

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Galderen sorkuntza ikasketa sakoneko tekniken bidez

Egilea

Maite Urrea Gorospe

Zuzendariak

Oier Lopez de Lacalle

Montse Maritxalar

Laburpena

Proiektu honetan ikasketa sakoneko tekniken bidezko galderen sorkuntza egiten da, galderen sorkuntzak alde pedagogikotik dakarren motibazioarengatik bultzatuta. Ikasketa sakonean oinarritutako metodoak ondo ulertzeko beharrezko oinarri teorikoaren azterketa sakona egin da, bai eta arkitektura espezifiko baten berrinplementazioa. Etorkizunean motibazio pedagogikoa helburu izanda, ikasketa sakonean oinarritutako galderen sorkuntzak izan ditzakeen lan-lerroak aztertzen dira baita ere.

Gaien aurkibidea

Laburpena	i
Gaien aurkibidea	iii
Irudien aurkibidea	vii
Taulen aurkibidea	ix
1 Sarrera	1
2 Galderen sorkuntza	3
2.1 Motibazio pedagogikoa	3
2.2 Galderen sorkuntza automatikoa	5
2.2.1 Esaldi mailan	7
2.2.2 Paragrafo mailan	9
2.2.3 Konzeptu-mapa batetik abiatuta	11
2.3 Ikasketa sakonaren bidezko galderen sorkuntzarantz	13
3 Ikasketa sakonaren bidezko galderen sorkuntza	15
3.1 Sare neuronalak	17
3.1.1 Neurona artifiziala	18
3.1.2 Aktibazio-funtzioak	18
3.1.3 Arkitekturak	20
3.2 Sare neuronalen ikasketa	22
3.2.1 Ikasketa-paradigmak	22
3.2.2 Kostu-funtzioak	23
3.2.3 Optimizazio-algoritmoak	25
3.2.4 Ereduaren orokortzea	29
3.3 Sare neuronal errekurrenteak	31

3.3.1	Sare errekurente simplea	32
3.3.2	LSTM	34
3.3.3	GRU	36
3.4	Kodetzaile-dekodetzaile arkitektura	37
3.4.1	Kodetzailea	37
3.4.2	Dekodetzailea	37
3.5	Atentzio-mekanismoa	39
3.6	Hitzen errepresentazioa	41
3.6.1	Indexatutako hiztegia	41
3.6.2	One-hot kodeketa	42
3.6.3	Esanahi-bektoreak	42
4	Diseinua eta inplementazioa	45
4.1	Datu-multzoa	45
4.2	Aurreprozesaketa	46
4.2.1	Datu-multzoaren aurreprozesaketa	46
4.2.2	Hiztegia	48
4.2.3	Esanahi-bektoreak	49
4.3	Arkitektura	49
4.3.1	Bi norantzako LSTMa	49
4.3.2	Pilaratutako LSTMa	49
4.3.3	Arkitekturaren xehetasunak	50
4.4	Entrenamendua	51
4.4.1	Kostu-funtzioa eta optimizazio-algoritmoa	51
4.4.2	Entrenamendua errazteko estrategiak	52
4.4.3	Eredu-aukeraketa	54
4.5	Sorkuntza	54
4.5.1	<i>Beam search</i>	55
4.5.2	Kopiatzeko mekanismoa	56
4.6	Inplementazioa	56
5	Ebaluazioa eta emaitzak	57
5.1	Ebaluazio automatikoa	57
5.1.1	BLEU	57
5.1.2	METEOR	58
5.1.3	ROUGE	59
5.1.4	Galderen erantzugarritasunean oinarritutako metrika	60
5.2	Eskuzko ebaluazioa	61
5.3	Esperimentazioa eta emaitzak	62

6	Hobekuntzak eta proposamenak	67
6.1	Galderen sorkuntzaren artearen egoeraren hobekuntzak	67
6.1.1	Hobekuntza orokorrak	68
6.1.2	Hobekuntzak arlo pedagogikoan	71
6.2	Hobekuntza berrien proposamena	72
6.2.1	Transformerra	72
6.2.2	Testuinguruaren araberako esanahi-bektoreak	73
7	Ondorioak eta etorkizunerako lana	77
7.1	Ondorioak	77
7.1.1	Proiektuaren ondorioak	77
7.1.2	Ondorio pertsonalak	78
7.2	Etorkizunerako lana	79
Eranskinak		
A	Proiektuaren Helburuen Dokumentua	83
A.1	Proiektuaren deskribapena eta helburuak	83
A.2	Proiektuaren plangintza	83
A.2.1	LDE diagrama	83
A.2.2	Lan-paketeak	84
A.2.3	Emangarriak	86
A.2.4	Mugarriak	87
A.2.5	Gantt diagrama	87
A.3	Lan Metodologia	88
A.3.1	Bilerak	88
A.3.2	Planifikatutako ordutegiak	88
A.4	Bideragarritasuna	88
A.5	Arriskuak eta prebentzioa	89
A.5.1	Arriskuak	89
A.5.2	Prebentzioa	89
Bibliografia		91

Irudien aurkibidea

2.1	Bloom-en taxonomia berrikusia (LW et al., 2001)	5
2.2	Testutik galderarako atazaren deskonposaketa	6
2.3	Heilman and A. Smith-en sistemaren galderak sortzeko faseak	7
2.4	Galdera-transduktorearen pausoak (Heilman and A. Smith, 2009). * ikur- rarak pauso horretan zero edo hainbat irteera egon daitezkeela adierazten du; parentesiek, berriz, pausoa ez dela beharrezkoa zenbait galdera sor- tzeko.	8
2.5	Esaldi bati galdera transduktorearen pausoak aplikatzearen adibidea (Heil- man and Smith, 2010)	9
2.6	Galderak sortzeko faseak (Mannem et al., 2010)	10
2.7	Kontzeptu-maparen adibidea. Termino-gakoak nodo beltzak dira.	12
3.1	Sare neuronal	17
3.2	Sare neuronal batek definitutako funtzioa	17
3.3	Neurona artifiziala	18
3.4	<i>Multilayer perceptron</i>	20
3.5	Sare artifizial konboluzionala	21
3.6	Sare errekorrentea	21
3.7	Gradientearen jaitziera aldagai bakarreko funtzioan	26
3.8	<i>Backpropagation</i> adibiderako sare neuronal eta ekuazioak	28
3.9	<i>Overfitting</i> eta <i>underfitting</i> sarearen ikasketan	30
3.10	Sare errekorrentea	31
3.11	Sare errekorrente destolestua	31
3.12	Sare errekorrente sinplea	32
3.13	<i>Backpropagation through time</i>	33
3.14	W_h -ren eraginaren grafo ordenatua	33
3.15	LSTM	35

3.16	Kodetzailea	38
3.17	Dekodetzailea	38
3.18	Kodetzaile-dekodetzaile arkitektura atentzio mekanismoarekin	40
3.19	<i>One-hot encoding</i>	42
3.20	<i>man, woman, king, queen</i> hitzen arteko analogia	43
4.1	Esaldi-galdera pareen arteko <i>non-stop</i> hitzen arteko gainezarpen proportzioa. (Du et al., 2017)	47
4.2	Bi norantzako LSTMa	50
4.3	Pilaratutako LSTMa	50
4.4	<i>Padding</i>	52
4.5	Dropout	53
4.6	<i>Teacher forcing</i>	53
4.7	<i>Beam search</i> adibidea (<i>beam size= 2</i>)	55
5.1	Gida-lerroak ebaluatzeko <i>Pybossa</i> -rako lehenengo bertsioa	63
5.2	Perplexity-aren konparaketa	65
6.1	Transformer (Vaswani et al., 2017)	73
A.1	Lanaren Deskonposaketa Egitura	84
A.2	Gantt diagrama	88

Taulen aurkibidea

2.1	Galdera sorkuntzarako datu-multzoen laburpena (Pan et al., 2019)	14
3.1	Indexatutako hiztegi baten adibidea	41
5.1	Emaitzak	64
5.2	Sarrerako hiztegien konparaketa	65
5.3	Irteerako hiztegien konparaketa	65
5.4	Du et al. eta gure ereduaren balidazioko <i>perplexity</i> -aren eboluzioen konparaketa	66
5.5	Gure ereduarekin lortutako galdera batzuen adibidea	66
6.1	Galderen sorkuntzaren artearen egoera (Pan et al., 2019). QW : galdehitzen sorkuntza, PC : paragrafo mailako testuingurua, CP : kopiatzeko mekanismoa, LF : ezaugarri linguistikoak, PG : <i>policy gradient</i>	71
A.1	Lan-pakete bakoitzari esleitutako denbora	87
A.2	Mugarriak	87

1. KAPITULUA

Sarrera

Galderak egitea jakin-mina, sormena, pentsamendu kritikoa eta ikerketarako jarrerarekin estu lotuta dagoen gizakion izaeraren berezko jarduera da. Hurrei begiratu besterik ez da egin behar, hitz egiten ikasten dutenetik milaka galdera egiten baitituzte. Eguneroko bizitza izaera desberdineko galderetaz dago beteta: “Zer gosalduko dut gaur?”, “Zein da bizitzaren zentzua?” eta abarrekoak. Hemen ikuspegi pedagogikotik esanguratsuak diren galderetan zentratuko gara.

