eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Diskurtsoko koherentzia erlazioak iragartzeko euskarazko sistema neuronala

Egilea

Erik Angulo Arnaiz

2021

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Diskurtsoko koherentzia erlazioak iragartzeko euskarazko sistema neuronala

Egilea

Erik Angulo Arnaiz

Zuzendariak

Ander Soraluze

Mikel Iruskieta

Laburpena

Dokumentu honetan, diskurtso-egitura automatikoki iragartzeko sistema nola sortu den azalduko da. Zehazki, Rhetorical Structure Theory (RST) hurbilpenean oinarritzen da sistema hori. Sistema horrek euskarazko testuak etiketatzen ditu eta diskurtso-egitura deskribatzen duen zuhaitz-egiturak sortzen ditu. Sistemak ikasketa automatikoko metodoak erabiltzen ditu, zehazki sare neuronalak. Azkenik, sistema hori ebaluatzeko metodoa eta sistema horrekin lortutako emaitzak azalduko dira, baita etorkizuneko lanak ere.

Eskerrak

Amari eta aitari, eman didaten sostengu guztiagatik, bai akademikoan bai pertsonalean.

Lagunei, unibertsitate bidaia dibertigarriagoa egiteagatik.

Ander Soraluze eta Mikel Iruskietari, proiektuaren zuzendariak, eskainitako laguntzagatik. Proiektuaren zuzendaritza ezin hobea izan da beraiei esker.

Ixa taldeari eta Ixakideei, proiektu hau garatzeko aukera emateagatik.

Informatika Fakultateko eta Bilboko Ingeniaritza Eskolako irakasleei, asko ikasi baitut haiei esker.

Gaien aurkibidea

Laburpena i

Eskerrak iii

Gaien aurkibidea v

Irudien aurkibidea ix

Taulen aurkibidea xi

1 Sarrera 1

1.1 Edukia 2

2 Proiektuaren Helburuen Dokumentua 3

2.1 Proiektuaren helburuak 3

2.2 Plangintza 5

2.2.1 LDE diagrama 5

2.2.2 Lan-paketeak 6

2.2.3 Gantt diagrama 8

2.2.4 Lan-metodologia 8

2.2.5 Arriskuak eta prebentzioa 9

2.2.6 Plangintzaren desbiderapena 10

3 Aurrekariak eta baliabideak **13**

3.1 Rethorical Structure Theory eta RST erlazioak 13

3.2 Artearen egoera 17

3.3 Baliabideak 18

3.3.1 Corpusa 18

3.3.2 Sare neuronalak 22

3.3.3 Trantsizioetan oinarritutako RST parserra orakulo dinamikoarekin 29

4 Datuen azterketa eta aurreprozesaketa **33**

4.1 Tokenizazioa 33

4.2 EDUak atributuekin aberastea 34

4.3 Entrenamendurako fitxategien formatuaren azterketa 36

4.4 Ebaluaziorako fitxategien formatuaren azterketa 38

4.5 Formatu aldaketa 40

4.5.1 Hurbilpen teknikoa 40

4.5.2 Inplementazioa 41

5 RST erlazioak iragartzeko sistema **45**

5.1 RST erlazioak ikasten 46

5.2 Esperimentazioa 48

6 Ebaluazioa eta emaitzak **51**

7 Analisi katea **57**

8 Ondorioak **59**

8.1 Proiektuaren ondorioak 59

8.2 Etorkizuneko lana 60

Eranskinak

A RST erlazioen taulak **65**

A.1 Aurkezpenezko erlazioak euskaraz 65

A.2 Edukizko erlazioak euskaraz 69

A.3 Multinuklear erlazioak euskaraz 71

B Ereduek iragarritako RST zuhaitz adibidea **75**

Bibliografia **81**

Irudien aurkibidea

2.1 Proiektuaren helburu nagusia deskonposatuta 4

2.2 Lanaren Deskonposaketa Egitura diagrama 5

2.3 Lan-paketeak garatzeko aurreikusitako beharrezko ordu kopuruen estimazioa 7

2.4 Proiektuaren Gantt diagrama 8

2.5 Lan-paketeak garatzeko behar izan diren ordu errealak eta aurreikusitako orduekin desbiderapena 10

3.1 RST zuhaitz-diagramaren egitura, atalen identifikazioekin, laktosari buruzko testua erabilita. 14

3.2 GMB0301 testuaren RST zuhaitz-diagrama. 16

3.3 GMB0301 testuko rs3 fitxategiaren RST zuhaitz-diagrama. 20

3.4 RSTTools programaren interfazea, segmentazioa egin ondoren RST zuhaitza eraikitzen. 23

3.5 McCulloch-Pitt proposatutako neurona artifizialaren funtzionamendua. . . 24

3.6 Multi-Layer Perceptronaren eskema. 25

3.7 RNN baten arkitektura, folded ezkerrean eta unfolded eskuinean. 27

3.8 LSTM unitatearen eskema, *t* unean. 28

3.9 LSTM unitate baten faseak, *t* unerako 29

4.1 RST informazioa duten formatu desberdinen zuhaitzak. 38

4.2 Formatu aldaketaren programaren atazak 41

6.1 Entrenatutako eredu bakoitzetik lortutako metriken emaitzak. 52

B.1 Eskuz etiketatutako RST zuhaitza: OSA12 testua. 76

B.2 A modeloak iragarritako RST zuhaitza: OSA12 testua. 77

B.3 E modeloak iragarritako RST zuhaitza: OSA12 testua. 78

B.4 G modeloak iragarritako RST zuhaitza: OSA12 testua. 79

B.5 H modeloak iragarritako RST zuhaitza: OSA12 testua. 80

Taulen aurkibidea

3.1 Trantsizio bidezko sistemaren adibidea RST diskurtso parserrarekin . . . 30

5.1 Train, dev eta test banaketetan RST diskurtso erlazioen maiztasuna 47

6.1 Relation puntuazio altuena lortutako ereduen zehaztasuna 53

6.2 Span altuena eta nuclearity altuenak lortutako ereduen zehaztasuna, hurrenez hurren. 53

6.3 Lortutako ereduen metriken alderaketa beste lanekiko 53

6.4 10 iteraziorekin entrenatutako bi eredu, lortutako ereduen zehaztasuna. . . 55

A.1 Euskarazko aurkezpenezko erlazioen arauak eta efektuak 68

A.2 Euskarazko edukizko erlazioen arauak eta efektuak 71

A.3 Euskarazko multinuklear erlazioen arauak eta efektuak 73

Zerrenden aurkibidea

3.1 Ingelesezko testu bat laktosari buruz. 14
3.2 GMB0301 testua: Estomatitis Aftosa Recurrente 16
3.3 GMB0301 testua RST informazioarekin, corpuseko formatuan (rs3) . . . 19
4.1 Tokenizazioa udpiperen bidez 34
4.2 GMB0301 testua tokenizatuta, .raw formatuan 35
4.3 Entrenatzeko formatuaren definizioa (tbk) 36
4.4 GMB0301 testua RST informazioarekin, entrenatzeko formatuan (tbk) . . 37
4.5 .brackets formatuaren definizioa 39
4.6 GMB0301 testua RST informazioarekin, formatu parentetikoan (brackets) 39
4.7 Formatu aldaketaren pseudokodea 42
B.1 OSA12 testua 75

1. KAPITULUA

Sarrera

Ixa taldea [1] UPV/EHU unibertsitateko ikerketa talde bat da. Bertan, hizkuntzaren prozesamenduaren arloko ikerketa egiten da. Ikerlerro inportanteenak dira, besteak beste, ikasketa automatikoa, itzulpen automatikoa, analisi morfologikoa, sintaktikoa, semantikoa, eta corpusak. Datu eta corpus berriak sortzeaz gain, tresnak eta aplikazioak ere sortzen dituzte, aurretik aipatutako atazetarako.

Rethorical Structure Theory [Mann and Thompson, 1988], diskurtso analisiaren arloari dagokio. RST testuak deskribatzeaz arduratzen da, zuhaitz eran adierazita. Orokorrean, testua segmentuetan banatzen da, mailaka antolatuta, eta maila bereko segmentu edota segmentu multzoak besteekiko erlazioen bidez konektatzen dira. Ixa taldeak testuen RST zuhaitzak, segmentuak eta erlazioak era automatikoan jartzeko tresna du, baina euskarazko testuak etiketatzeko ingelesezko corpusarekin entrenatuta dago.

Proiektu honen helburua euskarazko corpusa moldatzea da, euskarazko erlazioekin, tresna hori erabiliz euskarazko testuetatik ikas ditzan eta, ondorioz, euskarazko testuetan segmentuen arteko erlazioak era automatikoan esleitzeko.

[1] Ixa taldearen webgunea: http://http://ixa.si.ehu.es/

1.1 Edukia

Proiektu honen memoria horrela egituratuta dago:

- 2. Kapituluan, proiektuaren helburuak eta plangintza aurkezten dira.
- 3. Kapituluan, lanaren aurrekariak eta baliabideak aurkezten dira.
- 4. Kapituluan, datuen azterketa egiten da eta datuei aplikatu behar zaien aurreprozesaketa azaltzen da.
- 5. Kapituluan, ikasketa-prozesua azaltzen da.
- 6. Kapituluan, lortutako emaitzak aurkezten dira.
- 7. Kapituluan, lanean sortutako ereduak edozein testurekin erabiltzen ahalbidetzen duen sistema aurkezten da.
- Azkenik, 8. Kapituluan, proiektuaren ondorioak eta etorkizuneko lanak aurkezten dira.

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

Atal honetan, GrAL honen helburuak aurkeztuko dira. Gainera, helburu horiek betearazteko planifikazioa aurkeztuko da.

2.1 Proiektuaren helburuak

Proiektuaren helburu nagusia euskarazko testuen *Rethotical Structure Theory* (RST) zuhaitzak eta erlazioak iragartzea da.

Horretarako, lehendabiziko eginbeharrekoa euskaraz RST informazioarekin etiketatutako corpusa eskuratzea izango da. Corpusa dugula, hura aurreprozesatu beharko da RST iragartzeko sistema erabili ahal izateko. Pausu hau beteta, ondorengo helburua sistemarekin ereduak sortzea da. Horren ostean, ereduak ebaluatuko dira RST zuhaitzak eta erlazioak iragartzen hobekien aritzen den eredua hautatzeko. Azkenik, analisi katea sortuko da, eredurik hoberena integratuz, testu gordin batetik zuzenean iragarritako RST zuhaitza eta erlazioak lortzeko.

Beraz, proiektuaren helburu nagusia lortzeko, helburua 4 bloke nagusitan deskonposa daiteke, fasetan ordenatuta:

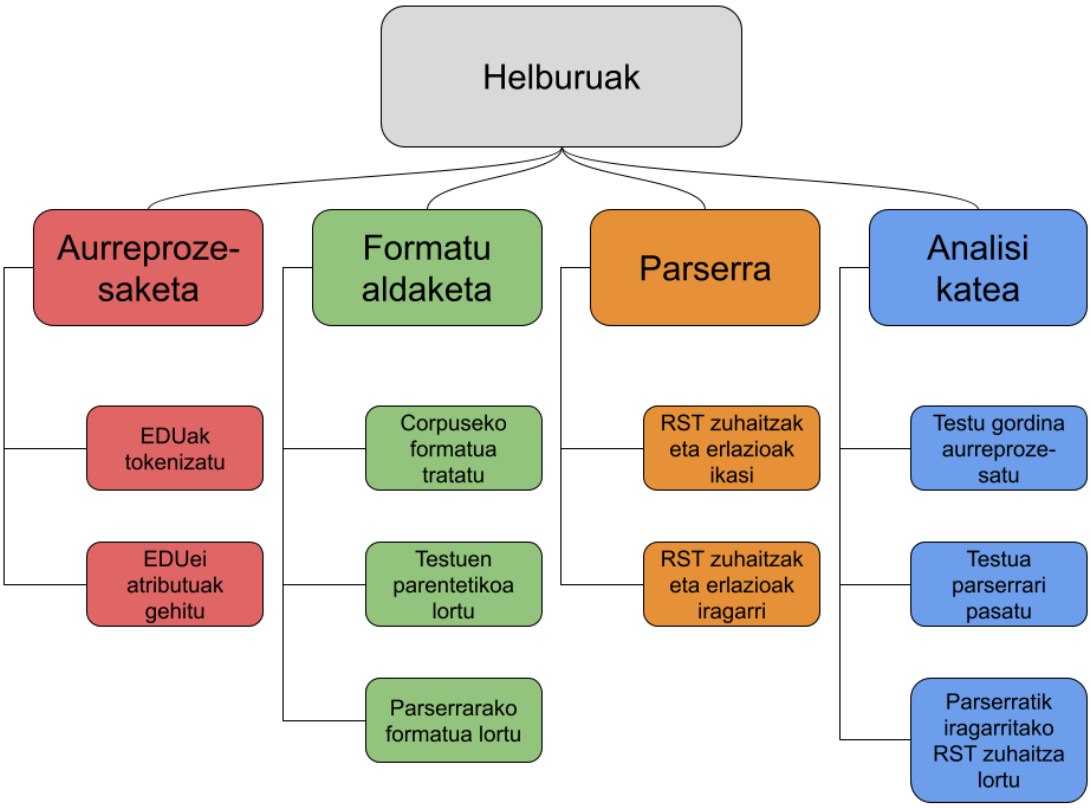
- **Aurreprozesaketa.** Hemen RST zuhaitzak eta erlazioak iragartzeko aurretik egin beharreko prozesuak egingo dira. Horien artean, EDUen (testuko segmentuen) tokenizazioa eta EDU bakoitzaren beharrezko atributuak lortzea.

- **Formatu aldaketa.** Corpuseko formatuan adierazitako RST zuhaitzetatik abiaturik, RST iragartzeko sistemak behar duen RST zuhaitzen formatua lortzea da atal honen helburua.

- **Parserra.** Hau da RST iragartzeko sistema. Ikasketa-fasea sare neuronalekin egingo da. Corpuseko RST zuhaitzak erabilita, ereduak sortuko dira testu berrietarako RST zuhaitzak eta erlazioak iragartzeko aukera izateko.

- **Analisi katea.** Testu gordin bati RST zuhaitza iragarriko zaio. Horretarako, testua aurreprozesatuko da, parserrara bidaliko da, eta parserrak bueltatzen duen iragarritako RST zuhaitza gordeko da.

Euskarazko testuen RST zuhaitzak eta erlazioak iragartzeko helburuaren deskonposaketa 2.1 Irudian ikus daiteke.



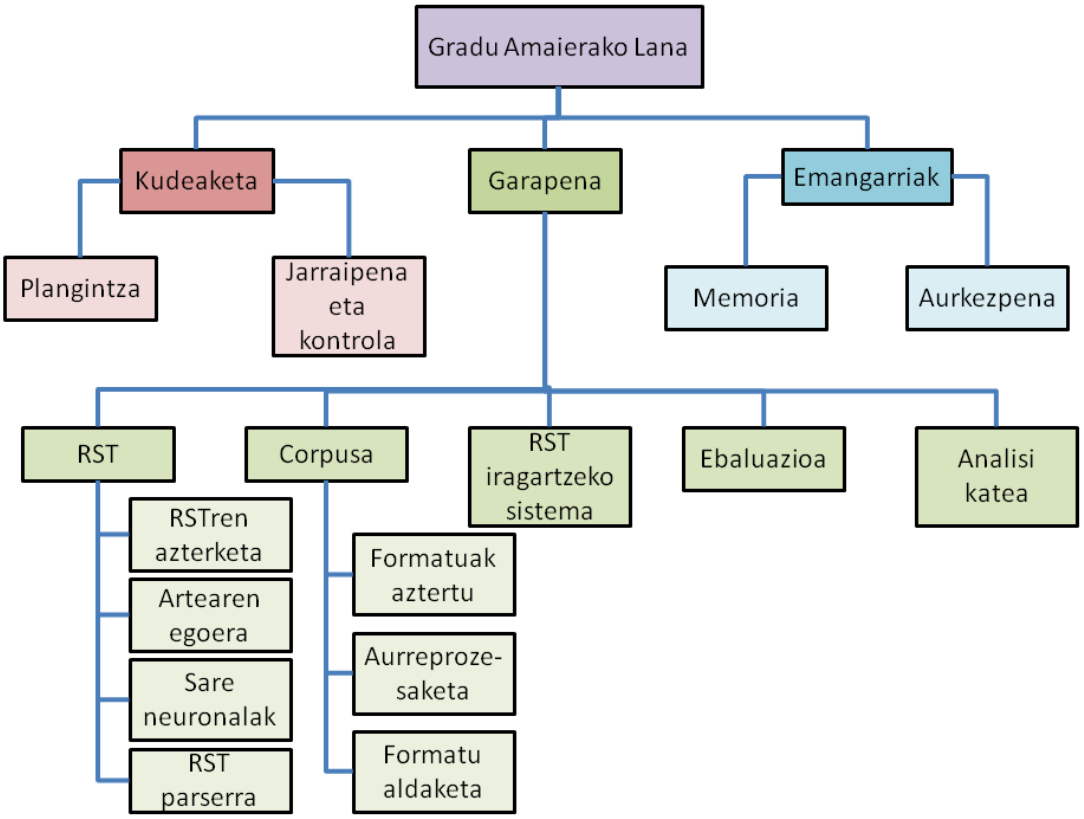
2.1 Irudia: Proiektuaren helburu nagusia deskonposatuta

2.2 Plangintza

Proiektuaren plangintza egiteko, lehenik eta behin proiektuaren atazak identifikatu dira, eta LDE diagrama batean aurkeztu dira. Ondoren, ataza bakoitzari dagokion ordu estimazioa kalkulatu da, azkenik Gantt diagrama sortzeko. Plangintzan proiektuaren garapenean egon daitezkeen arriskuak identifikatu dira ere.

2.2.1 LDE diagrama

LDE diagrama baten bidez, proiektuaren garapena lan-deskonposaketa egitura hierarkiko batean aurkezten da. GrAL honen LDE diagrama 2.2 Irudian ikus daiteke.



2.2 Irudia: Lanaren Deskonposaketa Egitura diagrama

Lan-deskonposaketa hiru lan-pakete nagusitan banatzen da: kudeaketari, garapenari eta emangarriei dagozkienak. Kudeaketa atalean, proiektuaren plangintza eta jarraipenarekin zerikusia duten atazak biltzen dira. Garapena atalean, egindako ikerketa, inplementazioa

eta diseinuarekin zerikusia duten atazak biltzen dira. Emangarrien atalean, proiektuaren memoriarekin eta defentsarekin zerikusia duten atalak biltzen dira.

2.2.2 Lan-paketeak

Azpiatal honetan 2.2 Irudian agertzen den LDE diagramako lan-paketeak azalduko dira, zehaztasun handiagorekin.

- **Plangintza.**

  Ataza honen bidez proiektuaren planifikazioa burutu da. Proiektuaren atazak definitu dira, ataza bakoitzaren iraupena estimatu da eta atazak Gantt diagrama batean kokatu dira hasiera eta bukaera datak zehazteko.

- **Jarraipena eta kontrola.**

  Plangintzan ezarritako epeak betetzen direla eta proiektua bideragarri mantentzen dela ziurtatuko da ataza honen bidez. Horretarako, proiektuko zuzendariekin bilerak egin ditugu. Bileren bidez zereginak era egokian garatzen ari garela bermatuko da ere.

- **RST.**

  Ataza honetan RSTri buruzko lanak irakurriko dira, teoria ulertzeko. Gainera, RSTren inguruan egin diren ikerketak irakurriko dira, artearen egoera barne. Beste alde batetik, RST iragartzeko garatu diren sistemen teknikak aztertuko dira. Tekniken artean sare neuronalak eta parserra aurkitzen dira, beraz teknika bakoitzeko azterketa egingo da.

- **Corpusa.**

  Ataza honek corpusarekin zerikusia duten azpiataza guztiak bilduko ditu. “Formatuak aztertu” azpiatazan, corpuseko formatua eta RST iragartzeko sistemak behar dituen formatuak aztertuko dira. “Aurreprozesaketa” azpiatazan corpusa eskuratu eta aurreprozesatzeko egin beharreko pausuak biltzen dira. “Formatu aldaketa” azpiataza corpusa RST iragartzeko sistemak behar dituen formatuan lortzean datza.

- **RST iragartzeko sistema.**

  Ataza honetan RST iragartzeko sistema martxan jarriko da. Gainera, RST zuhaitzak eta erlazioak iragarriko dituzten ereduak entrenatuko dira.

- **Ebaluazioa.**

  Ataza honetan, sortutako ereduak ebaluatuko dira, RST zuhaitzak eta ereduak iragartzen direla egiaztatuz. Lortutako emaitzak aztertuko dira, eta ereduen arteko konparaketak egingo dira, hoberena hautatzeko.

- **Analisi katea.**

  Ataza honetan zuzenean testu gordin bat jasota iragarritako RST zuhaitza eta erlazioak lortzen duen sistema bat sortuko da. Horretarako, jasotako testua aurreprozesatuko da, lortutako eredu hoberenarekin iragarpena egingo da, eta emaitza gordeko da.

- **Memoria.**

  Dokumentu hau bera da ataza honetan garatuko den emangarria. Dokumentuan proiektuan zehar egindako lana era sakon batean aurkeztuko da.

- **Aurkezpena.**

  Ataza honetan proiektuaren defentsa prestatuko da, beharrezko gardenkiak sortuz.

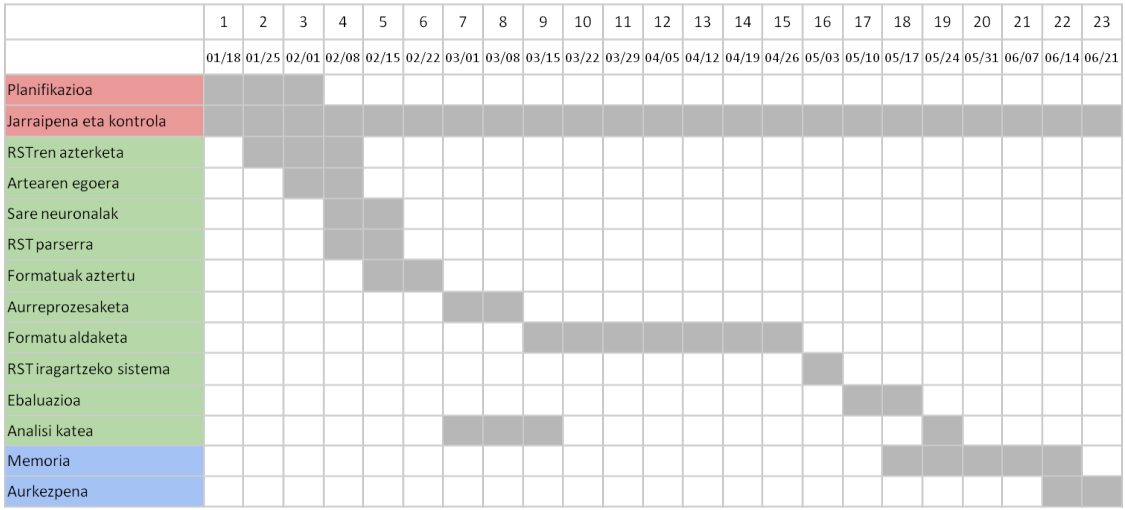
Atazak		Ordu estimatuak
	Totala	**310**
Kudeaketa		**30**
	Plangintza	10
	Jarraipena eta kontrola	20
RST		**40**
	RSTren azterketa	10
	Artearen egoera	10
	Sare neuronalak	10
	RST parserra	10
Corpusa		**100**
	Formatuak aztertu	10
	Aurreprozesaketa	10
	Formatu aldaketa	80
RST iragartzeko sistema		**15**
Ebaluazioa		**15**
Analisi katea		**20**
Emangarriak		**90**
	Memoria	70
	Aurkezpena	20

2.3 Irudia: Lan-paketeak garatzeko aurreikusitako beharrezko ordu kopuruen estimazioa

Lan-pakete bakoitza garatzeko aurreikusitako beharrezko ordu kopuruen estimazioa 2.3 Irudian ikus daiteke.

