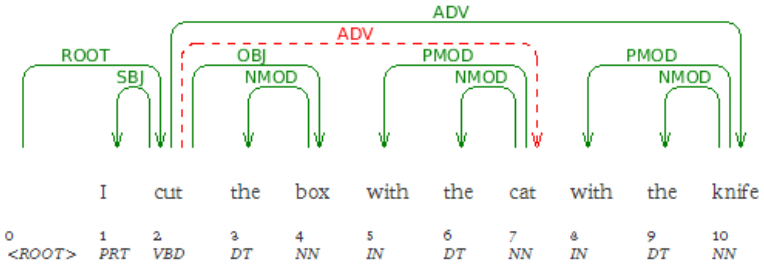
Lexikalizatutako analizatzaile sintaktikoek hitzak orokortzeko daukaten muga gainditze aldera, orain arte, kategoria, azpikategoria eta ezaugarri morfologikoak erabili izan dira, baina azken urteotan informazio semantikoa erabiltzeko ahalegin ugari egin dira. Informazio semantikoak nola lagun dezakeen ikusteko, preposizio-sintagma bi dituen *I cut the box with the cat with the knife* esaldia erabiliko da. 5.1a irudian, dependentzia-zuhaitza egokian, *with the cat* preposizio-sintagma *box* izenaren mende agertzen da, eta *with the knife* preposizio-sintagma, berriz, *cut* aditzaren mende agertzen dela ikus daiteke. 5.1b irudian analizatzaile baten irteera ikus daiteke. Gorriz dagoen erlazioan, analizatzaileak gaizki lotu duen erlazioa agertzen da, eta *with the cat* preposizio-sintagma *cut* aditzaren mende agertzen dela ikus daiteke. Zergatik egiten du gaizki? Analizatzailea kategoria-informazioa erabiltzen ari da eta *cat* eta *knife* hitzek *NN* izen kategoria bera dutenez, analizatzaileak ez dauka bien artean bereizteko informaziorik. 5.1c irudian, izen guztiei *WordNet*eko (Fellbaum, 1998) informazio zabalena gehitu zaie, hau da, *knife*ren ezaugarriei *artifact* eta *cat*en ezaugarriei *animal* gehitu zaie, eta analizatzailea bi hitzen artean bereizteko gai dela eta analisia era egokian gauzatu duela ikus daiteke.

Kapitulu honen helburu nagusia izango da informazio semantikoak dependentzia-ereduko analisian izan dezakeen eragina probatzea eta helburu horretarako hurrengo galderei erantzuteko ahaleginak egin dira:

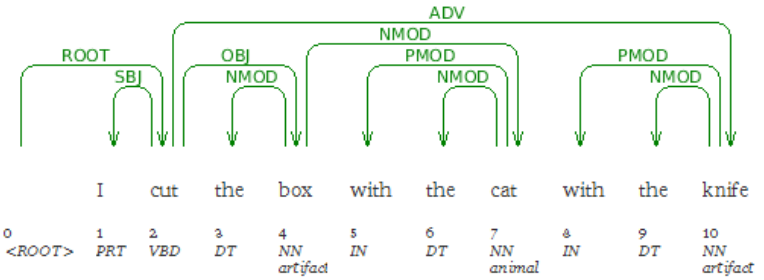
- **Informazio semantikoa analizatzaile sintaktikoa hobetzeko baliagarria da?** IXA taldean, euskararako baliabide semantikoak garatzen diren bitartean, ingeleserako eskuragarri dauden baliabide semantikoak erabiltzea pentsatu da. Informazio semantikoa gehitzeko orduan, oro har, bi iturri ezberdin erabiltzen dira: corpusak eta ezagutza-base lexiko-semantikoak (*WordNet* edo antzerako ontologiak). Galdera honi erantzuteko, iturri mota desberdineko eta ingeleserako eskuragarri dauden hurrengo baliabideak erabili dira:
  - (Koo *et al.*, 2008) lanean etiketatu gabeko corpusetik lortutako hitz-multzoak
  - (Agirre *et al.*, 2008) lanean *WordNet*etik (ingeleseko EBLStik) lortutako klase semantikoak
- **Zein mailatako informazio semantikoa erabiltzea komeni da?** *WordNet*eko hitz eta adierei buruzko informaziotik adierarik zabalena eta zorrotzena eta etiketatu gabeko corpusetik lortutako hitz-multzoe-



(a) dependentzia-zuhaitza egokia



(b) analizatzailearen irteera, informazio semantikorik gabe



(c) analizatzailearen irteera, informazio semantikoarekin

5.1 irudia – Izenei gehitutako *WordNet*eko informazio zabalena.

tatik maila desberdinetako hitz-multzoak erabili dira.

– **Benetako egoera batean baliagarriak dira?** Urre-patroiko kategoriak soilik erabili beharrean, kategoria etiketatzaile batek (ingelesez, *POS tagger*) iragarritako kategoria automatikoekin probatu da.
– **Maila desberdineko informazio semantikoak izaera desberdineko sistemetan eragin berdina izango ote du?** Izaera desberdineko hiru sistemekin lan egin da: trantsizioetan oinarritutako analiza-

tzaile determinista (MaltParser), grafoetan oinarritutako analizatzaile globala (MSTParser) eta trantsizioetan oinarritutako ZPar, bilaketa lokala, determinista eta jalea erabili beharrean, ikasketa diskriminatzaile globala eta *beam* bilaketa erabiltzen duena, alegia.

– **Zein bihurketa tresna da egokiena informazio semantikoarekin lan egiterakoan?** *Penn Treebank*a osagai-eredutik dependentzia-eredura bihurtzeko bihurtzaile anitz daude. Oro har, bihurtzaile mota bi bereizten dira: dependentzia-etiketa aberatsagoak eta finagoak ematen dituztenak, adibidez, *Pennconverter*[1] (Johansson eta Nugues, 2007b) eta dependentzia-etiketen multzo murriztua ematen dutenak, adibidez, *Penn2Malt*[2] (Yamada eta Matsumoto, 2003). Galdera honi erantzuteko, mota bakoitzeko *Penn2Malt* eta *Pennconverter* bihurtzaileak erabiliko dira.

– **Konbinaketan informazio semantikoa erabiltzea aberasgarria da?** Oinarrizko sistemen eta informazio semantikoekin entrenatutako sistemen irteerak bozketa bidezko MaltBlender tresnarekin konbinatu dira galdera honi erantzun ahal izateko.

Hurrengo ataletan zerrendatutako aurreko galdera horiei erantzuna emango zaie.

5.2 Antzerako lanak

Informazio semantikoa lortzeko erabilitako iturriaren arabera, talde nagusi bi daude, hots, ezagutza-base lexiko-semantikoetatik (*WordNet* edo antzerako ontologiatik) eta corpusetatik lortutako informazio semantikoa erabiltzen duten hurbilpenak.

1. **WordNet**, lexikalizatutako analizatzaileek hitzak orokortzeko daukaten muga gainditze aldera informazio semantikoa erabiltzeko ahalegin ugari egin dira (Magerman, 1995; Collins, 1996; Charniak, 1997; Collins, 2003). Lan batzuk aipatzearren, (MacKinlay *et al.*, 2012) lanean esaterako, hiperonimoak gehitzerakoan dependentzia-etiketetan errore tasa % 1 *F-score* neurrian gutxitu zuten *HPSG parse ranking*ean, nahiz eta beste metodo batzuetan zehaztasuna asko jaitsi; (Fujita *et al.*, 2010) lanean, ontologiatik guztiz desanbiguatutako adieretan oinarri-

[1] http://nlp.cs.lth.se/software/treebank_converter/
[2] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

tutako ezaugarriak erabiltzea eraginkorra zela probatu zuten analizatzaile aukeraketan (ingelesez, *parse selection*); eta (Agirre *et al.*, 2008) lanean, % 6,9 errore gutxiago izatea lortu zuten egitura osoan, eta *pp-attachment*en % 20,5 errore gutxiago izatea lortu zuten *Wall Street Journal*eko eta *SemCor*eko (Landes *et al.*, 1998) ebakiduraren gainean eta Bikel-en *parser* modeloarekin (Charniak, 2000; Bikel, 2004). Tesi-lan honetan, (Agirre *et al.*, 2008) lanean egindakoari jarraituz, *WordNet*eko klase semantikoak *Penn Treebank (PTB)* eta *SemCor/PTB* ebaketari gehitu zaizkie, eta dependentzietan oinarritutako izaera desberdineko sistemekin, urre-patroiko kategoriekin eta kategoria etiketatzaile batetik lortutako kategoria automatikoekin probatu dira.

2. **Corpusetik lortutako informazio semantikoa** erabiltzen dutenen artean, hitz-multzokatze lanak daude. (Koo *et al.*, 2008) lanean, *PTB* eta *Prague Dependency Treebank*ean agerpen gehien zituzten 800 hitzen hitz-multzoak gehitu zituzten. Hitz-multzoak lortzeko Brown-en algoritmoa (Brown *et al.*, 1992) eta etiketatu gabeko BLLIP corpusa (Charniak, 2000) erabili ziren (azken honek *Wall Street Journal*eko 43 milioiko hitz kopurua du). Analizatzaile sintaktikoan duen eragina probatzeko, lehen eta bigarren mailako faktorizazioko *Maximum Spanning Tree (MST)* algoritmoa eta ikasketa automatikorako *Averaged Perceptron* erabili zuten. Sartutako hitz-multzoen artean, informazio semantiko sendoko hitz-multzoak {apple, pear} edo {Apple, IBM} bezalakoak zeuden, bai eta informazio sintaktiko sendoko hitz-multzoak {of, in} bezalakoak ere. *PTB* eta *Prague Dependency Treebank* zuhaitz-bankuen gainean egindako esperimentuetan, hitz-multzoak eraginkorrak zirela probatu zen. Ildo beretik, (Suzuki *et al.*, 2009, Sagae eta Gordon, 2009, Candito eta Seddah, 2010 eta Täckström *et al.*, 2012) lanetan ere hitz-multzoak erabilgarriak direla probatu izan da. Tesi-lan honetan, (Koo *et al.*, 2008) lanean lortutako hitz-multzoak erabili dira.

5.3 Baliabideak

Atal honetan, esperimentuak gauzatzeko erabilitako baliabideak aurkeztuko dira:

5.3.1 Datuak eta bihurtzaileak

Esperimentuak egiteko *PTB* osoa (ikusi 2.2.2.1 atala) eta *SemCor/PTB* ebaketa erabili dira. *SemCor/PTB SemCor* corpusean (Landes *et al.*, 1998) eta *Penn Treebank* zuhaitz-bankuan agertzen den ebaketa da. Ebaketa hau erabiltzeko arrazoi nagusia da *SemCor*en zentzu-informazioa integratzen dela eta *Penn Treebank*ean egitura sintaktikoena. Horrela, (Agirre *et al.*, 2008) lanean, *SemCor*eko datuak eta *Penn Treebank*eko informazioaren nahasketa egin ostean eta token ezberdinak erabilitako esaldiak baztertu ostean, 8.669 esaldirekin eta 151.928 hitzekin osatutako *Semcor/PTB* ebaketa erabili da. *PTB* osoa osagai-ereduan etiketatuta dago eta MaltParser, MSTParser eta Zpar dependentzietan oinarritutako analizatzaileak dira. Hori dela-eta, osagai-zuhaitzak dependentzia-zuhaitz bihurtzeko bihurketa mota bi erabili dira:

- *Pennconverter*: *LTH* bihurtzailea ere deitzen da. Nahiz eta, beste bihurtzaile batzuk baino irteera finago eman, oro har, onenetarikoa da semantikaren prozesamendurako (Johansson eta Nugues, 2007a), egitura edo dependentzia-etiketa aberatsagoa eta finagoa lortzen baita. 42 dependentzia-etiketa sortzen ditu, eta horietatik batzuk urruneko erlazioetan ematen diren fenomenoak dira, topikalizazioa eta *wh-movement* bezalakoak. *LTH* bihurtzaile honek arku ez-proiektibo ugari sortzen ditu, *PTB* zuhaitz-bankuko entrenamenduko 39.832 esaldietatik 2.459 esaldi ez-proiektibo sortzen ditu, hau da, esaldien % 6,17. Esperimentuak egiteko, *CoNLL 2007 Shared Task on Dependency Parsing* lehiaketako konfigurazio[3] onena erabili da.
- *Penn2Malt* bihurtzailea eta Yamada eta Matsumoto-ren (2003) buru-erregela orokorrak erabiliz, 12 dependentzia-erlazioko multzo murriztua lortu da. Bihurketa honetan lortzen diren arku guztiak proiektiboak dira. Esperimentua egiteko erabilitako konfigurazioa[4] lan askotan erabili da, eta emaitzak antzerako lanekin (Koo *et al.*, 2008; Suzuki *et al.*, 2009; Koo eta Collins, 2010) konparatzeko baliagarri izango dira.

LTH bihurtzailearekin analisi sintaktikoan lortutako emaitzak *Penn2Malt* bihurtzailearekin lortutakoak baino txarragoak dira. *PTB* osoa erabiltzerakoan kategoria eta azpikategorian balio berdinak erabili dira, baina urre-patroiko kategoria balioak erabiltzeaz gain, *Perceptron* kategoria etiketatzailea (Collins, 2002) erabili da kategoria automatikoekin probatzeko. Kategoria

[3] java -jar pennconverter.jar -old*LTH* <tree >malt

[4] java -jar Penn2Malt.jar penn.23.1-2 headrules 3 2 penn

etiketatzailea entrenatzeko *Wall Street Journal*eko 2-21 sekzioak erabili dira, eta entrenamenduko eta probako multzoari automatikoki kategoria esleitzeko *jackkniffing 10-way* teknika baliatuta (Chen *et al.*, 2013) lanean lortutako kategoriak erabili dira. *PTB* osoarekin esperimentuak egiteko orduan, *Wall Street Journal*eko hiru multzo erabili dira: entrenatzeko 2-21 sekzioak, lantzeko 22. sekzioa eta proba egiteko 23. sekzioa. Ildo beretik, *SemCor/PTB* ebaketa hiru zatitan banatu da: entrenatzeko % 80, lantzeko % 10 eta azkeneko proba egiteko datuen % 10.

5.3.2 Informazio semantikoa

Analisi sintaktikoa hobetzeko, iturri desberdinetatik ateratako informazio semantikoa erabili da. Alde batetik, *WordNet*etik lortutako muturreko errepresentazio bi, *synset*ak (adiera zorrotzena) eta fitxategi semantikoak (adiera zabalena), erabili dira. Beste aldetik, corpusetik automatikoki lortutako hitz-multzoak erabili dira.

WordNet

WordNet synset edo sinonimo multzoen arabera antolatuta dagoen hierarkia kontzeptuala da. Adibidez, "labana, aizto –gauzak mozteko erabiltzen den tresna zorrotza–" adieran, "labana" eta "aizto" hitzek kontzeptu bera adierazten dute. *WordNet*eko erlazio semantiko garrantzitsuenetako bat sinonimia da. Sinonimia ale lexikalaren adieran dago, eta adiera horrek ale lexikal bat baino gehiago duenean, ale lexikalak multzokatu egiten dira. Baina sinonimia erlazioarekin batera, beste hainbat erlazio daude: hiperonimia-hiponimia erlazioak (*synset* orokorrenak *synset* zehatzagoekin lotzen dituztenak), antonimia, *synset* antonimoak lotzen dituztenak, eta abar. *WordNet* ontologia edo hierarkia bat da, eta hiperonimia-hiponimia harreman semantikoarekin hierarkian gora eta behera egiteko aukera dago. Arazoa maila egoki bat arte aukeratzean datza. Esperimentuak egiteko, (Agirre *et al.*, 2008) lanean *WordNet* 1.7tik ateratako muturreko errepresentazio semantikoak erabili dira, *synset*ak (zorrotzena) eta fitxategi semantikoak (zabalena). Adibidez, "labana" hitzaren kasuan, *synset*a "gauzak mozteko erabiltzen den tresna zorrotza" bada, fitxategi semantikoaren informazioa "IZEN.TRESNA" izango da. Horrez gain, fitxategi semantikoen eta hitzen arteko irudikapen nahastua edo hibridoa ere erabili da. Irudikapen hibridoak hitzaren adiera ezberdinen artean bereizteko balio du. Adibidez, *crane* hitza ingelesez zikoina edo garabia izan daiteke, eta *demolish a house with a crane* esaldian irudikapen hibri-

doak "CRANE+IZEN.TRESNA" edo "CRANE+IZEN.ANIMALIA" erabiliz gero sistemak hitzaren adiera ezberdinen artean hobeto bereizteko balio duen probatuko da. *WordNet* 1.7an, *synset* bakoitzari fitxategi semantiko bakarra dagokio. Kategoria sintaktiko eta semantikoetan sailkatutako 45 fitxategi semantiko daude: 1 adberbioena, 3 adjektiboenak, 15 aditzenak eta 26 izenenak. Adibidez, izenen fitxategi semantikoek ekintza, animaliak, tresnak, eta abar adierazten dituzten izenak sailkatzen dituzte (ikusi 5.1 taulan).

fitxategi semantikoen identifikatzailea	**Definizioa**
adj.all	adjektibo guztien multzokatzeak
adj.pert	adjektibo erlazionalak (ingelesez, *pertainyms*)
adj.ppl	partizipio adjektiboak
adv.all	adberbio guztiak
noun.act	ekintza edo akzioa adierazten duten izenak
noun.animal	animaliak adierazten dituzten izenak
noun.artifact	gizonak sortutako objektuen izenak
...	...
verb.consumption	jatea eta edatea adierazten duten aditzak
verb.emotion	sentimenduak adierazten dituzten aditzak
verb.perception	ikusmena, entzumena, sentimenduak adierazten dituzten aditzak
...	...