Galderak aspalditik erabili izan dira ikasketa-prozesuan. Antzinako Grezian filosofoek asko erabilitako metodoa da. Horren eredu Sokratesek sortutako metodo sokratikoa daukagu, non irakasleak, galderen erabileraren bitartez, ikaslea ezagutza lortzera laguntzen duen.

Galdera desberdinek maila desberdineko gaitasun kognitiboak eskatzen dituzte: gogortzea, ulertzea, hausnartzea eta abar. Galdera on eta sakonak egitea ez da gauza erraza; denbora eta praktikarekin hobetzen den jarduera da. Gainera, gaur egun dagoen hezkuntza-sisteman hobekuntzak egin badira ere, normalean, ez da ikaslearen jakin-mina pizten ezta galderak egitera bultzatzen.

Gauzak horrela, maila kognitibo desberdineko galderak era automatikoan sortzen dituen sistema bat garatzea interes handikoa litzateke. Hori bai, kontuan hartuz sistema ikasketa-prozesuan lagungarri den erreminta baino ez dela eta ez helburua; hau da, sistemak sortutako galderak erantzutea ez da helburu bakarra izango, ikasleak berak galdera horiek bere kabuz sortzen ikasi behar du ere. Galdera pedagogikoak testuinguru desberdinetan sor badaitezke ere, hemen testu bat emanik testuaren informazioarekiko erlazioa duten

galderak sortzean zentratuko gara. Baina nola egin dezake ordenagailu batek galdera bat sortzeko?

Orain arte garatutako galderen sorkuntzarako sistemak eskuz egindako erregeletan oinarritzen dira, eta gogoratzea (maila kognitibo baxuko gaitasuna) eskatzen duten galderak sortzen dituzte bakarrik. Galderen sorkuntza automatikorako sistema ideala lortzeko bidean alde pedagogikotik eman beharreko hurrengo pausoa ulermena lantzen duten galderak sortzea litzateke. Gainera, azken urteotan garrantzi handia hartu duten ikasketa sakoneko teknikak datu-multzo handietan orokortzeko daukaten gaitasuna dela eta, galderen sorkuntza automatikoan ikerketa-lerro berria zabaldu da.

Proiektu honetan ikasketa sakoneko teknikak erabiliz garatutako ingeleseko galderen sorkuntzarako eredu baten berrinplementazioa egiten da. Eredu horren funtzionamendua eta intuizioa ulertzeko beharrezkoak diren oinarrizko kontzeptu teorikoak era sakon batean lantzen dira; izan ere, etorkizunean ereduaren gainean aldaketak egin nahi izanez gero premiazkoa delako bere xehetasun guztiak ondo kontrolatzea. Azkenik, galderen sorkuntzaren artearen egoeran azkenengo urteetan egin diren hobekuntza berrien azterketa egiten da, bai eta zenbait hobekuntza berrien proposamena.

Laburbilduz, proiektu honen helburua ikasketa sakonaren bidezko galderen sorkuntzarako eredu bat berrinplementatzea da, horren funtzionamenduaren ulermen sakona lortu eta arlo pedagogikotik begira ereduari egin dakizkioken hobekuntzen azterketa egitea.

Memoria honetan proiektuaren nondik norakoak azaltzen dira, eta hainbat kapitulutan dago antolatuta. **2** kapituluan galderen sorkuntzarako motibazio pedagogikoa azaltzen da, bai eta, duela gutxi arte erabilitako erregeletan oinarritutako tekniken azterketa orokorra, eta ikasketa sakonean oinarritutako teknikak erabiltzen hastearen zergatia. **3** kapituluan galderen sorkuntza ikasketa sakonaren bidez egiteko ulertu beharreko oinarri teorikoak lantzen dira. **4** kapituluan proiektu honetan berrinplementatu den sistemaren diseinua eta xehetasunak aztertzen dira. **5** kapituluan berrinplementatutako sistema ebaluatzeko erabilitako metrikak ikusten dira, bai eta lortutako emaitzen azterketa. **6** kapituluan ikasketa sakonean oinarritutako galderen sorkuntzan azken urteetan egin diren hobekuntzen azterketa egiten da, eta beste zenbait hobekuntza berri proposatzen dira. Azkenik, **7** kapituluan proiektutik ateratako ondorioak eta etorkizunerako lana jasotzen dira.

2. KAPITULUA

Galderen sorkuntza

Kapitulu honetan galderen sorkuntzaren inguruko zenbait aspektu lantzen dira. Alde batetik, alde pedagogikotik galderek daukaten garrantzia aztertzen da. Beste aldetik, duela gutxi arte galderen sorkuntza automatikorako erabilitako erregeletan oinarritutako teknikak azterketa orokorra egiten da. Azkenik, galderen sorkuntzarako ikasketa sakonean oinarritutako teknikak erabiltzen hastearen zergatia eta emandako lehenengo pausoak aztertzen dira.

2.1 Motibazio pedagogikoa

Galderak pertsonen ezagutza ebaluatzeko baino balio ez dutela zabaldua dagoen pentsaera da ([Karpicke and J. Grimaldi, 2012](#)). Askotan ikasketa-prozesua informazioa eskuratu, kodetu eta gorde egiten duen prozesua bezala ikusi da, eta, beraz, informazioa kodetzeko erari erreparatzen zaio ikasketa-prozesua hobetzeko. Testuinguru horretan, galderak egitea ezagutza dagoen edo ez ikusteko baino ez dira erabiltzen.

Beste ikuspegi bat dago non ikastea iraganeko esperientziak orainean erabili ahal izateko gaitasuna bezala definitzen den. Ikuspegi horren arabera, ikasketa-prozesuan informazioaren kodeketa ez ezik, informazioaren berreskurapena ere ezinbestekoa suertatzen da. Hainbat ikerketa egin dira non berreskurapenaren praktika ikasketa-prozesuan lagungarria eta eraginkorra dela ondorioztatzen den ([Karpicke and J. Grimaldi, 2012](#); [Weinstein et al., 2018](#)). Galderak informazioaren berreskurapena lantzeko oso aproposak dira; hori

dela eta, galderen sorkuntza interes handiko gai bat izan da ikerkuntzan, bai arlo kognitiboan bai arlo pedagogikoan.

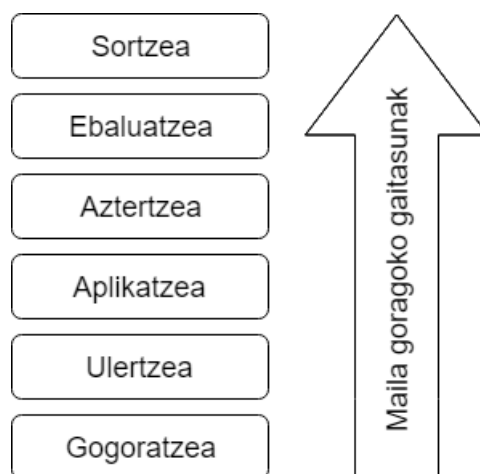
Berreskurapenaren praktikak burmuineko epe luzerako memoriatik lan memoriara eramateko prozesua lantzen du. Horrek ikasitakoa gehiago finkatzen du, etorkizunean informazio hori berreskuratzea errazago eginez, onurak ekarriz memorian, informazioaren transferentzian eta inferentzian. Gainera, berreskurapena arrakastatsua izan den edo ez ikusteak ikasketa-prozesuan zertan zentratu behar den jakitera laguntzen du; hau da, informazio baten berreskurapena egitea lortu ez bada edo gaizki egin bada, informazio hori hobeto ikasi behar dela esan nahi du.

Hala ere, berreskurapenaren praktika eraginkorra izateko zenbait baldintza bete behar dira. Alde batetik, berreskurapenaren arrakasta oso baxua bada, eta, gainera, berreskuratu beharreko informazioa zein den egiaztatzen ez bada, teknikak ez du onurarik ekartzen, baizik eta kontrakoa. Bestetik, berreskurapena arrakasta bermatuta dagoen egoera batean egiten bada, adibidez, esaldi bat irakurri bezain laster galdera egiten bada, litekeena da berreskurapena ondo egitea, baina ez epe luzerako memoriatik berreskuratu delako, baizik eta lan memorian denbora luzeagoz mantendu delako. Azkenik, aztertu da berreskurapen-prozesuan ikaslearengan nolabaiteko presioa badago, teknikaren eraginkortasuna jaitسي daitekeela ([Hinze and Rapp, 2014](#)), baina ikasitako informazioa galdetua izango dela pentsatzeak ere onurak ekar ditzake ([Weinstein et al., 2014](#)).