2.2.3 Gantt diagrama

Gantt diagrama batean, lan-pakete bakoitzeko bere hasiera eta bukaera datak aurkezten dira. 2.4 Irudian proiektu honetako lanaren jarraipenari dagokion Gantt diagrama aurkezten da. Proiektuaren hasiera data 2021.eko urtarrilaren 18a izan da, eta amaiera data 2021.eko ekainaren 27a da, defentsa baino lehen. Guztira, 23 astetan zehar garatuta izan da. Aste bakoitza zenbakituta dago diagraman, astearen asteleheneko egunaren data adieraziz.



2.4 Irudia: Proiektuaren Gantt diagrama

2.2.4 Lan-metodologia

Proiektu osoan zehar, proiektuaren jarraipena eta egoera aztertzeko, zuzendariekin bilerak adostu dira. Orokorrean, astero bildu gara era presentzialean Donostiako Informatika Fakultatean, eta ordubeteko iraupena izan dute. COVID-19ak eragindako osasun egoera dela eta, bilera batzuk *online* egin dira. Bileretatik kanpo komunikatzeko posta elektronikoa erabili da.

Garapenari dagokionez, Ixa taldeak eskainitako zerbitzarian lan egin da, taldeak garatutako RST iragartzeko sistema erabiltzeko aukera izateko.

GrALean zehar erabilitako baliabideak honakoak izan dira:

- **Overleaf.** [1] LaTeX-en idazteko hodeian dagoen zerbitzua da. Memoria bertan garatu da, eta zuzendariekin partekatuta dago.

- **OpenSSH.** [2] Ixa taldeko zerbitzarietara konektatzeko SSH (*Secure Shell*) protokoloaren bidez egingo da, bertako makinetan terminala erabiltzeko eta komandoak egikaritzeko. Konexioa gauzatzeko OpenSSH tresna erabili da.

- **WinSCP.** [3] Ixa taldeko zerbitzariaren eta bezero ordenagailuaren artean fitxategi transferentziak egiteko ahalbidetzen duen programa.

- **Google Drive.** [4] Proiektuan zehar garatutako kodea eta emangarriak (memoria eta aurkezpena) segurtasun-kopia bezala gordetzeko. Segurtasun-kopia pendrive batera kopiatuko da ere.

- **Microsoft Office.** [5] Aurkezpenerako gardenkiak prestatzeko programa hau erabili da. Gainera, memorian zehar aurki daitezkeen diagrama batzuk tresna honekin egin dira.

- **Blackboard Collaborate eta Google Meet** [6] [7]. Bilerak era birtualean egiteko erabili diren tresnak. Audioa, kamera eta ordenagailuko pantaila konpartitzen ahalbidetzen dituzte.

2.2.5 Arriskuak eta prebentzioa

Proiektua garatu bitartean, atzerapenak eragin dezaketen arriskuak gerta daitezke. Hori dela eta, gerta litezkeen arrisku garrantzitsuenak identifikatu dira eta bakoitza prebenitzeko edo kudeatzeko plan bat sortu da:

- **Informazio galera.** Memoria edota kode fitxategiak gordetzen dituzten disko gogorrak apurtzen badira, bertako informazioa galduko da. Prebentzio bezala, egunero garatutakoaren segurtasun-kopia egingo da pendrive batean eta Google Driven.

[1] https://www.overleaf.com/
[2] https://www.openssh.com/
[3] https://winscp.net/eng/index.php
[4] https://www.google.com/drive/
[5] https://www.office.com/
[6] https://www.blackboard.com/es-es/resources/blackboard-collaborate-overview
[7] https://meet.google.com

- **COVID-19ak eragindako osasun egoera.** Pandemiaren egoeraren arabera, GrAL-eko epeetan aldaketak egon daitezke. Prebenitzeko, kasu horretan plangintza alternatibo bat proposatuko da. Bestalde, plangintzan ezarritako epeetan desbiderapena sor daiteke birusarekin kontagiatuz gero. Atzerapen handia egotekotan, proiektua hurrengo deialdian aurkeztea baloratuko da.

2.2.6 Plangintzaren desbiderapena

Orokorrean, 2.3 Irudian aurkezten diren ataza bakoitzerako aurreikusitako estimatutako orduak bete dira. Hala ere, denbora desbiderapenak egon dira, 2.5 Irudian ikus daitezkeenak.

Atazak		Ordu estimatuak	Ordu errealak	Desbiderapena
	Totala	**310**	**353**	**43+**
Kudeaketa		**30**	**36**	**6+**
	Plangintza	10	10	-
	Jarraipena eta kontrola	20	26	6+
RST		**40**	**40**	**-**
	RSTren azterketa	10	10	-
	Artearen egoera	10	10	-
	Sare neuronalak	10	10	-
	RST parserra	10	10	-
Corpusa		**100**	**120**	**20+**
	Formatuak aztertu	10	10	-
	Aurreprozesaketa	10	10	-
	Formatu aldaketa	80	100	20+
RST iragartzeko sistema		**15**	**12**	**3-**
Ebaluazioa		**15**	**15**	**-**
Analisi katea		**20**	**20**	**-**
Emangarriak		**90**	**110**	**20+**
	Memoria	70	90	20+
	Aurkezpena	20	20	-

2.5 Irudia: Lan-paketeak garatzeko behar izan diren ordu errealak eta aurreikusitako orduekin desbiderapena

Kudeaketari dagokionez, bilerak egiteko 20 sesio planifikatu genituen, bat astero eta ordubetekoak. Hala ere, azkenean 6 bilera 2 ordutara arte luzatu dira. Beraz, “Jarraipena eta kontrola” atazean 6 orduko desbiderapena egon da.

Bestalde, formatu aldaketa lan-paketea garatzeko aurreikusitako denbora baino 20 ordu gehiago behar izan dira, inplementazio garaian agertutako arazoak konpontzeko.

Aldiz, RST iragartzeko sistemarekin ereduak sortzeko 3 ordu gutxiago behar izan dira.

Memoriaren kalitatea hobetu nahi izan da, diagrama, adibide eta azalpen argiak jarriz. Horregatik, estimatutako orduak baino 20 ordu gehiago behar izan dira.

Azkenik, proiektu hau garatzeko 353 ordu behar izan dira guztira.

3. KAPITULUA

Aurrekariak eta baliabideak

3.1 Rethorical Structure Theory eta RST erlazioak

Rethorical Structure Theory (RST) testu batean dauden parte ezberdinen artean egon daitezkeen erlazioak deskribatzen dituen teoria da, [Mann and Thompson, 1988]ek proposatutakoa. RSTk testuen koherentzia azaltzen du era hierarkiko batean. Arlo honetan, koherentziaren esanahia, haien artean kontraesanik edo aurkakotasunik sortzen ez duten bi osagaien arteko erlazio logikoa da. Teoriaren jatorria testuak era automatikoki sortzeko helburutik dator [Marcu, 2000b]. Gaur egun arte, RST aldatuz joan izan da teoria originala hobetzen duten hainbat moldaketekin [Taboada and Mann, 2006b]. Gainera, RST hizkuntzalaritzan balioztatua izan da.

Koherentziaren eta egitura hierarkiko honetan, RST zuhaitza deiturikoa, testuaren zatiak mailaka antolatzen dira. Testua zatietan ala segmentuetan banatzeko prozesuari segmentazioa deritzo, diskurtso-segmentazioa. Banandutako segmentuek sailkapen teoriko neutral bat jarraitu behar du, eta segmentuek osotasun funtzional independente izan behar dute, [Mann and Thompson, 1988]ek proposatutako segmentu definizioaren arabera. Segmentuak Elemmentary Discourse Unit (EDU) ere deitu daitezke. EDUa oinarrizko diskurtso unitatea da. EDUen multzoei unitate-multzo (*span*) deritzo. *Span*ak eta EDUak haien artean erlazionatzen dira koherentzia erlazioen bidez.

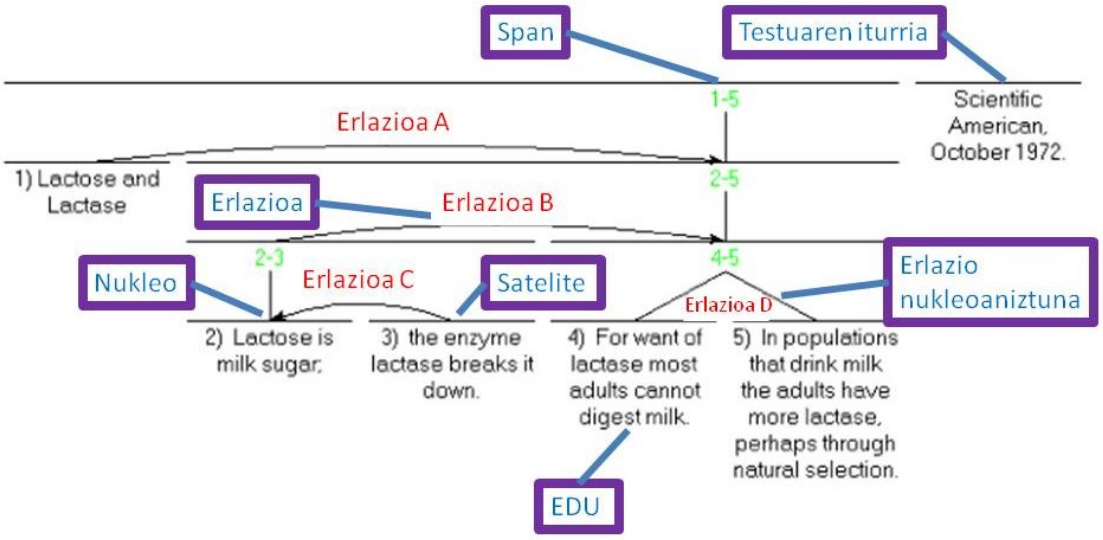
Erlazio bakoitzeko, gehienetan, lotutako segmentu bat nukleoa izango da eta bestea satelitea. Nukleoak testuaren ideia garrantzitsua du, sateliteak, ordea, nukleoaren esanahia aldatzen du. Nukleoek garrantzi handiagoa daukate testuan sateliteak baino, ondorioz ba-

tzutan sateliteak ematen duen informazioa ulergaitza da nukleoko informazioa izan ezean. Beste erlazio batzuk nukleoaniztunak (multinuklearrak) dira, honek esan nahi du lotzen dituen bi *span*ak edo segmentuak nukleoak direla.

RST zuhaitz-diagramak era askotan irudikatu daitezke eta adibide gisa 3.1 Irudikoa har daiteke. Erlazioak gezien bidez adierazten dira, eta geziak apuntatzen duen helburua *span*a ala nukleoa da, jatorria satelitea da, beraz. Erlazioa nukleoaniztuna bada, diagrametan bi segmentu ala *span* lotzen dituzten bi lerro zuzenekin adierazten da. Diagraman, EDU bakoitza zenbaki batekin identifikatzen da. Era berean, EDUak eta *span*ak batzen dituen *span*a identifikatzeko bere azpiko EDU eta *span*en EDUen zenbakiak izango ditu. Irudian erlazionatu gabeko EDU bat dago, testuaren iturria delako eta ez delako testuaren parte.

3.1 Zerrenda: Ingelesezko testu bat laktosari buruz.

```
1  Lactose and Lactase. Lactose is milk sugar; the enzyme lactase breaks it down. For want of
       lactase most adults cannot digest milk. In populations that drink milk the adults have more
       lactase, perhaps through natural selection. Norman Kretchmer, Scientific American, page 70,
       October 1972.
```



3.1 Irudia: RST zuhaitz-diagramaren egitura, atalen identifikazioekin, laktosari buruzko testua erabilita.

3.1 Irudian 3.1 Zerrendako testuaren RST zuhaitz-diagrama bat ikus daiteke. Bertan, EDUak (testua dutenak) eta *span*ak agertzen dira, zuhaitz hierarkia osatuz. Ikus daitekeenez, *span* azpian beste *span* edo EDUak daude, eta *span* horrek beste *span* batekin

erlazionatzen da koherentzia erlazio batekin. Adibidez, *Erlazioa A* erlazioaren gezia aztertuta, EDU_1 du satelite bezala, eta $span_{2-5}$ nukleo bezala, bere azpian dagoen $span_{2-3}$ eta $span_{4-5}$az osatuta. *Erlazioa D* erlazioa, aldiz, nukleoaniztuna da, eta gezi gabeko bi lerroekin adierazten dira bi nukleoak (EDU_4 eta EDU_5). Erlazio honek $span_{4-5}$ eratuko du hierarkiako goiko mailan, beste *span*ekin erlazionatuz.

RST teoria sortu zenetik, hasierako erlazioez gain beste erlazio-zerrenda berri batzuk proposatu ziren. Batez ere, hizkuntzalariek faltan botatzen zituzten fenomenoak deskribatzeko [Taboada and Mann, 2006b].

Erlazioak bi mota nagusitan sailka daitezke: nukleoaniztunak eta nukleo-satelite motakoak. Azken hauek edukizkoak eta aurkezpenezkoak izan daitezke. Edukizkoek izaera semantikoa dute, idazleak EDUen edo *span*en artean erlazioko izenaren efektua dagoela jakinarazten dio irakurleari; adibidez, *Ondorioa*. Aurkezpenezkoek, aldiz, izaera erretorikoa dute, unitateen arteko loturak irakurlearengan dagokion efektua eragiteko asmoa dute; adibidez *Laburpena*. Euskarazko erlazioei dagokionez, hainbat motatakoak daude [Iruskieta, 2014]. Adibidez: *elaborazioa, kontzesioa, lista, metodoa, kontrastea, helburua,* eta abar.

Bestalde, erlazio bakoitzak bere arauak eta efektuak ditu. Hona hemen, adibide gisa, *Laburpena* erlazioarena.

- **Erlazioa:** laburpena (aurkezpenekoa)
  - **Arauak Nukleoan:** EDU bat baino gehiagoz osatuta egon behar da, hau da, *span* batez
  - **Arauak Satelitean:** baldintzarik gabe
  - **Arauak Nukleo-Satelitean:** nukleoan idatzitakoaren sintesia satelitean agertzen da; beraz, sateliteko informazioa nukleokoa baino laburragoa da
  - **Efektua:** Irakurleak satelitean dagoena nukleoan dagoenaren laburpena dela onartzen du

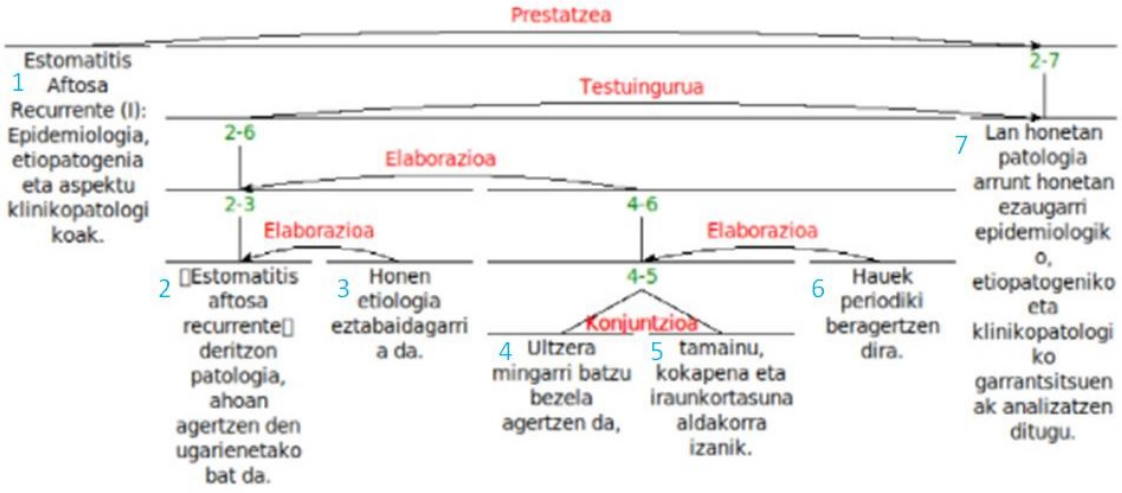
Eranskineko A Kapituluan euskarazko erlazio guztiak ikus daitezke.

Erlazioek sortzen duten efektuak eta RST zuhaitz-diagrama baliatuz analisi prozesua egiten da. Analisi prozesuaren helburua testu baten ulermena era egituratuan aurkeztea da. Analisia egiten duen pertsonari behatzailea ala etiketatzailea deitzen zaio. Testu baten

RST analisia ez da adiera bakarrekoa eta, behatzailearen interpretazioaren arabera, modu batera edo bestera adieraz daiteke. Gainera, behatzaile berak RST analisi desberdinak egin ditzake testuaren konplexutasunaren eta anbiguotasunaren arabera.

3.2 Zerrenda: GMB0301 testua: Estomatitis Aftosa Recurrente

```
1  Estomatitis Aftosa Recurrente (I): Epidemiologia, etiopatogenia eta aspektu klinikopatologikoak.
       "Estomatitis aftosa recurrente" deritzon patologia, ahoan agertzen den ugarienetako bat da.
       Honen etiologia eztabaidagarria da. Ultzera mingarri batzu bezela agertzen da, tamainu,
       kokapena eta iraunkortasuna aldakorra izanik. Hauek periodiki beragertzen dira. Lan honetan
       patologia arrunt honetan ezaugarri epidemiologiko, etiopatogeniko eta klinikopatologiko
       garrantsitsuenak analizatzen ditugu.
```



3.2 Irudia: GMB0301 testuaren RST zuhaitz-diagrama.

3.2 Zerrendan aurkeztutako testuaren analisiaren RST zuhaitz-diagrama 3.2 Irudian ikus daiteke. Bertan, etiketatzailearen arabera, EDU_1 eta $span_{2-7}$ lotzen dituen erlazioa *Prestatzea* da. EDU_1 testuko lehenengoa denez, etiketatzailearen arabera, segmentu horretako testuak irakurlea prestatzen du, irakurriko duen gaia iragarriz eta erlazio horretako nukleoko zatia ($span_{2-7}$) irakurtzeko interesa handituz. EDU_2 eta EDU_3 *Elaborazioa* erlazioarekin lotuta daude, eta irakurleak EDU_3k EDU_2 elaboratzen duelakoan dago behatzailea. EDU_4 eta EDU_5 *Konjuntzioa* erlazio nukleoaniztunarekin erlazionatuta daude, bi EDUek osotasun bat eratzen dutela eta elkarrekin erlazionaturik daudela ezagutzen duelako irakurleak. Osatzen duten $span_{4-5}$aren testu edukia EDU_6k elaboratzen du *Elaborazioa* erlazioaren bitartez. *Elaborazioa*rekin ere $span_{4-6}$a $span_{2-3}$arekin erlazionatzen da. Azkenik, hauek osatzen duten $span_{2-6}$a *Testuingurua* erlazioaren satelitea izango da, EDU_7a

nukleoa izanik. Kasu honetan, *span*ak dituen EDUak Nukleoko EDUa hobeto ulertzeko daude. Hau da, irakurleak EDU_7 ez du guztiz ulertuko $span_{2-6}$ko EDUko testuak irakurri gabe, hauek EDU_7rako testuingurua baitaukate.

3.2 Artearen egoera

RST teoria proposatu zenetik, hainbat lan eta aurrerapen egon dira. Garrantzitsuenetariko bat RSTko corpusak eratzea izan da. Izan ere, horiek gabe ezin izango lirateke RSTko beste aplikazio batzuk sortu. Gainera, RSTren inguruan hainbat *Shared-Task* [1][2][3] antolatu izan dira, zenbait hizkuntzetako RST corpusak eta ebaluatzeko tresnak eskuragarri utziz, RST aplikazioak sortzeko ala hobetzeko helburuarekin.

Corpus on bat hainbat behatzailek etiketatutako testuen RST zuhaitzekin eratuta dago. Orokorrean, corpusek dituzten testuak hainbat arlotakoak dira, horien artean, medikuntza, literatura, berriak, iritzi artikuluak, eta abar. Hau guztia biltzen duen corpusa badago euskaraz, eta gainera, Gradu Amaierako Lan hau garatzeko erabili izan da: Euskal RST Treebank [Iruskieta et al., 2013] [4], Ixa taldeak garatua. 3.2 Irudian agertzen den RST zuhaitza corpus honetatik eskuratu da.

Corpus hori baliatuta, GrAL honetan, euskarazko testuen RST erlazioak era automatikoan etiketatu nahi dira. Ildo honetan, RSTko zuhaitzak, segmentuak eta erlazioak iragartzeko badaude hainbat lan, adibidez, [Fu et al., 2016] eta [Braud et al., 2016a]. Bi lan horietan *Long Short-term Memory* sare neuronalak erabiltzea proposatzen dute, *embedding*ak baliatuz [Iruskieta and Braud, 2019].

Horretaz aparte, gainera, testuen RST zuhaitzak beste ataza batzuetan erabilgarriak dira. Horien artean, sentimenduen analisirako [Kraus and Feuerriegel, 2017], iritzi eta produktuen berrikuspen faltsuak detektatzeko [Popoola, 2017], eta RSTetan koherentzia aztertzeko [Skoufaki, 2020] proiektuak aurkitzen dira, adibidez. Euskarari dagokionez, sentimenduen analisirako [Alkorta et al., 2019] eta laburpenak egiteko [Atutxa et al., 2021] proiektuak egin dira ere. Aplikazio gehiago biltzen dituen dokumentua aztertu nahi izanez gero, ikusi [Taboada and Mann, 2006a].

[1] Shared-Task ikerketa arlo baten inguruan ikerketa taldeek problema zehatzak ebazteko antolatzen diren zereginak dira.

[2] 2019an antolatu zen Shared-Taskaren webgunea: https://sites.google.com/view/disrpt2019/shared-task

[3] 2021ean antolatu den Shared-Taskaren webgunea: https://sites.google.com/georgetown.edu/disrpt2021

[4] Euskal RST Treebankaren corpusa hemen atzigarri: https://ixa2.si.ehu.eus/diskurtsoa/

Azkenik, Simon Fraser Univertsityk mantentzen duen dibulgazioko RST web [5] orrialdeari esker, informazio ugari lor dezakegu RSTri buruz. Gainera, RSTko hainbat publikazio biltzen dituen bibliografia [6] badute ere.

3.3 Baliabideak

3.3.1 Corpusa

Euskarazko RST sistema garatzeko, ezinbestekoa da RST etiketez osatutako testu multzoa izatea. Hori dela eta, lan honen lehenengo betebeharra corpusa osatzea izango da.

Euskaraz idatzitako testuen corpusak lortzea erraza da. Euskaraz idatzitako testu literarioetatik, berrietatik eta bestelako testuetatik bil daiteke. Testu horien formatua testu laua da, inongo informazio gehigarririk gabe. Horrek esan nahi du testu horiek ez dutela RSTko inongo informaziorik ezta bestelako informazio linguistikorik ere.

Hori dela eta, RSTko behatzaile eta etiketatzaileek testu horiek RST informazioarekin aberastu behar dituzte. Prozesu hori testua segmentuetan banatzea (EDU), segmentuen artean *span*ak eratzea eta horien artean erlazioak jartzean datza. Jakina, beharrezkoa da etiketatzaileak hizkuntzaren eta RSTren ezagutza handia izatea kalitatezko analisiak lortzeko. Horregatik, normalean RST zuhaitz-diagramak hizkuntzalariek egiten dituzte.