5.1 taula – *WordNet*eko fitxategi semantikoen aukeraketa bat.

Klase semantikoa gehitzeko orduan, aukera asko daude: hitz guztien klase semantikoa gehitzea, lema berari klase semantiko bera edo ezberdina gehitzea, kategoriaren arabera klase semantikoa gehitzea, eta abar. Gure esperimentuak egiteko, hurrengo aukerak erabili dira:

- Izen, aditz, adjektibo eta adberbio kategorien klase semantikoak
- Soilik izen kategorien klase semantikoak
- Soilik aditz kategorien klase semantikoak

Horrela, bederatzi konbinaketarekin lan egin da:

1. guztien *synset*ak (GSS)
2. izenen *synset*ak (ISS)
3. aditzen *synset*ak (ASS)
4. guztien fitxategi semantikoak (GFS)
5. izenen fitxategi semantikoak (IFS)
6. aditzen fitxategi semantikoak (AFS)

7. hitzen forma eta guztien fitxategi semantikoak (HGFS)

8. hitzen forma eta izenen fitxategi semantikoak (HIFS)

9. hitzen forma eta aditzen fitxategi semantikoak (HAFS)

Adibidez, IFS klase semantikoek izenen adiera argituko dute, animalia, ekintza, eta abar den adieraziko dute. Hitz bakoitzari dagokion klase semantikoa esleitzeko orduan, hitz batek adiera bat baino gehiago izan dezake. Adibidez, *"crane"* hitza "zikoina" edo "garabia" izan daiteke eta *"demolish a house with a crane"* esaldian, *"crane"* hitzak bi esanahi izan ditzake, "TRESNA" edo "ANIMALIA". Testuinguruaren arabera desanbiguatuko duen metodoren bat behar da. *SemCor/PTB* ebaketan hitzen adiera-desanbiguaziorako (HAD) (ingelesez, *word sense disambiguation edo WSD*) 3 metodo erabili dira:

- Urre-patroiko *SemCor/PTB* ebaketa: *SemCor*en eskuz egindako adieren urre-patroiko etiketak daude, eta *SemCor PTB* zuhaitz-bankuaren azpimultzo bat izanik, adieren eta egitura sintaktikoen urre-patroiko etiketak izango dira.
- Esanahirik sarrienaren metodoa: hitz kategoria bati klase semantikoa gehitzeko orduan, *SemCor*en gehien agertzen den adiera gehitu da. Aditzak soilik infinitiboan agertzen zirela-eta, testuan agertzen ziren aditzaren forma erregularrak eta irregularrak ere tratatu ahal izateko kategoria etiketatzailea eta lematizatzailea den TreeTagger[5] (Schmid, 1995) erabili da. *PTB* osoa lematizatu ostean, klase semantiko gehiago lortu da (% 14).
- *Automatic Sense Ranking (ASR)*: hitz bakoitzaren adiera sarriena zein den erabakitzeko, (McCarthy *et al.*, 2004) datuak erabili dira.

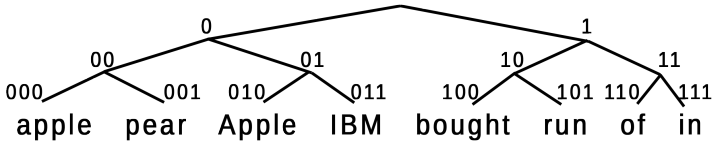
PTB osoan hitzen adiera-desanbiguaziorako esanahirik sarrienaren metodoa soilik erabili da, gainerako metodoak aplikatzeko daturik ez zegoelako.

Hitz-multzoak

Normalean, analisian hitz-formaren errepresentazio maila ez da nahikoa izaten; horregatik, beste errepresentazio maila batzuk erabiltzen dira, besteak beste, lema, kategoria, azpikategoria, eta abar. Gehien erabiltzen den errepresentazio maila hitzaren kategoria izaten da, baina kategoria erabiliz gero, "katua", "arkatza" eta "pertsona" bezalako hitzak maila berean geratzen dira.

[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Arazo honi aurre egiteko, tesi-lan honetan, hitzen antzekotasunaren errepresentazio maila erabili da. Bi hitzen arteko antzekotasuna bilatzeko algoritmo asko daude (Brown *et al.*, 1992; Li eta McCallum, 2005). Gure esperimentuak egiteko (Koo *et al.*, 2008) lanean lortutako hitz-multzoak erabili dira. (Koo *et al.*, 2008) lanean *Wall Street Journal*eko 43 milioiko hitz-kopurua duen etiketatu gabeko BLLIP corpusetik (Charniak, 2000) hitz-multzoak lortzeko, Brown-en algoritmoa (Brown *et al.*, 1992) inplementatzen duen Liangen (2005) aplikazioa erabili zuten. Algoritmoaren hasieran, hiztegiko hitz bakoitzak bere hitz-multzo propio ezberdina izango du. Behin eta berriro, algoritmoak hitz-multzo bikoteak (corpuseko testuaren probabilitatean gutxiengo jaitsiera eragiten dutenak) batuko ditu, klaseetan oinarritutako *bigram* hizkuntza-eredu baten arabera. Bikote pareak batuz, hitz-multzokatze hierarkikoa lortzen da (ikusi 5.2 irudian).



5.2 irudia – Brown-en hitz-multzoen zuhaitzaren adibide bat, Koo *et al.*-en (2008) lanetik ateratakoa.

Brown-en hitz-multzoen zuhaitzaren 5.2 irudian, hitz bakoitzak identifikatzaile propioa du, errotik hostora (hitzera) heltzeko bidea dena, eta bide hori bit-kate batekin adieraz daiteke, goiko irudian agertzen den modura. Bitarteko maila batera arte dauden nodoak multzokatzeko aukera dago, hau da, errotik sakonera batera arte dauden nodoak taldekatzeko aukera dago. Adibidez, errotik bigarren sakonera arte joanez gero edo lehenengo bi aurrizki bitak aukeratuz gero, lau hitz-multzo aurki daitezke: {apple, pear} 00 bit-katearekin lotuta, {Apple, IBM} 01 bit-katearekin lotuta, {bought, run} 10 bit-katearekin lotuta eta {of, in} 11 bit-katearekin lotuta daudenak. Horrela, luzera ezberdinetako bit-aurrizkiak aukeratuz, maila desberdineko hitz-multzoak lortuko dira (Miller *et al.*, 2004). Hitz-multzo hauen artean, informazio semantiko sendoko hitz-multzoak (adibidez, {apple,pear} eta {Apple, IBM}) edo informazio sintaktiko sendoko hitz-multzoak (adibidez, {of, in}) aurki daitezke. Gainera, hiztegi osoko hitzen hitz-multzoak ordezkatu beharrean, corpusean agerpen gehien dituzten hitzen hitz-multzoak ordezka daitezke. Zuhaitz-bankuan, Brown-en hitz-multzoen hierarkiatik ateratako informa-

zioa gehitzeko orduan, erabiliko den aurrizki kopurua eta corpusean gehien agertzen diren hitzen kopurua aukeratzea ez da lan tribiala. Adibidez, (Koo *et al.*, 2008) lanean, 100 eta 1.000 agerpen artean, analizatzaileen etekina hobea zela egiaztatu ostean, hiztegiko hitzak 1.000 klasetan banatu zituzten. Horrela, bit-kate guztia hartzerakoan ez da hitz-formaren baliokide izango. Gure tesi-lan honetan, (Koo *et al.*, 2008) lanean eta *PTB* corpusean agerpen gehien dituzten 1.000 hitzen hitz-multzoak erabili dira.

5.3.3 Analizatzaileak

Esperimentuak egiteko, izaera desberdineko dependentzietan oinarritutako 3 analizatzaile sintaktiko erabili dira:

- MaltParser: trantsizioetan oinarritutako analizatzaile sintaktikoen sortzailea da (ikusi 3.2.2.1 atala). *CoNLL* 2007 *Multilingual Dependency Parsing* lehiaketan ingeleserako MaltParser sistemarekin erabilitako konfigurazioan, ezaugarri berri bi gehitu dira, eta pila eta *buffer*eko buruan dauden hitzen klase semantikoak gehitu dira. Horrela, *CoNLL-X* formatuan agertzen den FEATS zutabean klase semantikoa gehitu ostean, konfigurazioan FEATS egozpen-funtzioari Stack[0] eta Input[0] helbide-funtzioak gehitu dizkiogu (ikusi 3.3.3 atalean).
- MSTParser: grafoetan oinarritutako analizatzaile sintaktikoen sortzailea da (ikusi 3.2.2.2 atala). Eisner-en algoritmoa (1996) erabiltzen duen bigarren mailako algoritmo proiektiboarekin lortu dira emaitzarik onenak. MSTParser sistemak ezaugarrien pisua ikasteko *Margin Infused Relaxed Algorithm (MIRA)* erabiltzen du. Klase semantikoak izan dezakeen eragina aztertzeko, sistema lehenetsiaren kodea aldatu da, arku bakoitzaren guraso eta umearen klase semantikoa hitz-forma eta kategoriarekin konbinatzeko.
- ZPar: grafoetan eta trantsizioetan oinarritutako analizatzaile sintaktikoen sortzaile hibridoa da (ikusi 2.3.3.2 atala). Zpar sistemak (Zhang eta Clark, 2008; Zhang eta Nivre, 2011) Arku Azkarra algoritmoaren trantsizio eta datu-egitura berdinak erabiltzen ditu (ikusi 3.2 taulan), baina une bakoitzean trantsizio egokiena zein den erabakitzen duen algoritmo jale eta lokala erabili beharrean, *beam-search* erabiliko da. *Beam-search* algoritmoak *parsing* urrats inkremental bakoitzean puntuazioak batu ostean, puntuazio maximoa lortzen duten B bide onenak gordetzen ditu. Bide onenen informazioa kudeatzeko egoera-elementuan gordetzen duen informazioa, besteak beste, sortutako zuhaitz-egi-

tura partziala, pila egitura, *buffer* egitura eta puntuazioa izango dira. Modu horretan, trantsizio posible guztien egoera-elementuak agenda batean gordetzen ditu, eta puntuazio altuena daukaten B onenak erabiliko dira hurrengo trantsizio-segida egiteko. Azkenean, agendako trantsizio-segida onena bukaerako egoerara heltzen denean, *beam-search* algoritmoak dependentzia-zuhaitza itzuliko du. *Beam-search* algoritmoak puntuazio funtzioa kalkulatzeko ezaugarri-bektore globala eta ikasitako parametro bektorea erabiliko ditu une oro. Alde batetik, ezaugarri-bektore globalak esaldi-egitura ezaugarri-bektore global batera bideratu ostean, ezaugarri-txantiloi batean definitutako ezaugarri guztietatik zenbat bete diren adierazten du; eta beste aldetik, parametro bektorea ikasteko, ikasketa automatiko gainbegiratua egiten duen *Averaged Perceptron* sailkatzailea erabiltzen du. *Averaged Perceptron* sailkatzaileak entrenamenduko adibideak ikasteko hainbat iterazio erabiltzeko aukera dauka. Entrenamenduko iterazio kopuru egokia zein den erabakitzeko, 1etik 25era bitarteko iterazio egin ostean, lantzeko multzoarekin lortutako emaitzarik onena erabili da, azkeneko test multzoarekin probatzeko. Zpar sistemako ezaugarri multzoari ezaugarri berri bi baino ez dizkiogu gehitu, pila eta *buffer*eko buruan dauden hitzen klase semantikoak. Zpar sistemarekin *Penn2Malt* bihurtzailearekin lortutako arkuak proiektiboak direnez, ZPar zuzenean erabili da, eta *LTH* bihurtzailearekin sortutako arku ez-proiektiboak proiektibo bihurtzeko, transformazio pseudo-proiektiboarekin konbinatu da.

5.3.4 Bozketa bidezko konbinaketa: MaltBlender

Aurreko ataletan erabilitako klase semantiko ezberdinak eta hauek probatzeko erabilitako analizatzaile ezberdinak aurkeztu dira. Analisia egiteko garaian, era independentean entrenatutako modeloak konbinatzeko MaltBlender (ikusi 2.3.3.2 atala) bozketa bidezko konbinaketa tresna erabili da. MaltBlender tresnarekin pisurik gabeko eta pisu-eskema ezberdinekin probatu ostean, etiketaren zuzentasunaren pisu-eskema erabiltzea erabaki da. Modelo aniztasunaren mesedetan analizatzaile modelo ezberdinekin konbinaketa ezberdinak egin dira:

- Oinarrizko sistemak: multzo honetan, MaltParser, MSTParser eta Zpar sistemekin lortutako oinarriak konbinatu dira.
- Sistema hedatuak: oinarrizko algoritmoei mota ezberdinetako klase semantikoak (*synset*ak, fitxategi semantikoak eta hitz-multzoak) gehitu

ostean, sortutako sistemak oinarrizko sistemekin konbinatu dira.

5.4 Esperimentuak eta emaitzak

Tesi-lan honetan, (Agirre *et al.*, 2011) eta (Bengoetxea *et al.*, 2014) lanetan egin diren esperimentuak aurkeztuko dira. 5.4.1 atalean, *WordNet*eko klase semantikoak eta hitz-multzoak probatzeko, hurrengo baliabideak erabili dira:

- *SemCor/PTB* ebaketa eta *PTB* osoko urre-patroiko kategoriak
- dependentzietan oinarritutako MaltParser sistema
- zuhaitz-bankuak osagai-eredutik dependentzia-eredu bihurtzeko *LTH* bihurtzailea

Eta baliabide hauekin emaitza onak lortu ostean, 5.4.2 atalean, informazio semantikoak analisian duen eragina orokortzeko, hainbat baliabide berri erabili dira:

- *PTB* osoko urre-patroiko kategoriak erabili beharrean etiketatzaile batekin lortutako kategoria automatikoak
- izaera desberdineko 3 sistema (Malt, Zpar eta MST)
- osagai-eredutik dependentzia-eredu bihurtzeko 2 bihurtzaile desberdin (*LTH* edo *Penn2Malt*)
- oinarrizko sistemen eta informazio semantikoekin entrenatutako sistemen irteerak bozketa bidezko MaltBlender tresna

5.4.1 Lehen proba. Informazio semantikoa MaltParser sistemarekin

Lehenengo proba honetan, zuhaitz-banku ezberdin bi erabili dira: alde batetik, SemCor/*PTB* ebaketa eta beste aldetik, *PTB* osoa. SemCor/*PTB* ebaketarekin eta *PTB* osoarekin egindako esperimentuak, hurrengo baldintza komun hauetan egin dira:

- Zuhaitz-bankuak osagai-eredutik dependentzia-eredu bihurtzeko *LTH* bihurtzailea erabili da.
- MaltParser sistema erabili da.
- Urre-patroiko kategoriak erabili dira.
- Esanahirik sarrienaren metodoa: hitz kategoria bati klase semantikoa gehitzeko orduan, *SemCor*en gehien agertzen den adiera gehitu da. Aditzak soilik infinitiboan agertzen zirela-eta, testuan agertzen ziren

aditzaren forma erregularrak eta irregularrak ere tratatu ahal izateko kategoria etiketatzailea eta lematizatzailea den TreeTagger erabili da.
- Sistemak neurtzeko LAS erabili da, eta egindako antzeko lanekin alderatzeko, puntuazio-marka ez da kontuan hartu.

Baina, badaude zuhaitz-bankuarekiko propioak diren baldintzak. Alda batetik, SemCor/*PTB* ebaketarekin egindako esperimentuak hurrengo baldintza berezietan egin dira:
- Hitzen adiera-desanbiguaziorako (HAD) 3 iturri ezberdin erabili dira: urre-patroiko informazioa semantikoa, esanahirik sarrienaren metodoa eta *ASR* desanbiguatze metodoa
- Klase semantikoak bakarka eta euren arteko ehunka konbinaketa probatu dira.
- *Cross-validation* teknika eta azkeneko proba egiteko, ebaluazio multzoa erabili da.

