

Informatika Ingeniaritzako Gradua Konputazioa

Gradu Amaierako Lana

Bi iturriko itzultzaile neuronal

Egilea

Bittor Alkain Genua

2020

Informatika Ingeniaritzako Gradua Konputazioa

Gradu Amaierako Lana

Bi iturriko itzultzaile neuronal

Egilea

Bittor Alkain Genua

Zuzendariak

Gorka Labaka Intxauspe, Olatz Perez de Viñaspre Garralda, Ander Soraluze Irureta

Laburpena

Lan honetan bi iturriko itzultzaile neuronal bat eraiki dut, *transformer* arkitektura erabiliz. Itzultzaileak testu beraren gaztelaniazko eta ingelesezko bertsioak jaso behar ditu, bi iturriak konbinatuz euskarazko itzulpena sortzeko. Corpus bereko datuekin entrenatutako iturri bakarreko sistemen aurrean, emaitza hobeak ematen ditu, eta itzultzaile arrunt batek hizkuntzaren anbiguitasunekin izan ditzakeen arazoetako batzuk konpontzen ditu.

Gainera, datuen aurreprozesaketaren barruan, corpus multiparalelo bateko parekatzeak hobetzeko metodo bat sortu dut, LASER hizkuntza anitzeko esaldi-bektoreak erabiliz. Hala ere, ez ditut espero bezain emaitza onak lortu, eta metodoaren erabilera ez da lagungarria izan itzultzailea entrenatzeko orduan.

Gaien aurkibidea

Laburpena	i
Gaien aurkibidea	iii
Irudien aurkibidea	v
Taulen aurkibidea	vii
1 Sarrera	1
2 Aurrekariak	5
2.1 Gaztelania-ingelesa-euskara corpusak	5
2.2 Tokenizazioa eta BPE	6
2.3 Transformer neurona-sarea	7
2.3.1 Bi iturriko transformer-a	10
2.3.2 Inferentziarako erabiltzea	11
2.4 LASER esaldi-bektoreak	12
3 Datuen azterketa eta aurreprozesaketa	13
3.1 Hizkuntzen arteko Corpora	13
3.1.1 Datuak eskuratu eta garbitzea	13
3.1.2 Parekatze-arazoak	15

3.1.3	Parekatzeak hobetzeko metodoa	17
3.2	OpenSubtitles	26
3.3	Tokenizazioa	29
4	Ikasketa-prozesua	33
4.1	Entrenamendua	33
4.2	Ebaluazioa	34
5	Esperimentuak	37
5.1	Parekatzeak hobetzeko metodoa	37
5.2	Iturri bateko eta bi iturriko ereduen konparazioa	39
5.3	Batua.eus itzultzailearekin konparazioa	40
6	Ondorioak eta etorkizunerako lana	47
6.1	Etorkizunerako lana	48
Eranskinak		
A	Proiektuaren kudeaketa	53
A.1	Helburuak	53
A.2	Atazak	54
A.2.1	Atazen deskribapena	54
A.2.2	Atazen garapen-denboraldiak	55
A.2.3	Atazei emandako denbora	56
A.3	Informazio- eta komunikazio-sistemak	56
A.3.1	Informazio-sistema	56
A.3.2	Komunikazio-sistema	57
Bibliografia		59

Irudien aurkibidea

2.1	Transformer-aren egitura orokorra.	8
2.2	Kodetzaileko eta deskodetzaileko geruza bakoitzaren egitura.	9
3.1	HACeko hutsuneen denbora-lerroa.	15
3.2	HACeko gaztelania-ingelesa parerako, hitz kopuruaren (bikoteko maximoa) eta batez besteko antzekotasunaren arteko erlazioa.	24
3.3	Hainbat luzeratarako aukeratutako atalaseak eta balioak hurbiltzeko kurba.	26
3.4	Metodoaren funtzionamenduaren eskema, kontrol-puntuak eta arazoek bihurtutako gehituta.	26
5.1	Epoka bakoitzaren ondorengo ebaluazioaren BLEU puntuazioak, HACeko jatorrizko datuekin egindako entrenamenduan, corpus bakoitzeko ebaluazio-multzorako eta multzo elkaturako.	38
5.2	Epoka bakoitzaren ondorengo ebaluazioaren BLEU puntuazioak, bi corpusetako datuekin eta bi iturriekin egindako entrenamenduan, corpus bakoitzeko ebaluazio-multzorako eta multzo elkaturako.	40
A.1	LDE diagrama.	54
A.2	Gantt diagrama.	58

Taulen aurkibidea

2.1	Hizkuntzen arteko Corpuseko bi segmentu-hirukote.	6
3.1	Segmentu huts bat duen kasua.	16
3.2	Elkarketarik behar ez duen egoera.	18
3.3	Segmentu bati dagokion testua (B) falta den egoera.	18
3.4	Hobekuntza-metodoa diseinatzeko probak.	20
3.5	Jatorrizko eta esaldikako banaketan bi adibide.	21
3.6	Euskarazko hiru lerro elkartu behar diren kasua.	22
3.7	Bi hizkuntzatan bi lerro elkartu beharreko egoera.	22
3.8	Margin-based score-ekin muturreko kasu batzuk, non gaizki parekatutako bikoteek ondo parekatutakoek baino puntuazio askoz altuagoak dituzten.	25
3.9	Elkartu beharreko bi lerro.	27
5.1	Proba bakoitzean entrenatzeko erabilitako instantzia kopurua.	38
5.2	HAC corpusarekin bakarrik entrenatuta lortutako emaitzak.	38
5.3	Iturri bateko eta bi iturriko ereduen emaitzak, bi corpusekin entrenatuta.	39
5.7	Prensa-oharretatik hartutako esaldiak.	42
5.4	Hizkuntzetako batean anbiguoak diren esaldi-bikoteak eta nire ereduaren itzulpenak.	44
5.5	Nire eredu onenaren eta Batua.eus itzultzailearen BLEU puntuazioak, corpus bakoitzeko ebaluazio-multzorako eta multzo elkarturako.	45

5.6	OpenSubtitles-eko ebaluazio-multzoko kasu batzuk, bi iturri erabiltzeari esker zuzen itzultzen direnak.	45
A.1	Ataza bakoitzerako, hasieran estimatutako denbora eta benetan emandakoa.	56

1. KAPITULUA

Sarrera

Azken hamarkadan, neurona-sare artifizialek aurrerapauso handiak ekarri dituzte adimen artifizialaren arlo askotan. Sare horiek, ataza bat egiteko, adibide kopuru handietatik automatikoki ikasten dute, aditu batek eskuz ezarritako arauetan oinarritu ordezt. Itzulpen automatikoaren kasuan, hizkuntzaren prozesamenduko beste ataza askotan bezala, *recurrent neural network* edo neurona-sare errepikariekin eta horien aldaerek izan zuten lehenik arrakasta. Azken urteetan, berriz, *transformer* izeneko arkitekturarekin lortu dira aurrerapen handienak, testuzko sekuentziak irakurri eta sortzeko oso eraginkorra baita.

GrAL honen abiapuntua 2019ko udan egin nuen lana da, IXA taldeak eskainitako praktiketan. Orduan landutako ataza postedizio automatikoa zen, eta bi iturriko transformer bat inplementatu nuen artearen egoerako sistema bat erreplikatzeko. Oraingo honetan, idatzitako kode hura beste datu batzuekin erabili dut, sareak bi iturriko itzulpena egiten ikas dezan. Itzulpen hauetan, sarrera gisa jatorrizko testua hizkuntza bakarrean jaso beharrean, bi hizkuntzatan jasotzen du sistemak, hau da, esaldi bererako bi iturri ditu. Adibidez:

- 1. sarrera: *Hemos viajado toda la noche.*
- 2. sarrera: *We've been traveling all night.*
- Irteera: *Gau osoan bidaiatu dugu.*

Horrelako sistema bat, lehen aldiz, [Och and Ney, 2001] artikulurako eraiki zuten. Itzul-tzaile estatistiko bat zen, ohikoena garai hartan. Artikuluan azaltzen duten moduan, iturri bat baino gehiago edukitzeak itzulpenaren kalitatea hobetu dezake. Adibidez, hitzen

adiera-desanbiguazioa errazten du: askotan, bi hizkuntzaren artean itzultzeko ebatzi beharreko anbiguotasunak ez dira existitzen beste hizkuntza batzuen artean. Esaterako, gaztelaniazko *nadie conocía el destino del avión secuestrado* euskarara itzultzeko, *destino* hori *helmuga* ala *patua* den erabaki behar da. Aldiz, jatorrizkoa ingelesezkoa bada, *destination* eta *fate* bereiziko dira. Azken urteetan egin diren hainbat lanetan erakutsi dute itzultzaile neuronaletan ere lagungarria dela jatorrizko hizkuntza bat baino gehiago erabiltzea [Zoph and Knight, 2016, Firat et al., 2016, Garmash and Monz, 2016, Dabre et al., 2017, Nishimura et al., 2018].

Mota honetako itzultzaileek izan dezakete benetako erabilpena. Izan ere, badira hizkuntza askotara itzuli ohi diren testuak: liburu ospetsuak; film, telesail edo bestelako bideoen azpтитuluak; nazioarteko enpresa eta erakundeen idatziak; erabiltzaile-interfazeak... Beraz, jatorrizko testuaz gain, itzulpen profesionalen bat dagoeneko eskuragarri dagoen kasuetan, bi iturriak aprobeitza daitezke beste itzulpen bat sortzeko.

Lan honetan, itzulpenetako hizkuntzak gaztelania, ingelesa eta euskara izatea erabaki dut, ezagutzen ditudan hizkuntzekin lan egiteak aukera ematen didalako datuak eta emaitzak sakonago aztertzeko, eta hizkuntza horietan behar adina datu aurkitu dudalako. Itzulpenaren noranzkoari dagokionez, euskara aukeratu dut helburuko hizkuntza bezala, eta beraz, gaztelania eta ingelesa iturri bezala. Izan ere, sortutako sistemaren praktikotasunari begira, aukera posibleetatik, ohikoagoa dirudi jada gaztelaniaz eta ingelesez dagoen testu bat euskarara itzuli nahi izateak.

[Zoph and Knight, 2016] lanean diotenez, iturri bakarreko itzulpenarekin konparatuta, hobekuntzak handiagoak dira jatorrizko hizkuntzak elkarren artean desberdinagoak direnean. Izan ere, elkar desanbiguatzeak aukera gehiago omen dute horrela. Aldiz, [Dabre et al., 2017] artikuluan, Indian hitz egiten diren 6 hizkuntza aztertuta, ondorioztatzen dute hobekuntzak handiagoak direla jatorrizko hizkuntzen antzekotasun lexikoa handiagoa denean. Beraz, zaila da auresaten zenbateraino lagunduko dioten gaztelaniak eta ingelesak elkarri euskarara itzultzeko atazan. Hori erakusteko, bi iturriko ereduaz gain, gaztelania-euskara eta ingelesa-euskara iturri bakarreko ereduak entrenatu ditut. Gainera, gaur egungo puntako itzultzaile batekin konparatu ditut emaitzak, zenbat hurbildu naizen ikusteko.

Ereduak entrenatu eta ebaluatzeak Hizkuntzen arteko Corpusetik eta OpenSubtitles corpusetik lortutako gaztelania-ingelesa-euskara hirukoteak erabili ditut. Bertako testua liburutatik eta azpтитuluetatik hartuta dagoenez, gehiena fikziozko lanetatik, entrenatutako ereduak antzeko testuak itzultzeko izango dira batez ere erabilgarriak. Esandakoaz gain, lana osatzeko, Hizkuntzen arteko Corpuseko datuen kalitatea hobetzen saiatu naiz, horre-

tarako metodo bat garatuz, jatorrizko datuak ez baitira guztiz fidagarriak.

Memoriaren egitura, hemendik aurrera, honakoa da: 2. kapituluaren aurrekariak daude, alegia, lanean zehar erabilitako datu, metodo eta tresna garrantzitsuenei buruzko alde zuruko azalpenak; 3. kapituluaren datuen azterketa eta erreproduzizioak azaltzen dira; 4. kapituluaren ikasketa-prozesua azaltzen da, ereduak entrenatzeko eta ebaluatzeko baldintzak zehaztuz; 5. kapituluaren egindako esperimenduak azaldu eta emaitzak komentatzen dira; azkenik, 6. kapituluaren lanaren ondorio orokorrak azaldu eta etorkizunerako geratzen den lana planteatzen da. Bukatzen da proiektuaren kudeaketari buruzko A. eranskina eta bibliografia.

2. KAPITULUA

Aurrekariak

Aurrekarien kapitulu honetan hauek aurkezten dira: erabilitako corpusak, tokenizaziorako teknikak, transformer neurona-sarea eta esaldien antzekotasuna neurtzeko LASER esaldi-bektoreak.

2.1 Gaztelania-ingelesa-euskara corpusak

Neurona-sareetan oinarritutako ereduak itzulpenak egiten ikasteko, datu kopuru handiak behar dira. Datu horiek testu berbera erakutsi behar dute gaztelaniaz, ingelesez eta euskaraz. Hortaz, corpus multiparaleloak behar dira, hau da, bi hizkuntza baino gehiagotan informazio bera ematen duten testu-multzoak. Horrelako datuak lortzeko aukera ematen duten bi corpus aurkitu ditut: Hizkuntzen arteko Corpora eta OpenSubtitles.

Corpus hauetan, testuak segmentu edo itzulpen-unitatetan banatuta daude, eta segmentu horiek hizkuntza desberdinen artean parekatuta daude. “Segmentu” deitzen diet, eta ez “esaldi”, ez dutelako zertan puntutik punturako esaldiak izan behar: izan daitezke esaldizatiak edo esaldi-multzo txikiak ere. Bi adibide daude [2.1](#) taulan.

Hizkuntzen arteko Corpora (HAC) [[Sarasola et al., 2015](#)] EHUko Euskara Institutuan sortutako corpora da. Bigarren bertsioa, 2017koa, lau hizkuntzatarako itzulitako 136 liburuz osatuta dago: euskarara, gaztelaniara, ingelesera eta frantsesera. Hala ere, lan honetarako ez ditut frantsesezko datuak erabili. Guztira, egileen arabera, 698.036 segmentu-laukote daude, eta horietako 624.988k (% 89,54) lau hizkuntzetako baliokideak dituzte. Gainera-

-El título de la redacción es: “The title of your composition is: -Zure idazlanaren izenburua hau da:
-¡Qué lección para ti, blanducho! Oímos también ruidos de pelea, de golpes, estruendo de sillas volcadas, una caída, gritos, jadeos. “What a lesson for you, you weakling!” We also hear the noise of a fight, blows, the crash of chairs being knocked over, a fall, shouts, panting. -Hori ikasgaia hiretzat, maskala halakoa! Liskar-hotsa entzuten dugu, eta danbatekoak, jaurtitako aulkien zartotsa, norbait erortzen, garrasiak, arnasestuak.

2.1 Taula: Hizkuntzen arteko Corpuseko bi segmentu-hirukote.

koei hizkuntzaren bateko baliokidea falta zaie. Datuak nola eskuratu eta garbitu ditudan azaltzen dut 3.1 atalean, baita segmentuen parekatzeak hobetzeko nire saiakera ere.

OpenSubtitles corpusa [Lison and Tiedemann, 2016], berriz, telesail eta filmetako azpitituluz osatuta dago, jendeak *opensubtitles.org* gunera igotakoak. Egileek azaltzen duten moduan, azpitituluen denbora-markei jarraituz daude parekatuta hizkuntza-pare guztietako esaldiak. 2018ko bertsioa erabili dut nik, lan hau egiteko momentuan dagoen azkena. Eskuragarri dauden fitxategietan ez daude gaztelania-ingelesa-euskara hirukoteak jada sortuta. Hortaz, horiek lortzeko, gaztelania-euskara bikoteetatik (793.593) eta ingelesa-euskara bikoteetatik (805.780) abiatu naiz. Aipatu nahi dut horiek direla corpusean euskarak dituen pare handienak, hurrengo Brasilgo portugesa - euskara (690.511 bikote) delarik. Orokorrean, esaldiak HACen baino motzagoak dira, eta elkarrizketa gehiago dago. Hirukoteak osatzeko eta datuak garbitzeko prozesua 3.2 atalean azaltzen dut.

2.2 Tokenizazioa eta BPE

Testua ereduari pasatu ahal izateko, lehenik tokenizatu eta indexatu egin behar da, hiztegi baten arabera. Hasteko, hitzak gainontzeko karaktereetatik banandu beharra dago, hau da, puntuazio-marka eta bestelako sinboloetatik. Horrela, sinbolo horiek hiztegiko indize propioa izango dute, eta sistemak behar bezala interpretatuko ditu. Gainera, hitzak minuskulaz jarri eta maiuskulak beste nolabait adieraztea komeni da. Izan ere, testuko maiuskulak eta minuskulak bere horretan utziz gero, hitz asko bi aldiz edo gehiago agertuko lirateke hiztegian: indize bat izango lukete esaldi hasieran maiuskulaz hasita agertzen direnerako, eta beste bat osorik minuskulaz agertzen direnerako, hiztegiko indizeak xahutuz eta sistemari pasatako informazioa ilunduz.

