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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Euskarazko Zuzentzaile Gramatikal Neuronala

Egilea

Ariane Méndez Amuchategui

2020

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

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Euskarazko Zuzentzaile Gramatikal Neuronala

Egilea

Ariane Méndez Amuchategui

Zuzendariak

Maite Oronoz

Gorka Labaka

Laburpena

Proiektu honetan ikasketa sakoneko teknikak erabiltzen dira, Transformer izeneko arkitektura neuronala zehazki, euskarazko errore gramatikalak automatikoki zuzendu ahal izateko. Itzulpen automatikoko hurbilpen bat erabilita, testu erroredunak, testu zuzenetara itzultzen dira. Euskarak ataza hau bereziki zaila egiten duten ezaugarri batzuk ditu, besteak beste, aditz laguntzaileak komunztadura egiten du subjektuarekin, objektuarekin eta zehar objektuarekin eta hauek esaldian ager daitezke edo ez (elipsia). Bestalde, hizkuntza "ergatiboa" izanik, aditza trantsitiboa izan edo ez, subjektuak marka ezberdina darama. Lan honetan esaldi zuzenetan erroreak txertatu ahal izateko esaldiek analisia behar ote duten aztertzen da eta erroreak automatikoki sortzen dituen sistema bat diseinatzen da, ondoren guk sortutako corpusekin zuzentzailea entrenatzeko.

Gaien aurkibidea

Laburpena i

Gaien aurkibidea iii

Irudien aurkibidea vii

Taulen aurkibidea ix

1 Sarrera 1

2 Proiektuaren Helburuen Dokumentua 3

2.1 Proiektuaren deskribapena eta helburuak 3

2.2 Plangintza 4

2.2.1 LDE diagrama 4

2.2.2 Lan-paketeak 5

2.2.3 Gantt diagrama 7

2.2.4 Lan-metodologia 8

2.2.5 Emangarriak 9

2.2.6 Arriskuak eta prebentzioa 9

2.3 Plangintzarekiko Desbiderapenak 10

3 Aurrekariak **13**

3.1 Itzulpen Automatikoa . . . 13

3.1.1 Erregeletan oinarritutako itzulpen automatikoa . . . 14

3.1.2 Corpusean oinarritutako itzulpen automatikoa . . . 15

3.2 Zuzenketa Gramatikala . . . 17

3.2.1 Datuetan oinarritutako zuzenketa gramatikala . . . 18

3.2.2 Itzulpen automatikoa eta zuzenketa gramatikala . . . 19

3.3 Hizkuntza-ereduak . . . 20

3.3.1 Kontaketan oinarritutako hizkuntza-ereduak . . . 20

3.3.2 Espazio jarraituko hizkuntza-ereduak . . . 23

4 Itzulpenerako eta Zuzenketarako Arkitektura Neuronalak **25**

4.1 Sare Neuronal Errekurrenteak . . . 26

4.2 Long-Short Term Memory . . . 28

4.3 Atentzioa . . . 29

4.4 Sare Neuronal Konboluzionalak . . . 31

4.5 Transformer arkitektura . . . 34

5 Diseinua eta Inplementazioa **37**

5.1 Erroreen analisia . . . 38

5.2 Datu-multzoa eta aurre-prozesaketa . . . 39

5.3 Erroreen sorkuntza . . . 40

5.3.1 Hizkuntza-eredua . . . 41

5.4 Corpus desberdinen diseinua eta sorkuntza . . . 42

5.4.1 Entrenamendurako corpusak . . . 42

5.4.2 Garapenerako corpusa . . . 45

5.4.3 Ebaluaziorako corpusa . . . 45

5.5 *Byte Pair Encoding* 47
5.6 Sistema 48
5.6.1 Konfigurazioa eta Inplementazioa 48
5.6.2 Entrenamendua 53
5.6.3 Sorkuntza 54

6 Ebaluazioa eta Emaitzak **57**
6.1 Ebaluazio-metrikak 57
6.1.1 ERRANT 57
6.1.2 GLEU 60
6.2 Garapenerako ebaluazioa 61
6.3 Ebaluazio finala eta Emaitzak 62

7 Ondorioak eta Etorkizunerako Lana **69**
7.1 Ondorioak 69
7.1.1 Proiektuaren ondorioak 69
7.1.2 Ondorio pertsonalak 70
7.2 Etorkizunerako Lana 71

Eranskinak

A Euskarazko erroreen sailkapena **75**

B Garapenerako ebaluazioan lortutako zuzenketak **85**

C Laugarren sistemak esaldi zuzenetarako proposatutako zuzenketak **91**

Bibliografia **93**

Irudien aurkibidea

2.1 Itzulpen automatikoa eta zuzenketa gramatikalaren arteko paralelismoa. . 3

2.2 Lanaren Deskonposaketa Egitura diagrama. 4

2.3 Lan-pakete bakoitzeko atazak ebazteko behar izango ditugun orduen aurreikuspena. 7

2.4 Gantt diagrama. 8

2.5 Lan-pakete bakoitzeko atazak ebazteko aurreikusitako denbora, atazak ebazteko benetan erabili den denbora eta bien arteko aldea, ordutan. . . . 11

3.1 Vauquois-en piramidea. 16

4.1 Sare Neuronal Errekurrenteen arkitektura. 27

4.2 Sare neuronal errekurrentearen erabilera itzulpen automatikoan. 27

4.3 LSTM sarearen arkitektura. 28

4.4 *Seq2Seq* modeloa atentzioa erabiliz. 29

4.5 Atentziodun itzulpen automatikoaren adibidea. Ingelesezko hitz bakoitza lortzeko euskarazko zein hitzetan jarri den atentzioa adierazten da. 30

4.6 Iragazkien erabilera Sare Neuronal Konboluzionaletan. 32

4.7 CNN-en erabilera hizkuntzaren prozesamenduan. 33

4.8 Wavenet sarea, testu-ahots eraldaketan erabiltzen den CNN baten adibidea. 33

4.9 Transformer ereduaren arkitektura. 34

4.10 Kodetzailea 3 hitzeko esaldi bat jasotzen. 35

4.11 Transformer arkitekturako kodetzaile- eta deskodetzaile-osagaien funtzionamendua zuzenketa gramatikalerako. Kasu honetan pila 3 elementuz osatuta dago. 36

5.1 Posizioaren araberako kodeketa-matrizea 5.1 eta 5.2 ekuazioek definitutako matrize konstantea da. Hitzen *embedding*-ari gehitzean lortzen den matrizeak hitzen esanahiaren eta posizioaren informazioa gordetzen du. . 49

5.2 Atentzioan jarraitzen den prozedura orokorra. 50

5.3 Atentzioan jarraitzen den prozedura 2 hitzeko esaldi baten kasuan. 51

5.4 Buru-Anitzeko Atentzio mekanismoa. 52

5.5 Hondar-konexioak eta normalizazioa kodetzailearen kasuan. 53

6.1 ERRANT-ek proposatutako anotazioak. 58

6.2 *Span-based Correction*, *Span-based Detection* eta *Token-based Detection* irizpideen konparazioa. 59

6.3 ERRANT-ek proposatutako anotazioak sistemak sarrerako-esaldi guztiak zuzenak direla antzematen duen kasuan. 64

6.4 ERRANT-ek proposatutako anotazioak sistemak sarrerako-esaldiak zuzenak direla kasu guztietan antzematen ez badu. 65

Taulen aurkibidea

3.1 Adibideetan oinarritutako itzulpen automatikoan erabiltzen den corpusaren adibidea. 16

5.1 Corpusen ezaugarriak: esaldi kopurua, hitz kopurua, token kopurua eta token horien artean desberdinak direnen kopurua. 39

5.2 Ausazko errore sorkuntzaren adibideak. 40

5.3 Diseinatutako corpusen ezaugarriak laburbiltzen dituen taula. "∼" ikurrak balio zehatza ezagutzen ez dela adierazten du, esaldi horiek esaldi erroredunak eta zuzenak nahastuta zituen datu-multzo batetik ausaz hautatu baitira. 42

5.4 err_zuz400K datu-multzoarekin lortutako zuzenketa batzuk. 44

5.5 BPE teknikaren adibidea. 47

6.1 Esaldi erroredunen, esaldi zuzenen eta sistemak proposatutako zuzenketen adibideak. 58

6.2 ERRANT tresna 6.1 irudiko fitxategiekin erabiliz lortutako irteera. 59

6.3 Entrenatutako sistema bakoitzak garapenerako esaldiei egindako zuzenketetan jasotako ERRANT ebaluazioa. 61

6.4 Automatikoki sortutako esaldi erroredunetan sistema bakoitzak lortutako emaitzak, ERRANT metrika erabiliz neurtuak. 63

6.5 ERRANT-ekin lortutako balioak 6.3 irudiko fitxategiak ebaluatzean. . . . 64

6.6 ERRANT-ekin lortutako balioak 6.4 irudiko fitxategiak ebaluatzean. . . . 65

6.7 Esaldi erroredun errealetan eta esaldi zuzenetan sistema bakoitzak lortutako emaitzak, ERRANT metrika eta *difflib* modulua erabiliz neurtuak. . . 65

6.8 Laugarren sistemak esaldi erroredun errealetan modu egokian zuzendutako 8 erroreak. 66

6.9 Laugarren sistemak esaldi zuzenetarako proposatutako aldaketa batzuk. Esaldi zuzenetarako proposatutako aldaketa guztiak C eranskinean aurki daitezke. 67

6.10 Sistema bakoitzak lortutako GLEU puntuazioa. Lehenengo errenkadan sistema entrenatzeko erabilitako corpusa zein izan den adierazten da. . . . 68

C.1 Laugarren sistemak esaldi zuzenetarako proposatutako zuzenketak. 92

1. KAPITULUA

Sarrera

Komunikazioa esanahia duen edozer adierazi, jaso eta trukatzean datzan prozesu sozial eta kontzientea da. Komunikazio prozesuan parte hartzen duten elementu nagusiak mezua (informazioa), igorlea (mezua bidaltzen duena) eta hartzailea (mezua jasotzen duena) dira. Igorleak mezu bat bidali nahi duen bakoitzean hainbat erabaki hartu behar ditu: zein hitz erabili, esaldiak nola eraiki, mezua hartzailearen ezagutza-mailarekin bat datorren testuinguru batera nola egokitu eta zein den gramatika aproposa, besteak beste. Erabaki hauek guztiak modu egokian hartzeak hartzailearen lana (mezua interpretatu eta ulertzea) erraztuko du eta komunikazioa arrakastatsua izateko aukerak nabarmenki handituko dira.

Igorleak igortzen duen mezuan gramatika-erroreak daudenean, hau da, hizkuntza ez zuzena erabiltzen denean, komunikazioa zaildu daiteke. Gramatika hizkuntza baten erabilera, hiztegia, esaldien konposaketa eta gainontzeko elementu sintaktikoak definitzen dituzten erregelen ikerketa da. Arlo honek morfologia eta sintaxia hartzen ditu bere baitan, kasu batzuetan fonetika ere kontuan hartzera helduz. Laburki azalduta, morfologia hitzen barne egitura aztertzen duen arloa da, hitz-unitateak definitu eta sailkatzeko eta hitz berriak sortzeko; sintaxia hitzak bateratzeko moduak (sintagmak eta esaldiak) aztertzen dituen arloa da eta fonetika gizakion arrazoibidearen soinu fisikoen azterketa da.

Hiru arlo hauek komunikazioaren oinarri dira. Morfologia eta sintaxia transmititu nahi den edozein mezuren funtsezko atalak direla ukaezina da, esaldiak (eta esaldiak osatzen dituzten hitzak) erabili gabe ezinezkoa litzatekeelako ezagutzen ditugun komunikatzeko modu nagusiak burutzea, hitz egitea eta idaztea alegia. Fonetika ezinbestekoa da hitzezko komunikazioan, baina lan honetan idatzizko komunikazioa bakarrik aztertuko dugu.

Argi dago, beraz, diskurtso batean erabiltzen den gramatikak igortzen ari den mezuan eragina duela. Transmititu nahi den informazioan gramatika okerra erabiltzen bada mezuaren kalitatea murrizten da eta aldi berean igorlearen ospea eta jakinduria zalantzatan jartzen dira. Igorleak ez badu bere hizkuntza ezagutzen, nola izango da gai hartzaileari mezua modu egokian transmititu edo azaltzeko?

Proiektu honetan, gramatika zuzena erabiltzeak duen garrantzia ikusita, euskaraz idatzitako testuentzako zuzentzaile gramatikal bat eraikitzea proposatzen dugu. Horretarako problema itzulpen automatikoko ataza bat izango balitz bezala planteatuko dugu. Itzulpen automatikoan jatorri-hizkuntza jakin bateko esaldi bat edukiko dugu eta helburua esaldi hori helburu-hizkuntza jakin batera (jatorri-hizkuntza ez den beste bat) itzultzea izango da. Zuzenketa gramatikalean jatorri-hizkuntzako esaldi bat eduki ordez akats gramatikalak dituen euskarazko esaldi bat edukiko dugu (hemendik aurrera “euskara erroreduna” deituko diogu) eta helburu-hizkuntzara itzultzea ordez, xedea gramatikalki zuzena den euskarazko esaldia (esaldi zuzena) lortzea izango da.

Memoria honetan lehenenik eta behin proiektuaren helburuen dokumentua aurki daiteke, 2. kapituluan, non lanaren deskribapena, helburuak, jarraitutako plangintza eta plangintza horrekiko desbiderapenak adierazten diren. 3. kapituluan proiektu honetan landutako tekniken aurrekariak adierazten dira, itzulpen automatikoak izandako eboluziotik abiatuz zuzenketa gramatikala burutzeko gaur egun erabiltzen diren metodoetara arte. Problema ebazteko erabili diren arkitekturak eta bakoitzaren ahuleziak gainditzeko egindako proposamenak 4. kapituluan azaltzen dira, gaur egun itzulpen eta zuzenketa lanetan gehien erabiltzen den arkitekturara heldu arte. 5. kapituluan proiektuaren diseinu eta inplementazioan landu diren atal guztiei buruz hitz egiten da, hala nola erabilitako datuak, corpus artifizialaren sorrera, datuetan erabilitako prozesamendurako tresnak, hautatutako arkitektura eta egindako entrenamendua. 6. kapituluan gure sistemaren kalitatea neurtzeko jarraitutako neurriei buruzko azalpena ematen da eta lortutako emaitzak aztertzen dira. Azkenik, 7. kapituluan, lan honekin ateratako ondorioei buruz eta etorkizuneko lanari buruz hitz egiten da.

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

2.1 Proiektuaren deskribapena eta helburuak

Helburu nagusia euskaraz idatzitako testuentzako zuzentzaile gramatikal bat inplementatzea da. Horretarako zuzenketa gramatikala itzulpen automatikoaren kasu berezi bezala planteatzea erabaki dugu. Itzulpen automatikoan jatorri-hizkuntza jakin batean idatzitako testu bat beste helburu-hizkuntza jakin batera itzultzen da. Zuzenketa gramatikala itzulpen bezala ulertzen badugu, jatorri- eta helburu-testuak hizkuntza berean idatzita egongo dira, jatorria testu erroreduna izango delarik eta helburua jatorrizko testuaren zuzenketa.

2.1 irudian ikusi daiteke adibide bat. Bertan, itzulpen automatikoaren kasuan, jatorri-hizkuntza gaztelania da eta helburu-hizkuntzako, kasu honetan euskara, itzulpena lortzen da; era berean, zuzenketa gramatikalaren kasuan, euskara erroredunetik abiatuta euskara zuzenerako "itzulpena" (zuzenketa) lortzen da.

2.1 Irudia: Itzulpen automatikoa eta zuzenketa gramatikalaren arteko paralelismoa.

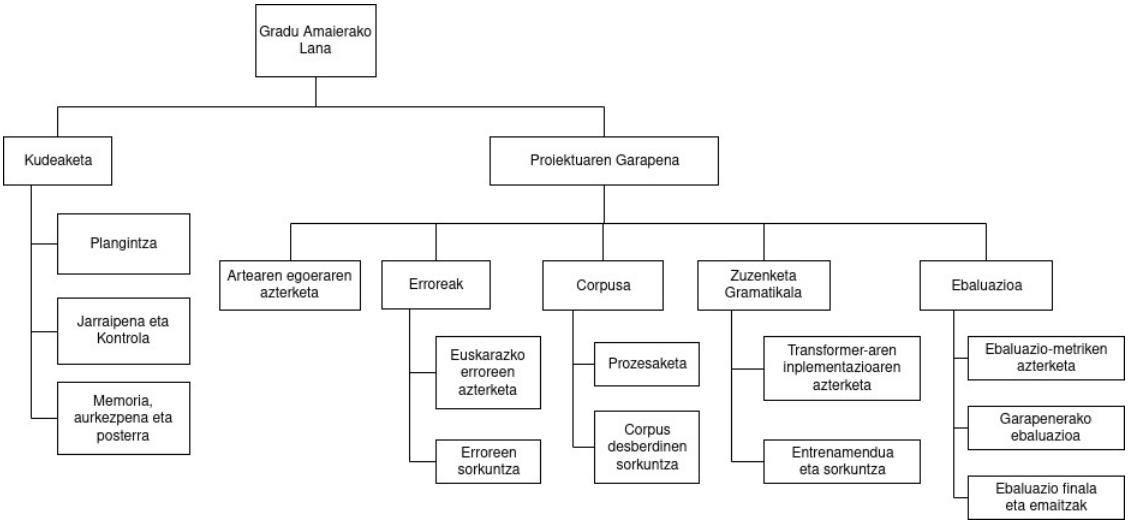
Bigarren mailako helburu bezala hurrengoak finkatu ditugu:

- Sorkuntzarako behar duten informazio kopuruaren arabera euskarazko erroreen sailkapen bat egitea.
- Zuzentzaile neuronal batek zenbat testu erroredun eta zenbat zuzen behar duen aztertzea.
- Zuzentzaile bakar batek errore guztiak zuzen ditzakeen edo errore mota bakoitzeko zuzentzaile bat behar ote den aztertzea.

2.2 Plangintza

2.2.1 LDE diagrama

2.2 irudian ikus daiteke gure proiekturako planteatutako Lanaren Deskonposaketa Egitura (LDE). Diagraman adierazten den bezala, alde batetik proiektuaren kudeaketa eta aurkezpena landu dira eta, bestetik, proiektuaren garapena. Kudeaketa atalean plangintza, memoria, defentsa eta posterrarekin zerikusia duten atazak biltzen dira. Garapen atalean egindako ikerketa, diseinua, inplementazioa eta ebaluazioarekin zerikusia duten atazak aurki daitezke. Diagraman agertzen diren lan-pakete guztiak zehaztasun handiagoarekin azaltzen dira 2.2.2 atalean.



2.2 Irudia: Lanaren Deskonposaketa Egitura diagrama.

2.2.2 Lan-paketeak

Atal honetan LDE diagraman (2.2 irudia) definitu diren lan-pakete eta atazen azalpen zehatzago bat ematen da. Horrez gain, 2.3 irudian, lan-pakete eta ataza bakoitza ebazteko behar izango dugun ordu-kopuruaren estimazioa erakusten da.

Plangintza

Ataza honetan proiektuaren planifikazioa burutu da: helburuen definizioa, lanaren banaketa, ataza bakoitzaren iraupenaren estimazioa, arrisku posibleen identifikazioa... Ezinbestekoa da faktore hauek guztiak zehaztea eta antolatzea, lana modu eraginkorrean aurrera eramateko eta emangarrien entrega-epeak arazorik gabe betetzeko.

Jarraipena eta Kontrola

Ataza honetan plangintzan zehaztutako mugarriak eta epeak betetzen ari garela eta proiektuaren bideragarritasuna mantentzen dela bermatuko da. Proiektuaren bizi-ziklo osoan zehar landuko dugu ataza hau, batez ere kontrol-bilerak eginez. Bilera hauek, salbuespenik ezean, astero burutuko dira eta bertan plangintza jarraitzen dela eta asteroko zereginak modu egokian gauzatu direla ziurtatuko da.

Memoria, aurkezpena eta posterra

Ataza honek proiektuaren emangarrietako batzuk biltzen ditu. Alde batetik memoriaren idazketa landuko da, dokumentu hau bera, proiektuan egindako lana eta erabilitako baliabideak azalduz. Bestetik, aurkezpena burutuko da, proiektuaren defentsarako beharrezko informazioa bertan adieraziz. Azkenik, proiektua aurkezteko posterra ere egingo da.

Artearen egoeraren azterketa

Ataza honetan gaur egun zuzenketa gramatikal automatikoa egiteko erabiltzen diren teknikak aztertuko dira eta proiektua burutzeko zein erabili aukeratuko da.

Euskarazko erroreen azterketa

Ataza honetan euskaraz egiten diren errore gramatikalak identifikatzea izango da helburua. Erroreak automatikoki sortu ahal izateko, euskaraz ager daitezkeen erroreak sailkatuko dira eta errore horiek sortzeko analisia beharrezkoa ote den erabakiko da.

Erroreen sorkuntza

Ataza honetan, aurreko atazako azterketan oinarrituta, erroreak automatikoki sortzeko kodea inplementatuko da.

Prozesaketa

Ataza honetan corpusaren prozesaketa landuko da. Alde batetik, esaldi erroredunak sortzeko baliatuko diren datu-multzoak aurre-prozesatuko dira, tokenizazio eta egitura finko batetik abiatzeko. Bestetik, diseinatutako corpus bakoitza entrenamendurako prestatuko da *Byte Pair Encoding* teknika erabiliz.

Corpus desberdinen sorkuntza

Ataza honetan proiektuan garatuko diren sistemak entrenatzeko erabiliko diren corpusak diseinatu eta sortuko dira. Behar izatekotan, uneko corpusari gehitu beharrekoa edo egin beharreko aldaketak erabakiko dira eta corpus berriak sortuko dira.

Transformer-aren inplementazioaren azterketa

Ataza honetan Transformer izeneko arkitekturaren inplementazio bat oinarri bezala hartu, aztertu eta ulertuko da. Hiperparametro batzuk zehaztu eta datuen irakurketa moldatu ondoren, inplementazio hori erabiliko da gure sistemak entrenatzeko. 4.5 atalean zehaztuko dugu hobeto zein ezaugarri dituen Transformer arkitektura neuronalak.

Entrenamendua eta sorkuntza

Ataza honetan zuzentzaile gramatikalaren azpian dagoen sare neuronalaren entrenamendua burutuko da eta sistemak euskaraz idatzitako esaldien zuzenketak sortzeko gai direla egiaztatuko da.

Ebaluazio-metriken azterketa

Ataza honetan zuzenketen kalitatea neurtzeko zein metrika erabiliko diren erabakiko da eta metrika horiek inplementatzen dituzten tresnak eta kodeak aztertuko dira.

Garapenerako ebaluazioa

Ataza honetan sistema bakoitzarekin esaldi-kopuru murriztu bat zuzenduko da automatikoki, behar izatekotan, zuzenketa horien kalitatean oinarrituta, hurrengo entrenamendurako corpusa diseinatzeko.

Ebaluazio finala eta emaitzak

Ataza honetan, ebaluazio-metrika desberdinak erabiliz, diseinatutako corpusekin entrenatutako sistemek sortzen dituzten zuzenketen kalitatea neurtuko da.

LAN-PAKETEA	DENBORA-AURREIKUSPENA (ordutan)
Kudeaketa	130
Plangintza	5
Jarraipena eta Kontrola	45
Memoria, aurkezpena eta posterra	80
Proiektuaren Garapena	170
Artearen egoeraren azterketa	10
Erroreak	20
Euskarazko erroreen azterketa	5
Erroreen sorkuntza	15
Corpusa	55
Prozesaketa	5
Corpus desberdinen sorkuntza	50
Zuzenketa Gramatikala	55
Transformer-aren inplementazioaren azterketa	20
Entrenamendua eta sorkuntza	35
Ebaluazioa	30
Ebaluazio-metriken azterketa	5
Garapenerako ebaluazioa	5
Ebaluazio finala eta emaitzak	20
GUZTIRA	300

2.3 Irudia: Lan-pakete bakoitzeko atazak ebazteko behar izango ditugun orduen aurreikuspena.

2.2.3 Gantt diagrama

2.4 irudian ikus daiteke proiekturako garatutako Gantt diagrama. Bertan lan-pakete guztiak agertzen dira, bakoitzaren iraupena adieraziz denboran zehar, hasiera- eta bukaera-data zehatzekin batera.

Errore sorkuntzari eta ebaluazio-metriken azterketari dagozkien atazetan bi hasiera- eta bukaera-data adierazi direla ikus daiteke. Ataza horiek bi fasetan banatu direlako gertatzen da hori: errore sorkuntzarako bi metodo desberdin garatu dira eta bi ebaluazio-metrika aztertu dira.

Lan-paketeak			Hasiera	Bukaera	2019				2020							
					09	10	11	12	01	02	03	04	05	06	07	08
Kudeaketa	Plangintza		19/09/20	19/09/30												
	Jarraipena eta Kontrola		19/09/20	20/08/31												
	Memoria, aurkezpena, posterra		20/03/18	20/08/31												
Proiektuaren Garapena	Artearen egoeraren azterketa		19/09/20	19/10/04												
	Erroreak	Euskarazko erroreen azterketa	19/10/04	19/10/11												
		Erroreen sorkuntza	19/10/11	19/11/08												
			20/05/26	20/06/25												
	Corpusa	Prozesaketa	19/10/11	20/05/26												
		Corpus desberdinen sorkuntza	19/11/08	20/06/25												
	Zuzenketa Gramatikala	Transformer-aren inplementazioaren azterketa	19/11/15	20/01/16												
		Entrenamendua eta sorkuntza	20/01/16	20/06/29												
	Ebaluazioa	Ebaluazio-metriken azterketa	20/02/27	20/03/25												
			20/05/22	20/06/08												
		Garapenerako ebaluazioa	20/05/22	20/07/06												
		Ebaluazio finala eta emaitzak	20/05/22	20/07/13												

2.4 Irudia: Gantt diagrama.

2.2.4 Lan-metodologia

Proiektu hau hizkuntzaren prozesamenduaren inguruan lan egiten duen IXA ikerketa-taldeak eskainitako 300 orduko praktika akademiko bezala garatu da. Lana IXA taldeko ikerlariek zuzendu dute eta ikerketa-taldearen baliabideak erabili dira proiektuan zehar, ikasketa prozesurako behar izan diren *Graphics Processing Unit-ak (GPU)* edo corpusak sortzeko erabilitako datuak adibidez.

Proiektuaren iraupen osoan zehar zuzendariekin bilerak adostu dira, proiektuaren jarraipena eta kontrola egiteko asmoz. Bilera hauek, salbuespen ezean, astero egin dira eta normalean ordubeteko iraupena izan dute. Hasiera batean bilerak Donostiako Informatika Fakultatean egin dira baina, ezusteko osasun-egoerak ekarritako itxialdia dela eta, azkenengo hilabeteetan bilerak *Blackboard Collaborate* plataforma erabiliz egin dira. Bilera hauetako bakoitzean proiektuaren uneko egoera aztertu da, arazorik egotekotan konponbidea topatu zaio eta hurrengo bilerarako zereginak zehaztu dira, ondoren ikasleak astean zehar zeregin horietan lan egiteko.

2.2.5 Emangarriak

Proiektu honetan bi motatako emangarriak sortu dira:

1. Proiektuaren garapenarekin zerikusia dutenak
    - Zuzentzaile gramatikala inplementatzen duen kodea.
    - Zuzentzaile gramatikalak entrenatzeko erabili diren lau corpus.
    - Aurreko corpusak sortzeko erabili den kodea.
    - Corpusak sortzeko erabili diren datu-multzoen ezaugarriak (esaldi kopurua, token kopurua, eta abar) zehazteko kodea.
    - Emaitzak aztertzeko erabili den kodea.
2. Proiektuaren kudeaketarekin zerikusia dutenak
    - Proiektuaren memoria, dokumentu hau bera.
    - Proiektuaren defentsan erabiliko diren gardenkiak.
    - Proiektua aurkezteko posterra.

2.2.6 Arriskuak eta prebentzioa

Garapenean zehar proiektuaren arrakasta arriskuan jarri dezaketen hainbat oztopo agertu daitezke. Hori ekiditeko, garrantzitsua da suertatu daitezkeen arazoak aurreikustea eta arriskuak ekiditen lagunduko duten prebentzio-neurriak hartzea. Atal honetan proiektuaren hasieran identifikatutako arriskuak eta prebentzio-neurriak adierazten dira.

Arriskuak

- Ikasketa sakoneko eredu bat erabiltzeak suposatzen duen ikasketa-denbora luzea.
- Ikaslearen ordenagailuak tamaina handiko datuak prozesatzeko gaitasuna ez izatea.
- Proiektuaren garapenerako garrantzitsuak diren datuen galera.

Prebentzioak

- Ikasketa prozesuak exekuzio luzeak egitea suposatzen duenean beste ataza batzuk aurreratuko dira paraleloan, denbora-galerarik ez edukitzeko.
- Tamaina handiko datuak prozesatzeko gaitasuna ziurtatzeko *Google Colaboratory* plataformak eskaintzen dituen GPU-ak erabiliko dira. Nahikoa ez izatekotan, ikasleari IXA taldeko zerbitzarietan kontu bat zabaldu zaio, bertan datuak gorde eta exekuzioak egin ahal izateko.
- Informazio-galera saihesteko kode- zein datu-fitxategi guztiak ikaslearen ordenagailuan gordetzeaz gain *Google Drive* plataformara igoko dira, beti hodeian eskuragarri edukitzeko. Fitxategi horietako batzuen hirugarren kopia bat IXA taldearen zerbitzarietan ere gordeko da.

2.3 Plangintzarekiko Desbiderapenak

2.5 irudian 2.2.2 atalean definitutako lan-pakete bakoitzeko atazak burutzeko hasieran egindako ordu-aurreikuspenaren eta benetan behar izan den ordu-kopuruaren arteko aldea ikus daiteke. Proiektua bukatutzat emateko 545 ordu behar izan direla estimatzen da. Hasiera batean estimatzen ziren 300 ordu behar izateaz gain beste 245 ordu behar izan dira lan hau bukatutzat emateko, proiektuaren iraupena aurreikusitakoaren ia bikoitza bihurtuz. Atal honetan proiektuaren iraupena luzatzearen arrazoiak eztabaidatzen dira.