Beste aldatik, *PTB* osoarekin egindako esperimentuak hurrengo baldintza berezietan egin dira:
- Informazioa semantikoa lortzeko esanahirik sarrienaren metodoa soilik erabili da, bestelako informaziorik ez genuelako.
- Klase semantikoak bakarka eta *SemCor/PTB*tik lortutako konbinaketa onenak probatu dira.
- Antzerako lanekin alderatzeko (Yamada eta Matsumoto, 2003; McDonald *et al.*, 2005; McDonald eta Pereira, 2006), entrenatzeko *Wall Street Journal*ko 2-21 sekzioak, lantzeko 22. sekzioa eta azkeneko proba egiteko 23. sekzioa erabili dira.

SemCor/PTB ebaketaren hurrengo emaitzak aipatuko dira:
- **Esanahirik sarrienaren metodoarekin**, ezaugarri bakun onena *GFS*-arekin 81,59 lortu da LAS neurrian, +0,49ko hobekuntza esanguratsuarekin eta klase semantikoekin ehunka konbinaketa egin ostean, emaitzarik onena $GFS + ASS + HIFS$ konbinaketarekin 81,92 lortu da LAS neurrian, +0,81 hobekuntza esanguratsuarekin.
- **Urre-patroiko metodoarekin**, ezaugarri bakun onena *GFS*arekin 82,05 lortu da LAS neurrian, +0,95 hobekuntza esanguratsuarekin. Beste bakarkako batzuk aipatzearren, $HGFS$ (+0,46), AFS (+0,42) eta ASS (+0,38) hobekuntzak aipa daitezke. Eta konbinaketa onena $GFS + ASS + HIFS$ konbinaketarekin 81,92 lortu da LAS neurrian, estatistikoki esanguratsua den +0,81eko hobekuntzarekin.
- ***ASR* metodoarekin**, ezaugarri bakun onena $HAFS$arekin 81,72 lortu da LAS neurrian, estatistikoki esanguratsua den +0,62ko hobekuntza-

HAD	**klase semantikoa**		***SemCor/PTB***	***PTB* osoa**
Oinarria			81.10	86,27
Esa.	*FS*	Guztiak	81,59 (+0,49)†	**86,63 (+0,36)**‡
		Izenak	81,38 (+0,28)	86,56 (+0,29)†
		Aditzak	81,52 (+0,42)†	86,34 (+0,07)
	SS	Guztiak	81,40 (+0,30)	86,53 (+0,26)†
		Izenak	81,39 (+0,29)	86,33 (+0,06)
		Aditzak	81,48 (+0,38)†	86,48 (+0,21)†
	HFS	Guztiak	81,57 (+0,46)†	86,50 (+0,23)†
		Izenak	81,40 (+0,30)	86,25 (-0,02)†
		Aditzak	81,42 (+0,32)	86,51 (+0,24)†
	Konb.	*GFS + ASS + HIFS*	**81,92 (+0,81)**‡	**86,60 (+0,33)**†
Urre.	*FS*	Guztiak	**82,05 (+0,95)**‡	
		Izenak	81,51 (+0,41)	
		Aditzak	81,51 (+0,41)	
	SS	Guztiak	81,18 (+0,08)	
		Izenak	81,40 (+0,30)	
		Aditzak	81,58 (+0,48)†	
	HFS	Guztiak	81,51 (+0,40)	
		Izenak	81,43 (+0,33)	
		Aditzak	81,51 (+0,41)†	
	Konb.	*GFS + IFS + AFS + GSS + HIFS*	**81,74 (+0,63)**†	
ASR	*FS*	Guztiak	81 (-0,10)	
		Izenak	80,97 (-0,13)	
		Aditzak	**81,66 (+0,56)**‡	
	SS	Guztiak	81,30 (+0,19)	
		Izenak	81,56 (+0,46)	
		Aditzak	81,49 (+0,39)	
	HFS	Guztiak	81,32 (+0,22)	
		Izenak	81,62 (+0,52)†	
		Aditzak	**81,72 (+0,62)**‡	
	Konb.	*AFS + ASS*	81,41 (+0,31)	

5.2 taula – MaltParser sistemarekin *SemCor/PTB* ebaketan eta *PTB* osoko test multzoarekin lortutako emaitzak (LAS) desanbiguatze-metodo desberdinekin (Urre. = Urre-patroi semantikoa, Esa. = Esanahirik sarriena).

rekin. Beste bakarkako batzuk aipatzearren, AFS (+0,56) eta $HIFS$ (+0,52) aipatuko dira. Eta konbinaketa onena $AFS + ASS$ konbinaketarekin 81,41 lortu da LAS neurrian, estatistikoki esanguratsua den +0,63ko hobekuntzarekin.

- **Bakun onena vs txarrena**, urre-patroiko informazio semantikoarekin eta GFS (+0,95) ezaugarri semantiko bakunarekin, emaitzarik onena lortu da. Aldiz, ASR metodoarekin GFS eta IFS klase semantikoekin -0,10 eta -0,13 balioak lortu dira, hurrenez hurren.
- **Konbinaketak**, ASR metodoarekin eta urre-patroiko informazio semantikoarekin ez dira ezaugarri bakunen emaitzak hobetu. Baina, guztietan hobekuntza esanguratsuak lortu dira. Adibidez: esanahirik sarriena teknikarekin eta $GFS + ASS + HIFS$ konbinaketarekin 81,92 (+0,81), eta urre-patroiko informazio semantikoarekin eta $GFS + IFS + AFS + GSS + HIFS$ konbinaketarekin 81,74 (+0,63) lortu dira, LAS neurrian.

PTB osoarekin esanahirik sarrienaren metodoa baino ez da erabili beste metodoen informazio faltagatik, eta **esanahirik sarrienaren metodoarekin** lortutako hurrengo emaitzak aipatuko dira:

- $HIFS$ ezaugarriaren emaitzan izan ezik, gainerako emaitzetan oinarria gainditzen da.
- Emaitzarik onena 86,63 (+0,36) izan da, GFS klase semantikoarekin.
- FS, SS eta HFS multzo semantiko bakoitzetik, GFS (+0,36), $HAFS$ (+0,24) eta GSS (+0,26) klase semantikoekin lortu dira hobekuntzarik onenak.
- Lehenengo aldia da LTH bihurtzailearekin PTB osoa dependentzia bihurtu ostean eta *WordNet*etik esanahirik sarrienaren metodoarekin[6] lortutako klase semantikoak gehitu ostean, emaitza estatistikoki esanguratsuak lortzen direla.

5.4.2 Bigarren proba. Kategoria automatikoekin lortutako klase semantikoak, Malt, MST eta ZPar sistemekin

Informazio semantikoak (Brown-en hitz-multzoak eta *WordNet*etik lortutako klase semantikoak) analizatzaile sintaktikoetan eduki dezakeen eragina bost

[6] Esanahirik sarrienaren metodoak kategoria zehaztea behar du, eta zeregin horretarako urre-patroiko kategoria erabili da.

ikuspuntu ezberdinetatik aztertu da:

- **Hitzen adiera-desanbiguaziorako (HAD) metodoa**: *PTB* zuhaitz-bankuko hitz-kategoria bakoitzari dagokion *WordNet* 1.7 bertsioko klase semantikoa esleitzeko orduan, hitz batek esanahi ugari izan ditzakeela kontuan izan behar da. Anbiguotasun hori gainditzeko, esanahirik sarrienaren metodoa erabili da.
- **Izaera desberdineko sistemak**: Dependentzietan oinarritutako izaera desberdineko *Malt*, *Zpar* eta *MST* sistemekin probatu da.
- **Bihurtzaileak**: *PTB* zuhaitz-bankua osagai-eredutik dependentzia-eredu bihurtzerakoan, bi bihurtzaile desberdinekin probatu dira. *LTH* bihurtzailearekin, dependentzia-etiketa aberatsagoak eta finagoak lortzen dira, eta *Penn2Malt* bihurtzailearekin, berriz, dependentzia-etiketa multzo murriztua lortzen da. *LTH* bihurketan arku ez-proiektiboak sortzen direnez eta *ZPar*ek arku proiektiboekin lan egiten duenez, pseudo-proiektibo algoritmoarekin konbinatu da.
- **Kategoria-etiketatzailearekin lortutako kategoria automatikoak edo urre-patroiko kategoriak**: Kategoria automatikoak lortzeko *Perceptron POS tagger* (Collins, 2002) kategoria etiketatzailea erabili da. *Perceptron POS tagger* entrenatzeko, *Wall Street Journal*eko 2-21 sekzioak erabili dira entrenamenduko multzoari eta proba multzoari modu automatikoan esleitzeko kategoria. *Perceptron POS tagger*aren zehaztasuna proba multzoarekin % 97,32koa da.
- **Sistemen konbinaketa**: Azkenik, analisi garaian, klase semantiko ezberdinekin entrenatutako modeloak konbinatzeko pisu-eskema ugari dituen MaltBlender (ikusi 2.3.3.2 atala) bozketa bidezko konbinaketa tresna erabili da, eta esperimentu guztietarako soilik pisu-eskema sinpleena den etiketaren zuzentasunaren pisu-eskema erabili da.

Esperimentu guztiak *PTB* osoarekin egin dira. Esperimentu guztietan *Wall Street Journal*eko sekzio hauek erabili dira: entrenatzeko 2-21 sekzioak, lantzeko 22.a eta azkeneko proba egiteko 23.a. Emaitzak estatistikoki esanguratsuak diren jakiteko, Bikelen ausazko analizatzaile sintaktikoen ebaluazioen konparadorea erabili da.

5.4.3 Informazio semantikoa banaka

Lehenengo eta behin, informazio semantikoa banaka probatu da. Alde batetik, urre-patroiko kategoria edo kategoria automatikoekin lortutako *synset*ak eta fitxategi semantikoak probatu dira, eta, bestetik, era automatikoan lortu-

tako *Brown*en hitz-multzoen hierarkia probatu da. 5.3 eta 5.4 tauletan, *LTH* eta *Penn2Malt* bihurtzaileekin *WordNet*etik ateratako klase semantikoak banaka probatu ostean, SS eta FS azpimultzo bakoitzeko onena agertzen da. Eta *Brown*en hitz-multzoen hierarkiatik, MaltParser eta MSTParser sistemekin bit tarte ugari probatu ostean lortutako emaitzarik onena agertzen da; Zpar sistemarekin, aldiz, bit guztiak erabiliz baino ez da probatu.

Kategoria		**Oinarria**	**FS**	**SS**	**Hitz-multzoak**
Urre-patroikoak	Malt	89,74	89,87 (+0,13)	89,86 (+0,12)	89,84 (+0,10)
	MST	91,82	92,03 (+0,21)	91,99 (+0,17)	**92,20 (+0,38)**†
	ZPar	92,83	92,91 (+0,08)	92,80 (-0,03)	92,91 (+0,08)
Automatikoak	Malt	88,46	88,49 (+0,03)	88,42 (-0,04)	88,59 (+0,13)
	MST	90,55	90,70 (+0,15)	90,47 (-0,08)	**90,88 (+0,33)**‡
	ZPar	91,52	91,65(+0,13)	**91,70 (+0,18)**†	91,74 (+0,22)

5.3 taula – *Penn2Malt* bihurtzailearekin eta banakako klase semantikoekin lortutako emaitzak LAS neurrian.

Kategoria		**Oinarria**	**FS**	**SS**	**Hitz-multzoak**
Urre-patroikoak	Malt	86,27	**86,63 (+0,36)**‡	**86,53 (+0,26)**†	**86,61 (+0,34)**‡
	MST	86,86	86,85 (-0,01)	86,64 (-0,22)	**87,64 (+0,78)**‡
	ZPar	90,48	90,54 (+0,06)	90,53 (+0,05)	90,59 (+0,11)
Automatikoak	Malt	84,95	85,12 (+0,17)	85,08 (+0,16)	85,13 (+0,18)
	MST	85,06	**85,35 (+0,29)**‡	84,99 (-0,07)	**86,18 (+1,12)**‡
	ZPar	89,15	89,33 (+0,18)	89,19 (+0,04)	89,17 (+0,02)

5.4 taula – *LTH* bihurtzailearekin eta banakako klase semantikoekin lortutako emaitzak LAS neurrian .

- ***Penn2Malt* bihurtzailearekin**, MST sisteman hitz-multzoak erabilita, 92,20 (+0,38) eta 90,88 (+0,33) emaitza esanguratsuak lortu dira, urre-patroiko kategoria eta kategoria automatikoekin, hurrenez hurren; eta, ZPar sisteman 91,70 (+0,18) eta 91,74 (+0,22) emaitzak lortu dira *synset*ak eta hitz-multzoak erabilita, hurrenez hurren, ezaugarri automatikoekin (ikusi 5.3 taula).
- ***LTH* bihurtzailearekin**, MST sisteman hitz-multzoak erabilita, 87,64 (+0,78) eta 86,18 (+1,12) emaitza esanguratsuak lortu dira, urre-patroiko kategoria eta kategoria automatikoekin, hurrenez hurren (ikusi 5.4 taula).
- ***LTH* eta *Penn2Malt* alderatuta**, espero zen bezala, klase semantikoekin lortutako hobekuntzak handiagoak izan dira *LTH* bihurtzailearekin.

- **Kategoria automatikoak eta urre-patroiko kategoriak alderatuta**, kategoria automatikoak erabiltzerakoan, soilik MST sisteman FS eta hitz-multzoekin hobekuntza estatistikoki esanguratsuak lortu dira. Urre-patroiko kategoriak erabiltzerakoan, berriz, Malt sisteman FS, SS eta hitz-multzoekin eta MST sisteman hitz-multzoekin hobekuntza estatistikoki esanguratsuak lortu dira.
- **Izaera desberdineko sistemak alderatuta**, ZPar sistema MST eta Malt sistemak baino zehatzagoa da, 4 puntu balio absolutuan.
- **Hobekuntzarik onena**: *LTH* bihurtzailea, kategoria automatikoak, MST sistema eta hitz-multzoak erabilita, +1,12ko hobekuntza esanguratsua lortu da.

5.4.4 Konbinaketak

Oinarrizko sistemak (informazio semantikorik ez daukatenak) eta sistema hedatuak (informazio semantikoa daukatenak) aurkeztu dira 5.4.3 atalean. Atal honetan, oinarrizko sistemen eta sistema hedatuen irteerak konbinatzeak zehaztasunean izan dezakeen eragina aztertuko da. Horrela, analizatzaile sintaktikoen irteerak konbinatzeko MaltBlender tresna erabiliko da. MaltBlender tresnarekin pisurik gabeko eta pisu-eskema ezberdinekin probatu ostean, dependentzia-etiketaren zuzentasunaren pisu-eskema erabiltzea erabaki da. Modelo aniztasunaren mesedetan, analizatzaile modelo ezberdinekin konbinaketa ugari egin dira:

- Oinarrizko sistemak: multzo honetan Malt, MST eta Zpar sistemekin lortutako oinarriak konbinatu dira.
- Sistema hedatuak: Surdeanu eta Manning-en (2010) lanean probatu zuten sistema desberdinen aniztasuna aberasgarria zela konbinaketan. Informazio semantiko desberdinekin sortutako modelo hedatuak oinarrizko sistemekin konbinatzeko orduan ehunka konbinaketa ez egiteko, hurrengo konbinaketak erabili dira:
  - Malt, MST eta ZPar sistema bakoitzeko, oinarrizko eta FS taldeko (GFS, IFS eta AFS) sistema hedatu onena aukeratu ostean konbinatu dira (guztira 6).
  - Malt, MST eta ZPar sistema bakoitzeko, oinarrizko eta SS taldeko (GSS, ISS eta ASS) sistema hedatu onena aukeratu ostean konbinatu dira (guztira 6).
  - Malt, MST eta ZPar sistema bakoitzeko, oinarrizko eta hitz-multzoko sistema hedatuak konbinatu dira (guztira 6).

Horrela, 3 oinarrizkoen (Malt + ZPar + MST) konbinaketa onenarekin alderatzeko aukera egongo da.

Analizatzaileak	**LAS**	**UAS**
Oinarrizko onena (ZPar)	91,52	92,57
Hedatu onena (ZPar + Hitz-multzoak)	91,74 (+0,22)	92,63
3 oinarrizkoen konbinaketa onena	91,90 (+0,38)	93,01
Hiruko konbinaketa onena: 3 oinarrizko + 3 FS hedatu	91,93 (+0,41)	92,95
Hiruko konbinaketa onena: 3 oinarrizko + 3 SS hedatu	91,87 (+0,35)	92,92
Hiruko konbinaketa onena: 3 oinarrizko + 3 hitz-multzo hedatu	91,90 (+0,38)	92,90

5.5 taula – *Penn2Malt* bihurtzailearekin eta ezaugarri automatikoekin egindako konbinaketak.