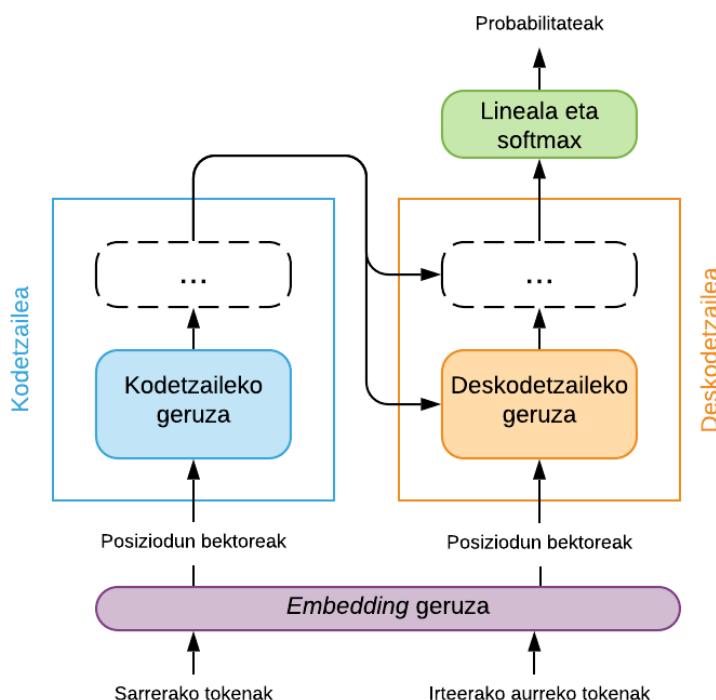
Behin testua hitzetan banatuta dagoela, hiztegia sortzeko modurik sinpleena hitz desberdin bakoitzari zenbaki bat esleitzea da. Hiztegiaren tamainari muga bat jartzen zaio, zenbakiak hitzen maiztasunaren arabera esleitzen dira, eta tamaina-mugaren ondorioz kanpoan geratzen diren hitz guztiei *hiztegiz kanpo* adierazten duen zenbaki bat esleitzen zaie. Bide hori ez da batere eraginkorra euskararen gisako hizkuntza eranskariekin aplikatzeko: alde batetik, lema baten forma bakoitzak (*handi*, *handitik*, *handiago*...) indize bat hartuz gero, hiztegia berehala beteko delako, eta bestetik, ikasketa zailtzen duelako. Izan ere, ereduak *etxe* eta *etxetik* hitzen arteko erlazioa ikasita ere, ez dio balioko *eskola* eta *eskolatik* hitzen artekoa argitzeko. Baina bestelako hizkuntzetan ere arazoak sortzen ditu: ikasitako ereduak ezingo ditu entrenamenduan ikusi ez dituen hitzak inola interpretatu, ezta sortu ere. Izen bereziak itzultzeko, esaterako, ohikoa litzateke hitz ezezagunak irakurri eta sortu beharra.

Horren aurrean, azpihitz bidez tokenizatzeko teknikak nagusitu dira azken urteetan, tartean nik erabili dudak BPE *byte pair encoding* edo byte-pare kodeketa. [Sennrich et al., 2016c] Teknika hau 90eko hamarkadan datu-konpresiorako asmatu zuten, baina hizkuntzaren prozesamendurako oso erabilgarria dela ikusi da. Algoritmoak, lehenik, hitzen hasieran karaktere berezi bat txertatzen du, kodeketa desagitea posible izateko. Ondoren, hiztegia ikasteko, testuko azpihitz ohikoenak binaka elkartzeko arauak sortzen ditu, karaktere solteetatik abiatuta, hiztegiaren tamaina bete arte. Horrela, hitz arruntenak hiztegian osorik agertuko dira, eta gainontzekoak, berriz, azpihitzetan banatuta. Honi esker, erabili beharreko hiztegiaren tamaina nabarmen murriztu daiteke, neurona-sarean parametro batzuk aurreztuz. Gainera, ez da token ezezagunik egongo, karaktere ezezagunik ez dagoen bitartean, behintzat.

Hortaz, edozein testu, BPE ereduaren bidez kodetuta, zenbaki-segida bihurtuko da, hori baita neurona-sareak behar duen formatua. Adibide bat ikus daiteke 3.3 atalaren bukaeran, erabilitako parametro zehatzen azalpenaren ondoren.

2.3 Transformer neurona-sarea

Transformer izeneko neurona-sarea [Vaswani et al., 2017] *seq2seq* edo sekuentziatik sekuentziarako eredua da, hau da, sekuentzia bat irakurri eta beste bat idazten du. Itzulpen automatikoaren kasuan, sekuentzia hori segmentu bati dagokion token-segida da, tokenizazioaren azalpenean esan bezala. Sareak kodetzaile-deskodematzaile forma du: zati batek sarrerako sekuentzia kodetzen du erabilgarriak zaizkion bektore batzuetan; beste zatiak,

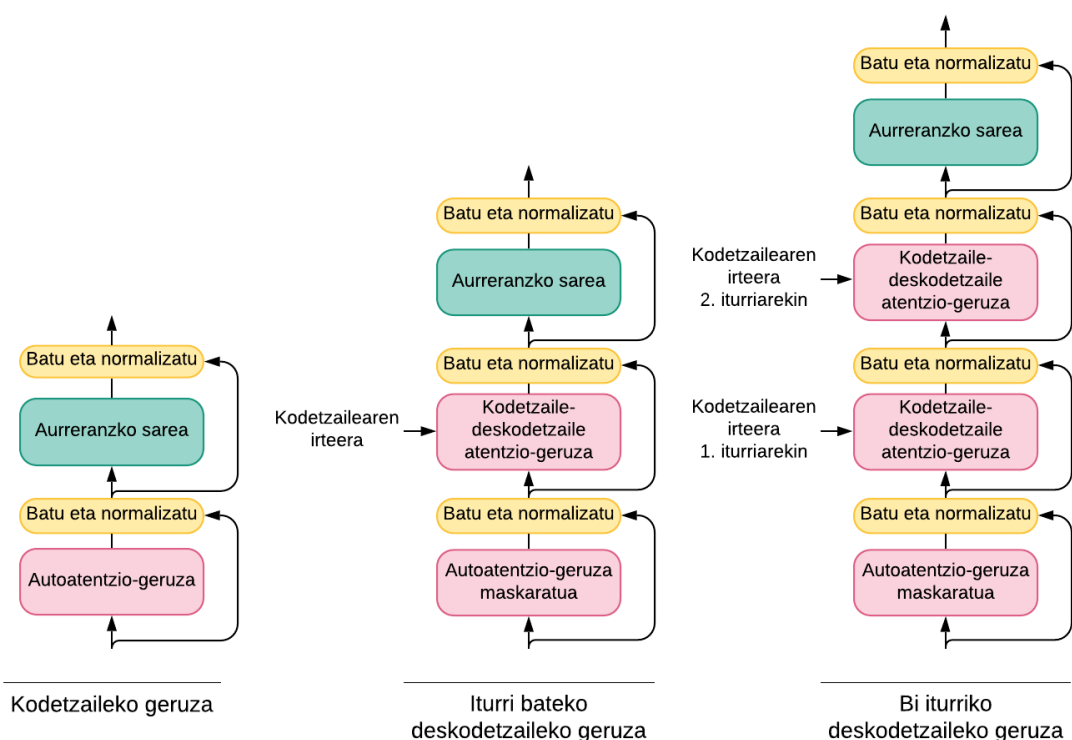


2.1 Irudia: Transformer-aren egitura orokorra.

berri, kodetutako bektore horiek hartu eta irteerako sekuentzia sortzen du. Horretarako, *self-attention* edo autoatentzio izeneko mekanismoa erabiltzen du behin eta berri sarean zehar. Egitura orokorra 2.1 irudian ikus daiteke.

Autoatentzio-geruza batek, labur esanda, bektore-multzo bat hartzen du, bektore bat sekuentziako posizio bakoitzeko, eta horietako bakoitza berriro kodetzen du, sekuentziako gainontzeko bektoreak kontuan izanda. Eragiketa horretan, bektore bakoitzeko, puntuazio bat kalkulatu da gainontzeko bektore bakoitzarentzat. Puntuazio horrek baldintzatzen du zenbateko pisua izango duten beste bektore horiek uneko posizioaren kodeketa berrian.

Gainera, autoatentzio-geruza arruntak erabili beharrean, **buru anitzeko atentzioa** erabiltzen du: autoatentzio-mekanismo bera behin baino gehiagotan aplikatzen da “buru” desberdinetan, hau da, parametro desberdinekin. Ondoren, buru guztietan lortutako bektoreak kateatu eta, matrize batekin biderkatuz, hasierako tamainako bektore bat lortzen da. Arkitektura honen sortzaileen arabera, buru bakoitzak ataza desberdin bat betetzen ikasten du, esaldien egitura sintaktiko eta semantikoarekin lotuta egon daitezkeenak.



2.2 Irudia: Kodetzaileko eta deskodetzaileko geruza bakoitzaren egitura.

Kodetzaileko geruza bakoitzak (2.2 irudiko ezkerrekoa) autoatentzio-geruza bat eta aurreranzko sare bat ditu (azken hori *feed-forward network* edo *multi-layer perceptron* ere deitua). Posizio bakoitzeko bektoreari independenteki aplikatzen zaio aurreranzko sare berbera. Kodetzailea horrelako hainbat geruzaz egon daiteke osatuta, bakoitzak sarrera bezala aurreko kodetzaile-geruzaren irteera hartzen duelarik.

Deskodetzaileak, azken kodetzailearen irteeraz gain, aurreko pausoetan idatzi dituen tokenak ere behar ditu sarrera gisa. Hortaz, deskodetzaileko geruza bakoitzak (2.2 irudiko erdikoak), lehenik, aurreko irteeren gaineko autoatentzio-geruza du. Horren emaitza kodetzaile-deskodetzaile atentzio-geruza batek erabiltzen du kodetzailearen irteeraren gaineko atentzioa kalkulatzeko. Kodetzailean bezalaxe, geruza bakoitzaren bukaeran aurreranzko sare bat dago, eta hainbat geruza bata bestearen ondoren jar daitezke deskodetzailea osatzeko.

Sarea entrenatzeko garaian, espero den irteera osorik pasatzen zaio deskodetzaileari, maskara bat aplikatuta eta tokenak posizio bat atzeratuta. Horrela, posizio bakoitzari dagozkion kalkuluak egitean, ezin du ikusi sortu beharreko hurrengo tokenen informaziorik.

Modu honetan, entrenamenduan, behin bakarrik exekututzen da deskodetzailea sekuentzia osorako. Aldiz, sistema inferentziarako erabiltzean, exekuzio bat behar du idazten duen token bakoitzeko.

Esandakoaz gain, kodetzailean zein deskodetzailean, atentzio-geruza eta aurreranzko sare bakoitzak *residual connection* edo **hondar-konexioa** du. Horrek esan nahi du geruzaren irteerari geruza beraren sarrera batzen zaiola. Ondoren, baturari **geruza-normalizazioa** aplikatzen zaio, hau da, posizio bakoitza adierazten duen bektorea normalizatzen da. Iru-dian “batu eta normalizatu” izena du pauso honek.

Sarrerako tokenak kodetzaileari pasatzeko, eta irteerakoak deskodetzaileari, lehenik bektore bihurtu behar dira, **embedding geruzan**. Geruza bera erabiltzen da bi kasuetan eta, beraz, itzulpenarako erabiltzean, sarrerako eta irteerako hizkuntzetarako hiztegi bera ikas-tea komeni da. Lortutako bektoreek dagozkien tokenen esanahia kodetu beharko lukete. Ondoren, bektore horiei sekuentzian duten posizioaren araberrako balio batzuk gehitzen zaizkie, *positional embeddings* edo posiziodun bektoreak lortzeko. Izan ere, hori gabe, sareak ez luke modurik tokenek sekuentzian duten ordena kontuan hartzeko.

Azkenik, **deskodetzailearen irteera** geruza lineal batetik (*single-layer perceptron*) pasatzen da, hiztegiaren tamainako irteera duena, softmax aktibazioarekin. Horrela, lortutako balioak probabilitateak dira: posizio bakoitzean, token mota posible bakoitza idatzi behar izateko probabilitatea. Eraitza hori, entrenamenduan, espero diren tokenekin konparatuko da, galera kalkulatu eta ereduaren parametroak doitzeko. Espero diren token horien bukaeran sekuentzia-bukaerako token berezi bat gehitzen da, ereduak deskodetketaren bukaera markatzen ikas dezan.

Transformer-ak, RNNek ez bezala, ez ditu tokenak banaka-banaka irakurri behar, aldi berean egin ditzake sekuentziako posizio desberdinei dagozkien kalkuluak. Horri esker, alde batetik, RNNek baino hobeto aprobetxatzen ditu gaur egungo GPUz osatutako gai-luak, kalkuluak paraleloan egiteko oso eraginkorrak direnak. Gainera, RNNek zailtasunak dituzte elkarrengandik urruti dauden tokenen arteko erlazioak behar bezala tratatzeko, informazioa “ahazteko” joera baitute. Transformer-ak, berriz, uneoro sekuentzia osoari dagokion informazioa erabil dezakeenez, ez du arazo hori.

2.3.1 Bi iturriko transformer-a

Sarreran azaldu bezala, lan honetan erabiltzen dudan transformer mota, bi iturrikoa, postedizio automatikoaren atazarako proposatu zuten [Juncys-Dowmunt and Grundkiewicz,

2018], ataza horretan bi informazio-iturri behar baitira: jatorrizko esaldia eta itzulpen akasduna. Oraingoan, berriz, arkitektura bera erabili dut, baina bi iturriak segmentu baten gaztelaniazko eta ingelesezko bertsioak dira, eta irteera, euskarazko bertsioa. Hortaz, ereduak ikasteko jasotzen duen instantzia bakoitza segmentu-hirukote bat da.

Bi iturriko transformer-ean, kodetzaileak ez du aldaketarik; gaztelaniazko eta ingelesezko segmentuekin kodetzaile bakar bat erabiltzen dut. Aldiz, deskodetzaileko geruza bakoitzean (2.2 irudiko eskuinekoa) beste kodetzaile-deskodetzaile atentzio-geruza bat gehitzen da. Lehen atentzio-geruza, deskodetzaile arruntean bezala, aurreko irteeren gainekoa da. Horren emaitza bigarren atentzio-geruzak erabiltzen du iturrietako baten kodeketaren gaineko atentzioa kalkulatzeko. Azkenik, hirugarrenak bigarrenaren emaitzarekin beste iturriaren kodeketaren gaineko atentzioa kalkulatu du. Hondar-konexioei esker, bi iturrien kodeketatik datorren informazioa irits daiteke deskodetzaile-geruza baten bukaerara.

Nire inplementazioa PyTorch-ekin¹ egin nuen, kodea idazteko eredu bat² jarraituta (momentu hartan PyTorch-ek ez zekarren *nn.Transformer* modulua). Ereduarekin alderatuta, deskodetzailea aldatu nuen bi iturrikoa izateko, eta embedding-geruza sarrera guztietarako partekatua izan zedin jarri nuen, PyTorch-en bertsio berrietan funtzionatzeko aldaketatxo batzuk egiteaz gain.

2.3.2 Inferentziarako erabiltzea

Sarea inferentziarako erabiltzeko, hau da, behin entrenatuta, beharrezko sarrera emanda irteerako sekuentzia oso bat lortzeko, modu desberdinetan egin daiteke deskodeketa.

Lehena *greedy* edo jalea da, sinpleena: pauso bakoitzean, deskodetzailea exekutatu eta irteeran probabilitate handiena lortu duen tokena aukeratzen da. Token berri bakoitzeko exekuzio bat egiten da, irteeran sekuentzia-bukaerako tokena aukeratzen den arte.

Beste aukera bat *beam search* erabiltzea da: pauso bakoitzean k sekuentzia hautagai daude (bilaketaren zabalera), eta horietako bakoitzari token bat gehitzeko aukera guztiak aztertzen dira. Aukera bakoitzaren probabilitatea eskuratzeko sekuentziako tokenek lortutako probabilitateen biderkadura kalkulatu da, eta k onenak aukeratzen dira prozesuarekin jarraitzeko. Praktikan, zenbakiak zerotik hurbilegi geratzea saihesteko, p probabilitateen biderkadurak maximizatu beharrean, $\log(p)$ balioen baturak minimizatzen dira, sailkapen

¹<https://pytorch.org/> Azken atzipena: 2020-08-19

²<https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec> Azken atzipena: 2020-06-29.

bera lortuko baita. Hautagai batek sekuentzia-bukaerako tokena badu, hurrengo pausoetan ez zaio tokenik gehituko, baina aukera bezala mantenduko da.

2.4 LASER esaldi-bektoreak

Sarreran aipatu bezala, HACeko datuen kalitatea hobetzen saiatu naiz, 3.1.3 atalean azaltzen dutan metodoaren bidez. Metodo horretan, hizkuntza desberdinetako segmentuen arteko antzekotasunak kalkulatzeko, aurretik entrenatutako neurona-sare bat erabili dut: Facebook Research-en LASER³ (*Language-Agnostic SEntence Representations*) [Artetxe and Schwenk, 2018] tresna. LASER-ek hizkuntza anitzeko kodetzailea du, 93 hizkuntza-tan entrenatua, tartean gaztelania, ingelesa eta euskara. 1024 elementuko *sentence embedding* edo esaldi-bektoreak sortzen ditu, alegia, esaldi oso baten esanahia tamaina finkoko bektore batean kodetu dezake. Bi esaldiren bektoreak eskuratuta, kosinu-antzekotasuna kalkulatu daiteke esaldien esanahiek antza duten edo ez jakiteko; berdin da bi esaldiak hizkuntza berean dauden ala ez.