Beraz, zerbait ikasteen informazio hori geroago galdetua izango dela gogoratuz, galdera horiek era estrategiko batean denboran zehar sakabanatuz eta informazio horren berreskurapenean inolako presiorik ez dagoela ziurtatuz, ikasketa-prozesurako oso teknika eraginkorra lortzen da galderen bitartez.

Berreskurapenaren praktikarako ez ezik, hausnarketara eramaten duten galderak erabiltzea oso lagungarria denaren ebidentzia aurkitu da baita ere ([Dunlosky et al., 2013](#)), ([Pashler et al., 2007](#)). Normalean, galderek eskatzen duten erantzuteko maila kognitibo neurtzeko [Bloom et al.-k \(1956\)](#) sortutako taxonomia hartzen da abiapuntu bezala. Taxonomia horrek zenbait berrikuste izan ditu, azkenengoa [LW et al.-k \(2001\)](#) egindako da. Bertan, gaitasun maila desberdineko sei prozesu kognitibo definitzen dira: gogoratzea, ulertzea, aplikatzea, aztertzea, ebaluatzea eta sortzea ([2.1 irudia](#)).

Ikasketa-prozesuan galderen erabileraren onuren ebidentzia handiak aurkitu badira ere, bai ikasgeletan bai ikasketa autonomoan gutxi erabiltzen den teknika da. Hainbat ikerketek ondorioztatu dute, oro har, gizakion galderak egiteko abilezia ez dela oso ona: ikasketa-prozesuan norberaren ezagutzaren defizientziak identifikatzea zaila egiten da



2.1 Irudia: Bloom-en taxonomia berrikusia (LW et al., 2001)

pertsona gehienontzat (Hacker et al., 1998), eta, gainera, bai ikasleek bai irakasleek galdera gutxi egiten dituzte (Graesser and Person, 1994). Bestalde, galderak egiteaz gain, horiek kalitatezkoak izatea premiazkoa da. Aipatutako arrazoi guztiak direla eta, galdera sorkuntzarako sistema automatizatuak onura handikoak izan daitezke.

Sistema automatizatuak sortutako galderak aipatutako Bloom-en taxonomiako maila kognitibo desberdineko galdera onak sortzea litzateke helburua. Baina hori lortzea ez da gauza erreza; izan ere, horretarako, alde batetik, sistemak sarrerako informazioaren ulermen sakona lortu behar du, eta, bestetik, sortutako galderak gramatikalki bai semantikoki egokiak izan behar dira.

Aztertuko den bezala, orain arte egindako sistemak egitatei buruzko galderetan oinarritzen dira; hau da, gogoratzea bakarrik eskatzen duten galderetan. Aipatu da galdera horiek oso erabilgarriak direla; hala ere, erantzuteko maila goragoko gaitasunak eskatzen dituzten galderak sortzen dituen sistema lortzea interesgarria izango litzateke. Beraz, hori lortzeko hurrengo pausoa informazioaren ulermena eskatzen duten galderak sortzea da.

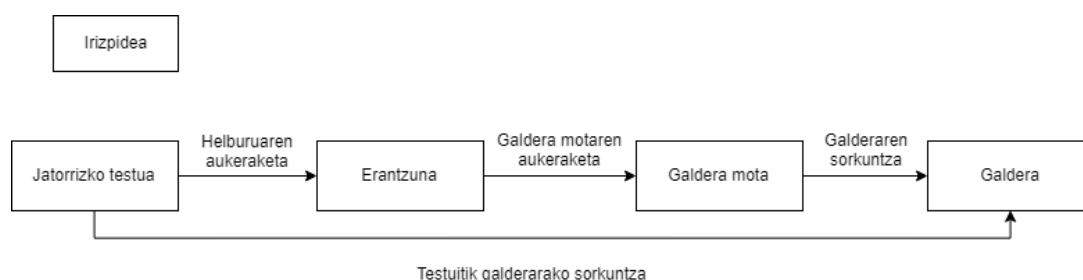
2.2 Galderen sorkuntza automatikoa

Galderen sorkuntzak Hizkuntzaren Prozesamenduko ikertzaileen artean eragin du interesa baita ere. Ez hainbeste motibazio pedagogikoarengatik, baizik eta galderen sorkuntza beste ataza batzuetarako erabilgarria delako. Elkarrizketa sistemetan, adibidez, *chatbot* bat gai izan behar da modu arin batean galderak sortzeko, bai elkarrizketa bat hasteko edota informazio espezifiko eskatzeko. Galderen sorkuntzaren atzarekin oso lotuta dagoen

galdera-erantzun sistementzako ere oso erabilgarria da; sistema horien helburua testu bat eta galdera bat emanik horren erantzuna aurkitzea da; beraz, galderen sorkuntza sistema horiek lortzeko behar diren datu-multzoak sortzeko erabil daiteke.

Galderen sorkuntza automatikoari bultzakada bat emateko 2009. urtean egin zen galderen sorkuntzaren inguruko 3. tailerrean [Rus and Graesser-ek \(2009\)](#) *The Question Generation Shared Task and Evaluation Challenge (QGSTEC)* sortu zuten. QGSTEC-aren helburua diziplina desberdineko ikertzaileak batzea zen, galderen sorkuntzaren inguruko oinarrizko aspektuak landu eta etorkizunerako lan plangintza bat egiteko. Bertan horrela definitu zen galderen sorkuntza: informazio-iturri bat eta galdera onak zer diren ebaluatzeko irizpide bat emanik, informazioari buruz galdetzen duten eta irizpidea betetzen duten galderak sortzea da.

QGSTEC-en galderen sorkuntzaren inguruko lau ataza landu ziren: testutik galderarako ataza, tutorial elkarrizketa ataza, ebaluazio-ataza eta kontsultatik galderarako ataza. Hemen testutik galderarako ataza aztertzen da ([2.2 irudia](#)).



2.2 Irudia: Testutik galderarako atazaren deskonposaketa

Tradizionalki galderen sorkuntza eskuz egindako erregeletan oinarritutako metodoekin egin da. Metodo horietan, oro har, sarrerako testua analizatzaile batekin prozesatzen da eta eskuz definitutako zenbait erregela erabiliz sortzen dira galderak.

Hemen, erregeletan oinarritutako galderen sorkuntzaren ikuspegi orokor bat izateko, sarrera mota desberdinetan oinarritzen diren hiru hurbilpen desberdin aztertuko dira, horretarako ingeleseko galderak sortzen dituzten hiru metodo zehatz azalduz: esaldi mailan ([Heilman and A. Smith, 2009](#)), paragrafo mailan ([Mannem et al., 2010](#)) eta kontzeptu-mapa batetik abiatuta ([M. Olney et al., 2012](#)).

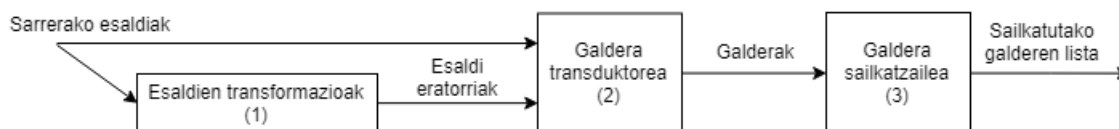
Hala ere, erregeletan oinarritutako era desberdineko hurbilpen asko sortu dira, bai eta euskararako ere. [Aldabe et al.-ek \(2011\)](#) zenbakizko entitatei buruzko euskarazko galderak

sortzeko sistema aurkezten dute. [Aldabe et al.-ek \(2013\)](#) euskarazko galderak sortzeko bi hurbilpen proposatzen dituzte, bat *chunk*-etan oinarritua eta bestea dependentzia sintaktikoetan. Sistema hauek [López Gazpio-k \(2013\)](#) eta [Madrado Azpiazu-k \(2013\)](#) Gradu Amaierako Lanetan aurkeztutakoak dira. [López Gazpio-k](#) *chunk*-etan oinarritutako hurbilpen bat implementatzeaz gain, [Heilman and A. Smith-ek](#) proposatzen duten moduan, sortutako galderen artean lehentasun bat ezartzen du. [Heilman and A. Smith-ek](#) ez bezala, lehentasuna *chunk* mailan egiten da eta ez esaldi mailan. [Madrado Azpiazu-k](#) berriz, dependentzietan oinarritutako galderen sorkuntza burutu zuen. Bere Gradu Amaierako Lanean galdera sortzaile bat aurkezteaz gain, testu bat emanda, testu horretako termino esanguratsuak ere antzeman zituen, [M. Olney et al.-ek](#) proposatutako hurbilpenaren lehenengo pausuak ezarriz. Bertan, galderak kontzeptu-mapa batetik sortzen dira.