Analizatzaileak	**LAS**	**UAS**
Oinarrizko onena (ZPar)	89,15	91,81
Hedatu onena (ZPar + FS)	89,33 (+0,18)	92,01
3 oinarrizkoen konbinaketa onena	89,15 (+0,00)	91,81
Hiruko konbinaketa onena: 3 oinarrizko + 3 FS hedatu	89,56 (+0,41)‡	92,23
Hiruko konbinaketa onena: 3 oinarrizko + 3 SS hedatu	89.43 (+0,28)	93,12
Hiruko konbinaketa onena: 3 oinarrizko + 3 hitz-multzo hedatu	89,52 (+0,37)†	92,19

5.6 taula – *LTH* bihurtzailearekin eta ezaugarri automatikoekin egindako konbinaketak .

Beste alde batetik, 5.5 eta 5.6 tauletan agertzen diren emaitzak, lantzeko datuekin lortutako onenak, test multzoarekin ebaluatu ostean lortutakoak dira. 5.5 taulan, *Penn2Malt* bihurtzailearekin 3 oinarrizko sistemen konbinaketarik onenak oinarrizko sistema onenak baino 0,38 puntu gehiago lortzen du edozein sistema hedaturekin egindako konbinaketa onenaren parean. Horrela, esan daiteke *Penn2Malt* bihurtzailearekin, klase semantikoa konbinaketan gehitzeak ez duela apenas eraginik analisian. 5.6 taulan, *LTH* bihurtzailearekin 3 oinarrizko sistemen konbinaketarik onenak ez du oinarrizko sistemarik onena hobetu. Hitz-multzoekin eta FS sistema hedatuekin egindako konbinaketetan, aldiz, emaitza estatistikoki esanguratsuak lortu dira, +0,37 eta +0,41, hurrenez hurren.

5.4.5 Analisia

Atal honetan, informazio semantikoa analizatzaile sintaktikoetan erabiltzeak non eta nola laguntzen duen aztertzeko datuen analisi sakonagoa egin da. Analizatzaile automatikoek duten arazoetako bat kategoria-etiketatzaileak hitz baten kategoria ez-zuzena ematerakoan gertatzen da. Adibidez, kategoria automatikoak erabiltzerakoan, ZPar sistema % 90,45etik % 89,12ra jaisten da. Informazio semantiko ezberdinek kategoria zuzenekin eta kategoria ez-zuzenekin duten eragina probatzeko, test multzoa bi azpimultzotan banatu da: kategoria erroreak dituen azpimultzoa (1.025 esaldi eta 27.300 hitzekoa) eta kategoria guztiak zuzenak dituen azpimultzoa (1.391 esaldi eta 29.386 hitzekoa).

Kat.	**Analizatzaileak**	**testa (LAS)**	**kat. zuzenen esaldiak (LAS)**	**kat. erroredun esaldiak (LAS)**
Urre	ZPar	90,45	91,68	89,14
Auto	ZPar	89,15	91,62	86,51
Auto	Hiruko konbinaketa onena: 3 oinarrizko + 3 FS	89,56 (+0,41)	91,90 (+0,28)	87,06 (+0,55)
Auto	Hiruko konbinaketa onena: 3 oinarrizko + 3 SS	89,43 (+0,28)	91,95 (+0,33)	86,75 (+0,24)
Auto	Hiruko konbinaketa onena: 3 oinarrizko + 3 hitz-multzo	89,52 (+0,37)	91,92 (+0,30)	86,96 (+0,45)

5.7 taula – Kategoria zuzenak eta ez-zuzenak dituzten azpimultzoen emaitzak, *LTH* bihurtzailearekin (parentesietan agertzen diren hobekuntzak, kategoria automatikoekin entrenatutako ZPar sistemarekiko dira).

Adierazitako 5.7 taulan agertzen diren esperimentu guztiak *LTH* bihurtzailearekin egin dira, *LTH* bihurtzailearekin lortu baitira hobekuntza handienak informazio semantikoa erabiltzerakoan. 5.7 taularen lehenengo bi lerroetan, ZPar oinarrizko sistemarekin, urre-patroiko kategoriekin eta kategoria automatikoekin lortutako emaitzak agertzen dira. Hurrengo hiru lerroetan kategoria automatikoekin lortutako oinarrizko eta informazio semantikoarekin aberastutako irteeren konbinaketak agertzen dira. 5.6 taulan ikusten da *LTH* bihurtzailearekin 3 oinarrizko sistemen konbinaketa onenak ez duela oinarrizko sistema onena hobetu, eta, hartara, sistema hedatuekin egindako konbinaketetan dauden irabaziak informazio semantikoa gehitu ostean gertatzen direla suposa daiteke. 5.7 taularen hirugarren lerroan 3 oinarrizko eta 3 *FS* sistemekin kategoria akatsak dituzten esaldien eta kategoria akats gabeko esaldien artean desoreka dagoela ikus daiteke. Horrela, akatsen azpimultzoan 0,55 hobetzen dela, eta, akats gabeko esaldietan, aldiz,

0,28 hobetzen dela ikus daiteke. Horregatik, *FS*etik jasotzen den informazioaren zati handi bat kategoria etiketatzaile automatikotik datozen akatsak arintzeko erabiltzen dela esan daiteke. Bestalde, *SS*etik eta hitz-multzotik datozen hobekuntzak kategoria akatsak dituzten esaldien eta akats gabeko esaldien artean orekatuta daude; izan ere, gerta daiteke *SS* eta hitz-multzoetatik kategoriarekiko ortogonalak izatea, eta, horregatik, akats eta akats gabeko azpimultzoetan jokaera berdintsua izatea. Horixe bera egiaztatzeko, analizatzaile bakunak era independentean probatu dira. Adibidez, MST sistemarekin eta *FS*ekin kategoria akatsekin irabaziak ia bikoiztu (+0,21etik +0,37ra) egiten dira, eta hitz-multzoen informazioa gehitu ostean irabaziak antzekoak direla ikus daiteke (+0,91 eta +1,33, hurrenez hurren). Lortutako irabaziak klase semantiko motarekin eta analizatzaile algoritmoarekin zerikusia dutela esan daitekeen arren, lan hau hasierako miaketa besterik ez da.

5.5 Ondorioak

Tesi-lan honetan, *WordNet*eko *synset*ak eta fitxategi semantikoak, eta, etiketatu gabeko corpus batetik lortutako hitz-multzoak dependentzietan oinarritutako analisi sintaktikoan gehitzeak dakartzan hobekuntzak probatu ditugu. Esperimentu asko egin dira *PTB* zuhaitz-bankua osagai-eredutik dependentzia-eredu bihurtzeko; bi bihurketa erabili dira eta kategoria etiketatzailearekin era automatikoan lortutako informazioa eta izaera desberdineko hiru analizatzaile erabili dira. (Agirre *et al.*, 2011) lanean, lehenengoz, *FS* fitxategi semantikoekin PTB osoan hobekuntza esanguratsuak lortu genituen, *LTH* bihurtzailearekin, MaltParser sistema, urre-patroiko kategoria eta esanahirik sarrienaren metodoa erabiliz. Baina kategoria etiketatzailearekin era automatikoan lortutako kategoriak erabiltzerakoan lortutako hobekuntza gehienak xumeak izan dira. Lortutako emaitzekin, sarreran egindako galderak erantzuten ahaleginduko gara:

– **Informazio semantikoa analizatzaile sintaktikoa hobetzeko baliagarria da? Benetako egoera batean baliagarriak dira?** Urre-patroiko kategoriak soilik erabili beharrean, kategoria etiketatzaile batek (ingelesez, *POS tagger*) iragarritako kategoria automatikoekin probatu da. Kategoria etiketatzaile batekin lortutako kategoria automatikoekin probatu ostean, hurrengo hobekuntza esanguratsuak lortu dira: MST sisteman hitz-multzoekin eta *LTH* eta *Penn2Malt* bihurkete-

kin, +1,12 eta +0,33, hurrenez hurren; ZPar sisteman *synset*ekin eta *Penn2Malt* bihurketekin +0,18 eta MST sisteman fitxategi semantikoekin eta *LTH* bihurketekin +0,29. Informazio semantikoa erabiltzeak dependentzia-ereduko analisi sintaktikoaren zehaztasuna hobetzen du, baina kategoria automatikoekin lan egiterakoan, kasu gehienetan oso txikiak eta ez-esanguratsuak izan dira.

- **Zein mailatako informazio semantikoa erabiltzea komeni da?** Eskuragai izan ditugun *WordNet*eko hitz eta adierei buruzko informaziotik adiera zabalena eta adiera zorrotzena duena, eta etiketatu gabeko corpusetik lortutako hitz-multzoetatik maila desberdinetako hitz-multzoak probatu ditugu. Ia emaitza guztiak positiboak izan dira. Kategoria automatikoak erabiltzerakoan, MSTParser sistemarekin, fitxategi semantikoekin eta hitz-multzoekin emaitza esanguratsuak lortu dira. Baina, oro har, kategoria automatikoak erabiltzerakoan, fitxategi semantikoekin eta *synset*ekin lortutako hobekuntzak ez dira esanguratsuak izan.
- **Maila desberdineko informazio semantikoak izaera desberdineko sistemetan eragin berdina izango ote du?** Izaera desberdineko hiru sistemekin lan egin da: trantsizioetan oinarritutako analizatzaile determinista (MaltParser), grafoetan oinarritutako analizatzaile globala (MSTParser) eta trantsizioetan oinarritutako ZPar, hau da, bilaketa lokala, determinista eta jalea erabili beharrean, ikasketa diskriminatzaile globala eta *beam* bilaketa erabiltzen duena. Oro har, ia emaitza guztiak positiboak dira. Baina hitz-multzoekin eta fitxategi semantikoekin, MSTParser sistemarekin baino ez dira emaitza esanguratsuak lortu.
- **Zein bihurketa tresna da egokiena informazio semantikoarekin lan egiterakoan?** *Penn Treebank*a osagai-eredutik dependentzia-eredura bihurtzeko bihurtzaile anitz daude. Oro har, bihurtzaile mota bi bereizten dira: dependentzia-etiketa aberatsagoak eta finagoak ematen dituztenak, adibidez, *Pennconverter* eta dependentzia-etiketen multzo murriztua ematen dutenak, adibidez, *Penn2Malt*. Galdera honi erantzuteko, *Penn2Malt* eta *Pennconverter* erabili dira. Oro har, informazio semantikoa erabili gabe, *Penn2Malt* bihurtzailearekin, *LTH* bihurtzailearekin baino emaitza hobeak plazaratu dira hiru sistemetan, bien arteko aldea, Malt (+3,51), MST (+5,49) eta ZPar (+2,37) izan dela. Baina, nahiz eta *LTH* bihurtzailearekin irteera baxuagoa lortu, semantikaren prozesamendurako onenetarikoa dela (Johansson eta Nu-

gues, 2007b) ikus daiteke, eta informazio semantikoa erabiltzerakoan aldeak gutxitu egin direla ikus daiteke. Emaitza batzuk aipatzearren: lehenengo aldia da, *LTH* bihurtzailearekin eta MaltParser sistemarekin, PTB osoan *WordNet*etik lortutako fitxategi semantikoekin emaitza estatistikoki esanguratsuak lortzen direla. Horrela, urre-patroiko kategoria erabilita eta esanahirik sarrienaren metodoa erabiliz, *PTB* osoan +0,36ko hobekuntza esanguratsua lortu genuen LAS neurrian. Baina kategoria etiketatzailearekin era automatikoan lortutako kategoriak erabiltzerakoan hobekuntza +0,17koa izan da. LTH bihurtzailearekin lortutako beste emaitza batzuk aipatzearren, hitz-multzoak eta fitxategi semantikoak erabiltzerakoan, MSTParser sistemarekin eta ezaugarri automatikoekin emaitza esanguratsuak lortu dira. *Penn2Malt* bihurtzailearekin bi emaitza azpimarratuko dira: MSTParser sistemarekin eta hitz-multzoak erabiltzerakoan lortutako +0,33ko hobekuntza esanguratsua, eta, ZPar eta *synset*ak erabilita +0,18 esangura txikiko hobekuntza.

– **Kategoria automatikoekin, konbinaketan informazio semantikoa erabiltzea aberasgarria da?** Oinarrizko sistemen eta sistema hedatuen (informazio semantikoekin entrenatutako sistemen) irteerak bozketa bidezko MaltBlender sistemarekin konbinatu dira. Sistema hedatuen irteerak konbinatzerakoan, bai *LTH* bihurtzailearekin, bai *Penn2Malt* bihurtzailearekin antzerako hobekuntzak lortu dira: fitxategi semantikoekin +0,41 bietan, *synset*ekin, +0,28 eta +0,35 eta hitz-multzoekin, +0,37 eta +0,38, hurrenez hurren. Sistema hedatuen irteerak konbinatzerakoan, *LTH* eta *Penn2Malt* bihurtzailearekin antzerako hobekuntza ez esanguratsuak lortu dira.

6

Ondorioak eta etorkizuneko lanak

IXA taldean, sintaxi konputazionalean, analisi sintaktiko partzialetik osorako bidean, hizkuntza-ezagutzan oinarritutako eta Murriztapen Gramatikan oinarritzen den Euskararako Dependentzia Gramatika Konputazionala (EDGK) (Aranzabe, 2008) sortu da, eta gramatika hori baliatuta garatu da analizatzaile sintaktiko osoa. Tesi-lan honek analisi sintaktiko partzialetik osorako bidean aukera berriak zabaltzen ditu. Helburu nagusia estaldura zabala izango duen dependentzietan oinarritutako analizatzaile sintaktiko estatistiko sendoak lortu eta analizatzaile sintaktikoaren zehaztasuna hobetzeko teknika ezberdinak erabiltzea izan da. Bidean, hurrengo galdera nagusiak erantzuten ahalegindu gara:

- Analizatzaileen sortzaileen egokitzapenak zenbateko eragina du analizatzailearen zehaztasunean?
- Euskara bezalako hizkuntza eranskarietan ezaugarri morfosintaktikoek zenbateko eragina dute analizatzailearen zehaztasunean?
- Ezaugarri morfosintaktiko automatikoak erabiltzeak zenbateko eragina du analizatzailearen zehaztasunean?
- Corpuseko tamainak badu eraginik analizatzailearen zehaztasunean?
- Analizatzailearen zehaztasuna hobetu ahal da zuhaitz-bankuan egindako zuhaitz-transformazioekin?
- Analizatzailearen zehaztasuna hobetu ahal da analizatzaileen pilaketa eta bozketa bidezko konbinaketarekin?
- Analizatzailearen zehaztasuna hobetu ahal da informazio semantikoa zuhaitz-bankuetan gehituz gero?

Hurrengo ataletan, lortutako emaitzekin atera diren ondorio nagusiak eta amaitu barik utzi diren etorkizuneko lanak zeintzuk izango diren azaldu dira.

6.1 Ondorio nagusiak

Atal honetan, aurreko atalean egindako galdera nagusi bakoitzetik atera diren ondorioak plazaratuko dira.

6.1.1 Analizatzaileen sortzaileen egokitzapena

Tesi-lan honek balio izan du euskarazko corpusak *CoNLL-X* formatura egokitzeko, dependentzietan oinarritutako analizatzaileen sortzaileek ikasteko behar duten euskarazko zuhaitz-bankua izateko. 2007an, lehen euskarazko corpusa (Aduriz *et al.*, 2006a) *CoNLL-X* formatura egokitu zen EZB I zuhaitz-bankua sortuz, dependentzietan oinarritutako analizatzaile sintaktikoen sortzaileek erabil zezaten, eta *CoNLL* 2007 *Multilingual Dependency Parsing* lehiaketan, beste 9 hizkuntzekin batera, 20 sistemen artean ebaluatu zezaten. 2010ean, dependentzia-ereduan etiketatuta dagoen euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe *et al.*, 2008) *CoNLL-X* formatura egokitu zen EZB II zuhaitz-bankua sortuz. Baita MaltParser eta MSTParser dependentzietan oinarritutako analizatzaileen sortzaileak ere euskarara egokitzeko balio izan du. MaltParser eta MSTParser instalatu ostean, hizkuntza ezberdinetan baliagarriak diren balioekin konfiguratuta datoz, baina ez datoz prestatuta euskara bezalako hizkuntza eranskari baterako. Nahiz eta orain optimizaziorako hainbat tresna agertzen ari diren (MaltOptimizer esate baterako), tresna hauek ere ez dute lan egiten espero bezain ondo, izan ere, hitzari eta kategoriari garrantzi handia ematen baitiote. Euskara hizkuntza eranskaria da, izan ere, testu-hitz baten lemak morfema asko har ditzake. Gainera, hitzari errekurtsiboki gehi diezazkiogu atzizkiak eta hitzak ezaugarri morfosintaktiko ugari hartzen ditu. Horrela, ezaugarri-modeloa definitzerakoan, lemak, hitzak baino protagonismo handiagoa hartuz gero, eta kasua eta mendeko perpausen informazioa FEATS zutabetik atera eta kategoriaren maila berdinean jarrita, emaitzak asko hobetzen direla probatu da. Adibidez, EZB I zuhaitz-bankuarekin eta MaltParser 0.4 balio lehenetsiekin, 65,08 lortzen da LAS neurrian, testean; aldiz, ezaugarri-modeloaren egokitzapenarekin 74,52ko oinarri sendoa lortu da, hau da, +9,44ko hobekuntza (ikusi 3.3.3 ataleko, 3.6 taulan); eta EZB II zuhaitz-bankuarekin eta

MSTParser balio lehenetsiekin, 72,63 lortzen da LAS neurrian testean; aldiz, ezaugarri-modeloaren egokitzapenarekin eta bigarren mailako algoritmo ez-proiektiboarekin 79,30eko oinarri sendoa lortu da, hau da, +6,67ko hobekuntza (ikusi 3.4 ataleko, 3.11 taulan). Azkenik, EZB II zuhaitz-bankuan eta MaltParser 1.4 bertsioarekin, ezaugarri-modeloaren egokitzapenarekin eta Nivre Arku Estandarra (LibLINEAR) + transformazio pseudo-proiektiboarekin 78,46ko oinarri sendoa lortu da (ikusi 3.3.3 ataleko, 3.6 taulan).