Adibidez, hona hemen “*Gau osoan bidaiatu dugu.*” esaldiaren bektoreak beste batzuekin duen kosinu-antzekotasuna:

- *Hemos viajado toda la noche.* → 0,9314
- *We’ve been traveling all night.* → 0,8797
- *Amak gorriturik dauzka begiak.* → 0,2952
- *La llegada a casa de la abuela* → 0,4174
- *We arrive from the Big Town.* → 0,4667

Ikusten denez, esaldiaren itzulpenek askoz antzekotasun handiagoa lortzen dute besteek baino. Horregatik, nire kasuan, segmentuen parekatze posibleak baloratzeko erabili dut neurri hau.

Gehienetan, esaldi-bektoreak lortzeko, egileek argitaratutako LASER liburutegiaren ordez, *laserembeddings*⁴ Pythoneko liburutegia erabili dut, erabiltzeko askoz errazagoa delako. Neurona-sare berberak erabiltzen ditu, baina desberdintasun batzuk ditu inplemtazioan. Hizkuntza gehienetarako, sortzen dituen bektoreak jatorrizkoen berdin-berdinak dira.

³<https://github.com/facebookresearch/LASER> Azken atzipena: 2020-08-19

⁴<https://github.com/yannvgn/laserembeddings> Azken atzipena: 2020-08-19

3. KAPITULUA

Datuen azterketa eta aurreprozesaketa

Kapitulu honetan, corpus bakoitzeko datuen eskuraketa, azterketa eta garbiketa azaltzen dira, HACeko parekatzeak hobetzeko saiakera barne. Ondoren, tokenizazioari buruzko xehetasunak ematen dira.

3.1 Hizkuntzen arteko Corpusa

3.1.1 Datuak eskuratu eta garbitzea

Hizkuntzen arteko Corpuseko datuak eskuratzeko bi bide daude: alde batetik, HACen webgunean¹ kontsulta daiteke, nahiz eta ez duen osorik deskargatzeko aukerarik ematen; bestetik, OPUS proiektuak² corpusa testu-fitxategietan deskargatzeko aukera ematen du, baina gaztelania-euskara eta ingelesa-euskara hizkuntza-pareak independenteki daude gordeta. Bigarren honetatik abiatuta gaztelania-ingelesa-euskara hirukoteak sortzen saiatzean, bistan geratu zen hizkuntza-pare bakoitzean milaka segmentu falta direla, pareetako batean segmentu-bikote batzuk eta bestean beste batzuk. Horrek parekatzea zaildu eta datu-kopuru erabilgarria nabarmen txikitzen du. Adibidez, corpuseko liburu handiena, *Elizen Arteko Biblia*, osorik falta da gaztelania-euskara parean. Ingelesa-euskara parean 30.986 bikote hartzen ditu. Gainera, badaude bi pareetan falta diren liburu-zatiak, HACen webguneko bertsioarekin alderatuta. Guztira, gaztelania-euskara parean 609.912 bikote

¹<https://www.ehu.eus/ehg/hac/> Azken atzipena: 2020-06-30.

²<http://opus.nlpl.eu/EhuHac.php> Azken atzipena: 2020-06-29.

daude, eta ingelesa-euskara parean 585.210. Errepikatuak kenduta 586.839 eta 563.633 dira.

Aldiz, webguneko bertsioak arazo gutxiago ditu. Voltaire-ren *Gutun Filosofikoak* liburua falta da esteka ez dabilelako eta hirukote askok hizkuntzaren bat hutsik dute parekatzeak ez daudelako guztiz zuzen eginda (OPUSeko bertsioan ere liburu hori falta da eta parekatze-arazo berberak daude). Hala ere, hirukoteak jada osatuta daude eta, orokorrean, testuak osorik daude. Gainera, esaldiak jatorrizko ordenan eta liburuka banatuta egoteak datuen analisia errazten du. Hori ikusita, bertsio hau erabiltzea erabaki nuen, datu gehiago lortzeko aukera ematen baitu. Horretarako, *web crawling* deritzon teknikaren bidez jaso ditut datuak: Pythonez idatzitako script bat egin dut, webgunea orriz orri irakurri, itzulpenak jaso eta fitxategietan gordetzen dituen. Emaiza 663.143 lerroko fitxategiak dira, segmentu bat jarrita lerro bakoitzean.

Behin datuak eskuratuta, hiru garbiketa sinple aplikatu dizkiot. Hirukote bat aurrekoaren errepikapena zenean, kendu egin dut: horrelako 13.069 zeuden, gehienak datuak webgunetik jaso ditudan moduagatik sortuak. Badaude beste 6052 hirukote errepikatu (edozein posiziotan), baina horiek ez ditut fase honetan kendu, aurrerago azaltzen dudak konponketa-algoritmoan lagungarriak izan daitezke eta. Ondoren, hiru hizkuntzetan hutsik zeuden hirukoteak kendu ditut: 150 ziren. Horrela, beraz, 649.924 geratzen dira.

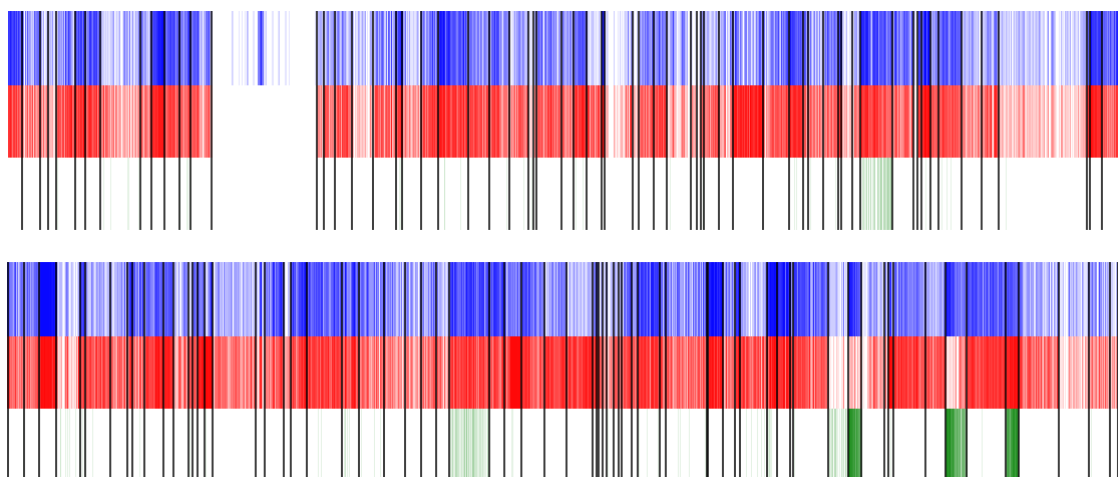
Garbiketarekin bukatzeko, euskal hitz asko azentu arraroekin idatzita zeudela ikus zitekeen. Ez dira akatsak: Joxe Austin Arrietak itzultako bi liburutan, *Denbora galduaren bila* - Swann-enetik eta *Hadrianoren oroitzapenak*, esaldi luze eta konplexuen irakurketa errazteko asmoz, erlatiboazko esaldietako iraganeko aditzek azentu bereizgarria dute³ (*da* + menderagailua → *den*, bereizten dira; baina *zen* + menderagailua → *zen*, berdin geratzen da ohiko idazkeran). Horrela, *zuèn* 456 aldiz agertzen da, *zèn* 341 aldiz, eta abar. Azentu horiek guztiak kentzea erabaki dut, ez baitira egungo euskara idatzian erabiltzen, eta beraz, ez baitut uste itzultzaile automatiko bati ikasgarazteak ezertan laguntzen duenik. Horretarako, *-àn*, *-èn*, *-òn* eta *-ùn* hitz-bukaeretakoa azentuak kendu ditut. Beste aukera bat euskarazko testuetatik azentu guztiak kentzea izan zitekeen, baina hori eginez gero atzeritar izenetako eta frantsesezko esamoldeetako azentuak ere galduko lirateke, besteak beste.

Idazkera ez-estandarrekin jarraituz, euskara batuan idatzita ez dagoen pasarte bat ere topatu dut. Dirudienez, Xabier Olarrak, *Ulises* liburuaren itzultzaileak, historiako hainbat

³Hemen aipatzen da: <https://kritikak.armiarma.eus/?p=3020> Azken atzipena: 2020-06-30.

euskal idazleren estiloa imitatzen du atal batean zehar.⁴ Hala ere, atal motza denez, eta berezitasunak txikiak, zegoen bezala utzi dut.

3.1.2 Parekatze-arazoak



3.1 Irudia: HACeko hutsuneen denbora-lerroa. Beltzez (goitik behera) liburuen mugak, urdinez (goian) gaztelaniazko hutsuneak, gorritz (erdian) ingelesezkoak eta berdez (behean) euskarazkoak. Hutsuneak elkarrengandik hurbil daudenean, kolorea ilunagoa da.

Corpusaren webgunearen arabera, segmentuen banaketa eta parekatzea automatikoki daude eginda; zuzentasun-maila % 91,53koa dela estimatzen dute egileek. Jaso ditudan datuak aztertuta, nahiz eta esaldien gehiengoa zuzen lerrokatuta egon, ikus daiteke erabilitako metodoak, edozein dela ere, arazo batzuk dituela. Lehenago aipatu bezala, hirukote batzuek hizkuntzaren bat hutsik dute: zehazki, 649.924tik 24.913k (% 3,83). Gaztelaniazkoa 15.674ri (% 2,41) falta zaie, ingelesezkoa 22.139ri (% 3,41) eta euskarazkoa 842ri (% 0,001). Horietako bakoitzak, gehienetan, parekatzean arazoren bat egon dela adierazten du.

Hutsuneek corpus osoan zehar duten banaketa ikus daiteke 3.1 irudiko denbora-lerroan: liburuak webgunean dauden ordena berean jarrita, goitik beherako marra beltzak liburuen arteko mugak dira, marra urdinak (goian) gaztelaniaz hutsik dauden hirukoteak, gorriak (erdian) ingelesez hutsik daudenak eta berdeak (behean) euskaraz hutsik daudenak. Marrek gardentasuna dute; horrela, hutsuneak elkarrengandik hurbil daudenean, kolorea

⁴Hemen aipatzen da:

<http://www.igela.eus/language/eu/2015/09/07/ulises-euskaratzearen-odisea/> Azken atzipena: 2020-08-16.

<i>es</i>	<i>en</i>	<i>eu</i>
Los caminantes son poco numerosos, el pueblo está silencioso.	There aren't many people in the streets.	Jende gutxi dabil kalean;
Se oye el ruido de nuestros pasos.	The town is very quiet.	hiria isilik dago.
	Our footsteps echo on the pavement;	Geure pausoen hotsa entzun dezakegu;
Caminamos sin hablar, nuestra madre en medio, entre nosotros dos.	we walk without speaking, Mother in the middle, between the two of us.	hitzik egin gabe goaz, Ama erdian, gu bion artean.

3.1 Taula: Segmentu huts bat duen adibidea. 1. lerroan hasten den arazo bat konpontzeko, 3. lerroan hutsune bat dago. 4.ean, berriro, hiru hizkuntzak zuzen parekatuta daude.

ilunagoa da irudian. Orokorrean, hutsuneak liburu guztietan zehar banatuta daudela ikus daiteke.

Salbuespen nabarmena Biblia da, corpuseko liburu luzeena, 30.499 hirukoterekin: ez du hutsunerik ingelesez eta euskaraz, eta gaztelaniaz 84 bakarrik. Dirudienez, segmentuak Biblia-txatalen arabera daude banatuta eta, horri esker, ez dago parekatze-arazorik. Gaztelaniaz hutsik dauden segmentuetako txatalak falta dira, baina horien aurreko eta hurrengo hirukoteek ez daukate arazorik.

Beste liburuetakoa kasu askotan, aldiz, hutsunea aurretik zetorren arazo bat konpontzeko dago, adibidez, 3.1 taulako kasuan. Bertan lau lerroko pasarte bat dago, segmentu huts batekin. Ikus daitekeenez, gaztelaniazko lehenengo segmentuak ingelesezko eta euskarazko bi hartzen ditu, parekatze-arazo bat sortuz. 3. lerroko hutsunearen eraginez, 4. lerroa berriro zuzen parekatuta dago.

Gainera, arazoak ez daude justu hutsuneen aurretik bakarrik. Liburu batzuetan, batez ere esaldi motz eta sinpleak dituztenetan, noizbehinka arazo gutxi batzuk baino ez daude; baina, beste batzuetan, parekatzeak arazo asko ditu. Itxura txarra zeukaten pasarte batzuk eskuz aztertu ditut, arazoaren tamaina baloratzeko: adibidez, Albert Camus-en *Arrotza* eleberriko 100 hirukoteko zati batean, 54 daude guztiz edo ia guztiz ondo parekatuta, 34k dute testu erabat desberdina gutxienez hizkuntzetako batean, 11k segmentuen zati bat bakarrik dute hiru hizkuntzetan, eta 1ek hutsune bat du. Beste kasu batean, Aleksandr Puxkin-en *Kapitainaren alaba* liburuko 100 hirukotetan, hauek dira emaitzak: 51 daude ondo, 21 gaizki, 23k dute zati bat ondo eta 5ek dute hutsuneren bat.

3.1.3 Parekatzeak hobetzeko metodoa

Aurreko atalean azaldutako arazoak ikusita, HACeko datuen kalitatea hobetzeko moduren bat bilatu dut. Hiru hizkuntzak parekatzeke tresnarik aurkitu ez dudanez, eginda dauden parekatzeak hobetzeko metodo bat garatzen saiatu naiz, corpus osoari aplikatzeko, Bibliari izan ezik (ezin baita hobetu). Hurrengo orrialdeetan, beraz, nire proposamena azaltzen dut, hainbat aldaera posiblerekin.

Ideiaaren oinarria da, testuak ia beti osorik daudenez, zuzen parekatutako hirukoteak sortzeko aukera dagoela hizkuntza bakoitzean beharrezkoak diren segmentuak elkartuz. Horrela, eskuragarri ditugun datuak ahalik eta ondoen aprobetxatuko genituzke. Beraz, ideia izan da segmentuen LASER esaldi-bektoreak lortzea, eta bektore horiek konparatuz parekatze posibleak baloratzea, egokienak aurkitzeko. Datuak garbitzeko beste aukera bat izan zitekeen antzekotasun txikiena duten hirukoteak baztertzea, besterik gabe, baina modu horretan lortutako datu baliagarrien kopurua askoz txikiagoa litzateke.

Azaltzen ditudan esperimentu guztiak *Arrotza* liburuko lehenago aipatutako 100 hirukoteen gainean egin ditut.⁵ Arazo asko dituen pasarte motza denez, emaitzak eskuz aztertzeko aukera ematen du.

Probatutako lehenengo ideiak oso gaizki funtzionatzen du. Lehen lerrotik hasita, hirukote bakoitza sortzeko, planteatzen ditudan aukerak dira hizkuntza bakoitzerako lerro kopuru hauek hartzea: 0-0-1, 0-1-1, 1-1-1, 1-1-2 edo 1-2-2, permutazio posible guztietan. Guztira 13 aukera, beraz. Zero hartzeak karaktere-kate hutsa uztea esan nahi du, eta bi hartzeak uneko lerroa eta hurrengo kateatzea. Zero hartzeko aukera sartu nuen, oso noizbehinka bada ere, hizkuntza batean esaldiren bat falta delako, eta beti hizkuntza batean bakarria hartzen dut beharrezkoak baino elkarketa gehiago ez egiteko (adibidez, ondo parekatutako bi lerro baditugu, bi lerroak elkartuta lortutako 2-2-2 parekatzea ere zuzena litzateke). Aukera bakoitza ebaluatzeko, sortutako kate bakoitzaren LASER bektoreak kalkulatu eta gaztelania-euskara eta ingelese-euskara kosinu-antzekotasunen batura kalkulatu dut. Esaldi bat kate huts batekin konparatzen den kasuei 0,5 balioa eman diet, gehiegi ez penalizatzeke. Behin hirukotea aukeratuta, hizkuntza bakoitzean, hartutakoen hurrengo lerroak aztertzen ditut. Proba exekutatzean, argi geratu zen sistema honek erabaki okerrak hartzen dituela.

Horren arrazoi nagusia da LASER esaldi-bektoreak nahiko fidagarriak direla esaldi bat (A) hartuta bere itzulpena (A') esaldi desberdin batetik (B') bereizteko, baina ez hainbes-

⁵Zehazki, “Ohean jarraitu dut, Mari-ren usaina sumatu dut ...” segmentuan hasi eta “Eskailera beltzak igotzerakoan ...” segmentuan bukatzen da, eleberriaren 1. zatiko 2. eta 3. atalen artean

1. hizkuntza	2. hizkuntza	3. hizkuntza
Uneko posizioa \rightarrow A	A'	A''
B	B'	B''
C	C'	C''

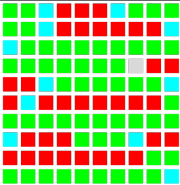

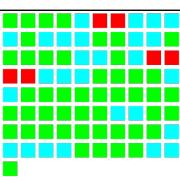



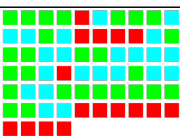
3.2 Taula: Elkarketarik behar ez duen egoera.