2.5 irudiari erreparatzen badiogu, argi dago ataza luzeena entrenamenduari eta sorkuntzari dagokiona izan dela, 200 ordutik gorako iraupena izan duela estimatzen delarik. Ataza hau ebazteko hainbeste denbora behar izatearen arrazoia zera da: egindako entrenamendu kopurua. Proiektu honetan lan egiten hasi ginean, helburua sistema bakarra entrenatzea zen, horretarako 1,5 milioi esaldi inguruko corpus bat erabiliz. Azkenean, zuzentzailearen kalitatea hobetu nahian, lau sistema entrenatu ditugu eta erabilitako corpus handienak 10 milioi esaldi ingurukoak izan dira. Datuen tamaina handitzeak ikasketa prozesurako behar den denbora ere asko luzatzea ekarri du.

Aurrekoaz gain, zuzentzailearen azpian dagoen sare neuronalaren inplementazioaren inguruan espero genuena baino gehiago lan egin behar izan dugu. Transformer arkitektura inplementatzeko gida batetik abiatuta kodea aztertu, ulertu eta datuen irakurketa gure beharretara moldatu ondoren, entrenamendu fasera pasa ginenean arazoak izan geni-

LAN-PAKETEA	DENBORA-AURREIKUSPENA (ordutan)	IRAUPEN ERREALA (ordutan)	DESBIDERAPENA (ordutan)
Kudeaketa	130	150	20
Plangintza	5	5	-
Jarraipena eta Kontrola	45	45	-
Memoria, aurkezpena eta posterra	80	100	20
Proiektuaren Garapena	170	395	225
Artearen egoeraren azterketa	10	10	-
Erroreak	20	20	-
Euskarazko erroreen azterketa	5	5	-
Erroreen sorkuntza	15	15	-
Corpusa	55	75	20
Prozesaketa	5	5	-
Corpus desberdinen sorkuntza	50	70	20
Zuzenketa Gramatikala	55	260	205
Transformer-aren inplementazioaren azterketa	20	30	10
Entrenamendua eta sorkuntza	35	230	195
Ebaluazioa	30	30	-
Ebaluazio-metriken azterketa	5	5	-
Garapenerako ebaluazioa	5	5	-
Ebaluazio finala eta emaitzak	20	20	-
GUZTIRA	300	545	245

2.5 Irudia: Lan-pakete bakoitzeko atazak ebazteko aurreikusitako denbora, atazak ebazteko benetan erabili den denbora eta bien arteko aldea, ordutan.

tuen. Sistema hainbat orduz entrenatzen egon ostean exekuzioa eten egiten zen memoria-arazoak zirela eta. Hasieran exekuzioak *Colab*-en egiten genituenez, plataformak doan eskaintzen dituen baliabideak agortzen ari ginela suposatu genuen, eta sistema IXA taldearen GPU-etan entrenatzera pasa ginen, baina ez ginen memoria-arazoa ebazteko gai izan. Konponbiderik aurkitzen ez genuela ikusita, Transformer-aren beste inplementazio bat erabiltzea erabaki genuen; memoriarekin zerikusia zuten arazoak mantentzen baziren datuen tamaina murriztu behar genuela ondorioztatu genuen. Azkenean, bigarren inplementazio honekin sistema entrenatzeko gai izan gara, baina kode berria aztertu, ulertu eta datu-irakurketarako moldatzeak denbora gehigarria suposatu du.

Corpus desberdinak sortzeko ere denbora asko behar izan dugu, ataza hori ebazteko 70 ordu inguru erabili ditugularik. Erroreak sortzeko bi metodo inplementatzeaz gain, sortu beharreko esaldi erroredun kopuru altua dela eta gertatu da hau. Gainera, esaldi erroredunak sortzeko bigarren metodoak exekuzio-denbora luzea behar du, esaldi bakoitzean errore gehiago txertatu behar direlako.

Aipatu berri ditugun desbiderapen guztiek ebaluazio fasera ekainaren bigarren astean heltzea eragin dute. Hori dela eta, hasiera batean asmoa Gradu Amaierako Lana uztailean defendatzea zen arren, defentsa irailera atzeratzea erabaki dugu. Horrela ebaluazio sakona egiteko eta memoria modu antolatu eta dotorean idazteko denbora ziurtatu dugu.

3. KAPITULUA

Aurrekariak

Sarreran (1. kapitulua) aipatu dugun bezala, itzulpen automatikoan erabiltzen diren teknikak zuzenketa gramatikalaren problema ebazteko erabili daitezke. Atal honetan lehenik eta behin itzulpen automatikoaren bilakaerari buruz hitz egingo dugu, ezagutzen diren aztarna zaharrenetatik abiatuz gaur egun gehien erabiltzen diren tekniketara arte. Ondoren, zuzenketa gramatikalean zentratuko gara, eremu honek historian zehar jasandako eboluzioa azalduz eta bereziki itzulpen automatikoa ebazteko teknikak zuzenketa gramatikalean nola aplikatu diren adieraziz. Hizkuntza-ereduei buruz ere hitz egingo dugu, itzulpen automatikoan zein zuzenketa gramatikalean erabiltzen baitira.

3.1 Itzulpen Automatikoa

Itzulpen automatikoaren arrastoa 9. mendera arte jarraitu daiteke, Al-Kindi izeneko eta kriptografiaren arloan lehen urratsak emateagatik ezaguna den filosofo, matematikari, fisikari eta musikariak kriptoanalisiaren inguruko hainbat teknika proposatu zituenean. Besteak beste, testu desberdinen arteko probabilitate estatistikoak alderatzeko eta hizkuntza desberdinetako patroiak eta ezaugarriak antzemateko gai izan zen [1]. 16 eta 17. mendeetan zehar, nazioarteko komunikazioa eta komunikazio zientifikoa hobetzeko asmoarekin, hizkuntza unibertsal eta logiko bat lortzeko proposamen asko egin ziren, batez ere hizkuntzen arteko lotura ebatziko zuten zenbakizko kodeekin erlazionatutako teknikak. Lehenengo "makina itzultzaileak" 20. mendean zehar agertu ziren [2]. Hala ere, ordenagailu elektronikoek hizkuntzen arteko itzulpena egiteko gaitasuna izatearen ideia ez zen

1946. urtera arte planteatu. Andrew D. Booth eta Warren Weaver izan ziren itzulpen automatikoari buruzko elkarrizketa hasi zutenak, ordenagailuak zenbakietan oinarritzen ez diren arloetan erabiltzeko nahiak bultzatuta. Weaver-ek itzulpena problema kriptografiko bat bezala lantzea iradoki zuen [3]. Hurrengo urteetan zehar itzulpen automatikoaren inguruko ikerketak aurrera jarraitu zuen. Itzulpen automatikoko sistema baten lehenengo erakustaldi publikoa 1954ko urtarrilaren 7an burutu zen New York-en. Bertan aurkeztutako sistemak 250 hitz besterik ez zituen eta aldez aurretik hautatutako errusiarrez idatzitako 49 esaldi ingelesera itzultzeko gai zen. Erakustaldi honek irismen handia izan zuen eta itzulpen automatikoari buruzko interesa mundu osoan zehar pizten lagundu zuen.

1966. urtean ALPAC txostena argitaratu zen. Txosten hau Estatu Batuetako gobernuaren agindupean idatzi zen eta bertan itzulpen automatikoak giza-itzulpenak baino emaitza txarragoak ematen zituela ondorioztatzen zen, garestiagoa eta motelagoa izateaz gain. Argitalpen honek kolpe gogorra suposatu zuen itzulpen automatikoaren komunitatean eta arloan lan egiten ari ziren talde gehienek ikerketa bertan behera uztea eragin zuen [4]. Orduz geroztik itzulpen automatikoaren arloan lan egiten zuten ikerlarien helburuak errealistagoak bihurtu ziren; jada asmoa ez zen estilo aldetik ezin hobeak ziren itzulpenak lortzea, ulergarriak eta fidagarriak ziren itzulpenak lortzea baizik.

3.1.1 Erregeletan oinarritutako itzulpen automatikoa

Erregeletan oinarritutako itzulpen automatikoa (ingelesez *Rule-based Machine Translation* edo RBMT) 70. hamarkadan planteatu zen lehenengoz. Mota honetako sistemek jatorri- eta helburu-hizkuntzen informazio linguistikoa dute oinarri bezala. Xedea jatorrieta helburu-esaldien egitura lotzea da, beti ere esanahia mantenduz. Horretarako, jatorrieta helburu-hizkuntzen hiztegi bat eta hizkuntzen semantika, morfologia eta sintaxia biltzen duten erregelak behar dira. Erregelen kasuan, zehazki, 3 sorta desberdin dira beharrezkoak: jatorrizko hizkuntzako esaldien egitura biltzen dutenak, helburu hizkuntzako esaldien egitura biltzen dutenak eta bi hizkuntzen egiturak lotzen dituztenak.

RBMT metodologiaren barnean 3 sistema mota desberdintzen dira: Sistema Zuzenak, Transferentzia-Sistemak eta Sistema Interlinguistikoak.

Sistema Zuzena

Izenak dioen bezala, estrategia zuzenena da: esaldia hitzez hitz itzultzen da hiztegi bat erabiliz eta kasu batzuetan morfologia zuzenketa txikiak eginez. Esaldiaren esanahia eta

hitzen arteko korrelazioa ez direnez kontuan hartzen, esaldien itzulpen finala kaskarra da, baina erabilgarria da zerrenden itzulpenerako, katalogoen kasuan adibidez.

Transferentzia-Sistema

Transferentzian oinarritutako sistemek hiru fase jarraitzen dituzte: analisia, transferentzia eta sorkuntza. Analisian jatorrizko hizkuntzan dagoen esaldiaren analisi linguistikoa egiten da, besteak beste morfologia, sintaxia, semantika eta kategoria gramatikalei buruzko informazio lortuz, hizkuntzaren barne-errepresentazio bat sortzeko. Transferentzia atalean aurreko fasean lortutako errepresentazioaren helburu-hizkuntzako errepresentazio baliokidea lortzen da, horretarako bi hizkuntzen egitura lotzen duten erregelak erabiliz. Sorkuntza-fasean lortutako azken errepresentazioari aldaketak egiten zaizkio helburu-hizkuntzaren arauen arabera (itzulitako hitzen generoaren eta komunztaduraren zuzenketa, adibidez), horrela itzulpen finala sortuz [5] [6].

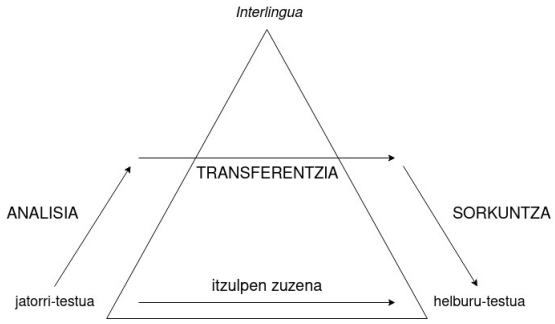
Sistema Interlinguistikoa

Estrategia interlinguistikoan bi fase bereiz daitezke: Analisia eta Sorkuntza. Lehenengo fasean jatorri-hizkuntzako esaldiaren analisia egiten da, bere eduki semantikoa ateratzen da eta *Interlingua* moduan errepresentatzen da. *Interlingua* deritzo jatorri- zein helburu-hizkuntzekiko dependentziarik ez duen hizkuntza berri bati eta jatorri-hizkuntzaren erdibideko barne-errepresentazio bezala erabiltzen da. Bigarren fasean *Interlingua* erabiltzen da helburu-hizkuntzako esaldiak sortzeko. Sistema Interlinguistikoen abantaila nagusia hurrengoa da: erdibideko errepresentazioa beste hizkuntzekiko independentea denez, jatorri-hizkuntza jakin bateko analisi-programa nahi beste helburu-hizkuntzarako erabili daiteke eta, era berean, helburu-hizkuntza jakin bateko sorkuntza-programa edozein jatorri-hizkuntzetako itzulpena lortzeko erabili daiteke [5].

Hiru sistema hauek sortzen dituzten barne-errepresentazioen alderaketa modu argian laburbildu daiteke Bernard Vauquois-en [7] piramideari esker (3.1 irudia).

3.1.2 Corpusean oinarritutako itzulpen automatikoa

Corpusean oinarritutako itzulpen automatikoan ez da beharrezkoa hizkuntzei buruzko informazio linguistikorik ezagutzea. Horren ordez, jatorri- eta helburu-hizkuntzen corpusa (esaldi multzoa) da ezinbestekoa. Bi hizkuntzetako datu-multzo paraleloak erabiltzen dituen metodo hau 1984. urtean erabili zen lehenengoz [8], azken hamarkadetan izandako aurrerakuntza teknologiko eta hizkuntza desberdinen arteko itzulpen kopuruaren areagotzeari esker guztiz nagusitu arte. Duela gutxi arte corpusarekin lan egiten duten itzulpen-



3.1 Irudia: Vauquois-en piramidea.

teknikak bi metodotan banatu izan dira: adibideetan oinarritutako metodoak eta estatistikan oinarritutako metodoak [9]. Hala ere, azken urteetan egindako aurreraketen ondorioz, hirugarren metodo bat ere aipatu beharra dago, sare neuronalak erabiltzen dituena, alegia.

Adibideetan oinarritutako itzulpen automatikoa

Adibideetan oinarritutako itzulpenaren ideiak gizakion itzulpenak egiteko moduan du funtsa: esaldi bat itzultzeko, analisi linguistiko bat egin ordez, esaldia zati txikiagotan banatzen dugu, zati horiek itzultzen ditugu eta gero analogia bidez elkartzen ditugu zati guztiak zentzuzko esaldi bat osatuz. Kontzeptu hau makinei aplikatzen badiegu, adibide-banku erraldoi bat eduki beharko dugu jatorri-hizkuntzako esaldiekin eta bakoitzaren itzulpenarekin. Esaldi bat itzuli nahi denean sistemak esaldia zatituko du eta adibide guztien artean bilatuko du zati bakoitzari semantikoki gehien hurbiltzen zaiona, irteera bezala zati guztien itzulpenak emanez.

Adibidez, suposa dezagun gaztelaniako "*Ha comprado un coche*" esaldia euskarara itzuli nahi dugula eta gure corpuseko adibideen artean 3.1 taulan ikus daitezkeen itzulpenak aurki daitezkeela.

Sistemak itzuli nahi den esaldia zatituko du ("*Ha comprado*" eta "*un coche*") eta zati bakoitzari dagokion itzulpen zatia lortuko du. Horrela, "*Ha comprado*" eta "erosi du" bateratuz eta "*un coche*" eta "Kotxe bat" bateratuz, "Kotxe bat erosi du" esaldia lortuko da (batzuetan post-prozesaketa egiten da komunztadura zuzentzeko).

Jatorrizko hizkuntza (gaztelania)	**Itzulpena (euskara)**
"Ha comprado una casa"	"Etxe bat erosi du"
"Tiene un coche"	"Kotxe bat dauka"

3.1 Taula: Adibideetan oinarritutako itzulpen automatikoan erabiltzen den corpusaren adibidea.

Itzulpen automatiko estatistikoa

Estatistikan oinarritutako itzulpen automatikoa *International Business Machines (IBM)* enpresako ikerlariek proposatu zuten lehenengoz 1990. urtean. Geroztik, azken urteotan metodo estatistikoek izandako garapenari esker, itzulpen automatiko estatistikoak erregeletan oinarritutako metodoak gailendu ditu.

Itzulpen automatiko estatistikoaren ideia nagusia hurrengoa da: helburu-hizkuntzako edozein esaldik jatorrizko esaldiaren itzulpena izateko probabilitate jakin bat du. Horrela bada, helburua probabilitate altueneko helburu-hizkuntzako esaldia aurkitzea da.

Mota honetako itzulpenean bi hizkuntzatako corpus paralelotik modelo estatistiko bat lortzen da, gero itzulpena egiteko erabiliko dena. Modelo estatistiko hori itzulpen-modeloak eta hizkuntza-ereduak[1] osatzen dute. Itzulpen-modeloa hizkuntzaren zati bat beste hizkuntza batera itzultzearen posibilitatea (baldintzapeko probabilitatea) kalkulatzeko erabiltzen da. Hizkuntza-eredua hitz, hitz-multzo edo esaldi bat helburu-hizkuntzaren parte izatearen probabilitatea neurtzeko erabiltzen da. Modelo estatistikoa corpusetik lortzen denez, corpusaren kalitateak zehazten du sistemaren eraginkortasuna. Normalean 2 milioi hitzetik gorako corpusak behar dira zentzuzko itzulpenak lortzen hasteko.

Itzulpen automatiko neuronala

Eredu estatistikoetan ez bezala, itzulpen automatiko neuronalean sare neuronalen bidez burutzen da ikasketa eta sarrerako esaldi bat emanda itzulpena lortzen da irteera bezala. Horretarako kodetzaile-deskodetzaile eredua erabiltzen da, non kodetzaileak sarrerako esaldia irakurri eta bektore-errepresentazio moduan kodetzen duen eta deskodetzaileak bektore-errepresentazio horretatik irteerako itzulpena lortzen duen.

3.2 Zuzenketa Gramatikala

Itzulpen automatikoaren kasuan gertatu zen antzera, zuzenketa gramatikal automatikoaren arazoa ebazteko lehenengo proposamenek erregela bidezko metodoak erabiltzea planteatzen zuten. Askotariko metodoak ziren hauek, patroien identifikazio eta *string*-en ordezkapena bezalako teknika sinpleetatik hasita, analisi sintaktikoa eta eskuz deskribatutako erregela gramatikalak erabiltzen zituzten metodoetara arte. Gaur egun oraindik erabiltzen dira erregeletan oinarritutako teknikak, inplementatzeko errazak direlako eta kasu batzuetan, hitzen ordenarekin zerikusia duten egoeretan adibidez, oso eraginkorrak suertatzen

[1] Hizkuntza-ereduen funtzionamenduari buruzko azalpen xehatua eskaintzen da 3.3 atalean.

direlako. Hala ere, existitu daitekeen errore bakoitzarentzat erregela bat definitzea ez denez bideragarria, kasu zehatzetarako bakarrik erabiltzen dira eta gainontzeko erroreen zuzenketari beste ikuspuntu batetik heltzen zaio.

3.2.1 Datuetan oinarritutako zuzenketa gramatikala

90. hamarkadan, anotatutako datuen kopurua nabarmenki handitzearekin batera, datumultzo handiei Ikasketa Automatikoko (ingelesez *Machine Learning*) teknikak aplikatzen zizkieten metodoak nagusitu ziren eta errore mota desberdinak ebazteko sailkatzaileak eraiki ziren [10] [11]. Bereziki artikuluekin eta preposizioekin lotutako erroreen kasuak landu ziren, alde batetik errore hauek ohikoenak direlako hizkuntza bat ikasten ari diren pertsonetan eta, bestetik, *Machine Learning* teknikekin erregelak erabiliz baino errazago antzematen eta zuzentzen direlako. Errore hauek zuzentzeko zuzenketarako hautagai guztien sorta bat definitzen da (adibidez existitzen diren artikulu guztien zerrenda), entrenamendurako adibideak ezaugarri linguistikoen bektore moduan adierazten dira, hala nola erlazio gramatikalak edo hitz auzokideak, eta ezaugarri hauetan oinarrituta sailkatzaileak entrenatzen dira. Erroreak hitz originala sailkatzaileak proposatutako hautagaiarekin alderatuz zuzentzen dira. Ezaugarri erabilgarrienak hitz-klasearen araberakoak direnez, beharrezkoa izaten da errore mota bakoitzerako sailkatzaile desberdinak entrenatzea.

Mota honetako zuzentzaileen ahulezia testuinguru lokala bakarrik kontuan hartzen dutela eta erroreak modu independentean tratatzen dituztela da, esaldi bakoitzean errore bakarra dagoela suposatuz. Ohiko konponbide bat sailkatzaile anitz eraikitzea eta *pipeline* sistema batean bata bestearen atzean erabiltzea da. Normalean sailkatzaileak eta erregelak elkartzen dira ataza honetarako [12] eta, sailkatzaileen ordenaren garrantzian erreparatzeaz gain, aurre-prozesaketa eta post-prozesaketa burutu behar dira. Hala ere, metodo honek ez ditu elkarri eragiten dioten erroreak ebazten.

Elkarri eragiten dioten erroreen arazoa konpontzeko hainbat teknika proposatu dira, *Beam Search* deskodetzaile bat erabiliz esaldi-mailako hautagaiak iteratiboki sortzea eta hauek hizkuntza-eredu bat erabiliz puntuatzea [13] edo inferentzia bateratua[2] erabiliz banakako sailkatzaileek sortutako funsgabetasunari irtenbidea aurkitzea [14], adibidez. Hala ere, hainbat errore zuzentzeko metodo zabalduena n-grama hizkuntza-ereduak erabiltzea da [15], kasu batzuetan sailkatzaileekin eta erregelekin konbinatuz [12] maiztasun gutxiko

[2]Modelo estatistikoen arloan inferentzia bateratua edo *joint inference* deritzo morfologia, semantika, sintaxia, pragmatika eta testuinguruari buruzko gainontzeko informazioa aldi berean aintzakotzat hartzeari, anbiguotasuna ebazteko asmoarekin.

hitz-konbinazioak eta hitz-konbinazio okerrak desberdintzeak suposatzen duen erronkari aurre egiteko.

3.2.2 Itzulpen automatikoa eta zuzenketa gramatikala

Errore gramatikalak zuzentzeko itzulpen automatikoaren arloan erabiltzen diren ereduak arrakastatsuak suertatzen dira errore multzo zabalagoak ebazteko gai direlako. Itzulpenean oinarritutako zuzentzaileek adibide paraleloetatik (testu erroreduna eta testu zuzena) patroiak eta erlazioak ikasten dituzte eta hauek erabiltzen dituzte testu erroreduneko ahalik eta errore gehien zuzentzeko.

2006. urtean erabili ziren lehenengoz itzulpen automatikoko teknikak erroreen zuzenketarako [16]. Zehazki itzulpen automatiko estatistikoan erabiliak diren tekniken bidez izen zenbakaitzen (ingelesez *mass nouns*) erroreak zuzendu ziren. Eskuragarri zegoen izen zenbakaitzen erabilera okerra erakusten duen esaldi kopuru murriztua dela-eta, esaldi zuzenetan erroreak txertatu ziren erregelak erabiliz, horrela entrenamendurako datuen tamaina handituz. Erroreen %61,81 zuzentzea lortu zen, corpus artifiziala erabiltzea baliagarria dela eta emaitza arrakastatsuak lortu daitezkeela frogatuz. Datu-multzoaren tamaina artifizialki handitzea problema gehienetan erabiltzen den praktika bat da, geroz eta datu gehiago izan, orduan eta doitasun altuagoa lortzen dela frogatu baita [17].

2011. urtean proposatu zen ingelesezko errore gramatikalak zuzentzea helburu zuen lehenengo lehiaketa, *Helping Our Own (HOO)* izenekoa [18]. Harrezkero ataza hau ebazteko lehiaketa gehiago proposatu dira, *CoNLL-2013 Shared Task* [19] eta *CoNLL-2014 Shared Task* [20] aipagarriak direlarik. Hizkuntzaren Ikasketa Konputazionalari (ingelesez *Computational Natural Language Learning*) buruzko konferentzietan aurkeztutako bi ataza hauetan partaide asko itzulpen automatikoko teknikez baliatu ziren, modelo estatistikoak uneko artearen egoerako emaitzak lortzeko gai zirela frogatuz.

Itzultzaile automatiko estatistiko eta hizkuntza-ereduez gain, lehiaketa hauetan sarritan proposatutako teknikak itzulpen automatiko neuronalarekin erlazionatutakoak dira. Gure proiekturako zein metodo erabili hautatzeko, azken urteotan zuzenketa gramatikalaren arloan aurkeztu diren hainbat lan aztertu ditugu. Lan hauetan akats gramatikalak zuzentzeko metodo desberdin asko proposatzen dira: zuzenketa estatistikoa eta neuronala konbinatzea [21], sare neuronal arkitekturak hizkuntza-ereduekin batera erabiltzea [22] eta sare neuronal arkitektura desberdinak beraien artean kateatzea [23], besteak beste. Proposamen hauek aztertu ondoren, zuzenketa gramatikal neuronala lan guztietan erabilia izan dela

antzeman dugu, bereziki 2019ko *BEA Shared Task* lehiaketako [23] parte-hartzaileen artean, non aurreko urteetako lehiaketetan ez bezala, talde guztiek neurona-sareak erabili zituzten. Sare neuronalak erabiliz artearen egoerako emaitzak lortu daitezkeela erreparatu dugunez, bide horretatik jarraitzea erabaki dugu. 4. kapituluan azalduko dugu zehazki itzulpen automatikoa eta zuzenketa gramatikala ebazteko proposatu diren sare neuronalen arkitekturaren bilakaera.

3.3 Hizkuntza-ereduak

Itzulpenean eta zuzenketan lortzen diren emaitzak helburu-hizkuntzaren parte ote diren jakitea garrantzitsua da, hau da, lortzen diren esaldiak helburu-hizkuntzarekiko nolabaiteko antzekotasuna dutela ziurtatzea. Horretarako hizkuntza-ereduak (HE), ingelesez *language models* (LM), erabiltzen dira. Hizkuntza-ereduek hitz-sekuentzia bat emanda banaketa probabilistiko bat kalkulatzen dute. Hizkuntza-eredu bat *H* hizkuntza jakin bateko testu askorekin entrenatzen badugu eta gero esaldi bati aplikatzen badiogu, esaldi horrek H hizkuntzakoa izateko duen probabilitatea ezagutu dezakegu.

"Hizkuntza-eredu" terminoa 80. hamarkadako hasieran ahotsaren ezagutzaren arloan erabiltzeko garatutako testu-sorkuntza eredu probabilistikoetan erabili zen lehenengoz [24]. Azken 40 urteetan hizkuntza-ereduak inplementatzeko modu desberdinak planteatu dira, gaur egun bi kategoria nagusitan banatu daitezkeelarik: kontaketan oinarritutako HE eta espazio jarraituko HE.

3.3.1 Kontaketan oinarritutako hizkuntza-ereduak

Kontaketan oinarritutako hizkuntza-ereduek (ingelesez *Count-based language models*) formulazio estatistikoa erabiltzen dute hitz-sekuentzia baten gaineko probabilitate-banaketa bateratua (ingelesez *joint probability distribution*) kalkulatzeko.

Demagun s sekuentzia bat emanda sekuentziako hurrengo hitza w hitza izatearen P($w|s$) probabilitatea kalkulatu nahi dugula. Probabilitate hau estimatzeko modu bat maiztasun kontaketa erlatiboa erabiltzea da, hau da, corpus handi batean (Google-eko bilaketak, adibidez) s sekuentziaren agerpen kopurua zenbatzea eta w hitza s sekuentziaren ondoren zenbatetan agertzen den zenbatzea. Ondoren, 3.1 ekuazioan adierazten den bezala kalkulatuko litzateke probabilitatea, non kontaketak adierazteko C erabiltzen den eta (sw) adierazpenak s sekuentziaren ondoren w hitza dagoela adierazten duen.

$$P(w|s) = \frac{C(sw)}{C(s)} \tag{3.1}$$

Metodo honek fidagarria dirudien arren, *"It was the best of times, the worst was over"* bezalako esaldi sinple eta arrunt batek ez du agerpenik Google-eko bilaketetan eta, beraz, ezin dezakegu kalkulu hau erabili sekuentzia guztientzako.

Proposatutako beste aukera bat *s* hitz-sekuentzia baten probabilitate bateratua kalkulatzea da, horretarako *s* sekuentziaren hitz kopurua adina hitz duten sekuentzia guztietatik *s*-rekin zenbat datozen bat neurtuz. *s*-ren agerpen kopurua *s* sekuentziak bezain beste hitz dituzten sekuentzia kopuruarekin zatitu beharko genuke, baina hori estimazio handiegia egitea izango litzateke.

Aukera desberdin bat katearen erregela erabiltzea da. Demagun $P(w_1, w_2, w_3, ..., w_n)$ adierazpenak *n* hitzez osatutako sekuentzia baten probabilitatea adierazten duela. Katearen erregela aplikatuz, 3.2 ekuazioan agertzen den bezala kalkulatu dezakegu probabilitate hori. Hala ere, ez daukagu modurik $P(w_k|w_1^{k-1})$ zehazki kalkulatzeko. Arazo honi aurre egiteko n-grama eredua erabiltzea planteatzen da.

$$\begin{aligned} P(w_1, w_2, w_3, ..., w_n) &= P(w_1) \times P(w_2|w_1) \times P(w_3|w_1, w_2) \\ &\times, ..., \times P(w_n|w_1, w_2, w_3, ..., w_{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned} \tag{3.2}$$

N hitzez osatutako sekuentzia bati deritzogu n-grama. Eredu honen motibazioa hurrengoa da: sekuentziako aurreko hitz guztiak ezagututa hurrengo hitzaren probabilitatea kalkulatu ordez, sekuentzia osoaren hurbilpena egin dezakegu azken hitzak soilik erabiliz. Hitz baten probabilitatea aurreko hitzekiko menpekoa besterik ez dela dioen teoriari Markov-en hipotesia deritzo. Iraganean gehiegi atzera joan gabe etorkizunaren probabilitatea iragarri daitekeela da Markov-en eredu probabilistikoen funtsa.

Markov-en lehen ordena, beraz, sekuentziaren aurreko hitza soilik kontuan hartzeari deritzo. Bigarren ordenan aurreko bi hitzak hartuko dira kontuan, hirugarren ordenan aurreko hiru hitzak, eta abar. Era berean, bigrama ereduan bi hitzekin egiten da hurbilpena, uneko hitza eta sekuentziako aurreko hitza erabiliz, 3.3 ekuazioan ikusten den bezala. Beraz, Markov-en lehen ordena (3.4 ekuazioa[3]) eta bigrama eredua bat datoz. Bigrama eredua-

[3] $P(STOP|w_n)$: azken hitza ezagututa sekuentziaren amaiera izatearen probabilitatea.

ren oinarria jarraituz, trigrama ereduan sekuentziako aurreko bi hitzak aztertzen dira eta, orokortuz, n-grama ereduan sekuentziaren aurreko *n-1* hitzetan erreparatzen da.