MaltParser eta MSTParser analizatzaileen sortzaileak euskarara egokitzeko bidea ez da erraza izan, baina egokitzapenak estaldura zabaleko hainbat analizatzaile sintaktiko sortzeko aukera eman du.

6.1.2 Ezaugarri morfosintaktikoen ikerketa

Euskara, morfologia aldetik, oso aberatsa da; hau da, hitz batek hainbat ezaugarri morfosintaktiko izaten ditu. Hori dela-eta, euskarak ezaugarri morfosintaktikoen multzotik ezaugarriak kentzeko, gehitzeko, konbinatzeko, eta bestelakoak egiteko aukera eskaintzen du. Ezaugarri morfosintaktikoak 26 klasetan multzokatu ostean, klase bakoitza bakarka edo klase ezaugarriak konbinatuta, MaltParser eta MSTParser sistemetan probatu dira, eta bidean hurrengo galderak erantzuten saiatu gara:

- **Zein da analizatzailearen zehaztasuna gehien igotzen duen konbinaketa murriztua? Ezaugarri morfosintaktiko guztietatik, zeintzuk dira analizatzailearen zehaztasunean ia eraginik ez dutenak, eta baztertu ahal direnak?** Kasua eta mendeko perpausen informazioa izeneko ezaugarriak osagarriak dira, bakarka ematen duten hobekuntza batzerakoan mantendu egiten baita, eta ezaugarri gabeko sistemarekin alderatuta, 8 eta ia 9 puntuko hobekuntza lortzen da bi sistemetan (ikusi 3.5 atalean). Kasua eta mendeko perpausen informazio konbinaketari gehitutako gainerako ezaugarriek analisian eragin txikiagoa dute.
- **Erabili ahal da konbinaketa murrizturen bat 26 ezaugarriak ordezkatzeko?** Lehenengo, 26 ezaugarriak bakarka, gero, bakarka lortutako ezaugarririk onena gainerako beste 25 ezaugarriekin konbinatu da. Binakako konbinaketarik onena lortu ostean, gainerako 24 ezaugarriekin konbinatu da. Modu horretan jokatu da harik eta lau ezaugarri onenak konbinatzera heldu arte. MaltParser sisteman, kasua, mendeko perpausen informazioa, numeroa eta mugatasunaren konbinaketa murriztuekin, eta MSTParser sisteman, kasua, mendeko perpausen in-

formazioa, numeroa eta nork marka konbinaketa murriztuekin 26 ezaugarri klaseekin bezainbeste lortu da.

– **Lortutako konbinaketa murriztuek izaera ezberdineko sistemetan antzeko portaera dute?** Galdera honi erantzuteko izaera desberdineko sistema bi erabili dira: batetik, trantsizioetan oinarritutako eta izaera lokala duen MaltParser sistema, eta, bestetik, grafoetan oinarritutako eta izaera globala duen MSTParser sistema. MaltParser eta MSTParser sistemekin probatu ostean, lortutako konbinaketa murriztuek izaera ezberdineko sistemetan antzeko portaera dutela esan daiteke. Gainera, konbinaketa murriztua erabiltzeak entrenamendu fasea azkartu eta optimizazio fasea erraztu eta hobetu egiten duela probatu da. Adibidez, soilik EZB II zuhaitz-bankuan lema eta hitza zutabeak trukatuta eta azpikategoriaren zutabean, kategoria, azpikategoria, kasua eta mendeko perpausen informazioa kateatuta, +3,10eko hobekuntza lortu da MaltOptimizer tresnarekin. Horrez gain, ezaugarri automatikoak lortzen direnean analizatzaile morfologikoaren desanbiguazioan zer-nolako ezaugarriekin lan egin behar den adieraziko digute.

6.1.3 Ezaugarri morfosintaktiko automatikoak

Euskara bezalako hizkuntza eranskariak oso anbiguoak dira, informazio morfosintaktiko guztia (kasu-marka, numeroa edo aspektua bezalakoak) kontuan hartuz gero, analizatzaile morfologikoaren anbiguotasuna 2,65ekoa da. Urre-patroiko ezaugarriak erabili beharrean, ezaugarri automatikoak erabiltzerakoan, analisian zenbateko galera gertatzen ote den erantzuten saiatu gara. Horretarako, zuhaitz-bankuko esaldiak, analisi morfologiko eta desanbiguazio moduluak ematen dituen aukeretatik lehenengo aukera (sarriena) hartuta, 2 puntuko galera gertatzen dela ikusi da, izaera desberdineko MSTParser eta MaltParser sistemekin. Jaitsiera handiagoa espero genuen analizatzaile morfologikoaren anbiguotasuna 2,65ekoa baita. Jaitsiera baxu honen arrazoietako bat izan daiteke, desanbiguatzaile morfologikoaren lan sendoa eta errore morfologiko batzuk gardenak izatea analizatzaile sintaktikoarentzat.

6.1.4 Zuhaitz-bankuen tamaina

CONLL 2007 Multilingual Dependency Parsing lehiaketan (Nivre *et al.*, 2007a) lanean agertzen ziren emaitzak ikusirik, euskara, arabiar eta greko bezalako hizkuntza eranskariek entrenatzeko zuhaitz-banku handiagoen beharra zuten,

hau da, EZB I zuhaitz-bankuaren tamaina (57.000 hitzekoa) handiago izan behar zela ikusi zen. Orain, aitzitik, EZB II zuhaitz-bankua dago (150.000 hitzekoa), baina egindako probetan garbi ikusten da hobekuntza gehiago lor daitekeela zuhaitz-bankuaren tamaina handituz gero. Sistema eta algoritmo guztiekin entrenamenduko esaldi kopurua bikoizterakoan, gutxienez 1,30eko irabaziak gertatzen direla ikusi da. Gutxiengo irabazia 4000 esalditik 9100 esaldira pasatzerakoan gertatu da.

6.1.5 Zuhaitz-transformazioak

(Collins, 2009) lanean zuhaitzen aurre-prozesaketa erabili zen analizatzailearen emaitzak hobetzeko helburuarekin. Euskararako zuhaitz-bankuaren etiketatze moduak eta euskara hizkuntza buru-azkena den ezaugarria kontuan izan dira euskarako zuhaitz-bankuan transformazio ugari egiteko (ikusi 4.1 atala). Horrela, analisia hobetzeko euskararako egokiagoak izan daitezkeen transformazio pseudo-proiektiboa, sintagmen transformazioa, mendeko perpausaren transformazioa eta koordinazio-transformazioak probatu dira.

- **Transformazio pseudo-proiektiboak:** EZB II zuhaitz-bankua, % 1,18 arku ez-proiektibo eta % 13,28 esaldi ez-proiektiboekin, zuhaitz-banku ez-proiektiboen artean koka dezakegu. Algoritmo ez-proiektiboek algoritmo proiektiboak baino emaitza hobeak ematen dituzte. Transformazio pseudo-proiektiboak analisian dependentzia-etiketa konplexuak gehitu eta analisia neketsuago egiten badu ere, transformazio pseudo-proiektiboarekin hobekuntza esanguratsua lortzen dela probatu da.
- **Sintagmen transformazioa:** EZB I zuhaitz-bankuan, izen kategoriako hitzen zehaztasuna % 66koa da, eta zuhaitz-bankuko kategoriarik ugariena dela jakinda, ia errore guztien % 50 dela esan daiteke. Errore honen zergatia hau dela uste da: izenak berak kasu-markarik ez duenez (beste hitz batean dago), eta dependentzia-etiketa kasu gramatikalaren informazioarekin ezartzen denez, izena aditzarekin lotzerakoan dependentzia-etiketa okerrekoarekin egiten da batzuetan. Ondorioz, hitzak morfemetan banatzeak izan dezakeen eragina aztertzera eraman gaitu. EZB I eta II zuhaitz-bankuak etiketatzerakoan sintagmen buru bezala esanahi semantikoa duen hitza jarri da. 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 di-

rela ikusirik, hitzaren azken morfema banatu ostean, sintagmen burua, buru semantikoa izan beharrean, sintagmen buru sintagmen azkeneko kokagunea hartzen duen hitzaren morfema printzipala izanez gero, analisian hobekuntza esanguratsuak lortzen direla probatu da. Sintagmen transformazioak arku ez-proiektiboak sortzen ditu eta transformazio pseudo-proiektiboarekin batera emaitzak hobetzen dira.

- **Mendeko perpausen transformazioa:** Oro har, euskaraz, mendeko perpausak aditz nagusi edo aditz laguntzaileari morfema bat gehituz sortzen dira. 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. EZB I eta II zuhaitz--bankuetan, mendeko perpausak aditz nagusiaren (buru semantikoaren) inguruan antolatuta daude. Baina gehienetan mendeko perpausen morfema (buru sintaktikoa) aditz laguntzaileari lotuta etortzen da; hau da, ez dator bat buru semantikoarekin. Horrela, sintagma bakunak eta mendeko perpausak hobeto bereizteko eta mendeko morfemaren informazio sintaktikoa bere burutik hurbilago egoteko, mendeko morfema bere aditz laguntzailetik banatu eta mendeko perpausaren buru jarri ostean, hobekuntza xumeak gertatu dira. Esan beharra dago, transformazio honek arku ez-proiektiboak sortzen dituela eta transformazio pseudo-proiektiboarekin batera emaitzak hobetzen direla.
- **Koordinazioaren transformazioa:** Hitzen artean erlazio bitarra argia denean, dependentzietan oinarritutako teoriek ondo baino hobeto lan egiten dute. Baina koordinazioaren kasuan, juntagailuaren eta juntagaien artean erlazio bitarrak sortzeko orduan, teoria edo estilo ezberdinak sortu dira. Koordinazio estilo ezberdinekin (Praga, Melcuk eta Mel'cuk Estiloaren Simetrikoarekin) probatu ostean, emaitzarik onenak Mel'cuk Estiloaren Simetrikoarekin lortu dira. Horrela, euskara buru-azkena izanik koordinazioaren buru, koordinazioaren azken osagaia jartzeak analizatzailearen lana zehatzagoa dela probatu du. Oro har, koordinazioa hizkuntzaren ezaugarrietara egokituz gero, analisian emaitza hobeak lortzen direla esan dezakegu.
- **Transformazioen konbinaketa eta tamaina:** Sintagmen transformazioa, mendeko perpausen transformazioa, koordinazio transformazioa eta transformazio pseudo-proiektiboa bata bestearen atzetik aplikatu behar dira. Konbinaketan transformazioen ordenak garrantzia du, izan ere, transformazio pseudo-proiektiboa sintagmen transforma-

zioa eta mendeko perpausen transformazioaren ostean joan behar da, sintagmen eta mendeko perpausen transformazioek hainbat arku ez-proiektibo sortzen baitute. Esperimentu guztietan, transformazioen konbinaketa bateragarria dela probatu da, emaitza esanguratsuak lortu baitira. Algoritmo gehienetan, transformazioen konbinaketaren eragina zuhaitz-bankua bikoiztearen heinekoa dela esan daiteke.

EZB zuhaitz-bankuei egindako zuhaitz-transformazioekin hobekuntza esanguratsuak lortu dira bai urre-patroiko ezaugarriekin, bai ezaugarri automatikoekin. Ondorioz, teknika hauek erabilgarriak direla egiaztatu da.

6.1.6 Pilaketa

Pilaketan, analizatzaile bi bateratzeko, lehenengo analizatzailearen irteeran lortutako informazioa bigarren analizatzailearen sarrera aberasteko erabili da. Orain arte, aberasteko erabili diren ezaugarriak (egitura ezaugarriak) hizkuntzarekiko independenteak izan dira. Gure ekarpena ezaugarri linguistikoak erabiltzea izan da, hau da, lehenengo analizatzailearen irteeran lortutako ezaugarri morfosintaktikoak (kasua eta numeroa bezalakoak) dependentzia-zuhaitzetatik goratzeko printzipio linguistikoak (sintagma, koordinazio eta aditz-taldekoak) erabiltzea. Analizatzailearen zehaztasuna hobetzeko pilaketa erabili da, eta bidean, hurrengo galderei erantzuna ematen ahalegindu gara:

- **Zein ezaugarrirekin lortzen dira emaitzarik onenak pilaketan? Ezaugarri linguistikoekin edo egitura ezaugarriekin? Bakarkako zein ezaugarrirekin edo konbinaketa ezaugarrirekin lortzen dira emaitza hobeak?** Ezaugarri linguistikoak eta egitura ezaugarriak banaka eta konbinatuta goratzeak zehaztasunean duen eragina aztertu ostean, guztien konbinaketarekin emaitzarik onenak lortu dira. Oro har, egitura ezaugarriekin, ezaugarri linguistikoekin baino emaitza hobeak lortu dira.
- **Zein da hobea, Bengoetxea eta Gojenola-ren (2009a) proposamena, arku zuzenetatik ikasten duen pilaketa, ala Martins *et al.*-en (2008) proposamena, analizatzailearen irteeraren arkuetatik ikasten duena?** Oro har, Martins *et al.*-en (2008) proposamenak Bengoetxea eta Gojenola-ren (2009a) proposamenak baino emaitza hobeak eman ditu. Esaldi berriak analizatzerakoan, informazio morfosintaktikoa lehenengo analizatzailearen irteeraren arkuetatik

goratu ostean, bigarren analizatzailearen zehaztasuna hobetzen dela probatu da.

- **Zein da onena pilaketan? Izaera berdineko sistemekin ala izaera desberdineko sistemekin eraikitako pilaketa? Izaera berdineko baina algoritmo desberdineko sistemekin eraikitako pilaketa?** Oinarriarekiko hobekuntzarik onena pilaketaren lehen eta bigarren fasean erabilitako sistemak izaera desberdinekoak direnean gertatzen da, (McDonald eta Nivre, 2011) lanean probatu zuten bezala. Oinarriarekiko bigarren hobekuntzarik onena, bi faseetan, izaera berdineko sistema baina algoritmo ezberdinak erabiltzerakoan lortu dira. Hobekuntza txikienak bi faseetan izaera berdineko sistema eta algoritmo berdina erabiltzerakoan lortu dira. Horrela, sistema eta algoritmo ezberdinen osagarritasuna pilaketan mesedegarria dela esan dezakegu.
- **Zuhaitz-bankuaren tamainak eraginik ote du pilaketan?** Zuhaitz-bankuaren tamainak pilaketan duen garrantzia aztertu da, $Stacklazy_{Stacklazy}$ (sistema-algoritmo berdina, bi faseetan) pilaketaren bitartez eta zuhaitz-banku txikiekin hobekuntza handiena lortu da; $Stacklazy_{MST}$ eta $Stacklazy_{Covington}$ (sistema-algoritmo ezberdinak, bi faseetan) pilaketekin, aldiz, hobekuntzarik handienak zuhaitz-banku handiekin lortu dira.
- **Urre-patroiko ezaugarriekin lortutako hobekuntzak ezaugarri automatikoekin ere gertatzen dira?** Bai, pilaketa ezaugarri automatikoekin ere erabil daitekeela probatu da; emaitza bat aipatzearren, $Stacklazy_{MST}$ pilaketa eginez +1,72 puntuko hobekuntza lortu da LAS neurrian.
- **Pilaketa teknika euskararako analizatzaile sintaktikoa hobetzeko baliagarria da? Zenbateraino?** Oro har, zuhaitz-banku osoari erreparatuz gero, pilaketak emaitza esanguratsuak eman ditu, baina zuhaitz-transformazioan lortutakoak baino apalagoak dira. Salbuespen bakarra $Stacklazy_{MST}$ pilaketa izan da, zuhaitz-transformazioan lortutakoak baino hobeak lortu baitira.