1. hizkuntza	2. hizkuntza	3. hizkuntza
Uneko posizioa \rightarrow A	A'	A''
C	B'	B''
D	C'	C''

3.3 Taula: Segmentu bati dagokion testua (B) falta den egoera.

te esaldiaren itzulpenari soberan dagoen zati bat kateatu zaion ala ez bereizteko. Alegia, $A \leftrightarrow A'$ esaldien antzekotasuna eta $A \leftrightarrow A'+B'$ esaldiena kalkulaturik, maiz ematen dio bigarrenari puntuazioa altuxeagoa. Arazo hori saihesteko, pauso bakoitzean esaldiak elkartu ala ez erabakitzeke, honakoa egin dut: 3.2 taulako egoeran, adibidez, A' eta A'' -rekin batera A ala $A+B$ hartu erabakitzeke, $A \leftrightarrow A' \leftrightarrow A''$ eta $A+B \leftrightarrow A' \leftrightarrow A''$ konparatu beharrean, $B \leftrightarrow B' \leftrightarrow B''$ eta $C \leftrightarrow B' \leftrightarrow B''$ konparatu ditut, hau da, aukera bakoitza hartuta geratzen diren **hurrengo esaldiak**. Konparazio hori askoz ere fidagarriagoa da, eta kalkulatu beharreko esaldi-bektoreen kopurua erdia bakarrik da, elkarketan bektorerik ez delako kalkulatu.

Bigarren probarako, beraz, modu horretan aukeratu ditut elkarketak. Gainera, aukera kopurua 7ra murriztu dut, zero hartzeko aukera kenduz, sistema honetan zentzurik ez duelako: 3.3 taulako egoeran, B esaldia falta denez, sistemak $C \leftrightarrow C' \leftrightarrow C''$ hirukotearen antz handia detektatu eta, beraz, A esaldia $A'+B'$ eta $A''+B''$ esaldiekin parekatuko du. Ez da soluzio ideala, baina ez du beste arazorik sortzen eta, gainera, oso gutxitan falta dira esaldiak. Probaren emaitzak 3.4 taulako 2. errenkadan daude. Hemendik aurrerako proba bakoitzean, lortutako parekatzeen zenbakizko emaitzak jarri ditut: lehenago esan bezala, hirukotea “ondo” dagoela diot parekatzea guztiz edo ia guztiz zuzena bada, “gaizki” dagoela hizkuntza batean gutxienez testua erabat desberdina bada, eta “zati bat ondo” dagoela esaldien zati bat behintzat hiru hizkuntzetan agertzen bada. Hala ere, erabakiak hartzeko irizpide nagusia ez dira kopuru horiek izan, baizik eta aukera bakoitzarekin sortzen eta konpontzen diren arazoen azterketa. Izan ere, akats txiki batek hurrengo erabakietan eragin eta zenbakiak asko alda ditzake, hain datu gutxi aztertuta.

-	Jatorrizkoa	Ondo: 54 Gaizki: 34 Zati bat ondo: 11 Hutsak: 1	
(2)	es-eu + en-eu puntuazioa	Ondo: 40 (% 51,3) Gaizki: 13 (% 16,7) Zati bat ondo: 25 (% 32,1) Guztira: 78	
(3)	es-eu + en-eu + es-en punt.	Ondo: 48 (% 59,3) Gaizki: 6 (% 7,4) Zati bat ondo: 27 (% 33,3) Guztira: 81	
(4)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK)	Ondo: 47 (% 60,3) Gaizki: 13 (% 16,7) Zati bat ondo: 18 (% 23,1) Guztira: 78	
(5)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera	Ondo: 33 (% 47,1) Gaizki: 9 (% 12,9) Zati bat ondo: 28 (% 40,0) Guztira: 70	
(6)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera Hasierako 8-8-6 hitzak	Ondo: 38 (% 55,1) Gaizki: 14 (% 20,3) Zati bat ondo: 17 (% 24,6) Guztira: 69	
(7)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera Hasierako 9-9-7 hitzak	Ondo: 26 (% 40,6) Gaizki: 16 (% 25,0) Zati bat ondo: 22 (% 34,4) Guztira: 64	

(8)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera $x = 0,04, y = 0,1$ penaliz.	Ondo: 60 (% 72,3) Gaizki: 6 (% 7,2) Zati bat ondo: 17 (% 20,5) Guztira: 83	
(9)	Aurrekoa bezala baina $y = 0,08$. Emaitzak berdinak dira.		
(10)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera $x = 0,04, y = 0,07$ penaliz.	Ondo: 63 (% 75,9) Gaizki: 2 (% 2,4) Zati bat ondo: 18 (% 21,7) Guztira: 83	
(11)	es-eu + en-eu + es-en punt. Banaketa esaldika (NLTK) 3 elkartzeko aukera $x = 0,04, y = 0,06$ penaliz.	Ondo: 61 (% 74,4) Gaizki: 1 (% 1,2) Zati bat ondo: 20 (% 24,4) Guztira: 82	

3.4 Taula: Hobekuntza-metodoa diseinatzeko probak. Eskuineko irudietan, karratutxo berdeak hirukote zuzenak dira, gorriak okerrak, urdinak zati bat ondo dutenak, eta grisa hutsunea duena.

Hirugarren proban, gaztelania-euskara eta ingelesa-euskara antzekotasunez gain, batuketan **gaztelania-ingelesa ere gehitu** dut. Erredundantea izango zela uste nuen, baina, dirudienez, puntuazio fidagarriagoak sortzen ditu, akats gutxiago eraginez, 3.4 taulako 3. errenkadan ikus daitekeen bezala.

Datuei begiratuta ikusi nuen gauza bat da, kasu batzuetan, lerroen jatorrizko mugek parekatzea zailtzen dutela, arazoak konpontzeko lerro asko elkartzera behartuz. Horregatik, lerroen jatorrizko banaketa erabili beharrean, **esaldiak NLTK⁶ liburutegiarekin banatu** ditut 4. proban. 3.5 taulako bi kasuetan ikus daiteke nola, jatorrizko banaketan ordez NLTK-renak erabilia, parekatze-arazoak konpontzen diren: nahikoa da hizkuntza bakoitzetik esaldiak banaka hartzea. Noski, beste kasu batzuetan ondo eginda zeuden banaketak desegin ditzake. 3.4 taulako 4. errenkadan daude emaitzak.

Bosgarren proban, 2 lerro elkartzeaz gain, **3 elkartzeko aukera** ere onartu dut. Horrela, hirukote bakoitza osatzeko 19 aukera desberdin daude. Izan ere, badaude 3.6 taulakoa bezalako kasu gutxi batzuk, 3 lerro elkartzea behar dutenak. Aldi berean, baina, aukera

⁶<https://www.nltk.org/> Azken atzipena: 2020-08-19

<i>es</i>	<i>en</i>	<i>eu</i>
había poca gente y apurada. • Pasó primero una familia que iba de paseo:	First of all there came a family, going for their Sunday-afternoon walk;	Paseatzen zebiltzan familiak, bi mutiko marinelez jantzita eta neskatala bat oinetako beltzekin. •
dos niños de traje mariner, los pantalones sobre las rodillas, un tanto trabados dentro de las ropas rígidas, y una niña con un gran lazo color de rosa y zapatos de charol. •	two small boys in sailor suits, with short trousers hardly down to their knees, and looking rather uneasy in their Sunday best;	
	then a little girl with a big pink bow and black patent-leather shoes. •	
Uno hasta llegó a gritarme: “¡Les ganamos!” • Dije:	One of them looked up at me and shouted, “We licked them!” • I waved	Haietako batek deiadar egin dit: “xehatu ditugu”. • Eta nik:
“Sí”, sacudiendo la cabeza. •	my hand and called back, “Good work!” • From	“bai”, buruari eraginaz. •
A partir de ese instante los automóviles comenzaron a afluir. •	now on there was a steady stream of private cars. •	Une honetatik aurrera autobusen emana hasi da. •

3.5 Taula: Jatorrizko eta esaldikako banaketen bi adibide. Bakoitzean, taulako lerro horizontalek jatorrizko lerroen banaketa markatzen dute, eta • sinboloek, NLTK-rekin lortzen den esaldikako banaketa.

gehiago emateak erabaki okerrak hartzeko probabilitatea handitzen du kasu gehienetarako. 3.4 taulako 5. errenkadan daude emaitzak.

Aurreko probetan erabilitako metodoak badu arazo bat, akats txiki asko eragiten dituena. Demagun 3.7 taulako egoera dugula. Parekatze zuzena sortzeko, sistemak honakoa egin beharko luke hurrengo bi pausoetan:

1. Puntuazio altuena: $BC \leftrightarrow B' \leftrightarrow B''$. Sortutako hirukotea: $A \leftrightarrow A' \leftrightarrow A''$.
2. Puntuazio altuena: $D \leftrightarrow D' \leftrightarrow D''$. Sortutako hirukotea: $BC \leftrightarrow B'+C' \leftrightarrow B''+C''$.

Kontua da BC-k antzekotasun handia duela bai B' eta bai C'-rekin, bietaz osatuta dagoelako. Beraz, beste hau gertatzea ere nahiko probablea da:

1. Puntuazio altuena: $BC \leftrightarrow C' \leftrightarrow C''$. Sortutako hirukotea: $A \leftrightarrow A'+B' \leftrightarrow A''+B''$.
2. Puntuazio altuena: $D \leftrightarrow D' \leftrightarrow D''$. Sortutako hirukotea: $BC \leftrightarrow C' \leftrightarrow C''$.

<i>es</i>	<i>en</i>	<i>eu</i>
Un poco más tarde pasaron los jóvenes del arrabal, de pelo lustroso y corbata roja, chaqueta muy ajustada, bolsillo bordado y zapatos de punta cuadrada.	Next came a group of young fellows, the local “bloods,” with sleek oiled hair, red ties, coats cut very tight at the waist, braided pockets, and square-toed shoes.	Geroxeago Faubourgko gazteak pasatu dira.
		Ile lakatuak eta gorbata gorriak zituzten.
		Zamarra gerri estuak eta oinetako mutur koadratuak.

3.6 Taula: Euskarazko hiru lerro elkartu behar diren kasua.

<i>1. hizkuntza</i>	<i>2. hizkuntza</i>	<i>3. hizkuntza</i>
Uneko posizioa → A	A'	A''
BC	B'	B''
D	C'	C''
E	D'	D''

3.7 Taula: Bi hizkuntzatan bi lerro elkartu beharreko egoera.

Ez da oso arazo larria, 2. pausoarekin posizio egoki batera itzultzen baita, baina erabat zuzen ez dauden parekatze batzuk sortzen dira. Hori konpontzeko probatutako lehen ideia da **esaldien hasierak bakarrik aztertzea**. Hau da, lerro osoen esaldi-bektoreak lortu eta konparatu beharrean, lerro bakoitzeko hasierako hitzekin bakarrik egitea. Horrela, ondo funtzionatuz gero, BC lerroa aztertzean, B-ri dagokion zatia izango genuke batez ere kontuan. Nahiz eta hizkuntza bakoitzak hitz-ordena desberdina duen, orokorrean, esaldi luzeetan, informazioaren ordena antzekoa da. Proba pare bat egin ditut modu honetan (3.4 taulako 6. eta 7. errenkadak): lehena gaztelaniaz hasierako 8 hitz hartuta, ingelesez beste 8 eta euskaraz 6; bigarrena 9, 9 eta 7 hartuta. Hizkuntzen arteko hitz kopuruen proportzioa aukeratzeko, corpus osoan hizkuntza bakoitzak duen hitz kopuruari begiratu diot: gutxi gorabehera, euskarak halako 1,3 dute beste bi hizkuntzek.

Neurri batean funtzionatzen du, baina bide honetatik ez jarraitzea erabaki genuen. Izan ere, oso zaila da esaldiaren hasiera noraino hartu erabakitzea. Aukeratutako hitz kopuruek itxuraz zentzua zuten aztertutako esaldietarako, baina beharbada ez dira egokiak beste batzuetarako. Liburu bakoitzaren estiloak eragina izan lezake horretan. Kopuru finkoak erabili ordez, beste aukera bat izan daiteke puntuazio-markak kontuan hartzea, baina hori ere zehazteko zaila da.

Planteatutako bigarren ideia da **lerro gehien elkartzen dituzten aukeren puntuazioak**

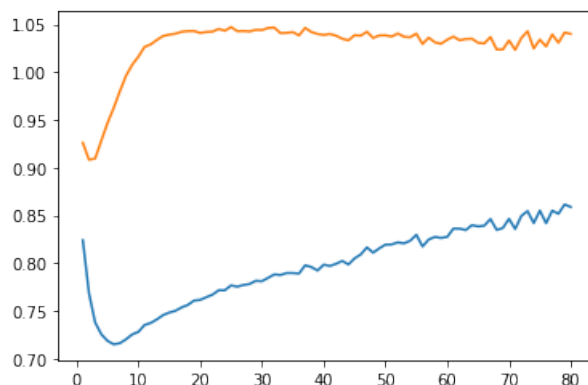
pixka bat penalizatzea. Honek bi helburu betetzen ditu. Alde batetik, lehen aipatutako 3.7 taulakoa bezalako kasuetan, zeinetan bi aukerak puntuazio berdintsuak lortzea espero dugun, erabakia aukera zuzenaren aldekoa izango litzateke, hau da, 1. pausoan 1-1-1 hartzearen aldekoa. Bestalde, 3 lerro elkartzeko aukera onartzeak arriskua duela aipatu dut lehenago, eta honekin murriztu egiten da. Izan ere, aukera horien puntuazioak besteenak baino nabarmen altuagoa izan beharko du, penalizazioaren ondoren ere altuena izateko. Egindako probetan, penalizazio hori aukeraren puntuazioari biderkatzen zaion faktorea da. Faktorea horrela zehazten dut: aukera bakoitzari, hizkuntza batean 2 lerro elkartzeagatik x kentzen diot, eta 3 lerro elkartzeagatik, y . Beraz, honela kalkulatu da aukera bakoitzaren penalizazio-faktorea:

- 1-1-1: 1 (penalizaziorik ez).
- 1-1-2, 1-2-1, 2-1-1: $1 - x$.
- 1-2-2, 2-1-2, 2-2-1: $1 - 2x$.
- 1-1-3, 1-3-1, 3-1-1: $1 - y$.
- 1-3-3, 3-1-3, 3-3-1: $1 - 2y$.
- 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, 3-2-1: $1 - x - y$.

Hasieran $x = 0,02$ eta $y = 0,05$ kentzaileekin probatu nuen, eta eragin txikiegia zutela erabaki nuen. Egin ditudan hurrengo proben emaitzak 3.4 taulako 8., 9., 10., eta 11. errenkadetan daude. Azkenean, $x = 0,04$ eta $y = 0,6$ parametroak aukeratu ditut. Noski, benetan ataza honetarako egokienak diren balioak aurkitzeko, balio gehiagorekin probatu beharko litzateke, eta askoz datu gehiagorekin.

Orain arteko proba guztien irudietan ikusten den moduan, parekatze okerrak bata bestearen atzetik egon ohi dira. Izan ere, lerro bat gaizki sortzen denean, sistema posizio oker batean geratzen da, eta ez du beti izango hurrengo pausoan berreskuratzeko aukera. Zenbat eta luzeagoa izan konpondu beharreko testua, orduan eta arrisku handiagoa egongo da sistema bere onera itzuli ezinik geratzeko, behin eta berriro parekatze okerrak sortzen. Arazoa konpontzeko, jatorrizko hirukote batzuk **kontrol-puntu** moduan jarri ditut, hobekuntza-metodoa hirukote horien artean bakarrik aplikatzeko. Hizkuntza batean hurrengo lerroa kontrol-puntuak den momentuan, beste hizkuntzetan esaldi gehiago geratzen badira, elkartu egiten dira; horrela, hiru hizkuntzetan hurrengo lerroa kontrol-puntu izango da.

Jatorrizko hirukote bat kontrol-puntu izateko, bere hiru bikoteen antzekotasunek atalase bat gainditu behar dute. Antzekotasun hori, orain arte bezala, LASER bektoreen kosinu-



3.2 Irudia: HACeko gaztelania-ingelesa parerako, hitz kopuruaren (bikoteko maximoa) eta batez besteko antzekotasunaren arteko erlazioa. Urdinez kosinu-antzekotasuna eta laranja margin-based score-a.

antzekotasuna da. Atalasea, balio finkoa izan beharrean, segmentuen luzeraren arabera izatea erabaki dut. Izan ere, orokorrean, segmentuen luzera handitu ahala, puntuazioek gora egiten dute, 3.2 irudian ikusten den moduan. Kasu batzuk begiratuta, ez dirudi segmentu luzeak motzak baino hobeto parekatuta daudenik, igoera hori justifikatzeko.