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1}) \tag{3.3}$$

$$\prod_i^n P(w_i|w_{i-1}) \times P(STOP|w_n) \tag{3.4}$$

Bigrama eredua adibide batekin ulertzeko, *"The sky is blue"* esaldiaren probabilitatearen kalkuluaren adierazpena ikus daiteke 3.5 ekuazioan[4].

$$P(The|START) \times P(sky|The) \times P(is|sky) \times P(blue|is) \times P(STOP|blue) \tag{3.5}$$

n-gramen probabilitatea estimatzeko egiantza handieneko estimazioa (ingelesez *maximum likelihood estimation* edo MLE) erabiltzen da, hau da, n-gramaren agerpen kopurua normalizatzen da. Adibidez, bigramen kasuan (3.6 ekuazioa), w_n hitzaren probabilitatea kalkulatu nahi badugu aurreko w_{n-1} hitza emanda, $w_{n-1}w_n$ bigramaren agerpen kopurua zenbatuko dugu eta lehenengo hitz bezala w_{n-1} hitza duten bigrama guztien kopuruarekin zatituz normalizatuko dugu[5].

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{3.6}$$

Kontaketetan oinarritutako eredu hauek arazo bat dute: posible da entrenamenduko datuetan n-grama baten agerpen kopurua 0 izatea baina bestelako datuetan, probarako esaldietan adibidez, agertzea. Kasu hori gertatzekotan, n-grama horren probabilitatea 0 izango litzateke eta esaldiaren probabilitate totala kalkulatzeko n-gramen probabilitateen biderketa erabiltzen dugunez (3.4 ekuazioa), esaldiaren probabilitatea ere 0 izango litzateke. Horrelako kasuetan leunketa teknikak erabiltzen dira, elementu bakoitzari probabilitate masa txiki bat gehitzea adibidez [25].

Leunketak laguntzen duen arren, dimentsionalitatearen arazoa nabarmena da n-grama hizkuntza-ereduetan, hitzekin osatu daitezkeen sekuentzia desberdinen kopurua erraldoia baita. Adibidez, 10 hitzeko sekuentziekin lan egin nahi badugu 100.000 hitzeko hiztegia

[4] P(*The*|START): sekuentziaren hasieran gaudela jakinda lehenengo hitza *"The"* izatearen probabilitatea.

[5] Hitz jakin batekin hasten diren bigrama guztien kopurua hitz jakin horren unigrama kopurua da.

duen corpus batean, 10^{50} sekuentzia posible existitzen dira. Gainera, n-gramak zenbatzean *string* zehatzak bakarrik kontatzen direnez, ez da inolako informazio linguistikorik (*"cat"* eta *"kitten"* hitzen antzekotasuna, adibidez) kontuan hartzen. Ereduek informazio linguistikoa ikasteko asmoz eta dimentsionalitatearen arazoa saihesteko orokortze hobea lortzeko helburuarekin proposatzen dira espazio jarraituko hizkuntza-ereduak.

3.3.2 Espazio jarraituko hizkuntza-ereduak

Espazio jarraituko hizkuntza-ereduek (ingelesez *Continuous-space language models*) sare neuronalek errepresentazio jarraituak (*embedding*-ak[6]) ikasteko duten gaitasuna erabiltzen dute dimentsionalitatearen arazoari aurre egiteko. Sare neuronalak erabiliz hitzak sareko pisuen konbinazio ez-lineal bezala adierazi daitezke. Eredu hauek hizkuntza-eredu neuronal bezala ere dira ezagunak.

Bi talde nagusitan banatzen dira hizkuntza-eredu neuronalak. Alde batetik, aurreranzko barreiatze-sareetan (ingelesez *feed forward networks*) oinarritutako hizkuntza-ereduak aurki daitezke, entrenamenduan n-grama baten agerpenik ez egoteak suposatzen duen arazoa ebazteko proposatuak. Bestetik, sare neuronal errekurrenteetan[7] oinarritutako hizkuntza-ereduak, n-gramak erabiliz testuingurua aurreko *n* hitzetara soilik mugatzeak dakarren testuinguru mugatuaren arazoa konpontzeko helburuarekin.

Adibide bat 2003. urtean aurkeztutako hizkuntza-eredu neuronal probabilistikoa da, aurreko *n-1* hitzak ezagututa eta 3 geruzako aurreranzko barreiatze-sare bat erabiliz hurrengo hitzaren probabilitate banaketa ikasten duena [26]. Lan honetan, hiztegiko hitz bakoitzari *embedding* bat egokitzen zaio eta hurrengo hitzaren probabilitatea testuinguruko hitzen *embedding*-en mapaketa egiten duen f funtzio baten bidez kalkulatzen da. Eredua gai da aldi berean hitzen *embedding*-ak eta f funtzioaren parametroak ikasteko.

Espazio jarraituko hizkuntza-ereduen desberdintasun nagusia erabiltzen duten testuinguruaren luzera da. Aurreranzko barreiatze-sareetan oinarritutako hizkuntza-ereduetan testuinguruaren tamaina finkatu beharra dago; sare errekurrenteak erabiltzen dituzten hizkuntza-ereduetan, ordea, ez da testuingurua mugatu behar, konexio errekurrenteak erabiliz informazioa sarean barrena zikloak eginez mantendu daitekeelako eta neurona bakoitzak memoria bat bezala jokatuko duelako [27].

[6] *Embedding* deitzen zaie hitzen esanahi linguistikoari buruzko ahalik eta informazio gehien ematen duten zenbakizko errepresentazioei (normalean bektoreak).

[7] Sare neuronal errekurrenteei buruzko azalpen sakonagoa aurki daiteke 4.1 atalean.

4. KAPITULUA

Itzulpenerako eta Zuzenketarako Arkitektura Neuronalak

Zuzenketa gramatikal neuronalean, aukeratutako sare neuronal arkitektura esaldi erroredun eta zuzenduen bikoteekin entrenatu ondoren, sarrerako esaldi erroredun bat emanda esaldi horren zuzenketa irteera bezala lortzea da gure helburua. Horretarako, 3.1.2 atalean aipatu den bezala, kodetzaile-deskodetzaile eredua erabiltzen da.

Kodetzaile-deskodetzaile egitura sekuentziekin lan egitea eskatzen duen edozein problemarako da baliagarria. Sekuentziekin lan egitea deritzogu sarrera edo/eta irteera bezala sekuentzia bat onartzen duen edozein ataza ebazteari. Aukera desberdinak aurki ditzakegu: (1) sarrera-sekuentzia bat izatea eta irteera-balio bakarra izatea, (2) sarrera-balio bakarra izatea eta irteera-sekuentzia bat izatea eta (3) sarrera- zein irteera-sekuentziak izatea. (1) kasuaren adibide bat eguraldi-iragarpena da, non sarrerako sekuentzia aurreko egun kopuru jakin batean izandako eguraldia den eta irteerako balioa hurrengo egunean egingo duen eguraldiaren iragarpena den. Estiloaren arabera musika automatikoki sortzea izan daiteke (2) kasua, sarrera adibidez "rock musika" delarik eta irteera sortutako noten segida. (3) kasua itzulpen automatikoan, ahotsaren ezagupenean eta testu-ahots eraldaketan ematen da, besteak beste, eta gure lanaren xedea ere bada.

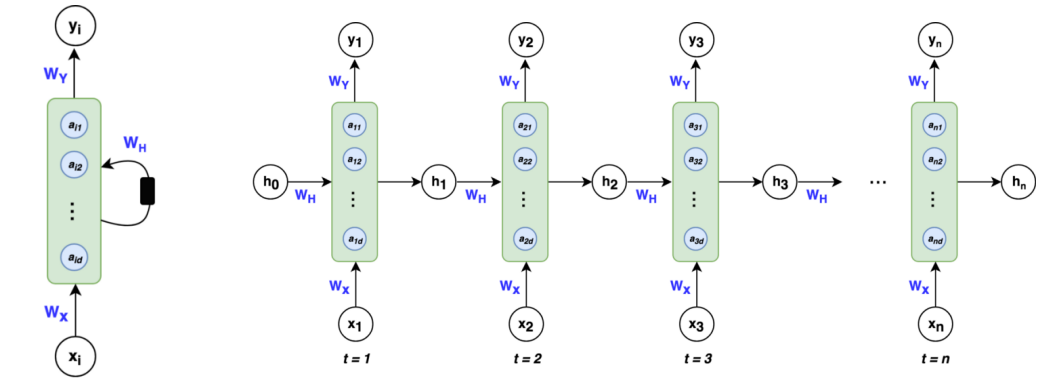
Azken kasu honetan aipatutako atazak ebazteko sekuentziatik-sekuentziara (ingelesez *sequence-to-sequence* edo *Seq2Seq*) modeloak erabiltzen dira. Jatorrizko esanahia galdu gabe sekuentziak eraldatzeko gai izateko, modelo hauek nolabaiteko memoria bat behar dute. Demagun hurrengo testua daukagula: *"One Direction" are an English-Irish pop boy band. The band was formed on The X Factor in 2010.* Testu hau beste edozein hizkuntza-

ra itzuli nahiko bagenu garrantzitsua izango litzateke bigarren esaldiko "*the band*" hitzek lehenengo esaldiko "*One Direction*" taldeari erreferentzia egiten diotela jakitea. Orokorrean, sekuentziekin lan egitean, modeloek esaldien arteko dependentziak eta konexioak ezagutu behar dituzte. Sare Neuronal Errekurrenteak erabiltzea da ohikoa arazo honi aurre egiteko.

4.1 Sare Neuronal Errekurrenteak

Sare Neuronal Errekurrenteak, ingelesez *Recurrent Neural Networks* edo RNN, begiztaz osatuta daude, denboran zehar informazioa gorde ahal izateko. 4.1 irudian ikus dezakegu RNN motako sareen arkitektura. Sare hauek tolestuta (4.1a azpi-irudia) zein hedatuta (4.1b azpi-irudia) irudikatu daitezke. Adierazpen hedatua denbora-pausu bakoitzeko egoera irudikatzeko erabiltzen da. Irudietako biribil urdinak neurona, unitate edo nodo ezkutuak (*hidden neurons/units/nodes*) dira (unitate ezkutu kopurua sarearen arkitektura diseinatzean erabakitzen da) eta bloke berdeak, egoera ezkutuak (*hidden states*) izenekoak, neuronen gain eragiten duten aktibazio-funtzioak dira. Sare hedatuari erreparatzen badiogu, t momentuan, x sarrera kontuan hartzeaz gain, *t-1* momentuko informazioa ere erabiltzen da aurreko egoera ezkutuko h bektorea sarrera bezala hartuz. h bektore horretan *t-1* egoera ezkutuko irteeraren informazioa dago gordeta. Garrantzitsua da guztira d neurona daudela gogoratzea, 4.1a azpi-irudian ikus daitezkeenak hain zuzen ere. Honek zera esan nahi du: sare hedatuko t bakoitzean irudikatzen den $a_{t,1}$ neurona eta *t+1*-eko $a_{t+1,1}$ neurona nodo bera direla (berdina betetzen da neurona eta t guztientzako). Horrela bada, h bektore eta x sarrera bakoitza beti sartzen dira neurona berberetara, baina iterazio bakoitzean x sarrerako sekuentziaren hurrengo elementua eta h aurreko iterazioan gordetako informazioaren errepresentazioa direlarik. Era berean, RNN-aren pisuak gordetzen dituzten W_x, W_y eta W_h matrizeak ere berberak dira pausu bakoitzean eta sare osoan zehar partekatzen dira, iterazio bakoitzean pisuak eguneratuz.

4.2 irudian ikus daiteke RNN-aren erabilera itzulpen automatikoaren arloan. Kodeketa faseko pausu bakoitzean x sarrera euskarazko "ni ikaslea naiz" esaldiko hitz bat izango da eta h bektorea aurretik ikusitako hitz guztien informazio kodetua izango da; t = 2 pausuan, adibidez, x_2 "ikaslea" hitza da, h_1 bektorea "ni" hitzaren errepresentazio kodetua da eta t = 3 pausura sartuko den h_2 bektorea "ni" eta "ikaslea" hitzen informazio kodetua da. Deskodetzailearen fasean, euskarazko esaldi osoaren informazio kodetua jasoko da h_3 bektorean eta informazio horretatik abiatuta pausu bakoitzean ingelesezko "*I am a student*" esaldiko hitz bakoitza lortuko da irteera bezala.



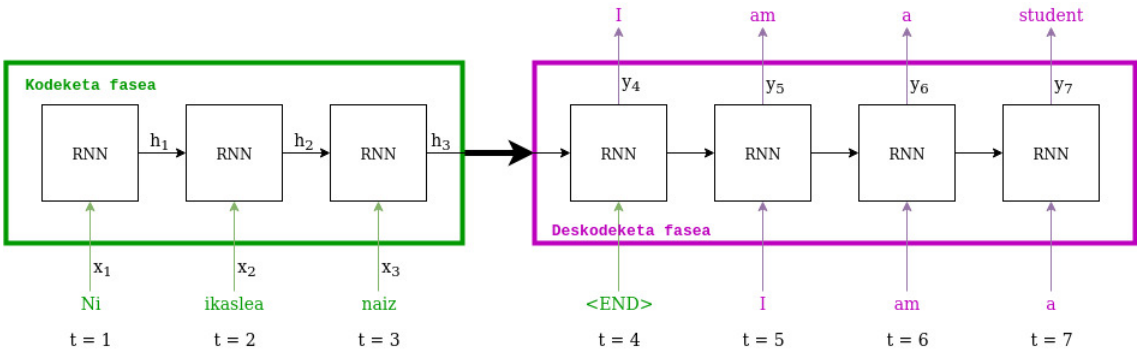
(a) RNN tolestua.

(b) RNN hedatua.

4.1 Irudia: Sare Neuronal Errekurrenteen arkitektura.

Iturria: https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f

Sare neuronal errekurrenteek esaldietako edozein dependentzia ikasteko gaitasuna dutela dirudien arren, dependentzia luzeekin arazoak sortzen dira. Demagun RNN bat erabili nahi dugula esaldi bateko azken hitza zein den asmatzeko. Esaldia *The sun is in the...* bada, ez da zaila azken hitza "*sky*" dela asmatzea, esaldiko hitz esanguratsuak ("*the sun*") eta asmatu beharreko hitza ("*sky*") hurbil daudelako eta, beraz, sare errekurrenteak hurbil edukiko du hitz garrantzitsuen informazioa. Esaldi luzeagoetan dependentziak agertzen direnean, ordea, ez da hain erraza. Adibidez, *I was born in Italy and lived there until I was 15. I speak fluent...* testuaren kasuan, argi dago lortu beharreko hitza hizkuntza bati dagokiola, baina zein hizkuntza den adierazten digun hitza, kasu honetan "*Italy*", urrunegi dago eta asmatu beharreko posiziora heltzen denerako sareak dagoeneko ahaztu du herrialdearen informazioa. Geroz eta luzeagoa izan saretik igarotzen den sekuentzia, orduan eta errazagoa izango da informazioa sekuentzian barrena galtzea. Arazo honi konponbidea topatzeko *Long-Short Term Memory* sareak erabiltzen dira.

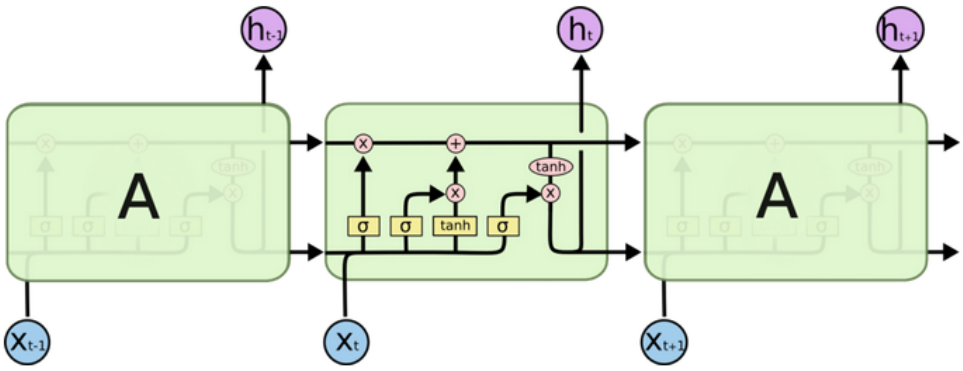


4.2 Irudia: Sare neuronal errekurrentearen erabilera itzulpen automatikoan.

4.2 Long-Short Term Memory

Long-Short Term Memory edo LSTM arkitektura Hochreiter and Schmidhuber-ek aurkeztu zuten 1997. urtean [28]. Sare hauek garrantzitsua den informazioa gogoratzeko eta hain garrantzitsua ez dena ahazteko gaitasuna dute. RNN-en kasuan informazio berria jasotzen den bakoitzean aurretik ezagutzen den informazio guztia eraldatzen da eta jaso berri den informazio guztia gehitzen da, garrantzizkoa zer den kontuan hartu gabe. LSTM-ek, ordea, gelaxka-egoerak (*cell states*) izeneko mekanismo batzuen bidez, informazioari eragiketa desberdinak aplikatzen dizkiote baliozkoa dena soilik gogoratzeko.

Mekanismo hau ulertzeko 4.3 irudia aztertu dezakegu. Berdez irudikatutako gelaxka-egoera bakoitzak hiru sarrera jasotzen ditu: x_t kanpoko sarrera (gure kasuan zuzendu nahi den esaldiko hitz bat), denbora laburreko egoera (*short-term state*) bektorea eta denbora luzeko egoera (*long-term state*) bektorea. Bi bektore hauek aurreko gelaxkatik jasotzen dira eta bertan unera arte aztertutako azpi-sekuentziaren informazio garrantzitsua biltzen da. Gelaxka bakoitzean sarrerako 3 bektoreen transformazioak egiten dira irudian σ ikurrarekin adierazten diren ateak erabiliz. Ate hauek sigmoide funtzioa erabiltzen dute, irteera bezala 0 eta 1 arteko balio bat ematen duen funtzioa. Balio honek sarean zehar bektoreetako bakoitzaren zenbat informazio barreiatu behar den adieraziko du, bakoitzaren garrantziaren arabera. Sigmoide funtzioarekin 1 balioa lortzen bada, atea itxita egongo da eta informazio guztia pasako da aurrera. 0 balioa lortzen bada, atea guztiz zabalik egongo da eta informazio hori ez da erabiliko (ahaztu egingo da). Beraz, sigmoidearekin lortutako balioa 1-etik hurbil egoteak uneko informazioa garrantzitsua dela adierazten du.



4.3 Irudia: LSTM sarearen arkitektura.

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

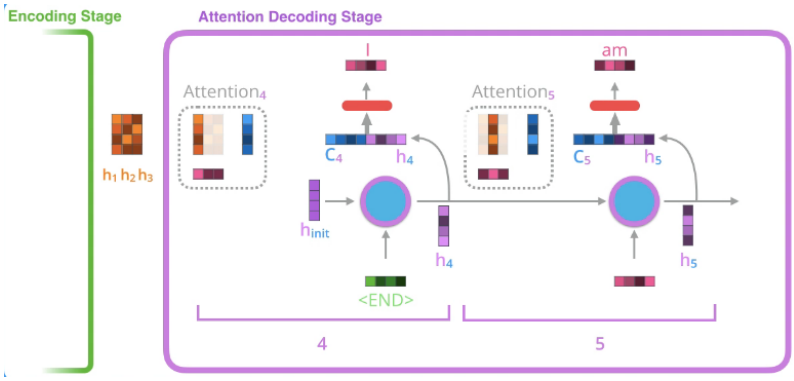
Esaldi luzeekin LSTM-ek RNN-ek baino arrakasta handiagoa duten arren, esaldiak oso luzeak badira LSTM-ek ere ez dute lortzen erlazio eta dependentzia guztiak ikastea, prozesatzen ari den uneko hitzetik oso urrun dagoen hitz baten testuingurua gogoratzearen probabilitatea esponentzialki txikitzen delako bi hitzen arteko distantzia handitu ahala. Gainera, LSTM zein RNN-ekin suertatzen den beste arazo bat paralelizatzeko zailtasuna da, esaldi osoak prozesatzeko hitzez hitz egin beharra dagoelako. Menpeko hitzen arteko distantziaren arazoari aurre egiteko atentzio-mekanismoa erabiltzea proposatu zen.

4.3 Atentzioa

Atentzio-mekanismoa sare neuronaletan erabiltzen den teknika bat da, jasotako informazioaren azpi-atal jakin batean fokua jartzeko erabiltzen dena. Gure lanaren kasuan, foku hori zuzendu nahi den esaldiaren hitz jakin batzuetan jarriko litzateke. Teknika hau Bahdanau et al., 2014 [29] eta Luong et al., 2015 [30] lanetan proposatu zen lehenengoz eta oso baliagarria suertatu da itzulpen automatikoaren kalitatea hobetzeko.

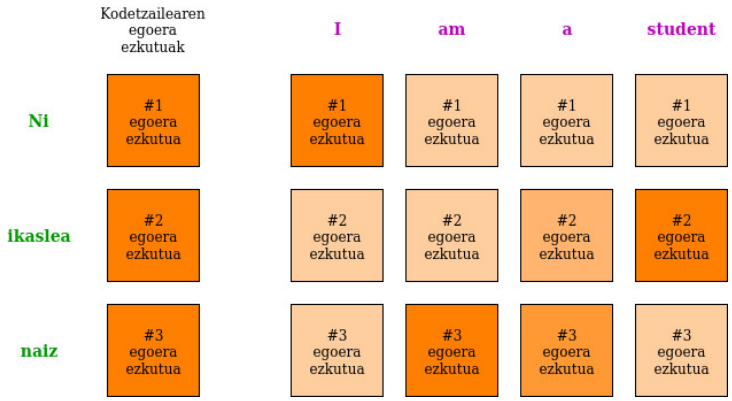
Atentzioa sare errekurrenteekin erabiltzen badugu, kodeketa fasean esaldi osoa egoera ezkutu bakarrean kodetu ordez, hitz bakoitzak bere errepresentazio kodetua izango du eta errepresentazio guztiak pasatuko zaizkio deskodetzaileari. Sarrerako esaldiko hitz guztietan informazio garrantzitsua egon daitekeenez, helburua deskodetzaileak hitz guztiak kontuan hartzea da, irteerako esaldia zehatzagoa izan dadin.

Deskodetzaileak jarraitzen duen prozesua 4.4 irudian ikus daiteke. Demagun t = 4 unean gaudela, deskodetze prozesuaren hasieran. Lehenengo RNN-ak <END> tokena eta des-



4.4 Irudia: *Seq2Seq* modeloa atentzioa erabiliz.

Iturria: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/



4.5 Irudia: Atentziodun itzulpen automatikoaren adibidea. Ingelesezko hitz bakoitza lortzeko euskarazko zein hitzetan jarri den atentzioa adierazten da.

kodetzailearen hasierako egoera ezkutua (h_{init}) jasotzen ditu eta informazio horrekin h_4 egoera ezkutua sortzen du. Ondoren, uneko hitza lortzeko beharrezko testuingurua errepresentatzen duen C_4 bektorea lortzen da (Attention$_4$ gelaxkako bektore urdina). Bektore hori lortzeko, kodetzailetik jasotako egoera ezkutuei (h_1, h_2 eta h_3) puntuazio bat esleitzen zaie, puntuazio horiei *Softmax*[1] funtzioa aplikatzen zaie, egoera ezkutu bakoitza dagokion *Softmax* balioarekin biderkatzen da eta biderketekin lortzen diren bektoreak gehitzen dira. Azkenik, C_4 testuinguru-bektorea eta h_4 egoera ezkutua kateatzen dira. Lortutako bektoreak emango digu uneko hitza, aurreranzko barreiatze-sare bat erabiliz, adibidez.

4.2 irudian aurkeztutako adibidera bueltatuz, 4.5 irudian ikus dezakegu atentzioa zein hitzetan jarriko litzatekeen pausu bakoitzean. Egoera ezkutu bakoitzak kolore laranjaren intentsitate desberdina du, intentsitate altuagoak atentzio-maila altuagoa adierazten duelarik. Horrela bada, ingeleseko "*I*" hitza lortzeko, jatorriko euskarazko esaldiaren hitzen artean "Ni" hitzari eskaini behar zaio atentzio gehien. Era berean, "*am*" hitza lortzeko informazio gehiena "naiz" hitzetik eskuratu behar da eta "*student*" hitza lortzeko "ikaslea" hitzetik. Berezia da "*a*" hitza sortzearen kasua, singularrean hitz egiten ari garela bi hitzek adierazten baitute: "ikaslea" eta "naiz".

Atentzio-mekanismoak dependentzien arazoa murrizten du, baina ez du paralelizazioarekin laguntzen. Tamaina handiko datu-multzoentzat paralelizatzeko gaitasun ezak konputazio denbora luzeegia suposatu dezake. Hori ekiditeko Sare Neuronal Konboluzionalak erabiltzea proposatzen da.

[1] *Softmax* funtzioak zenbaki errealen bektore bat normalizatzen du, bektoreko elementu bakoitzari sarrerako balioaren esponentzialarekiko proportzionala den probabilitate bat esleituz. Hau da, *Softmax* funtzioa aplikatu ondoren, bektoreko elementu bakoitzaren balioa (0,1) tartean egongo da eta bektoreko elementu guztien batura 1 izango da.

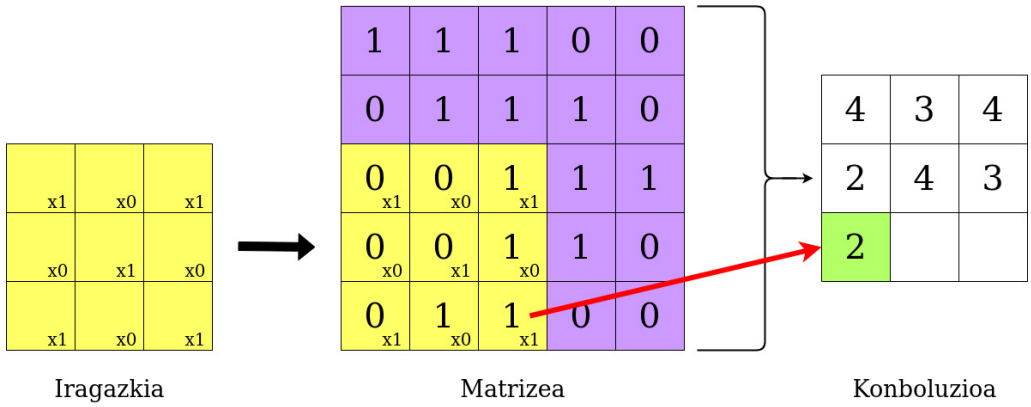
4.4 Sare Neuronal Konboluzionalak

Sare Neuronal Konboluzionalak, ingelesez *Convolutional Neural Networks* edo CNN, konboluzioa erabiltzen duten sare neuronal sakonak dira. Konboluzioa bi funtzioren eragiketa matematikoa da, bietako batek bestearen gain eragiten duen aldaketa adierazten duen hirugarren funtzio bat ematen duena. Mota honetako sareen erabilera ohikoena irudien analisiaren arloan ematen da, baina hizkuntzaren prozesamenduan ere erabili daitezke, irudi bateko pixelek osatzen duten matrizearekin lan egin ordez hitzen bektoreek osatzen dutenarekin lan eginez.

CNN-ak hainbat konboluzio-geruzaz osatutako sareak dira. Geruza bakoitzean iragazki desberdinak (ingelesez *filter* edo *kernel*) aplikatzen dira eta ondoren aktibazio-funtzioak erabiltzen dira geruza bakoitzeko irteera kalkulatzeko. Iragazkien funtzionamendua ulertzeko 4.6 irudia lagungarria da. Iragazkia horiz irudikatzen da eta morez irudikatutako matrizearen gain aplikatzen da. Matrize horren balioek irudien pixelak adierazten dituzte edo, hizkuntzaren prozesamenduaren kasuan, hitzen bektoreak. Iragazkiaren balioak entrenamendu garaian ikasten dira eta matrizean barrena biderkatzen dira, konboluzio-matrizea lortuz. Adibide honetan iragazkia 3x3 tamainakoa denez, lehenengo pausuan iragazkiaren elementu bakoitza matrizeko lehenengo 3 errenkada eta lehenengo 3 zutabeetako dagokion elementuarekin biderkatuko da eta ondoren 9 balio horiek gehituko dira, konboluzio-matrizeko lehenengo balioa lortuz. Bigarren pausuan iragazkia posizio bat mugituko dugu eskuinerantz eta prozesua errepikatuko dugu, konboluzio-matrizeko bigarren elementua lortuz. 4.6 irudian, zehazki, 7. pausua da irudikatzen dena. Pausu honetan matrizeko 1, 2 eta 3 zutabeetako eta 3, 4 eta 5 errenkadetako balioak iragazkiaren balioekin biderkatzen dira eta biderketen emaitzak gehitzen dira, 4.1 ekuazioan ikusten den bezala. Emaitza, konboluzio-matrizeko 7. posizioari (3. errenkadako lehenengo balioa) dagokiona, berdez irudikatzen da 4.6 irudian.

$$\begin{bmatrix} 0\cdot 1 & 0\cdot 0 & 1\cdot 1 \\ 0\cdot 0 & 0\cdot 1 & 1\cdot 0 \\ 0\cdot 1 & 1\cdot 0 & 1\cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow[elementuak\ gehitu]{} 0\times 7+1\times 2=2 \tag{4.1}$$

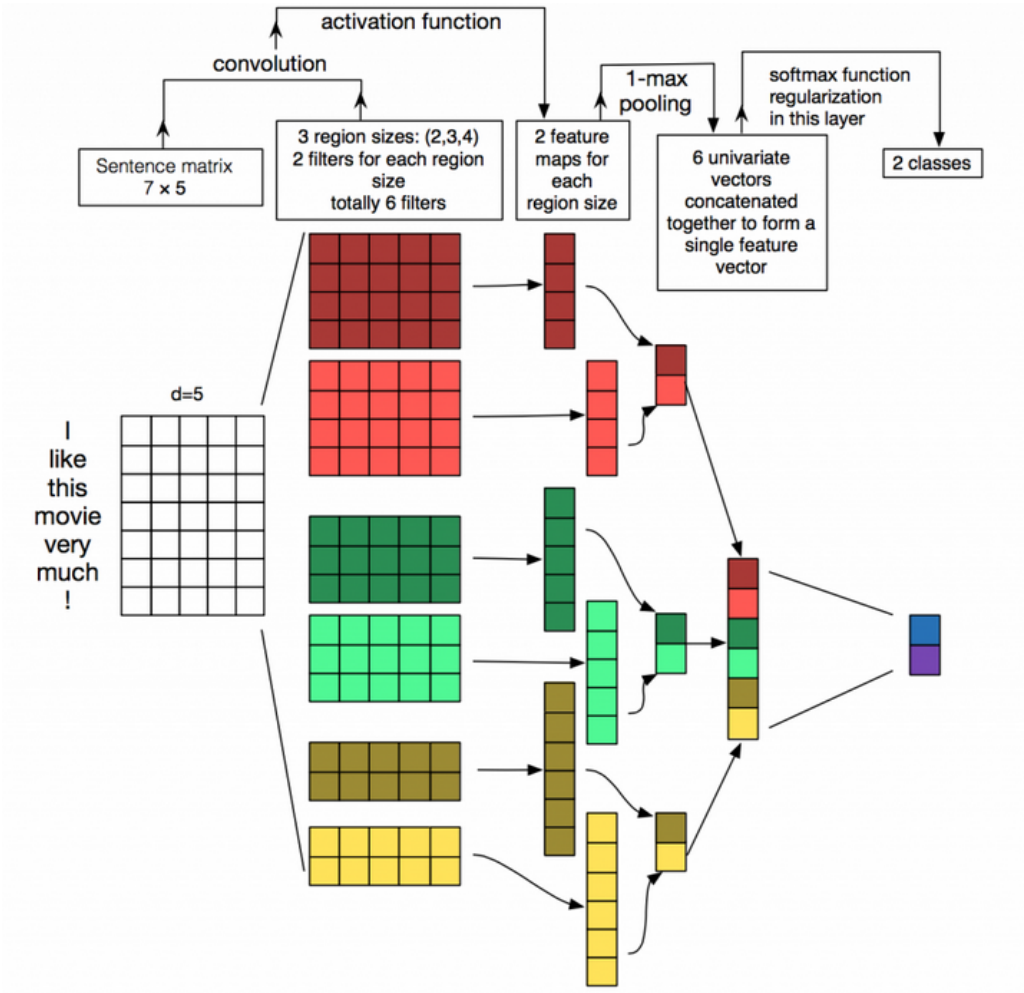
Geruza bakoitzeko iragazkiak aplikatu ondoren normalean *pooling* geruzak erabiltzen dira konboluzioaren elementu bakarra hautatzeko. Ohikoena balio maximoa duen elementua (*max-pooling*) edo elementu guztien batezbestekoa (*avg-pooling*) aukeratzea da.