2.2.1 Esaldi mailan

[Heilman and A. Smith-ek \(2009\)](#) esaldien ganean galderak sortzeko metodoa proposatzen dute. Galderen sorkuntza irakurrizko ulermena ebaluatu eta praktikatzeko bideratzen dute eta egitatei buruzko galderak sortzean zentratzen dira. Horretarako, helburu orokorreko erregelak erabiltzen dituzte esaldi deklaratiiboak galderetara transformatzeko. Gainera, *gainsortu eta sailkatu* teknika erabiltzen dute; hau da, sorkuntzan hainbat galdera sortu eta haien sailkapen bat egiten dute galderarik hoberenak aukeratzeko.

Galderak sortzeko prozesua hiru pausotan banatzen dute (2.3 irudia): sarrerako esaldiaren transformazioa, galdera-transduktorea eta galderen sailkapena. Esaldien egitura sintaktikoa irudikatzen duen Peen Treebank erako zuhaitz sintaktikoak erabiltzen dituzte kategoria gramatikalarekin batera.



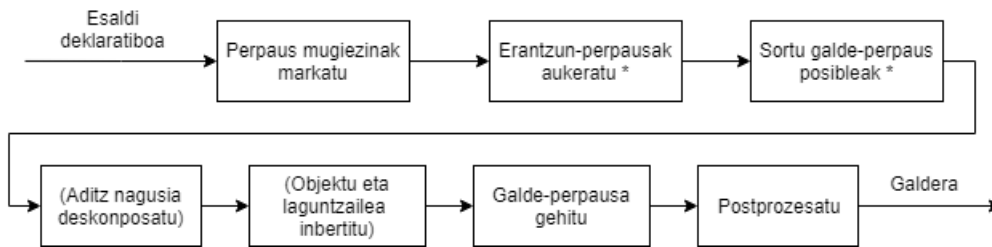
2.3 Irudia: [Heilman and A. Smith-ek](#) sistemaren galderak sortzeko faseak

Esaldien transformazioa

Sarrerako esaldia transformatzen da esaldi deklaratiiboa sinplifikatuz. Horretarako, testu laburpen eta konpresioan erabiltzen diren zenbait teknika aplikatzen dira, besteak beste, perpausen eta aposizioen sinplifikazioa, eta konjuntzioen biltzea.

Galdera-transduktorea

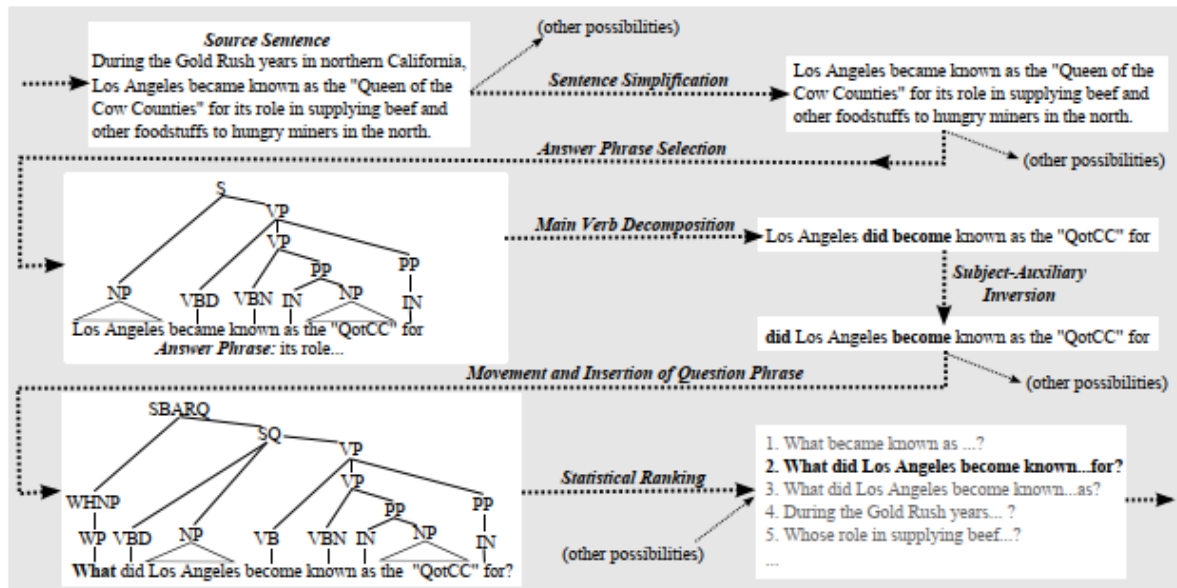
Esaldi deklaratihoa galdera batera transformatzen da ondo definitutako transformazio sintaktikoak eginez. Horretarako, ondoko pausoak ematen dira (2.4 irudia):



2.4 Irudia: Galdera-transduktorearen pausoak (Heilman and A. Smith, 2009). * ikurrak pauso horretan zero edo hainbat irteera egon daitezkeela adierazten du; parentesiek, berriz, pausoa ez dela beharrezkoa zenbait galdera sortzeko.

- Perpaus mugiezinak markatu: erantzuna izan ezin diren perpausak markatzen dira. Perpaus horien nodoen etiketak aldatzen dira esaldiaren zuhaitz sintaktikoan.
- Erantzun-perpausak aukeratu: erantzuna izan daitezkeen perpausak bilatzen dira. Erantzun-perpaus bakoitzeko zuhaitz sintaktikoaren kopia bat egiten da, zuhaitz bakoitzean dagokion erantzun-perpausa ezabatuz.
- Galde-perpaus posibleak sortu: erantzun-perpaus bakoitzeko galde-perpaus posibleak bilatzen dira. Galde-perpausa galde-hitz batek osatzen du.
- Aditz nagusiaren deskonposaketa: objektu-laguntzaile inbertsioa egin ahal izateko aditz laguntzailea edo modala agertzen ez bada, aditz nagusia deskonposatzen da.
- Objektu-laguntzaile inbertsioa: objektu-laguntzaile inbertsioa egiten da bai-ez era-ko galderetan edota subjektua ez daukaten izen-sintagmetan.
- Galde-perpausa gehitu: galde-perpaus bakoitza zuhaitz sintaktikoaren kopia batera gehitzen da galdera sortzeko.
- Postprozesaketa: formatu eta puntuazio egokiak bermatzeko egiten den postprozesaketa.

2.5 irudian sarrera esaldi bati galdera transduktorearen pausoak aplikatzearen adibide bat azaltzen da (Heilman and Smith, 2010).



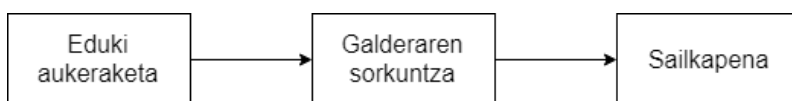
2.5 Irudia: Esaldi bati galdera transduktorearen pausoak aplikatzearen adibidea (Heilman and Smith, 2010)

Galdera sailkatzailea

Esaldi bakoitzeko hainbat galdera sortzen dira, haietatik askok erroreak dituzte; beraz, galdera guztien arteko sailkapen bat egiten da galdera hoberenak hartzeko. Galderen sailkapena egiteko Collins and Koo-ek (2005) proposatutako sailkatzaile diskriminatzailea erabiltzen dute. Erregresio logistikoko eredu batean oinarrituz sarrera testua eta galdera emanik onargarrtasun probabilitate bat kalkulatzen da, eta horren arabera sailkatzen dira galderak.

2.2.2 Paragrafo mailan

Mannem et al.-ek (2010) garatutako sistema "The Question Generation Shared Task Evaluation Challenge 2010 (QGSTEC'10)" lehiaketan aurkeztutakoa da. Bertan, paragrafoetatik galderak sortzeko atazan parte hartu zuten. Sistemak, paragrafo bat emanik, paragrafo horretako sei galdera esanguratsuenak sortu behar zituen. Beraz, sorkuntza esaldi mailan izan arren, esaldien esanahia ulertzea bilatzen duen sistema da eta horren arabera, sei esanguratsuenak lortu. Proposatzen duten galdera sorkuntzarako sistemak hiru fase ditu (2.6 irudia): eduki aukeraketa, galderen sorkuntza eta sailkapena.