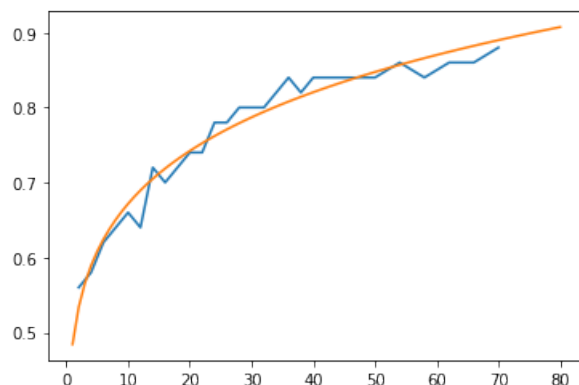
Alternatiba moduan, [Artetxe and Schwenk, 2019] artikuluan proposatzen duten *margin-based scoring*-a ere probatu dut, puntuazio bakoitza kalkulatzeko corpuseko segmentu gehiago kontuan hartzen dituen. LASER liburutegian inplementatuta dator corpus paralelo baterako puntuazio horiek lortzeko aukera. Baina, neurri honekin ere, igoera nabaria ikus daiteke 20 hitzetik beherako bikoteetan, hau da, corpuseko ugarienetan; beraz, honekin ere, atalase finkoak ez dirudi egokia. Kasu konkretu batzuk ikus daitezke 3.8 taulan. Horregatik, eta puntuazio honek kalkulu gehiago behar dituelako, kosinu-antzekotasun arrunta erabili dut.

Aipatutako luzeraren arabera atalasea finkatzeko, hurrengo prozedura erabili dut: hitz kopuru desberdinetarako (2tik 50era binaka, 50etik 70era launaka, beti hirukoteko balio maximoari begiratu), corpuseko hirukoteak antzekotasun txikienaren arabera tartetan banatu ditut ([0,5, 0,52), [0,52, 0,54), [0,54, 0,56), etab.), eta tarte bakoitzerako ausaz aukeratutako 40 hirukote aztertu ditut, gehienez ere guztiz ondo parekatu gabeko 4 hirukote dituen lehen tarte bilatuz. Tarte horretako balio minimoa aukeratu dut uneko luzerarako atalase gisa. Emaitzak 3.3 irudian daude. Ondoren, lortutako balioen forma kontuan izanda, $y = ax^b + c$ kurba bat doitu dut automatikoki balio horietara, luzera guztietarako atalase bezala erabiltzeko. Kalkulu hauetan guztietan, hizkuntzaren bat hutsik duten hirukoteak ez ditut kontuan hartu.

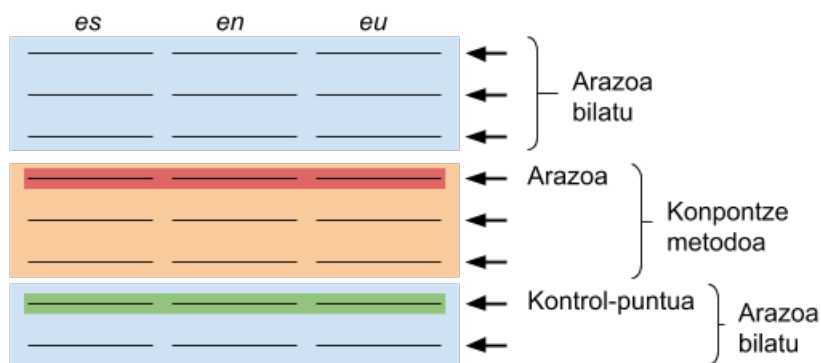
Puntuazioa	Segmentu-bikotea
0,7557	Sale. He goes out.
0,6864	Nada. Deus ez.
0,7610	“Thank you.” -Eskerrik asko.
0,8927	admiten, sí, que éstas han sido producidas por variación, pero se niegan a hacer extensiva la misma opinión a otras formas poco diferentes. Nevertheless, they do not pretend that they can define, or even conjecture, which are the created forms of life, and which are those produced by secondary laws.
0,8784	Acompañado por el viejo Rufo de Éfeso, especialista en tisis, fue personalmente a esperar al puerto de Bayas a mi frágil Elio César. Tibureko klima, Erromakoa baino hobea izanik ere, ez da, alabaina, behar bezain gozoa alborengo gaitzez jota daudenentzat;
0,9129	nay, that it is utterly impossible there should be any such thing, so long as that word is taken to denote an unthinking substratum of qualities or accidents wherein they exist without the mind. Baina agertua denez nolakotasun bat ere, ezta hauetarikorik ere, ezin dela existitu hautematen dituen izpiritu edo adimenduan izan ezik, ateratzen da ez dugula arrazoi bat ere “materiaren” existentzia suposatze-ko.

3.8 Taula: Margin-based score-ekin muturreko kasu batzuk, non gaizki parekatutako bikoteek ondo parekatutakoek baino puntuazio askoz altuagoak dituzten.

Kontrol-puntuak gehituta ere, hobekuntza-metodoa gainontzeko lerro guztietan aplikatzea gehiegizkoa izan liteke, ondo dauden zatiak hondatzeko arriskua baitago. Horregatik, beste irizpide bat erabili dut konponketa non hasi erabakitzeko: jatorrizko lerro bakoitzean, lehenago azaldutako modu berean, elkarketarik behar duen edo ez aztertzen du sistemak. Ez badu behar, alegia, 1-1-1 hartzeko aukera onena bada, ezer aldatu gabe aurrera egiten du. Bestela, arazoren bat dagoela esan nahi du. Orduan, lerro horretatik hasi eta hurrengo kontrol-punturainoko blokean esaldiak NLTK-rekin berriz banatu eta hobekuntza-metodoa aplikatzen du. Kontrol-puntura iristean, hizkuntzaren batean 3 esaldi baino gehiago elkartu behar badira, ziur aski zerbait ondo joan ez den seinale denez, sortutako blokea baztertu eta jatorrizko lerroak uzten ditu. Hizkuntza batean hurrengo lerroa hutsik dagoen momentuan, konponketa derrigorrez hasten du. 3.4 irudian funtzio-namenduaren eskema bat dago.



3.3 Irudia: Hainbat luzeratarako aukeratutako atalaseak (urdinez) eta balioak hurbiltzeko kurba (laranjaz).



3.4 Irudia: Metodoaren funtzionamenduaren eskema, kontrol-puntuak eta arazoen bilaketa gehituta.

3.2 OpenSubtitles

OpenSubtitles corpusa ere OPUS proiektuaren webgunean⁷ eskuragarri dago. Aurrekarie-tan aipatu bezala, gaztelania-ingelesa-euskara segmentu-hirukoteak lortzeko gaztelania-euskara (793.593 bikote, 791 film edo telesail-ataletakoak) eta ingelesa-euskara (805.780, 797koak) bikoteetatik abiatu naiz. Hizkuntza-pare bakoitzak hiru fitxategi ditu: hizkun-tza bakoitzeko testuarekin fitxategi bana, eta hirugarren bat identifikadoreekin. Azken horretan ikus daiteke lerro bakoitza zein film edo ataletakoa den (IMDb identifikado-rea), azpтитuluak hartu diren fitxategiaren izena, eta lerro bakoitza jatorrizko fitxategiko zenbatgarren azpтитuluari dagokion (identifikazio-zenbaki bat edo gehiago izan daitezke,

⁷<http://opus.nlpl.eu/OpenSubtitles-v2018.php> Azken atzipena: 2020-07-10.

zenbait lerro jatorrizko azpititulu batzuk elkartuz sortuta baitago). Informazio hori erabili dut bi pareetako bikoteak parekatu eta hirukoteak sortzeko.

Hasteko, bi hizkuntza-pareek dituzten film edo atal komunak 770 dira. Gainontzekoetatik, noski, ezin da hirukoterik lortu. Ondoren, film edo atal bakoitzean, bi pareetako euskarazko bertsioa berdina izatea komeni da, parekatzea errazteko, baina hau ez da beti horrela. Izan ere, OpenSubtitles-en, film edo atal baten azpitituluen bertsio bat baino gehiago egon daiteke hizkuntza berean (bat bestearen zuzenketa delako, bakoitza eskualde bateko bikoizketari dagokiolako...). Corputa sortzean, hizkuntza-pare, film/atal eta hizkuntza bakoitzeko azpitituluen bertsio bat aukeratu zuten, eta desberdinak izan daitezke hizkuntza-pare desberdinen artean. Gaztelania-euskara eta ingelesa-euskara pareen kasuan, nahiz eta 22 film edo atalen kasuan fitxategien izenak desberdinak izan, ia beti bi fitxategietako azpitituluak berdinak direla dirudi, bi kasutan izan ezik: *Theeb* (2014) eta *The Lobster* (2015) filmen euskarazko itzulpenak nahiko desberdinak dira bi hizkuntza-pareetan, nahiz eta, itxuraz, bi bertsioak guztiz zuzenak izan. Horrelako kasuetan ere hirukoteak sortzeko metodo bat garatu liteke, aurreko atalean erabilitako esaldi-bektoreen laguntzaz, adibidez. Hala ere, kasu honetan, datuen oso zati txikia zenez, ez dut bide horretatik jarraitu.

Kontuan izateko hurrengo gauza da, film edo atal bererako, pareetako batean bestean ez dauden esaldi batzuk egon daitezkeela. Hori gertatzeko arrazoietakoa bat da, adibidez, film batean ingelesez idatzitako hitz batzuk pantailan agertzen baldin badira, testu hori ez dela ingelesezko azpitituluetan agertuko, baina bai agian euskarazkoetan eta gaztelaniazkoetan. Ondorioz, gaztelania-euskara bikotean agertuko litzateke, baina ez ingelesa-euskara bikotean. Beste esaldi batzuen kasuan, ziur aski, corputa sortzean denbora-markak ez zetozen bat behar bezain ondo, eta horregatik baztertuko zituzten.

Beraz, hirukoteak sortzeko, euskarazko azpitituluen identifikadoreak erabili ditut, bi pareetan agertzen diren esaldiak aurkitzeko. Kasu batzuetan, lerro batzuk elkartu beharra dago, bi pareetako euskarazko testuek guztiz bat egiteko. Adibide bat dago 3.9 taulan.

es-eu Testua	IDak	en-eu Testua	IDak
Kaixo! Arratsalde on!	1, 2	Kaixo!	1
Zer moduz atzo?	3	Arratsalde on! Zer moduz atzo?	2, 3

3.9 Taula: Elkartu beharreko bi lerro.

Adibideko kasuan, lerroak elkartu beharra dago bi aldeetan zehazki euskarazko testu bera lortzeko, [1, 2, 3] identifikadoreak izango dituenak. Zehazki, film edo atal bakoitzean bat

egiten duten esaldi guztiak lortzeko, azpiko sasikodean agertzen den algoritmoa jarraitu dut. Demagun A eta B hizkuntza-pare bakoitzeko lerroen identifikadoreak dituzten listak direla (aurreko adibidean, $A = [[1, 2], [3]]$ eta $B = [[1], [2, 3]]$). Hasi baino lehen, lista bakoitza lerro bakoitzeko lehen identifikadorearen arabera ordenatu dut. Ia denak lehendik ordenatuta zeuden, baina bazeuden salbuespen gutxi batzuk.

```

1:  $uneko_A, uneko_B = \text{lehena}(A), \text{lehena}(B)$ 
2: while not bukatu_da( $A$ ) or bukatu_da( $B$ ) do
3:   if  $uneko_A == uneko_B$  then
4:     if testuak_berdinak( $uneko_A, uneko_B$ ) then
5:       gorde_esaldiak( $uneko_A, uneko_B$ )
6:        $uneko_A, uneko_B = \text{hurrengo}(A), \text{hurrengo}(B)$ 
7:     else if  $uneko_A[0] < uneko_B[0]$  then
8:        $uneko_A = \text{hurrengo}(A)$ 
9:     else if  $uneko_B[0] < uneko_A[0]$  then
10:       $uneko_B = \text{hurrengo}(B)$ 
11:    else if  $uneko_A \subset uneko_B$  then
12:       $uneko_A += \text{hurrengo}(A)$ 
13:    else if  $uneko_B \subset uneko_A$  then
14:       $uneko_B += \text{hurrengo}(B)$ 
15:    else
16:      if gehitutako azkena Atik then
17:         $uneko_A = \text{aurrekoa}(A)$ 
18:         $uneko_B = \text{hurrengo}(B)$ 
19:      else
20:         $uneko_B = \text{aurrekoa}(B)$ 
21:         $uneko_A = \text{hurrengo}(A)$ 

```

Behin zenbakiak bat egiten dutela, lortutako euskarazko testuak ere berdinak direla egiaztatu beharra dago (sasikodeko 4. lerroa). Alde batetik, badaude lau film zeinetan, nahiz eta bi pareetako euskarazko itzulpenak berdinak izan, esaldi bakoitzaren zenbakiak ez datozen bat askotan bi fitxategien artean eta, beraz, zenbakien arabera parekatze gutxi egin daitezkeen. Berez, antzeko algoritmo bat erabili ahalko litzateke testuaren arabera parekatzeak lortzeko, baina nahiko kasu gutxi direnez, ez dut egin. Film horietaz gain, lehenago aipatu bezala, badaude beste bi euskarazko bi itzulpen desberdinekin. Kasu horietarako, begiratu nuen ea esaldi baliokideek zenbaki berberak dauzkaten, testua berdina izateko baldintza kenduta parekatzeak lortu ahal izateko, baina ez da hala. Beraz, film

horietatik hirukote gutxi batzuk baino ezin dira lortu, euskarazko esaldi baten bi bertsioak eta bi zenbakiak, kasualitatez bada ere, bat datozenean. Gainontzeko film edo ataletan, oso gutxitan esaldi batzuk baztertu beharra dago, testuek zehazki kointziditzen ez dutelako. Guztira, 732.144 hirukote lortzen dira.

Behin hirukoteak lortuta, datuak gainetik aztertu ditut, arazoak dituen film edo atalik dagoen ikusteko. 7 aurkitu ditut, hirukoteen gehiengoan hizkuntza bat besteekin gaizki parekatuta dutenak. Beraz, horiek kentzea erabaki dut. Gainontzekoetan, itxuraz, ez dago arazo handirik; corpuseko bikoteak sortzeko denbora-marken arabera metodoa nahiko fidagarria dela dirudi. Garbiketa hori eginda, 726.471 hirukote geratzen dira. Errepikatuak kenduta, 679.341 dira.

Azkenik, aipatzekoa da azpтитuluen idazketan etenpuntuak erabiltzeko ohitura dagoela esaldi batek hurrengo azpтитuluan jarraitzen duela markatzeko. Horrelako kasu batzuk ikus daitezke lortutako datuetan. Adibidez, hurrengo hirukotean euskaraz jarritako etenpuntuak soberan daudela dirudi:

Sí, no creas que no lo noté cuando nos conocimos.

Yeah, don't think I didn't notice that little item the first time we met.

Bai, ez pentsatu... ez nuela antzeman zera txiki hori elkar ezagutu genuenean.

Entrenamendurako datuetan horrelako asko egongo balira, beharbada, ikasitako ereduak beharrezkoak ez diren etenpuntuak idatziko lituzkete. Hala ere, datuetatik soberan daudenak bakarrik kentzeko modu errazik aurkitu ez dugunez, zeuden bezala utzi ditut guztiak. Egindako probetan, ereduak sortzen dituzten testuei begiratuta, ez dirudi arazo hori agertu denik.

3.3 Tokenizazioa

Aurrekarietan azaldu bezala, tokenizazioaren lehen pausoa hitzak eta gainontzeko karaktereak banatzea da. Horretarako, OpenNMT-ko tokenizatzaila ([pyonmttok](https://github.com/OpenNMT/Tokenizer)⁸ libururegia) erabili dut. Entrenamendurako datuak sortzeko, parametro hauek aukeratu ditut:

- `mode="conservative"`: OpenNMT tokenizazio estandarra, hitzen arabera.
- `joiner_annotate=True`: tokenizazioa desegin ahal izateko, karaktere berezi bat

⁸<https://github.com/OpenNMT/Tokenizer> Azken atzipena: 2020-08-19

jartzen du sortu den zuriunearen ondoan, garrantzi txikieneko karaktereari itsatsita (mother's eyes are red. → mother •'• s eyes are red •.).

- `case_markup=True`: testua minuskulaz jarri eta token bereziak gehitzen ditu, hurrengo hitza maiuskulaz hasten dela adierazteko (`<mrk_case_modifier_C>`) eta osorik maiuskulaz idatzita dagoen tarte bat markatzeko (`<mrk_begin_case_region_U>` eta `<mrk_end_case_region_U>`).
- `soft_case_regions=True`: osorik maiuskulaz dagoen tarte bat, hainbat hitzekoa denean ere, hasieran eta bukaeran bakarrik markatzen du.

Ondoren, BPE hiztegia ikasi eta testua tokenizatzeko YouTokenToMe⁹ liburutegia erabili dut. Eredua ikasteko proba bakoitzeko entrenamendu-multzoa erabili dut, parametro hauekin:

- `vocab_size=20_000`: hiztegiaren tamainak hitzen zatikatzeko-maila baldintzatzen du. Azterketa sakonagoa egin beharko litzateke kopuru egokiena zein den jakiteko.
- `coverage=0.9999`: $[0, 1]$ tartean, karaktere guztietatik, ereduak jasoko duen zatia. Adibidez, HAC corpusarekin bakarrik ikasitako BPE ereduak, *ü* karakterea hiztegitik kanpo geratzen da, nahiz eta hitz batzuetan agertu, nahikoa maiztasun ez duelako.