RST Euskal Treebanka [Iruskieta et al., 2013] izan da lehenengoetariko euskarazko RST corpusa osatzen. Corpusa [7] hainbat arloetako testuez osatuta dago: medikuntza, liburuen iritziak, egunkari idatzien berriak, terminologia eta artikulu zientifikoak. Gainera, testu horiek guztiak 2021eko *DISRPT Shared-Task*ean daude eskura [8], beste testu gehiagorekin.

RST informazioa duten corpuseko testuak *rs3* formatuan daude gordeak. Formatu hori *xml* formatuan oinarritzen da. Gordetzen den informazioa bi ataletan banatzen da, burukoa (*header*) eta gorputza (*body*). Burukoan, (*relations* etiketarekin) RST erlazioen zerrenda aurkitzen da, eta gorputzean honako informazio hau: *i*) testuaren EDUak, *ii*) EDUek satelite ala nukleo funtzioa duten eta zein koherentzia erlazio duten, eta *iii*) EDUak haien

[5] SFUren RST web orrialdea, ingelesez, euskaraz, gazteleraz, portuguesaz eta frantsezaz: https://www.sfu.ca/rst/index.html

[6] RST sakontzeko bibliografia: https://www.sfu.ca/rst/05bibliographies/bibs/RST_bibliography.pdf

[7] Euskal RST Treebankaren corpusa hemen atzigarri: https://ixa2.si.ehu.eus/diskurtsoa/

[8] https://github.com/disrpt/sharedtask2021 euskararen atalean.

artean elkartuta sortzen diren *span*ak. Adibide gisa, 3.2 Irudian agertzen den RST zuhaitz-diagramaren *rs3* formatua 3.3 Zerrendan ikus daiteke.

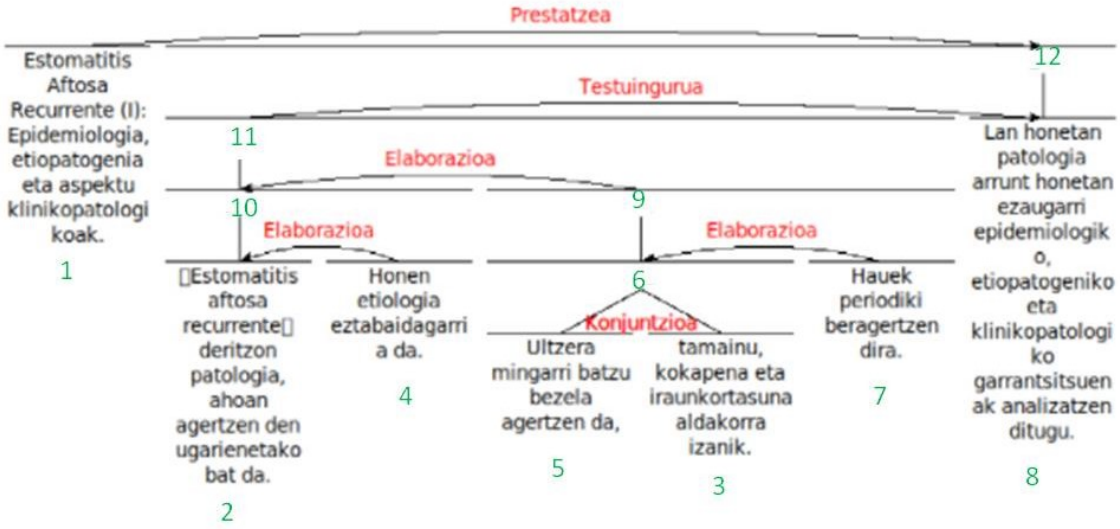
3.3 Zerrenda: GMB0301 testua RST informazioarekin, corpuseko formatuan (rs3)

```
<rst>
  <header>
    <relations>
      <rel name="elaborazioa" type="rst" />
      <rel name="prestatzea" type="rst" />
      <rel name="testuingurua" type="rst" />
      <rel name="konjuntzioa" type="multinuc" />
    </relations>
  </header>
  <body>
    <segment id="1" parent="12" relname="prestatzea">Estomatitis Aftosa Recurrente (I):
     Epidemiologia, etiopatogenia eta aspektu klinikopatologikoak.</segment>
    <segment id="2" parent="10" relname="span"> ‘‘Estomatitis aftosa ’’recurrente deritzon
     patologia, ahoan agertzen den ugarienetako bat da.</segment>
    <segment id="4" parent="2" relname="elaborazioa"> Honen etiologia eztabaidagarria da.</
     segment>
    <segment id="5" parent="6" relname="konjuntzioa"> Ultzera mingarri batzu bezela agertzen da,<
     /segment>
    <segment id="3" parent="6" relname="konjuntzioa"> tamainu, kokapena eta iraunkortasuna
     aldakorra izanik.</segment>
    <segment id="7" parent="6" relname="elaborazioa"> Hauek periodiki beragertzen dira.</segment>
    <segment id="8" parent="12" relname="span"> Lan honetan patologia arrunt honetan ezaugarri
     epidemiologiko, etiopatogeniko eta klinikopatologiko garrantsitsuenak analizatzen ditugu.</
     segment>
<group id="6" type="multinuc" parent="9" relname="span" />
<group id="9" type="span" parent="10" relname="elaborazioa" />
<group id="10" type="span" parent="11" relname="span" />
<group id="11" type="span" parent="8" relname="testuingurua" />
<group id="12" type="span" />
  </body>
</rst>
```

Adibide honetan, sinplifikatzeko, burukoa (*header*) atalean, bakarrik testuan erabiltzen diren RST erlazioak agertzen dira *relations* etiketaren barnean, baina normalean RSTko informazioa duten testuak eraikitzeko erabiliko diren erlazio guztiak aipatzen dira. *Body* etiketaren barnean, bi atal bereizten dira: *segment*, EDUei dagokiena, eta *group*, *span*ei dagokiena. Biek parametro berak dituzte, *id*, *parent* eta *relname*, baina *group*ek bat gehiago du: *type*. Hori, *span*a osatzen duen beste EDUa ala *span*a RSTko erlazio nukleobakar ala nukleoaniztun batekin lotuta dagoen adierazteko erabiltzen da.

Aipatu diren gainerako parametroek (*id*, *parent* eta *relname*) hau adierazten dute hurrenez hurren:

- Identifikadorea (***Id***): EDUak eta *span*ak identifikatzeko erabiltzen dira eta ez dute zertan ordenan egon behar.
- Gurasoa (***Parent***): zein *span*en barnean dagoen adierazten du. Horretarako, *parent*ek *span* horren *id*a izan behar du.
- Erlazioa ***Relname***k EDUaren ala *span*aren nukleartasuna adierazten du. "*span*"hitza izatekotan bertan, nukleoa da, bestela, RST erlazio horren satelitea izango da.
- Root: *Parent* atributurik ez duen *group* etiketa *Root*a izango da, hau da, maila goreneko *span*a.



3.3 Irudia: GMB0301 testuko rs3 fitxategiaren RST zuhaitz-diagrama.

3.3 Zerrendan agertzen den *rs3* formatuan adierazita dagoen RST zuhaitza 3.3 Irudian era bisualean ikus daiteke. Agertzen diren zenbaki berdeak *rs3* fitxategian agertzen diren EDUen eta *span*en *id*ak adierazten dituzte. *rs3* formatutik zuhaitz-diagrama lor daiteke, eta, era berean, zuhaitz-diagramatik *rs3* formatua ere lor daiteke.

i) Zuhaitz-diagramatik *rs3* formatuko fitxategira pasatzeko prozesua.

Zuhaitz-diagrama batetik abiatuta *rs3* formatudun fitxategia sortzeko lehenengo pausua zuhaitz-diagraman ageri diren erlazio guztiak burukoan (*relations* etiketaren barnean) jartzea da. Ondoren, *body* etiketan EDUetarako *segment*ak sortzen dira, EDUko testuekin eta

id bat esleitzen zaie, errepikatuta ez dagoena. *Span*entzako gauza bera egingo da, baina *group* etiketa erabiliz, eta *root span*a sortzen da, zuhaitzeko unitate gorena.

Parent atributuak jartzeko, 3.3 Irudian agertzen diren EDUen eta *span*en *id* zenbaki berak erabiliko ditugu. Irudia jarraituz, zuhaitz-diagraman azpitik hasita, EDU_2, bere gainean agertzen den *span*aren *id*a (10) izango da *parent* atributua (ikusi fitxategiko 12. lerroa), *span* hori osatzen duten azpiko mailako EDUen nukleoa izan behar baita beti. Irudia jarraituta, EDU hori beste EDU_4rekin erlazionatuta dago *Elaborazioa* erlazio batekin, beraz satelite den EDUaren *parent* atributuak nukleoa den EDUaren *id*a (2) izango du (ikusi 13. lerroa), eta satelitea denez *relation* atributuan erlazio hori finkatzen da.

*Span*ekin gauza bera gertatzen da, irudiko $span_{10}$ aztertuta, 10 *parent* balioa izango du $span_9$k, biek erlazionatuta baitaude eta satelitea baita (ikusi 19. lerroa). Gainera, $span_{10}$ *parent* balio gisa 11 izango du, $span_{11}$ osatzen duen bi *span*en artean nukleoa delako (ikusi 20. lerroa).

Nukleoaniztunen kasuan, osatzen duten EDU_3 eta EDU_5 nukleoak direnez, biek bere gainekoo *span*aren *id*a izango dute *parent* balio gisa, eta biek *relname* atributuan erlazio nukleoaniztuaren izena izango dute (ikusi 15. eta 16. lerroak). Azkenean, EDU eta *span* guztiei *id*ak jarrita, dagozkien *parent*ekin lotuta eta nukleoak eta sateliteat definituta, edozein RST zuhaitz-diagramaren *rs3* fitxategia sor daiteke.

ii) *Rs3* formatuko fitxategitik zuhaitz-diagrama lortzeko prozesua.

3.3 Zerrendako *rs3* fitxategitik 3.3 Irudian agertzen den RST zuhaitz-diagrama lortzeko, zuhaitzaren egitura goitik behera sortu behar da. 22. lerroko *root span*etik abiatuta, bere *id*a (12) *parent* gisa zeintzuk duten begiratzea izango da lehenengo pausua. Hauek 11. eta 17. lerroan agertzen diren EDUak dira, eta irudian kokatzen baditugu, EDU_1 eta EDU_8 dira. Lehenengoa satelitea da *Prestatzea* erlazioa duelako eta $span_{12}$ bere nukleoa izango da. Ordea, beste EDUa (EDU_8), $span_{12}$ osatzen duen EDU nukleoa da, eta 8 *parent* gisa duen *span*a $span_{11}$ satelitea da, *Testuingurua* erlazioarekin (21. lerroa). Honen *id*a 11 da, eta 20. lerroko $span_{10}$ 11 du *parent* gisa, beraz $span_{11}$ osatzen duten nukleoa da. Satelitea aurkitzeko, $span_{10}$aren *id*a *parent* gisa duten EDUa edota *span*a aztertu behar dira. Kasu honetan, 12. lerroko EDU_2 eta 19. lerroko $span_9$ dira. Azken honek erlazioa duenez, $span_{11}$ren satelitea da, *Elaborazioa* erlazioarekin lotuz.

Erlazio nukleoaniztunak irudikatzeko, *type* atributuan *multinuc* agertzen denean, bere *id*a parent gisa duten EDU ala *span*ak aztertu behar dira. Kasu honetan, 18. lerroko $span_6$k

betetzen du. Hemendik abiatuta, 6 *parent*a duten hiru EDU ditugu (14, 15 eta 16. lerrokoak, EDU_5, EDU_3 eta EDU_7 hain zuzen ere). $span_6$ nukleoa da baina kasu honetan 3 EDUek dituzte erlazioa *relname* atributuan. Horretarako, *relations* etiketan erlazioen *type* atributua ea *rst* ala *multinuc* motatakoa diren ikusi behar da. *Konjuntzioa* erlazioa *multinuc* denez, horiek izango dira $span_6$ osatzen duten EDU_5 eta EDU_3, erlazio horrekin. EDU_7, aldiz, $span_6$ren satelitea izango da, bere *relname* atributuan dagoen *Elaborazioa* erlazioarekin lotuta.

Laburbilduz, RST zuhaitz-diagrama sortzeko, edozein EDU edo *span* hartuta, bere *id*a *parent* gisa duten EDU ala *span* aztertzea izango da, zuhaitzean kokatzeko eta hura eratzen joateko, eta *type* aztertu erlazio nukleoaniztunak identifikatzeko.

RSTTool

RSTTool tresnak [O'Donnell, 2000] *rs3* formatuan dauden fitxategien zuhaitz-diagramak irudikatzen ditu. Gainera, RST zuhaitz-diagramak etiketatzeko eta *rs3* formatuan gordetzeko aukera ematen du. Hori dela eta, RSTrekin lan egiteko oso erreminta ahaltsua da.

Programaren dokumentazioa eta tresna bera eskura daude.[9] Programa hori erabiltzeko tutorialak ere badaude.[10] [11] Programa horren interfaze grafikoa 3.4 Irudian ikus daiteke.

3.3.2 Sare neuronalak

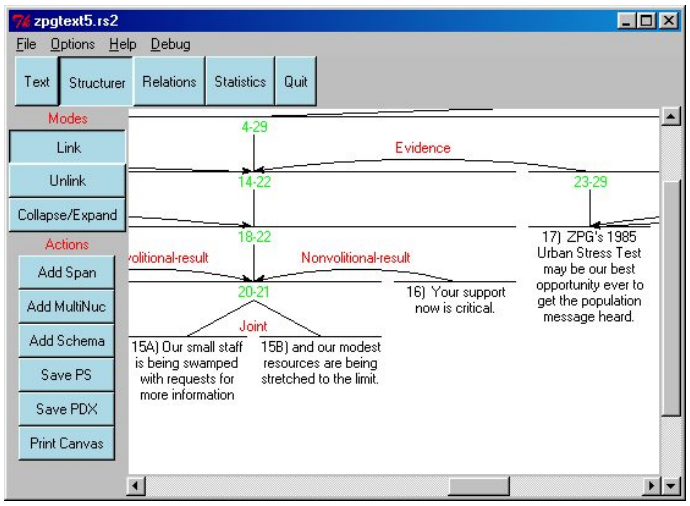
GrALean sare neuronalak erabiliko dira RST erlazioak iragartzeko. Sare neuronalen bidez, corpuseko RST zuhaitzak ikasi egingo dira. Ikasketa-prozesua bukatzean, sare neuronalari RST erlazioak eta zuhaitzak iragartzeko eska dakioke. Hau hobeto ulertzeko, lehenik eta behin sare neuronalen egitura eta funtzionamendua azalduko da, sare neuronalen historian sortutako lehenengo eredutik abiatuta.

Giza burmuina ezagutzen den sistema ahaltsu adimentsuena da. Gainera, oraindik ez da bere funtzionamendua zehatz-mehatz ezagutzen. Horregatik, gizakiak hainbat ikerketa egin ditu bere historian zehar burmuina hobeto ulertzeko. Burmuina neuronaz osatuta dago, baina neurona era konputazionalean eta matematikoan aurkeztu zen lehenengo ikerketa [McCulloch and Pitts, 1943] izan zen.

[9] RSTTool programa: http://www.wagsoft.com/RSTTool/index.html

[10] Programaren erabilpen tutoriala, 1. zatia, portugesez: https://www.youtube.com/watch?v=YWW4WG4NYTY

[11] Programaren erabilpen tutoriala, 2. zatia, portugesez: https://www.youtube.com/watch?v=g33qjTK0PEc



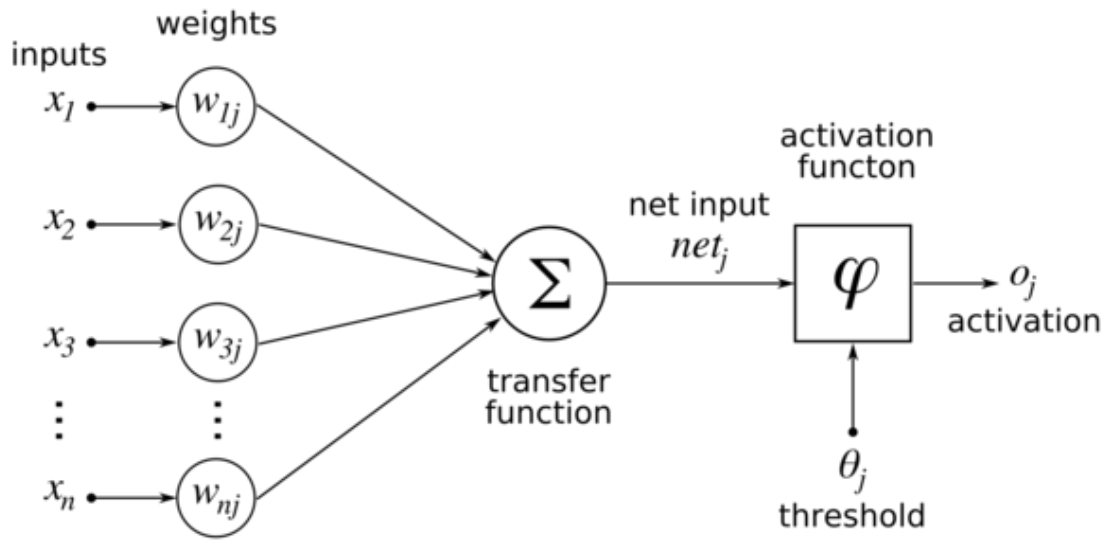
3.4 Irudia: RSTTools programaren interfazea, segmentazioa egin ondoren RST zuhaitza eraikitzen.

Proposatutako modelo hori (gaur egun McCulloch-Pitt bezala ezagutzen dena) neurona bakar batez osatuta dago. Ondorengo urteetan, neurona gehiagoz osatutako modeloak (sare neuronalak) garatu dira [Schmidhuber, 2015]. Baina ez da izan azken urteotara arte sare neuronalek ikaragarrizko bultzada izan dutela, hainbat ikerketatik lortu diren sistema arrakastatsuak lortu direlako [Dargan et al., 2019], batez ere konputazio aukera eta abiadura handiagoa dagoelako.

McCulloch-Pitt proposatutako modelora bueltatuz, neurona modeloa 3.5 Irudian ikus daiteke. Sarrerako balio batzuk sartzen dira neuronara, eta balio horiek aurredefinitutako pisuekin biderkatzen dira. Ondoren, balio guztien gehiketa egiten da, eta emaitza aktibazio funtzioan ebaluatzen da. Funtzio honen sarrera balioa atalase bat gainditzen badu, bere irteera 1 izango da, kontrako kasuan 0. Balio hori izango da neuronaren emaitza, funtzio inhibitzailea 0 bada. Bere formula matematikoa 3.1 Ekuazioan ikus daiteke, ondorengo aldagaiak kontuan izanda: y: irteera, x_i: sarrera, θ: aktibazio funtzioaren atalasea, w_i pisuak eta z_j funtzio inhibitzailea.

$$y = \begin{cases} 1, & \text{if } \sum_i w_i x_i \geq \theta \wedge z_j = 0, \forall j \\ 0, & \text{bestela} \end{cases} \tag{3.1}$$

Neurona modelo honetatik abiatuta, pertzeptroia definitu zen [Rosenblatt, 1958], eta dagoeneko neurona artifizial bezala hartu da. McCulloch-Pitt modeloarekin alderatuta, inhibitzaile funtzioa desagertzen da, eta pisuak aurredefinitutako balioak izan ordez, ajusta-



3.5 Irudia: McCulloch-Pitt proposatutako neurona artifizialaren funtzionamendua.

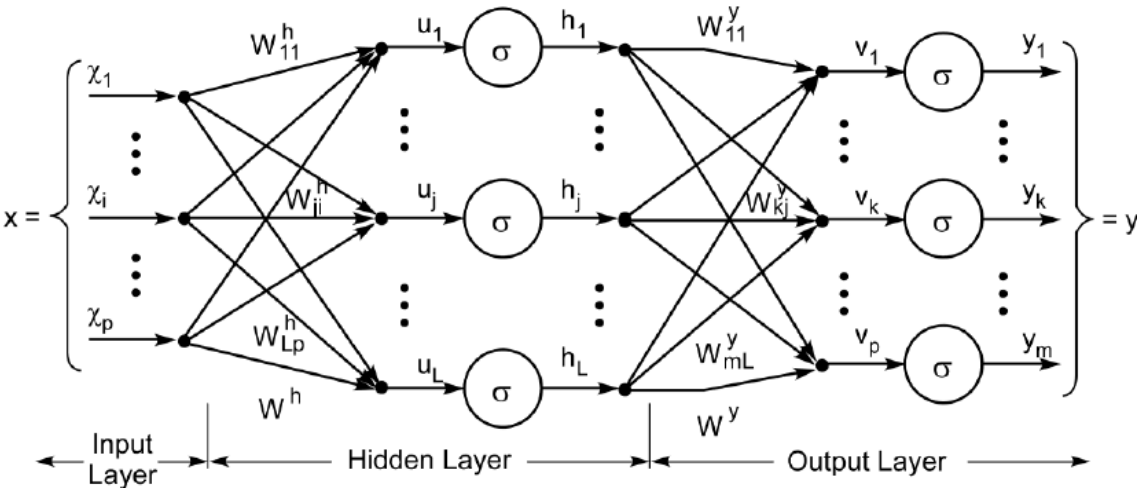
Iturria: [Kusmartsev and Kusmartsev, 2016]

tu daitezke. Neurona bakoitzaren pisua neurona horren influentzia adieraziko du. Gainera, pisuekin biderkatzeaz gain, *bias* deitzen den balio bat batu daiteke neuronako balioa handitzeko. Beste alde batetik, neuronaren aktibazio funtzioak ez du zertan linearra izan behar. Honen adibidea *sigmoide* funtzioa da. *Sigmoide*aren funtzioa: $f(x) = \frac{1}{1+e^{-x}}$. Aktibazio funtzio honetaz gain gehiago erabiltzen dira [Nwankpa et al., 2018].

Pertzeptroiak funtzio lineal bat ikasteko gai dira, pisuak moldatuz. Jasotako sarrera bati aktibazio funtzioaren irteeraren arabera (demagun 1 ala 0 balioa) eta guk sarrera hori erlazionatzen dugun balioaren arabera (demagun 1 edo 0), pisuak moldatuko ditu klasifikazioa ondo egiteko, momentu honetan ikasitako modeloa (funtzio lineala) gordez. Hala ere, funtzio lineal batekin batzutan ez da nahikoa sailkapenak egiteko.

Horretarako, Multi-Layer Perceptron (MLP) modeloa erabiltzen da. Hau neurona gehiagoz osatuta dago (sare neuronal bat eratuz), eta ondorioz funtzio konplexuagoak ikasteko gai da. Neuronak geruzaka antolatuta daude, eta geruza desberdinen artean konektatzen dira. Neurona bakoitzaren egoera aktibazio funtzio (σ) baten bitartez adieraziko da, irteera neuronarekin bezala. Geruzei *layers* ere deitzen zaie. Irteera geruza eta sarrera geruza artean dauden geruzei ezkutatutako geruzak deitzen zaie, *hidden layers* ingelesez. MLP baten eskema eta bere geruzak 3.6 Irudian ikus daitezke. Kasu honetan, hainbat irteera neurona daude, eta barneko ezkutatutako geruza bakarra du.

MLP sare neuronalak normalean sailkapen edo iragartzeko atazetarako erabiltzen dira.