2.6 Irudia: Galderak sortzeko faseak (Mannem et al., 2010)

Eduki aukeraketa

Sortuko diren galderen erantzuna diren paragrafoko atalak identifikatzen dira horretarako zenbait irizpide jarraituz:

- Derrigorrezko argumentuak: paragrafoko esaldi bakoitzeko predikatuko derrigorrezko argumentuak erantzun posible bezala kontsideratzen dira, non derrigorrezko argumentuak rol semantikoa duen argumentua den.
- Aukerako argumentuak: zenbait aukerako argumentu informatiboak dira, egokiak eginez galderaren erantzuna izateko.
- Aditz kopulatiboak: Aditzaren eskuineko argumentua ia beti aproposa izaten da erantzuna izateko, beti ere esaldi existentziala ez bada. Adibidez, hondoko esaldia izanik: “*Latent semantic analysis (LSA) is a technique in natural language processing*”, aditzaren eskuineko argumentua: “*a technique in natural language processing*”, galdera izateko hautagai ona da.

Galderaren sorkuntza

Pauso honetan, lehenengo, aditz-gune nagusia identifikatzen da, eta, ondoren, paragrafoko esaldi bakoitzari zenbait transformazio aplikatzen zaizkio galdera sortzeko.

- Aditz-gune nagusiak identifikatu: aditz-gunea aditz nagusia eta berarekin ager daitezkeen laguntzaile eta modalek osatzen dute. Aurreko pausoen identifikatutako erantzun posibleen predikatuan aditz-gunea bilatzen da, galdera sortzeko egin behar den transformazioa egiteko beharrezkoa delako.
- Transformazioak: predikatua eta erantzun helburuaren arabera hainbat transformazio aplikatzen dira galdera sortzeko.

Paragrafoko lehenengo esaldia galdera orokorra sortzeko erabiltzen da. Galdera orokor horren irismena paragrafo osokoa izango da.

Sailkapena

Sortutako galdera guztiekin zerrenda sailkatua egiten da eta sei hoberenak hartzen dira bakarrik. Galderak bi pausotan sailkatzen dira.

- Lehenengo fasea: galderak predikatuaren zuhaitz sintaktikoaren sakoneraren arabera sailkatzen dira.
- Bigarren fasea: sailkapen berdina jaso duten galderak izenordain kopuruaren arabera ordenatzen dira, izenordain gutxiago dutenei puntuazio altuagoa emanez.

2.2.3 Konzeptu-mapa batetik abiatuta

[M. Olney et al.](#)-ek (2012) testuaren edukia adierazteko kontzeptu-mapa batean oinarritzen den metodoa proposatzen dute. Metodoa bi pausotan banatzen da: lehenengo, domeinu baterako kontzeptu-mapa sortzen da testuetatik abiatuta era automatiko batean, eta, ondoren, kontzeptu-mapa erabiltzen da galderak sortzeko, horretarako, eskuz egindako erregela-patroiak erabiliz.

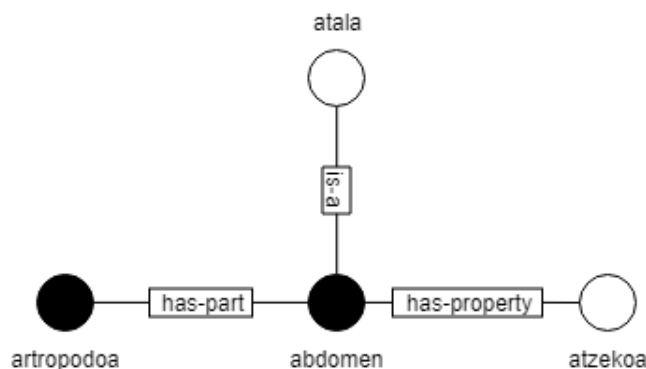
Erauzle automatikoa

Erauzle automatikoak *termino-erlazio-termino* hirukoteak sortzen ditu erauzteko zenbait algoritmo erabiliz, eta kontzeptu-mapa batez baliatzen da hauek adierazteko. Testu batean informazioa testu osoan zehar egon daiteke sakabanatuta. Testu osoko kontzeptu-mapa bat sortzearen abantaila zera da: nahiz eta testuan oso bereizirik agertu, termino berdinari buruzko bi hirukote erlazionaturik daude.

Erabiltzen duten kontzeptu-mapak bi elementu nagusi ditu:

- Termino-gakoak: pedagogikoki adierazgarriak diren terminoak dira; izan ere, galdera esanguratsuak galdetzeko gaiaren inguruko ideia gakoak identifikatu behar dira. Hirukote baten hasiera termino-gako bat izan behar da; amaierako nodoa, aldiz, termino-gako, beste hitz batzuk edo proposizio oso bat izan daiteke.
- Erlazio-ertzak: terminoen arteko erlazioak adierazten dituzte; egileek 252 erlazio desberdin identifikatu eta eskuz egindako multzokatze bat egin ondoren 30 erlazio erabiltzea erabakitzen dute.

Termino-gakoak eta erlazio-ertzak 2.7 irudiko adibidean agertzen den bezala erlazionatzen dira. Irudiko termino-gakoak *artropodoa* eta *abdomen* dira, erlazio-ertzak *has-part*, *has-property* eta *is-a*; *atala* eta *atzekoa* amaierako nodoak dira. Beraz, erauzle automatikoak sortutako hirukoteak *artropodoa-has-part-abdomen*, *abdomen-is-a-atala* eta *abdomen-has-property-atzekoa* liriateke eta horietatik abiatuta irudiko kontzeptu-mapa sortzen da.



2.7 Irudia: Kontzeptu-maparen adibidea. Termino-gakoak nodo beltzak dira.

Galderen sorkuntza

Galderak sortzeko bi eratarara egin da, batetik hirukote indibidualak erabiliz eta bestetik zenbait hirukote konbinatuz. Azkeneko erak kontzeptu-maparen gaineko nolabaiteko arrazoiketa sartzen du.

- Hirukote indibidualak: hirukote indibidualak erabiliz espezifikotasun maila desberdineko galderak sortzen dira; hala nola, aholkuak, oharrak, etab. Horiek hirukotearen ertz erlazioaren arabera galdera-ereduak erabiliz sortzen dira.
- Zenbait hirukote: zenbait hirukote erabilia hiru galdera mota sortzen dira: testuinguru egiaztapena, aukeraketa derrigortua eta erregela kausala.

Testuinguru-egiaztapenean testuinguru bat erakutsi eta ondoren egiaztapen-galdera bat egiten da. Adibidez: “*An ecosystem is a community. Is that true for a forest?*”

Aukeraketa derrigortuan propietate bat bi terminotako zeinek betetzen duen galde-tzen da. Adibidez: “*What resides in skin, melanin or phytochrome?*”

Erregela kausala kontzeptu-mapako zenbait nodoren arteko erlazio kausalean oinarritzen da. Adibidez: “*How do bacteria produce energy?*” galdera horrek *bacteria*, *food* eta *energy* termino-gakoak erlazionatzen ditu.

2.3 Ikasketa sakonaren bidezko galderen sorkuntzarantz

Erregeletan oinarritutako metodoek egitateei buruzko galderetan nahiko emaitza onak lortzen badituzte ere, eskuz egindako erregeletan oinarritzen dira. Erregela horiek sortzea lan neketsua da eta adituen ezagutza beharrezkoa da. Gainera, sortutako erregelak beste domeinu batean aplikatu nahi direnean, gehienetan ez dira aplikagarriak beste domeinu horretara.

Maila goragoko gaitasunak behar dituzten galderak sortzeko, sarrerako informazioaren ulermen sakona egiten duten sistemak behar dira eta erregeletan oinarritutako metodoek ez dute hori egiten. Beraz, hori lortzeko beste erako metodoak bilatzea ezinbestekoa suertatzen da.

Hori dela eta, eta makina bidezko itzulpenean sare neuronaletan oinarritutako hurbilpenek (Sutskever et al., 2014) izandako arrakasta dela medio, Serban et al.-ek (2016) lehenengo aldiz erabili zuten ikasketa sakona galderen sorkuntza egiteko. M. Olney et al.-ek egindakoaren antzera, galderak sortzeko kontzeptu-mapatik lortutako hirukoteetatik abiatzen dira, eta sare neuronalen erabilerarekin 30 milioi galdera-erantzun dituen *corpus* bat sortzen dute.

Du et al.-ek (2017) lerro beretik doan eredia sortzen dute, baina kasu honetan ikasketa sakonean oinarritzen den eredurako sarrera esaldiak edo paragrafoak erabiltzen dituzte, hirukoteak erabili ordez. Gradu Amaierako Proiektu hau garatzeko azken hauek esaldi mailan egindako lana hartu da oinarritzat.