Gainera, BPE aplikatu baino lehen, OpenNMT-ko tokenizatzaileak sortutako token bereziak nahiko luzeak zirenez, laburtu egin ditut:

- `<mrk_case_modifier_C>` → `<C>`
- `<mrk_begin_case_region_U>` → ``
- `<mrk_end_case_region_U>` → `<E>`

Horrela, gordetako fitxategien tamaina nabarmen txikiagotzen da, eta BPE hiztegian token horietarako bakarrik sortutako arau-multzo bat egotea saihesten da.

Adibide moduan, hona esaldi bat tokenizazioaren aurretik eta ondoren:

Esperamos un poco y después entramos en el jardín, rodeamos la casa,
nos agachamos

⁹<https://github.com/VKCOM/YouTokenToMe> Azken atzipena: 2020-08-19

_⟨C _esper amos _un _poco _y _después _entra mos _en _el _jardín _●,
_rode amos _la _casa _●, _nos _ag ach amos

Token bakoitza zenbaki bihurtuta, honela geratzen da:

[72, 1341, 601, 159, 1417, 112, 1284, 7017, 529, 136, 153, 8679, 78, 5378, 601, 132,
1161, 78, 778, 338, 969, 601]

4. KAPITULUA

Ikasketa-prozesua

4.1 Entrenamendua

Entrenatutako sareen tamaina definitzeko, transformer-a aurkezten duten [Vaswani et al., 2017] artikuluko *base* ereduko hiperparametroak erabili ditut. Guztira, iturri bakarreko ereduak 64.640.544 parametro entrenagarri dituzte, eta bi iturrikoek 70.950.432, deskodetzaileko geruzetako atentzio-geruza gehigarrien ondorioz handitzen baita kopurua.

Posiziodun bektoreen kalkulua inplementatzeko moduaren ondorioz, sareak hartuko dituen sekuentzien luzera maximoa aurretik zehaztu behar da. Muga hori 120 tokenetan ezarri dut, nahikoa baita oso esaldi luzeak ere onartzeko. Hizkuntzetako batean luzera-muga gainditzen duten instantziak baztertu egin ditut entrenamendurako. Modu berean, hizkuntzaren bat hutsik zeukatenak ere kendu ditut.

Sareak entrenatzeko Adam optimizazio-algoritmoa erabili dut. PyTorch-eko parametro lehenetsiak erabili ditut, ikasketa-tasaren kasuan izan ezik, 0,001en ordez 0,0001 erabili behar izan baitut. Galera-funtzioa entropia gurutzatua da, eta *batch* tamaina 90ekoa. Eredutako bakoitza 20 epokaz entrenatu dut; epoka bakoitzaren ondoren kontrol-puntu modura ikasitako parametroak fitxategi batean gorde ditut eta ereduaren beste datu batzuekin ebaluatu ditut.

Entrenamendua Google Colab zerbitzuan egin dut, NVIDIAren Tesla P100 edo Tesla T4 GPUekin, une bakoitzean erabilgarri zegoenaren arabera. apex.amp¹ liburutegiari esker,

¹<https://nvidia.github.io/apex/amp.html> Azken atzipena: 2020-08-19

16 biteko doitasuna erabili dut 32koaren ordeztu, exekuzioak azkartuz eta memoriaren eta biltegiatzearen beharrak murriztuz. Zehazki, O2 optimizazio-maila erabili dut. Egindako esperimendu guztietatik, gehien iraun duen entrenamendua 29 ordu ingurukoa izan da.

4.2 Ebaluazioa

Ikasitako ereduak ebaluatzeko, bi datu-multzo hartu ditut, corpus bakoitzetik bat. Behar bezala ebaluatu ahal izateko, ebaluazio-multzoetako instantziek zuzen lerrokatutako hirukoteak izan behar dute. Hori bermatzeko, honela osatu ditut multzoak: corpuseko hirukote guztien artetik ausaz 2000 hautagai hartu, LASER esaldi-bektoreen arabera puntuatu eta 1000 onenak balidazio-multzorako hartu ditut. Puntuatzeko, hirukoteko bektoreen kosinuantzekotasuna bikoteka kalkulatu eta batu egin dut, parekatzeak hobetzeko metodoan bezalaxe. Hizkuntzaren bat hutsik duten hirukoteak zuzenean baztertu ditut.

Corpus bakoitzean, aipatutako 2000 hautagai horiek ez beste guztiak utzi ditut entrenamendurako. Ebaluazio-multzoan sartu ez diren 1000 hautagaiak, puntuazio txarrenak dituztenak, ez ditut entrenamendu-multzoan sartu. Izan ere, hautagai horien artean gaizki parekatutako hirukoteen proportzioa handia izatea espero da, batez ere HAC corpusaren kasuan.

Aurretik azaldu bezala, ereduak irakur ditzaketen segmentuek luzera-muga dute. Ebaluaziorako datuen kasuan, iturrietako batek muga gainditzen duenean, segmentuaren hasierako tokenak eman dizkiot ereduari, kopuru maximoa bete arte. Muga gainditzen dutenen kopurua esperimendu bakoitzean erabilitako BPE ereduaren arabera da, baina, kasurik txarrenean, gaztelaniazko 14 segmentu eta ingelesezko 12 moztu behar izan dira, denak HAC corpusekoak.

Ereduek sortutako testuen kalitatea neurtzeko BLEU puntuazioa [Papineni et al., 2002] erabili dut, NLTK liburutegiaren bidez. Metrika horrekin, sortutako segmentu baten eta erreferentziako itzulpen errealean arteko antzekotasuna kalkulatu da. Erreferentziako itzulpen bat baino gehiago erabiltzeko aukera dago, baina kasu honetan bakarra erabili ahal izan dut, datuetan jasotako euskarazko itzulpena. Puntuazioa corpus bakoitzeko ebaluazio-multzorako kalkulatu dut, eta baita bi multzoak elkartuta ere. Azken hori erabili dut ereduak konparatzeko irizpide modura.

BLEU kalkulatu baino lehen, erreferentziako itzulpenak OpenNMT-rekin tokenizatu ditut, baina hitzak banatzeko pausoa bakarrik eginda, alegia, tokenizazioa desgiteko karaktererik gehitu gabe (`joiner_annotate=False`) eta maiuskulak bere horretan utzita

(`case_markup=False`). Izan ere, azken urteetako lanetan ebaluaziorako erabiltzen diren tresna estandarrek honen antzeko tokenizazioa egiten dute. Ereduak sortutako testuaren kasuan, BPE kodeketa eta jatorrizko tokenizazioa desegin eta beste parametro hauekin berriro tokenizatu beharra dago, konparazioa egin ahal izateko.

Entrenamenduan zehar, epoka bakoitzaren ondoren egindako ebaluazioan, greedy deskodeketa erabili dut, azkar exekutatzen delako. Ebaluatutako kontrol-puntuetatik, puntuazio onena lortzen duena gordetzen dut. Horrez gain, beste eredu bat ere gordetzen dut, parametroen batezbestekoa kalkulatz kontrol-puntu onenaren, aurreko epokakoaren eta hurrengo epokakoaren artean (*checkpoint averaging* edo *checkpoint smoothing* esaten zaiona). Gordetako bi eredu horiekin, greedy deskodeketaz gain beam search ere erabili dut, $k = 4$ zabalerarekin. Exekuzioa luzeagoa da, baina emaitza hobeak ematen ditu.

5. KAPITULUA

Esperimentuak

Kapitulu honetan hainbat esperimentu aurkezten dira, parekatzeak hobetzeko metodoa lagungarria den probatzeko, bi iturriko eredu iturri bateko ereduarekin konparatzeko, bi iturriak konbinatzeko gai dela egiaztatzeko eta Batua.eus itzultzailearekin konparatzeko.

5.1 Parekatzeak hobetzeko metodoa

Lehen esperimentuan egiaztatu dut HACeko datuak nire metodoarekin prozesatzeko itzultzailearen ikasketan laguntzen duen ala ez. Horretarako, bi iturriko eredu bat entrenatu dut HACeko entrenamendu-multzoko jatorrizko datuekin, eta beste bat nire metodotik pasatutako datuekin. Jatorrizko datuekin emaitza hobeak lortzen zirela ikusita, 3. proba bat egin dut, jatorrizko datuak eta prozesatuak elkartuta. Izan ere, bi datu-multzoak konparatuta, ikus zitekeen bazeudela multzo batean ondo eta bestean gaizki parekatutako hirukoteak, bai multzo baten eta bai bestearen alde. Beraz, biak elkartuta, zuzen parekatutako informazio kopurua handiagoa da. Hiru probetan HACeko entrenamendu-multzotik ikasitako BPE eredu erabili dut tokenizaziorako.

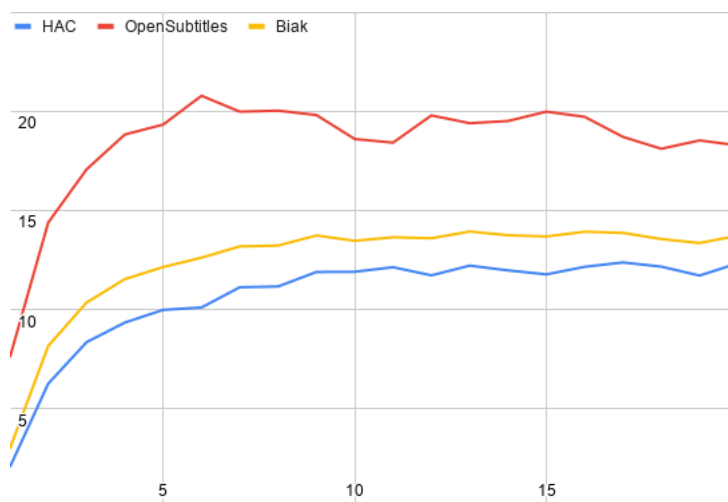
Proba hauetako datu kopuruari buruzko informazioa [5.1](#) taulako lehen zutabeetan dago, eta ebaluazioaren emaitzak [5.2](#) taulan. Emaitzei begiratuta, argi dago nire metodoarekin datuak prozesatzea ez dela lagungarria izan ereduaren entrenatzeko. Jatorrizko datuak eta prozesatuak elkartuta ere, jatorrizkoekin bakarrik lortzen diren antzeko emaitzak lortzen dira. Hori ikusita, hurrengo esperimentuetan HACeko jatorrizko datuak bakarrik erabili ditut.

	HAC			HAC+OpenSubtitles
	Jatorrizkoak	Nire metodoa	Biak elkartuta	
Hasieran	647.924	582.284	1.230.208	1.372.210
Errepikatuak	−6052	−5529	−470.057	−53.496
Hutsak	−24.185	−6853	−24.185	−24.185
Luzeegiak	−3992	−7542	−8790	−4089
Entrenatzeko	641.872	562.360	727.176	1.290.440

5.1 Taula: Proba bakoitzean entrenatzeko erabilitako instantzia kopurua.

		Jatorrizkoak	Nire metodoa	Biak elkartuta
Kontrol-puntu bakarra	Greedy	13,94	13,73 (−0,21)	14,07 (+0,13)
	Beam search	14,58	13,96 (−0,62)	14,34 (−0,24)
Checkpoint averaging	Greedy	14,43	14,04 (−0,39)	14,52 (+0,09)
	Beam search	14,91	14,35 (−0,56)	14,86 (−0,05)

5.2 Taula: HAC corpusarekin bakarrik entrenatuta lortutako emaitzak. Egindako hiru entrenamenduetarako, modu desberdinetan ebaluatuz lortutako BLEU puntuazioak agertzen dira, bi corpusetako ebaluazio-multzo elkarturako. Jatorrizko datuekin lortutako emaitzekiko diferentzia erakusten da beste bi zutabeetan.



5.1 Irudia: Epoka bakoitzaren ondorengo ebaluazioaren BLEU puntuazioak, HACeko jatorrizko datuekin egindako entrenamenduan, corpus bakoitzeko ebaluazio-multzorako eta multzo elkarturako.

Horrez gain, 5.1 irudian ikus daiteke, jatorrizko datuekin egindako probarako, nolako eboluzioa duen datu-multzo bakoitzarekin egindako ebaluazioak entrenamenduan zehar. Argi ikusten da OpenSubtitles corpora itzultzeko errazagoa dela, puntuazio altuagoa lor-

		es+en→eu	es→eu	en→eu
Kontrol-puntu bakarra	Greedy	15,88	14,48	12,95
	Beam search	16,42	15,16	12,96
Checkpoint averaging	Greedy	16,32	15,01	13,25
	Beam search	17,08	15,35	13,46

5.3 Taula: Iturri bateko eta bi iturriko ereduen emaitzak, bi corpusekin entrenatuta. Egindako hiru entrenamenduetarako, modu desberdinetan ebaluatuz lortutako BLEU puntuazioak agertzen dira, bi corpusetako ebaluazio-multzo elkarturako.

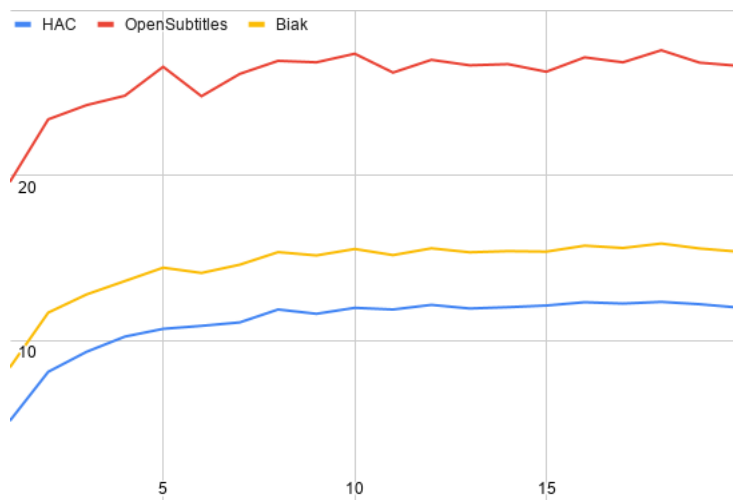
tzen baita corpus horretako ebaluazio-multzoan HACekoan baino, nahiz eta, eredia entrenatzeko, ez den OpenSubtitles-eko daturik erabili. Gainera, lehenago lortzen dira emaitza onenak. Aldiz, HACeko ebaluazio-multzoan, entrenamenduak jarraitu ahala, emaitzek pixkanaka hobetzen jarraitzen dute. OpenSubtitles-eko datuek 6. epokatik aurrera duten jaitsierak adierazi lezake ereduak momentu horretan orokortzeko gaitasun handiagoa dela, eta geroago, HACeko estilora gaindoitzen dela, nolabait.

5.2 Iturri bateko eta bi iturriko ereduen konparazioa

Bigarren esperimentuan erakutsi dut bi iturriko itzultzailea ikasita emaitza hobeak lortzen direla iturri bakarrekoarekin baino. Horretarako, alde batetik, bi iturriko eredu bat entrenatu dut, HACeko eta OpenSubtitles-eko entrenamendu-multzoak elkartuta. Horrez gain, gaztelania-euskara eta ingelesa-euskara iturri bakarreko ereduak ere entrenatu ditut. Kasu guztietan, hiru hizkuntzetako entrenamendu-multzo osotik ikasitako BPE eredia erabili dut, eta bi iturriko eredurako baztertutako instantzia guztiak iturri bakarrekoak entrenatzeko ere baztertu ditut, hau da, hiru hizkuntzetako batean segmentu hutsa edo luzeegia dutenak. Horrela, hiru ereduak baldintza berdinetan entrenatu ditut.

Proba hauetako datu kopuruari buruzko informazioa 5.1 taulako azken zutabea dago, eta ebaluazioaren emaitzak 5.3 taulan. Bertan, garbi ikusten da bi iturrikoak emaitza hobeak lortzen dituela. Horrez gain, iturri bakarrekoen artean, gaztelaniazkoak emaitza hobeak ditu ingelesezkoak baino. Desberdintasun horren arrazoa hizkuntza-pare bakoitzean itzulpenak duen berezko zailtasuna izan daiteke, baina baita erabilitako datuen estiloa ere.

Aurreko esperimentuan bezala, 5.2 irudian ikus daiteke, bi iturriekin egindako probarako, nolako eboluzioa duen datu-multzo bakoitzarekin egindako ebaluazioak entrenamenduan zehar. HACeko datuekin bakarrik entrenatutako ereduak ez bezala, entrenamenduak au-



5.2 Irudia: Epoka bakoitzaren ondorengo ebaluazioaren BLEU puntuazioak, bi corpusetako datuekin eta bi iturriekin egindako entrenamenduan, corpus bakoitzeko ebaluazio-multzorako eta multzo elkaturako.

rrera egin ahala, bi corpusetako ebaluazioak hoberanzko joera du. Gainera, OpenSubtitles-en gaineko emaitzak aurreko esperimentuan baino nabarmen hobeak dira.

Orain arteko analisi kuantitatiboaz gain, kasu konkretuekin erakutsi nahi izan dut nire itzultzailea bi iturrietako informazioa konbinatzeko gai dela. Horretarako, gaztelaniaz zein ingelesez anbiguoak diren esaldi batzuk idatzi ditut, bigarren iturriarekin desanbiguatzeko direnak, bi modu posibletan. Esaldi-bikote horiek ikasitako eredu onenarekin itzuli ditut (bi corpusetan entrenatua, kontrol-puntu batezbestekoa eginda, beam search bidez deskodetua). Emaitzak 5.4 taulan daude. Ikusten denez, espero bezala, itzultzailea gai da bi iturrietako informazioa erabili eta esaldiari esanahi zuzena emateko.