3.6 Irudia: Multi-Layer Perceptronaren eskema.

Iturria: [Faghfouri and Frish, 2011]

Adibide sinple bat: txakurraren datuetatik bere arraza iragartzea. Eskuratzen den datu basetik, txakur bakoitzeko bere arraza, tamaina, buztanaren luzera, kolorea, belarrien tamaina, zaunka kopurua egunean zehar, eta abar atributuak jaso. Kualitatiboki badaude datuak, bakoitzari balio bat esleitzen bazaio. Atributu adina sarrera neurona egongo dira sarrera geruzan, eta irteera geruzan arraza adina. Horrela, datuetatik gure sare neuronalak neuronen pisuak eta *bias* balioak ikasiko ditu (entrenamendu fasea deritzo ere) atributuak arrazekin erlazionatzeko, hau da, funtzio bat. Behin entrenamendu prozesua amaituta, beste txakur berri baten atributuak pasata, txakur horri ikasitako modeloaren neurona pisuak erabilita, arraza bat esleituko dio.

MLPetaz gain, beste arkitektura, helburu eta funtzio desberdinak dituzten sare neuronalak sortu egin izan dira [Jain et al., 1996]. Hiru ezkutatutako geruza edo gehiago dituzten sare neuronalei sare neuronal sakonak deitzen zaie ere, ingelesez *deep neural network*. Sare hauek ikasteko prozesuari *Deep Learning* deitzen zaio, hau da, ikasketa sakona. Sare neuronalak eta zehazki *Deep Learning* sakonki azaltzen dituen erreferentzia bibliografikoa honakoa da: [Goodfellow et al., 2016].

Ikasketa sakoneko sare neuronalak bultzada handia izan dute azken urteotan, eta hainbat arkitektura berri sortu dira, helburu edo ataza konkretuen ebazpena hobetzeko [Wang and Raj, 2017]. Adibidez, sare konboluzionalak (CNN) irudiak sailkatzeko erabiltzen dira normalean. Sare neuronal errekurrenteak, aldiz, datu sekuentzialetan iragarpenak egiteko erabiltzen dira. Sare moten adibideak, zehaztapenak eta horien paper originalak

dokumentuetan [12] aurki daitezke.

Long Short-Term Memory sare neuronal errekurrenteak

Long Short-Term Memory neural networks (LSTM) [Hochreiter and Schmidhuber, 1997] sare neuronal errekurrenteetan (RNN) [Elman, 1990] oinarritzen da. RNNetako ezkutatutako geruzetako neuronak errekurrenteak dira, hau da, ezkutatutako geruzen neuronetako balioak irteera neuronalera pasatzeaz gain, neurona berera bueltatzen dira input gisa hurrengo iterazioan, beste sarrera datu batzuk sartzean. Honen errepresentazio matematikoa 3.2 Ekuazioan ikus daiteke. h ezkutatutako geruzetako neuronak adierazten ditu, y sare neuronalaren irteera, t denbora unea da, X sarrera balioak, w_h ezkutatutako geruzen neuronen pisuak eta w_r ezkutatutako geruzen neuronen pisu errekurrenteak.

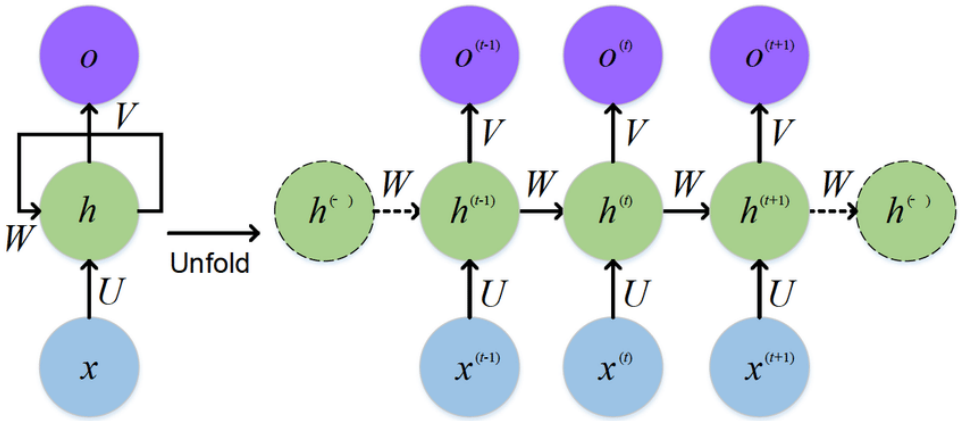
$$\begin{aligned} h_t &= \sigma(W_h X + W_r h^{t-1}) \\ y &= \sigma(W_y h^t) \end{aligned} \tag{3.2}$$

RNN sarearen erabileraren adibide sinple bat eguraldi iragarpena da. Asteleheneko eguraldia jakinda, asteartekoa iragartzen saiatuko da, baina asteartea pasata, bai asteleheneko eta astearteko eguraldia kontuan hartuko ditu asteazkeneko eguraldi iragarpena emateko. Hizkuntzaren prozesamenduaren kasuan, esaldi baten hurrengo hitza iragartzea esaldiko aurreko hitzak jakinda izan daiteke aplikazio bat. Horrenbestez, RNNetan sekuentzia baten denbora une batean iragartzeko, sekuentziako aurreko denbora uneak kontuan hartzen dira.

Iterazio bakoitza, informazioa sarrera neuronetara heltzen den t unea izango da. Iterazioak daudenez, RNNak denbora unitate bakoitzeko egoeran adieraz daitezke zabalduta (*unfolded*), bere arkitektura eran adierazi ordez tolestuta (*folded)*. Ikusi 3.7 Irudia. Irudian agertzen den h, *hidden layer*a, neurona batez ala askoz eratuta egon daiteke.

Edozein kasutan, RNNen sekuentziak ez dute zertan norabide bakarrekoak izan behar. Bidirekzionalak izan daitezke [Schuster and Paliwal, 1997], *Bidirectional* RNN deiturikoak (BRNN). RNNren arkitektura bera du. RNNko ezkutatutako geruza mantentzen da *forward layer* izenarekin, baina beste ezkutatutako geruza bat sartzen da, kontrako noranzkoan, *backward layer* deiturikoa. Honek esan nahi du t unean, *forward layer*en $t-1$ uneko informazioa jasotzen duela, eta *backward layer*n, aldiz, $t+1$ uneko informazioa.

[12] Web honetan sare neuronal bakoitzaren diagramak, zehaztapenak eta paper originalak ikus daitezke: https://www.asimovinstitute.org/neural-network-zoo/



3.7 Irudia: RNN baten arkitektura, folded ezkerrean eta unfolded eskuinean.

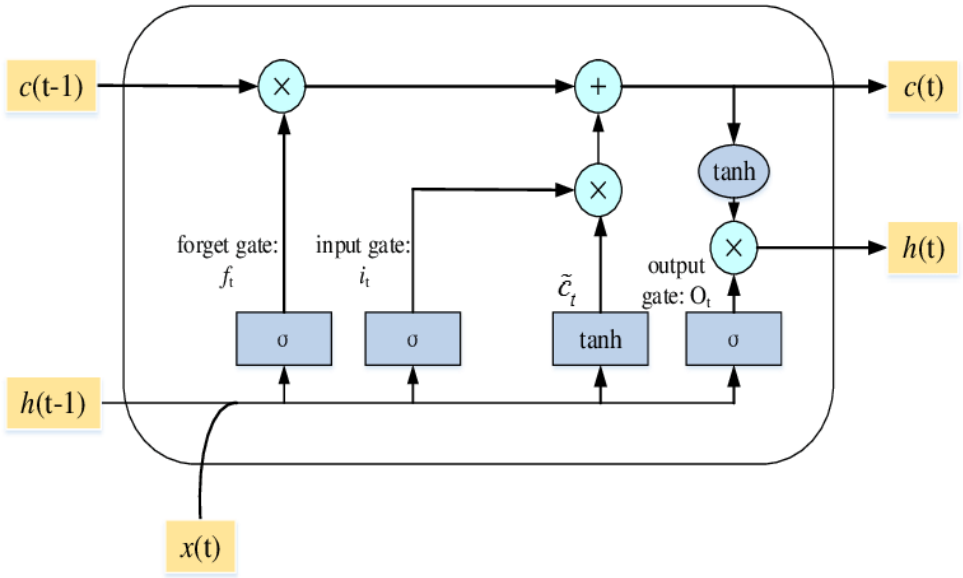
Iturria: [Feng et al., 2017]

Sarrerako neuronak bi geruzekin konektatuta daude, eta bi geruzen emaitzak multzokatzen dira irteera neuronekin, bakoitza bere pisuekin biderkatuta.

LSTM, berez, RNN bat da, baina dituen neuronak memoria erabiltzen dute: LSTM unitateak. LSTMaren ideia unitateak garrantzitsua dena ikastea da eta zer ez hartu kontuan, eta ikasitakotik zer ahaztu. LSTM unitatea 3.8 Irudian ikus daiteke. Memoria unitate honek epe laburreko egoera (*short term*) h eta epe luzeko egoera (*long term*) c izango ditu, irudiko beheko eta goiko fluxua, hurrenez hurren. Epe luzeko egoeran, informazioa t unetik $t+1$ unera pasa daiteke ia aldatu gabe, horregatik bere izena. Epe laburreko egoeran, ordea, $t-1$ uneko informazioa x sarrerako t unearekin kontuan hartzen da.

Epe luzeko egoeran informazioa gehitzeko edota kentzeko, ateak (*gates*) deritzen estrukturen bidez egin behar da. Ateak *sigmoide layer* baten bidez osatuta daude, eta emaitza epe luzeko egoerarekin biderkatuko da puntuz puntuz. Hori dela eta, *sigmoide*aren irteeraren arabera, zenbat informazio pasako den adieraziko da, 1 informazio guztia pasa eta 0 informaziorik ez pasa. LSTMk hiru ate ditu, irudian *sigmoide* (σ) biderketa operazioarekin jarraian ikus daitezke. Bi ate epe luzeko egoeraren fluxua kontrolatzeko dira, eta bestea epe laburreko fluxua.

LSTMk t une bakoitzeko egiten dituen pausuak 3.9 Irudian ikus daitezke. Hasteko, epe luzeko egoeran zer informazio mantenduko ala ezabatuko den erabakitzen da. Ahazteko ate (*Forget gate*) f_t bat arduratzen da horretaz. h_{t-1} eta x_t aztertuta, ateak 0 (ahaztu) edo 1 (mantendu) bueltatuko du epe luzeko egoeran dagoen atributu bakoitzeko. Fase hau eta bere errepresentazio matematikoa 3.9a Irudian ikus daiteke.



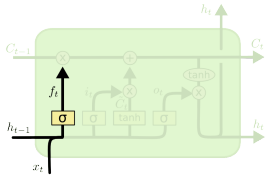
3.8 Irudia: LSTM unitatearen eskema, t unean.

Iturria: [Yuan et al., 2019]

Ondoren, epe luzeko egoeran zer informazio berri gehituko den erabakiko da. Alde batetik, sarrera ate i_t batek erabakiko du zein atributu eguneratuko diren, ateak 0 (ez eguneratu) edo 1 (eguneratu) itzuliz atributu bakoitzeko. Beste alde batetik, $\tilde{C}_t$ *tanh* geruza baten bidez eguneratuko diren atributuen balioak izango ditu. Ikusi formula 3.9b Irudian.

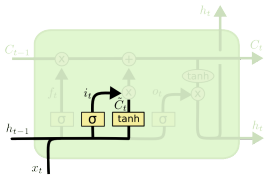
Jarraitzeko, epe luzeko egoera c_{t-1} eguneratuko c_t da, 3.9c Irudiko formula jarraituz. Hau da, aurreko pausuetan erabakitako ezabatuko den informazioa ezabatuko da, eta erabakitako gehituko den informazioa gehituko da. c_t epe luzeko egoeraren atributu balioak $t+1$ unean erabiliko dira LSTM unitatearen epe luzeko egoeraren sarrera gisa.

Amaitzeko, LSTM unitatearen irteera erabaki behar da, bai irteera geruzerako baita t uneko epe laburrerako egoera bezala, h_t. Hau $t+1$ unean LSTM unitatearen epe laburreko egoeraren sarrera izango da. Irteera balio hau lehendabizi irteera ate (*output layer*) o_t batetik pasako da, zein atributu balio irtengo diren jakiteko. Ondoren, epe luzeko egoerari *tanh* aplikatuko zaio, balioak $[-1,1]$ tartean egoteko. Bukatzeko, o_t atetik irten denarekin biderkatuko da, emaitza h_t irteera izanik. Formula matematikoa 3.9d Irudian azter daiteke.



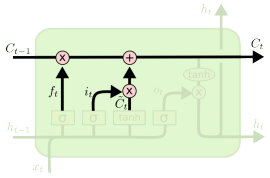
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

(a) Epe luzeko egoeran mantendu eta ezabatuko den informazioa erabaki



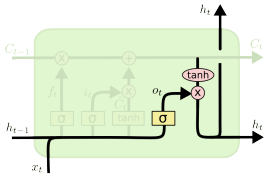
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

(b) Epe luzeko egoeran gehituko den informazio berria erabaki



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

(c) Epe luzeko egoera berritu t unerako



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

(d) LSTM unitatearen irteera erabaki

3.9 Irudia: LSTM unitate baten faseak, t unerako

Iturria: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

3.3.3 Trantsizioetan oinarritutako RST parserra orakulo dinamikoarekin

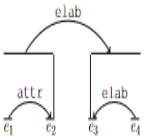
Parser baten ataza hizkuntza baten esaldi bakoitzari esanahia ematea da [Frazier, 1979]. Beste era batean esanda, esaldi bakoitza deskribatzea da helburua. Esaldi batek subjektua eta predikatua du, eta bakoitza elementu txikiagotan banatzen da. Esaldien deskribapen hau zuhaitz egituren bidez adierazten da.

Gure kasuan, parserra erabiliko da EDUetatik erlazioak eta nukleartasuna iragartzeko, lortzen den zuhaitza RST-zuhaitz bat izanik. Ataza honetarako RST parserrak egin dira [Yu et al., 2018].

RST parserra trantsizioetan oinarritutakoa da. Trantsizioetan oinarritutako parserrek egitura proiektiboetan funtzionatzen dute bakarrik. RST zuhaitzak proiektiboak dira, EDU eta *span* guztietara bidea baitago burutik (*root*etik) hasita.

RST parserrak desplazatu-laburtu (*shift-reduce*) sistema trantsizionala erabiltzen du ere [Sagae and Lavie, 2005]. Trantsizio sistemak bi datu-egitura ditu: pila (*stack*) bat eta ilara (*queue*) bat. Hauek konfigurazioa eratzen dute. Pilak RST zuhaitz partzialak izango ditu eta ilarak aztertu gabeko EDUak. Hasieran, pila hutsik egongo da eta ilara dokumentu osoa izango da. RST parserrak iteratiboki konfigurazioan ekintzak egikarituko ditu, konfigurazio berriak sortuz, azkenean amaitu egoerara ailegatu arte, hots, ilara hutsik egongo da eta pilak elementu bakarra izango du, RST zuhaitzaren *root*a. Ekintzak bi motatakoak izan daitezke:

- **Desplazatu (SH):** Ilaratik EDU bat despilaratu (*pop*) eta pilan pilaratu (*push*), nodo bakarreko azpizuhaitza bihurtuz.
- **Laburtu (RD):** Pilako lehenengo bi azpizuhaitzak azpizuhaitz bakar batean konbinatzen ditu RST diskurtso erlazio batekin. Erlazioan SN, NS edo NN agertuko da, azpizuhaitz bakoitza satelitea (S) ala nukleoa (N) den adieraziz.

Pausua	Pila	Ilara	Ekintza	Erlazioak	Zuhaitza
1	$\emptyset$	e_1, e_2, e_3, e_4	SH	$\emptyset$	
2	e_1	e_2, e_3, e_4	SH	$\emptyset$	
3	e_1, e_2	e_3, e_4	RD(attr,SN)	$\emptyset$	
4	$e_{1:2}$	e_3, e_4	SH	$\widehat{e_1 \boldsymbol{e_2}}$	
5	$e_{1:2}, e_3$	e_4	SH	$\widehat{e_1 \boldsymbol{e_2}}$	
6	$e_{1:2}, e_3, e_4$	$\emptyset$	RD(elab,NS)	$\widehat{e_1 \boldsymbol{e_2}}$	
7	$e_{1:2}, e_{3:4}$	$\emptyset$	RD(elab,SN)	$\widehat{e_1 \boldsymbol{e_2}}, \widehat{\boldsymbol{e_3} e_4}$	
8	$e_{1:4}$	$\emptyset$	AMAITU	$\widehat{e_1 \boldsymbol{e_2}}, \widehat{\boldsymbol{e_3} e_4}, e_{1:2}\widehat{\mathbf{e_{3:4}}}$	

3.1 Taula: Trantsizio bidezko sistemaren adibidea RST diskurtso parserrarekin

Iturria: [Yu et al., 2018]

3.1 Taulan adibide bat ikus daiteke. Taulako ekintzak jarraituz, taulako eskuineko RST zuhaitz-diagrama da sortu dena. Pausu bakoitzean, ilararen eta pilaren egoerak ikus daitezke, hau da, konfigurazioak. 1. pausuan pila hutsik dago, beraz, pilan sartzen dugu ilarako lehenengo EDUa azpizuhaitz bezala. 2. pausuan, ekintza bera egiten da. 3. pausuan, laburketa egiten da pilako lehenengo bi azpizuhaitzekin, eta azpizuhaitz berri bat sortzen da *attr* erlazioarekin, *span* bat osatuz, eta azpizuhaitz berriko ezkerreko hostoa satelitea izanik eta eskuinekoa nukleoa. Beraz, 4. pausuan pilan azpizuhaitza sortu dela ikus dezakegu, eta erlazio zutabean satelitea eta nukleoa. 4. eta 5. pausuan, ilarako bi EDUak pilan sartzen dira, eta 6.ean bi horiek (lehenengo biak baitira) erlazionatzen dira berriro, laburketa ekintza exekutatu delako. Beste azpizuhaitz berri bat sortzen dute, *elab* erlazioarekin eta lehenengoa nukleoa eta bigarrena satelitea izanik. Azkenik, 7. pausuan pilan geratzen diren bi azpizuhaitzak erlazionatzen dira *elab* erlazioarekin, beste *span* bat eratuz. *Span* hau izango da zuhaitzaren *roota*; izan ere, pilan azpizuhaitz bakarra geratzen delako eta ilara hutsik dagoelako. Bukatzeko, 8. pausuan amaitze pausua gertatzen da, non taulako eskuineko zuhaitz diagrama eraiki daitekeen.

Noski, pausuak beste orden batean exekutatu izan balira, beste zuhaitz bat lortu izango genuke. Azken finean, bi ekintza horiekin, EDU horiekin egin daitezkeen zuhaitz konbinazio guztiak sor daitezke. Hori dela eta, RST zuhaitzak sortzeko eta erlazioak iragartzeko tran-

tsizio bidezko sistema aproposa da. Hala ere, iragarri nahi den zuhaitza sortzeko ekintzak nolabait erabaki behar dira.

Horretarako, orakuloa erabiltzen da. Orakuloak zein ekintza eta noiz erabili erabakitzeaz arduratzen da. Egokiak diren RST zuhaitzetatik (hizkuntzalariek sortutakoak, adibidez) konfigurazioak lortzeko mapaketa bat egiten du. Hau da, zuhaitz horiek sortzeko jarraitu behar dituen ekintzak ikasten ditu.

Orakuloa estatikoa izan daiteke. Honek esan nahi du konfigurazio bakoitzeko, ekintza bat itzuliko duela. Orakulo mota hau determinista da, eta limitazioak dakartza. Ikasi ez dituen konfigurazioak gertatzen badira, arazoak izango ditu iragartzeko prozesuan, oker egiten badu hurrengo pausuan zailagoa izango baitu zuhaitz egoki bat eratzen.

Aldiz, orakulo dinamikoak konfigurazio bakoitzeko ekintzen zerrenda bat itzultzen du, ekintza bakoitza hura izateko zuhaitz zuzena sortzeko ekintzen parte probabilitatea eman-da. Beraz, nahiz eta ez diren beti zuhaitz berberak lortuko, okertzeko joera murriztuko da.

Orakuloak ekintzak erabakitzeko, ezarritako erregelen bidez jokatu dezake, edota sare neuronalak entrenatuz ekintzak erabakitzen ikas ditzan, RST zuhaitzetatik ikasiz.

GrAL honetan, testuen RST zuhaitzak iragartzeko, orakuloak ekintzak erabakiko ditu sare neuronalak erabilita.

4. KAPITULUA

Datuen azterketa eta aurreprozesaketa

Euskal RST Treebankeko corpusa eskuratuta, parserraren sare neuronala entrenatzeko eta ebaluatzeko beharrezkoak diren fitxategien formatuak aztertuko dira eta corpusa aurreprozesatuko da, behar diren formatuak lortzeko.

4.1 Tokenizazioa

Corpusaren aurreprozesaketaren lehenengo pausua EDUak tokenizatzea da. Honek esan nahi du EDUko testua segmentatu eta tokenizatu behar dela. Tokenak testuan agertzen diren hitzak, karaktereak (euro ikurra, emotikonoa, etab.), puntuazio-markak edo azpihitzak izan daitezke. Orokorrean, esaldia hitzetan eta puntuazio-marketan banatzea izango da. Testu-segmentuak edo EDUak perpaus independenteetan banatzea da.

Dena den, tokenizazioaren arauak eta irizpideak norberak ezartzen ditu egin behar duen atazaren arabera, era zuzen bakarra ez dagoelako. Adibidez, *"Bihotz-maiztasuna 54 tau/min da."* esaldian, argi dago *da* tokena dela, baina *bihotz-maiztasuna* tokenean erabaki behar da ea bakarra (hitz bera) ala bi diren: *bihotz* eta *maiztasuna*. Aldrebes ere gerta daiteke, *arnas maiztasun* hitzekin token bat era daiteke. *54 tau/min*ekin ere erabaki behar da ea zenbakiak eta neurriak banandu. Medikuntzan aritzekotan agian ez da interesgarria tokenetan banatzea informazioa ez galtzeko.

Tokenizazioa egiteko erregela bidezko sistemak eraiki ahal dira, baina tokenizatzeko ere-

duak ikasi daitezke ere. Corpuseko EDUak tokenizatzeko, *Universal Dependencies (udpipe)* [Straka and Strakova, 2017] tresna erabili da, euskaraz aritzeko eredua baliatuz [1].

4.1 Zerrenda: Tokenizazioa udpiperen bidez

```
1  1  Honen  hau DET _  Case=Gen|Definite=Def|Number=Sing  2  nmod  _  SpacesBefore=\s
2  2  etiologia  etiologia  NOUN  _  _  0  root  _  _
3  3  eztabaidagarria eztabaidagarri  ADJ _  Case=Abs|Definite=Def|Number=Sing  2  amod  _  _
4  4  da  izan  VERB  _  Aspect=Prog|Mood=Ind|Number[abs]=Sing|Person[abs]=3 2  cop _
       SpaceAfter=No
5  5  .  .  PUNCT  _  _  2  punct  _  SpacesAfter=\n
```

4.1 Zerrendan, 3.3 Irudian agertzen den EDU_4ko testuaren ("Honen etiologia eztabaidagarria da.") *udpipe* tresnak ematen duen emaitza da. Bigarren zutabean, tokenak agertzen dira. Beste zutabeetan bestelako datuak agertzen dira, beste hizkuntzaren prozesamenduko lanetarako baliagarriak izan daitezkeenak. Horien artean, hirugarren zutabeko lema (hitzaren forma basea, hiztegian agertzen dena) eta laugarren zutabeko *Part of Speech* (POS), hau da, hitzen kategoria (aditza, izena, puntuazio-marka, izenordaina, izenondoa, aditzondoa, determinatzailea), adibidez.