6.1.7 Bozketa bidezko konbinaketa

Analizatzaile sintaktikoaren zehaztasuna hobetzeko bozketa bidezko konbinaketa probatu da. Sistema ezberdinen irteerak konbinatzeko, MaltBlender

bozketa bidezko sistema erabili da. Bozketa bidezko konbinaketa sistemarekin lortutako emaitzak, oinarrizko sistema batekin alderatuz gero, izugarrizko hobekuntzak lortzen direla probatu da (Sagae eta Lavie, 2006; Hall *et al.*, 2007). Konbinatzeko erabili ditugun sistemen irteerak, alde batetik, oinarrizko algoritmoekin osatutako sistemak izan dira, eta, bestetik, sistema hedatuekin (zuhaitz-transformazio eta ezaugarri linguistikoen pilaketarekin) osatutako sistemak. Konbinatzerakoan, bi algoritmo desberdin erabili ditugu, batetik, 3tik $x$ra arteko konbinaketa posible guztiak jasotzea izan da (analizatzaileen irteera ez-onenen konbinaketa, ausazko onena bezala deitu dena), eta , bestetik, x sistema onenen arteko gehikuntza konbinaketa, x onenen konbinaketa bezala deitu dena izan dira. Emaitzak aztertuta, hurrengo galderei erantzuna ematen ahalegindu gara:

- **Konbinatzeko garaian, analizatzaileen LAS puntuazioaren araberako ordenak edo aniztasunak (osagarritasunak) garrantzi handiagoa du?** x onenen konbinaketa eta ausazko onenen konbinaketa probatu dira. Oinarrizko algoritmoekin 12 aldaera ezberdineta-ko analizatzaileen multzoa osatu da eta irteera multzo posible guztiak (3, ...,12) konbinatu ostean, hurrengo ondorioak atera ditugu: ausazko onena metodoarekin, x onenen konbinaketarekin baino emaitza hobeak lortu dira; ausazko onena metodoarekin 6 analizatzaileren irteerekin nahiko da; aldiz, x onenen konbinaketarekin 10 analizatzaileren irteerak behar izaten dira; eta, azkenik, ausazko onena metodoarekin lortutako konbinaketa onena osatzen duten analizatzaileen multzoan, inoiz ez direla x onenen konbinaketa multzoa agertzen gertatu da. Horrela, bozketa sistemetan LAS neurria soilik kontuan hartu beharra baztertuko da, sistema konbinatu onenen artean, beti aurkitzen baitugu LAS sailkapenaren bukaeran daudenak; horretaz gainera, beste faktore garrantzitsu batzuk aurki daitezke: bateragarritasuna, aldaera eta kalitatea (Surdeanu eta Manning, 2010).
- **Oinarrizko sistemekin, zein da hobekuntza? Zein sistemaren artean ematen dira emaitzarik onenak?** 12 oinarrizko sistemen irteerak ausazko onena konbinaketarekin, urre-patroiko ezaugarriekin eta ezaugarri automatikoekin 82,99 eta 81,17 lortu dira LAS neurrian, hurrenez hurren, eta bakarkako oinarrizko sistema onenarekin alderatuz gero ia 3 puntuko hobekuntza lortu da. Konbinaketa onena osatzen duten sistemei erreparatuz, alde batetik esan daiteke MSTParser eta MaltParser izaera desberdineko (globala eta lokala) sistemak agertzen direla beti; beste aldetik, esaldiak ezkerretik eskuinera edo alderantziz

entrenatutako analizatzaileen irteerak konbinatutako sistemak agertzen dira, nahiz eta eskuinetik ezkerrera entrenatutako sistemen irteeren emaitzak LAS neurrian askoz ere baxuagoak izan.

- **Zein da hobekuntza, sistema hedatuen irteerak konbinaketan sartzean? Zein sistemaren artean ematen dira emaitzarik onenak?** Esperimentuak egiteko 26 analizatzaile ezberdin lortu dira (oinarrizkoak + hedatuak), baina 26 sistemaren azpimultzo posible guztiak kalkulatzeko mementuan, esperimentuen kopurua izugarri igotzen zenez, 14 sistema onenak erabiltzea erabaki da. 14 sistema onenen artean aniztasuna mantentzen dela probatu da, urre-patroiko ezaugarriekin 5 zuhaitz-transformazio, 7 pilaketa eta oinarrizko 2 sistemaren irteerak bildu dira, eta, ezaugarri automatikoekin, 5 zuhaitz-transformazio, 6 pilaketa eta oinarrizko 3 sistemen irteerak bildu dira. 14 sistemen (oinarrizko eta hedatuen) konbinaketa kontuan hartuta, 9 sistemaren konbinaketatik onena lortutako emaitza aztertzerakoan, 5 zuhaitz-transformazio, 6 pilaketa eta oinarrizko 3 sistemen irteeren konbinaketa multzotik, konbinaketa onena 4 zuhaitz-transformazio, 2 pilaketa eta oinarrizko 3 sistemen irteeren multzoarekin lortu da; hau da, azken buruan, aniztasuna mantentzen dela esan daiteke. Hemen ere, oinarrizko analizatzaileen konbinaketan bezala MSTParser eta MaltParser izaera desberdineko (globala eta lokala) sistemak eta norabide desberdineko sistemak aurkituko ditugu. Sistema hedatuak gehituz egindako konbinaketetan, urre-patroiko ezaugarriak eta ezaugarri automatikoekin eta ausazko onenaren konbinaketarekin +3,07 eta +4,14ko irabaziak lortu dira; oinarrizko sistemak konbinatu direnean, aldiz, +2,82ko irabaziak baino ez dira lortu.
- **Konbinaketa egoera erreal batean erabili ahal da?** Urre-patroiko etiketak erabili beharrean, etiketa automatikoak erabiliz gero hobekuntza esanguratsuak lortu dira. Emaitza bat aipatzearren, ausazko onena metodoarekin 82,40 (+4,14 oinarri onenarekiko) lortu da.

Euskarako analisi sintaktikoa hobetzeko datuetan eta informazio linguistikoan oinarritutako teknikak (zuhaitz-transformazioak, pilaketa eta bozketa sistema) erabiliz zehaztasuna handitzea lortu da.

6.1.8 Informazio semantikoa

Orain arte, analisirako ezaugarri morfosintaktikoen informazioa erabili da;

alabaina, analizatzaile sintaktikoaren lana hobetzeko informazio semantikoa erabiltzea pentsatu da. IXA taldean, euskararako baliabide semantikoak garatzen diren bitartean, ingeleserako eskuragarri dauden baliabide semantikoak erabili dira. Informazio semantikoa gehitzeko orduan, iturri mota desberdineko eta ingeleserako eskuragarri dauden hurrengo baliabideak erabili dira:

- (Koo *et al.*, 2008) lanean etiketatu gabeko corpusetik lortutako hitz-multzoak
- (Agirre *et al.*, 2008) lanean *WordNet*etik lortutako fitxategi semantikoak eta *synset*ak

Informazio semantikoak dependentzietan oinarritutako analisi sintaktikoan izan dezakeen eragina aztertzeko, *PTB* egitura eredutik dependentzia eredura bihurtzeko bihurketa bi, kategoria-etiketatzailearekin era automatikoan lortutako informazioa eta urre-patroiko kategorien informazioa, eta izaera desberdineko hiru analizatzaile erabili ditugu, hurrengo galderei erantzuna ematen ahalegintzeko:

- **Informazio semantikoa analizatzaile sintaktikoa hobetzeko baliagarria da? Benetako egoera batean baliagarriak dira?** Urre-patroiko kategoriak soilik erabili beharrean, kategoria etiketatzaile batek (ingelesez, *POS tagger*) iragarritako kategoria automatikoekin probatu da. Kategoria etiketatzaile batekin lortutako kategoria automatikoekin probatu ostean, hurrengo hobekuntza esanguratsuak lortu dira: MST sisteman hitz-multzoekin eta *LTH* eta *Penn2Malt* bihurketekin, +1,12 eta +0,33, hurrenez hurren; ZPar sisteman *synset*ekin eta *Penn2Malt* bihurketekin +0,18 eta MST sisteman fitxategi semantikoekin eta *LTH* bihurketekin +0,29. Informazio semantikoa erabiltzeak dependentzia-ereduko analisi sintaktikoaren zehaztasuna hobetzen du, baina kategoria automatikoekin lan egiterakoan, kasu gehienetan oso txikiak eta ez-esanguratsuak izan dira.
- **Zein mailatako informazio semantikoa erabiltzea komeni da?** Eskuragai izan ditugun *WordNet*eko hitz eta adierei buruzko informaziotik adiera zabalena eta adiera zorrotzena duena, eta etiketatu gabeko corpusetik lortutako hitz-multzoetatik maila desberdinetako hitz-multzoak probatu ditugu. Ia emaitza guztiak positiboak izan dira. Kategoria automatikoak erabiltzerakoan, MSTParser sistemarekin, fitxategi semantikoekin eta hitz-multzoekin emaitza esanguratsuak lortu dira. Baina, oro har, kategoria automatikoak erabiltzerakoan, fitxategi semantikoekin eta *synset*ekin lortutako hobekuntzak ez dira esangura-

tsuak izan.

- **Maila desberdineko informazio semantikoak izaera desberdineko sistemetan eragin berdina izango ote du?** Izaera desberdineko hiru sistemekin lan egin da: trantsizioetan oinarritutako analizatzaile determinista (MaltParser), grafoetan oinarritutako analizatzaile globala (MSTParser) eta trantsizioetan oinarritutako ZPar, hau da, bilaketa lokala, determinista eta jalea erabili beharrean, ikasketa diskriminatzaile globala eta *beam* bilaketa erabiltzen duena. Oro har, ia emaitza guztiak positiboak dira. Baina hitz-multzoekin eta fitxategi semantikoekin, MSTParser sistemarekin baino ez dira emaitza esanguratsuak lortu.
- **Zein bihurketa tresna da egokiena informazio semantikoarekin lan egiterakoan?** *Penn Treebank*a osagai-eredutik dependentzia-eredura bihurtzeko bihurtzaile anitz daude. Oro har, bihurtzaile mota bi bereizten dira: dependentzia-etiketa aberatsagoak eta finagoak ematen dituztenak, adibidez, *Pennconverter* eta dependentzia-etiketen multzo murriztua ematen dutenak, adibidez, *Penn2Malt*. Galdera honi erantzuteko, *Penn2Malt* eta *Pennconverter* erabili dira. Oro har, informazio semantikoa erabili gabe, *Penn2Malt* bihurtzailearekin, *LTH* bihurtzailearekin baino emaitza hobeak plazaratu dira hiru sistemetan, bien arteko aldea, Malt (+3,51), MST (+5,49) eta ZPar (+2,37) izan dela. Baina, nahiz eta *LTH* bihurtzailearekin irteera baxuagoa lortu, semantikaren prozesamendurako onenetarikoa dela (Johansson eta Nugues, 2007b) ikus daiteke, eta informazio semantikoa erabiltzerakoan aldeak gutxitu egin direla ikus daiteke. Emaitza batzuk aipatzearren: lehenengo aldia da, *LTH* bihurtzailearekin eta MaltParser sistemarekin, PTB osoan *WordNet*etik lortutako fitxategi semantikoekin emaitza estatistikoki esanguratsuak lortzen direla. Horrela, urre-patroiko kategoria erabilita eta esanahirik sarrienaren metodoa erabiliz, *PTB* osoan +0,36ko hobekuntza esanguratsua lortu genuen LAS neurrian. Baina kategoria etiketatzailearekin era automatikoan lortutako kategoriak erabiltzerakoan hobekuntza +0,17koa izan da. LTH bihurtzailearekin lortutako beste emaitza batzuk aipatzearren, hitz-multzoak eta fitxategi semantikoak erabiltzerakoan, MSTParser sistemarekin eta ezaugarri automatikoekin emaitza esanguratsuak lortu dira. *Penn2Malt* bihurtzailearekin emaitza bi azpimarra daitezke: MSTParser sistemarekin eta hitz-multzoak erabiltzerakoan lortutako +0,33ko hobekuntza esanguratsua, eta, ZPar eta *synset*ak erabilita +0,18ko esangura txikiko

hobekuntza.

- **Kategoria automatikoekin, konbinaketan informazio semantikoa erabiltzea aberasgarria da?** Oinarrizko sistemen eta sistema hedatuen (informazio semantikoekin entrenatutako sistemen) irteerak bozketa bidezko MaltBlender sistemarekin konbinatu dira. Sistema hedatuen irteerak konbinatzerakoan, bai *LTH* bihurtzailearekin, bai *Penn2Malt* bihurtzailearekin antzerako hobekuntzak lortu dira: fitxategi semantikoekin +0,41 bietan, *synset*ekin, +0,28 eta +0,35 eta hitz-multzoekin, +0,37 eta +0,38, hurrenez hurren. Sistema hedatuen irteerak konbinatzerakoan, *LTH* eta *Penn2Malt* bihurtzailearekin antzerako hobekuntza ez-esanguratsuak lortu dira.

6.2 Etorkizuneko lanak

Tesi-lan honek jarraian zerrendatuko diren ikerketa-lerro berriak zabaltzen ditu:

- **Zuhaitz-bankua:** ikusi da, oraindik zuhaitz-bankuaren tamaina handituz gero, zehaztasuna handitu daitekeela; horrela, IXA taldean, euskarazko EPEC-DEP zuhaitz-bankua (Aldezabal *et al.*, 2009; Aranzabe, 2008) handitzen den heinean, *CoNLL-X* formatura egokitzeko beharra ikusten da. Euskarazko zuhaitz-bankua literatura-testu eta egunkariko testuekin osatuta dago; mota bakoitzeko testuak banatuz gero, mota berdineko testuen eragina aztertu, eta zuhaitz-bankuaren tamainaren eragina aztertzeko aukera ematen du.
- **Ezaugarri automatikoak:** Analisi morfologiko eta desanbiguazio moduluak ematen dituen aukeretatik lehenengo aukera (sarriena) erabili da baina etorkizunean beste aukera batzuk ere probatu daitezke. Adibidez, analizatzaile sintaktikoa eta desanbiguatzaile morfologikoa bateratzea (Cohen eta Smith, 2007, Goldberg eta Tsarfaty, 2008, Lee *et al.*, 2011) edo orain erabiltzen ari garen modulua analizatzaile sintaktikoan eragin gehien dituzten ezaugarriak (kategoria, kasua eta mendeko perpausen informazioa) are gehiago kontuan hartzeko moldatzea, eta abar.
- **Konbinaketan:** tesi-lan honek ikerketa lerro berri bat zabaltzen du, zer-nolako analizatzaileak diren eraginkorrenak konbinatzeko garaian, bateragarritasun edo aniztasun neurrietan oinarriturik. Adibidez, (Goldberg eta Elhadad, 2010) lanean analizatzaile sintaktiko ezberdinen arteko ezberdintasunak eta komunean duten egiturazko joerak erabil di-

tzakegu.

- **Euskararako beste analizatzaile sintaktikoen sortzaileak:** Gaur egun, grafoetan eta trantsizioetan oinarritutako modeloen onurak sistema batean bateratzen dituzten sistemak aurki ditzakegu, hala nola, Zpar (Zhang eta Clark, 2008; Zhang eta Nivre, 2011). Horretaz gainera, badira sistema biak integratzen dituztenak, eta horietan zein erabili nahi den aukera ematen digutenak, adibidez, Mate (Bohnet eta Nivre, 2012).
- **Semantika:** *WordNet* ontologia edo hierarkia bat da, eta hiperonimia-hiponimia harreman semantikoarekin hierarkian gora eta behera egiteko aukera eskaintzen du etorkizunari begira, beste maila egoki batera heldu arte, edo bitarteko bestelako errepresentazio semantikoak erabiliko dira. Halaber, egindako esperimentuak euskaran bertan eta bestelako hizkuntzetan probatzeko aukera ematen du.

Bibliografia

Abaitua J. *Complex predicates in Basque: From lexical forms to functional structures. University of Manchester Ph. D.* Doktoretza-tesia, dissertation in Linguistics, 1988.

Aduriz I. Eusmg: morfologiatik sintaxira murriztapen gramatika erabiliz. *Euskararen desanbiguazio morfologikoaren tratamendua eta azterketa sintaktikoaren lehen urratsak. Doktoretza-tesia, Euskal Herriko Unibertsitatea (UPV/EHU)*, 2000.

Aduriz I., Agirre E., Aldezabal I., Alegria I., Ansa O., Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar M., Oronoz M., Sarasola K., Soroa A., eta Urizar R. A framework for the automatic processing of Basque. *Proceedings of the Workshop on Lexical Resources for Minority Languages*, Granada, Spain, 1998.

Aduriz I., Agirre E., Aldezabal I., Alegria I., Arregi X., Arriola J.M., Artola X., Gojenola K., Sarasola K., eta Urkia M. A word-grammar based morphological analyzer for agglutinative languages. *Proceedings of the International Conference on Computational Linguistics (COLING). ISBN: 1-55860-717-X. Vol. 1.*, 1–7, Saarbrucken, Germany, August 2000.