5.3 Batua.eus itzultzailearekin konparazioa

Bukatzeko, nire itzultzailea gaur egungo puntako gaztelania-euskara itzultzaile neuronal batekin konparatu dut, zenbateraino hurbiltzen den ikusteko. Horretarako, Vicomtech-en Batua.eus erabili dut, API bat eskuragarri baitauka, ebaluazio-multzo osoak itzultzeko erabili dudana. Emaitzak 5.5 taulan daude. Ikusten denez, nire itzultzailearen puntuazioak, txikixeagoak izan arren, nahiko hurbil geratzen dira, ebaluaziorako datu hauekin behintzat.

Itzultzaile bakoitzak ebaluazioan sortzen dituen testuei begiratuta, orokorrean, Batua.eus-

enek kalitate handiagoa dute, batez ere esaldi konplexuenetan. Hala ere, noizbehinka, bi iturri erabili izanak lagundu duen kasu batzuk daude. Bereziki, maiz samar agertzen da gaztelaniazko *usted* tratamenduak 3. pertsonako adizki eta posesibo berberak erabiltzeak sortzen duen arazoa, 5.6 taulan jasotako kasuetan ikusten den moduan. Gainera, jatorrizko testua ingelesez egongo balitz ere, badago antzeko arazo bat: *you* singularra eta plurala ez bereiztea. Hori erakusteko, Google Translate-k ingelesetik sortutako itzulpenak ere gehitu ditut taulan. Ondorioz, 2. pertsona plurala behar den kasu batzuetan, bi iturriak konbinatu behar dira itzulpen zuzena ziur lortzeko.

Azkenik, ikasitako itzultzailearen mugak erakusteko, entrenamenduan erabilitako testu motatik urruti dauden esaldi batzuk itzultzen probatu dut. Horretarako, gaztelaniazko eta ingelesezko bertsioa duten hiru prentsa-ohar hartu ditut: bat Apple-rena, beste bat Microsoft-ena eta azkena Espainiako Gobernuarena. Bertatik, esaldi zail bana aukeratu eta bi iturriko itzultzailearekin eta Batua.eus-ekin itzuli ditut, 5.7 taulan agertzen den moduan.

Jatorrizkoa (es)	Durante toda la semana, los desarrolladores de Apple han interactuado con más de 1.000 ingenieros de Apple a través de los nuevos foros para desarrolladores y las sesiones individuales, en las que han profundizado en las funciones más recientes que llegan con macOS Big Sur, iOS 14, iPadOS 14, watchOS 7 y tvOS 14.
Jatorrizkoa (en)	All week Apple developers have been engaging with more than 1,000 Apple engineers via the all-new Developer Forums and one-on-one Developer Labs, diving deep into the newest capabilities coming to macOS Big Sur, iOS 14, iPadOS 14, watchOS 7, and tvOS 14.
Nirea (es+en→eu)	<i>Astero Appleko garatzaileak 1.000 ingeniari baino gehiagorekin sartu dira Appleteko foro berrietan eta saio indibidualetan zehar, zeinetan sakondu baitira MACHego Big Hego, 1000 Pad14, PADk, PADk, 7 eta 14 telebista.</i>
Batua.eus (es→eu)	<i>Aste osoan zehar, Appleko garatzaileek Appleko 1.000 ingeniari baino gehiagorekin hitz egin dute garatzaileentzako foro berrien eta banakako saioen bidez. Horietan, Big Sur makosarekin, iOS 14rekin, iPadOS 14rekin, watchOS 7rekin eta tvOS 14rekin iristen diren funtzio berrietan sakondu dute.</i>

Jatorrizkoa (es)	Los juegos con la insignia Optimizado para Xbox Series X pueden contar con todo tipo de mejoras, desde prácticamente eliminar los tiempos de carga a través de la Arquitectura Xbox Velocity, mejorar los gráficos o trazado de rayos DirectX acelerado por hardware impulsado por nuestra GPU personalizada de próxima generación, hasta velocidades de cuadro más estables y a menudo, más altas, de hasta 120 fps.
Jatorrizkoa (en)	Games featuring the Optimized for Xbox Series X badge can showcase anything from virtually eliminating load times via the Xbox Velocity Architecture, heightened visuals and hardware-accelerated DirectX ray-tracing powered by our custom, next generation GPU, to steadier and often higher framerates up to 120fps.
Nirea (es+en→eu)	<i>Xbox Telesairako baikor Optimizatutako Jokoek, Xbox Dougeten arkitektura Xbox Velocity arkitektuaren garaia kenduta, XAzelerare-ko grabatuak eta X transputamenduak, gure hurrengo belaunaldiek eraginda, hurrengo belaunaldi garaia eta 120 f0era arte.</i>
Batua.eus (es→eu)	<i>Xbox Series X-erako intsignia Optimizatuko jokoek mota guztietako hobekuntzak izan ditzakete, hala nola karga-denborak ia desagerraraztea Xbox Velocity arkitekturaren bidez, grafikoak edo Zuzeneko x izpien trazadura hobetzea hardware bidez, gure hurrengo belaunaldiko GPU pertsonalizatuak bultzatuta, baita koadro-abiadura egonkorragoak eta sarritan altuagoak ere, 120 fps-raino.</i>
Jatorrizkoa (es)	El Consejo de Ministros ha aprobado un Real Decreto-ley que recoge medidas para reactivar la economía en los ámbitos de los transportes y de la vivienda y afrontar el impacto del coronavirus.
Jatorrizkoa (en)	The Council of Ministers approved a Royal Decree-Law that contains measures to reactivate the economy in the fields of transport and housing and to address the impact of the coronavirus.
Nirea (es+en→eu)	<i>Ministroen Kontseiluak Errege Dekretu bat onartu du (neurriak hartzen ditu garraioetako eta etxebizitzaren eremuetan ekonomia berreraikitze eta koronavirusaren inpaktuari aurre egiteko.</i>
Batua.eus (es→eu)	<i>Garraioen eta etxebizitzaren alorretan ekonomia suspertze eta koronavirusaren eraginari aurre egiteko neurriak jasotzen dituen Errege Lege Dekretua onartu du Ministroen Kontseiluak.</i>

5.7 Taula: Prentsa-oharretatik hartutako esaldiak.

Ikusten denez, HAC eta OpenSubtitles corpusekin entrenatutako itzultzaile honek ez du edozein testurekin ondo funtzionatzen. Zati batzuk nahiko ondo itzulita daude, baina beste batzuekin arazoak izan ditu, batez ere 2. esaldian. Aipatzekoa da, BPE tokenizazioari esker, entrenamenduan ikusi gabeko termino batzuk itzultzeko gai izan dela, adibidez, *del coronavirus + of the coronavirus* → *koronavirusaren*, ia zuzen itzuli eta behar bezala deklinatu duena.

Ya lo había dicho. She had already said it. <i>Esan zuen lehen ere.</i>	Ya lo había dicho. I had already said it. <i>Esan dut lehen ere.</i>
Esta alianza costó más de lo que esperaba . This ring cost more than I expected . <i>Eraztun hau espero nuena baino garesti- tiagoa da.</i>	Esta alianza costó más de lo que esperaba . This alliance cost more than he expected . <i>Aliantza hau espero baino garesti- tiagoa da.</i>
Escondí rápidamente la carta . I quickly hid the letter . <i>Gutuna azkar ezkutatu nuen.</i>	Escondí rápidamente la carta . I quickly hid the card . <i>Azkar ezkutatu nuen karta.</i>
Nadie conocía el destino del avión se- cuestrado. No one knew the fate of the hijacked plane. <i>Inork ez zekien hegazkin bahituaren pa- tua.</i>	Nadie conocía el destino del avión se- cuestrado. No one knew the destination of the hi- jacked plane. <i>Inork ez zekien hegazkin bahituaren hel- muga.</i>
El tiempo vuela como una flecha Time flies like an arrow <i>Denbora gezi batek bezala hegan doa.</i>	A las moscas del tiempo les gusta una flecha Time flies like an arrow <i>Denboraren euliei gezi bat gustatzen zaie.</i>
Todavía estaba tocando cuando yo llegué. He was still playing when I arrived. <i>Oraindik jotzen ari zen ni iritsi nintze- nean.</i>	Todavía estaba jugando cuando yo lle- gué. He was still playing when I arrived. <i>Jolasean ari zen ni iritsi nintzenean.</i>
Le dio de comer a su gato. He fed her cat food. <i>Katuari jaten eman zion.</i>	Le dio comida de gato a ella. He fed her cat food. <i>Katu-jana eman zion.</i>
El crystal se rompió en pedazos. The glass broke into pieces. <i>Kristala pusketan puskatu zen.</i>	El vaso se rompió en pedazos. The glass broke into pieces. <i>Edalontzia puskatu egin zen.</i>

5.4 Taula: Hizkuntzetako batean anbiguoak diren esaldi-bikoteak eta nire ereduaren itzulpenak. Lehen laurak gaztelaniaz anbiguoak dira, eta gainontzekoak ingelesez. Anbiguotasuna hitz jakin batek sortzen duenean, nabarmenduta dago.

	Nirea (es+en→eu)	Batua.eus (es→eu)
HAC	13,72	14,56
OpenSubtitles	28,39	29,40
Biak	17,08	17,98

5.5 Taula: Nire eredu onenaren eta Batua.eus itzultzailearen BLEU puntuazioak, corpus bakoitzeko ebaluazio-multzorako eta multzo elkaturako.

OpenSubtitles (es)	No lo sabían, así que los perdono.
OpenSubtitles (en)	You didn't know that, so I forgive you.
OpenSubtitles (eu)	Ez zenekiten, beraz barkatzen dizuet.
Nirea (es+en→eu)	<i>Ez zenekiten, beraz barkatu egiten zaituztet.</i>
Batua.eus (es→eu)	<i>Ez zekiten, beraz, barkatzen diet.</i>
Google (en→eu)	<i>Ez zenekien hori, beraz barkatzen dizut.</i>
OpenSubtitles (es)	Estas personas no son sus enemigos.
OpenSubtitles (en)	These people are not your enemy.
OpenSubtitles (eu)	Pertsona hauek ez dira zuen etsaiak.
Nirea (es+en→eu)	<i>Pertsona hauek ez dira zuen etsaiak.</i>
Batua.eus (es→eu)	<i>Pertsona horiek ez dira bere etsaiak.</i>
Google (en→eu)	<i>Pertsona hauek ez dira zure etsaia.</i>
OpenSubtitles (es)	- ¿Con quién habla?
OpenSubtitles (en)	- Who is she talking to?
OpenSubtitles (eu)	-Norekin ari da hizketan?
Nirea (es+en→eu)	<i>- Norekin ari da hizketan?</i>
Batua.eus (es→eu)	<i>- Norekin ari zara?</i>
Google (en→eu)	<i>- Norekin hitz egiten du?</i>
OpenSubtitles (es)	- ¿Quiénes son?
OpenSubtitles (en)	Who are you?
OpenSubtitles (eu)	- Nor zarete?
Nirea (es+en→eu)	<i>- Nortzuk zarete?</i>
Batua.eus (es→eu)	<i>- Nortzuk dira?</i>
Google (en→eu)	<i>Nor zara?</i>
OpenSubtitles (es)	¿Cuánto tiempo hace que se esconde en mi piano?
OpenSubtitles (en)	How long have you been hiding in my piano?
OpenSubtitles (eu)	Zenbat denbora daramazu nire pianoan ezkutatuta?
Nirea (es+en→eu)	<i>Zenbat denbora daramazu nire pianoan?</i>
Batua.eus (es→eu)	<i>Noiztik dago nire pianoan ezkutatuta?</i>
Google (en→eu)	<i>Zenbat denbora daramazu ezkutatzten nire pianoan?</i>

5.6 Taula: OpenSubtitles-eko ebaluazio-multzoko kasu batzuk, bi iturri erabiltzeari esker zuzen itzultzen direnak.

6. KAPITULUA

Ondorioak eta etorkizunerako lana

Lan honetan erakutsi dut, aurretik beste lan batzuek esandakoa berretsiz, bi hizkuntzatako iturriak erabiltzea lagungarria dela neurona-sare bidezko itzulpenarako. Gaztelania eta ingelesetik euskararako itzulpenari dagokionez, badaude eredu erabilgarri bat entrenatzeko behar beste datu eskuragarri. Azpimarratuko nuke emaitza onak lortu direla datu kopuru ez oso handiarekin: gehienez 1,3 milioi hirukote. Gainera, hizkuntza horietan iturri bakarreko itzulpenak izan ditzakeen arazo konkretu batzuk erakutsi ditut, bi iturri erabiltza konpontzen direnak. Hala ere, nabarmentzekoa da horrelako eredu bat entrenatzeko datuak lortzea zailagoa dela iturri bakarreko baterako baino, eta erabilgarritasuna ere mugatuagoa dela, jatorrizko testu bera bi hizkuntzetan behar izateagatik.

Bestalde, Hizkuntzen arteko Corpuseko parekatzeak hobetzeko diseinatutako metodoa ez da lagungarria izan itzultzailea entrenatzeko orduan. Aztertutako pasarte jakin batzuetako parekatzeak nabarmen hobetzen baditu ere, dirudienez, ez da behar bezain ondo orokortzen corpus osora. Gainera, agian, jatorrizko datuetako akats kopurua ez da ikasketan kalte nabarmena eragiteko bezain handia. Dena den, hizkuntza anitzeko esaldi-bektoreei buruz ikasitakoa baliagarria izan liteke beste lan baterako.

Alde pertsonalean, proiektu honek aukera eman dit interesatzen zaidan gai bat lantzeko, itzulpen automatikoarena. Gainera, ideia propioekin esperimendu ahal izan dut, puntako teknologia aztertu eta erronka berriei aurre egin; tartean, dokumentu hau idaztea, orain arte egin behar izan dudana luzeena.

Lana egin bitartean sortutako kodea GitHub-eko biltegi batera¹ igo dut. Bertara bi iturriko

¹<https://github.com/bitalkain/GrAL>

itzultzailea erraz probatzeko aukera ematen duen notebook bat ere igo dut, ereduaren parametroekin batera.

6.1 Etorkizunerako lana

Lehenik, bi iturriko itzultzailea eraikitze moduari dagokionez, badaude alternatiba batzuk, entrenamendurako datu gehiago erabiltzeko aukera ematen dutenak. [Firat et al., 2016] artikuluan proposatutakoari jarraituz, GrAL honetako bezalako bi iturriko itzultzaile bat sortzeko, nahikoa litzateke gaztelania-euskara eta ingelesa-euskara corpusekin entrenatzea, hirukoteen beharrik gabe. Hala ere, lan horretan sare errepikariak erabiltzen direnez, aldaketa batzuk beharko lirateke bertako estrategia transformer-ekin erabiltzeko.

Beste aukera bat [Nishimura et al., 2018] artikulukoa da: ahal den kasuetan hirukoteak erabiltzen dituzte entrenatzeko, baina iturri bakarreko instantziak ere gehitzen dituzte, falta den iturriko sarreran token bakarra utziz, segmentua hutsik dagoela adierazten duena. Erraza litzateke, modu honetan, OpenSubtitles-eko gaztelania-euskara edo ingelesa-euskara parean bakarrik dauden film eta atalak gehitzea entrenamendu-multzora, eta emaitzak hobetzen al diren egiaztatzea.

Horren aldaera bat izan daiteke *back-translation* [Sennrich et al., 2016b] delakoa erabiltzea, hau da, falta den iturria hutsik utzi ordez, itzultzaile arrunt bat erabiltzea euskaratik falta den hizkuntzara itzultzeko. Baita ere, euskarazko testu elebakarretik abiatu eta bi iturriak artifizialki lortzen probatu daiteke. Izan ere, datu kopurua handitzeko teknika honek emaitza onak eman izan ditu.

Bestalde, eredu berririk ikasi gabe, baliteke emaitzak hobetzeko aukera egotea, deskodeketa [Chen et al., 2017] artikuluan planteatzen den moduan eginez. Bertan diotena da, entrenamenduko kontrol-puntuen batez besteko parametroak kalkulatzek baino emaitza hobeak ematen dituela kontrol-puntu bakoitzeko ereduarekin iragarpena egin eta lortutako probabilitateen batezbestekoa kalkulatzek, *ensemble* bat osatuz. Kontrol-puntuak erabiltzeko ideiarekin jarraituz, entrenamendu-prozesua ere aldatu liteke, [Wei et al., 2019] artikuluko metodoari jarraituz, oraindik ere emaitza hobeak ematen baititu.

Esandako guztiaz gain, sortutako itzultzailea baldintzatzen duten hiperparametro asko daudenez, balio hobeak aurkitzeko bilaketak ere egin litezke. Besteak beste, hurrengoe-kin: neurona-sarearen osagaien tamainak, BPE hiztegiaren tamaina, azken eredua osatzeko aukeratutako kontrol-puntuak eta beam search-aren zabalera.