4.6 Irudia: Iragazkien erabilera Sare Neuronal Konboluzionaletan.

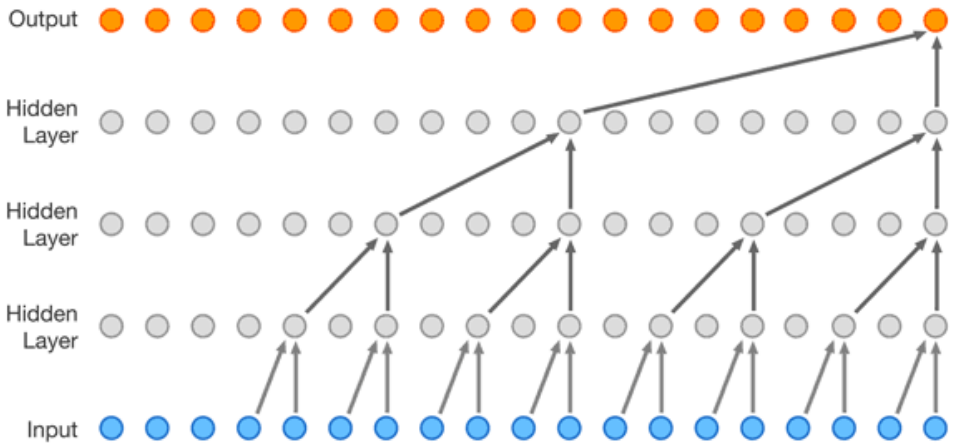
Hizkuntzaren prozesamenduan aplikatutako CNN bat ikus dezakegu 4.7 irudian. Aztertzen den esaldia "*I like this movie very much!*" da eta helburua sailkapen bitarra egitea da, esaldia positiboa edo negatiboa den erabakitzeko. Matrizearen zabalera 5 tamainakoa da eta luzera esaldiaren token kopuruak zehazten du, kasu honetan 7. Iragazkien tamaina zehazteko matrizearen zabalera mantentzen da eta luzera aldatzen da, normalean 2-5 tartean, une bakoitzean 2-5 hitz aztertu ahal izateko. Kasu honetan 6 iragazki aplikatzen dira: 4x5 tamainako bi (gorriak), 3x5 tamainako bi (berdeak) eta 2x5 tamainako beste bi (horiak). Hitzen bektoreen matrizearekin eta iragazki hauetako bakoitzarekin konboluzioa eta aktibazio-funtzioa aplikatzen dira eta tamaina desberdineko 6 bektore lortzen dira, irudian *feature maps* bezala adieraziak. Bektore hauetako bakoitzean *max-pooling* aplikatzen da, horrela 6 elementuko bektore bat lortuz. Azkenik, *Softmax* funtzioa erabiltzen da bektoreko elementu bakoitzari probabilitate bat esleitzeko eta horrela esaldia sailkatu ahal izateko.

CNN sareek paraleloan lan egin dezakete esaldi bateko hitz bakoitza aldi berean prozesatu daitekeelako, aurreko hitzen dependentziarik gabe. Horrez gain, irteerako hitzen eta sarrerako edozein hitzen arteko distantzia log(N) ordenakoa da, 4.8 irudian ikusi daitekeen bezala. Distantzia hau sare errekurrenteetako hitzen arteko distantzia baino hobea da, lehen N ordenako distantziarekin ari baikinen lanean. Dena den, sare konboluzionalek ez dute dependentzien arazoa konpontzen, bereziki esaldi luzeen kasuan. Orain arte aipatu ditugun oztopo guztiei batera aurre egiteko CNN-ak eta atentzio-mekanismoa konbinatzen dituen eredu bat proposatu zen: Transformer-a.



4.7 Irudia: CNN-en erabilera hizkuntzaren prozesamenduan.

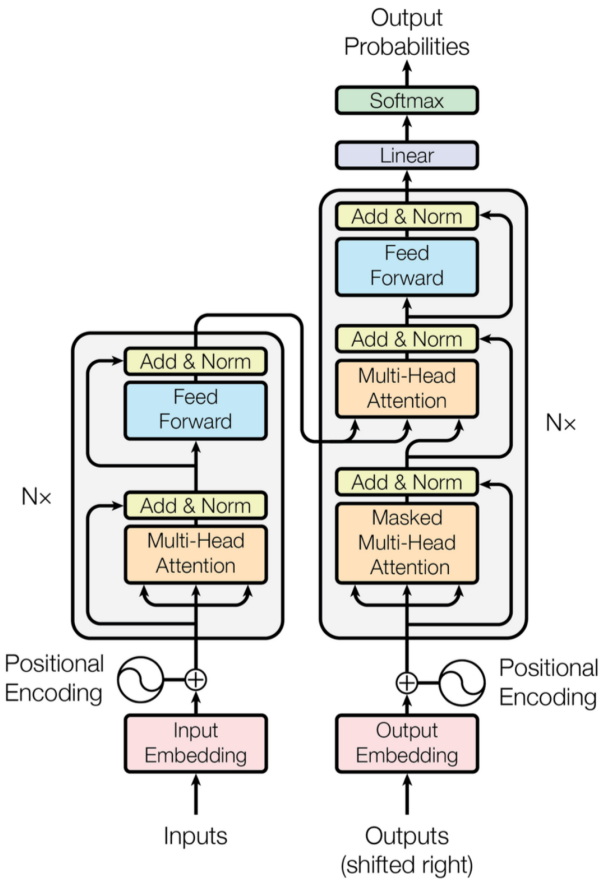
Iturria: Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.



4.8 Irudia: Wavenet sarea, testu-ahots eraldaketan erabiltzen den CNN baten adibidea.

Iturria: https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

4.5 Transformer arkitektura



4.9 Irudia: Transformer ereduaren arkitektura.

Iturria: A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.Ñ. Gomez, Ł. Kaiser,and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems.

Transformer arkitektura 2017. urtean aurkeztu zen ordura arte sekuentziekin lan egiteko existitzen ziren ereduen ahuleziei aurre egiteko asmoz. Aipatu dugun bezala, aurretik ezagutzen ziren modeloek sare errekurrente edo konboluzional konplexuak eta atentzio mekanismoak elkartzen zituzten sekuentziekin zerikusia zuten atazak ebazteko. Transformer arkitekturan, ordea, atentzioa soilik erabiltzea proposatzen da [31].

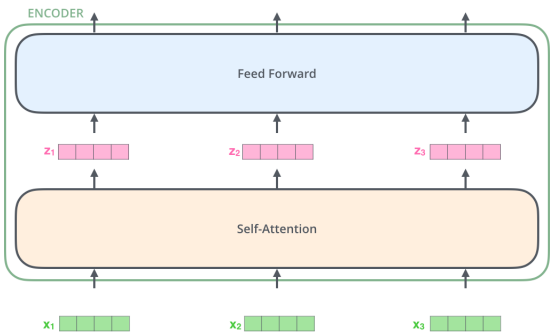
Transformer-ak sekuentzia bat beste sekuentzia batean bilakatzen du horretarako kodetzaileak eta deskodetzaileak erabiliz. 4.9 irudian ikusi dezakegu Transformer ereduaren arkitektura. Ezkerrean kodetzaile-osagaia irudikatzen da eta eskuinean deskodetzaile-osagaia. Kodetzaile-osagaia kodetzaile-pila bat da eta deskodetzaile-osagaia deskodetzaile-pila bat. Kodetzaileak sarrera-esaldiak jasotzen ditu eta deskodetzaileak helburu-esaldiak.

Esaldiak ezin dira zuzenean *string* moduan erabili, ordenagailuek eta Ikasketa Automatikoan erabiltzen diren ereduek ezin dutelako testua gizakiok bezala irakurri eta ulertu. Ondorioz, esaldiak testu bezala erabili ordez, hauen *embedding*-ak erabiltzen dira. *Embedding* hauei *Positional Encoding* gehitzen zaie, esaldien sekuentzia gogoratu dezakeen sare errekurrenterik erabiltzen ez denez, hitz bakoitzaren posizio erlatiboaren informazioa ere kontuan hartzeko.

Kodetzaile-pila osatzen duten kodetzaile guztiek egitura berdina daukate, baina ez dituzte pisuak partekatzen. Kodetzaile bakoitza bi azpi-ataletan banatzen da: Buru Anitzeko Atentzioa (*Multi-Head Attention*) eta aurreranzko barreiatze-geruza. Sarrera-esaldien *embedding*-ak atentzio geruzatik pasatzen dira lehenengo, esaldiko hitz bakoitza kodetzeko orduan garrantzia duten esaldiko gainontzeko hitzak ere kontuan hartzeko, eta ondoren aurreranzko barreiatze-geruzatik pasatzen dira. Deskodetzaile bakoitzak ere bi atal horiek ditu, baina horrez gain beraien artean beste atentzio geruza bat dago, deskodetzaileak sarrera-esaldiko zati garrantzitsuetan arreta jartzeko erabiltzen dena.

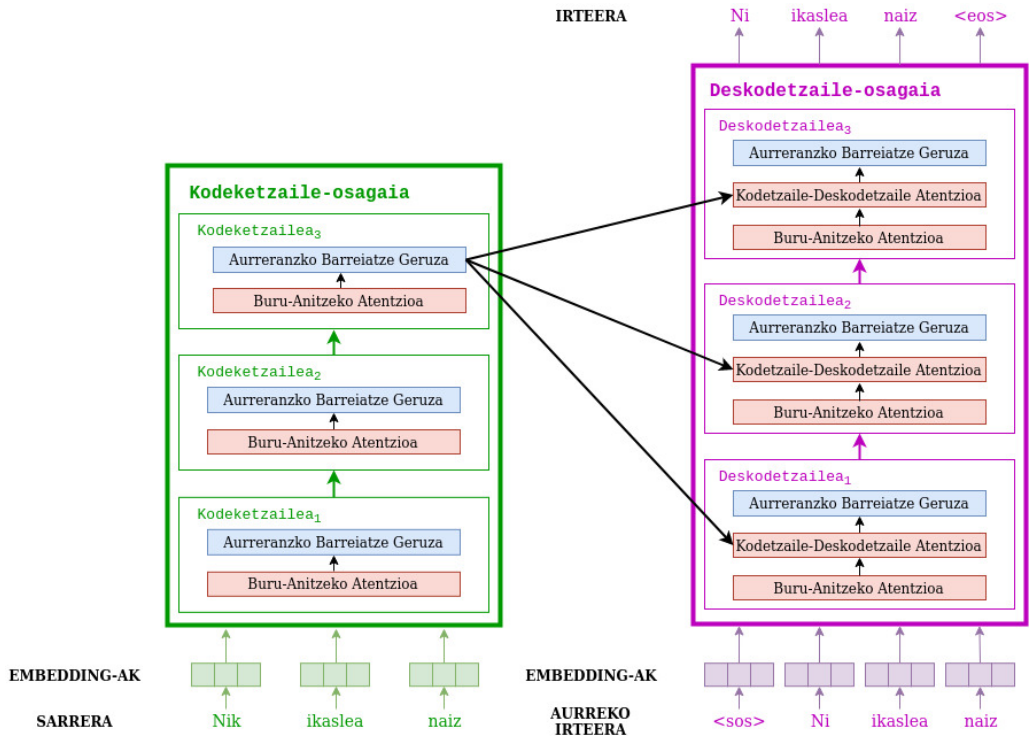
4.10 irudian ikusi daitekeen bezala, hitz bakoitzak bere bide propioa jarraitzen du kodetzailean. Atentzioaren barruan bide hauen arteko dependentziak existitzen dira baina aurre barreiatzean guztiz independenteak dira, paralelizatzeko aukera emanez. Paralelizaziorako ahalmen hau da Transformer-aren abantaila nagusienetako bat.

Entrenamendua burutu ondoren, sistemari sarrera-esaldi bat emanda (gure kasuan esaldi erroreduna) eta esaldi-hasiera (*start-of-sentence (sos)*) tokenetik abiatuz, sistema helburu-esaldiaren (gure kasuan zuzenketa) lehenengo hitza sortzeko gai izan beharko litzateke. Ondoren, esaldi-hasiera eta lehenengo hitza emanda, bigarren hitza iragarriko luke. Prozesu hau errepikatuko litzateke esaldi-amaiera (*end-of-sentence (eos)*) tokena lortu arte (4.11 irudia).



4.10 Irudia: Kodetzailea 3 hitzeko esaldi bat jasotzen.

Iturria: http://jalammar.github.io/illustrated-transformer/



4.11 Irudia: Transformer arkitekturako kodetzaile- eta deskodetzaile-osagaien funtzionamendua zuzenketa gramatikalerako. Kasu honetan pila 3 elementuz osatuta dago.

Nabarmentzekoa da deskodetzailearen sarrera, hau da, helburu-esaldia, eskuinerantz mugitzen dela posizio bat. Hau egiten da modeloak ez dezan besterik gabe esaldi hori kopiatu eta horren ordez kodetzaileko eta deskodetzaileko sekuentzia jakin batzuk emanda hurrengo hitza iragartzen ikasi dezan. Mugitzerakoan libre uzten den posizio horretan txertatzen da esaldi-hasiera tokena.

Transformer ereduaren funtzionamendua sakonago azaltzen da 5.6.1 atalean.

5. KAPITULUA

Diseinua eta Inplementazioa

Kapitulu honetan zuzentzaile gramatikala inplementatzeko jarraitutako prozesua deskribatzen da. Alde batetik, entrenamendurako erabilitako datuen sorkuntza azaltzen da: corpusaren egitura definitzea, esaldiko errore-proportzioa erabakitzea eta esaldi erroredun eta zuzenen bikote kopurua zehaztea, besteak beste. Bestetik, entrenamendua burutzeko erabilitako arkitekturaren funtzionamendua, entrenamenduaren ezaugarriak eta zuzenketen sorkuntza azaltzen dira.

Proiektua garatzeko erabili den programazio-lengoaia *Python* izan da. Inplementazioa *Google Colaboratory (Colab)* plataforman egin da, *Python* kodea nabigatzailean idatzi eta exekutatzea ahalbidetzen duena. *Colab* erabili ahal izateko ez da inolako aurrekonfiguraziorik behar eta bere abantaila nagusia *Python* osatzen duten liburutegi eta pakete guztiak erabili daitezkeela da, inolako instalaziorik gabe. Gainera, GPU-ak erabiltzea ere ahalbidetzen du.

Gure proiektuan erabili ditugun tresnen artean *numpy*[1] eta *pandas*[2] liburutegiak aurki daitezke, baina bereziki aipagarriak dira *Pytorch* liburutegia, ikasketa automatikorako diseinatua, eta *torchtext* paketea, datuen prozesaketa lantzeko tresna anitzez eta hizkuntza naturaleko datu-multzo ezagunez osatua.

[1] Dimentsio anitzeko *array*-ekin lan egiteko liburutegia, maila altuko funtzio matematikoak eskaintzen dituena.

[2] Datuak maneiatzeko eta hauen analisia egiteko maila altuko tresna.

5.1 Erroreen analisia

Zuzenketarekin hasi aurretik, garrantzitsua da euskaraz existitzen diren errore gramatikalen azterketa egitea, mota guztietako erroreekin lan egingo dugun edo mota jakin batean zentratuko garen erabakitzeko.

Gure lehenengo lana euskarazko errore motak sailkatzea izan da. Etiketatutako euskarazko testu erroredun multzo handirik ez daukagunez, entrenamendurako erroreak guk sortu beharko ditugu eta, beraz, errore motak sailkatzeko irizpidea hurrengoa izan da: errore mota hori sortzeko esaldiaren analisirik egin behar ote den. Garrantzitsua da sorkuntza-analisia eta zuzenketa-analisia ez nahastea. Mota jakin bateko erroreak sortzeko hitzzerrenda bat erabili dezakegu, analisirik gabe zerrendako hitzak testu zuzenean txertatuz adibidez, baina errore hori antzeman eta zuzentzeko esaldiaren analisia egin behar izatea gerta daiteke. Guk egindako errore moten sailkapena sorkuntzarako analisiari dagokio.

Demagun "Ni Donostian bizi naiz" esaldia daukagula. Esaldi hori gramatikalki zuzena da. Suposa dezagun esaldi horretatik abiatuz komunztadura motako errore bat duen esaldi bat sortu nahi dugula, "Nik Donostian bizi naiz" adibidez. Errore hori sortzeko jatorrizko esaldiaren analisi bat burutu behar izan dugu aditza eta subjektuaren kasua zein den identifikatzeko eta ondoren subjektuaren kasua aldatu, aditzaren kasuarekin bat etorri ez dadin. Hau kontuan izanda, aditzaren eta subjektuen kasuaren komunztadura, analisia behar duen errore gramatikal bezala sailkatuko genuke. Demagun orain "Berak nik baino hobeto egin du" esaldi zuzena daukagula. Ohiko errorea adberbioaren ordez antzekoa den adjektiboa erabiltzea da, hau da, "Berak nik baino hobe egin du" esatea. Horrelako erroreak sortzeko, "hobeto" hitza aurkitzen dugun esaldietan "hobe" hitzarekin ordezkatu besterik ez dugu egin behar, eta alderantziz. Beraz, adberbioak eta adjektiboak nahastuz sortzen diren erroreak esaldiaren analisirik egin gabe sortu daitezkeela esan dezakegu.

Euskaraz aurki daitezkeen erroreen zerrenda Euskarazko erroreen sailkapena lanetik [32] lortu dugu. Bertan, B eranskinean, euskarazko erroreak 7 kategoria nagusitan sailkatzen dira eta errore bakoitzaren adibideak adierazten dira. Zerrenda honetan oinarrituz burutu dugu sorkuntzarako analisia behar duten erroreen eta behar ez dutenen sailkapena (A eranskinean eskuragarri).

Sailkapen horretan, 33 motako erroreak sortzeko esaldiaren analisi bat egin beharko genukeela ondorioztatu dugu. Guztira 60 errore gramatikal definitzen direla kontuan izanda, erdia baino gehiago sortzeko analisia egin beharko genuke. Gainera, sortzeko analisia behar duten errore gehienak komunztadura motakoak dira, gure hizkuntzan errore ohikoen

artean aurki daitezkeenak hain zuzen ere. Hori horrela bada, gizakiok idatzitako testuetan maizen egiten ditugun erroreek komunztadurarekin zerikusia dutenez[3], sortu beharko genituzkeen esaldi gehienek komunztadura errore errealen antza eduki beharko lukete, eta horrek analisi lan handia suposatuko luke. Hori ekiditeko, hasiera batean ausazko erroreak sortzea erabaki dugu (5.3 atalean azaldutako metodoa erabiliz), erroreak modu zentzudunean sortzeko esaldien analisia behar ote den erreparatu gabe. Hala ere, analisia erabiliz errore zentzudunagoak sortuko genituzkeela badakigu, eta horrek zuzenketen kalitatea hobetuko luke. Hori dela eta, errore zentzudunak sortzeko moduak eztabaidatu ditugu, etorkizuneko lanean (7.2 atala) deskribatzen direnak.

5.2 Datu-multzoa eta aurre-prozesaketa

Zuzentzailea entrenatzeko esaldi erroredun eta bere zuzenketaren bikoteak beharko ditugu. Bikote horiek lortzeko euskaraz idatzitako albiste eta artikuluetatik abiatuko gara. Testu hauek informazio-iturri ofizialetan argitaratuta daudenez, erabiltzen duten euskara zuzena dela suposatuko dugu eta gure lana bertako esaldiak moldatuz esaldi erroredunak sortzea izango da.

Gure jatorrizko corpusa 3 albiste eta artikulu bildumaz osatuta dago. Lehenengoa Euskal Irrati Telebistako (EiTB) testuen multzoa da, 477.072 esaldiz osatutakoa. Bigarrena Elhuyarren euskarazko web-corpus elebakarra da[4], 6.880.348 esaldiz osatua. Azkenik, Egunkaria-ko testuak ere erabili ditugu, hauek 1.426.527 esaldi dituztelarik. Horrela bada, guztira 8.783.947 esaldi zuzeneko datu-multzoarekin egingo dugu lan. Corpus bakoitzaren ezaugarrien laburpena 5.1 taulan bildu dugu.

	EiTB	**Elhuyar**	**Egunkaria**
Esaldi kopurua	477.072	6.880.348	1.426.527
Hitz kopurua	8.231.866	124.065.599	23.546.175
Token kopurua	10.056.271	148.522.349	27.424.277
Token desberdin kopurua	288.954	2.814.209	625.749

5.1 Taula: Corpusen ezaugarriak: esaldi kopurua, hitz kopurua, token kopurua eta token horien artean desberdinak direnen kopurua.

[3] Baieztapen honen adibide bat ikusi daiteke 5.4.3 atalean. Bertan ebaluaziorako corpusa sortzen da eta corpus horretan erabiltzen diren esaldi errealetan (gizakiok idatziak) aurkitzen diren erroreen artean %60a baino gehiago komunztadura erroreak dira.

[4] Informazio gehiago: http://webcorpusak.elhuyar.eus/sarrera_elebakarra.html

Erroreak sortzen hasi aurretik, lehenengo pausua corpusaren aurre-prozesaketa bat egitea izan da, horretarako Moses Machine Translation[5] tresna erabiliz. Tresna honekin testu guztiak hitz-mailan tokenizatu ditugu eta letra larriak letra xehe bihurtu ditugu.

5.3 Erroreen sorkuntza

5.1 atalean azaldutako sailkapenari esker errore mota asko modu zentzudunean sortu ahal izateko analisia beharrezkoa dela ikusi dugu. Lan honetan, erroreen sorkuntza sinplifikatze aldera, esaldietan ausazko zarata aplikatuz sortu ditugu erroreak. Ez dugu kontuan hartu erroreak zentzuarekin sortzeko analisia behar den edo ez; erroreak ausaz sortu ditugu eta gerta liteke komunztadura errore bat sortu izana (sorkuntza-analisia beharko lukeena) edo adjektibo bat adberbio baten truke jarri izana (sorkuntza-analisia beharko ez lukeena), adibidez. Horretarako corpusa esalditan tokenizatu dugu eta esaldi bakoitzean ausazko erroreak sortu ditugu.

Lau errore mota definitu ditugu: ezabatzea, gehitzea, ordezkatzea eta trukatzea. Ezabatzearen kasuan ausaz aukeratzen dugu jatorrizko esaldiko hitz bat eta hitz hori ezabatuz sortzen dugu esaldi erroreduna. Gehitzeari dagokionez, corpus osoko 100.000 hitz maizenen artean bat aukeratzen dugu ausaz eta esaldi originalaren ausazko posizio batean txertatzen dugu. Ordezkatze motako erroreetan esaldi originaleko hitz bat hautatzen dugu ausaz eta hitz horren ordez corpus osoko 100.000 hitz maizenen artean ausazko bat jartzen dugu. Azkenik, trukatzearen kasuan, ausaz aukeratzen ditugu esaldi originaleko 2 hitz eta beraien posizioak aldatzen ditugu. 5.2 taulan ikus daiteke errore hauetako bakoitzaren adibide bana.

Errore mota	Zuzena	Erroreduna
Ezabatu	kirol aerobikoa **egin** dezakeela esan diote .	kirol aerobikoa dezakeela esan diote .
Gehitu	kirol aerobikoa egin dezakeela esan diote .	kirol aerobikoa egin **etxe** dezakeela esan diote .
Ordezkatu	**kirol** aerobikoa egin dezakeela esan diote .	**etxe** aerobikoa egin dezakeela esan diote .
Trukatu	**kirol** aerobikoa egin dezakeela **esan** diote .	**esan** aerobikoa egin dezakeela **kirol** diote .

5.2 Taula: Ausazko errore sorkuntzaren adibideak.

[5]Github-eko esteka: https://github.com/moses-smt/mosesdecoder/tree/master/scripts

Erroreak sortzeko EiTB-ko eta Elhuyarreko corpusak erabili ditugu eta ausaz aukeratu dugu iterazio bakoitzean corpus horietako zein esaldi eraldatuko dugun erroreduna bihurtzeko. Esaldi bat ausaz hautatzen den bakoitzean definitutako 4 erroreen artean zein aplikatuko zaion ere ausaz aukeratu dugu eta sortutako esaldi erroreduna esaldi zuzenaren parean gorde dugu, horrela esaldi erroredun-zuzen bikoteak lortuz. Prozesu hau 1,5 milioi bikote lortu arte errepikatu dugu. Garrantzitsua da kontuan hartzea 1,5 milioi bikote horietako esaldi erroredunek ez dutela zertan errore bakarra izan. Eraldatuko den esaldia ausaz erabakitzen denez, posiblea da iterazio batean aurretik eraldatua izan den esaldi bat egokitzea eta horri beste errore bat aplikatzea.

Erroreak sortzeko kodea *Google Colaboratory* plataforman inplementatu dugu. Ondoren, 1,5 milioi esaldi erroredunak sortzeko, IXA taldeko zerbitzarietan exekutatu dugu kodea.

5.3.1 Hizkuntza-eredua

5.3 atalean deskribatutako errore-sorkuntza metodoarekin posible da esaldi bakarrari hainbat errore aplikatzea. Aukera horri esker maila desberdineko esaldi erroredunak sortzeko gai izango gara, baina beti ziurtatu behar dugu esaldiaren zentzua mantentzen dela. Esaldi batean errore asko sortzen baditugu azkenean ausazko hitz-sekuentzia batekin ariko gara lanean eta hori ezingo genezake euskarazko esaldi bezala kontsideratu. Kasu hau saihesteko hizkuntza-eredu bat erabili dugu.

Gure kasuan kenlm hizkuntza-eredua[6] 5.2 atalean deskribatutako EiTB-ko eta Elhuyarreko corpusekin (erroreak sortzeko erabili dugun datu-multzo bera) entrenatu dugu. Eredu hau kontaketetan oinarritutakoa da eta 5-gramak erabili dira probabilitateak kalkulatzeko. Ondoren, guk sortutako esaldi erroredunei aplikatu diegu, euskara izatearen probabilitatea kalkulatzeko eta euskarazko esaldien "antza" ez duten esaldiak baztertzeko.

Errore-sorkuntza automatikoan hitzak ezabatu eta gehitzen ditugunez esaldi originalek eta erroredunek ez dute luzera berdina izango. Ondorioz, alderaketa bidezkoa egiteko asmoz, hizkuntza-ereduarekin lortutako probabilitateak normalizatu ditugu probabilitatea esaldiaren luzerarekin zatituz. Hauek izango dira esaldiak onartu edo baztertzeko erabiliko ditugun probabilitateak.

Probabilitate normalizatuak eta esaldien ezaugarriak aztertu ondoren onartuko diren esaldiak aukeratzeko hurrengo irizpideak jarraitu ditugu:

[6] Github-eko esteka: https://github.com/kpu/kenlm

- 7 hitz baino gutxiago dituzten esaldiak baztertu dira normalean zentzuzko esaldi bat osatzeko 7 hitz edo gehiago behar direlako.
- Esaldiak onartzeko atalasea -0,4 izan da. -0,4 baino txikiagoa den probabilitate normalizatua duten esaldiak baztertu dira zerrendak edo zentzurik gabeko hitz eta puntuazio-ikur sekuentziak direlako.

5.4 Corpus desberdinen diseinua eta sorkuntza

5.4.1 Entrenamendurako corpusak

Hasierako corpusaren prozesaketa burututa eta erroreak sortuta, hurrengo ataza entrenamendurako corpusak diseinatzea izan da. Guztira 4 corpus sortu ditugu, err_zuz400K, err_zuz5M, err_zuz8M eta err+_zuz8M bezala identifikatuko ditugunak. Izen hauek bi ezaugarri deskribatzen dituzte: azpimarraren aurretik errore-sorkuntza metodoa adierazten da eta azpimarraren atzean corpus bakoitzak duen esaldi zuzenen kopuruaren hurbilpen bat egiten da. Datu-multzo hauen ezaugarrien laburpena 5.3 taulan ikus daiteke.