Aduriz I., Aranzabe M.J., Arriola J.M., Atutxa A., de Ilarraza D.A., Ezeiza N., Gojenola K., Oronoz M., Soroa A., eta Urizar R. Methodology and steps towards the construction of epec, a corpus of written basque tagged

at morphological and syntactic levels for automatic processing. *Language and Computers*, 56(1):1–15, 2006a.

Aduriz I., Aranzabe M., Arriola J.M., Díaz de Ilarraza A., Gojenola K., Oronoz M., eta Uria L. A cascaded syntactic analyser for Basque. In Gelbukh A., editor, *Computational Linguistics and Intelligent Text Processing: 5th International Conference CICLing2004, Seoul, Korea, February 15-21*, 2945 lib. of *Lecture Notes in Computer Science*, 124–134. Springer-Verlag GmbH, 2004. ISBN 3-540-21006-7.

Aduriz I., Esparza I., Bengoetxea K., eta Gojenola K. Migración de una gramática sintáctica parcial entre dos formalismos de unificación. *Procesamiento del lenguaje natural, nº 37*, 2006b.

Agirre E., Baldwin T., eta Martinez D. Improving parsing and PP attachment performance with sense information. *Proceedings of ACL-08: HLT*, 317–325, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Agirre E., Bengoetxea K., Gojenola K., eta Nivre J. Improving dependency parsing with semantic classes. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 699–703, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernández G., Lersundi M., *et al.*. Edbl: a general lexical basis for the automatic processing of basque. *Proceedings of the IRCS Workshop on linguistic databases*, 2001.

Aldezabal I. *Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitz-bankua eta gramatika konputazionala.* Doktoretza-tesia, PhD Thesis, Euskal Filologia Saila (UPV/EHU), 2004.

Aldezabal I., Aranzabe M.J., Arriola J.M., eta de Ilarraza A.D. 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, 2009.

Aldezabal I., Gojenola K., eta Sarasola K. Baterakuntzan oinarritutako euskararen analizatzailea: oinarrizko"patr"gramatika. *Euskal Gramatikari eta*

literaturari buruzko Jardunaldiak XXI. mendearen atarian (I-II), 37–60. Euskaltzaindia, 2003.

Aranzabe M.J., Arriola J.M., eta de Ilarraza A.D. Theoretical and methodological issues of tagging noun phrase structures following dependency grammar formalism. *Gramatika Jaietan. Patxi Goenagaren omenez, Julio Urkixo Euskal Filologi Mintegiaren Urtekariaren Gehigarriak, LI*, 57–69, 2008.

Aranzabe M. *Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitz-bankua eta gramatika konputazionala.* Doktoretza-tesia, PhD Thesis, Euskal Filologia Saila (UPV/EHU), 2008.

Arriola J.M. *Euskal Hiztegia-ren azterketa eta egituratzea ezagutza lexikalaren eskuratze automatikoari begira. Aditz-adibideen analisia Murriztapen Gramatika baliatuz, azpikategorizazioaren bidean.* Doktoretza-tesia, Filologia eta Historia-Geografia Fakultatea. UPV/EHU, 2000.

Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Labaka G., Sologaistoa A., eta Soroa A. A framework for representing and managing linguistic annotations based on typed feature structures. *Proceedings of Recent Advances on NLP (RANLP05)*, Borovets, Bulgaria, 2005.

Attardi G. eta Dell'Orletta F. Reverse revision and linear tree combination for dependency parsing. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, 261–264. Association for Computational Linguistics, 2009.

Ballesteros M. eta Nivre J. Maltoptimizer: A system for maltparser optimization. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 2757–2763, 2012.

Bengoetxea K., Agirre E., Nivre J., Zhang Y., eta Gojenola K. On wordnet semantic classes and dependency parsing. *Proceedings of Association for Computational Linguistics*, 649–655, Baltimore, Maryland, USA, 2014.

Bengoetxea K. eta Gojenola K. Desarrollo de un analizador sintáctico estadístico basado en dependencias para el euskera. *Revista de Procesamiento del lenguaje natural, nº 38*, 2007.

Bengoetxea K. eta Gojenola K. Application of feature propagation to dependency parsing. *Proceedings of the 11th International Conference on Parsing Technologies*, 142–145. Association for Computational Linguistics, 2009a.

Bengoetxea K. eta Gojenola K. Exploring treebank transformations in dependency parsing. *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP*, 2009b.

Bengoetxea K. eta Gojenola K. Application of different techniques to dependency parsing of basque. *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, 31–39. Association for Computational Linguistics, 2010.

Bengoetxea K., Gojenola K., eta Casillas A. Testing the effect of morphological disambiguation in dependency parsing of basque. *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, 28–33. Association for Computational Linguistics, 2011.

Bikel D.M. A distributional analysis of a lexicalized statistical parsing mode. *EMNLP*, 182–189, 2004.

Böhmová A., Hajič J., Hajičová E., eta Hladká B. The pdt: a 3-level annotation scenario. *Treebanks: Building and using parsed corpora*, 103–127. Springer, 2003.

Bohnet B. Very high accuracy and fast dependency parsing is not a contradiction. *Proceedings of the 23rd International Conference on Computational Linguistics*, 89–97, 2010.

Bohnet B. eta Nivre J. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1455–1465, 2012.

Bresnan J. *Optimality Theory: Phonology, Syntax and Acquisition*. Oxford University Press, Oxford, 2000.

Brown P.F., Desouza P.V., Mercer R.L., Pietra V.J.D., eta Lai J.C. Class-based n-gram models of natural language. *Computational linguistics*, 18 (4):467–479, 1992.

Buchholz S. eta Marsi E. Conll-x shared task on multilingual dependency parsing. *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 149–164. Association for Computational Linguistics, 2006.

Candito M. eta Seddah D. Parsing word clusters. *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, 76–84, Los Angeles, CA, USA, June 2010. Association for Computational Linguistics.

Carreras X. Experiments with a higher-order projective dependency parser. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, Czech Republic, June 2007.

Carroll J., Briscoe T., eta Sanfilippo A. Parser evaluation: a survey and a new proposal. *1st International Conference on Language Resources and Evaluation*, 447–454, Granada, Spain, 1998.

Cassel S. *MaltParser and LIBLINEAR–Transition-based dependency parsing with linear classification for feature model optimization*. Doktoretza-tesia, Master's thesis, Uppsala University, Uppsala, Sweden, 2009.

Chang C.C. eta Lin C.J. Training v-support vector classifiers: theory and algorithms. *Neural computation*, 13(9):2119–2147, 2001.

Charniak E. Statistical parsing with a context-free grammar and word statistics. *AAAI/IAAI*, 2005:598–603, 1997.

Charniak E. A Maximum-Entropy-Inspired parser. *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, 132–139, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Charniak E. eta Johnson M. Coarse-to-fine n-best parsing and maxent discriminative reranking. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 173–180. Association for Computational Linguistics, 2005.

Chelba C., Engle D., Jelinek F., Jimenez V., Khudanpur S., Mangu L., Printz H., Ristad E., Rosenfeld R., Stolcke A., *et al.*. Structure and performance of a dependency language model. *EUROSPEECH*, 1997.

Chen W., Zhang M., eta Zhang Y. Semi-supervised feature transformation for dependency parsing. *EMNLP*, 1303–1313, 2013.

Chomsky N. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956.

Chomsky N. *Aspects of the theory of syntax*. MIT Press, Cambridge, Massachusetts, 1965.

Chomsky N. *Lectures on Government and Binding*. Foris Publications, The Netherlands, 1981.

Chu Y.J. eta Liu T.H. On the shortest arborescence of a directed graph. *Scientia Sinica, 14*, 1965.

Cohen S.B. eta Smith N.A. Joint morphological and syntactic disambiguation. *EMNLP*, 2007.

Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 1–8. Association for Computational Linguistics, 2002.

Collins M. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637, December 2003. ISSN 0891-2017.

Collins M. *Head-Driven Statistical Models for Natural Language Parsing*. Doktoretza-tesia, PhD Dissertation, University of Pennsylvania, 2009.

Collins M., Ramshaw L., Hajič J., eta Tillmann C. A statistical parser for czech. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 505–512. Association for Computational Linguistics, 1999.

Collins M.J. A new statistical parser based on bigram lexical dependencies. *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 184–191. Association for Computational Linguistics, 1996.

Cortes C. eta Vapnik V. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

Covington M.A. A fundamental algorithm for dependency parsing. *Proceedings of the 39th Annual ACM Southeast Conference*, 95–102, 2001.

Crammer K. eta Singer Y. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

Crammer K. eta Singer Y. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991, 2003.

Culotta A. eta Sorensen J. Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

Ding Y. eta Palmer M. Synchronous dependency insertion grammars: A grammar formalism for syntax based statistical mt. *Workshop on Recent Advances in Dependency Grammars (COLING)*, 90–97, 2004.

Dreyer M., Smith D.A., eta Smith N.A. Vine parsing and minimum risk reranking for speed and precision. *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 201–205. Association for Computational Linguistics, 2006.

Duchier D. eta Debusmann R. Topological dependency trees: A constraint-based account of linear precedence. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 180–187. Association for Computational Linguistics, 2001.

Edmonds J. Optimum branchings. *Journal of Research of the National Bureau of Standards, 71B, 14*, 1967.

Eisner J. Three new probabilistic models for dependency parsing: An exploration. *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Copenhagen, 1996.

Eisner J. Bilexical grammars and their cubic-time parsing algorithms. *Advances in Probabilistic and Other Parsing Technologies*, 29–61. Springer, 2000.

Eryiğit G., Nivre J., eta Oflazer K. Dependency parsing of turkish. *Computational Linguistics*, 34(3):357–389, 2008.

Ezeiza N. *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile sintaktiko sendo eta malgua.* Doktoretza-tesia, University of the Basque Country, Donostia, 2002.

Ezeiza N., Alegria I., Arriola J.M., Urizar R., eta Aduriz I. 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. Association for Computational Linguistics, 1998.

Fan R.E., Chang K.W., Hsieh C.J., Wang X.R., eta Lin C.J. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–.1874, 2008.

Fellbaum C. *WordNet: an electronic lexical database and some of its applications.* MIT Press, Cambridge, Mass, 1998.

Freund Y., Schapire R., eta Abe N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Fujita S., Bond F., Oepen S., eta Tanaka T. Exploiting semantic information for hpsg parse selection. *Research on Language and Computation*, 8(1):1–22, 2010.

Gazdar G., Klein E., Pullum G., eta Sag I. *Generalized Phrase Structure Grammar.* Harvard University Press, Harvard, 1985.

Gildea D. eta Palmer M. The necessity of parsing for predicate argument recognition. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 239–246. Association for Computational Linguistics, 2002.

Goenaga P. *Gramatika bideetan.* Erein, 1980.

Gojenola K. *Euskararen sintaxi konputazionalerantz. Oinarrizko baliabideak eta beren aplikazioa aditzen azpikategorizazio-informazioaren erauzketan eta erroreen tratamenduan.* Doktoretza-tesia, Informatika Fakultatea. Euskal Herriko Unibertsitatea, Donostia, 2000.

Goldberg Y. eta Elhadad M. Easy first dependency parsing of modern hebrew. *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, 103–107. Association for Computational Linguistics, 2010.

Goldberg Y. eta Tsarfaty R. A single generative model for joint morphological segmentation and syntactic parsing. *Proceedings of ACL-HLT*, 2008.

Gómez-Rodríguez C. eta Nivre J. A transition-based parser for 2-planar dependency structures. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1492–1501. Association for Computational Linguistics, 2010.

Hajic J. Building a syntactically annotated corpus: The prague dependency treebank. *Issues of valency and meaning*, 106–132, 1998.

Hall J., Nilsson J., Nivre J., Eryigit G., Megyesi B., Nilsson M., eta Saers M. Single malt or blended? a study in multilingual parser optimization. *Proceedings of the CoNLL Shared Task EMNLP-CoNLL*, 2007.

Hall J. eta Nivre J. A generic architecture for data-driven dependency parsing. *NODALIDA2005*, page 47, 2005.

Hall K. eta Novák V. Corrective modeling for non-projective dependency parsing. *Proceedings of the Ninth International Workshop on Parsing Technology*, 42–52. Association for Computational Linguistics, 2005.

Hellwig P., Ágel V., Eichinger L., Eroms H.W., Heringer H.J., eta Lobin H. Dependency and valency: An international handbook of contemporary research, vol. 1, 2003.

Henderson J.C. eta Brill E. Exploiting diversity in natural language processing: Combining parsers. *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing*, 187–194, 1999.

Hudson R. Word grammar, 1998.

Hudson R.A. *English word grammar*, 108 lib. Basil Blackwell Oxford, 1990.

Johansson R. eta Nugues P. Investigating multilingual dependency parsing. *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 206–210. Association for Computational Linguistics, 2006.

Johansson R. eta Nugues P. Extended constituent-to-dependency conversion for English. *Proceedings of NODALIDA 2007*, 105–112, Tartu, Estonia, May 25-26 2007a.

Johansson R. eta Nugues P. Lth: semantic structure extraction using nonprojective dependency trees. *Proceedings of the 4th International Workshop on Semantic Evaluations*, 227–230. Association for Computational Linguistics, 2007b.

Johansson R. eta Nugues P. Dependency-based syntactic-semantic analysis with propbank and nombank. *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, 183–187, 2008.

Joshi, Aravind, eta Krishna. *Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?* University of Pennsylvania, Moore School of Electrical Engineering, Department of Computer and Information Science, 1985.

Kaplan R. eta Bresnan J. Lexical-functional grammar: A formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, 29–130, 1982.

Karlsson F., Voutilainen A., Heikkila J., eta Anttila A. *Constraint Grammar: Language-independent System for Parsing Unrestricted Text.* Prentice-Hall, Berlin, 1995.

Karttunen L., Gaál T., eta Kempe A. Xerox finite state tool. Barne-txostena, Xerox Research Centre Europe, 1997.

Klein D. eta Manning C.D. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 3–10, 2002.

Koo T., Carreras X., eta Collins M. Simple semi-supervised dependency parsing. *Proceedings of ACL-08: HLT*, 595–603, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Koo T. eta Collins M. Efficient third-order dependency parsers. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1–11, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Kruijff G.J. *A categorial-modal logical architecture of informativity.* Doktoretza-tesia, 2001.

Landes S., Leacock C., eta Tengi R. Building a semantic concordance of english. *WordNet: An electronic lexical database and some applications*, 199–216, 1998.

Lee J., Naradowsky J., eta Smith D.A. A discriminative model for joint morphological disambiguation and dependency parsing. *ACL-HLT*, 2011.

Li W. eta McCallum A. Semi-supervised sequence modeling with syntactic topic models. *Proceedings of the National Conference on Artificial Intelligence*, 20 lib., page 813. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Liang P. *Semi-supervised learning for natural language.* Doktoretza-tesia, Massachusetts Institute of Technology, 2005.

MacKinlay A., Dridan R., McCarthy D., eta Baldwin T. The effects of semantic annotations on precision parse ranking. *First Joint Conference on Lexical and Computational Semantics (*SEM)*, 228–236, Montreal, Canada, June 2012. Association for Computational Linguistics.

Magerman D.M. Statistical decision-tree models for parsing. *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 276–283. Association for Computational Linguistics, 1995.

Marcus M.P., Marcinkiewicz M.A., eta Santorini B. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19 (2):313–330, 1993.

Martins A.F.T., Das D., Smith N.A., eta Xing E.P. Stacking dependency parsers. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 157–166, Honolulu,HI, 2008. Association for Computational Linguistics.

Maruyama H. Constraint dependency grammar. Barne-txostena, Technical Report# RT0044, IBM, Tokyo, Japan, 1990.

McCarthy D., Koeling R., Weeds J., eta Carroll J. Finding predominant word senses in untagged text. *Proceedings of the 42nd Meeting of the*

Association for Computational Linguistics (ACL'04), Main Volume, 279–286, Barcelona, Spain, July 2004.

McDonald R., Lerman K., eta Pereira F. Multilingual dependency analysis with a two-stage discriminative parser. *Proceedings of CoNLL 2006*, 2006.

McDonald R., Crammer K., eta Pereira F. Online large-margin training of dependency parsers. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 91–98. Association for Computational Linguistics, 2005.

McDonald R. eta Nivre J. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37 (1):197–230, 2011.

McDonald R.T. eta Pereira F.C. Online learning of approximate dependency parsing algorithms. *EACL*, 2006.

Mel'čuk I. *Dependency Syntax: Theory and Practice*. State University of New York Press, 1988.

Miller S., Guinness J., eta Zamanian A. Name tagging with word clusters and discriminative training. *HLT-NAACL*, 4 lib., 337–342, 2004.

Müller S. The babel-system: An hpsg prolog implementation. *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, 263–277, 1996.

Nilsson J., Nivre J., eta Hall J. Graph transformations in data-driven dependency parsing. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 257–264. Association for Computational Linguistics, 2006.