Bukatzeko, aipatzekoa da ikasitako ereduek esaldi batzuk hika idazten dituztela. Hori arazo bat izan daiteke testu luze bat segmentuka itzuli nahi bada, esaldi batzuk zuka eta besteak hika agertuko liratekeelako, koherentzia mantendu gabe. Gainera, ziur aski, ereduak ez du bereiziko solaskidea gizonezkoa ala emakumezkoa den, adizki egokiak erabiltzeko. Beraz, hori konpontzeko, [Sennrich et al., 2016a] artikuluan planteatzen dutena euskaraz aplikatu daiteke, arau batzuen bidez entrenamenduko segmentuetan hika idatzita daudenak markatuz. Gero, erabiltzaileak aukeratu ahal izango luke itzulpena zein erregistrotan nahi duen. Dena den, ebaluazio-multzoetako itzulpenei begiratuta, esaldi gutxi batzuk bakarrik idazten ditu hika HAC corpusarekin entrenatutako eredu onenak, eta are gutxiago OpenSubtitles-etik ere ikasi duenak.

Eranskinak

A. ERANSKINA

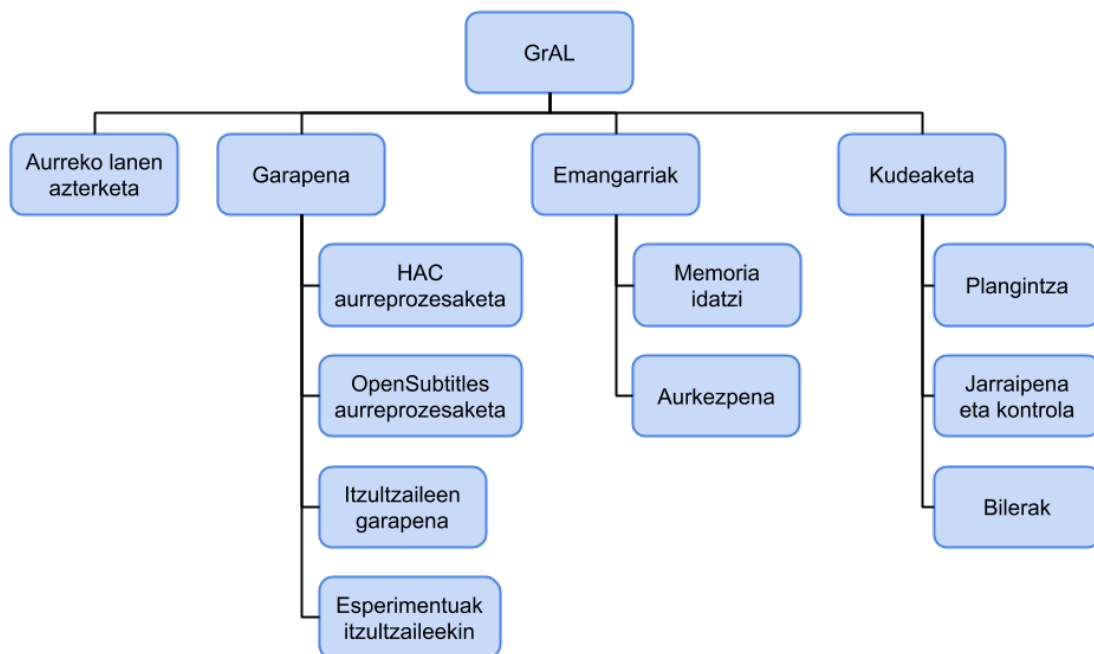
Proiektuaren kudeaketa

Eranskin honetan proiektuaren kudeaketari buruzko informazioa dago. Lehenik, proiektuaren helburu orokorrak azaltzen dira. Ondoren, egindako lana osatzen duten atazak zehaztu eta bakoitzaren garapen-denboraldiak eta emandako denbora erakusten dira, izandako desbiderapenen azalpenekin. Azkenik, erabilitako informazio- eta komunikazio-sistemak aipatzen dira.

A.1 Helburuak

Lan honen lehen helburua aurretik inplementatua nuen bi iturriko transformer-a ataza berri baterako berrerabiltzea izan da. Iturri anitzeko itzulpenak beste hizkuntza batzuetan emaitza onak eman dituela ikusita, eta ezagutzen ditudan hizkuntzekin erabiltzeko moduko bi datu-multzo handi aurkituta (HAC eta OpenSubtitles), ataza hori lantzea erabaki genuen. Beraz, lanaren helburu nagusia izan da, alde batetik, gaztelaniazko eta ingelesezko iturriak konbinatuz euskarazko testua sortzen duen itzultzaile bat lortzea eta, bestetik, bi iturriak erabiltzeak dakarren onura aztertzea. Azken hori lortzeko, iturri bakarreko ereduak ere entrenatu ditut, eta emaitzak Batua.eus itzultzaileak lortzen dituenekin konparatu ditut.

Transformer-a entrenatu ahal izateko, gaztelania-ingelesa-euskara segmentu-hirukoteak eskuratu behar izan ditut. Horretarako, corpus bakoitzaren kasuan, datuak eskuratzeko modu posibleak aztertu ditut, corpusak ahalik eta hoberen aprobeitzatzeko helburuarekin.



A.1 Irudia: LDE diagrama.

Horrekin lotuta, HAC corpuseko parekatzeek dituzten arazoak ikusita, horiek hobetzeko moduak aztertu ditut.

Proiektu hau gradu amaierako lan bat izanik, egindako lan guztia eta lortutako emaitzak azaltzen dituen memoria idatzi behar izan dut, dokumentu hauxe bera. Bukatzeko, aurkezpen bat egin beharko dut epaimahai baten aurrean.

A.2 Atazak

A.2.1 Atazen deskribapena

Proiektuaren lan-deskonposaketaren egitura (LDE) [A.1](#) irudian dago. Jarraian, egindako atazak azaltzen dira:

- **Aurreko lanen azterketa:** iturri anitzeko itzulpen automatikoari buruz argitaratu diren artikuluak irakurtzea, testuak parekatzeko tresnak bilatzea, LASER esaldibektoreei buruz irakurtzea eta etorkizunerako lana proposatzeko beste gai batzuei buruz informazioa bilatzea.

- **HAC aurreprozesaketa:** corpora eskuratzeko aukerak aztertzea, datuak webgune-tik lortzeko programa idatzi eta exekutatzeko, datuen garbiketa, parekatze-arazoen azterketa eta parekatzeak hobetzeko metodoaren garapena.
- **OpenSubtitles aurreprozesaketa:** datuen azterketa, hirukoteak osatzea eta datuen garbiketa.
- **Itzultzaileen garapena:** itzultzaileak entrenatu eta ebaluatu ahal izateko kodea idatzi eta probatzea, tokenizazioari dagokiona barne, aurretik postedizio automatikoari buruzko lanerako eginda neukana moldatu eta osatuz.
- **Esperimentuak itzultzaileekin:** ereduak modu desberdinetan entrenatu eta ebaluatzea, emaitzak aztertzea eta Batua.eus-en APIrako sarrera lortu eta erabiltzea.
- **Memoria idatzi:** egindako lana eta lortutako emaitzak azaltzen dituen memoria idatzea, bertan gehitzeko irudiak sortzea eta LaTeX-en hainbat gauza nola egin biltzea barne.
- **Aurkezpena:** lanaren defentsarako diapositibak eta esan beharrekua prestatu eta epaimahaiaren aurrean aurkeztea.
- **Plangintza:** hasierako plangintza egitea eta beharrezkoa izatean eguneratzea.
- **Jarraipena eta kontrola:** proiektuaren garapenaren jarraipena eta egunero ataza bakoitzari emandako denboraren kontrola.
- **Bilerak:** tutoreekin biltzea egindako lana azaldu, zalantzak argitu eta hurrengo pausoak zehazteko.

A.2.2 Atazen garapen-denboraldiak

Ataza bakoitzarekin zein astetan aritu naizen [A.2](#) irudiko Gantt diagraman agertzen da. Diagrama hori egindako lanaren jarraipenari dagokio, ez hasierako plangintzari. Hasiera batean, lana uztailean aurkeztea zen ezarritako helburua. Azkenean, ordea, apirilean hasita hilabeteko tartea hartu nuen geratzen zitzaizkidan irakasgaietan zentratzeko, lana bukatzeko data atzeratuz.

Ataza	Hasierako estimazioa (h)	Emandako denbora (h)
Aurreko lanen azterketa	10	9
HAC aurreprozesaketa	45	84
OpenSubtitles aurreprozesaketa	45	17
Itzultzaileen garapena	30	21
Esperimentuak itzultzaileekin	30	26
Memoria idatzi	100	111
Aurkezpena	15	15*
Plangintza	5	5
Jarraipena eta kontrola	5	5
Bilerak	15	15
Guztira	300	308

A.1 Taula: Ataza bakoitzerako, hasieran estimatutako denbora eta benetan emandakoa. (*) Txosten hau idazteko momentuan aurkezpena egin gabe dagoenez, estimazioa agertzen da berriro.

A.2.3 Atazei emandako denbora

Ataza bakoitzari eskaini beharreko denboraren hasierako estimazioa eta benetan emandako denbora agertzen dira [A.1](#) taulan. Desbiderapen handiena corpus bakoitzaren aurreprozesaketari eskainitako denboran egon da. Izan ere, hasiera batean suposatu nuen OpenSubtitles-etik hirukoteak lortzea lan zailagoa izango zela eta, agian, hizkuntza desberdinetako testuak konparatzeko tresnaren bat beharrezkoa izango zela. Azkenean, ordea, datu kopuru handia lan gutxiagorekin lortu dut. Horregatik, denbora gehiago eskaini diot HACen aurreprozesaketari, LASER esaldi-bektoreak erabiliz parekatzeak hobetzeko modua bilatuz, nahiz eta ez zen beharrezkoa itzultzaileak ikasteko eta, gainera, ez dudana lortu emaitza onak ematea corpus osorako.

A.3 Informazio- eta komunikazio-sistemak

A.3.1 Informazio-sistema

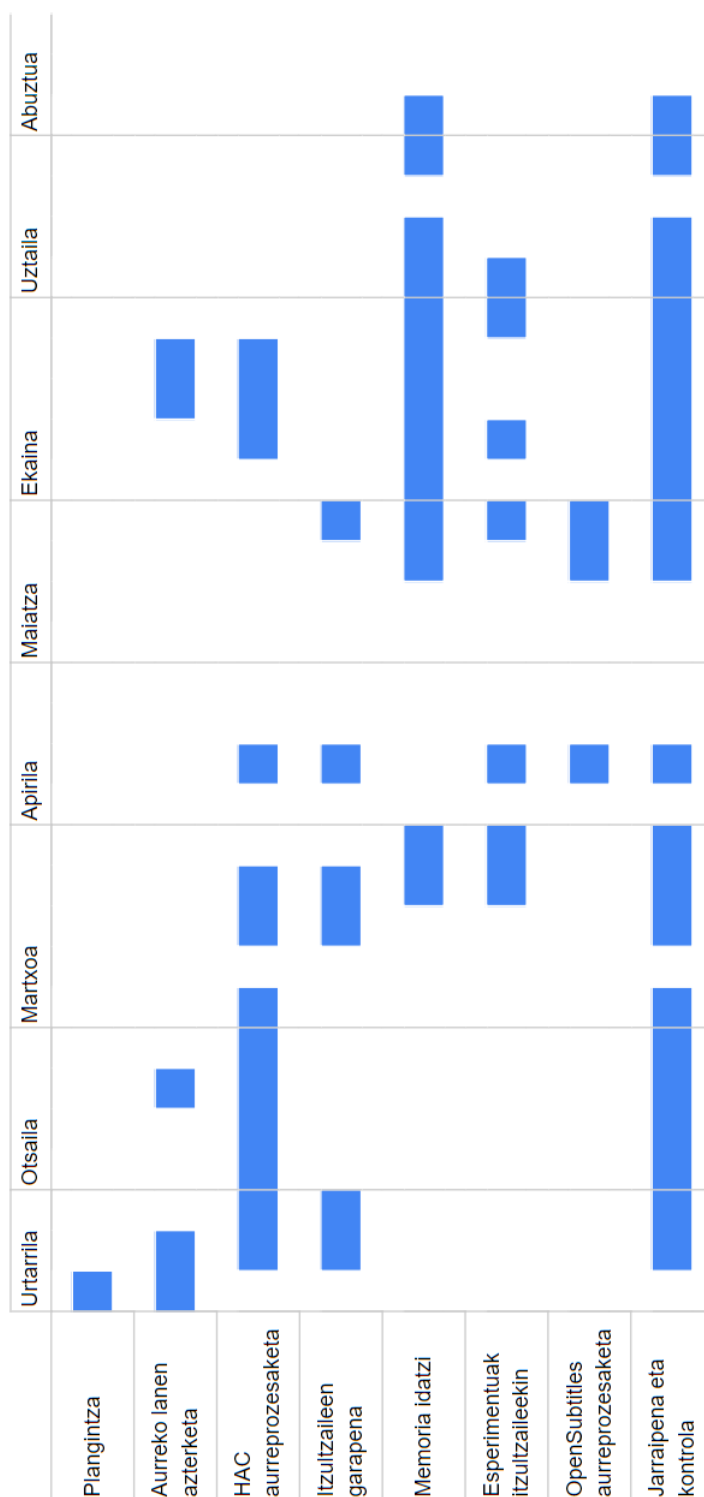
Kodearen garapena Google Colab tresnaren bidez egin dudanez, idatzitakoa automatikoki gorde da Google Drive-n. Bertan, tutoreekin partekatutako karpeta batean gorde ditut idatzitako notebook-ak, eta nire unitatean bestelako fitxategiak: adibidez, entrenamendu eta ebaluaziorako testuak, entrenatutako ereduaren parametroen fitxategiak eta LASER tresnak sortutakoak. Horietako batzuk biltegiatze-espazio handia hartzen dutenez, eta Google-k

dohainik 15 GB besterik eskaintzen ez dituenek, noizbehinka fitxategi batzuk ordenagailu batean deskargatzen eta txandaka berriro igotzen aritu behar izan dut.

Memoria LaTeX-en idazteko Overleaf erabili dut, proiektua tutoreekin partekatuta, eta irudi batzuk sortzeko Google Drawings eta Lucidchart. Lanaren jarraipena egiteko, Google Keep-en idatzi dut egun bakoitzean egindakoa, ondoren informazioa Google Sheets-era pasatu eta antolatzeko. Kasu horietan guztietan ere, sortutako fitxategiak hodeian gordetzen dira, eta edonondik atzitu daitezke. Noizean behin, memoria, notebook-ak eta jarraipenaren oharra nire eramangarrian deskargatu ditut, badaezpada.

A.3.2 Komunikazio-sistema

Lanarekin aritu naizen denboraldietan, bilerak izan ditut tutoreekin ia astero, guztira 21. Hasierako bilerak presentzialak izan ziren, eta hurrengoak Jitsi Meet bidezko bideodeiak, COVID-19aren aurka hartutako neurriak zirela eta. Bileretatik kanpo komunikatzeko posta elektronikoa erabili dugu. Horrez gain, memoriari buruzko iruzkinak Overleaf zerbitzuan bertan utzi dizkirate tutoreek.



A.2 Irudia: Gantt diagrama.

Bibliografia

- [Artetxe and Schwenk, 2018] Artetxe, M. and Schwenk, H. (2018). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464.
- [Artetxe and Schwenk, 2019] Artetxe, M. and Schwenk, H. (2019). Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.
- [Chen et al., 2017] Chen, H., Lundberg, S., and Lee, S.-I. (2017). Checkpoint ensembles: Ensemble methods from a single training process.
- [Dabre et al., 2017] Dabre, R., Cromieres, F., and Kurohashi, S. (2017). Enabling multi-source neural machine translation by concatenating source sentences in multiple languages.
- [Firat et al., 2016] Firat, O., Sankaran, B., Al-onaizan, Y., Yarman Vural, F. T., and Cho, K. (2016). Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.
- [Garmash and Monz, 2016] Garmash, E. and Monz, C. (2016). Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan. The COLING 2016 Organizing Committee.
- [Junczys-Dowmunt and Grundkiewicz, 2018] Junczys-Dowmunt, M. and Grundkiewicz, R. (2018). MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on*

Machine Translation: Shared Task Papers, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

- [Lison and Tiedemann, 2016] Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- [Nishimura et al., 2018] Nishimura, Y., Sudoh, K., Neubig, G., and Nakamura, S. (2018). Multi-source neural machine translation with missing data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 92–99, Melbourne, Australia. Association for Computational Linguistics.
- [Och and Ney, 2001] Och, F. J. and Ney, H. (2001). Statistical multi-source translation. In *MT Summit VIII: Machine Translation in the Information Age, Proceedings*, pages 253–258, Santiago de Compostela, Spain.
- [Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- [Sarasola et al., 2015] Sarasola, I., Salaburu, P., and Landa, J. (2015). *Hizkuntzen Arteko Corpusa (HAC)*. UPV/EHU (Euskara Institutua).
- [Sennrich et al., 2016a] Sennrich, R., Haddow, B., and Birch, A. (2016a). Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.
- [Sennrich et al., 2016b] Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- [Sennrich et al., 2016c] Sennrich, R., Haddow, B., and Birch, A. (2016c). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.Ñ., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[Wei et al., 2019] Wei, H.-R., Huang, S., Wang, R., Dai, X.-y., and Chen, J. (2019). On-line distilling from checkpoints for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1932–1941, Minneapolis, Minnesota. Association for Computational Linguistics.

[Zoph and Knight, 2016] Zoph, B. and Knight, K. (2016). Multi-source neural translation.