Hasiera batean gure helburua milioi eta erdi esaldi inguru erabiliz entrenamendu bakarra egitea zen. Esaldi horiek automatikoki sortutako 1,2 milioi esaldi erroredun (esaldi erroredunez eta esaldi zuzenez osatutako bikoteak) eta 400.000 esaldi zuzenetan (esaldi zuzena birritan edukiko duten bikoteak) banatzen dira[7] eta err_zuz400K datu-multzoa osatzen dute. Hasierako intuizioak erroreak zuzentzeko tresna bat garatzeko sistemaren entrenamenduan zuzenketarik behar ez duten esaldiak gehitzeak zentzu askorik ez daukala esan arren, garrantzitsua da esaldi zuzenak ere erabiltzea, beharrik ez dagoen egoeretan

	Esaldi kopurua	**Esaldi erroredun kopurua**	**Esaldi zuzen kopurua**	**Esaldi kopuru finala**
err_zuz400K	1.600.000	1.200.000 ∼	400.000 ∼	1.582.029
err_zuz5M	6.600.000	1.200.000 ∼	400.000 ∼ + 5.000.000	6.519.020
err_zuz8M	9.906.875	1.200.000 ∼	400.000 ∼ + 8.306.875	9.797.734
err+_zuz8M	9.916.784	1.132.837	8.783.947	9.812.089

5.3 Taula: Diseinatutako corpusen ezaugarriak laburbiltzen dituen taula. "∼" ikurrak balio zehatza ezagutzen ez dela adierazten du, esaldi horiek esaldi erroredunak eta zuzenak nahastuta zituen datu-multzo batetik ausaz hautatu baitira.

[7] Kopuruak ez dira zehatzak, esaldi erroredunez eta zuzenez osatutako 2 milioi esaldiko corpus batetik ausaz 1,6 milioi esaldi hautatu baitziren.

sistemak aldaketarik egiten ez duela ziurtatu nahi dugulako. Ez dugu sistemak beti zerbait zuzendu behar duela ikastea nahi; erroredun esaldiak identifikatzea eta hauetan zuzenketa aproposa aplikatzea nahi dugu.

Sortutako err_zuz400K datu-multzoarekin sistema entrenatu ostean esaldi kopuru murriztu batekin (5.4.2 atalean azaltzen den corpusa) ebaluazio sinple bat egin dugu[8] eta 5.4 taulan ikusten direnak izan dira lortutako zuzenketa batzuk. Emaitza horiei erreparatuz hobekuntza tartea handia dela antzeman dugu.

Zuzenketa kaskarren arrazoia azaltzeko hipotesi bat entrenamenduan erabilitako esaldi kopurua txikiegia izatea da. Entrenamendurako esaldi gutxi badaude, sistemari egitura berri bat aurkeztuko zaio ikusten duen esaldi bakoitzean eta, ondorioz, ezin izango ditu patroiak eta egitura zehatzak ikasi.

Hipotesi hau egiazkoa ote den frogatzeko err_zuz5M izeneko 2. datu-multzoa sortu dugu. Hori egiteko err_zuz400K datu-multzoko esaldiei 5.2 atalean aurkeztutako Elhuyarreko corpusetik ausaz hautatutako 5 milioi esaldi zuzen gehitu dizkiegu. Horrela, 6,6 milioi esaldi edukiko ditu err_zuz5M datu-multzoak.

Entrenamendua err_zuz5M datu-multzoarekin egin ostean zuzenketen kalitatea hobetu dela antzeman dugunez, hirugarren corpus bat sortzea erabaki dugu, oraingoan 10 milioi esalditara hurbilduz. Beraz, err_zuz8M datu-multzoa lortzeko err_zuz400K corpuseko esaldi guztiak, Elhuyarreko webcorpus osoa (6.880.348 esaldi) eta Egunkaria-ko corpus osoa (1.426.527) elkartu ditugu, guztira 9.906.875 esaldi bilduz.

Bigarren hipotesi bat ere landu dugu: automatikoki sortutako esaldi erroredunak sinpleegiak izatea. 5.4 taulan ikusi daitekeen bezala, kasu askotan sistemak proposatutako zuzenketa esaldi okerraren berdin-berdina da. Baliteke sortutako esaldi erroredun gehienetan errore bakarra egotea eta, ondorioz, sistemak errore hori zuzendu beharrean esaldia kopiatu besterik ez egitea. Suposa dezagun 10 hitzeko esaldi bat dugula eta sortutako errorea esaldiko hitzetako bat beste hitz batekin ordezkatzea izan dela. Zuzenketa bezala esaldi eraldatua bera proposatzen badugu, esaldi originaleko 10 hitzetatik 9 asmatuko genituzke, hau da, %90eko arrakasta edukiko genuke. Arrazoi horregatik posiblea da sistemak kasu gehienetan aukera hoberena esaldia kopiatzea dela ikastea.

Arazo honi aurre egiteko azkenengo datu-multzo bat sortzea erabaki dugu, err+_zuz8M izenekoa, erroreak automatikoki sortzeko metodoa pixka bat aldatuz (horregatik erabiltzen dugu "err+" corpusaren izenean). 5.3 atalean definitutako 4 erroreak mantendu ditugu eta erroreak sortzeko erabilitako corpusa berdina izan da, baina kasu honetan iterazio

[8] Ebaluazio hau sakonago lantzen da 6.2 atalean.

Esaldi erroreduna	Zuzenketa automatikoa
nik ane deitzen naiz	nik ane deitzen naiz
ni oporretan joango dira	ni oporretan joango dira
zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke ?
euskal kultur erakundea , iparraldeko egiteko kultur erakunde nagusia duten da .	euskal kultur dago , iparraldeko kultur erakunde nagusia duten dago .
bestalde . gorpua eskatu epailearen esku gelditu da ,	bestalde , gorpua , epailearen esku gelditu da .
errespetatzen al duzu amatasun eta aitatasun baimena ?	errespetatzen al duzu amatasun eta aitatasun baimena ?
hauek ere , eskean ibiltzeaz gain , dantzan ziren .	hauek ere , eskean ibiltzeaz gain , dantzan izan ziren .

5.4 Taula: err_zuz400K datu-multzoarekin lortutako zuzenketa batzuk.

bakoitzean ausaz aukeratutako esaldian errore bakarra sortu ordez, esaldiaren %25a eraldatu dugu. Honek zera esan nahi du: 4 hitzeko esaldi batean errore bat gehituko dugu, 8 hitzeko esaldi batean 2 errore, eta abar[9]. Metodo honekin 1.132.837 esaldi erroredun sortu ditugu.

Hirugarren datu-multzoarekin zuzenketak are gehiago hobetu direla ikusita eta erroreak automatikoki sortzeko bigarren teknika erabiliz lortutako esaldiak milioi bat baino gehiago izanez, err+_zuz8M datu-multzoa osatzeari ekin diogu. Egindako proba desberdinetan erabilitako esaldi kopurua handitu ahala zuzenketen kalitatea hobetu dela ikusita, automatikoki sortu berri ditugun 1.132.837 esaldi erroredunei 5.2 atalean aipatutako corpusetako esaldi zuzen guztiak (8.783.947 esaldi) gehitu dizkiegu. Horrela bada, err+_zuz8M datu-multzoa 9.916.784 esaldiz osatuta dago.

5.3 taula aztertzen badugu, corpus bakoitzerako deskribatu dugun esaldi kopurua eta taulako azken zutabea, esaldi kopuru finala adierazten duena, bat ez datozela ikus dezakegu. Guk diseinatutako datu-multzoek lehenengo zutabean adierazten den esaldi kopurua dute (kopuru hori da atal honetan definitu duguna), baina sistema entrenatzeko erabili aurretik esaldi luzeegiak baztertzea erabaki dugu. Zehazki, 90 hitz baino gehiago dituzten testuak utzi ditugu alde batera, oso zaila delako luzera horretako esaldi zentzudunak topatzea. 90 hitz baino gehiagoko sekuentzia bat aurkitzen bada ziurrena hitz soltez osatutako zerrenda bat izatea edo gaizki tokenizatutako testu bat izatea da. Horrela bada, 5.3 taulako azken zutabean adierazten den esaldi kopurua da entrenamendu bakoitzean erabilitako esaldi kopuru finala.

[9]Esaldiaren hitz kopurua 4ren multiploa ez izatekotan errore kopurua borobildu dugu. 10 hitzeko esaldi batean 2,5 errore sortu beharko genituzke; 2 sortu ditugu. 15 hitzeko esaldi batean 3,75 errore sortu beharko genituzke; 4 sortu ditugu.

5.4.2 Garapenerako corpusa

Garapenerako corpusa 5.4.1 atalean definitutako corpus bakoitzarekin sistema bat entrenatu ondoren, sistema horren kalitatea neurtzeko erabili dugun datu-multzoa da. Corpus hau 26 esaldiz osatuta dago, horien artean sistemek entrenamendu garaian dagoeneko ikusi dituzten esaldi batzuk aurkitu daitezkeelarik.

Datu-multzoa osatzeko erabilitako 26 esaldiak 3 multzotan banatu ditzakegu: guk asmatutako esaldi erroredunak (10), automatikoki sortutako esaldi erroredunak (13) eta esaldi zuzenak (3). Automatikoki sortutako esaldi erroredunak dira sistemak entrenamendu garaian ikusitakoak.

Garapeneko corpuseko esaldiekin sistema bakoitza ebaluatzean lortutako emaitzei buruz 6.2 atalean hitz egiten da eta sistema bakoitzak 26 esaldi hauentzako proposatutako zuzenketak C eranskinean aurki daitezke.

5.4.3 Ebaluaziorako corpusa

Ebaluaziorako corpusa entrenamendu guztiak burutu ondoren sistema bakoitzak egiten dituen zuzenketen kalitatea neurtzeko erabili dugun esaldi-multzoa da.

Corpus hau osatzen hasteko abiapuntua errore errealak dituzten esaldiak biltzea izan da. Errore erreal deritzogu kasu errealetan egin diren erroreei, hau da, euskaltegietan egin diren esaldiei adibidez. Guztira kasu errealetan egindako 466 esaldi erroredun bildu ditugu, hurrengo iturrietatik lortuak (informazio hau Maite Oronoz-en doktoretza tesitik [33] eskuratu da):

- Euskaltegietakoak. Euskaltegi hauetako testuak jaso dira, besteak beste: Nafarroako AEK-koak, Trintxerpeko AEK euskaltegikoak, irakasleen trebakuntzarako Irale programakoak eta Donostiako Ilazki euskaltegikoak.
- EuskaraZ zerrenda. EuskaraZ zerrenda[10] 1996an sortu zen eta euskararekin loturiko informazioa trukatzea zuen helburu. Zerrenda horretatik 2000 eta 2001 urteetan idatzitako posta elektronikoko mezuak jaso dira.
- Euskara teknikoa. Euskal Herriko Unibertsitatean euskara teknikoa izeneko irakasgaia irakasten duten irakasle batzuen eskutik jaso dira haien ikasleen lanak.

[10] Informazio gehiaho `http://www.sarean.com/artxiboak/000401.html` helbidean.

- Karrera bukaerako proiektuak. Informatika Fakultateko ikasleek karrera amaitu ahal izateko, proiektu bat garatu eta honi buruzko txosten bat idatzi behar izaten dute. Euskaraz idatzitako hainbat txosten jaso dira erroreen bila. Txostenak zuzendariek zuzendu aurretiko bertsioak dira.

466 esaldi hauetan egindako akatsak errore motaren arabera sailkatu ditugu, mota bakoitzaren proportzioa zenbatekoa den neurtzeko. Corpus honetan aurki daitezkeen erroreen artean errore mota bakoitzaren proportzioa honakoa da:

- Komunztadura-erroreak: %60,96
- Datak: %8,77
- Mendeko esaldiekin zerikusia dutenak: %5,92
- Determinatzailearen erabilera okerra: %5,26
- Postposizio-lokuzioen erabilera okerra: %5,26
- Mugatasuna: %3,07
- Koma: %2,63
- Bestelakoak[11]: %8,11

Errore hauen artean komunztadura-erroreak eta mendeko esaldiekin zerikusia duten erroreak dira zuzentzen zailenak, esaldiaren nolabaiteko analisia egin behar delako errorea identifikatu eta zuzendu ahal izateko. Komunztaduraren kasuan bereziki, komunztadura ondo dagoen jakiteko esaldian aditz laguntzaileko elementu guztiak aztertu behar dira eta euskaraz elipsia dela-eta, ez dute zertan denak agertu. Gainera, absolutibo-plurala eta ergatibo-singularrak forma bera hartzen dute euskaraz ("-ak") eta anbiguotasun horrek zaildu egiten du lana. Hori kontuan hartuta, gure sistementzat esaldi erroredun errealen zuzenketa perfektua egitea ataza zaila izango dela ondorioztatu dezakegu, komunztadura-erroreek eta mendeko esaldiekin egiten diren erroreek errore totalen %65a baino gehiago osatzen baitute.

Errore errealez gain sistemaren portaera errore automatikoekin eta esaldi zuzenekin ere aztertu nahi dugu, horiek baitira entrenamenduan erabili direnak. Hori lortzeko, errore

[11] "Bestelakoak" multzoaren barruan "hauekin guztiekin" eta "guzti hauek" hitz-bikoteen erabilera okerra, "ikusi" hitzarekin zerikusia duten akatsak, "ezer" eta "ezer ez" txarto erabiltzea edo "hobe" eta "hobeto" hitzak nahastean sortzen diren erroreak biltzen dira, besteak beste.

errealen proportzio berdina mantenduz, automatikoki 466 esaldi erroredun sortu ditugu (5.3 atalean azaldutako metodoa erabiliz) eta hasierako corpusetik (EiTB, Elhuyar eta Egunkaria-ko testuak biltzen dituena) 466 esaldi hautatu ditugu ausaz. Ebaluazio corpusak, hortaz, 1.398 esaldi ditu. Corpus honekin lortutako emaitza finalak 6. kapituluko 6.3 atalean azaltzen dira.

5.5 *Byte Pair Encoding*

Entrenamendua hasi aurretik, entrenamendurako corpusa prozesatzeko azken metodo bat erabili dugu: *Byte Pair Encoding* (BPE). Teknika hau baliatu dugu entrenamendurako erabili nahi dugun hiztegiaren tamaina finkatzea ahalbidetzen digulako. Gainera, metodo honi esker ez ditugu ohikoak ez diren hitzak baztertzen. Suposa dezagun gure corpusa hitzetan tokenizatzen dugula eta guztira 10.000 token dituela. Gure hiztegiaren tamaina 8.000 hitzetara mugatu nahiko bagenu, BPE aplikatu gabe 2.000 hitz bertan behera utziko genituzke. BPE aplikatuz, ordea, 10.000 hitzak erabiliko dira, batzuk hainbat tokenetan zatituko diren arren. Jarraian azalduko dugu zatiketa hau nola burutzen den.

BPE teknikan, abiapuntu bezala, corpuseko karaktere bakoitza token bat dela suposatzen da eta ondoren, iterazio bakoitzean, token bikote maizenak elkartzen dira. Algoritmoak iterazioak egiten ditu token kopurua adierazitako hiztegiaren tamainarekin bat etorri arte, gure kasuan 32.000. Hala ere, algoritmoaren funtzionamendua ulertzeko, azter dezagun hiztegi tamaina txikiagoa, 8 adibidez, finkatzen duen kasu zehatz bat. 5.5 taulari erreparatzen badiogu, 0. iterazioan karaktere bakoitza token bat da, guztira 10 token desberdin daudelarik: "t", "h", "e", "m", "a", "n", "i", "s", "v" eta zuriunea. 1. iterazioan "t" eta "h" tokenak elkartzen dira "th" tokena osatuz eta 2. iterazioan "a" eta "n" tokenak batzen dira, "an" tokena eratuz. 3. iterazioan "th" eta "e" tokenak elkartzen dira, prozesua 8 token desberdinekin bukatuz: "the", "m", "an", "i", "s", "n", "v" eta zuriunea.

Iter.	Token bikote maizena	Elkarketa	Esaldia
0			t h e _ m a n _ i s _ i n _ t h e _ v a n
1	t h	th	th e _ m a n _ i s _ i n _ th e _ v a n
2	a n	an	th e _ m an _ i s _ i n _ th e _ v an
3	th e	the	the _ m an _ i s _ i n _ the _ v an

5.5 Taula: BPE teknikaren adibidea.

5.6 Sistema

5.6.1 Konfigurazioa eta Inplementazioa

Zuzentzailearen garapenerako hautatutako arkitektura 4.5 atalean azaldutako Transformer arkitektura izan da. Dagoeneko inplementatutako kodea[12] hartu dugu abiapuntu bezala. Atal honetan helburua ez da Transformer eredua hutsetik inplementatzea izan, dagoeneko inplementatutako kode bat sakonki aztertzea eta ulertzea baizik. Hori dela eta, jatorrizko kodean egin dugun aldaketa bakarra, hiperparametro batzuen balioak egokitzeaz gain, datuen irakurketari dagokio.

Egokitu ditugun hiperparametroak hurrengoak izan dira:

- Hiztegiaren tamaina. 32.000 hitzeko hiztegia finkatu dugu.
- *Batch* tamaina. 4.096 elementuko *batch*-ak definitu ditugu. Tamaina handia ezartzea erabaki dugu exekuzio denbora murrizteko helburuarekin.
- *Epoch* kopurua. Entrenamendua 10 *epoch* egin ondoren gelditzea erabaki dugu.
- Esaldien luzera maximoa. 90 hitz baino gehiagoko testuak baztertu ditugu.

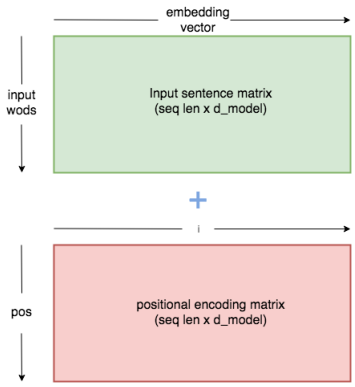
Entrenamendurako datuak irakurtzeko, lehenik eta behin, sortutako corpusak *pandas* erabiliz irakurri ditugu. Ondoren, BPE teknika aplikatu dugu YouTokenToMe[13] tresna erabiliz, hitzen tokenizazioa modu optimoan egiteko. Azkenik, *torchtext* paketeko *data* modulua erabili dugu, datuak *tensor* moduan erabiliko den *dataset* batean gordetzeko. Sistemak *dataset* horretatik jasotzen ditu datuak.

Jarraian, Transformer eredua aztertzearen eta ulertzearen helburua bete dela ziurtatzeko, ereduaren funtzionamendua modu sakonean azalduko dugu.

Aurretik aipatu dugun bezala, Transformer-ak kodetzaile- eta deskodetzaile-pila bana ditu. Gure kasuan, pila horietako bakoitzak 6 elementu edukitzea erabaki dugu, hori baita artikulu originalean proposatzen dena.

Kodetzaile eta deskodetzaile bakoitzak jasotzen duen bektorearen tamaina (d_{model}) 512 da. Lehenengo kodetzailearen eta deskodetzailearen kasuan bektore hau uneko esaldiko

[12] Github-eko esteka: https://github.com/SamLynnEvans/Transformer
[13] Github-eko esteka: https://github.com/VKCOM/YouTokenToMe



5.1 Irudia: Posizioaren araberako kodeketa-matrizea 5.1 eta 5.2 ekuazioek definitutako matrize konstantea da. Hitzen *embedding*-ari gehitzean lortzen den matrizeak hitzen esanahiaren eta posizioaren informazioa gordetzen du.

Iturria: https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec

hitzen *embedding*-ek osatzen dute eta hurrengo kodetzaile eta deskodetzaileen kasuan aurrekoaren irteerak. Hau da, lehenengo kodetzaileak E esaldi jakin baten *embedding*-ak jasotzen ditu, bigarren kodetzaileak lehenengo kodetzailearen irteera, hirugarren kodetzaileak bigarren kodetzailearen irteera, eta abar, horrela seigarren kodetzailera heldu arte (berdina gertatzen da deskodetzaileekin).

Ereduak esaldiak ahalik eta hoberen uler ditzan, hitz bakoitzaren esanahia (*embedding*-ek gordetzen dutena) ezagutzeaz gain, hitz bakoitzak esaldian duen posizioa ere ezagutu behar du. Horretarako *Positional Encoding* edo Posizioaren Araberako Kodeketa erabiltzen da. Hitz bakoitzaren *embedding*-ari hitzaren posizioa edo hitzen arteko distantzia kodetzen duen matrize bat (PE) gehitzen zaio (5.1 irudia). Matrize hau 5.1 eta 5.2 ekuazioetan adierazten den bezala kalkulatzen da, non *pos* hitzaren posizioa den eta *i* aldagaiak *embedding* bektorearen dimentsioan zehar posizioa adierazten duen.

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}}) \tag{5.1}$$

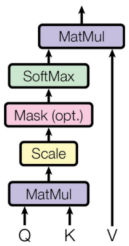
$$PE(pos, 2i+1) = cos(pos/10000^{2i/d_{model}}) \tag{5.2}$$

Sarrerako hitz bakoitzarentzat 3 bektore sortzen dira: *Query* (Q), *Key* (K) eta *Value* (V). *Query* bektoreak uneko hitzaren informazioa gordetzen du eta *Key* eta *Value* bektoreek ikusitako hitz guztien informazioa biltzen dute. 5.2 irudian ikus dezakegu kodetzailean eta deskodetzailean dagoen atentzio-geruzan bektore hauekin jarraitzen den prozesua. Hala ere, atentzioaren funtzionamendua pausuz pausu azaltzeko, 5.3 irudia erabiliko dugu, non bi hitzez osatutako esaldi baten adibidea aztertzen den.

Atentzio-geruza bakoitzean Transformer ereduak hiru pisu-matrize ikasten ditu: W_Q, W_K eta W_V. Matrize hauek eta hitz bakoitzaren *embedding*-a (irudiko x_1 eta x_2) biderkatuz Q, K eta V bektoreak (irudiko q_1, q_2, k_1, k_2, v_1 eta v_2) lortzen dira, ondoren puntuazio edo *score* bat kalkulatzeko erabiltzen direnak. Puntuazio honek uneko esaldiko hitz jakin bat kodetzen den bitartean esaldiko gainontzeko hitzei zenbateko arreta eman behar zaien adierazten du eta Q eta K bektoreen biderketa eskalarra kalkulatuz lortzen da. Adibide honetan esaldiko lehenengo hitzarekin ari gara lanean, beraz, kalkulatzen diren balioak hurrengoak dira: 1. posizioko hitzaren Q eta K (irudiko q_1 eta k_1) bektoreen arteko biderketa eskalarra eta 1. posizioko hitzaren Q eta 2. posizioko hitzaren K (irudiko q_1 eta k_2) bektoreen arteko biderketa eskalarra. Ondoren, gradiente egonkorrak lortzeko, lortutako puntuazioak K bektoreen dimentsioaren erro karratuarekin ($\sqrt{d_k}$) zatitzen dira (irudian erro karratua 8 da K bektoreen dimentsioa 64 delako). Eragiketa horren emaitzari *Softmax* aplikatzen zaio, uneko posizioan hitz bakoitzaren zenbat informazio kodetuko den zehaztuz. Gure adibidean, lehenengo posizioan kodetuko den informazioaren %88a *"Thinking"* hitza izango da eta %12a *"Machines"* hitza. Ondoren *Softmax*-arekin lortutako balioak eta V bektorea biderkatzen dira. Horrela garrantzitsuak diren hitzen balioak mantenduko dira eta gainontzeko hitzak "desagertuko" dira, oso balio txikiekin biderkatzen direlako. Azkenik, lortutako bektoreak (irudian v_1 eta v_2) gehitzen dira, uneko posizioko atentzio-geruzaren irteera lortuz.

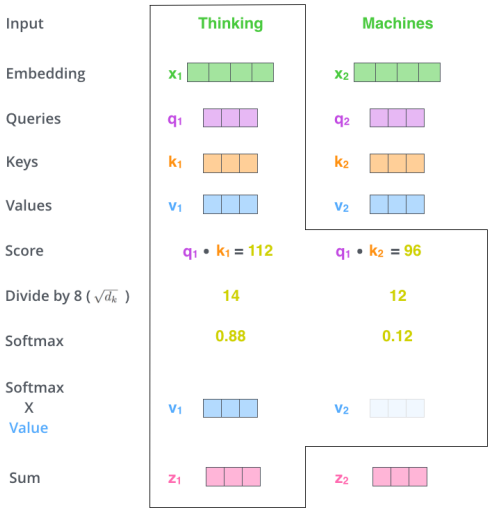
Inplementazioan, eragiketa hauek bektoreekin egin ordez, matrizeekin egiten dira. Matrizeak lortzeko *embedding* guztiak X matrize batean elkartzen dira eta matrize hori W_Q, W_K eta W_V matrizeekin biderkatzen da, horrela Q, K eta V matrizeak lortuz. Matrizeekin lan eginez, azaldu berri dugun prozesua ekuazio bakarrean (5.3 ekuazioa) bildu daiteke.

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5.3}$$



5.2 Irudia: Atentzioan jarraitzen den prozedura orokorra.

Iturria: A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.Ñ. Gomez, Ł. Kaiser,and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems.



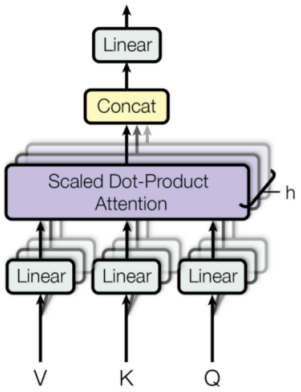
5.3 Irudia: Atentzioan jarraitzen den prozedura 2 hitzeko esaldi baten kasuan.

Iturria: http://jalammar.github.io/illustrated-transformer/

Aipatzekoa da atentzio-geruzan "Buru-Anitzeko" atentzio-mekanismoa erabiltzen dela (5.4 irudia). Honek zera esan nahi du: azaldu berri dugun atentzioaren prozedura hainbat aldiz egiten dela pisu-matrize desberdinekin. Buru-anitzekin lan egitearen arrazoia zera da: adierazgarritasun-maila desberdineko erlazioak antzematen direla. Adibidez, atentzio-buru batzuek arreta hurrengo hitzetan jartzen duten bitartean, beste buru batzuek aditz eta objektu-zuzenen arteko harremana nabarmenduko dute [34]. Jatorrizko artikuluan 8 atentzio-buru erabiltzea proposatzen da.

8 atentzio-bururekin lan egiteak atentzio-geruzaren irteera 8 matrize izango direla esan nahi du eta geruza honen ondoren dagoen aurreranzko barreiatze-sarea matrize bakarra (hitz bakoitzeko bektore bat) jasotzeko prestatuta dago. Arazo honi aurre egiteko matrize guztiak kateatzen dira eta ereduak ikasi beharreko laugarren matrize batekin (W_O) biderkatzen dira. Horrela buru guztien informazioa bateratzen duen matrize bakarra lortzen da, aurreranzko barreiatze-sareari pasatzen zaiona.

Kodetzaile eta deskodetzaile bakoitzeko elementuek, hau da, atentzioak eta aurreranzko barreiatze-sareak, hondar-konexio bat daukate, normalizazio pausu batek jarraitua. Atentzioak jasotzen duen sarrera-matrizea eta atentzioaren irteera-matrizea gehitu eta normalizatzen dira aurreranzko barreiatze-sarera (edo hurrengo atentziora, deskodetzailearen kasuan) pasatu aurretik eta, era berean, aurreranzko barreiatze-sarearen sarrera- eta irteera-matrizea gehitu eta normalizatzen dira pilako hurrengo kodetzaile edo deskodetzailera pasatu aurretik. 5.5 irudian ikusi daiteke prozesu hau kodetzaile baten kasuan, 5.3 irudiko adibidearekin jarraituz.



5.4 Irudia: Buru-Anitzeko Atentzio mekanismoa.

Iturria: A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.Ñ. Gomez, Ł. Kaiser,and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems.

Kodetzaile-pilako azken elementuak esaldi bat prozesatu ostean, lortzen diren K eta V matrizeak deskodetzaile bakoitzeko bigarren atentzioari[14] pasatzen zaizkio, deskodetzailea sarrera-esaldiko elementu garrantzitsuetan arreta jartzeko gai izan dadin. Deskodetzaileak irteera bezala ematen duen hitz bakoitza hurrengo pasuan sarrera bezala jasotzen du, pausuz pausu esaldia osatuz, irteera bezala *end of sentence* tokena lortu arte.

Inplementazioan landutako beste atal bat maskarak izan dira. Matrizeei maskarak gehitzeak bi helburu ditu. Lehenengoa, kodetzailean zein deskodetzailean, sarrera-esaldiko *padding*[15] elementuetan arretarik ez jartzeko. Bigarrena, deskodetzaileak hitz bat iragartzerakoan, helburu-esaldiko hurrengo hitzak begiratzea saihesteko (hitz bat iragartzeko deskodetzaileak kodetzaileko irteerak eta iragartzen ari den hitzaren aurretik dauden hitzak besterik ezingo ditu erabili).

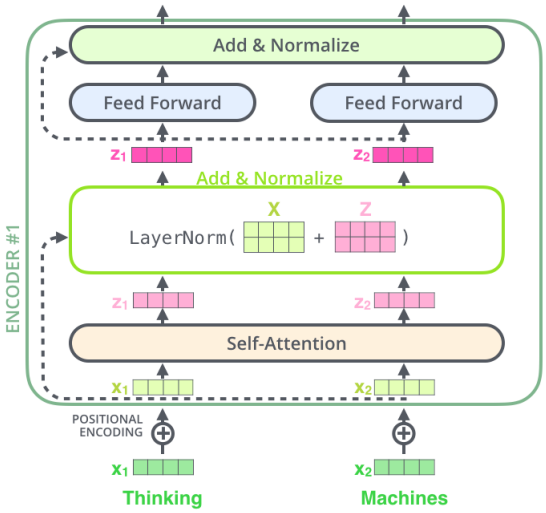
Pausu bakoitzean iragartzen den hitza zein den erabakitzeko, deskodetzaile-pilaren irteera bektorea *Linear* geruza batetik pasatzen da, hiztegiaren tamainako bektore bat[16] ematen duena. Bektore horretako posizio bakoitza hiztegiko hitz bati dagokio. Ondoren, bektorea *Softmax* geruza batetik pasatzen da, horrela bektoreko posizio bakoitzeko balioa probabilitate bihurtuz. Probabilitate handieneko posizioari dagokion hitza da deskodetzaileak iragartzen duena[17].

[14] Atentzio honek buru-anitzeko atentzioak bezala lan egiten du, baina Q matrizea aurreko geruzatik jasotzen du eta K eta V matrizeak kodetzailetik.

[15] Sare neuronal batek jasotzen dituen ezaugarri- eta datu-bektore guztiek luzera berdina izan behar dute. *Padding* deritze finkatutako luzera baino laburragoak diren bektoreei gehitzen zaizkien betetze-elementuei.

[16] Mota honetako bektoreei *logits vector* deitzen zaie.

[17] Azalpen hau egoera sinpleenari dagokio. Inplementazioan probabilitate handieneko tokena soilik hautatu ordez, hainbat token hautatzen dira pausu bakoitzean. Prozesu hau 5.6.3 atalean azaltzen da.