Nilsson J., Nivre J., eta Hall J. Generalizing tree transformations for inductive dependency parsing. *Proceedings of the 45th Conference of the ACL*, 2008a.

Nilsson J., Nivre J., eta Hall J. Generalizing tree transformations for inductive dependency parsing. *Proceedings of the 45th Conference of the ACL*, 2008b.

Nilsson P. eta Nugues P. Automatic discovery of feature sets for dependency parsing. *Proceedings of the 23rd International Conference on Computational Linguistics*, 824–832. Association for Computational Linguistics, 2010.

Nivre J. Dependency grammar and dependency parsing. *MSI report*, 5133 (1959):1–32, 2005.

Nivre J. An efficient algorithm for projective dependency parsing. *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, 2003.

Nivre J. Constraints on non-projective dependency parsing. *EACL*, 2006.

Nivre J. Incremental non-projective dependency parsing. *HLT-NAACL*, 396–403, 2007.

Nivre J. Non-projective dependency parsing in expected linear time. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, 351–359. Association for Computational Linguistics, 2009.

Nivre J., Hall J., Kübler S., McDonald R., Nilsson J., Riedel S., eta Yuret D. The conll 2007 shared task on dependency parsing. *Proceedings of EMNLP-CoNLL*, 2007a.

Nivre J., Hall J., eta Nilsson J. Memory-based dependency parsing. *Proceedings of CoNLL*, 49–56, 2004.

Nivre J., Hall J., eta Nilsson J. Maltparser: A data-driven parser-generator for dependency parsing. *Proceedings of LREC*, 6 lib., 2216–2219, 2006a.

Nivre J., Hall J., Nilsson J., Chanev A., Eryiğit G., Kübler S., Marinov S., eta Marsi E. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007b.

Nivre J., Hall J., Nilsson J., Eryiğit G., eta Marinov S. Labeled pseudo-projective dependency parsing with support vector machines. *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 221–225. Association for Computational Linguistics, 2006b.

Nivre J., Kuhlmann M., eta Hall J. An improved oracle for dependency parsing with online reordering. *Proceedings of the 11th international conference on parsing technologies*, 73–76. Association for Computational Linguistics, 2009.

Nivre J. eta McDonald R. Integrating graph-based and transition-based dependency parsers. *Proceedings of ACL*, 2008.

Nivre J. eta Nilsson J. Pseudo-projective dependency parsing. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 99–106. Association for Computational Linguistics, 2005.

Oflazer K. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Comput. Linguist.*, 22(1): 73–89, 1996. ISSN 0891-2017.

Oronoz M. *Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura.* Doktoretzatesia, 2008.

Palomar M., Civit M., Díaz de Ilarraza A., Moreno L., Bisbal E., Aranzabe M., Ageno A., Martí M.A., eta Navarro B. 3LB: Construcción de una base de árboles sintáctico-semánticos para el Catalán, Euskera y Castellano. *Revista de Procesamiento del lenguaje natural, n^o 35*, 2004.

Petrov S., Barrett L., Thibaux R., eta Klein D. Learning accurate, compact, and interpretable tree annotation. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 433–440. Association for Computational Linguistics, 2006.

Petrov S. eta Klein D. Improved inference for unlexicalized parsing. *HLT-NAACL*, 404–411, 2007.

Pollard C. *Head-driven phrase structure grammar.* University of Chicago Press, 1994.

Riezler S., King T.H., Kaplan R.M., Crouch R., Maxwell J.T. III, eta Johnson M. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. *Proceedings of the 40th Annual*

Meeting on Association for Computational Linguistics, ACL '02, 271–278, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Sagae K. eta Gordon A. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. *Proceedings of the Eleventh International Conference on Parsing Technologies*, 2009.

Sagae K. eta Lavie A. Parser combination by reparsing. *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 129–132. Association for Computational Linguistics, 2006.

Schmid H. Treetagger: a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43, 1995.

Sgall P., Hajicovà E., eta J.Panevovà. *The meaning of the sentence in its semantic and pragmatic aspects*. Springer, 1986.

Shalev-Shwartz S., Crammer K., Dekel O., eta Singer Y. Online passive-aggressive algorithms. *Advances in neural information processing systems*, page None, 2003.

Shieber S.M. *An Introduction to Unification-Based Approaches to Grammar*. University of Chicago Press., 1986.

Shinyama Y., Sekine S., eta Sudo K. Automatic paraphrase acquisition from news articles. *Proceedings of the second international conference on Human Language Technology Research*, 313–318. Morgan Kaufmann Publishers Inc., 2002.

Starosta S. *The case for lexicase: An outline of lexicase grammatical theory*. Pinter London, New York, 1988.

Surdeanu M. eta Manning C.D. Ensemble models for dependency parsing: Cheap and good? *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference (NAACL-2010)*, Los Angeles, CA, June 2010.

Suzuki J., Isozaki H., Carreras X., eta Collins M. An empirical study of semi-supervised structured conditional models for dependency parsing. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language*

Processing, 551–560, Singapore, August 2009. Association for Computational Linguistics.

Täckström O., McDonald R., eta Uszkoreit J. Cross-lingual word clusters for direct transfer of linguistic structure. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 477–487, Montréal, Canada, June 2012. Association for Computational Linguistics.

Tapanainen P. *The Constraint Grammar parser CG-2*. Publications of the University of Helsinki, 27, Helsinki, 1996.

Tapanainen P. eta Järvinen T. A non-projective dependency parser. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, 64–71, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

Tarjan R.E. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.

Taylor A., Marcus M., eta Santorini B. The penn treebank: an overview. *Treebanks*, 5–22. Springer, 2003.

Tesnière L. *Elements de syntaxe structurale*. Klincksieck, Paris, 1959.

Tiedemann J. Parallel data, tools and interfaces in opus. *LREC*, 2214–2218, 2012.

Titov I. eta Henderson J. Fast and robust multilingual dependency parsing with a generative latent variable model. *Proceedings of the CoNLL 2007 Shared Task*, 2007.

Urizar R. *Euskal lokuzioen tratamendu konputazionala*. Doktoretza-tesia, Ph. D. thesis, UPV-EHU, 2012.

Wang M., Smith N.A., eta Mitamura T. What is the jeopardy model? a quasi-synchronous grammar for qa. *EMNLP-CoNLL*, 7 lib., 22–32, 2007.

Wu Y.C., Lee Y.S., eta Yang J.C. The exploration of deterministic and efficient dependency parsing. *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 241–245. Association for Computational Linguistics, 2006.

Yamada H. eta Matsumoto Y. Statistical dependency analysis with support vector machines. *Proceedings of the 8th IWPT*, 195–206. Association for Computational Linguistics, 2003.

Zhang Y. eta Clark S. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 562–571. Association for Computational Linguistics, 2008.

Zhang Y., Clark S., *et al.*. Shift-reduce ccg parsing. *ACL*, 683–692, 2011.

Zhang Y. eta Nivre J. Transition-based dependency parsing with rich non-local features. *ACL (Short Papers)*, 188–193, 2011.

Glosategia

adabegi

Grafo bateko puntu berezia da, beste puntu batzuekin lotuta agertzen dena.

analizatzaile aukeraketa (*parse selection*)

Esaldi bat emanda, eta zuhaitz-sintaktiko posible guztien artetik probabilitate altuena duen zuhaitza aukeratzen duen prozesua.

corpus etiketatua

Informazio linguistikoarekin aberastutako corpusa. Hainbat zeregineta-rako baliatzen dira, etiketen arabera.

corpus gordina

Inongo markaketarik gabeko corpusa.

determinista

Trantsizio egoera batetik beste batera pasatzeko normalena da trantsizio aukera bat baino gehiago egotea. Orakuluak bide bat aukeratu ostean, algoritmoak ez ditu atzeko bideak gehiago aztertuko. Sistema honek trantsizio segida bakarra bilatzen du, azkenengo trantsiziora heldu arte.

dependentzia-egiturak

Dependentzietan oinarritutako esaldi-egitura.

dependentzia-zuhaitza

Zuhaitz bat elementuak hierarkikoki gordetzen dituen datu egitura ez lineal bat da. Adabegi multzo bat, zuzendutako arku talde bat. Arku bakoitzak bi adabegi konektatzen ditu guraso-umea erlazioaren bidez. Zuhaitz batean gurasorik ez daukan adabegi bakarra erroa da. Adabegi bakoitza (erroa izan ezik) arku baten bidez konektatua dago eta guraso bakarra izango du. Umeak ez dauzkaten adabegiei hostoak deritzaie. Bi adabegi guraso berdinaren umeak badira, senideak direla esaten da. Ume baten gurasoaren gurasoari aitona deituko diogu.

ezagutza-base lexiko-semantikoa (EBLS)

Hizkuntza ulertu ahal izateko, ordenagailuan hitzei buruz jakin beharreko guztia egon beharko litzateke. Baliabide lexikal egituratuak ditugu, hitzei eta adierei buruzko informazioa dutenak. Adierak klase/azpiklase hierarkien inguruan antolatzen dira.

ezaugarri automatikoak dituen zuhaitz-bankua

Zuhaitz-bankuko esaldiak analizatzaile morfologikotik pasa eta esaldiko hitz bakoitzari dagokion kategoria gramatikala eta marka lexikalak esleitu ostean, lortutako zuhaitz bankua izango da.

entrenatzeko corpusa (*train*)

Corpus osoaren zati handi bat, % 75 - % 80 inguru, erabiltzen da entrenatzeko. Bertatik ikasi behar du sistemak, beraz, garrantzitsua da entrenatzeko zatian ahalik eta kasu gehien, ahal izanez gero, denak agertzea. Sistema batek entrenatzeko corpusean aurkitzen ez duena ezingo du ikasi, eta ikusi ez duen kasu berri bat datorkionean ezingo du emaitza zuzenik eman. Hori dela eta, corpusaren banaketa egiterakoan, inportantea da dependentzia guztietako instantzia edo adibideak hartzea. Saiatu behar dugu dependentzia bakoitzeko dagoen adibide kopuruaren proportzioa mantentzen. Corpusak ausaz aukeratzen dira.

fitxategi semantikoak (*semantic file*)

*WordNet*etik lortutako muturreko errepresentazio zabalena. 45 fitxategi semantiko daude (1 adberbioena, 3 izenondoenak, 15 aditzenak eta

26 izenenak) kategoria sintaktiko eta semantikoetan sailkatuak. Adibidez, izenen fitxategi semantikoek, ekintza, animaliak, tresnak eta abar adierazten dituzten izenak sailkatzen dituzte.

grafo

Egitura matematikoa, erpin eta arku deituriko elementuz osatua.

garapenerako corpusa (*development*)

Entrenamendua egin eta gero, komenigarria da algoritmoen parametroak ondo doitzea, eta horretarako, corpusaren zati bat erabiltzen da. Zati honi garapenerako corpusa deituko diogu. Normalean, zati hau entrenamendurako corpusetik hartzen da (% 10) eta ez da erabiltzen hasierako ikasketa fasean. Lehenengo entrenamendua egin eta gero, zatitxo honetan probatzen da eta, emaitzen arabera, algoritmoaren parametroak egokitzen dira. Prozesu hau hainbat aldiz errepika daiteke ebaluatu aurretik. Sistema egokia dela erabakitzen denean, testerako corpusa erabil daiteke azken ebaluazioa egiteko.

hitz-multzokatze ataza (*word clustering*)

Hitz-multzokatzea, hitz formaren errepresentazio mailaren eskasia gainditzeko, beste errepresentazio maila batzuk erabiltzen dira: lema, kategoria, azpikategoria, eta abar, hitzaren kategoria da gehien erabiltzen dena. Baina kategoria erabiliz gero, katua, arkatza eta pertsona bezalako hitzak maila berean geratzen dira. Arazo honi aurre egiteko, lan honetan hitzen antzekotasunen errepresentazio maila erabili da. Hitzen antzekotasuna probabilitatean oinarritutako hitz-multzokatzea da. Bi hitzen arteko antzekotasuna bilatzeko eta hitz-multzokatzeak egiteko algoritmo asko daude.

hiperonimia (*hypernymy*)

Kontzeptu orokorrenak kontzeptu zehatzagoekin lotzen dituen erlazio semantikoa. Adibidez, auto hitza taxi hitzarekiko hiperonimiako erlazioan dago.

hiponimia (*hyponymy*)

Kontzeptu zehatzenak kontzeptu orokorragoekin lotzen dituen erlazio semantikoa. Adibidez, taxi hitza auto hitzarekiko hiponimiako erlazioan dago.

hitzen adiera-desanbiguazio, HAD (*word sense disambiguation*, WSD)

Konputazio-metodoak erabiliz hitzen agerpenei adiera egokia esleitzen dien prozesua.

informazioaren berreskurapen, IB (*information retrieval*, IR)

Erabiltzaile baten informazio-beharra asetuko duten dokumentuak bilatzeko prozesua. Prozesu honetan erabiltzaileak bere informazio-beharra lengoaia naturaleko adierazpen baten bidez edo termino solte batzuen bidez adieraziko dio IB sistema bati. IB sistemak informazio-behar horretan oinarritutako kontsulta bat sortuko du, eta hor eskatzen den informazioa eduki dezaketen dokumentuen berri emango dio erabiltzaileari.

jalea

Ez dago atzera biderik, dependentzia-arkuak erabaki ostean, ezingo da aldatu.

kategoria-etiketatzailea (*POS tagger*)

Hitz bakoitzari dagokion kategoria gramatikala edota bestelako marka lexikalen bat esleituko dion aplikazioa da.

k geruzako balidazio gurutzatua (*k fold cross-validation*)

Ikasketa automatikoko teknikak aplikatzeko beharrezkoa da, abiapuntu gisa, corpus egituratua eta tamaina handikoa izatea. Zenbat eta handiagoa izan corpusa, orduan eta ikasketa sakonagoa egiteko aukera izango dugu eta, ondorioz, sistema hobea lortuko dugu. Askotan, ordea, errealitatea beste bat da, eta corpusak txikiak izaten dira. Horrelakoetan, ikasteko % 75 edo % 80 erabili behar dugu, gerta baitaiteke informazio nahikorik ez izatea ezagutza corpusetik lortzeko. Hori dela eta, entrenamendurako eta testerako corpusa zatitzeko orduan, beste aukera bat ere badago. Aukera honi *cross-validation* esango diogu. Ideia da corpusa antzeko tamaina duten k zatitan banatzea; gero, k aldiz egiten da entrenamendu fasea, aldi bakoitzean k-1 datu-multzoa erabiliz eta kanpoan utzi den datu-multzoa testerako erabiliz. Horrela eginez gero, guztira k sailkatzaile ditugu. Sailkatzaile hauek konbinatu daitezke emaitza bakarra emateko. Ohikoa da 10 multzoko *cross-validation* erabiltzea.

lema (*lemma*)

Hitza atzizki flexiborik gabe, hiztegietan sarrera gisa dagoen hori.

ontologia (*ontology*)

Mundu errealaren eskema kontzeptuala, non hitzekin izendatzen ditugun kontzeptuak modu hierarkikoan antolatuta dauden.

SemCor

*WordNet*eko adierekin eskuz etiketatutako ingeleseko corpus bat.

sinonimo-multzoa (*synset*)

WordNeteko kontzeptu lexikal edo adiera bati dagokiona. Hau hainbat ale lexikal edo *variant*ek osatzen dute.

testerako corpusa (*test*)

Entrenatzeko eta garapenerako corpusekin sailkatzailea sortu eta gero, jakin behar da sailkatzaile hori zenbateraino den ona, hau da, zenbateko estaldura edo zehaztapena lortzen dugun, sortu dugun sistemarekin. Hori dela-eta, jatorrizko corpusaren gainerako zatia, hau da, entrenatzeko erabili ez duguna, sistema testatzeko edo probatzeko erabiltzen da. Corpusaren zati honetan adibide edo instantzia bakoitzak dependentzia eta burua definituta izango ditu, baina sailkatzaileak ez du dependentzia eta buru hori kontuan izaten iragarpena egiteko, baizik eta, ikasitakoan oinarrituz emango du bere ustez instantzia bakoitzari dagokion dependentzia eta burua. Gero, sailkatzaileak iragarritako dependentzia eta burua, dependentzia eta buru errealekin konparatzen dira eta emaitzak ateratzen dira, asmatze-tasak eta bestelako neurriak kalkulatuz.

urre-patroi (*gold standard*)

Automatikoki eskuratutako emaitzak ebaluatu ahal izateko, eskuz sortzen diren emaitza prototipikoak dira.

WordNet

Ingeleseko hitz eta adierei buruzko informazioa duen ezagutza-base lexikal bat. Izen, aditz, adjektibo eta adberbioak aurkitzen dira bertan *synset* delakoen arabera antolatuta eta hainbat erlazio semantikorekin lotuta.

zuhaitz-banku (*treebank*)

Sintaktikoki etiketatutako corpusa.