4.2 EDUak atributuekin aberastea

EDU tokenizatuez gain, parserrerako baliagarriak izan daitezkeen EDUekin erlazionatuta dauden atributuak lortuko dira, Ixa taldean garatutako tresnen bidez. EDUen tokenizazioa eta atributuak dituen formatua (*raw*) lortuko da, atributuak honakoak izanik:

- EDUaren lehenengo hiru tokenak (bakoitza atributu bat)
- EDUaren lehenengo hiru tokenen POSa (bakoitza atributu bat)
- EDUaren azken tokena
- EDUaren azken tokenaren informazioa morfosintaktikoa (POS)
- EDUaren luzera kualitatiboki:
  - *veryshort*: <=5

[1] Tresna honen dokumentazioa eskuragarri dago UDPIPE liburutegiaren dokumentazioan: https://ufal.mff.cuni.cz/udpipe/1

  - *short*: 6-15
  - *long*: 16-25
  - *verylong*: >=26

- EDUa osatzen duten sintagmetako gehienez hiru token eta horien POSa
- EDUaren kokapena testu osoarekiko (*first* lehenengoa bada, *first-middle* lehenengo erdialdean badago, *second-middle* bigarren erdialdean badago, eta *last* azkena bada)
- Zenbakiak agertzen diren ala ez
- Portzentaiak agertzen diren ala ez
- Datak agertzen diren ala ez
- Diru-kantitateak agertzen diren ala ez

3.3 Irudian agertzen den GMB0301 testua *raw* formatuan 4.2 Zerrendan ikus daiteke.

4.2 Zerrenda: GMB0301 testua tokenizatuta, .raw formatuan

```
1 word pos prefw1 prefw2 prefw3 prefp1 prefp2 prefp3 suffw1 suffp1 len headw1 headw2 headw3 headp1
      headp2 headp3 head position numb perc money date\n
2 estomatitis__aftosa__recurrente__-LRB-__i__-RRB-__:__epidemiologia__,
      __etiopatogenia__eta__aspektu__klinikopatologikoak__.  EDU Estomatitis Aftosa  Recurrente
      PROPN   PROPN   PROPN   .   PUNCT   short   Epidemiologia   <NA>    <NA>    NOUN    <NA>
      <NA>    IN-HEAD first   <NA>    <NA>    <NA>    <NA>
3 estomatitis__aftosa__recurrente__deritzon__patologia__,
      __ahoan__agertzen__den__ugarienetako__bat__da__. EDU Estomatitis aftosa  recurrente PROPN
          NOUN    NOUN    .   PUNCT   short   <NA>    <NA>    <NA>    <NA>    <NA>    <NA>    OUT-
      HEAD    first-middle    <NA>    <NA>    <NA>    <NA>
4 honen__etiologia__eztabaidagarria__da__.    EDU Honen   etiologia   eztabaidagarria DET NOUN
      ADJ .   PUNCT   veryshort   etiologia   <NA>    <NA>    NOUN    <NA>    <NA>    IN-HEAD
      first-middle    <NA>    <NA>    <NA>    <NA>
5 ultzera__mingarri__batzu__bezela__agertzen__da__,   EDU Ultzera mingarri    batzu   NOUN    ADJ
      DET ,   PUNCT   short   agertzen    <NA>    <NA>    VERB    <NA>    <NA>    IN-HEAD second-
      middle   <NA>    <NA>    <NA>    <NA>
6 tamainu__,__kokapena__eta__iraunkortasuna__aldakorra__izanik__. EDU tamainu ,   kokapena    NOUN
          PUNCT   NOUN    .   PUNCT   short   izanik  <NA>    <NA>    VERB    <NA>    <NA>    OUT-
      HEAD    second-middle   <NA>    <NA>    <NA>    <NA>
7 hauek__periodiki__beragertzen__dira__.  EDU Hauek   periodiki   beragertzen DET ADV VERB    .
      PUNCT   veryshort   beragertzen <NA>    <NA>    VERB    <NA>    <NA>    IN-HEAD end <NA>
      <NA>    <NA>    <NA>
8 lan__honetan__patologia__arrunt__honetan__ezaugarri__epidemiologiko__,
      __etiopatogeniko__eta__klinikopatologiko__garrantsitsuenak__analizatzen__ditugu__.    EDU
      Lan honetan patologia   NOUN    DET NOUN    .   PUNCT   short   analizatzen <NA>    <NA>
      VERB    <NA>    <NA>    IN-HEAD last    <NA>    <NA>    <NA>    <NA>
```

4.3 Entrenamendurako fitxategien formatuaren azterketa

Sistema entrenatzeko, corpuseko EDUen testuak erabili ordez, EDU tokenizatu eta aurreko atalean aipatutako atributuak erabiliko dira ikasketa hobetzeko. Beraz, corpuseko fitxategien hasierako formatutik (*rs3*) entrenatzeko formatura pasatzean (*tbk*), EDU tokenizatuen formatuan *raw* agertzen den edukia erabiliko da.

Entrenatzeko formatuan, zuhaitz egitura *xml* eran gordetzen da, eta zuhaitza era bitarrean adierazita egongo da, hots nodo bakoitzeko bi hosto gehienez. 4.3 Zerrendan *tbk* fitxategiaren deskribapen formala aurkeztuko da. Eskuineko geziak tabulazioa adierazten du. Ikus daitekeenez, etiketetan bi *span* edota EDU lotzen dituen erlazioak agertzen dira, nukleo-satelite ordenarekin nukleobakarrak badira. Etiketaren barnean, erlazioa osatzen duten EDUak edota *span*ak agertzen dira tokenizatuta *raw*eko edukia izanik. Horietakoren bat *span*a bada, *span*a osatzen duten *span* edota EDUak lotzen dituen erlazioaren etiketa irekiko da, barnean tabulatuta hurrengo *span*ak edo EDUak agertuz.

4.3 Zerrenda: Entrenatzeko formatuaren definizioa (tbk)

```
root ← <ROOT>
        → span
       </ROOT>

span ← <erlazioa>
        → edukia
        → edukia
       </erlazioa>

erlazioa ←  SNnukbak
          | NSnukbak
          | NNnukaniz

edukia ←  tokens
        | span

nukbak ← {edozein erlazio nukleobakarra}
nukaniz ← {edozein erlazio nukleoaniztuna}

tokens ← {EDUari dagokion testua tokenizatuta eta .raweko edukia}
```

3.3 Irudian agertzen den GMB0301 testua *tbk* formatuan 4.4 Zerrendan azter daiteke.

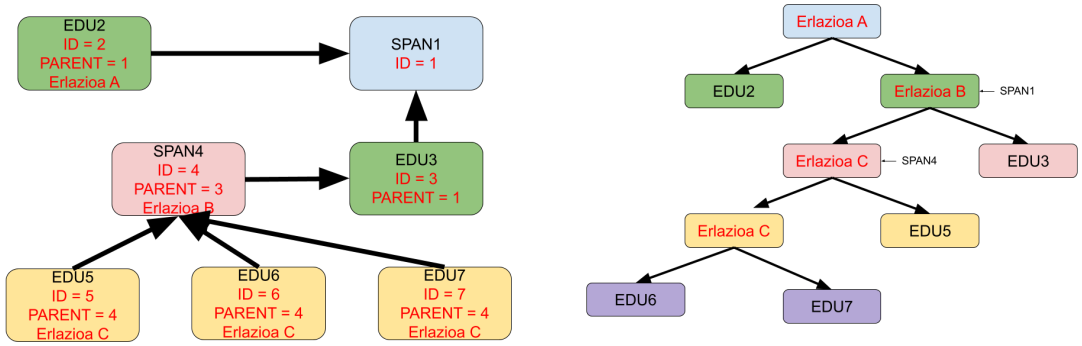
4.4 Zerrenda: GMB0301 testua RST informazioarekin, entrenatzeko formatuan (tbk)

```
<ROOT>
    <SNprestatzea>
        estomatitis__aftosa__recurrente__-LRB-__i__-RRB-__:__epidemiologia__,
    __etiopatogenia__eta__aspektu__klinikopatologikoak__.  EDU Estomatitis Aftosa  Recurrente
    PROPN   PROPN   PROPN   .   PUNCT   short   Epidemiologia   <NA>    <NA>    NOUN    <NA>
    <NA>    IN-HEAD first   <NA>    <NA>    <NA>    <NA>
        <SNtestuingurua>
            <NSelaborazioa>
                <NSelaborazioa>
                    estomatitis__aftosa__recurrente__deritzon__patologia__,
    __ahoan__agertzen__den__ugarienetako__bat__da__. EDU Estomatitis aftosa  recurrente  PROPN
       NOUN    NOUN    .   PUNCT   short   <NA>    <NA>    <NA>    <NA>    <NA>    <NA>    OUT-
    HEAD    first-middle    <NA>    <NA>    <NA>    <NA>
                    honen__etiologia__eztabaidagarria__da__.    EDU Honen   etiologia
    eztabaidagarria DET NOUN    ADJ .   PUNCT   veryshort   etiologia   <NA>    <NA>    NOUN
    <NA>    <NA>    IN-HEAD first-middle    <NA>    <NA>    <NA>    <NA>
                </NSelaborazioa>
                <NSelaborazioa>
                    <NNkonjuntzioa>
                        ultzera__mingarri__batzu__bezela__agertzen__da__,   EDU Ultzera mingarri
         batzu   NOUN    ADJ DET ,   PUNCT   short   agertzen    <NA>    <NA>    VERB    <NA>
    <NA>    IN-HEAD second-middle   <NA>    <NA>    <NA>    <NA>
                        tamainu__,__kokapena__eta__iraunkortasuna__aldakorra__izanik__. EDU
    tamainu ,   kokapena    NOUN    PUNCT   NOUN    .   PUNCT   short   izanik  <NA>    <NA>
    VERB    <NA>    <NA>    OUT-HEAD    second-middle   <NA>    <NA>    <NA>    <NA>
                    </NNkonjuntzioa>
                    hauek__periodiki__beragertzen__dira__.  EDU Hauek   periodiki   beragertzen
    DET ADV VERB    .   PUNCT   veryshort   beragertzen <NA>    <NA>    VERB    <NA>    <NA>
    IN-HEAD end <NA>    <NA>    <NA>    <NA>
                </NSelaborazioa>
            </NSelaborazioa>
            lan__honetan__patologia__arrunt__honetan__ezaugarri__epidemiologiko__,
    __etiopatogeniko__eta__klinikopatologiko__garrantsitsuenak__analizatzen__ditugu__.    EDU
    Lan honetan patologia   NOUN    DET NOUN    .   PUNCT   short   analizatzen <NA>    <NA>
    VERB    <NA>    <NA>    IN-HEAD last    <NA>    <NA>    <NA>    <NA>
        </SNtestuingurua>
    </SNprestatzea>
</ROOT>
```

Sistema entrenatzeko corpusa osatzen duten fitxategiak *rs3* formatutik *tbk* formatura aldaketa nola gauzatu den aipatu aurretik bi formatuen arteko desberdintasunak azalduko dira. Formatu bakoitzaren ezaugarriak, 4.1 Irudian konpara daitezke non diagrama bezala adierazita agertzen diren.

4.1a Irudian *rs3* fitxategia nodoez osatutako zuhaitza modura adierazita ikus daiteke, eta 4.1b Irudian, berriz, *tbk*ri dagokion nodoen zuhaitza. Bi diagramak mailaka antolatuta daude, maila bakoitzean dauden elementuak gaineko mailako nodoaren erlazio bidez lo-

tuta. *rs3* formatuari dagokion diagramak 3 maila ditu eta *tbk*ri dagokiona 5. Koloreak nodo bakoitzaren azpinodoak adierazten dituzte, geziekin konektatuta.



(a) Nodoen zuhaitza *.rs3*ko formatua jarraituz. **(b)** Nodoen zuhaitza *.tbk*ko formatua jarraituz.

4.1 Irudia: RST informazioa duten formatu desberdinen zuhaitzak.

Bien arteko desberdintasun bakarra RST zuhaitzen errepresentazioa da. Adibidez, $span_1$ aztertuta, 4.1a Irudian nodo urdinari dagokio, eta *span* hori osatzen duen nukleoa (nodo berdea, EDU_2) zuzenean dago, baina satelitea eta nodo berdea lotuko duen erlazioa lortzeko nodo arrosa ($span_4$) kontuan hartu behar da, hau da, beste azpinodo bat aztertu. Aldiz, 4.1b Irudian eskuineko nodo berdeari dagokio ($span_1$) eta zuzenean dugu bere azpinodo arrosak ($span_4$, EDU_3) lotuko dituen erlazioa.

Era sinplean esanda, formatu aldaketaren programaren helburua 4.1a Irudiko eskema izatetik 4.1b Irudiko eskema lortzea da.

4.4 Ebaluaziorako fitxategien formatuaren azterketa

Sistema entrenatu ostean, lortutako eredua ebaluatzeko fitxategiak behar dira. Horretarako, testuak formatu parentetikoan (*brackets*) adierazita egongo dira. RST zuhaitz egituraren hierarkia parentesien bidez kontrolatzen da, formatuaren izenak aipatzen duen bezala. Parentesi bat irekitzean EDU bat ala *span* bat egon daiteke. EDUa bada, tokenizatuta agertuko da, gure kasuan azpimarren bitartez. *Span* bat egotekotan barnean bi parentesi irekiko dira, beste EDU edota *span* egonik. Gainera, RST erlazioa eta nukleo-satelite ordena espezifikatzen da, EDUen ordena irakurleak irakurtzeko ordenan agertu behar baitira. 4.5 Zerrendan *brackets* formatuaren definizio formala agertzen da.

4.5 Zerrenda: .brackets formatuaren definizioa

```
span ← (erlazioa edukia edukia)

erlazioa ←  SNnukbak
          | NSnukbak
          | NNnukaniz

edukia ←  (EDU testua)
        | span

nukbak ← {edozein erlazio nukleobakarra}
nukaniz ← {edozein erlazio nukleoaniztuna}

testua ← {EDUari dagokion testua}
```

3.3 Irudian agertzen den GMB0301 testuaren errepresentazioa *brackets* formatuan 4.6 Zerrendan ikus daiteke. Zerrendako testua irakurri heinean, irudian agertzen den zuhaitza jarraitu behar da, ezkerretik eskuinera eta goitik behera. Zerrendako *bracket*sen agertzen den lehenengo RST erlazioa *Prestatzea* da, *SN*k adieraziz irekiko den lehenengo parentesia satelitea dela eta hurrengoa, aurreko parentesia itxi ostean, nukleoa. Satelitean, irudiko $span_1$ agertzen da. Nukleoa *span* bat denez, $span_{12}$, *bracket*sen *span* hori osatzen duten *span*ak izango ditu. *Testuingurua* erlazioan, satelitea EDU bat da, baina lehendabizi satelitea aztertu behar da, $span_{11}$. Berdin $span_{10}$arekin, *Elaborazioa* erlazioko nukleoa aztertu behar baita. Ondoren, EDU_2ko eta EDU_4ko testuak agertuko dira bakoitza bere parentesian, eta *Elaborazioa* itxi ondoren, aurreko *Elaborazioa* satelitea osatzen jarraituko da.

4.6 Zerrenda: GMB0301 testua RST informazioarekin, formatu parentetikoan (brackets)

```
(SNprestatzea (EDU estomatitis__aftosa__recurrente__-LRB-__i__-RRB-__:__epidemiologia__,
    __etiopatogenia__eta__aspektu__klinikopatologikoak__.) (SNtestuingurua (NSelaborazioa (
    NSelaborazioa (EDU estomatitis__aftosa__recurrente__deritzon__patologia__,
    __ahoan__agertzen__den__ugarienetako__bat__da__.) (EDU
    honen__etiologia__eztabaidagarria__da__.)) (NSelaborazioa (NNkonjuntzioa (EDU
    ultzera__mingarri__batzu__bezela__agertzen__da__,) (EDU tamainu__,
    __kokapena__eta__iraunkortasuna__aldakorra__izanik__.)) (EDU
    hauek__periodiki__beragertzen__dira__.))) (EDU
    lan__honetan__patologia__arrunt__honetan__ezaugarri__epidemiologiko__,
    __etiopatogeniko__eta__klinikopatologiko__garrantsitsuenak__analizatzen__ditugu__.)))
```

4.5 Formatu aldaketa

Atal honetan, tokenizazioaren ondoren egin behar diren formatu aldaketak nola inplementatu diren azalduko da. Horretarako hartu behar izan diren diseinu erabakiak eta bestelako xehetasunak aipatuko dira.

Esan beharra dago formatu aldaketaren atazaren esfortzua proiektuaren denbora handiena izan dela. Izan ere, *rs3* formatuaren errepresentazio era, kodeketa eta kasu guztiak ulertu behar izan dira. Gainera, kodea sortu, moldatu eta testeatu behar izan da, era egokian funtzionatzen duela frogatu arte.

Prozesu osoa garatzeko, script bat sortu da. Programa honek *rs3* fitxategiak hartuko ditu, eta irteera gisa *tbk*, *brackets*, *raw* eta *txt* formatuak sortuko ditu. *txt* formatuan *rs3*an agertzen den EDU guztien testua ordenan agertuko da, RST zuhaitz egiturarik eta informaziorik gabe.

4.5.1 Hurbilpen teknikoa

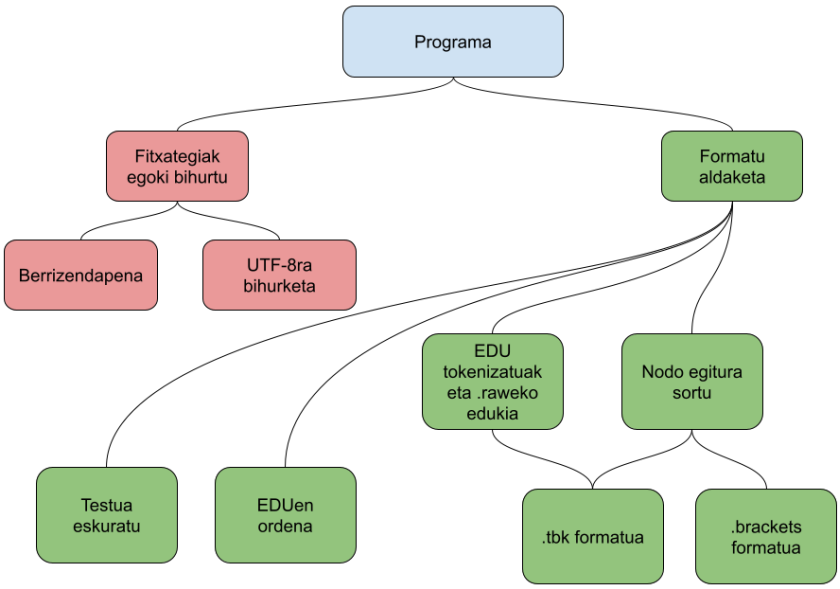
Lehenengo programazio lengoaia erabaki behar izan da. Ataza txikietarako, Linuxeko bash lengoaian scriptak egin dira, eta formatu aldaketaren atazarako, Python lengoaia. Erabaki hau, batez ere, Ixa taldean garaturiko beste programak nagusiki bi lengoaia horietan daudelako izan da.

Hala ere, diseinu erabaki garrantzitsuena programa atazetan banatzea izan da. Honen diagrama 4.2 Irudian ikus daiteke.

Basheko scriptak fitxategiak era sinplean tratatu behar izan direnean erabili dira. Bi script sortu dira (4.2 irudian gorriz), bat fitxategien izenak berrizendatzeko beharrezkoa denean eta bestea fitxategiak UTF-8 formatura pasatzeko.

Formatu aldaketa egiteko programa izango da Pythonen idatzita egongo dena. Corpuseko formatutik (*rs3*), sistema entrenatzeko formatua (*tbk*) lortzeko, tarteko pausu bezala, klase bat sortu da RSTko zuhaitz egitura gordetzeko, nodoak balira bezala.

Barneko Nodo klaseko zuhaitz egitura betetzeko, programa errekurtsibo baten bitartez egitea erabaki da. Osatuta dagoen zuhaitz egitura errekurtsiboki irakurtzeak formatu parentetikoa (*brackets*) eta entrenatzeko formatua (*tbk*) eraikitzen lagunduko du. Gainera, corpuseko formatuaren (*rs3*) RST zuhaitza errekurtsiboki irakurriz ziurta dezakegu Nodo klaseko zuhaitz egitura betetzen ari dela. Nodo klase honen egitura 4.1b Irudian ikus dai-



4.2 Irudia: Formatu aldaketaren programaren atazak

teke, gero hortik abiatuta azkeneko *tbk* eta *brackets* formatuak lortuko lirateke. Errekurtsibitateari esker programaren kodearen ulermena naturalagoa izango da RST diagramak zuhaitzak direlako.

Bestalde, corpuseko formatua (*rs3*) *xml* formatuan oinarritzen denez, *xml* tratatzen laguntzen duten hainbat liburutegi daude Pythonentzako. Aukeratutakoa ElementTree [2] izan da. Liburutegi honi esker, etiketa guztiak iteratu ditzakegu edota etiketa bat jaso atributu balio baten arabera. Hau oso erabilgarria izango da EDUen edota *span*en *parent* eta *id* balioak erlazionatzeko.

4.5.2 Inplementazioa

Basheko programen inplementazioa honela egin da. Direktorioan, fitxategi bakoitzeko, fitxategiaren izenean kategoria eta identifikazioa bakarrik utzi dira eta edukiaren kodeketaren arabera, behar izanez gero, UTF-8rako bihurketa egin da.

Formatu aldaketa egiteko programa ataza independentetan banatu da, (4.2 Irudian berdez agertzen diren paketeak), hau da, programa nagusi bat dago ataza horiek ordena egokian

[2]ElemenTree liburutegiaren dokumentazioa: https://docs.python.org/3/library/xml.etree.elementtree.html

exekutatzen dituena azken emaitza lortu arte. Programaren pseudokodea 1 Algoritmoan azter daiteke.

Algorithm 1 Formatu aldaketaren pseudokodea

Require: .rs3 fitxategi bat ala .rs3 fitxategiak dituen direktorio bat
Ensure: .rs3 fitxategi guztien .tbk, .brackets, .raw eta .txt

```
1:  function FORMATUALDAKETA(path)
2:      fitxategiak ← LORTURS3(path)
3:      formatuak ← []
4:      for all fitx ∈ fitxategiak do
5:          rs3 ← ELEMENTTREE(fitx)
6:          ordenak ← ORDENAEDU(rs3)
7:          txt ← TXTLORTU(ordenak)
8:          FORMATUAKGORDE([txt])
9:          hasiera ← ROOT(rs3)
10:         zuhaitza ← ZUHAITZANODOAK(hasiera, rs3)
11:         brackets ← PARENTETIKOA(zuhaitza)
12:         raw ← RAWLORTU(zuhaitza)
13:         tbk ← TBKLORTU(zuhaitza, raw)
14:         formatuak ← GEHITU(formatuak, brackets, raw, tbk)
15:     end for
16:     bracketsGuztiak, rawGuztiak, tbkGuztiak ← KONKATENATU(formatuak)
17:     FORMATUAKGORDE([bracketsGuztiak, rawGuztiak, tbkGuztiak])
18: end function
```

Honako hauek dira formatu aldaketa egiteko jarraitu behar diren urratsak:

1. **Fitxategiak.** Programari argumentutzat pasatu zaion bidea (*patha*) aztertu, *lortuRS3*rekin. Direktorio bat bada, programak barneko *rs3* fitxategiak hartuko ditu kontuan, bestela, bidean (*patha*ean) adierazitako fitxategia (*rs3*) erabiliko du.