5.5 Irudia: Hondar-konexioak eta normalizazioa kodetzailearen kasuan.

Iturria: http://jalammar.github.io/illustrated-transformer/

5.6.2 Entrenamendua

Tranformer-aren funtzionamendua azaldu ondoren, azter dezagun nola jarduten duen entrenamendu garaian W_Q, W_K, W_V eta W_O matrizeak ikasteko. Gainbegiratutako ikasketa lantzen ari garenez, hau da, esaldi bakoitzaren zuzenketa ezagutzen dugunez, desiratutako zuzenketa hori eta sistemak lortzen duen zuzenketa alderatu ditzakegu. Entrenamenduko iterazio bakoitzean, lortu nahi diren zuzenketen eta sistemak lortu dituen zuzenketen arteko galera kalkulatzen da eta matrizeen pisuak eguneratzen dira, galera hori minimizatzen saiatzeko.

Gure kasuan, galera kalkulatzeko, *Pytorch* liburutegiko *cross_entropy* funtzioa erabili dugu. Funtzio honek *Softmax* funtzioaren logaritmoa eta probabilitate-galera logaritmiko negatiboa (ingelesez *negative log likelihood loss* edo NLL) bateratzen ditu. Beste era batera esanda, lehenengo Transformer-aren irteeraren *Softmax* probabilitateen logaritmoak kalkulatzen dira eta, ondoren, logaritmo horiek erabiliz, galera kalkulatzen da probabilitate-galera logaritmiko negatiboaren irizpidea jarraituz.

Probabilitate-galera logaritmiko negatiboak nola funtzionatzen duen azaltzeko, demagun gure ereduaren irteera [0,3 , 0,1 , 0,5 , 0,1] probabilitateak direla. Helburu-irteera (kasu optimoan ereduak iragarri beharko lukeena) [0 , 0 , 0 , 1] bada, NLL 5.4 ekuazioan adierazten den bezala kalkulatzen da. Ordea, helburu-irteera [0 , 0 , 1 , 0] bada, 5.5 ekuazioan islatzen den emaitza lortzen da. Bi kasuetan lortutako balioak alderatuz, iragarpena geroz eta hobea izan ahala NLL galera baxuagoa dela ikus dezakegu. Galera geroz eta

baxuago izateak zera adierazten du: ereduaren parametroak (ikasi behar diren matrizeen pisuak) datuen ezaugarriak bereganatzeko eta datu horien distribuzioa jarraitzen duten datuak (gura kasuan zuzenketa egokiak) sortzeko gai direla.

$$NLL = -ln(0 \cdot 0,3 + 0 \cdot 0,1 + 0 \cdot 0,5 + 1 \cdot 0,1) = -ln(0,1) = 2,3 \tag{5.4}$$

$$NLL = -ln(0 \cdot 0,3 + 0 \cdot 0,1 + 1 \cdot 0,5 + 0 \cdot 0,1) = -ln(0,5) = 0,69 \tag{5.5}$$

5.4.1 atalean diseinatutako datu-multzo bakoitzarekin entrenamendu bana burutu dugu, horrela entrenatutako 4 sistema desberdin lortuz. Sistemak entrenatu ahal izateko beharrezkoa izan da IXA taldearen GPU-ak erabiltzea, esku tartean genuen datuen tamaina dela eta. Kasu guztietan entrenamendua 10 *epoch* egin ondoren moztu dugu, alde batetik momentu horretara heldutakoan galera nahiko egonkorra delako, eta bestetik, datu-multzo handiekin entrenatzeak suposatzen duen denbora-kostua dela eta. 4.096 ezarri dugu *batch* tamaina bezala, aldi berean ahalik eta esaldi gehien prozesatu ahal izateko, eta hala ere, entrenamendu-denbora oso luzea behar izan dugu: datu-multzo txikienarekin, 1,6 milioi esaldi ingurukoa, 735 minutu (12 ordu eta 15 minutu) behar izan dira sistema entrenatzeko; 10 milioi inguruko datu-multzoen kasuan entrenamendu-denbora 3.389 minutura (2 egun, 8 ordu eta 29 minutu) igotzen da.

5.6.3 Sorkuntza

Entrenamenduaren ondoren sistema bakoitzaren funtzionamendua frogatu dugu, hau da, euskarazko esaldien zuzenketa egin dugu. Helburua ahalik eta zuzenketa hoberenak lortzea da. Gure kasuan, zuzenketa hoberena probabilitate altuena duen irteera-esaldia da.

Probabilitate handieneko esaldia aurkitzeko, esaldi posible guztiak sortu beharko lirateke (hiztegi bateko hitz guztien konbinazio guztiak eginez) eta ondoren esaldi horietako bakoitza ebaluatu beharko litzateke probabilitate handienekoaren bila. Esaldi posible guztiak sortzea eta aztertzea NP problema bat da, hau da, ez dakigu problema modu zehatzean eta denbora polinomikoan ebazteko algoritmorik existitzen den. Hori dela eta, bilaketa-heuristikoko algoritmo bat erabili dugu, emaitza hoberena aurkituko dela ziurtatzen ez duten arren, modu eraginkorrean emaitza onak lortzeko gai diren algoritmoak baitira.

Gure kasuan, sarrera-esaldi bat emanda bere zuzenketa sortzeko, *Beam Search* algoritmoa erabiltzea erabaki dugu. 5.6.1 atalean esandakoa kontuan hartuta, pausu bakoitzean deskodetzaileak probabilitate altueneko hitza iragartzen badu, intuizioak dio esaldi osoa sortu dezakegula iterazio bakoitzean probabilitate altueneko tokena hautatzen badugu,

esaldiaren amaierara heldu garela adierazten duen tokena lortu arte. Pausu bakoitzean probabilitate altuena duen hitza soilik kontuan hartzeak, ordea, ez du zertan erabaki optimoa izan esaldi oso bat eraikitzen ari bagara. *Beam Search* algoritmoak, baldintzapeko probabilitatean oinarrituta, esaldia osatzeko hainbat hitz hautatzen ditu pausu bakoitzean. Iterazio bakoitzean aukeratzen den hitz kopurua *beam width* izeneko parametro baten bidez zehazten da; gure inplementazioan 4 balioa esleitu diogu.

Demagun ingelesezko *"I am to visit my parents"* esaldi erroreduna zuzenduna nahi dugula, 100 tokeneko hiztegi batekin ari garela lanean eta *beam width* parametroari 4 balioa eman diogula. Azter dezagun pausuz pausu deskodetzailearen sorkuntza prozesua.

Lehenengo pausuan hiztegiko 100 tokenen artean probabilitate handiena lortu duten 4 tokenekin geratuko gara. Suposa dezagun 4 token horiek *"I"*, *"My"*, *"We"* eta *"Me"* hitzak izan direla (5.6 ekuazioa, non x sarrera-esaldiko informazioa den).

$$P(y_1|x) = [I, My, We, Me] \tag{5.6}$$

Bigarren pausuan 4 bilaketa egingo dira, hitz horietako bakoitza esaldiko lehenengo hitza dela suposatuz, probabilitate handieneko hurrengo hitza zein den topatzeko. Kasu bakoitzean kalkulatuko diren probabilitateak 5.7 ekuazio-multzoan ikus daitezke. Bilaketa bakoitzean 100 probabilitate kalkulatzen direnez (hiztegiko token bakoitzarentzat probabilitate bana), guztira 400 probabilitate kalkulatuko dira.

$$\begin{aligned} &P(y_2|x, I) \\ &P(y_2|x, My) \\ &P(y_2|x, We) \\ &P(y_2|x, Me) \end{aligned} \tag{5.7}$$

400 probabilitate horien artean 4 altuenak hautatuko dira. Suposa dezagun probabilitate altuena lortu duten hitz-bikoteak *"I am"*, *"I will"*, *"My parents"* eta *"We will"* direla. Ikus daitekeenez, lehenengo pausuan probabilitate altua lortu duen *"Me"* hitza ez da bigarren pausuan aukera optimoen artean agertu eta, ondorioz, esaldi zuzenduaren lehenengo hitz bezala baztertuko dugu.

Hirugarren pausuan prozesua errepikatuko da, oraingoan probabilitate altueneko 4 hitzbikoteak ezagututa, probabilitate altueneko hirugarren hitza bilatuz (5.8 ekuazio-multzoa).

$$\begin{aligned} &P(y_3|x, I\ am) \\ &P(y_3|x, I\ will) \\ &P(y_3|x, My\ parents) \\ &P(y_3|x, We\ will) \end{aligned} \tag{5.8}$$

Prozesu honekin jarraituta, amaieran 4 esaldi lortuko ditugu. Demagun hurrengo 4 esaldiak lortu ditugula, probabilitate altuena duenetik probabilitate baxuena duenera ordenatuta: *"I am visiting my parents."*, *"I am going to visit my parents."*, *"I will visit my parents."* eta *"I will go visit my parents."*. 4 esaldiak gramatikalki zuzenak diren arren, probabilitate handiena lortu duena *"I am visiting my parents."* izan da eta, hortaz, hori izango da sistemak proposatuko duen zuzenketa.

6. KAPITULUA

Ebaluazioa eta Emaitzak

Kapitulu honetan gure sistemak sortutako zuzenketen kalitatea neurtzeko erabilitako metrikak azaltzen dira. Jarraian, corpus desberdinak diseinatzeko gidalerro bezala erabili den eskuzko ebaluazioaren inguruan jarduten da. Azkenik, ebaluaziorako corpusean lortutako emaitzei buruz hitz egiten da.

6.1 Ebaluazio-metrikak

6.1.1 ERRANT

Gure sistema ebaluatzeko 2019an ospatutako *BEA Shared Task* [23] txapelketan (aurretik 3.2.2 atalean aipatutakoa) parte-hartzaileen sistemak sailkatzeko erabilitako metodoa aztertu dugu. Lehiaketara aurkeztutako sistemak ebaluatzeko *Error Annotation Toolkit* edo ERRANT tresna[1] erabiltzen da. Tresna honek esaldi erroredun bat eta dagokion zuzenketa alderatzen ditu eta zuzenketan proposatutako aldaketak anotatzen ditu. ERRANT-ek anotazioak biltzen dituen fitxategi bat sortzen du, non esaldi erroredun-zuzen bikote bakoitzerako "S" eta "A" eremuak aurki daitezkeen. "S" letrarekin hasten diren lerroetan esaldi erroreduna adierazten da. "A" letrarekin hasten diren lerroetan esaldi erroredunerako proposatutako zuzenketak anotatzen dira. "A" motako lerro bakoitzean hainbat balio adierazten dira, horien artean errore mota eta anotatzailearen identifikadorea, adibidez.

[1] Github-eko esteka: https://github.com/chrisjbryant/errant

Esaldi erroreduna	Esaldi zuzena	Zuzenketa automatikoa
mahaia hurbil dira	mahaia hurbil dago	mahaia hurbil dago
umeak triste dago	umeak triste daude	umeak pozik daude
ni oporretan joango da	ni oporretan joango naiz	aulkia hurbil dago

6.1 Taula: Esaldi erroredunen, esaldi zuzenen eta sistemak proposatutako zuzenketen adibideak.

Guk kontuan hartu beharrekoak "A" lerro bakoitzeko lehenengo eta hirugarren posizioan dauden balioak dira, zuzenketaren posizioa (zuzenketa zein tokenetan hasten den eta zein tokenetan amaitzen den adierazten da zenbaki bidez) eta proposatutako zuzenketa, hurrenez hurren. "S" esaldi erroredun batek "A" zuzenketa bat baino gehiago izan ditzake.

Demagun 6.1 taulako esaldiekin ari garela lanean. 6.1a irudian ERRANT-ek esaldi erroredunak eta zuzenak alderatuz egindako anotazioa ikus dezakegu eta 6.1b irudian esaldi erroredunak eta sistemak emandako zuzenketak alderatuz egindakoa. Azter dezagun bigarren esaldiaren kasua. "S" esaldi erroreduna "umeak triste dago" da. 6.1a irudian esaldi zuzenarekin, "umeak triste daude", egiten da konparazioa. Anotazioan ikus daitekeenez, zuzenketa 2. tokenetik hasita 3. tokenera egiten da, hau da, 2. tokena ordezkatzen da, eta proposatutako zuzenketa "daude" da. Esaldi erroreduneko 0. tokena "umeak" da, 1. tokena "triste" eta 2. tokena "dago". Beraz, zuzenketa hurrengoa da: "dago" ordez "daude" erabili. 6.1b taulako anotazioan esaldi erroreduna "umeak pozik daude" esaldiarekin alderatzen da, sistemak proposatutako zuzenketarekin hain zuzen ere. Kasu honetan bi anotazio ditugu: 1. tokena ("triste") "pozik" tokenarekin ordezkatzea eta 2. tokena ("dago") "daude" tokenarekin ordezkatzea.

6.1 irudiko bi fitxategiak lortu ondoren ERRANT-ek *Span-based Correction* irizpidea eta F0,5 metrika erabiliz ebaluatuko ditu.

Span-based Correction zer den azaltzeko 6.2 irudia erabili dezakegu. Kasu horretan, esaldi erroreduna "*I often look at TV*" da eta lortu nahiko genukeen zuzenketa [2, 4, *watch*]

```
S mahaia hurbil dira
A 2 3|||R:NOUN|||dago|||REQUIRED|||-NONE-|||0

S umeak triste dago
A 2 3|||R:NOUN|||daude|||REQUIRED|||-NONE-|||0

S ni oporretan joango da
A 3 4|||R:OTHER|||naiz|||REQUIRED|||-NONE-|||0
```

(a) Esaldi erroreduna vs esaldi zuzena.

```
S mahaia hurbil dira
A 2 3|||R:NOUN|||dago|||REQUIRED|||-NONE-|||0

S umeak triste dago
A 1 2|||R:NOUN|||pozik|||REQUIRED|||-NONE-|||0
A 2 3|||R:NOUN|||daude|||REQUIRED|||-NONE-|||0

S ni oporretan joango da
A 0 1|||R:OTHER|||aulkia|||REQUIRED|||-NONE-|||0
A 1 2|||R:OTHER|||hurbil|||REQUIRED|||-NONE-|||0
A 2 4|||R:OTHER|||dago|||REQUIRED|||-NONE-|||0
```

(b) Esaldi erroreduna vs zuzenketa automatikoa.

6.1 Irudia: ERRANT-ek proposatutako anotazioak.

Original	I often look at TV	**Span-based**	**Span-based**	**Token-based**
Reference	[2, 4, watch]	**Correction**	**Detection**	**Detection**
Hypothesis 1	[2, 4, watch]	Match	Match	Match
Hypothesis 2	[2, 4, see]	No match	Match	Match
Hypothesis 3	[2, 3, watch]	No match	No match	Match

6.2 Irudia: *Span-based Correction*, *Span-based Detection* eta *Token-based Detection* irizpideen konparazioa.

Iturria: https://www.cl.cam.ac.uk/research/nl/bea2019st/#eval

da, esaldi zuzena "*I often watch TV*" dela adierazten duena. *Token-based Detection* irizpidea betetzeko ordezkatu beharreko tokenaren hasiera asmatu behar da, hau da, aldaketa 2. posizioan hasten dela antzeman. *Span-based Detection* betetzeko aldaketaren posizio zehatza lortu behar da, hau da, 2. tokenetik 4. tokenera arte ordezkatu behar dela zehaztu. *Span-based Correction* (ERRANT-ek zuzenketen kalitatea neurtzeko erabiltzen duena) gainditzeko, posizio zehatza asmatzeaz gain token zehatza asmatu beharra dago, hau da, lortu nahi den zuzenketa eta sistemak proposatutakoa berdin-berdinak izan behar dira.

ERRANT tresnak, *Span-based Correction* irizpidea jarraituz, 6.2 taulan ikusten diren datuen balioak kalkulatzen ditu anotatutako bi fitxategiekin (balio hauek zehazki 6.1 irudiko fitxategiei dagozkie).

Lehenengo lortutako *True Positive* (TP), *False Positive* (FP) eta *False Negative* (FN) kopurua adierazten da. 6.1 irudiko fitxategien kasuan 2 TP dauzkagu, hau da, asmatu diren kasuak: lehenengo esaldian "dago" jartzea "dira" ordez eta bigarren esaldian "daude" jartzea "dago" ordez. 4 FP kasuak zuzentzaile automatikoak proposatu dituenak baina benetako zuzenketan ez daudenak dira: bigarren esaldian "triste" ordez "pozik" jartzea, hirugarren esaldian "ni" ordez "aulkia" jartzea, hirugarren esaldian "oporretan" ordez "hurbil" jartzea eta hirugarren esaldian "joango da" ordez "dago" jartzea. FN zuzenketa errealean egiten diren baina sistemak antzeman ez dituen zuzenketak dira, gure kasuan hirugarren esaldian "da" ordez "naiz" erabiltzea (sistemak ez du horrelako zuzenketarik proposatu).

TP	FP	FN	Prec	Rec	F0,5
2	4	1	0,3333	0,6667	0,3704

6.2 Taula: ERRANT tresna 6.1 irudiko fitxategiekin erabiliz lortutako irteera.

Ondoren doitasuna (ingelesez *precision (prec)*) eta estaldura (ingelesez *recall (rec)*) erakusten dira, 6.1 eta 6.2 ekuazioak erabiliz kalkulatzen direnak, hurrenez hurren. Doita-

sunak sistemak proposatutako zuzenketa guztien artean egokiak zenbat diren neurtzen du eta estaldurak egin beharreko zuzenketa guztien artean sistemak zenbat egin dituen.

$$Doitasuna = \frac{TP}{TP+FP} \tag{6.1}$$

$$Estaldura = \frac{TP}{TP+FN} \tag{6.2}$$

Azkenik, F0,5 metrikarekin lortutako puntuazioa erakusten da. F0,5 metrikak doitasuna estaldura baino bi aldiz gehiago puntuatzen du eta 6.3 ekuazioan adierazten den bezala kalkulatzen da. Balio hau da *BEA Shared Task* txapelketan parte-hartzaileen *ranking*-a egiteko erabili dena eta guk gure sistemaren kalitatea neurtzeko erabili duguna.

$$F_{0,5} = (1+0,5^2) \cdot \frac{doitasuna \cdot estaldura}{(0,5^2 \cdot doitasuna) + estaldura} \tag{6.3}$$

6.1.2 GLEU

Generalized Language Evaluation Understanding edo GLEU errore gramatikalen zuzenketa ebaluatzeko erabiltzen den metrika[2] bat da [35]. *Bilingual Evaluation Understudy* edo BLEU metrikaren ([36]), gaur egun itzulpengintza-automatikoa ebaluatzeko erabiltzen den metodo nagusiaren aldaera bat da.

BLEU metrikak automatikoki sortutako itzulpenen kalitatea neurtzen du, puntuazio altuena giza-itzulpenetik hurbilen dauden itzulpen automatikoek jasotzen dutelarik. Honek zera esan nahi du: itzulpen-automatikoko atazetan BLEU puntuazioa kalkulatzeko sarrera-esaldia (itzuli nahi dena) ez dela beharrezkoa. Zuzenketa gramatikalaren kasuan, ordea, ez da gauza bera gertatzen; itzulpen-automatikoan sarrera-esaldiko hitz bat ez aldatzeak kasu gehienetan errorea suposatzen duen bitartean, zuzenketa gramatikalean sarrera-esaldiko hitz batzuk besterik ez dira aldatu behar. Sarrera-esaldia ez denez kontuan hartzen, zuzenketa automatikoa eta giza-zuzenketa hizkuntza berean idatzita daudenez eta esaldiko hitz gehienak ez direnez aldatu behar, normalean zuzenketa gramatikal automatikoek BLEU puntuazio oso altua jasotzen dute. Arazo horri aurre egiteko, GLEU metrikan, BLEU metrikan ez bezala, sarrera-esaldia ere hartzen da kontuan eta n-gramen doitasuna eraldatzen da pisu gehiago emateko zuzenketa automatikoan zein giza-zuzenketan aurki

[2]GLEU erabiltzeko Github-eko esteka: https://github.com/cnap/gec-ranking

	Entrenamendurako erabilitako corpusa			
	err_zuz400K	err_zuz5M	err_zuz8M	err+_zuz8M
TP	5	**11**	10	10
FP	39	18	**14**	23
FN	29	**23**	24	24
Doitasuna (prec)	0,1136	0,3793	**0,4167**	0,303
Estaldura (rec)	0,1471	**0,3235**	0,2941	0,2941
F0,5	0,119	0,3667	**0,3846**	0,3012

6.3 Taula: Entrenatutako sistema bakoitzak garapenerako esaldiei egindako zuzenketetan jasotako ERRANT ebaluazioa.

daitezkeen baina jatorrizko-esaldian ez dauden n-gramei eta zuzenketa automatikoan zein jatorri-esaldian agertzen diren baina giza-zuzenketan aurkitu ezin daitezkeen n-gramak zigortzeko.

6.2 Garapenerako ebaluazioa

Ebaluazio hau 5.4.1 atalean aipatutako ebaluazioari dagokio eta entrenamendurako corpus desberdinak diseinatzeko gakoa izan da. Sistema bakoitza entrenatu ondoren 26 esaldi tan (5.4.2 atalean definitutako corpusa) ebaluatu dugu ERRANT erabiliz eta lortutako F0,5 puntuazioari erreparatuz erabaki dugu sortu beharreko hurrengo corpusaren ezaugarriak zeintzuk izan beharko liratekeen. Beraz, ebaluazio hau proiektuaren diseinu eta inplementazio osoan zehar burutu da, iterazioak eginez, aurreko sistemarekin lortutako emaitzak hobetu nahian.

Entrenatutako sistema bakoitza probarako 26 esaldiak zuzentzeko erabili ondoren[3], lortutako zuzenketak eta zuzenketa errealak ERRANT erabiliz anotatu ditugu eta puntuazioak kalkulatu ditugu (6.3 taula). Azter ditzagun puntuazio horiek 5.4.1 atalean corpus berriak sortzeko aipatu ditugun arrazoiak berresteko.

err_zuz400K datu-multzoarekin entrenatutako sistemarekin lortutako emaitzak ikusita argi dago kaskarrak direla: 34 akatsetik 5 zuzentzea besterik ez da lortu eta gainera beharrezkoak ez diren 39 zuzenketa proposatu dira. Hau argi eta garbi islatzen da F0,5 puntuazioan, 0,119 oso balio baxua baita. Lehenengo corpusak 1,6 milioi esaldi dituenez, emaitza hauek hobetzeko aukera bat esaldi kopurua handitzea izango dela susmatu dugu.

[3] 26 esaldi hauetan sistema bakoitzak egindako zuzenketa B eranskinean ikus daiteke.

Ebaluatutako hurrengo sistema err_zuz5M datu-multzoarekin entrenatutakoa izan da, kasu honetan erabili den esaldi kopurua 6,5 milioi baino pixka bat gehiago delarik. Sistema honekin emaitzak hobetu dira, 34 zuzenketatik 11 asmatuz eta beharrezkoak ez diren zuzenketa askoz gutxiago, 18 zehazki, proposatuz. Horrela 0,3667 F0,5 puntuazioa lortu da. Entrenamendurako esaldi kopurua handitzea bide zuzena dela ikusita, esaldi gehiago gehitzea erabaki dugu.

Burututako hurrengo ebaluazioa entrenamendurako datu-multzo bezala err_zuz8M corpusa (10 milioi esalditik hurbil dagoena) duen sistemarena izan da. Oraingoan asmatutako zuzenketa kopurua antzekoa mantendu da (10 asmatu dira, aurreko sistemarekin baino 1 gutxiago) eta beharrezkoak ez diren zuzenketa kopurua jaitsi da. Emaitza hauekin F0,5 puntuazioa pixka bat igotzea lortu da, 0,3846 baliora arte.

Azkenik err+_zuz8M datu-multzoarekin entrenatutako sistema ebaluatu da, lehenengo sistemaren emaitzak hobetzeko beste proposamen bat jarraitzen duena: entrenamendurako corpusean automatikoki sortutako esaldi erroredunen errore-kopurua handitzea. Hala ere, hirugarren sistemarekin puntuazio altuena lortu denez, sistema hau entrenatzeko erabilitako esaldi kopurua ere 10 milioitik hurbil dago. Kasu honetan ere egin beharreko 34 zuzenketetatik 10 asmatu dira baina beharrezkoak ez diren zuzenketak 23ra igo dira. Horrela bada, laugarren sistemaren F0,5 puntuazioa 0,3012 baliokoa da; lehenengo sistemarena baino dezente hobea, baina ez bigarren eta hirugarren sistemen parekoa.

Emaitza hauek guztiak ikusita, sistemak sortutako zuzenketen kalitatea igotzea entrenamendurako esaldi kopuruaren menpekoa dela ondorioztatu dugu, esaldiko errore kopurua handitzea oso lagungarria ez izatearekin batera. Hirugarren sistemak lortu du puntuazio altuena, entrenamendurako corpusean 10 milioi esaldi dituena eta erroreak sortzeko metodo sinplea erabiltzen duena. Euskarazko corpus zuzen gehiago ez dugunez, ezin dugu entrenamendurako esaldien kopurua handitzen jarraitu eta, beraz, zuzenketa gramatikal hoberenak hirugarren sistemarekin lortzen direla esango dugu.

6.3 Ebaluazio finala eta Emaitzak

Sistema guztiak entrenatu eta bakoitzarekin hasierako ebaluazio sinple bat (6.2 atalean aipatutakoa) egin ostean, ebaluazio finala egitera pasa gara. Ebaluazio honetan 5.4.3 atalean aurkeztutako corpusa erabili dugu gure sistemek sortzen dituzten zuzenketen kalitatea neurtzeko.

	Automatikoki sortutako esaldi erroredunak					
	TP	**FP**	**FN**	**Prec**	**Rec**	**F0,5**
err_zuz400K	104	1227	631	0,0781	0,1415	0,0858
err_zuz5M	159	247	576	0,3916	0,2163	0,337
err_zuz8M	168	**140**	567	**0,5455**	0,2286	**0,427**
err+_zuz8M	**184**	201	**551**	0,4779	**0,2503**	0,4044

6.4 Taula: Automatikoki sortutako esaldi erroredunetan sistema bakoitzak lortutako emaitzak, ERRANT metrika erabiliz neurtuak.

Lehenik eta behin ebaluazio-corpuseko automatikoki sortutako 466 esaldi erroredunekin egin dugu saiakera, gure sistemak ere automatikoki sortutako esaldi erroredunekin entrenatu direnez, zuzenketa hoberenak esaldi hauekin lortu beharko liratekeelako.

Sistema bakoitza esaldi hauekin frogatu ostean lortutako ERRANT metrikaren balioak 6.4 taulan ikusi daitezke (lehenengo zutabean adierazten dena sistema bakoitza entrenatzeko erabilitako corpusa da).

Ikus dezakegunez, 6.2 atalean egindako ebaluazioan lortutako emaitzekin planteatutako hipotesia bete da, zuzenketa hoberenak hirugarren sistemak egiten dituela hain zuzen ere. Kasu honetan, zuzendu beharreko 735 erroreetatik 168 zuzentzea lortu da eta gainontzeko sistemek baino askoz zuzenketa oker gutxiago proposatu dira.

Hala ere, nabarmentzekoa da laugarren sistemaren portaera, hirugarren sistemarekiko aldea 0,02 puntukoa besterik ez baita izan. Sistema hau izan da egin beharreko zuzenketa gehien asmatu dituena, 735etik 184 zehazki.

Automatikoki sortutako esaldiekin proba egin ondoren, gainontzeko esaldietan sistemen portaera ere berdintsu mantentzen ote den aztertu dugu.

Puntu honetara helduta, garrantzitsua da ERRANT-en berezitasun bat aipatzea. Tresna hau erroreak anotatzeko prestatuta dagoenez, errorerik ez dagoen kasuetan ez du berdin funtzionatzen. Badakigu ERRANT-ek anotazioak egiteko sarrera-esaldia eta esaldi horretarako proposatutako zuzenketa alderatzen dituela. Sistemak sarrera-esaldi bat zuzentzat jotzen duenean ez dio inolako aldaketarik egingo eta, beraz, sarrera- eta irteera-esaldiak berdinak izango dira. Bi esaldiak berdinak direnean anotazioak honako itxura dauka:

```
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0
```

Errorearen mota "noop" izateaz gain errorea -1 tokenean hasi eta -1 tokenean bukatzen dela adierazten da, hau da, errorerik ez dagoela.

Sistemak ondo ikasi badu, zuzenak diren 466 esaldietarako ez luke zuzenketarik proposatu beharko eta, ondorioz, anotazio guztiak mota honetakoak izango lirateke. Kasu horretan 6.3 irudian ikusten diren itxurako bi fitxategi edukiko genituzke eta ERRANT-ek 1eko F0,5 puntuazioa emango luke (6.5 taula).

```
S batzuk kanoetan joan ginen , beste batzuk
basotik zehar .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S eta telebista autonomikoaren proiektua martxan
dagoenez , horrek du lehentasuna " , adierazi du
nuria iturriagagoitia bozeramaileak .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0
```

(a) Esaldi erroreduna vs esaldi zuzena.

```
S batzuk kanoetan joan ginen , beste batzuk
basotik zehar .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S eta telebista autonomikoaren proiektua martxan
dagoenez , horrek du lehentasuna " , adierazi du
nuria iturriagagoitia bozeramaileak .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0
```

(b) Esaldi erroreduna vs zuzenketa automatikoa.

6.3 Irudia: ERRANT-ek proposatutako anotazioak sistemak sarrerako-esaldi guztiak zuzenak direla antzematen duen kasuan.

TP	FP	FN	Prec	Rec	F0,5
0	0	0	1,0	1,0	1,0

6.5 Taula: ERRANT-ekin lortutako balioak 6.3 irudiko fitxategiak ebaluatzean.

Hala ere, badakigu gure sistemak ez dituela kasu guztietan zuzenketa perfektuak sortzen eta, beraz, normala da esaldi batzuen kasuan beharrezkoak ez diren zuzenketak proposatzea. Demagun 6.4 irudiaren kasuan gaudela, non sistemari dagoeneko zuzenak diren 4 esaldi zuzentzea eskatu zaion. Ikus daitekeenez, 3 esaldi zuzenak direla antzeman du eta ez du aldaketarik proposatu, baina bigarren esaldian bi errore daudela adierazi du. Bi fitxategiak ERRANT erabiliz ebaluatzen baditugu (6.6 taula), faltsu positibo hauek bakarrik zenbatzen dira. Zuzenak direla antzeman diren esaldien kasua "noop" bezala anotatzen denez ez dira TP bezala zenbatzen eta, ondorioz, lortzen den F0,5 puntuazioa 0 da.