Esan daiteke erregeletan oinarritutako sistemak ikuspegi erredukzionista duela, alde bategatik, zer galdetu behar den aukeratzen da, eta, bestetik, galdetu beharreko hori nola galdetu behar den. Ikasketa sakonean oinarritutako sistemak, berriz, batera ikasten dute bai zer eta bai nola galdetu.

Ikasketa sakona erabiliz sortutako galderak ez dira oinarritzen eskuz egindako erregeletan baizik eta datuetan; beraz, era horretan galdera onak sortzeko kalitatezko datuak behar dira. Izan ere, Hizkuntzaren Prozesamenduan, eta, oro har, beste edozein ataza ikasketa sakonaren bidez burutzeko, kalitatezko eta tamaina handiko datuen multzoa behar da.

Galderen sorkuntzarekin erlazio handia duen beste ataza bat, galdera-erantzun sistemak dira, eta ataza batentzako datuak beste atazarentzako datu bezala erabili daitezke. Azken urteetan bi atazentzako hainbat datu-multzo sortu dira, ondokoak ezagunenetarikoak izanik: SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), RACE (Lai

et al., 2017), MS MARCO (Bajaj et al., 2016), LearningQ (Chen et al., 2018) eta NarrativeQA (Kociský et al., 2018). 2.1 taulan datu-multzo horiei buruzko zenbait datu biltzen dira, hala nola, datu-multzoko galderek eskatzen duten maila kognitiboa, erantzun mota, domeinua, eta abar.

Cognitive Level	Dataset / Contributor	Answer Type	Domain	Statistics		
				Documents	Questions	Q./Doc
Shallow	SQuAD (Rajpurkar et al., 2016)	text span	Wikipedia	20,958	97,888	4.67
	NewsQA (Trischler et al., 2017)	text span	News	12,744	119,633	9.39
Medium	MS MARCO (Nguyen et al., 2016)	human generated	Web article	1,010,916	3,563,535	3.53
	RACE (Lai et al., 2017)	multiple choice	Education	27,933	72,547	2.60
Deep	LearningQ (Chen et al., 2018)	no answer	Education	10,841	231,470	21.35
	NarrativeQA (Kociský et al., 2018)	human generated	Story	1,572	46,765	29.75

2.1 Taula: Galdera sorkuntzarako datu-multzoen laburpena (Pan et al., 2019)

3. KAPITULUA

Ikasketa sakonaren bidezko galderen sorkuntza

Ikasketa sakona azken urteotan indarra hartu duen teknika bat bada ere, 1940ko hamarkadan ditu bere hastapenak. Konputagailuek ataza adimentsuak gauzatzeko grinatik jaio zen eta bere garapenean zehar izen desberdinak jaso ditu izandako ikuspegi desberdinen eraginez. 2000ko hamarkadan ikasketa sakona terminoa lehen aldiz erabili zen sare neuronal artifizialen arloan [Aizenberg et al.](#)-en (2000) eskutik.

Ikasketa sakonarekin batera, adimen artifiziala eta ikasketa automatikoa gero eta gehiago erabiltzen diren terminoak dira, askotan bata bestearekin nahastuz, baina ez dira gauza bera. Adimen artifiziala izaki bizidunen prozesu arrazional eta deduktiboak konputagailuen bidez gauzatzea helburu duen informatikako adarra da. Hori lortzeko hurbilpenetako bat ikasketa automatikoa da. Hurbilpen horren helburua ez da ataza espezifikoko bat ebazteko algoritmoa sortzea baizik eta konputagailuak datuetatik, hau da, esperientziatik, modu automatikoan ikastea. Ikasketa sakona, berriz, ikasketa automatikoko teknika multzo bat da. Gizakien burmuinean eta horrek informazioa prozesatzeko duen eran inspiratutako eredu konputazionalan, sare neuronal artifizialetan, oinarritzen da. Ikasketa sakonak hasieran gizakion burmuina erreplikatzeko helburu bazuen ere, gaur egun printzipio orokorrago bati egiten dio erreferentzia non estatistika eta matematika aplikatuaren ezagutza erabiltzen den oinarri gisa.

Ikasketa sakonak hasieran arrakasta handia izan ez bazuen ere, konputagailu ahaltsuagoen etorrerak eta datu-multzo hobeko eta handiagoen sorrerak ikasketa sakonari etekin handiagoa ateratzea ahalbidetu dute. Izan ere, era desberdineko alor askotan, besteak beste, konputagailu bidezko ikusmenean, hizkuntzaren prozesamenduan, patroien errekonozizi-

menduan, optimizazioan, etabarretan, ikasketa sakonaren erabilerak artearen egoera era esanguratsu batean hobetu du.

Aipatutako alorretako bat Hizkuntzaren Prozesamendua (HP) da, ingelesez *Natural Language Processing (NLP)*. Hizkuntza gizakiok komunikatzeko erarik zabalduena izanik, garrantzi handiko gaia da eta ahalegin handiak egin dira kalitatezko datu-multzoak sortzeko. Horrek bultzakada handia eman dio HPari eta arrakastaz aplikatu da ikasketa sakona HPko ataza askotan, besteak beste, itzulpen automatikoan, sentimenduen analisisian eta hizketa-ezagutzan. Nola ez, galderen sorkuntza automatikoan ere sare neuronaletan oinarritutako arkitekturak erabiltzen hasi dira.

Ikasketa sakonaren bidezko galderen sorkuntza, ingelesez *Neural Question Generation (NQG)*, honela formalizatzen da:

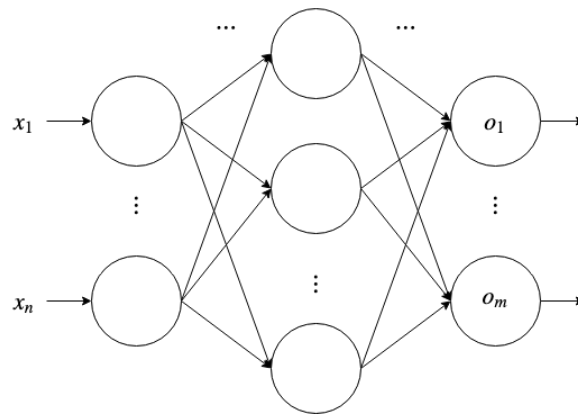
Izan bitez $x = (x_1, \dots, x_n)$ sarrerako esaldia, non x_i esaldiko token bat den. Helburua esaldi horretan dagoen informazioarekin erlazioa duen $y = (y_1, \dots, y_m)$ galdera sortzea da, horretarako, $P(y | x)$ egiantza-funtzio logaritmiko baldintzatua maximizatzen duen galdera bilatuz:

$$\bar{y} = \arg \max_y P(y | x) = \arg \max_y \prod_{t=1}^m P(y_t | x, y_{<t})$$

Beraz, ikasketa sakonaren bidez galderak sortzeko, ereduak egiantza-funtzio logaritmiko baldintzatua datuetatik ikasi behar du. Kapitulu honetan ikasketa sakonaren bidez helburu hori nola lortzen den ulertzeko oinarritzko teoria aztertzen da, horretarako hurrengo puntuak landuz: sare neuronalak, sare neuronalen ikasketa, sare neuronal errekurrenteak, kodetzaile-dekodemak arkitektura, atentzio-mekanismoa eta hitzen errepresentazioa.

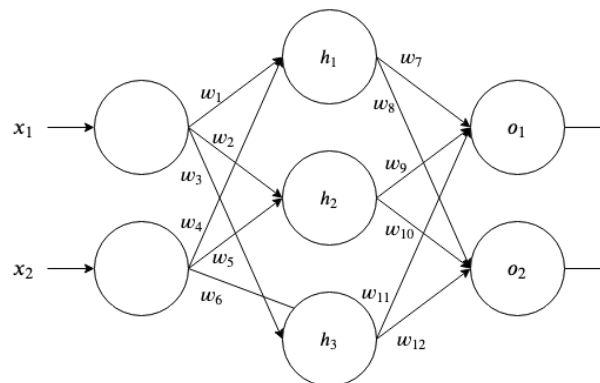
3.1 Sare neuronalak

Sare neuronal artifizialak burmuin biologikoen funtzionamenduan oinarritutako eredu konputazionalak dira. Neurona artifizialez osatutako grafo zuzendu bat bezala adierazi ohi dira, non sareak $x \in \mathbb{R}^n$ sarrera jaso eta $o \in \mathbb{R}^m$ irteera sortzen duen (3.1 irudia).