2. ***rs3* kargatu.** *rs3* fitxategi bakoitzeko beharrezko formatuak eskuratuko dira. Horretarako, *ElementTree* liburutegiari dei egiten zaio; horrek *rs3* fitxategia *xml* gisa analizatzen du eta ElementTree klaseko objektu bat itzultzen du. Horrela,*xml* etiketen, atributuen eta testuen kontsultak egiteko aukera izango dugu.

3. **EDUak ordenatuta gorde.** *ordenatuEDU* funtzioarekin, *rs3*-n agertzen diren EDUak gordetzen dira, irakurtzeko ordenan.

   EDUen ordena egokia lortzeko hiztegi bat sortzen da, non EDUko testua pasata dagokion zenbakia itzultzen du. Zenbaki horrek testuan EDUak duen posizioa adierazten du, goranzko ordenan, adibidez, testua osatzen duen lehenengo EDUari

1 zenbakia dagokio. *Span*en ordena jakiteko, *span*a osatzen duen edozein EDUren posizio zenbakia hartuko da, azken finean ordena jakiteko zenbakiak konparatu baino ez da egin behar. Hau da, zenbaki txikiena duen EDUa edota *span*a, lehenago jarri behar da testua osatzeko unean.

4. **Testua lortu.** Testua EDUetan banatuta izan ordez, guztien testua lotu testu bakarra eta originala izateko, esaldiz esaldi eta paragrafoekin. Hau *txtLortu* funtzioarekin lortzen da, eta ondorengo *formatuakGorde* funtzioarekin testua *txt* formatuan gordetzen da, irakurritako *rs3* fitxategiaren izena mantenduz.

5. **Goreneko elementua eskuratu *rs3*tik.** *Root* funtzioaren bitartez egiten da. *rs3* formatuan adierazitako RST zuhaitza irakurtzeko balioko du.

6. **Nodo egitura duen zuhaitza eratu.** Aurretik lortutako *root* elementuarekin Nodo klase gorena osatuko dugu. Nodo horren izena zuhaitzeko bi hostoak lotzen dituen erlazioa izango da. Hosto bat EDU bat bada, nodoaren izena testua izango da. Aldiz, *span* bat bada, programari dei errekurtsibo bat egingo da *span* hori *root*tzat hartuz.

   Zuhaitzaren bi hostoak osatzeko, EDU edota *span*en posizioaren arabera, lehenago agertzen dena zuhaitz bitarreko ezkerreko nodoa osatuko du, ondoren agertzen dena eskuinekoa osatuz. Posizioak *ordenaEDU* funtziotik eskuratzen dira. Honek, gainera, nukleo-satelite ordenan eragiten du. Hori dela eta, nodoaren erlazioaren izenean *SN* ala *NS* gehituko zaio, satelitea eta nukleoaren hurrenkera adieraziz. Erlazio nukleoaniztunetarako, *NN* gehitzen da biak nukleoak dira eta.

   4.1 Irudiko adibidea jarraituz, 4.1a eredutik (*rs3*) 4.1b eredura (*tbk*) pasatzeko, ondorengo pausuak egiten dira:

   (a) 4.1a Irudian *Root* nodoaren *ID*a *parent* gisa duten *span*ak edota EDUak aztertu, hau da, $span_1$i apuntatzen dizkioten gezien abiapuntu nodoak (EDU_2 eta EDU_3).

   (b) EDU_2 nodoak erlazioa duenez, sortu behar dugun 4.1b ereduan nodo bat sortuko dugu, *Erlazioa A* izenarekin (nodo urdina), eta nodo horren hosto batean EDU_2a jarriko dugu.

   (c) 4.1a Irudiko EDU_3k ez du erlaziorik $span_1$en parte delako, nukleoa hain zuzen ere. Bere *ID*a *parent* gisa duten nodoak aztertuta, $span_4$ (nodo arrosa) bakarrik topatzen dugu, eta ondorioz, hori $span_1$en satelitea izango da. Nodo horiek *Erlazioa B*rekin lotuta daudenez, gure 4.1b ereduko zuhaitzean *Erlazioa A* nodoaren azpian jarriko dugu *Erlazioa B* izenarekin, bere hosto bat EDU_3

izanik (nodo arrosa). Beste hostoan 4.1a Irudian dagoen EDU_3rekin dagoen satelitea joango da, baina *span* bat denez ($span_4$), 4.1b Irudian *span* hori lotzen dituen erlazioa joango da.

(d) 4.1a aztertuta, $span_4$ak hiru azpinodo ditu, *Erlazioa C* erlazio nukleoaniztunarekin (hirurek erlazio bera baitute, hirurak dira nukleoak). Beraz, 4.1b Irudian geratzen zen hostoan *Erlazioa C* joango da (nodo arrosa), 4.1a irudiko $span_4$ osatzen duten nodoak zuhaitz bitarra eratzen jarraituz.

(e) Azkenik, 4.1b Irudiko ereduan erlazio-nodo bakoitzeko hostoak EDUen ordenan ezarri beharko lirateke, hau da, testuan agertzen diren ordenaren arabera hostoen ezker eta eskuin aldeak erabakiko dira.

Pausu horiek diagramak jarraituz azaldu dira, baina, berez, *rs3* fitxategiko *xml*a aztertuta Nodo klaseko zuhaitz bitarra eratuko da.

7. **brackets formatua sortu.** Nodo egituratik *brackets* formatua sortuko da. Hierarkia bera osatzen dute, baina mailak parentesien bidez kontrolatuko dira. Sakonera bidez zuhaitz bitarra ezkerretik eskuinera irakurriz joango gara formatua eraikitzen.

8. **raw formatua sortu.** Horretarako, lehendabizi *Universal Dependencies* tresna erabilita EDUak tokenizatuko dira, eta ondoren, *raw* fitxategian agertzen diren atributuak lortuko dira.

9. **tbk formatua sortu.** Nodo egitura baliatuz, zuhaitz bitarra ezkerretik eskuinera sakonera bidez irakurriz sortuko dugu *tbk* formatua. Ondoren, EDUen testua ordezkatuko da *raw*en agertzen den EDU horri dagozkion atributuekin eta tokenizazioarekin.

10. **Fitxategien edukia kateatu.** Sare neuronalak erabiltzeko, corpusa fitxategi bakarrean mantendu behar dugu. Horregatik, testu bakoitzetik lortutako formatu desberdinen edukia kateatu egingo da, *brackets*, *raw* eta *tbk* formatuko fitxategietan testu guztiak izateko.

11. **Fitxategiak gorde.** Kateamendua egin ondoren fitxategiak gordeko dira. Horrela, corpusaren aurreprozesaketa pausua bukatzen da, eta emaitza erabilgarria izango da hurrengo pausurako: RST iragartzeko sistema euskarazko testuekin martxan jartzeko.

5. KAPITULUA

RST erlazioak iragartzeko sistema

RST erlazioak iragartzeko sistemaren, parserraren, helburua RST zuhaitzak eta erlazioak iragartzea da. Horretarako, parserrako orakuloak RST zuhaitzak eraikitzeko burutu beharko ekintzak ikasi behar ditu. Ikasketa-prozesua sare neuronalen bidez egingo da, corpuseko RST zuhaitzak baliatuz.

Erabili den RST parserra Ixa taldean garatutakoa [Iruskieta and Braud, 2019] da. Segmentazioa, berriz, [Atutxa et al., 2019] lanean ageri den sistemarekin burutu da eta diskurtso parserra [Braud et al., 2017] lanean ageri denarekin. RST parserra C lengoaian idatzita dago eta bi modulu ditu: ereduaren entrenamendua egiteko modulua, eta iragarpena egiteko modulua. Bi exekutagarriz osatuta dago, neural network train, *nnt* eta neural network parser, (*nnp*). Lehenengoak, RST erlazioak iragartzeko gai diren ereduak ikastea izango du helburu eta bigarrenak RST erlazioak iragartzea. Parserraren arkitekturari dagokionez, trantsizio bidezkoa da, orakulo dinamikoarekin.

Orakuloak RST zuhaitzak ikasteko sare neuronalak erabiliko ditu. Orakuloa entrenatzeko era bat LSTMak erabiltzea [Braud et al., 2016b] da. Parser honek zuhaitzak eratzeko heuristikoak eta erlazioak iragartzeko LSTMak hierarkikoki antolatuta erabiltzen ditu.

Hala ere, parserrean arkitektura desberdinak probatu dira, adibidez, MLP motako sare neuronalak erabiliz LSTMen ordez. Arkitektura aldaketa horrekin parserra sinplifikatu da eta artearen egoerarekin alderatuta emaitza hobeak lortu dira.

Hori dela eta, GrAL honetan [Iruskieta and Braud, 2019] lanean azaltzen den RST parserra erabiliko da. Parser hori, [Braud et al., 2017] artikuluan aipatzen den parserrean oinarritzen da.

Gainera, parser horiekin euskarazko corpusa erabilita hainbat esperimentu egin dira, eta horrek ahalbidetzen digu GrAL honetan lortutako emaitzak lan haiekin konparatzea.

Gure sistemaren ereduak sortu ahal izateko MLParen neuronen parametroak ikasiko ditu parserrak. Behin eredu bat ikasita, testu baten EDUak jasota, orakuloak zuhaitzak sortzeko ekintzak eta RST erlazioak iragarriko ditu, azkenean RST zuhaitza sortuz.

5.1 RST erlazioak ikasten

Ikasketa prozesua martxan jarri baino lehen, entrenamendurako sistemaren (*nnt*) hiperparametroak finkatu behar dira. Horretarako, hiperparametro guztien konbinazio ezberdinak frogatuko dira egokienak aukeratu ahal izateko. Hortaz, sare neuronala kutxa beltz bat bezala erabiltzen da, barneko arkitektura moldatu gabe.

Ikasketa-prozesua egiteko eta ebaluatzeko, corpusa hiru zatitan banatu dugu: *train*, *dev* eta *test*. *Train* zatian sistema entrenatzeko fitxategiak egongo dira, *dev* zatian garapenerako fitxategiak, entrenatu ahala ereduen errendimendua aztertzeko, eta *test* zatian ereduak ebaluatzeko fitxategiak. Corpuseko 164 fitxategietatik *train* multzoan 116 egongo dira (%70,7), *dev* zatian 20 (%12.2) eta *test*ean 28 (%17.1).

Beraz, banaketa egiteko jarraitu den irizpidea erlazioen agerpena ahalik eta proportzionalkien izatea izan da (banaketaren ehunekoak kontuan hartuta), nolabaiteko oreka lortzeko. Hau da, erlazio baten agerpen guztietatik, %70,7 *train*ean, %12.2 *dev*en eta %17.1 testean agertzea izango litzateke egokiena, banaketako proportzioak mantenduz. Banaketetan agertzen diren erlazioen kopurua 5.1 Taulan ikus daiteke.

5.1 Taulako "Guztira" zutabeak corpus osoan zehar erlazio bakoitzaren agerpen kopurua adierazten du. Erlazioak balio horren arabera ordenatuta daude handienetik txikienera. Gehien agertzen diren erlazioak gutxi gora behera banaketaren proportzioak jarraitzen dituzte, *Elaborazioa* erlaziotik *Kausa* erlaziora arteko tartean ikus daitezkeen moduan. Aldiz, kopuru gutxiagotan agertzen diren erlazioen proportzioa banaketan zehar mantentzea zailagoa izan da.

Hala ere, kontuan hartu behar da erlazio baten proportzioa orekatzeak beste erlazio baten proportzioan aldaketa eragingo duela.

Hori dela eta, agerpen gehien dituzten erlazioetan banaketaren proportzioa mantentzea erabaki da posible den heinean, erlazio horietan kritikoagoa baita. Beraz, banaketaren proportziotik erlazioek desbiderapen txikia izatea lortu nahi da. Aldi berean, banaketako

Erlazioa	Guztira	Train	Dev	Test
Elaborazioa	805	516	122	167
Lista	511	385	57	69
Same-unit	389	265	28	96
Prestatzea	369	219	83	67
Konjuntzioa	301	198	34	69
Helburua	243	188	32	23
Sekuentzia	235	172	32	31
Zirkunstantzia	219	121	17	81
Kontrastea	211	149	10	52
Ondorioa	208	136	58	14
Metodoa	174	129	20	25
Testuingurua	170	121	5	44
Kausa	157	111	16	30
Ebaluazioa	134	17	1	116
Kontzesioa	114	63	2	49
Interpretazioa	79	46	1	32
Justifikazioa	62	26	-	36
Birformulazioa	59	24	1	34
Antitesia	44	18	6	20
Baldintza	44	27	6	11
Arazo-soluzioa	43	33	-	10
Ebidentzia	38	21	1	16
Disjuntzioa	37	18	4	15
Ahalbideratzea	19	17	-	2
Bateratzea	18	-	8	10
Laburpena	16	14	-	2
Motibazioa	15	9	2	4
Alderantzizko-baldintza	8	5	-	3
Ez-baldintzatzailea	7	2	-	5
Birformulazioa-nn	4	4	-	-
Aukera	2	1	-	1

5.1 Taula: Train, dev eta test banaketetan RST diskurtso erlazioen maiztasuna

proportzioa mantentzen ez duten erlazioen kopurua ahalik eta txikiena izaten saiatu gara. Izan ere, agerpen desorekak arazoak ekar ditzake entrenamenduan eta gaizki ikasteko arazoa egon daiteke.

5.2 Esperimentazioa

Atal honetan, egindako esperimentuetan erabili diren parametroak azalduko dira. Sei dira eredu ezberdinak sortzeko kontuan izan ditugun hiperparametroak:

- **Embeddings:** Facebook
- **Iterations (it):** [1..10]
- **Learning-rate (lr):** [0.01, 0.02]
- **Decrease-constant (dc):** [0, 1e-5, 1e-6, 1e-7]
- **Hidden-layers (h):** [64, 128, 256]
- **Beam:** [1, 2, 4, 8, 16, 32]

*Embedding*ak hitzak errepresentatzeko era bat dira, horretarako, zenbakizko bektoreak erabiltzen dira. Bi hitzen bektore errepresentazioen arteko diferentzia zenbat eta txikiagoa izan, bi hitz horiek antzekoagoak izango dira. Guk Facebook enpresak egindakoak erabili ditugu, hitzen arteko erlazioak ikasita dituztelako dagoeneko.

Iterations parametroak, entrenamenduan egingo den iterazio kopurua adierazten du. Guk 1etik 10erako balioak probatuko ditugu.

Learning-rate hiperparametroarekin sare neuronala ikasteko prozesuan bere pisuak aldatu behar direnean zenbateko aldaketa izango duten adierazten du, 0 eta 1 tartean adierazita. Balioak gero eta handiagoa, orduan eta ikasketa optimo bat galtzeko aukera handiagoa izango da. Era berean, balioak gero eta txikiagoak, ikasketa orduan eta motelagoa izango da. Horregatik, 0.01 eta 0.02 balioak erabili ditugu.

*Decrease-constant*ak *learning-rate* parametroaren balioa jaisteaz arduratuko da, ikasketa aurrera doala. Ez da komeni zenbaki altuak izatea, bestela *learning-rate* hiperparametroa erabat jaitsiko da entrenamenduan, ikasketa geldiarazteko arriskuarekin. Hori dela eta, 0 balioa kontuan hartzen dugu aldaketarik ez egoteko *learning-rate* hiperparametroan, eta $1e-5$, $1e-6$ eta $1e-7$ aldaketa txikiak egoteko.

*Hidden-layers*en bidez ezkutuko geruza bat gehituko zaio arkitekturari, balioak adierazten duen neuronen kopuruarekin. 64 128 eta 256 neurona kopuruak probatuko ditugu.

*Beam*ek MLPren zenbat irteera kontuan hartuko diren adierazten du. 1 balioarekin orakulo estatikoa izango dugu, eta 1 baino gehiagoko balioekin adina irteera izango dira, bakoitzaren probabilitatearekin, orakulo dinamikoan aukeraketa egitean eragina izanik. 1etik 32ra arteko balioak frogatu ditugu 2ko progresio geometrikoa jarraituz.

Sortutako ereduak hiperparametro horien balio guztien konbinazioak eginez sortu dira. Guztira, 1440 eredu ezberdin entrenatu dira.

6. KAPITULUA

Ebaluazioa eta emaitzak

Atal honetan, parserrarekin sortutako ereduak ebaluatuko dira eta RST zuhaitz eta erlazioak iragartzen onenak direnak hautatuko dira bukaerako sistema eraikitzeko.

Sortutako eredu guztiak parserreko *nnp* sistemarekin probatuko dira. Eredu bakoitza ebaluatzeko, ordea, *test* multzoko fitxategiak erabili dira. RST zuhaitzak kuantitatiboki ebaluatzeko, literaturan erabiltzen diren metrika hauen bitartez egingo da [Marcu, 2000a]: *span*, *nuclearity* eta *relation*. Balio horiek zenbat eta altuagoak, eredua orduan eta hobea izango da.

Ebaluaziorako, gure sistemak lortutako emaitza *gold* fitxategiekin, hau da, eskuz ondo etiketatuta daudenekin, konparatzen ditu. Hau da, modu automatikoan lortu diren *brackets* fitxategiak, eskuz etiketatutako *brackets* fitxategiekin. Metrikek bi zuhaitz horien arteko adostasuna adieraziko dute.

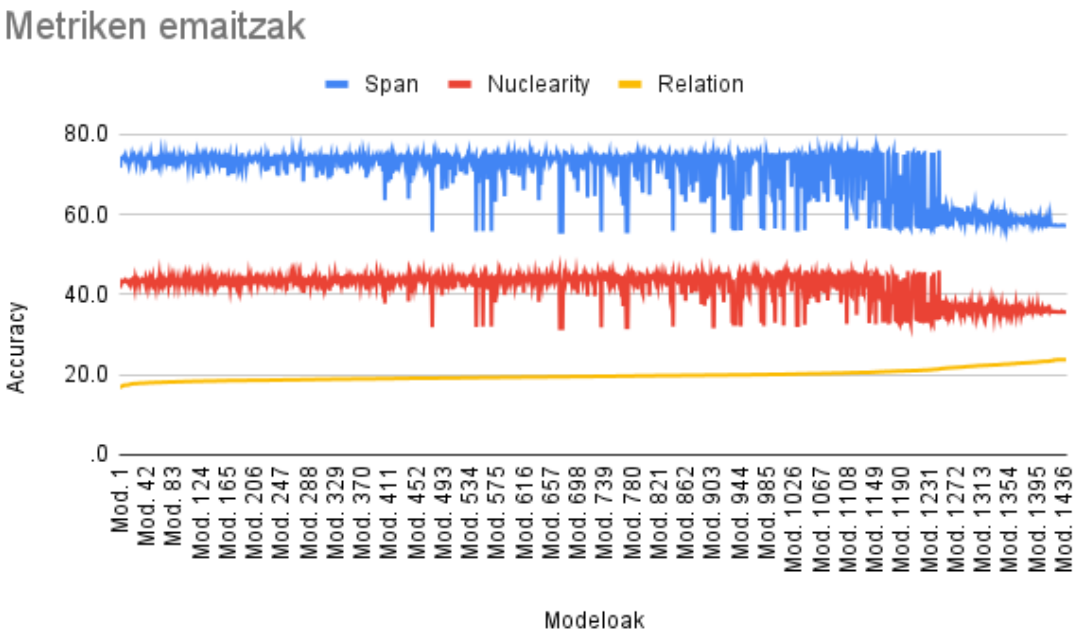
Beraz, *gold* eta iragarritako zuhaitzak konparatuz, *Span* metrikak, EDU edota *span*en multzokatze ala hierarkia ebaluatzen du, bi zuhaitzen arteko adostasuna adieraziz; *nuclearity*k erlazio baten bidez lotutako EDU edota *span*ak satelite-nukleo, nukleo-satelite ala nukleo-nukleo bi zuhaitzetan berdinak diren ebaluatzen ditu; eta *relation* balioak bi zuhaitzetan agertzen diren erlazioen adostasuna ebaluatzen ditu.

Eredu onenak hautatzerako orduan, *relation* puntuazioari begiratu zaio, bereziki erlazioetan puntuazio altua izateak ahalbidetzen duelako RST zuhaitz-diagramak sortzeko eredu egokia izatea. Era horretan, nahiz eta hierarkia hain ona ez izan, EDUen arteko erlazioak egokiagoak ala naturalagoak izango dira. Aitzitik, *span* puntuazio altu bat izateak EDUen

arteko hierarkia hobea izatea eragingo du, baina *relation* puntuazioa baxua bada, gerta daiteke EDUen artean askotan erlazio mota bera jartzea.

Beraz, lortutako ereduak *relation* puntuazioaren arabera ordenatzen badira, 6.1 Irudian ikusten den grafikoa lortuko da, entrenatutako 1440 ereduen metriken balioekin. Orokorrean, ia eredu gehienekin antzeko emaitzak lortu dira. X ardatzean adierazitako 1. eredutik 1231.era arte, *span* metrikan batezbesteko % 74ko zehaztasuna lortu da, *nuclearity*n % 44 eta *relation*en % 20.

Begi bistakoa da *relation* balio altuagoak lortu dituzten ereduek (X ardatzeko 1231. eredutik aurrera) *span* eta *nuclearity* balio baxuagoak lortu dituztela. Horren arrazoia, eredu horien sare neuronaletan topa dezakegu. Entrenamenduan, sare neuronalaren pisuak aldatzean, *relation* metrikaren zuzentasuna handitu da, baina *span* eta *nuclearity* metriken zuzentasuna kaltetuz.



6.1 Irudia: Entrenatutako eredu bakoitzetik lortutako metriken emaitzak.

Relation puntuazio altuena lortu duten lau eredurik onenak (A, B, C eta D letrekin izendatuak) eta beraien metrika balioak 6.1 Taulan erakusten dira. Lau eredu horietan erabilitako hiperparametroak taulan ageri dira. A eredua izan da hoberena, *relation* aldagaian baliorik altuena lortuta, % 23,77 zehaztasuna, hain zuzen ere. B eta C ereduek, A ereduak baino emaitza txarragoak lortzen dituzte metrika gehienetan (C ereduaren *nuclearity* salbues-

Eredua	Ereduen hiperparametroak	Span	Nuclearity	Relation
A	lr=0,01, dc=1e-5, h=64, it=2, beam=1	% 57,29	% 35,74	% 23,77
B	lr=0,02, dc=1e-7, h=64, it=1, beam=1	% 57,22	% 35,64	% 23,74
C	lr=0,02, dc=1e-5, h=128, it=1, beam=1	% 57,27	% 35,79	% 23,52
D	lr=0,01, dc=1e-6, h=64, it=2, beam=8	% 59,1	% 36,34	% 23,39

6.1 Taula: Relation puntuazio altuena lortutako ereduen zehaztasuna

pena izanik). D ereduak berriz, *span* eta *nuclearity* aldagaietan, A ereduak baino balio altuagoak ditu, 1,81 eta 0,6 puntu, hurrenez hurren. Agian, D eredua A ereduarekin batera erabiltzea interesgarria izango litzateke, *span* eta *nuclearity* nahiko hobetzen baitira *relation* aldagaian jaitsitako balioarekin alderatuz (0,38).

Eredua	Ereduen hiperparametroak	Span	Nuclearity	Relation
E	lr=0,02, dc=1e-7, h=256, it=5, beam=8	% 75,98	% 45,93	% 21,4
F	lr=0,02, dc=0, h=64, it=9, beam=32	% 75,05	% 46,29	% 20,29

6.2 Taula: Span altuena eta nuclearity altuenak lortutako ereduen zehaztasuna, hurrenez hurren.

Bestalde, 6.2 Taulan *relation* puntuazio eskasagoak lortu dituzten bi eredu aurkezten dira. Alde batetik, E ereduak *span* puntuazio altuena lortu duena da, eta bestetik, F ereduak *nuclearity* puntuazio altuena lortu duena da.