Hau kontuan hartuta, ezin izango dugu F0,5 puntuazioa erabili esaldi zuzenen gaineko zuzenketa automatikoa ebaluatzeko. Hala ere, ERRANT baliagarria suertatuko zaigu esaldi zuzenen kasuan beharrezkoak ez diren zenbat zuzenketa proposatzen diren aztertzeko (FP kopurua zenbatuz).

Horrez gain, esaldi zuzenen kasuan, sekuentziak alderatzeko erabiltzen den *difflib* moduluaz baliatu gara sistema bakoitzak asmatutako zuzenketa kopurua zenbatzeko, sarrerako-esaldiari aldaketarik egin ez zaizkion kasuak alegia.

Esaldi errealekin sistema bakoitzak lortutako ERRANT ebaluazioa eta esaldi zuzenekin lortutako aldaketarik gabeko zuzenketak eta faltsu positiboak 6.7 taulan ikus daitezke.

```
S alde handiz irabazi zuen caruanaren alderdiak ,
botoen % 58,35 bilduta , hots , duela lau urte
baino % 8,4 gehiago .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S horregatik haurren aferak ebaztea tokatzen
zaien helduei eskatzen ahal zaien gauza bakarra
horixe da : haurren interesei begiratzea , soilik
, helduen printzipio , interes , ideologia eta
aukerak bazter utzirik .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S batzuk kanoetan joan ginen , beste batzuk
basotik zehar .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S eta telebista autonomikoaren proiektua martxan
dagoenez , horrek du lehentasuna " , adierazi du
nuria iturriagagoitia bozeramaileak .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0
```

(a) Esaldi erroreduna vs esaldi zuzena.

```
S alde handiz irabazi zuen caruanaren alderdiak ,
botoen % 58,35 bilduta , hots , duela lau urte
baino % 8,4 gehiago .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S horregatik haurren aferak ebaztea tokatzen
zaien helduei eskatzen ahal zaien gauza bakarra
horixe da : haurren interesei begiratzea , soilik
, helduen printzipio , interes , ideologia eta
aukerak bazter utzirik .
A 3 4|||R:NOUN|||mugatzen|||REQUIRED|||-NONE-|||0
A 6 7|||R:OTHER|||,|||REQUIRED|||-NONE-|||0

S batzuk kanoetan joan ginen , beste batzuk
basotik zehar .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0

S eta telebista autonomikoaren proiektua martxan
dagoenez , horrek du lehentasuna " , adierazi du
nuria iturriagagoitia bozeramaileak .
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||0
```

(b) Esaldi erroreduna vs zuzenketa automatikoa.

6.4 Irudia: ERRANT-ek proposatutako anotazioak sistemak sarrerako-esaldiak zuzenak direla kasu guztietan antzematen ez badu.

TP	FP	FN	Prec	Rec	F0,5
0	2	0	0,0	1,0	0,0

6.6 Taula: ERRANT-ekin lortutako balioak 6.4 irudiko fitxategiak ebaluatzean.

Egoera hauetan emaitzak harrigarriak suertatu dira, ez baita aurretik genuen hipotesia bete. Automatikoki sortutako esaldi erroredunen kasuan emaitza hoberenak lortu dituen sistemak esaldi erroredun errealen kasuan emaitza kaskarrenak lortzen dituela antzeman dugu, egin beharreko zuzenketa guztietatik 2 soilik asmatzen dituelarik. Hala ere, F0,5 puntuazio baxuena lortu arren, beharrezkoak ez diren zuzenketa gutxien proposatzen dituen sistema da. Esaldi zuzenen kasuan arrakasta handiagoa du, 428 esaldik aldaketarik behar ez dutela detektatuz eta behar ez diren 48 zuzenketa bakarrik proposatuz.

Laugarren sistema da esaldi errealen eta zuzenen kasuan puntuazio altuena lortu duena. Alde batetik, esaldi erroredun errealei dagokienez, 8 errore zuzentzea lortu du. Bestetik,

	Esaldi erroredun errealak						**Esaldi zuzenak**	
	TP	**FP**	**FN**	**Prec**	**Rec**	**F0,5**	**Zuz**	**FP**
err_zuz400K	5	611	784	0,0081	0,0063	0,0077	49	1131
err_zuz5M	3	254	786	0,0117	0,0038	0,0083	301	289
err_zuz8M	2	**174**	787	0,0114	0,0025	0,0067	428	48
err+_zuz8M	**8**	230	**781**	**0,0336**	**0,0101**	**0,023**	**447**	**33**

6.7 Taula: Esaldi erroredun errealetan eta esaldi zuzenetan sistema bakoitzak lortutako emaitzak, ERRANT metrika eta *difflib* modulua erabiliz neurtuak.

ID	Esaldi erroreduna	Esaldi zuzena	Zuzenketa automatikoa
1	nik ez nago konforme	ni ez nago konforme .	nik ez nago konforme .
2	horregatik udaletxeari ~~ekonomiko~~ laguntza ekatzen diogu .	horregatik udaletxeari laguntza ekonomikoa eskatzen diogu .	horregatik , udaletxeari laguntza ekatzen diogu .
3	eskutitz hau idatzi baion lehen udaletxeko gizarte laguntzaileari egon ginen , berarentza aldizkaria idea ona iruditzen zitzaizun	eskutitz hau idatzi **baino** lehen udaletxeko gizarte laguntzailearekin egon ginen , berari aldizkaria idea ona iruditu zitzaion	eskutitz hau idatzi **baino** lehen , udaletxeko gizarte laguntzaileari begira egon ginen , berarentza aldizkaria idea ona iruditzen zitzaizun .
4	bestalde ez zitzaidan gustatu .	bestalde , ez zitzaidan gustatu .	bestalde **,** ez zitzaidan gustatu .
5	bi liburu~~a~~ irakurri ditut .	bi liburu irakurri ditut .	bi liburu irakurri ditut liburua .
6	pentsatzen dute ~~zerbait~~ gizakian , ekintza indarkeritsuak egitera , eramaten duen	pentsatzen dute gizakian zerbaitek , indarkeria ekintzak egitera , eramaten duela	pentsatzen dute gizakian , ekintza indarkeritsuak egitera , eramaten duen zerbait .
7	baina beste aukera dago	baina beste aukera **bat** dago	baina beste aukera **bat** dago .
8	hau dela eta , segun zer garraio erabiltzen duzun horrelakoak izango zara	hau dela eta , segun **eta** zer garraio erabiltzen duzun horrelakoa izango zara	hau dela eta , segun **eta** zer garraio erabiltzen duzun horrelakoak izango dira .

6.8 Taula: Laugarren sistemak esaldi erroredun errealetan modu egokian zuzendutako 8 erroreak.

esaldi zuzenen kasuan, 466 esalditik 447 zuzenak direla antzematea lortu du eta gainontzekoetan beharrezkoak ez diren 33 zuzenketa besterik ez ditu proposatu.

6.8 taulan ikusi daitezke sistemak gizaki batek egingo lukeen bezala detektatu eta zuzendu dituen erroreak. Kontuan hartu behar da sistemak zuzendutakoak errore indibidualak direla, ez esaldi osoak. Asmatutako kasuak berdez adierazi dira taulan. Zuzenketa hitz edo karaktere bat ezabatzea den kasuetan "Esaldi erroreduna" zutabean adierazi da ezabatutakoa marratuz. Gainontzeko kasuetan gehitutako hitza edo ikurra "Zuzenketa automatikoa" eta "Esaldi zuzena" zutabeetan azpimarratu da. Esaldietan zuzentzea lortu ez dena gorriz nabarmendu da "Esaldi erroreduna" eta "Zuzenketa automatikoa" zutabeetan. Gehitu gabe geratu diren hitzak laranjaz adierazi dira "Esaldi zuzena" zutabean eta gehitu direnak baina beharrezkoak ez zirenak urdinez "Zuzenketa automatikoa" zutabean.

Antzeman ez dituen errore asko dauden arren, badira sistemak zerbait ikasi duela adierazten diguten aztarnak. 3. esaldian, adibidez, ez du lortu "laguntzaileari" ordez "laguntzailearekin" hitza jarri behar dela detektatzea, baina "laguntzaileari" hitza mantenduta hurrengo hitza "egon" ezin dela izan antzeman du eta "begira" hitza gehitu du tartean, horrela zentzuduna den "laguntzaileari begira egon" sekuentzia osatuz. 8. esaldiaren kasua ere da aipagarria. Bilatzen zen zuzenketa "horrelakoak" hitza ordez "horrelakoa" hitza erabiltzea zen, "zara" hitzarekin bat etortzeko. Sistemak ez du aldaketa hori egitea lortu eta "horrelakoak" hitza mantendu du, baina "horrelakoak" eta "zara" bateragarriak ez direla ikasi du eta "horrelakoak" hitzari dagokion aditz-laguntzailea, kasu honetan "dira", topatu du. Aldaketa horrekin, sistemak proposatutako esaldi osoa gramatikalki zuzena da. Hala ere, ebaluaziorako erabiltzen ari garen irizpidea jarraituta negatiboki ebaluatzen da, ez datorrelako proposatzen den zuzenketarekin bat.

ID	Esaldi zuzena	Zuzenketa automatikoa
1	gutxienez historia alde dauka alavesek , coruñako deportivo ez baita mendizorrotzan irabazteko gai izan .	gutxienez historia alde dauka alavesek , coru ,ako deportivo ez baita mendizorrotzan irabazteko gai izan .
2	elkarri : "nafarroa bake prozesutik at nahi dute " aierdiren ustez , iruñean bake mahaia behar da	elkarri : "nafarroa bake prozesutik at nahi dute " aierdiren ustez , iruñean bake mahaia hartu behar da .
3	maiatzaren 12an donibane lohizunen eta iruñean manifestazioak eginen dituzte .	maiatzaren 12an donibane lohizunen eta iru :ean manifestazioak eginen dituzte .
4	las desventuras de beavers : 17:00. la niña de tus ojos : 20:00 .	las desventuras de beavers : 17:00. la ni ,a de tus ojos : 20:00 .
5	bigarren txanda suertatu zitzaien neurtuena beti gazte eta abadiñori .	bigarren txanda suertatu zitzaien neurtuena beti gazte eta abadi ,ori .

6.9 Taula: Laugarren sistemak esaldi zuzenetarako proposatutako aldaketa batzuk. Esaldi zuzenetarako proposatutako aldaketa guztiak C eranskinean aurki daitezke.

Sistemak esaldi zuzenak zuzentzeko orduan izandako portaera aztertzen badugu, 33 faltsu positibo horiek 19 esalditan gertatzen direla ikus dezakegu[4]. Are gehiago, ia kasu guztietan, proposatutako zuzenketak pertsona-izenak edo gaztelaniazko hitzak eraldatzeko asmoz egin dira. 6.9 taulan erakusten dira proposatutako aldaketetako batzuk. Kasu hauek aztertuz sistemak "ñ" karakterearekin arazoak dituela ikus dezakegu, esaldi gehienen kasuan letra hori duten hitzetan aurkitzen baitugu sistemak proposatzen duen zuzenketa. Baliteke inplementazioan antzeman ez dugun kodeketa arazoren baten ondorio izatea, baina badira letra horrekin arazoak erakusten ez dituzten kasuak, 6.9 taulako 2. esaldia adibidez. Hori kontuan hartuta, gure corpusek "ñ" karaktere gutxi dituztela pentsa dezakegu eta, ondorioz, sistemak entrenamendu garaian ez duela letra horren erabileraren adibide nahikorik eduki bertatik ikasi ahal izateko, gaztelaniako azentu-markarekin gertatzen den bezala. Hau guztia aztertuta, sistemak euskarazko esaldi zuzenak antzematen ikasi duela esan dezakegu, karaktere batzuek zailtasunak sortzen dituzten arren.

Emaitza hauek aztertu ondoren, kasu errealetan kalitate hobereneko zuzenketak laugarren sistemak ematen dituela ikusi dugu. Hori kontuan hartuta, errore automatikoak sortzeko diseinatutako bigarren metodoa, esaldiko errore kopurua handitzen zuena, kasu errealetan baliagarria dela ikusi dugu, hasiera batean automatikoki sortutako esaldi erroredunak zuzentzeko eraginkorra ez zirudien arren.

Esaldien hiru multzoak banaka ebaluatu ondoren esaldi guztiak batera ebaluatu ditugu, sistema bakoitzaren zuzenketen kalitate orokorra neurtzeko. Esaldi zuzenen kasuan ezin dugunez F0,5 puntuazioa erabili kalitatea neurtzeko, 1.398 esaldiak aldi berean ebaluatzeko GLEU metrika erabili dugu. 6.10 taulan ikusi daiteke sistema bakoitzak lortutako puntuazioa.

[4] C eranskineko C.1 taulan aurki daitezke esaldi zuzenetarako proposatutako zuzenketa guztiak.

err_zuz400K	err_zuz5M	err_zuz8M	err+_zuz8M
0,5429	0,7441	0,7759	0,7761

6.10 Taula: Sistema bakoitzak lortutako GLEU puntuazioa. Lehenengo errenkadan sistema entrenatzeko erabilitako corpusa zein izan den adierazten da.

Metrika honekin ere laugarren sistema suertatu da garaile, 0,7761 puntuazioa lortu duelarik. Dena den, hirugarren sistemak 0,7759 puntuazioa lortu du, 0,0002 gutxiago besterik ez. Aurretik ikusi dugunez, hirugarren sistemak sortzen ditu zuzenketa hoberenak automatikoki sortutako esaldien kasuan eta laugarren sistemak esaldi errealen kasuan (zuzenak zein erroredunak). Emaitza hauei laugarren sistema automatikoki sortutako esaldien kasuan emaitza hoberenak lortzetik 0,02 puntura egon dela eta hirugarren sistema esaldi erroredun errealen kasuan faltsu positibo gutxien proposatzeaz gain esaldi zuzenen kasuan gehienek zuzenketarik behar ez zutela antzemateko gai izan dela gehitzen badiegu, ulergarria da esaldi guztiak bateratzerakoan bi sistemek pareko puntuazioa lortzea.

Lau sistemak bi ebaluazio-metrikekin probatu ondoren, hoberena laugarren sistema dela esango dugu. Azken finean, gure helburua esaldi errealak zuzentzea da, eta kasu horretan laugarren sistema da puntuazio hoberenak jasotzen dituena ERRANT metrikaren arabera. Gainera, GLEU puntuazio altuena jasotzen duena ere bada, hirugarrenarekiko oso alde txikia eduki arren. Hortaz, etorkizuneko lanari begira, laugarren sistema hartuko dugu abiapuntu bezala eta honen emaitzak hobetzen saiatuko gara.

7. KAPITULUA

Ondorioak eta Etorkizunerako Lana

Kapitulu honetan proiektua amaitu ostean lortutako ondorioak aztertzen dira, bai lanari buruzkoak baita ondorio pertsonalak ere. Gainera, proiektuaren ildoan aurreratzen jarraitzeko eta lortutako emaitzak hobetzeko etorkizunerako lana proposatzen da.

7.1 Ondorioak

7.1.1 Proiektuaren ondorioak

Orokorrean, hasieran planteatutako helburuak bete direla esan dezakegu. Lehenik eta behin, euskaraz egiten diren erroreen analisia eta sailkapena burutu dugu, eta horretan oinarrituta erabaki dugu proiektua nola bideratu. Horrez gain, erroreak automatikoki sortzeko metodoak inplementatzeko gai izan gara, corpus desberdinak sortzea ahalbidetu diguna. Corpus horietako bakoitza sortzeko esaldi-kopuru desberdinak erabili ditugu eta horien artean erroredunak diren esaldien portzentaia ere aldatu dugu, entrenamendurako datu-multzo optimoenaren bila. Bestalde, Transformer arkitekturan sakondu dugu, ereduaren egitura zein inplementazioa ulertuz eta aldaketa txikiak eginez. Azkenik, entrenatutako sistemak probatu ditugu eta emaitzak interpretatzeko gai izan gara, jasotako zuzenketek gustatuko litzaigukeena baino kalitate txarragoa izan arren.

Lortutako emaitzei erreparatuz, zalantzarik gabe esan dezakegu errore gramatikalak zuzentzeko diseinatutako sistema bat garatzeko entrenamendurako corpusaren tamaina oso

handia izan behar dela, gutxienez 10 milioi esaldi ingurukoa. Gure ebaluazioaren kasuan, bai automatikoki sortutako esaldiekin hoberen funtzionatu duen sistema (hirugarrena) baita esaldi errealekin hoberen funtzionatu duen sistema ere (laugarrena) 10 milioi esaldi inguruz osatutako datu-multzoekin entrenatu dira. Gainera, bi kasuetan corpus horietako esaldietatik 8,5 milioi baino gehiago esaldi zuzenak dira, beraz, entrenatzeko esaldi zuzenen proportzio oso altu bat erabiltzearen garrantzia ere nabarmendu beharra dago.

Aurrekoaz gain, errore errealen parekoak diren erroreak erabiltzeak duen garrantziaz ohartu gara. Guk sortutako erroreak artifizialak dira eta sistemak mota horretako erroreak zuzentzen ikasi arren, oso zaila da errealitatean horrelako erroreak dituzten esaldiak aurkitzea eta, ondorioz, kasu askotan sistema ez da gai ikasitakoa aplikatzeko.

7.1.2 Ondorio pertsonalak

Ikasketa pertsonalari dagokionez, proiektu hau esperientzia aberasgarria izan da niretzat, bai arloaren inguruan lortutako ezagutzagatik eta baita proiektu handi batean lan egiteak suposatu duen ikasketagatik ere.

Proiektu honetan lan egiten hasi nintzenean ez nuen ia ezagutzarik ikasketa sakonaren arloan. Aurrerago, *Machine Learning and Neural Networks* irakasgaian, gaiari buruzko informazio gehiago lortu nuen, arloaren inguruan interesa pizteko baliagarria izan zena. Hala ere, ikasgai horretan burututako proiektuak dimentsio txikikoak ziren eta, beraz, ez nuen proiektu errealen dimentsioa ezagutzeko aukerarik izan. Lan honek faltan botatzen nuen ezagutza hori lortzen lagundu dit, gaur egun artearen egoeran dagoen eredu baten inplementazioa modu sakonean aztertzeaz gain, ikasketa-denbora errealak ezagutzea eta hauekin lan egiteak suposatzen dituen erronkak ikastea ahalbidetu baitit.

Horrez gain, proiektu hau graduko beste hainbat irakasgaitan landutako kontzeptuak erabiltzeko aukera izan da. Datu Meatzaritza, Konputazio Eredu Abstraktuak edo Bilaketa Heuristikoak dira irakasgai horien adibide, baina aurretik aipatutako *Machine Learning and Neural Networks* irakasgaiaz gain, arlo baliagarrienak Hizkuntzaren Prozesamendua eta Proiektuen Kudeaketa izan dira. Hizkuntzaren Prozesamendua irakasgaiari esker, ikasketa sakoneko hainbat gai gogoratzeaz bestalde, corpusak lantzeko erak, ebaluazioteknika desberdinak eta hizkuntza-ereduen erabilera ikasi ditut, proiektu honetan oso baliagarriak izan direnak. Proiektuen Kudeaketa arloari dagokionez, ezinbestekoa izan da plangintza egituratzeko eta kontrol sendoa jarraitzeko.

Graduko irakasgaietan jasotako ezagutzaz gain, proiektu hau aurretik egindako enpresako praktika batzuetan ikasitakoa erabiltzeko baliagarria izan da. Enpresako praktikak horietan dimentsio handiko proiektu batean lehenengoz lan egin nuenez, kalitatezko emaitza batzuk lortu ordez, helburua ezagutza lortzea izan zen. Ezagutza hori mesedegarria izan zait proiektu honetako hainbat atazatan, batez ere corpus handiaren prozesamenduan zerikusia dutenetan.

Proiektuaren kudeaketari dagokionez, memoria hau idaztea hasiera batean pentsatutakoa baino erronka handiagoa izan dela aitortu beharra dut. Zaila izan da egindako lanaren azalpena egituratzea, txosten honetako kapitulu bakoitza ahalik eta koherenteena eta ulergarriena izan dadin. Gainera, egunero idazteko motibazioa mantentzea ere gogorra izan da, batez ere kurtsoaren eta proiektuaren amaiera izateak dakarren nekeagatik. Memoria hau idazteak, idatzitakoa berriz irakurtzearen eta zuzentzearen garrantzia gogorarazteaz gain, egindako lana transmititzeko gaitasuna izatea premiazkoa dela erakutsi dit. Emaitza onak jasotzea ez da nahikoa; egindakoa modu formalean azaltzeko gaitasuna ezinbestekoa da, aldi berean beste pertsonentzako irakurketa erraza eta ulergarria dela ziurtatuz.

Azkenik, proiektu hau esperientzia baliagarria izan da ikerketa-talde baten funtzionamendua ezagutzen hasteko eta bertan lan egiten duten zenbait pertsona ezagutzeko. Hizkuntzaren Prozesamenduaren arloan lan egiten duten ikerlariekin lan egiteak eta graduan izen bereko irakasgaia landu izanak arloaren inguruko jakin-mina sortu dit eta gaiaren inguruan ikasten jarraitzea gustatuko litzaidakeela erabakitzeko lagungarria izan da.

7.2 Etorkizunerako Lana

Lortutako emaitzak ikusita, argi dago oraindik bide luzea geratzen dela euskarazko errore gramatikalen kalitatezko zuzenketa eskaintzeko gai izango den sistema bat garatzeko. Jarraian egindako lanetik abiatuta hurrengo pausuak zeintzuk izan liratekeen deskribatuko dugu.

Lehenengo pausua errore zentzudunagoak sortzea izango litzateke. Hori lortzeko, hitz mailako erroreak sortzeko metodo berri bat garatu beharko genuke, oraingoan errore errealak imitatuz. Badakigunez euskaraz egiten diren errore gramatikalen kopuru oso handi bat komunztadura motakoak direla, esaldi zuzenen analisi bat egin beharko genuke eta, adibidez, aditzen kasua edo numeroa aldatu. Horrela esaldi originaleko elementuekin (subjektu, objektu, predikatu, eta abar) komunztadura eza ziurtatuko genuke. Adibide ho-

ri jarraituz, banaka definitu beharko genituzke errore-sorkuntza metodoak A eranskineko sailkapenean analisia behar dutela erabaki dugun errore mota bakoitzeko.

Hala ere, hitz mailakoak ez dira esaldi batean sortu daitezkeen errore bakarrak. Erroreak karaktere mailakoak ere izan daitezke, hau da, errore ortografikoak. Sistemak errore hauek zuzentzen ikasteko entrenamendurako corpusean txarto idatzitako hitzak (eta haien zuzenketa) ere gehitu beharko genituzke. Errore ortografikoak automatikoki sortzeko orain arte erabili dugun errore sorkuntza metodoa erabili ahalko genuke, baina oraingoan esaldi osoan aplikatu ordez, hitz bakoitzean aplikatuz. Beste era batera esanda, txarto idatzita egotea nahi dugun hitz bakoitzeko ezabatu, gehitu, ordezkatu eta trukatu eragiketak aplikatuko genizkioke hitz horretako karaktereei.

Errore zentzudunak sortzeko beste aukera bat *confusion set*-ak erabiltzea izango litzateke. *Confusion set* hauek hitz jakin batekin nahastu daitezkeen (lexikoki zein fonetikoki) hitzez osatuta egongo lirateke. Normalean nahasten diren hitzen informazioa zuzentzaile ortografikoetatik lortu ahalko litzateke, adibidez. Esaldi erroredunak sortzeko garaian, hitzak ausaz ordezkatu beharrean, uneko hitza bere *confusion set*-eko hitz batekin ordezkatuko litzateke, gizakiok egiten ditugun erroreen antza duten esaldi erroredunak lortuz. Metodo hau 2019ko *BEA Shared Task* lehiaketan [23] lehenengo postua lortu zuen taldeak erabilia izan zen.

Entrenamendurako corpuseko esaldi erroredunak sortzeko metodoez gain, corpus beraren tamaina handitzea ere interesgarria izango litzateke, esaldi gehiagorekin sistemaren zuzenketen kalitateak gora egiten jarraitzen ote duen aztertzeko. Proba hori egin ahal izateko euskaraz idatzitako corpus gehiago bildu beharko genuke. Aukera bat Wikipediako edizioak, hau da, testu beraren bertsio desberdinak haietan egindako zuzenketekin, erabiltzea izango litzateke, beste hizkuntza batzuekin egiten den bezala.

Azkenik, errore mota bakoitzerako sistema bat entrenatzen saiatu ahalko ginateke, sistema bakoitza mota jakin bateko erroreez soilik osatutako corpus batekin entrenatuz. Besteak beste, komunztadura zuzentzen aditua den sistema bat entrenatu genezake, errore ortografikoak zuzentzeko gaitasuna duen beste bat eduki genezake eta determinanteekin zerikusia duten erroreak antzeman eta zuzentzeko gai den beste bat. Errore-mota bakoitzean aditua den sistema bana edukita, zuzentzaile guztien *pipeline* bat egin ahalko litzateke. Esaldi erroreduna zuzentzaile guztietatik barrena bidalita, metodo honek esaldi bateko errore guztiak zuzentzeko gaitasuna ote duen aztertuko genuke.

Eranskinak

A. ERANSKINA

Euskarazko erroreen sailkapena

Eranskin honetan egindako euskarazko erroreen sailkapena adierazten da. Erroreak 5 talde nagusitan banatzen dira: (1) Lexikoak, (2) Morfologikoak, sintaktikoak eta morfosintaktikoak, (3) Nozioak, (4) Semantikoak eta (5) Puntuazio ikurrak.[1]

Talde bakoitzean multzo horren barnean aurki daitezkeen errore desberdinak adierazten dira eta bakoitzari buruzko informazio zehatza jarraian azaltzen diren 5 zutabeen bidez ematen da:

- Errorea: Zutabe honetan uneko errore zehatza eta bere kodea (urdinez) adierazten dira, Euskarazko erroreen sailkapena lanean [32] definitutako kodeak jarraituz.
- Azalpena: Zutabe honetan erroreetako batzuen azalpena ematen da, errorea zerk sortzen duen argiago adierazteko.
- Adibidea: Zutabe honetan uneko errorea duten esaldien adibideak ematen dira.
- Analisia?: Zutabe honetan erroreak sailkatzen dira. Uneko errorearen motako errore bat automatikoki sortzeko esaldiaren analisi bat egitea beharrezkoa bada "BAI" balioa ezarri zaio eta "EZ" balioa aurkako kasuan.
- Oharrak: Zutabe honetan bestelako edozein anotazio ematen da, erroreak automatikoki sortzeko beharko liratekeen datuak adibidez.

[1] Euskarazko erroreen sailkapena lanean [32] erroreak 7 taldetan banatzen dira; ortografia erroreak eta estilo kontuak sailkapen honetatik kanpo utzi ditugu.

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
(1) LEXIKOAK	Maileguak (LMA-M)		afamatu; moskeatu	EZ	Ohiko maileguen zerrenda bat beharko litzateke errorea antzemateko
	Konposizioa eta eratorpena (KONERA)		afaltzaile; haurtoki	EZ	zerrenda bat beharko litzateke errorea antzemateko ~~ Zuzentzaile ortografikoak detektatuko du
	Generoa (GENE)	Generoa gaizki erabiltzea	aitak etorri dira oporretatik; musika klasika zoragarria da	BAI	Baztertuta
	Esamoldeak eta Kolokazioak (ESAMOL)		lur eta zur; siesta bota	EZ	Esamolde zuzen eta okerren zerrenda
(2) MORFOLOGIKOAK, SINTAKTIKOAK, MORFOSINTAKTIKOAK	Deklinabidea (DEKL)	Deklinabide kasu okerrak	Jonen autoaz etorri naiz; prest dago guri zirkora eramateko	BAI	ABS-ERG INE (non) - INS (zertaz)
	Deklinabidea (INS)	Instrumentaltasuna	haizkorarekin ebaki dugu	BAI	INS (zertaz) - Norekin
	Determinantea (DETMK)	Determinante mugatzailea kendu	txokolate nahi dut	BAI	
	Determinantea (DETMG)	Determinante mugatzailea gehitu	zer ordua da?	BAI	

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Izenordainak (IZEORD)	Izen ordainen erabilera okerra	bere buruari ikusi da/bera ikusi da; elkarri hitz egin dute	BAI	
	Izenordainak (IZEORDZG)	Izen ordain zehaztugabeak	ez dut nahi ezer ez	EZ	
	Adjektiboak eta adberbioak (ADJADB)	Adjektiboa eta adberbioa nahastea	hobe egin du	EZ	erroreak sortzeko adberbioa dagoen tokian adjektiboa jarri eta alderantziz (adb: hobe/hobeto ordezkatu)
	Adjektiboak eta adberbioak (ADJIA)	Adjektiboa izenaren aurretik jartzea	ekonomiko arazoak dauzkate	BAI ?	Errorea sortzeko ordena (+ deklinabidea ?) aldatu
	Adjektiboak eta adberbioak (ADJGRA)	Adjektibo graduatzaileen erabilera okerra	hobeago	EZ	“hobe” edo “hobeto” aurkitzen den esaldietan “hobeago” jarri errorea sortzeko
	Adjektiboak eta adberbioak (ADJGN)	Adjektibo galdetzaileen eta nolakotzaileen erabilera okerra	nola dira zure gurasoak?	???	
	Aposizioak (APOS)		Zure lagunari, Dublinen bizi dena, sari bat eman diote	BAI	
	Postposizioak (POST)		Izaskun buruz hizketan ibili da	BAI	

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Aditza (ADAM)	Aditzaren denbora, aspektua edo modua nahastea	hori zen bere herritik gehien gustatzen zaiona; goaz mendira?	BAI	
	Aditza (PARADIG_N_N-NK)	Nor eta nor-nork nahastea	ez da funtzionatzen	BAI	
	Aditza (PARADIG_N_N-NI)	Nor eta nor-nori nahastea	nagusiari zuzendu da	BAI	
	Aditza (PARADIG_N-NI_N-NK)	Nor-nork eta nor-nori nahastea	niri hori ez zait molestatzen	BAI	
	Aditza (PARADIG_N-NK_N-NI-NK)	Nor-nork eta nor-nori-nork nahastea	Joni ikusi diot	BAI	
	Aditza (PARADIG_N-NI_N-NI-NK)	Nor-nori eta nor-nori-nork nahastea	gustatzen dit	BAI	
	Komunztadura (KOMSIN)	Sintagma barruko komunztadura eza	gurasoak eta lagunez mintzatu zara; guk geu	BAI	
	Komunztadura (KOMAPOS)	Komunztadura eza aposizioan	zure laguna, Dublinen bizi denari, sari bat eman diote	BAI	
	Komunztadura: Aditza - Subjektua (KOMPAS-NUM)	Komunztadura eza perpausean, aditza eta subjektuaren artean, numeroari dagokionean: aditzean subjektuari dagokion numeroa eta subjektuan agertzen den numeroa ez datoz bat	aurrerapen handia daude; gizonek egin du	BAI	

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Komunztadura: Aditza - Subjektua (KOMPAS-KAS)	Komunztadura eza perpausean, aditza eta subjektuaren artean, kasuari dagokionean: aditzean subjektuari dagokion kasua eta subjektuan agertzen den kasua ez datoz bat	zuk etorri zara; nik esnatu naiz	BAI	
	Komunztadura: Aditza - Objektua (KOMPAO-NUM)	Komunztadura eza perpausean, aditza eta objektuaren artean, numeroari dagokionean	eman dizut liburuak; nik etxeak ikusi dut	BAI	
	Komunztadura: Aditza - Objektua (KOMPAO-KAS)	Komunztadura eza perpausean, aditza eta objektuaren artean, kasuari dagokionean	nik etxeek ikusi ditut	BAI	
	Komunztadura: Aditza - Zehar-objektua (KOMPAZO-NUM)	Komunztadura eza perpausean, aditza eta zehar objektuaren artean, numeroari dagokionean	ziberespazioan dabiltzan pertsonei dagokion izena da; emaitzei dagokiona	BAI	dagokion/dagokien ordezkatu soilik?
	Komunztadura: Aditza - Zehar-objektua (KOMPAZO-KAS)	Komunztadura eza perpausean, aditza eta zehar objektuaren artean, kasuari dagokionean	nik haiek eman diet	BAI	
	Komunztadura: Aditza - Predikatua (KOMPAP)	Komunztadura eza perpausean, aditza eta predikatuaren artean	gure erleak oso soziablea dira	BAI	

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Komunztadura (KOMM)	Komunztadura eza mendekoetan	goxoki asko jaten duen umeek kariesa dute	BAI	
	Mendeko esaldiak (MEN-KON)	Konpletiboak	ez dut uste etorriko dela; gizon horrek esan du nik istilua izan dut	BAI	
	Mendeko esaldiak (MEN-ZG)	Zehar-galderak	ez dakit nor da; galdetu ea joango bada	BAI	
	Mendeko esaldiak (MEN-HEL)	Helburuzkoak	etxera noa afaltzeko; paseatzeko joan dira	???	Nahikoa da “-era” bukaera duten hitzak “-eko” bukaerarekin ordezkatzea?
	Mendeko esaldiak (MEN-KAU)	Kausazkoak	zergatik ez zinen joan? zergatik ez neukan gogorik; ez delako etorri ez dugu ikusi	BAI	
	Mendeko esaldiak (MEN-BAL)	Baldintzazkoak	edukiz gero dirua; kontuz ez ibiliz gero	BAI	
	Mendeko esaldiak (MEN-ERL)	Erlatibozkoak	mutil bat etorri da zu ezagutzen zaituela; nik eman dizut liburua polita da	BAI	
	Mendeko esaldiak (MEN-DEN)	Denborazkoak	helduko denean, abisatu; ikusiko dudanean esango diot	???	“-ten” edo “-tzen” bukaera duten hitzak ordezkatu “-ko” bukaera jarriz???