3.1 Irudia: Sare neuronal

Sare neuronalak funtzio bat balira bezala ikus daitezke. Grafoko nodo bakoitza, neurona artifizial bakoitza, funtzio bat da eta nodoen arteko loturek neurona baten irteerak hurrengo neuronan duen w_i pisua adierazten dute. Adibidez, 3.2 irudiko sarean o_1 neuronaren sarrerek h_1, h_2 eta h_3 dira eta w_7, w_9 eta w_{11} pisuekin biderkatzen dira hurrenez hurren; ondoren, horren baturari φ funtzioa aplikatzen zaio.



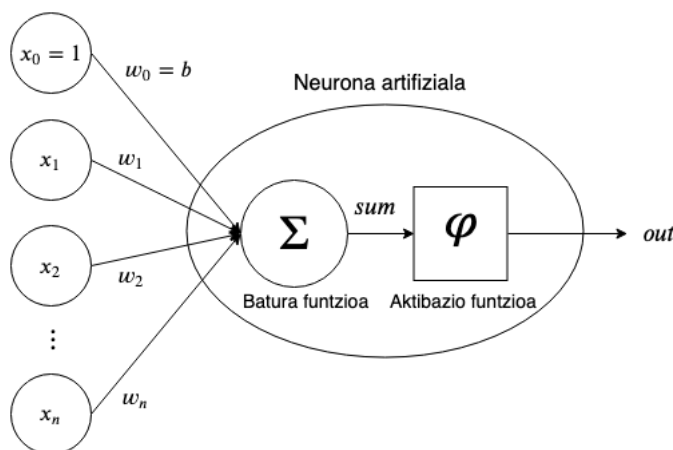
$$\begin{aligned}
 h_1 &= \varphi(w_1 \cdot x_1 + w_4 \cdot x_2) & o_1 &= \varphi(w_7 \cdot h_1 + w_9 \cdot h_2 + w_{11} \cdot h_3) \\
 h_2 &= \varphi(w_2 \cdot x_1 + w_5 \cdot x_2) & o_2 &= \varphi(w_8 \cdot h_1 + w_{10} \cdot h_2 + w_{12} \cdot h_3) \\
 h_3 &= \varphi(w_3 \cdot x_1 + w_6 \cdot x_2)
 \end{aligned}$$

3.2 Irudia: Sare neuronal batek definitutako funtzioa

3.1.1 Neurona artifiziala

Neurona artifiziala burmuin biologikoaren neuronen portaera modelatzen duen funtzio matematikoa da. Neurona artifizialak zenbait sarrera jaso, bakoitza pisu batekin biderkatu eta guztien baturari atalase bat gehitzen dio, azkenik, lortutako batura aktibazio-funtzioa izeneko funtzio ez-lineal batetik pasatzen du. Beraz, (x_1, \dots, x_n) sarrera-balioak eta φ aktibazio-funtzioa izanik, eta $x_0 = 1$ eta $w_0 = b$ definituz, baturari b balio atalasea gehitzeko, neurona artifizialak definitzen duen funtzioa honakoa da (3.3 irudia):

$$out = \varphi \left(\sum_{i=0}^n w_i x_i \right)$$



3.3 Irudia: Neurona artifiziala

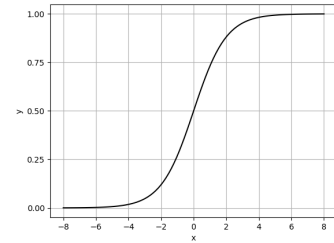
Sare neuronaletan neurona artifizialak era hierarkiko batean antolatzen dira. Bi neurona geruza berekoak direla esaten da biek sarrera berdinak jasotzen badituzte.

3.1.2 Aktibazio-funtzioak

Aktibazio-funtzioa oso elementu garrantzitsua da, sareak definitzen duen funtzioan ez-linealtasunaren propietatea sartzen duelako, eta hori delako sareari ahalmena ematen dio. Neurona artifizial batean erabiltzen den φ aktibazio-funtzioa era desberdinetara definitu daiteke. Garrantzitsua da funtzio hau deribagarria izatea, geroago ikusiko den bezala, sare neuronalaren ikasketarako funtzioaren deribatua behar delako. Hauek dira gehien erabiltzen diren aktibazio-funtzio batzuk:

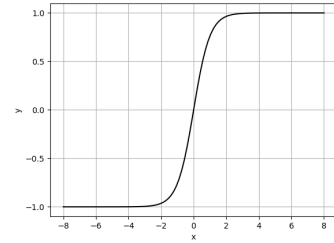
Funtzio sigmoidea

$$\sigma : \mathbb{R} \rightarrow (0, 1), \quad \sigma(x) = \frac{1}{1 + e^{-x}}$$



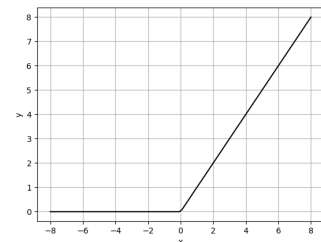
Tangente hiperbolikoa

$$\tanh : \mathbb{R} \rightarrow (-1, 1), \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



ReLU

$$ReLU : \mathbb{R} \rightarrow [0, \infty), \quad ReLU(x) = \max(0, x)$$



ReLU funtzioa ez da deribagarria $x = 0$ puntuan, baina $x < 0$ denean deribatua 0 da, eta $x > 0$ denean 1. Beraz, $x = 0$ puntuan funtzioak ez du deribaturik, baina $[0, 1]$ azpidiferentziala dauka eta tarte horretako edozein balio hartu daiteke azpideribatu bezala ikasketa fasean erabiltzeko. Oro har 1 balioa aukeratzeko da eta, beraz, honela gelditzen da funtzioaren deribatua:

$$ReLU'(x) = \begin{cases} 0 & \text{baldin } x < 0 \\ 1 & \text{baldin } x \geq 0 \end{cases}$$

Softmax

$$\text{softmax} : \mathbb{R}^n \rightarrow [0, 1]^n, \quad \text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}, \quad i = 1, \dots, n \text{ izanik}$$

Softmax funtzioa ez da aplikatzen neurona artifizial bakar baten gainean, baizik eta geruza bereko neurona guztien gainean. Izan ere, bektore bati *softmax* funtzioa aplikatuz gero, irteerako bektorea probabilitate-banaketa bezala interpreta daiteke, ondoko propietateak betetzen dituelako:

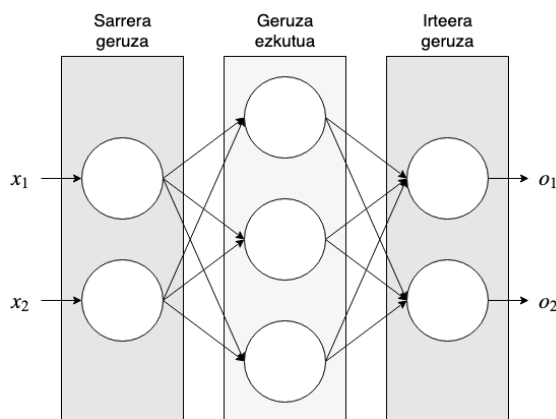
1. $\forall i \in [1, \dots, n] \text{softmax}(x_i) \geq 0$
2. $\sum_{i=1}^n \text{softmax}(x_i) = 1$

3.1.3 Arkitekturak

Neuronen arteko konexio desberdinen arabera sare neuronalen arkitektura desberdinak definitzen dira. Sareak arkitektura bat edo beste izateak aproposagoa egiten du ataza desberdinen ikasketarako. Hurrengoak arkitektura erabilienetako batzuk dira:

Multilayer perceptron

Arkitekturarik ezagunena *multilayer perceptron* (MLP) da. Arkitektura honetan, neuronen arteko loturetan ez da ziklorik sortzen eta neuronak geruza desberdinetan antolatzen dira. Lehenengo geruzari sarrera-geruza deritzo, azkeneko geruzari irteera-geruza eta erdiko geruzei geruza ezkutua. Geruza bateko neuronek aurreko geruzako neurona guztiekin daukate lotura (3.4 irudia).



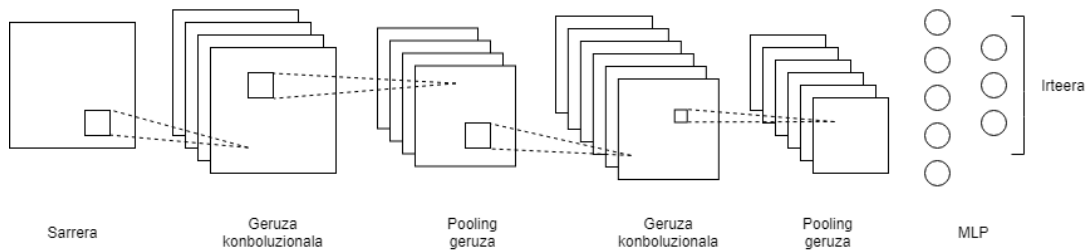
3.4 Irudia: *Multilayer perceptron*

Sare artifizial konboluzionala

Sare artifizial konboluzionalaren funtzionamendua burmuin biologikoak ikusmenerako erabiltzen dituen neuronen portaeran oinarritzen da. Konputagailu bidezko ikusmeneko atazetan oso eraginkorra den arkitektura bat da.