Bistan da 6.2 Taulako ereduen *relation* puntuazioa 6.1 Taulako ereduen *relation* puntuazioa baino txikiagoa dela, 3 puntuko aldea du batak bestearekiko. Hala eta guztiz ere, *span* eta *nuclearity* balioen hobekuntza askoz handiagoa da, 18,69 puntuko aldea A eta E ereduen artean, *span* neurrian eta 10,55 puntuko aldea A eta F ereduen artean, *nuclearity* neurrian.

Dena den, hasieran aipatu dugun bezala, A eredua izango litzateke aproposena erabiltzeko orduan *relation* balioa altuena duelako. Eredu guztien metriken zehaztasunak aztertuta, ordea, agian interesgarriagoa izango da *relation* txikiagoa duen eredu bat hautatzea *span* eta *nuclearity* adostasuna askoz handiagoa bada.

Eredua	Span	Nuclearity	Relation
A	% 57,29	% 35,74	% 23,77
E	% 75,98	% 45,93	% 21,4
[Braud et al., 2017]	% 78,6	% 53,0	% 26,4
[Iruskieta and Braud, 2019]	% 78,98	% 55,02	% 34,78

6.3 Taula: Lortutako ereduen metriken alderaketa beste lanekiko

Euskarazko testuen RST zuhaitzak iragartzeko egindako beste ikerketa batzuetan lortutako emaitzak, A eta E ereduekin alderatuta, 6.3 Taulan ikus daitezke. E eredua gertu geratu da [Braud et al., 2017] eredutik metriken balioak aztertuta. [Iruskieta and Braud, 2019] ereduak, ordea, A eta E ereduak baino errendimendu altuagoa lortu du hiru metriketan, baina batez ere *relation* metrikan.

Sistemen errendimendua modu kualitatibo batean ere azter daiteke. Adibide gisa, B Eranskinean, besteak beste, A eta E ereduek iragarritako RST zuhaitzekin batera, eskuz etiketatutako zuhaitza ikus daiteke. Adibiderako erabili den testua, B.1 Zerrendan ageri da, eskuzko etiketazioari dagokion zuhaitza B.1 Irudian, eta A eta E ereduen zuhaitzak B.2 Irudian eta B.3 Irudian hurrenez hurren.

A eredua izan da *Relation* puntuazio altuena lortu duena, baina B.2 Irudiari erreparatuta, errendimendu txarra daukala esan daiteke. Azken finean, bakarrik lehenengo *Prestatzea* erlazioa ikasi du, eta gainerako guztietan *Elaborazioa* erlazioa iragarri du. *Prestatzea* erlazioa lehenengo EDUan agertzean eta *Elaborazioa* erlazioa hainbatetan agertzeak zentzua du RST testu askok ezaugarri horiek dituztelako. Hala ere, egia da beti *Elaborazioa* erlazioa jartzeak ez duela zentzurik. E ereduari dagokion B.3 Irudia aztertuta, gutxienez beste erlazio batzuk jartzeko gai izan dela ikusten da. Gainera, *Sekuentzia* erlazioarekin lotutako bi EDUak asmatu ditu RST zuhaitz originalean (B.1 Irudia) ikus daiteken bezala.

Beraz, *Elaborazioa* askotan iragartzeak (*Elaborazioa* baita testuaren garapena egiteko erlaziorik erabiliena) *relation* metrikaren balioa igotzeak eragin du, RST zuhaitzetan gehien agertzen den erlazioa delako. E ereduak beste erlazio desberdin batzuk iragarri ditu, eta A ereduarekin konparatuta, iterazio gutxiagorekin entrenatua izan da. Hori dela eta, 10 iteraziorekin entrenatutako bi eredu aztertuko dira, ea askotariko RST erlazioak iragartzen diren, nahiz eta *relation* puntuazioa apur bat txikiagoa izan.

Horretarako 6.1 Irudian agertzen diren ereduetatik 10 iterazio dituzten modeloak aukeratuko dira, grafikan ikusten den *span* eta *nuclearity* balioak jaitsi baino lehen (1231. ereduaren aurretik daudenak). 10 iteraziorekin entrenatutako ereduak aukeratzeko arrazoia zentzuzkoa da: gero eta iterazio gehiagorekin, orduan eta ereduak gehiago ikasteko aukera du. Horregatik A ereduak (2 iterazio bakarrik) baino errendimendu hobea izatea espero da. Hala ere, gero eta iterazio gehiagorekin gainikasketa [1] (ingelesez *overfitting*) gertatzeko aukera handituko da. Aukeratutako 10 iterazioz entrenatutako ereduak 6.4 Taulan ikus daitezke, bakoitzean erabilitako hiperparametroekin.

[1] Gainikasketa egoera batean eredu batek ikasteko fitxategien ezagutza handia da, beraz inoiz ikusi ez duen egoera batekin arazoak izango ditu iragartzeko. Beste era batean esanda, ez du orokortzeko gaitasunik.

Eredua	Ereduen hiperparametroak	Span	Nuclearity	Relation
G	lr=0,01, dc=0, h=128, it=10, beam=4	% 74,08	% 44,44	% 19,53
H	lr=0,01, dc=1e-5, h=256, it=10, beam=1	% 74,59	% 42,50	% 18,85

6.4 Taula: 10 iteraziorekin entrenatutako bi eredu, lortutako ereduen zehaztasuna.

G eta H ereduek B.1 Zerrendako testutik iragarritako RST zuhaitzak B.4 Irudian eta B.5 Irudian ikus daitezke, hurrenez hurren. G ereduaren kasuan, erlazio anitzagoak iragartzen dituela ikusten da. H ereduaren kasuan, aldiz, zuhaitzaren hierarkia A, E eta G ereduekin konparatuz desberdina da (azkeneko EDUak ez daude zuhaitzeko azken mailan, baizik eta gorago).

Iragarritako RST zuhaitzak *gold* zuhaitzekin bat ez datozen arren, interesgarriagoak izango dira erlazio eta hierarkia aniztasuna iragartzen dituzten ereduak. Izan ere, metriken balioez gain, iragarritako zuhaitzak ere ebaluatu beharko dira, horrelako fenomenoak ekiditeko eta baliogarriak izan daitezkeen ereduak beste ikerketetan erabiltzeko.

Ahalik eta emaitzarik esanguratsuena lortzeko ebaluazioa egin dugu. Horregatik, metriken bidez ebaluatuak izan dira ereduak, hau da, kuantitatiboki. Hala ere, ereduak kualitatiboki ebaluatzea zuhaitz diagrama eta erlazioak aztertuta interesgarria da ere, aztertu den fenomenoa identifikatzeko eta erabiliko den eredua hautatzeko. Beraz, bai metrikak baita testu batzuen RST zuhaitzen diagramak aztertzea erabakigarria izango da ereduak hautatzeko eta erabiltzeko edo baztertzeko.

7. KAPITULUA

Analisi katea

RST egiturak iragartzeko tresna erreal eta erabilgarria eraikitzeko, testu gordinetik (*txt*) abiatzen den sistema behar da.

Aurreko ataletan, RST eredurik onena lortzeko urratsak eman dira, eta orain, eredu hori oinarri hartuta, hasieratik bukaerarako sistema edo tresna osatuko dugu. Ikusi dugun bezala, ereduen sarrerako fitxategiak *raw* formatuan egon behar dute, eta ondorioz, behar diren moldaketak egin beharko dira testu gordinetik abiatu nahi badugu. Hori dela eta, analisi katea deritzon tresna sortu da.

Tresna honi esker, testu gordina (*txt*) aurreprozesatuko da *raw* formatua lortu arte. Gainera, eredu onena analisi katean integratuko denez, era zuzenean ereduak iragarritako RST zuhaitza eta erlazioak lortuko dira. Laburbilduz, testu gordin batetik RST zuhaitz eta erlazioak iragarriko dira zuzenean. Horregatik, analisi katea da erabiltzaileentzako prest dagoen sistema.

Gainera, direktorio batean dauden *txt* motako fitxategi guztiak prozesatu nahi badira, ez da beharrezkoa izango banan banan egitea, programa behin bakarrik exekutatuta, fitxategi guztien RST zuhaitzak *rs3* formatuan lortuko direlako.

Analisi katearen prozesu osoa atazetan banatzen da, era jarraituan exekutatuz:

1. **Tokenizazioa.** Sarrerako testua (*txt*) tokenizatuko da.

2. **Segmentazioa.** Testua EDUetan segmentatuko da, hau da, aurretik lortutako tokenak segmentuetan banatuko dira. Horretarako [Atutxa et al., 2019] segmentatzailea erabiltzen da. Segmentuak identifikatzeko BIO notazioa erabiltzen da, B-SEG

tokena segmentuaren hasiera adierazteko, I-SEG tokena segmentuaren parte dela adierazteko, eta O-SEG segmentuaren amaiera adierazteko.

3. **Tokenen informazio gehigarria lortu.** *Universal Dependencies* liburutegia erabiliko da aurretik lortutako segmentuetako tokenei *lemma* eta POSa esleitzeko.

4. **EDUak *raw* formatuan lortu.** EDUekin eta EDUak aberasteko atributuekin, Ixa taldean inplementatutako funtzio baten bidez, *raw* formatuan dagoen fitxategia lortuko da.

5. **RST zuhaitza iragarri.** Lortutako *raw* formatuko fitxategia parserrako *nnp* exekutagarriari pasatuko zaio, lortutako eredurik hoberena erabilita. Horrek, RST zuhaitzaren hierarkia eta EDUen arteko erlazioak iragarriko ditu. Emaitza *brackets* formatuan adierazita egongo da.

6. **Iragarritako zuhaitza *rs3* formatuan lortu.** Ixa taldean inplementatutako funtzio baten bidez, *brackets* formatutik *rs3* formatura pasako da.

RST zuhaitza bistaratu. Azkenik, lortutako RST zuhaitzak RSTTool tresnarekin *.rs3* formatuan irudika daitezke. Pausu hau, ordea, ezin da testu guztietarako automatikoki egin RSTTool tresnak ez dituelako kontsolatik erabiltzeko komandoak inplementatuta, hau da, eskuz programa erabilita testuak banan banan irudikatuz egin daiteke bakarrik.

Hala ere, analisi katetik lortutako azken formatua *rs3*a izatea egokia da, RST zuhaitzak aurkezteko formatu estandarra delako. Hori dela eta, programa edo funtzio berri bat garatzen bada *rs3*ak era automatikoan irudi bihurtzeko, analisi katean programa edo funtzio hori gehitzea besterik ez litzateke egin behar.

8. KAPITULUA

Ondorioak

8.1 Proiektuaren ondorioak

Ez dago zalantzarik proiektuaren helburua lortu egin dela; hots euskarazko testuetatik RST zuhaitzak iragarriko dituen sistema martxan jartzea. Horretarako, lehendabizi, RST [Mann and Thompson, 1988] eta parserraren oinarri teorikoak aztertu ditugu. Honi esker, RST iragartzeko sistemaren funtzionamendua uler dezakegu, eta testuen RST zuhaitzak interpretatu ditzakegu, testuetatik iragarritako RST zuhaitzen kalitatea eta zuzentasuna ebaluatzeko aukera emanik.

RST zuhaitzak iragarriko dituen sistemaren parserra [Iruskieta and Braud, 2019] entrenatzeko, hainbat arloetako testuak eskuratu ditugu Euskal RTS Treebanketik eta corpusa osatu dugu. Hala ere, corpusaren formatua aldatu behar izan dugu parserreko entrenamendua egin ahal izateko.

Aipatu beharra dago formatu aldaketaren prozesuak izan duen garrantzia. Alde batetik, pausu hau lortu ezean ezin izango genuke parserra erabili *rs3* formatudun corpusekin. Beste alde batetik, formatu aldaketari esker egindako esperimentuak erreproduzigarriak dira, erabilitako parserrarekin beste edozeinek egindako esperimentuak edota beste corpus bat erabiltzeko ahalbidetuz.

Corpusaren formatu aldaketa eta gero parserrako sare neuronala entrenatu dugu. Hiperparametro desberdinekin hainbat eredu sortu ditugu. Ereduak ebaluatzerako orduan, aztertu dugu *Relation* puntuazioak ez direla izan oso altuak, batez ere beste ikerketetan lortuta-

koekin konparatuz [Braud et al., 2017] [Iruskieta and Braud, 2019]. Gainera, *relation* balio altuenak lortu dituzten ereduak (A adibidez) errendimendu txarragoa izan dute beti erlazio bera iragartzeagatik. Gure kasuan, *relation* balio baxuagoak lortu dituzten ereduak (G adibidez) errendimendu hobea dituztela iruditu zaizkigu, hierarkia eta erlazio barietatea egon delako. Beraz, ohartu gara ereduak RST zuhaitzen bitartez kualitatiboki ebaluatzeak duen garrantzia.

Hala ere, lortutako ereduak ez dira izan adierazgarriak, iragarritako RST zuhaitzen adostasuna orokorrean ez datozelako bat eskuz etiketatutako testu bereko RST zuhaitzarekin. Alabaina, ezin dugu baztertu corpusaren tamainaren eragina. Corpus handiago bat izatekotan ereduen errendimendua handitu lezake. Dena den, formatu aldaketaren programa gogobetekoa izan da, beraz atea irekita uzten dugu etorkizuneko lan bezala euskararako RST iragartzeko eredu hobeak lortzea.

Azkenik, analisi katea proposatzen dugu, era erraz batean edonork lortu egin ditugun ereduak probatu nahi baditu berak nahi dituen testuekin. Horregatik, analisi katea GrAL honetako atalik adierazgarrienatzat jotzen dut.

8.2 Etorkizuneko lana

Atal honetan, GrALean lortutakotik abiatuta etorkizun lan gisa planteatu litezkeen ikerketak aurkeztuko dira.

Corpusa handitu

Argi dago parserraren funtzionamendua hobetzeko ereduetan *relation* puntuazio altuagoak lortu behar direla, betiere ereduen ebaluazio kualitatiboa egokia bada (hau da, aztertutako erlazio bera iragartzearen fenomenoa ez bada gertatzen). Beraz, etorkizuneko lan bat corpus handiago batekin entrenamendua egitea izango litzateke.

Are gehiago, *train*, *dev*, eta *test* banaketa beste era batean planteatuta edota beste irizpide batzuk jarraituta banatzeak agian ereduen errendimendua handitu lezake. Adibidez, corpus handiago batekin *train*, *dev* eta *test* zatiketa arloka eta tokenak eta erlazioak kontuan hartuz egin liteke.

Ereduen hiperparametroak

Hiperparametro balio desberdinekin edota erabili ez diren beste hiperparametroekin (adibidez, *dropout*) ereduak sortu eta ebaluatu litezke. Alabaina, kontuan hartu behar da konputagailuaren konputazio ahalmena, zeren eta balio eta atributu gehiagorekin probatzeak exekuzioaren denbora handituko da, hiperparametroen konbinazio guztien ereduak sortuz gero.

Analisi katean RST zuhaitz-diagrama irudiak lortzeko integrazioa

Analisi katea prest dago testuetatik RST zuhaitzak iragartzeko *rs3* formatuan gordez. Hala ere, era jarraituan lortzen den *rs3*tik bertan adierazita dagoen zuhaitz-diagrama irudi batean gordetzea interesgarria izango litzateke, RSTTool tresna erabili gabe fitxategi bakoitzeko. Horretarako, programa bat gara liteke ataza hori egiteko.

Galdera-erantzun sistema

RSTko erlazioetatik galdera-erantzun sistema bat garatzea aproposa izan liteke. Adibidez, 3.1 irudiko *Elaboration* erlazioa aztertuta, norbaitek zer den laktosa galdetzen badu, bere sateliteko EDUa erantzun geniezaioke. RST zuhaitz-diagramak testuetatik sortzen dituen sistema on bat izatekotan, informazio horretaz baliatzen den sistema bat sortzea interesgarria izango litzateke.

Eranskinak

A. ERANSKINA

RST erlazioen taulak

Hona hemen euskarazko RST erlazioen arauak eta efektuak, [Iruskieta, 2014]tik aterata.

A.1 Aurkezpenezko erlazioak euskaraz

Aurkezpenezko erlazioen definizioak			
Erlazioa	**Arauak Sn eta Nn**	**Arauak S-Nn**	**Efektua**
Antitesia	N-n: idazleak N-rekiko aldeko iritzia du	N eta S aurkaritzako erlazioan daude eta bateraezinak dira; beraz, ezinezkoa da biekiko (N eta S) aldeko iritzia edukitzea. S-ren aldeko iritzia izatean, ezin da N-ren aldeko iritzia izan. Aurkaritza erlazio horretan, irakurleak N-rekiko duen aldeko iritzia handitzen du	Bateraezinak diren bi egoeren aurrean irakurlearen iritzi positiboa handitzen da N-rekiko

Testuingurua	N-n: S irakurri arte irakurleak ez du N ulertuko guztiz	S irakurtzeak N edo N-ko elementuren bat hobeto ulertzea dakar	N hobeto ulertzeko irakurlearen aldeko iritzia handitzen da
Kontzesioa	N-n: idazleak N-rekiko aldeko iritzia du; S-n: idazleak ez du erakusten S onartzen ez duenik	Idazlearentzat N eta S onargarriak izan badaitezke ere, bien artean aurkakotasunak egon daitezkeela onartzen du; irakurleak aurkakotasunezko egoera hori onartzean, N-rekiko aldeko iritzia handitzen du	N eta S onargarriak izan arren, aurkakotasunak daude bi egoeretan. Aurkakotasun horrek irakurlearengan N-rekiko aldeko iritzia handitzea dakar

Ahalbideratzea	N-n: gauzatu gabeko ekintza bat aurkezten da N-n irakurleari zuzendua (eskaintza baten onarpena barnean duela); S-n: Irakurleak S ulertzean N-n aurkezten den gauzatu gabeko ekintza gauzatzeko aldeko iritzia handitzen du	Proposaturiko ekintza gauzatzeko irakurlearen aldeko iritzia handitzen da	N gauzatu gabe dago eta S-k N gauzatzeko modua erakusten du
Evidentzia	N-n: gerta liteke irakurleak ez izatea froga nahikorik N-rekiko aldeko iritzia izateko; S-n: irakurleak S-rekiko aldeko iritzia du.	Irakurleak S ulertzean N-rekiko aldeko iritzia handitzen du	Irakurleak S-n ditu frogak N sinesteko edo N-rekiko aldeko iritzia handitzeko
Justifikazioa	Ez dago baldintzarik	Irakurleak S ulertzean, idazleak N aurkezteko egokitasuna areagotzen da	Irakurleak idazleari N aurkezteko egokitasuna onartzen dio

Motibazioa	N-n: N gauzatu gabeko ekintza da eta irakurlea ekintzaren egilea (eskaintza baten onarpena ere badago)	S ulertzeak irakurleari N-n proposatu zaion ekintza egiteko aldeko iritzia edo nahia handitzen dio	N-n proposaturiko ekintza egiteko irakurlearen gogoa handitzen du
Prestatzea	Ez dago baldintzarik	Testuan S dago N-ren aurretik; S-rekin irakurleari N-n dagoen informazioa aurreratzen edo interesa pizten dio idazleak	Irakurleari N irakurtzeko, interesa, prestutasuna edota orientazioa handitzen zaio
Birformulazioa	Ez dago baldintzarik	S-rekin N-n dagoena beste era batera formulatzen da. Testuaren tamainari dagokionez, S eta N tamaina berekoak dira; baina nukleartasunari dagokionez, N idazlearentzat S baino garrantzitsuagoa da	Irakurleak S N-ren bestelako formulaziotzat hartzen du
Laburpena	N-n:unitate bat baino gehiagoz osatuta egon behar da	N-n idatzitakoaren sintesia agertzen da S-n; beraz, S-ko informazioa N-koa baino laburragoa da	Irakurleak S-n dagoena N-n dagoenaren laburpena dela onartzen du

A.1 Taula: Euskarazko aurkezpenezko erlazioen arauak eta efektuak

A.2 Edukizko erlazioak euskaraz

Edukizko erlazioen definizioak			
Erlazioa	Arauak Sn eta Nn	Arauak S-Nn	Efektua
Zirkunstantzia	S-n: S gauzatuta dago	Irakurleak S-n deskribatutako zirkunstantzietan interpretatu behar du N	N interpretatzeko zirkunstantzia S-k ematen diola onartzen du irakurleak
Baldintza	S-n: S egoera hipotetikoa, etorkizunekoa edo gauzatu gabekoa da	N gauzatuko da, baldin eta S gauzatzen bada	N S-k baldintzatzen duela onartzen du irakurleak
Elaborazioa	Ez dago baldintzarik	N-n aurkeztutako gaiaren edo egoeraren ezaugarriren bat garatzen da S-n edo N-tiko inferentzia aurkezten da S-n, erlazio hauen arabera: multzoa :: kidea; abstraktua :: adibidea; osoa :: zatia; prozesua :: urratsa; objektua :: atributua; orokorra :: espezifikoa	S-n aurkeztutako egoerak N-ko ezaugarriren bat garatzen duela onartzen du irakurleak. Irakurleak garatutako elementua edo gaia identifikatzen du

Ebaluazioa	Ez dago baldintzarik	Idazleak N-rekiko duen aldeko iritzia aurkezten du S-k	S-n N ebaluatzen dela onartzen du irakurleak
Interpretazioa	Ez dago baldintzarik	Idazleak N-n ez dauden ideiak erlazionatzen ditu S-rekin	N-n ez dauden ideia-multzoak S-rekin erlazioa duela onartzen du irakurleak
Metodoa	N aktibitatea da	S-n metodoa edo instrumentua aurkezten da, zeinaren bidez N egikaritzen den	Irakurleak onartzen du S-n aurkezturiko metodoak edo instrumentuak N posible egiten duela
Kausa	N-n: N-n arrazoia aurkezten da	N gauzatzeko arrazoia S-n agertzen da; S aurkeztu gabe irakurleak ezingo luke jakin zergatik gertatu den N; N S baino garrantzitsuagoa da idazlearen helburuetarako	N-n gertatzen den egoeraren kausa S dela onartzen du irakurleak
Ondorioa	S-n: S-n ekintza edo egoera sortu den arrazoia aurkezten da	N-k eragin zezakeen S; S baino garrantzitsuagoa da N idazlearen helburuetarako	Irakurleak onartzen du N dela S-ren kausa edo S dela N-ren ondorioa

Aukera	N-n: N gauzatu gabeko egoera da; S-n: S gauzatu gabeko egoera da.	N gauzatzeak S gauzatzea galarazten du	N-ren gauzatzeak S-ren ez gauzatzearekin duen mendeko erlazioa onartzen du irakurleak
Helburua	N-n: N aktibitatea da; S-n: S gauzatu ez den egoera da	N-ko aktibitatearekin gauzatuko da S	N-ko aktibitatea S gauzatzeko egin dela onartzen du irakurleak
Arazo-soluzioa	S-n: S-n arazoa aurkezten da	N da S-n aurkezturiko arazoaren soluzioa	S-n aurkezturiko arazoaren soluzioa N dela onartzen du irakurleak
Ez-baldintzatzailea	S-n: baliteke S-k N-ren egikaritzean eragitea	S-k ez du N baldintzatzen	Irakurleak onartzen du N ez duela S-k baldintzatzen
Alderantzizko Baldintza	Ez dago baldintzarik	N gauzatzen da baldin eta ez bada S gauzatzen	N gauzatuko dela onartzen du irakurleak baldin eta bakarrik S ez bada gauzatzen