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Mendeko esaldiak (MEN-KONT)	Kontzesiboak	izan arren berandu, joan egingo naiz; ez bada ere oso handia, guztiok sartuko gara	EZ	Ordena aldatzea nahikoa errorea sortzeko
	Mendeko esaldiak (MEN-MOD)	Moduzkoak	Jon zu bezala da; ez dira uste nuen bezala	EZ ?	“bezala” eta “bezalakoak” ordezkatu
	Mendeko esaldiak (MEN-KONP)	Konparaziozkoak	zuk baino dirua gehiago daukat	EZ	
	Mendeko esaldiak (MEN-NOM)	Nominalizazioa	ezin nuen liburua irakurtzen	BAI	
	Perpausen egitura (HITZOR-S)	HItzen ordena, sintaxiari dagokionean	jakin dudanez auzokide baten bitartez Udalak dirua eskatzen du; arrantza motak erabiltzen zirenak, oso erle fina ez baitzen	EZ	Hitzen ordena aldatzearekin nahikoa
	Juntagailuak eta lokailuak (JUNLOK)			EZ	
(3) NOZIOAK	Denbora (DENNOZ)		ostirala afaria daukat; egun guztietan joaten naiz	EZ ?	
	Ordua (ORDNOZ)		zazpiak t’erdietan goaz; bost t’erditan	EZ ?	
	Data (DATNOZ)		Donostia, 1995eko urtarrilak 15ean; Donostian, 1995eko martxoaren 22	EZ ?	
	Eguraldia (EGUNOZ)		atertu da; euria ari da	EZ	Eguraldiarekin zerikusia duten hitzen zerrenda bat beharko litzateke?

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Sentipenak (SENNOZ)		gosea daukagu; beldurra det	EZ	
	Zenbatasuna (ZENNOZ)		hiru kilo sagarrak; bero/gose/antz asko	EZ	
	Zenbakiak (ZENB)		mila bederatziehun pezeta; hiru mila eta zazpiehun eta hogei	EZ	
(4) SEMANTIKOAK	Hitz mailakoak - Sasi-adiskideak	Hitz bat beste batekin nahastu, ezberdintasunak jakin gabe	xelebre (célebre); azienda (hacienda)	EZ	
	Hitz mailakoak - Pare dikotomikoak	Fonetikoki berdinak edo antzekoak diren hitzak, baina ezberdin idatzi eta esanahi ezberdina dutenak	hura - ura; ari - hari; hasi - hazi; atso - atzo; arrastaka - arrakasta; arazo - azaro	EZ	Erroreak sortu hitz bikoteak bata bestearen ordez jarriz
	Esaldi mailakoak		mahaia edan nuen	BAI ~	Testuingurua aztertu behar da
	Esamolde edo egitura zuzenak ez dagokien egoeratan/momentutan erabiltzea		jaten ari den bati “bejondeizula”; arratsaldean “egun on”	BAI	Testuingurua aztertu beharra dago
(5) PUNTUAZIO IKURRAK	Puntuazio ikurra behar ez denean erabiltzea (PIE-K) (PIE-P) (PIE-BP) (PIE-HP) (PIE-GH)	K: koma P: puntua BP: bi puntu HP: hiru puntuak GH: galdera/harridura		EZ	Errorea sortu daiteke esaldi batean puntuazio ikurrak ausaz gehituz

	ERROREA	Azalpena	Adibidea	Analisia?	Oharrak
	Puntuazio ikurra behar denean ez erabiltzea (PIEE-K) (PIEE-P) (PIEE-BP) (PIEE-HP) (PIEE-GH9)	K: koma P: puntua BP: bi puntu HP: hiru puntuak GH: galdera/harridura		EZ	Errorea sortu daiteke esaldi batean dauden puntuazio ikurrak kenduz
	Puntuazio ikurrak ordezkatzea edo nahastea (PIO_P-K) (PIO_K-P) (PIO_KB-P)	P-K: puntua beharrean koma K-P: koma beharrean puntua KB-P: koma beharrean bi puntu		EZ	Errorea sortu daiteke esaldi batean aurkitzen den puntuazio ikur bakoitza beste baten ordez trukatuz
	Puntuazio ikurrak: parentesiak, komatxoak, barrak, gidoiak… ez ixtea edo irekitzea (PII)			EZ	

B. ERANSKINA

Garapenerako ebaluazioan lortutako zuzenketak

Eranskin honetan proiektua burutu ahala egindako garapenerako ebaluazioetan lortutako zuzenketak erakusten dira, sistema bakoitzaren zuzenketa-gaitasunaren adibide bezala. 6.2 atalean aipatu den bezala, ebaluazio honetarako 26 esaldi erabili dira, hiru multzotan banatuak: guk asmatutako esaldi erroredunak, automatikoki sortutako esaldi erroredunak eta esaldi zuzenak. Esaldi hauek jarraian erantsitako taulako lehenengo zutabean adierazi dira, "Sarrerako esaldia" izenekoan. Esaldi bakoitzaren hasieran parentesi artean adierazi da esaldi-mota, hurrengo kodeak erabiliz:

- EE: Errore erreala. Guk asmatutako esaldi erroredunak; kasu errealetan aurki daitezkeen esaldi erroredunen adibideak.
- EA: Errore automatikoak. 5.3 atalean azalduko metodoa erabiliz sortutako esaldiak.
- Z: Esaldi zuzenak.

Taularen lehenengo errenkadan sistema bakoitza entrenatzeko erabilitako corpusa zein izan den adierazten da.

Sarrera-esaldia	err_400K	err_5M	err_8M	err+_8M
(EE) nik ane deitzen naiz	nik ane deitzen naiz .	nik ane deitzen naiz ?	nik ane deitzen naiz .	nik ane deitzen naiz .
(EE) ni oporretan joango dira	ni oporretan joango dira .	ni oporretan joango dira .	ni oporretan joango dira .	ni oporretan joango dira .
(Z) zerbait aldatu mesedez	zerbait aldatu mesedez .	zerbait aldatu mesedez mesedez zerbait aldatu mesedez zerbait aldatu mesedez ?	zerbait aldatu mesedez .	zerbait aldatu mesedez mesedez .
(EA) mendearen amaieran , berriz zeuden mañerun , tabakoa , olioa eta batez ere ardoa ekoizten zuten enpresak , .	mendearen amaieran , berriz , mañerun , tabakoa , badu eta batez ere ardoa ekoizten zuten enpresak , .	mendearen amaieran , berriz , mañerun , tabakoa , olioa eta batez ere ardoa ekoizten zuten enpresak zeuden .	mendearen amaieran , berriz , mañerun , tabakoa , olioa eta batez ere ardoa ekoizten zuten enpresak zeuden .	mendearen amaieran , berriz zeuden mañerun , tabakoa , olioa eta batez ere ardoa ekoizten zuten enpresak , .
(EE) ez dakit zer gehiago joan paper honetan	ez dakit zer gehiago joan paper honetan paper honetan .	ez dakit zer gehiago joan paper honetan .	ez dakit zer gehiago joan paper honetan .	ez dakit zer gehiago joan den paper honetan .
(EE) nola zuzenduko du zerbait ez badu ulertzen etxe esaten dena	nola apain du zerbait ez badu ulertzen etxe esaten dena ?	nola zuzenduko du zerbait ez badu ulertzen , esaten dena ?	nola zuzenduko du zerbait ez badu ulertzen etxe esaten dena ?	nola zuzenduko du zerbait ez badu ulertzen zer esaten dena ?
(EE) ez dakit zer pasatzen den azken aldi hontan jendea hasi dela dantzatzen sarritan	ez dakit zer pasatzen den azken aldi hontan jendea hasi dela dantzatzen .	ez dakit zer pasatzen den azken aldi hontan jendea hasi dela dantzatzen sarritan .	ez dakit zer pasatzen den azken aldi hontan jendea hasi dela dantzatzen sarritan .	ez dakit zer pasatzen den azken aldi hontan jendea hasi dela dantzatzen sarritan
(EE) zerbit ikusi al duzue ?	16.t ikusi al duzue ?	zerbit ikusi al duzue ?	zerbit ikusi al duzue ?	zerbit zer ikusi al duzue ?

(EE) zer gehigo proposatu daiteke	zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke .	zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke ?
(EE) zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke ?	zer gehigo proposatu daiteke , ezta ?
(EA) euskal kultur erakundea , iparraldeko egiteko kultur erakunde nagusia duten da .	euskal kultur dago , iparraldeko kultur erakunde nagusia duten dago .	euskal kultur erakundea , iparraldeko kultur erakunde nagusia da .	euskal kultur erakundea , iparraldeko kultur erakunde nagusia izan da .	euskal kultur erakundea da , iparraldeko kultur erakunde nagusia , da .
(EA) baionako udalak hartu aginduta , zubia zirkulazioari hori dago .	baionako udalak hartu dira zebilen , zubia .	baionako udalak aginduta , zubia zirkulazioari hori dago .	baionako udalak hartu aginduta , zubia zirkulazioari hori dago .	baionako udalak aginduta , zubia zirkulazioari lotuta dago .
(EA) honetan baten azaldu dirusarreren inguruko zurrumurruen aurrean izango ministro ohia haserre agertu da sare sozialetan .	honetan , dirusarreren inguruko zurrumurruen aurrean izango da ministro ohia " .	aste honetan , .	honetan , dirusarreren inguruko zurrumurruen aurrean izango duten ministro ohia haserre agertu da sare sozialetan .	kasu honetan , dirusarreren inguruko zurrumurruen aurrean , ministro ohia haserre agertu da sare sozialetan .
(EA) herrialde horiek asko duenez merke . nahi dute ekoiztu	herrialde horiek asko merke - nahi dute idatzi nahi dute mendian .	herrialde horiek asko duenez merke ekoiztu nahi dute .	herrialde horiek asko duenez merke ekoiztu nahi dute .	herrialde horiek asko eta merke ekoiztu nahi dute .
(EA) bestalde . gorpua eskatu epailearen esku gelditu da ,	bestalde , gorpua , epailearen esku gelditu da .	bestalde , gorpua epailearen esku gelditu da .	bestalde , gorpua epailearen esku gelditu da .	bestalde , gorpua epailearen esku gelditu da .
(Z) errespetatzen al duzu amatasun eta aitatasun baimena ?	errespetatzen al duzu amatasun eta aitatasun baimena ?	errespetatzen al duzu amatasun eta aitatasun baimena ?	errespetatzen al duzu amatasun eta aitatasun baimena ?	errespetatzen al duzu amatasun eta aitatasun baimena ?

(EA) ustezko erasotzaileak zion eman zuen gain burdinezko barra bat jaurti , ondoren , egindako baina ez zion makilaz .	ustezko erasotzaileak , eman zuen gain burdinezko barra bat jaurti zion ondoren , baina ez zion makilaz .	ustezko erasotzaileak , eman zuen gain burdinezko barra bat jaurti zion ondoren , baina ez zion makilaz .	ustezko erasotzaileak diote eman zuen gain burdinezko barra bat jaurti , baina ondoren , baina ez zion makilaz .	ustezko erasotzaileak makilaz eman zuen burdinezko barra bat jaurti ondoren , baina , ez zion eman .
(Z) argi daukadana lasaiago ibili nahi dudala da .	argi daukadana lasaiago ibili nahi dudala da .	argi daukadana egiten aldekoak ibili nahi dudala da .	argi daukadana lasaiago ibili nahi dudala da .	argi daukadana lasaiago ibili nahi dudala da .
(EA) hauek ere , eskean ibiltzeaz gain , dantzan ziren .	hauek ere , eskean ibiltzeaz gain , dantzan izan ziren .	hauek ere , eskean ibiltzeaz gain , dantzan ziren .	hauek ere , eskean ibiltzeaz gain , dantzan ziren .	hauek ere , eskean ibiltzeaz gain , dantzan ziren .
(EA) ez da zeharo hilgarria , baina gaixotu askoz egiten du .	ez da zeharo hilgarria , baina klasikoak askoz egiten du .	ez da zeharo hilgarria , baina gaixotu askoz egiten du .	ez da zeharo hilgarria , baina gaixotu askoz egiten du .	ez da zeharo hilgarria , baina gaixotu askoz egiten du .
(EA) gainerako kondekorazioak ministro-agindu bidez emango dira eta justizia ministerioaren informazio aldizkarian esker dira .	gainerako kon " kon .razioak ministro : : , eta justizia ministerioaren informazio aldizkarian esker dira .	gainerako kondekorazioak ministro-agindu bidez emango dira eta justizia ministroaren informazio aldizkarian esker dira .	gainerako kondekorazioak ministro-agindu bidez emango dira eta justizia ministerioaren informazio aldizkarian esker dira .	gainerako kondekorazioak ministro-agindu bidez emango dira eta justizia ministerioaren informazio aldizkarian esker dira .
(EA) euskaraz dokumentu dituztenak txostenak aurkeztea eta hizkuntza berean erantzunak jasotzeko eskubidea .	euskaraz dokumentu horren txostenak , eta hizkuntza berean .	euskaraz egingo dituztenak txostenak aurkeztea eta hizkuntza berean erantzunak jasotzeko eskubidea .	euskaraz dokumentu eta txostenak aurkeztea eta hizkuntza berean erantzunak jasotzeko eskubidea .	euskaraz dokumentu txostenak aurkeztea eta hizkuntza berean erantzunak jasotzeko eskubidea da .
(EE) zenbat egun falta dute eskolak bukatzeko ?	zenbat egun falta dute eskolak bukatzeko ?	zenbat egun falta dute eskolak bukatzeko ?	zenbat egun falta dute eskolak bukatzeko ?	zenbat egun falta dute eskolak bukatzeko ?

(EA) aldizkari baten azala baino gehiago mendean izango da lehen orrialdea , bere maketazioaren arabera .	aldizkari baten azala baino gehiago izango da lehen orrialdea , bere errenzioaren arabera .	aldizkari baten azala baino gehiago mendean izango da lehen orrialdea , bere maketazioaren arabera .	aldizkari baten azala baino gehiago izango da lehen orrialdea , bere maketazioaren arabera .	aldizkari baten azala baino gehiago mendean izango da lehen orrialdea , bere maketazioaren arabera .
(EE) etxera garaiz heltzen saiatu da baina ez lortu du trena berandu heldu delako	etxera garaiz geratzen saiatu da baina ez lortu du trena berandu heldu delako .	etxera garaiz heltzen saiatu da baina ez lortu du trena berandu heldu delako .	etxera garaiz heltzen saiatu da baina ez lortu du trena berandu heldu delako	etxera garaiz heltzen saiatu da baina ez lortu du trena berandu heldu delako
(EA) gaur bularreko aurkako minbiziaren nazioarteko eguna da , eta hainbat ekitaldi egingo dituzte , gaixotasuna garaiz atzematea ezinbestekoa aldarrikatzeko .	gaur bularreko , minbiziaren nazioarteko eguna da , eta hainbat ekitaldi egingo dituzte , gaixotasuna garaiz atzematea ezinbestekoa aldarrikatzeko .	gaur bularreko aurkako minbiziaren nazioarteko eguna da , eta hainbat ekitaldi egingo dituzte , gaixotasuna garaiz atzematea ezinbestekoa aldarrikatzeko .	gaur bularreko minbiziaren aurkako nazioarteko eguna da , eta hainbat ekitaldi egingo dituzte , gaixotasuna garaiz atzematea ezinbestekoa aldarrikatzeko .	gaur bularreko aurkako minbiziaren nazioarteko eguna da , eta hainbat ekitaldi egingo dituzte , gaixotasuna garaiz atzematea ezinbestekoa aldarrikatzeko .

C. ERANSKINA

Laugarren sistemak esaldi zuzenetarako proposatutako zuzenketak

Eranskin honetan laugarren sistemak, err+_zuz8M corpusarekin entrenatutakoak, esaldi zuzenetarako proposatutako zuzenketa guztiak erakusten dira. Laugarren sistemak ebaluazio corpuseko 466 esaldi zuzenetatik 19 esalditan egiten ditu zuzenketak. 19 esaldi horietan 33 zuzenketa proposatzen ditu, C.1 taulan bildutakoak hain zuzen ere (aurretik 6.3 ataleko 6.9 taulan ikusi ditugu zuzenketa hauetako batzuk). Lehenengo zutabean esaldi originala erakusten da, zuzenketarik behar ez duena, eta bigarren zutabean sistemak egindako zuzenketa. Aldatutako edo gehitutako hitzak laranjaz nabarmendu ditugu.

Esaldi zuzena	Zuzenketa automatikoa
1begoña vicario zinemagile esperimentalista n euskal animazio zinemari buruzko ikuspegi bat .	1bego : 1bego , vicario zinemagile esperimentalista n euskal animazio zinemari buruzko ikuspegi bat .
hala ere , "liskar politikoak alde batera uzteko garaia " dela esan zuen knörrek eta gaineratu zuenez , europan kontzertuaren "adiera bakarreko defentsa egin behar dute instituzioek , interes politikoak eta sinesmenak alde batera utziz ".	hala ere , "liskar politikoak alde batera uzteko garaia " dela esan zuen kn , celarrek eta gaineratu zuenez , europan kontzertuaren "adiera bakarreko defentsa egin behar dute instituzioek , interes politikoak eta sinesmenak alde batera utziz ".
hermann heinzel o la peregrinación de las grullas , el hombre pingüino eta nuestra cigüeña dira dokumentalen izenburuak , eta termostato 7 frantziako film laburra .	hermann heinzel o la peregrinaci ,n de las grullas , el hombre ping sistema , eta nuestra cig lane : , eta dokumentalen izenburuak , eta termostato 7 frantziako film laburra .
euskal departamenduaren aldekoek palestina eta israel zein hego euskal herriko "zatiketa eta enfrentamendu harremana eredu "dutela adierazi zuen atzo françois bayrou pirinio atlantikoetako kontseilu nagusiko buruak .	euskal departamenduaren aldekoek palestina eta israel zein hego euskal herriko "zatiketa eta enfrentamendu harremana eredu "dutela adierazi zuen atzo fran , susana bayrou pirinio atlantikoetako kontseilu nagusiko buruak .
gutxienez historia alde dauka alavesek , coruñako deportivo ez baita mendizorrotzan irabazteko gai izan .	gutxienez historia alde dauka alavesek , coru ,ako deportivo ez baita mendizorrotzan irabazteko gai izan .
elkarri : "nafarroa bake prozesutik at nahi dute " aierdiren ustez , iruñean bake mahaia behar da	elkarri : "nafarroa bake prozesutik at nahi dute " aierdiren ustez , iruñean bake mahaia hartu behar da .
garbi dago athleticek coruñan baino zerbait gehiago erakutsi beharko duela , erasoan batez ere , bartzelonari aurre egiteko .	garbi dago athleticek coruon baino zerbait gehiago erakutsi beharko duela , erasoan batez ere , bartzelonari aurre egiteko .
juan gonzalez lahoz sendagileak atzo iruñean aditzera eman zuenez , oso litekeena da lau-sei urte barru merkatuan hiesaren aurkako botika berriak eskuratu ahal izatea .	juan gonzalez lahoz sendagileak atzo iru ,ean aditzera eman zuenez , oso litekeena da lau-sei urte barru merkatuan hiesaren aurkako botika berriak eskuratu ahal izatea .
maiatzaren 12an donibane lohizunen eta iruñean manifestazioak eginen dituzte .	maiatzaren 12an donibane lohizunen eta iru :ean manifestazioak eginen dituzte .
las desventuras de beavers : 17:00. la niña de tus ojos : 20:00 .	las desventuras de beavers : 17:00. la ni ,a de tus ojos : 20:00 .
cuadernos de cinematografía aldizkariko zuzendaria) , javier agirresarobe , michel lamarque , xabi hiriart eta aitzpea goenaga dira obra osatzen duten lanen egileak .	cuadernos de cinemato de ,a aldizkariko zuzendaria) , javier agirresarobe , michel lamarque , xabi hiriart eta aitzpea goenaga dira obra osatzen duten lanen egileak .
foru erkidegoko sozialistek garrantzi handia eman diote biltzarraren lehen bilera iruñean egiteari .	foru erkidegoko sozialistek garrantzi handia eman diote biltzarraren lehen bilera iru ,ean egiteari .
askiko litzake beartzungo errepidea luzatzea mugaraño , lau kilometrokoen bat .	askiko litzake beartzungo errepidea luzatzea mugara , lau kilometrokoen artean .
handien mailan , berriz , iñaki apalategi , nerea elustondo , iñaki gurrutxaga , gorka iribar , iñigo izagirre , beñat lizaso , jon martin , eta joseba otaegi ariko dira .	handien mailan , berriz , iaki apalategi , nerea elustondo , i ,aki gurrutxaga , gorka iribar , i , er : jon izagirre , be : , justizia lizaso , jon martin , eta joseba otaegi ariko dira .
bigarren txanda suertatu zitzaien neurtuena beti gazte eta abadiñori .	bigarren txanda suertatu zitzaien neurtuena beti gazte eta abadi ,ori .
honako mutil hauek jardun zuten atzo urdaibairen tostetan : ibon gondra , asier elkoro , jon beitia , imanol mejias , patxi egurrola , gorka olazar , jon egurrola , asier peña , iker zabala , egoitz gallo , zigor uriondo , .	honako mutil hauek jardun zuten atzo urdaibairen tostetan : ibon gondra , asier elkoro , jon beitia , imanol mejias , patxi egurrola , gorka olazar , jon egurrola , asier peñññññña , iker zabala , egoitz gallo , zigor uriondo , .
liburuaren azoka , berriz , ekainean ospatuko dute iruñeko liburu saltzaileek , urtero bezala .	liburuaren azoka , berriz , ekainean ospatuko dute iru ,eko liburu saltzaileek , urtero bezala .
izan ere , ostegunero bezala , hainbat proba egin zituen manek atzo ere , eta hamaikako hau erabili zuen ; herrera : gañan , karmona , tellez , torres mestre ; desio , pablo , astudillo , azkoitia ; magno eta salinas .	izan ere , ostegunero bezala , hainbat proba egin zituen manek atzo ere , eta hamaikako hau erabili zuen ; herrera : ga ,an , karmona , tellez , torres mestre ; desio , pablo , astudillo , azkoitia ; magno eta salinas .
bi argazki erakusketa zabaldu berri ditu iruñeko planetarioak	bi argazki erakusketa zabaldu berri ditu iru bi planetarioak .

C.1 Taula: Laugarren sistemak esaldi zuzenetarako proposatutako zuzenketak.

Bibliografia

[1] Q. DuPont, "The cryptological origins of machine translation: From al-kindi to weaver," *Amodern*, January 2018.

[2] W. J. Hutchins, "Machine translation: A brief history," in *Concise history of the language sciences*, pp. 431–445, Elsevier, 1995.

[3] W. J. Hutchins, "Chapter 2: The precursors and the pioneers," in *Machine translation: past, present, future*, pp. 5–18, Ellis Horwood Chichester, 1986.

[4] W. J. Hutchins, "Chapter 9: Strategies and methods since the mid 1960s," in *Machine translation: past, present, future*, pp. 122–135, Ellis Horwood Chichester, 1986.

[5] M. A. Chéragui, "Theoretical overview of machine translation," *Proceedings ICWIT*, p. 160, 2012.

[6] S. Sreelekha, "Statistical vs rule based machine translation; a case study on indian language perspective," *arXiv preprint arXiv*, vol. 1708, 2017.

[7] C. Boitet, "Bernard vauquois'contribution to the theory and practice of building mt systems," *Early Years in Machine Translation: Memoirs and Biographies of Pioneers*, vol. 97, p. 331, 2000.

[8] M. Kay and M. Röscheisen, "Text-translation alignment," *Computational linguistics*, vol. 19, no. 1, pp. 121–142, 1993.

[9] Z. Dajun and W. Yun, "Corpus-based machine translation: Its current development and perspectives," in *International Forum of Teaching and Studies*, vol. 11, p. 90, American Scholars Press, Inc., 2015.

[10] N.-R. Han, M. Chodorow, and C. Leacock, "Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus.," in *LREC*, 2004.

[11] J. Tetreault and M. Chodorow, "The ups and downs of preposition error detection in esl writing," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 865–872, 2008.

[12] D. Dahlmeier, H. T. Ng, and E. J. F. Ng, "Nus at the hoo 2012 shared task," in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pp. 216–224, 2012.

[13] D. Dahlmeier and H. T. Ng, "A beam-search decoder for grammatical error correction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 568–578, Association for Computational Linguistics, 2012.

[14] Y. Wu and H. T. Ng, "Grammatical error correction using integer linear programming," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1456–1465, 2013.

[15] M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. B. Dolan, D. Belenko, and L. Vanderwende, "Using contextual speller techniques and language modeling for esl error correction," in *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[16] C. Brockett, W. B. Dolan, and M. Gamon, "Correcting esl errors using phrasal smt techniques," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 249–256, Association for Computational Linguistics, 2006.

[17] T. Mizumoto, Y. Hayashibe, M. Komachi, M.Ñagata, and Y. Matsumoto, "The effect of learner corpus size in grammatical error correction of esl writings," in *Proceedings of COLING 2012: Posters*, pp. 863–872, 2012.

[18] R. Dale and A. Kilgarriff, "Helping our own: The hoo 2011 pilot shared task," in *Proceedings of the 13th European Workshop on Natural Language Generation*, pp. 242–249, Association for Computational Linguistics, 2011.

[19] H. T. Ng, S. M. Wu, Y. Wu, C. Hadiwinoto, and J. Tetreault, "The CoNLL-2013 shared task on grammatical error correction," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, (Sofia, Bulgaria), pp. 1–12, Association for Computational Linguistics, Aug. 2013.

[20] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The conll-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14, 2014.

[21] R. Grundkiewicz and M. Junczys-Dowmunt, "Near human-level performance in grammatical error correction with hybrid machine translation," *arXiv preprint arXiv:1804.05945*, 2018.

[22] D. Alikaniotis and V. Raheja, "The unreasonable effectiveness of transformer language models in grammatical error correction," *arXiv preprint arXiv:1906.01733*, 2019.

[23] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The bea-2019 shared task on grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75, 2019.

[24] J. Allan, J. Aslam, N. Belkin, C. Buckley, J. Callan, B. Croft, S. Dumais, N. Fuhr, D. Harman, D. J. Harper, *et al.*, "Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002," in *ACM SIGIR Forum*, vol. 37, pp. 31–47, ACM New York, NY, USA, 2003.

[25] D. Jurafsky and J. H. Martin, *Speech and Language Processing.*, ch. Chapter 3: N-gram Language Models. October 2019.

[26] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[27] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," in *Twelfth annual conference of the international speech communication association*, 2011.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[30] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.Ñ. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[32] I. Aldabe, I. Aldezabal, M. Aranzabe, B. Arrieta, A. Díaz de Ilarraza, K. Gojenola, M. Maritxalar, M. Oronoz, and A. Otegi, "Euskarazko erroreen sailkapena,"

[33] M. Oronoz, *Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura*. PhD thesis, 2008.

[34] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," *arXiv preprint arXiv:1906.04341*, 2019.

[35] C.Ñapoles, K. Sakaguchi, M. Post, and J. Tetreault, "Ground truth for grammatical error correction metrics," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 588–593, 2015.

[36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.