A.2 Taula: Euskarazko edukizko erlazioen arauak eta efektuak

A.3 Multinuklear erlazioak euskaraz

Erlazio nukleoaniztunen definizioak		
Erlazioa	Arauak nukleo bakoitzean	Efektua
Konjuntzioa	N guztiek osotasun bat osatzen dute. Osotasun horretan N baten rola beste N guztiekin konparagarria da.	N-ek osotasun bat osatzen dutela eta elkarren artean erlazionaturik daudela ezagutzen du irakurleak
Kontrastea	Bi N daude, ez gehiago; N horien arteko erlazioak honakoak izan daitezke: a) antzekotzat ulertzen dira zenbait ezaugarritan, b) ezberdintzat ulertzen dira zenbait ezaugarritan, c) ezberdintasuna da beste antzeko ezaugarriekin konparatzen dena.	Aurkaritzazko konparazioaren berdintasunak eta ezberdintasunak onartzen ditu irakurleak
Disjuntzioa	N bat beste(ar)en alternatibatzat (ez derrigorrez esklusiboa) aurkezten da	N guztiak alternatiboak direla onartzen du irakurleak
Bateratzea	Ez dago baldintzarik	Ez dago baldintzarik
Lista	N guztiek elkarren artean ezaugarriren bat konpartitzen dute eta, gainera, N guztiek zerrenda bat osatzen dute	Zerrenda bateko elementuak direla ezagutzen du irakurleak

Birformulazio nukleoaniztuna	N bat berregiten da beste N batekin eta idazlearen helburuetarako bi N horien garrantzia maila berekoa da	Berregindako N-ek garrantzia bera dutela ezagutzen du irakurleak
Sekuentzia	N guztien artean segida erlazioa dago	N guztien arteko segida erlazioa ezagutzen du irakurleak

A.3 Taula: Euskarazko multinuklear erlazioen arauak eta efektuak

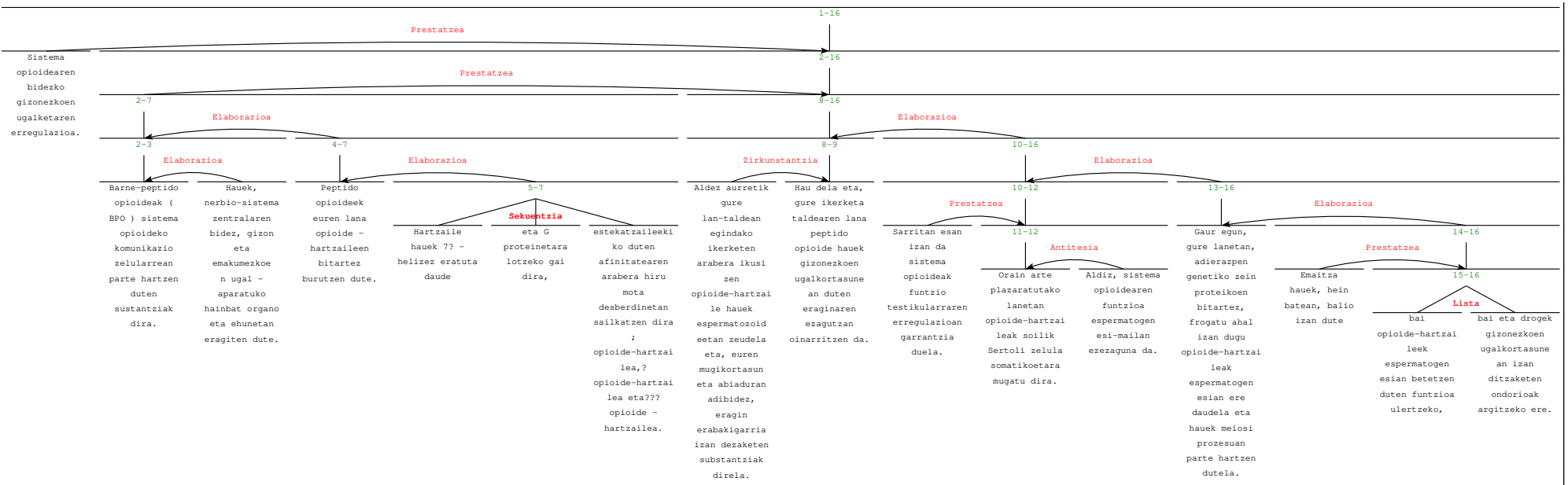
B. ERANSKINA

Ereduek iragarritako RST zuhaitz adibidea

Adibide gisa, erabili den testua B.1 Zerrendan irakur daiteke. Eskuz etiketatutako RST zuhaitza B.1 Irudian ikus daiteke. 6 Atalean aurkeztutako A, E, G eta H ereduek iragarritako RST zuhaitza B.2 Irudian, B.3 Irudian, B.4 Irudian eta B.5 Irudian, hurrenez hurren.

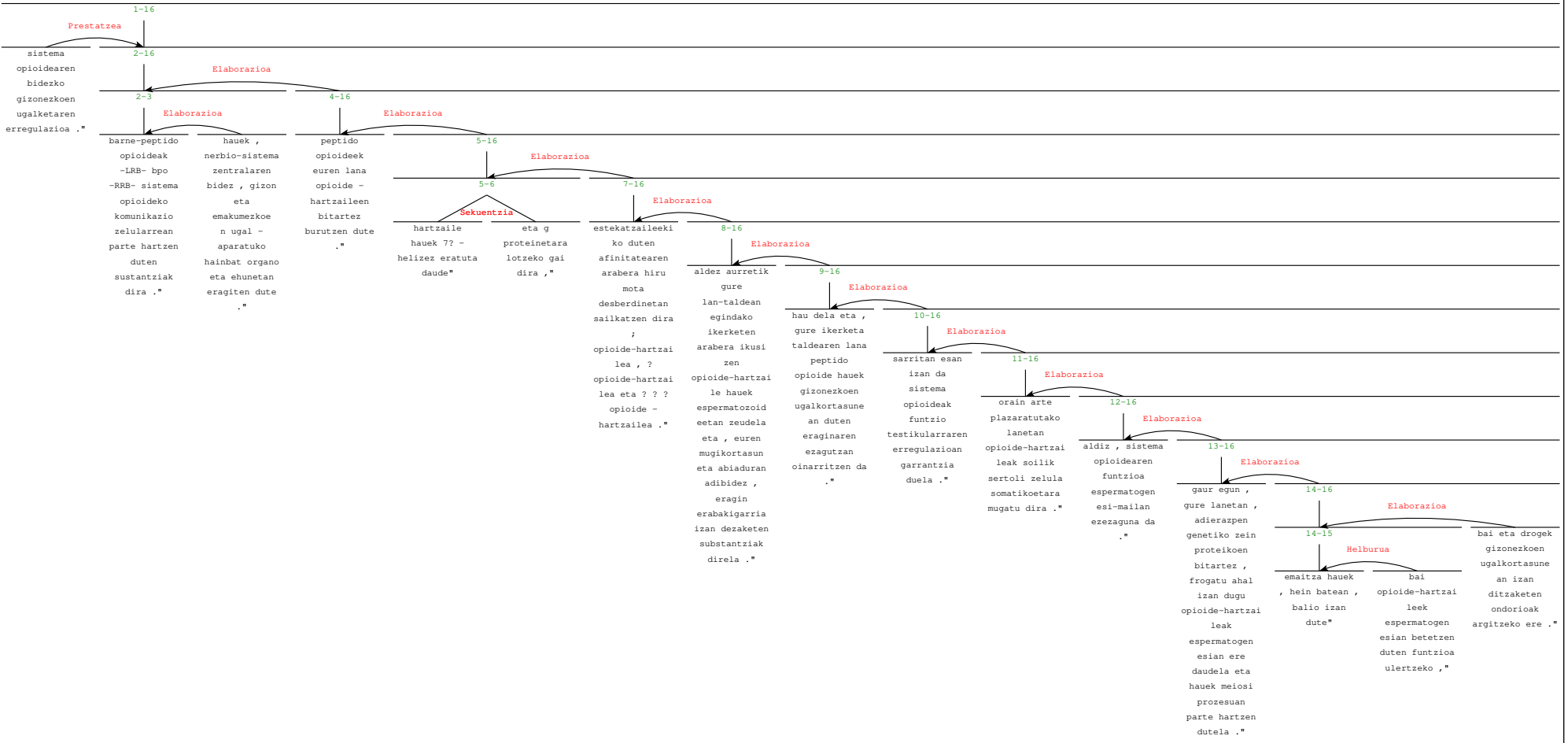
B.1 Zerrenda: OSA12 testua

```
1 Sistema opioidearen bidezko gizonezkoen ugalketaren erregulazioa. Barne-peptido opioideak ( BPO )
      sistema opioideko komunikazio zelularrean parte hartzen duten sustantziak dira. Hauek,
     nerbio-sistema zentralaren bidez, gizon eta emakumezkoen ugal - aparatuko hainbat organo eta
      ehunetan eragiten dute. Peptido opioideek euren lana opioide - hartzaileen bitartez
     burutzen dute. Hartzaile hauek 7? - helizez eratuta daude eta G proteinetara lotzeko gai
     dira, estekatzaileekiko duten afinitatearen arabera hiru mota desberdinetan sailkatzen dira
     ; opioide-hartzailea,? opioide-hartzailea eta??? opioide - hartzailea. Aldez aurretik gure
     lan-taldean egindako ikerketen arabera ikusi zen opioide-hartzaile hauek espermatozoideetan
     zeudela eta, euren mugikortasun eta abiaduran adibidez, eragin erabakigarria izan dezaketen
     substantziak direla. Hau dela eta, gure ikerketa taldearen lana peptido opioide hauek
     gizonezkoen ugalkortasunean duten eraginaren ezagutzan oinarritzen da. Sarritan esan izan da
      sistema opioideak funtzio testikularraren erregulazioan garrantzia duela. Orain arte
     plazaratutako lanetan opioide-hartzaileak soilik Sertoli zelula somatikoetara mugatu dira.
     Aldiz, sistema opioidearen funtzioa espermatogenesi-mailan ezezaguna da. Gaur egun, gure
     lanetan, adierazpen genetiko zein proteikoen bitartez, frogatu ahal izan dugu opioide-
     hartzaileak espermatogenesian ere daudela eta hauek meiosi prozesuan parte hartzen dutela.
     Emaitza hauek, hein batean, balio izan dute bai opioide-hartzaileek espermatogenesian
     betetzen duten funtzioa ulertzeko, bai eta drogek gizonezkoen ugalkortasunean izan
     ditzaketen ondorioak argitzeko ere.
```

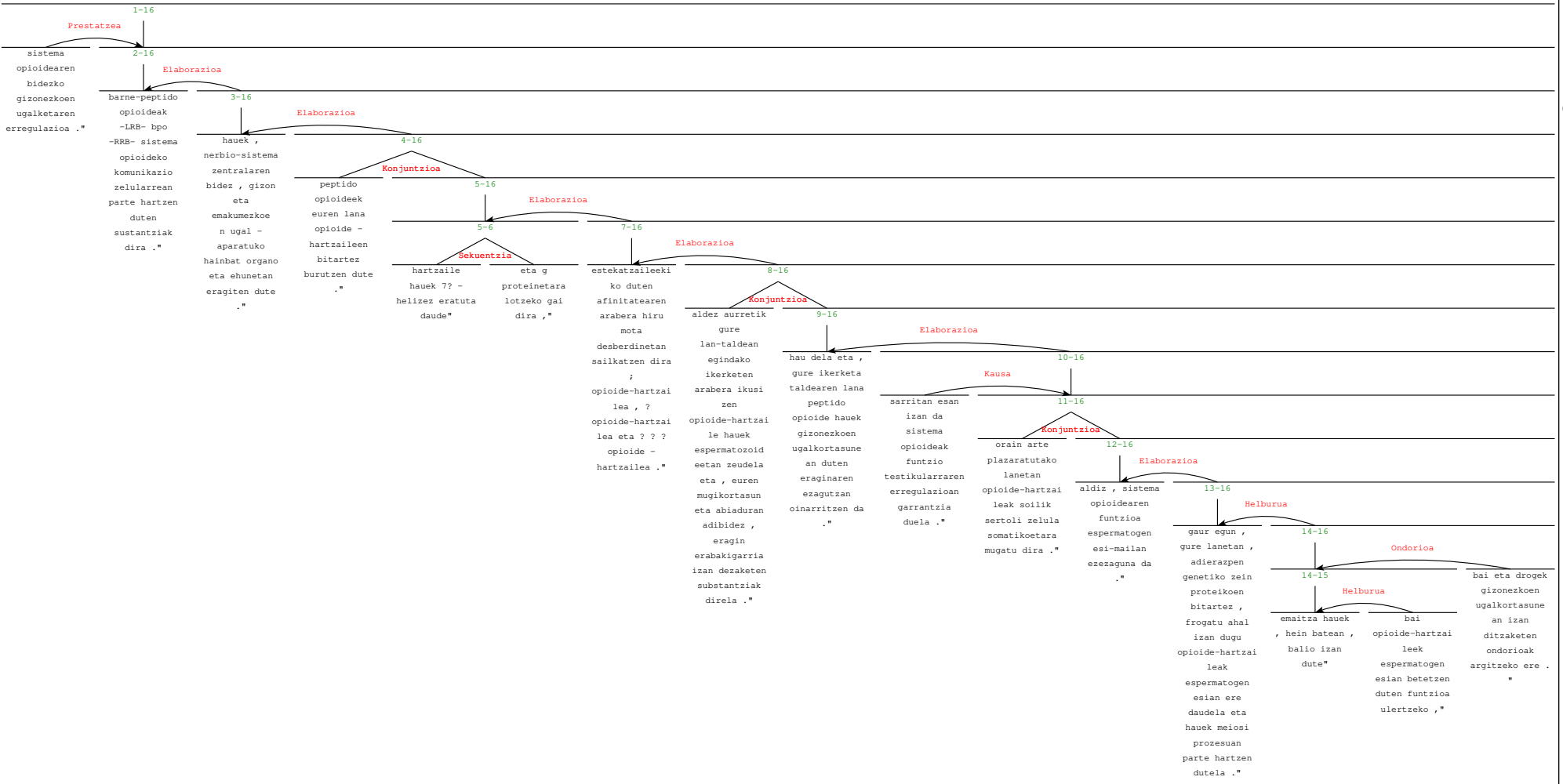


B.1 Irudia: Eskuz etiketatutako RST zuhaitza: OSA12 testua.

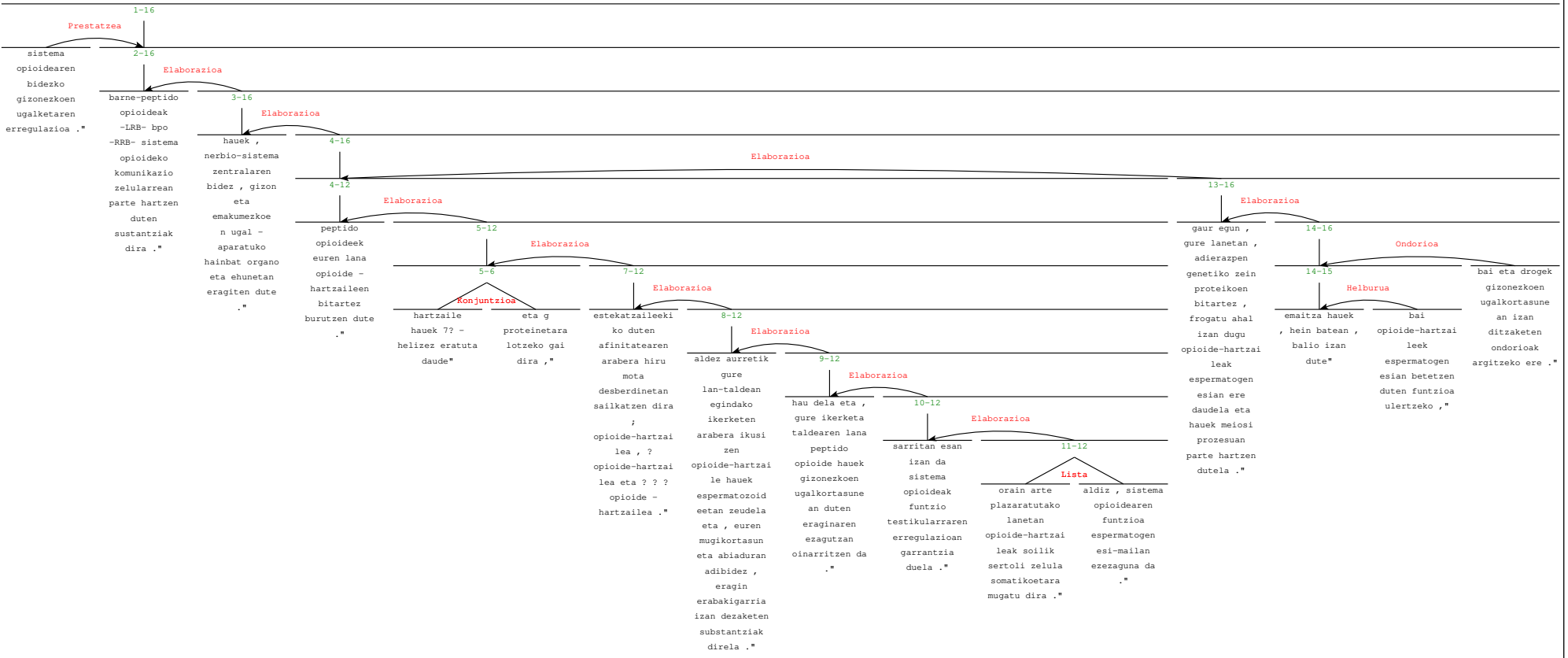
B.2 Irudia: A modeloak iragarritako RST zuhaitza: OSA12 testua.



B.3 Irudia: E modeloak iragarritako RST zuhaitza: OSA12 testua.



B.4 Irudia: G modeloak iragarritako RST zuhaitza: OSA12 testua.



B.5 Irudia: H modeloak iragarritako RST zuhaitza: OSA12 testua.

Bibliografia

[Alkorta et al., 2019] Alkorta, J., Gojenola, K., and Iruskieta, M. (2019). Towards discourse annotation and sentiment analysis of the Basque opinion corpus. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 144–152, Minneapolis, MN. Association for Computational Linguistics.

[Atutxa et al., 2019] Atutxa, A., Bengoetxea, K., Diaz de Ilarraza, A., and Iruskieta, M. (2019). Towards a top-down approach for an automatic discourse analysis for basque: Segmentation and central unit detection tool. *PLOS ONE*, 14(9):1–25.

[Atutxa et al., 2021] Atutxa, U., Molina-Villegas, A., and Iruskieta Quintian, M. (2021). Generación automática de meta-resúmenes para la evaluación del manejo de estructuras discursivas y coherencia en el alumnado. *Procesamiento del Lenguaje Natural*, pages 165–175.

[Braud et al., 2017] Braud, C., Coavoux, M., and Søgaard, A. (2017). Cross-lingual rst discourse parsing.

[Braud et al., 2016a] Braud, C., Plank, B., and Søgaard, A. (2016a). Multi-view and multi-task training of RST discourse parsers. In *Conference on Computational Linguistics (CoLing)*, pages 1903 – 1913, Osaka, Japan.

[Braud et al., 2016b] Braud, C., Plank, B., and Søgaard, A. (2016b). Multi-view and multi-task training of RST discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913, Osaka, Japan. The COLING 2016 Organizing Committee.

[Dargan et al., 2019] Dargan, S., Kumar, M., Ayyagari, M. R., and Kumar, G. (2019). A survey of deep learning and its applications: A new paradigm to machine learning. pages 1–22.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

[Faghfouri and Frish, 2011] Faghfouri, A. and Frish, M. (2011). Robust discrimination of human footsteps using seismic signals. *Proc SPIE*.

[Feng et al., 2017] Feng, W., Guan, N., Li, Y., Zhang, X., and Luo, Z. (2017). Audio visual speech recognition with multimodal recurrent neural networks. pages 681–688.

[Frazier, 1979] Frazier, L. (1979). On comprehending sentences: Syntactic parsing strategies. *ETD Collection for University of Connecticut*.

[Fu et al., 2016] Fu, X., Liu, W., Xu, Y., Yu, C., and Wang, T. (2016). Long short-term memory network over rhetorical structure theory for sentence-level sentiment analysis. In Durrant, R. J. and Kim, K.-E., editors, *Proceedings of The 8th Asian Conference on Machine Learning*, volume 63 of *Proceedings of Machine Learning Research*, pages 17–32, The University of Waikato, Hamilton, New Zealand. PMLR.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

[Iruskieta, 2014] Iruskieta, M. (2014). *Pragmatikako erlaziozko diskurtso-egitura: deskribapena eta bere ebaluazioa hizkuntzalaritza konputazionalean*. PhD thesis.

[Iruskieta et al., 2013] Iruskieta, M., Aranzabe, M., Diaz de Ilarraza, A., Gonzalez-Dios, I., Lersundi, M., and Lopez de Lacalle, O. (2013). The rst basque treebank: an online search interface to check rhetorical relations. In *IV Workshop A RST e os Estudos do Texto*, page 40–49, Fortaleza, CE. Sociedade Brasileira de Computaçao. Outubro 21-23.

[Iruskieta and Braud, 2019] Iruskieta, M. and Braud, C. (2019). EusDisParser: improving an under-resourced discourse parser with cross-lingual data. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 62–71, Minneapolis, MN. Association for Computational Linguistics.

[Jain et al., 1996] Jain, A., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: a tutorial. *Computer*, 29(3):31–44.

[Kraus and Feuerriegel, 2017] Kraus, M. and Feuerriegel, S. (2017). Sentiment analysis based on rhetorical structure theory: Learning deep neural networks from discourse trees. *Expert Systems with Applications*, 118.

[Kusmartsev and Kusmartsev, 2016] Kusmartsev, V. and Kusmartsev, F. (2016). Modelling a network where the opinion of each unit varies according to a majority ruling of its neighbouring units.

[Mann and Thompson, 1988] Mann, W. and Thompson, S. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text, 8 (3)*, pages 243–281.

[Marcu, 2000a] Marcu, D. (2000a). The rhetorical parsing of unrestricted texts: a surface-based approach. *Computational Linguistics*, 26(3):395–448.

[Marcu, 2000b] Marcu, D. (2000b). *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[Nwankpa et al., 2018] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning.

[O'Donnell, 2000] O'Donnell, M. (2000). Rsttool 2.4: A markup tool for rhetorical structure theory. In *Proceedings of the First International Conference on Natural Language Generation - Volume 14*, INLG '00, page 253–256, USA. Association for Computational Linguistics.

[Popoola, 2017] Popoola, O. (2017). Using Rhetorical Structure Theory for detection of fake online reviews. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 58–63, Santiago de Compostela, Spain. Association for Computational Linguistics.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408.

[Sagae and Lavie, 2005] Sagae, K. and Lavie, A. (2005). A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132, Vancouver, British Columbia. Association for Computational Linguistics.

[Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681.

[Skoufaki, 2020] Skoufaki, S. (2020). Rhetorical structure theory and coherence break identification. *Text and Talk*, 40:99–124.

[Straka and Strakova, 2017] Straka, M. and Strakova, J. (2017). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. pages 88–99.

[Taboada and Mann, 2006a] Taboada, M. and Mann, W. (2006a). Applications of rhetorical structure theory. *Discourse Studies - DISCOURSE STUD*, 8:567–588.

[Taboada and Mann, 2006b] Taboada, M. and Mann, W. (2006b). Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies - DISCOURSE STUD*, 8.

[Wang and Raj, 2017] Wang, H. and Raj, B. (2017). On the origin of deep learning.

[Yu et al., 2018] Yu, N., Zhang, M., and Fu, G. (2018). Transition-based neural RST parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[Yuan et al., 2019] Yuan, X., Li, L., and Wang, Y. (2019). Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, PP:1–1.