Sare konboluzionala iragazki konboluzionalak aplikatzen dituen hainbat geruzaraz dago osatua; geruza bakoitzaren ondoren, mapaketa bat egiten duen funtzio ez-lineal bat aplikatu ohi da (3.5 irudia).

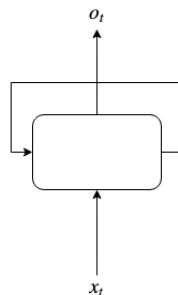
Ikusmen artifizialeko atazetan erabiltzeko sortu bazen ere, hizkuntzaren prozesamenduan ere erabiltzen den arkitektura da. Ikusmen artifizialean iragazkia irudian zehar bi ardatzetan mugituz aplikatzen da; testua prozesatzeko, berriz, ardatz bakarrean. Sare konboluzionalak testua prozesatzeko erabiltzean, iragazkiak n-gramen gainean aplikatzen dira.



3.5 Irudia: Sare artifizial konboluzionala

Sare errekurrentea

Sare errekurrentean neuronen arteko loturak atzeranzkoak izan daitezke (3.6 irudia). Era horretan sareak nolabaiteko memoria lortzen du, eta horrek sekuentziak prozesatzea ahalbidetzen du. Aurrerago era sakonago batean aztertuko dira sare hauek, testua prozesatzeko aproposak direlako.



3.6 Irudia: Sare errekurrentea

3.2 Sare neuronalen ikasketa

Sare neuronalek sarrera bat jaso eta neuronen arteko loturen w_i pisuekin biderkatuz, batura eginez eta aktibazio-funtzioa aplikatuz irteerako balio bat sortzen dute; beraz, sarearen irteera sareko pisuen balioen arabera izango da. Sareko pisu guztien multzoa, $\theta = \{w_i\}$, sarearen parametroak dira eta ebatzi nahi den atazarako emaitzarik hoberenak lortzen dituen parametro multzoa aurkitzeari sarearen entrenamendua edo ikasketa deritzo.

Sarearen entrenamendua egiteko datu-multzo bat erabiltzen da. Datu-multzoak ezezaguna den distribuzio bat jarraitzen du. Sareak distribuzio hori ikasi behar du eskuragarri dagoen laginetik abiatuta.

Parametro multzo bat ebatzi nahi den atazarako egokia den edo ez zehazteko funtzio bat definitzen da, kostu-funtzioa hain zuzen. Funtzioak saretik espero den irteera eta lortutako irteeraren arteko errorea definitzen du.

Parametro egokiak era eraginkor batean aurkitzeko optimizazio-algoritmoak erabiltzen dira, kostu-funtzioa minimizatuko dutena. Optimizazioan funtzio baten maximo edo minimo globala aurkitzea da helburua. Estimazioan, zehazki funtzio-estimazioan, sarrera-datu batzuen eta beraien irteeren artean dagoen erlazioa hurbiltzen duen funtzioa aurkitzea da helburua. Sare neuronalen ikasketan funtzio baten estimazioa egin nahi bada ere, optimizazio-algoritmoak erabiltzen dira hori lortzeko.

3.2.1 Ikasketa-paradigmak

Ikasketarako erabiltzen den datu-multzoaren izaeraren arabera hiru paradigma nagusi bereiz daitezke sare neuronal bat eta, oro har, ikasketa automatikoko eredu bat entrenatzera-koan:

- Ikasketa gainbegiratua: sarrera eta espero den irteera eskuragarri daude eta ereduak bien arteko lotura modelatzen du.
- Ikasketa ez-gainbegiratua: sarrerako datuak besterik ez daude eta ereduak datu horien intereseko propietateren bat lortzen du, irteera zein den aldeaz aurretik ezagutu gabe.
- Berrindartze bidezko ikasketa: sarrera-datuak agente batek jasotzen ditu dagoen ingurunearen elkarrekintzatik. Helburua agenteak hartu beharreko erabakiak ikastea

da, erabaki horien bidez luzerako kostu-funtzio baten balioa ahalik eta baxuena mantenduz.

Galderen sorkuntzaren kasuan, sarrera esaldi edo paragrafo bat dat eta espero den irteera esaldi edo paragrafo horri dagokion galdera, beraz, ikasketa gainbegiratuko ataza bat da.

3.2.2 Kostu-funtzioak

Kostu-funtzioa sare neuronala ebaluatzen duen funtzio bat da; hori dela eta, kostu-funtzio egokia aukeratzea premiazkoa da sareak ondo ikasteko. Oro har, propietate ezagunak dituzten kostu-funtzioak erabiltzen dira, nahiz eta berriak definitu daitezkeen. Ikasketa gainbegiratuan gehien erabiltzen diren kostu-funtzio batzuk ondokoak dira:

Batezbesteko errore karratua

Batezbesteko errore karratua erregresioa egin behar den atazetan erabili ohi den kostu-funtzioa da. n datu eta $i = 1, \dots, n$ rentzat y_i , i .garren sarrerarekin espero den irteera eta \hat{y}_i , i .garren sarrerarekin ereduak lortutako irteera izanik, espero den balioa eta lortutako balioen arteko diferentzien karratuen batezbestekoa kalkulatzeko du ondoko formularekin:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Entropia gurutzatua

Entropia gurutzatua sailkapena egin behar den atazetan erabili ohi da. Ataza horietan sarearen irteera sailkapenerako klase posible guztien gaineko probabilitate-banaketa bezala interpretatzen da. Ideia banaketa horren eta benetako irteerek jarraitzen duten banaketaren arteko diferentzia ahalik eta txikiena izatea da, horretarako entropia gurutzatua erabiliz.

Entropiak y probabilitate-banaketa batean dagoen ziurgabetasun maila neurtzen du:

$$H(y) = \sum_i y_i \log \frac{1}{y_i} = - \sum_i y_i \log y_i$$

Entropia gurutzatuak bi distribuzioren arteko antzekotasuna neurtzen du:

$$H(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i$$

Entrenamendurako adibide guztien entropia gurutzatuaren batura totala erabil daiteke kostu-funtziotzat. Era horretan, sareak ikasten duen probabilitate-banaketa eta benetako probabilitate-banaketa ahalik eta antzekoen izateko:

$$H(\{y^{(n)}\}, \{\hat{y}^{(n)}\}) = \sum_n H(y^{(n)}, \hat{y}^{(n)})$$

Beste ikuspegi batetik pentsa daiteke hobe dela ereduaren ahalmen iragarkorra maximizatzea. Horretarako egiantza-funtzioa erabil daiteke, eredu batek benetako irteera sortzeko probabilitatea neurtzen duelako. Normalean suposatzen da datuak independenteak direla, beraz, adibide guztien gaineko egiantza-funtzioa adibide bakoitzaren egiantza-funtzioaren produktua bezala deskonposatzen da:

$$L(\{y^{(n)}\}, \{\hat{y}^{(n)}\}) = \prod_n L(y^{(n)}, \hat{y}^{(n)})$$

Alde batetik, konputazionalki batura egitea biderketa egitea baino errazagoa denez, eta, bestetik, funtzio bat maximizatzea funtzio horren logaritmoa maximizatzearen baliokidea denez, aurreko ekuazioari logaritmoa aplikatzen zaio. Gainera, funtzioa kostu-funtzio bezala interpretatzeko eta helburua funtzioa minimizatzea izateko, egiantza-funtzio logaritmikoari balio negatiboa aplikatzen zaio. Beraz, egiantza-funtzio logaritmiko negatiboa kostu-funtzio bezala erabili daiteke, eta honelakoa da:

$$-\log L(\{y^{(n)}\}, \{\hat{y}^{(n)}\}) = - \sum_n \log L(y^{(n)}, \hat{y}^{(n)})$$

non $y^{(n)}$ ren egiantza-funtzioa $\hat{y}^{(n)}$ benetako irteera den klaseari esleitzen dion probabilitatea den, eta honela adieraz daitekeen:

$$\log L(y^{(n)}, \hat{y}^{(n)}) = \sum_i y_i^{(n)} \log \hat{y}_i^{(n)}$$

Ikus daiteke egiantza-funtzio negatiboa eta adibide guztien gaineko entropia gurutzatua-

ren batura funtzio berdina direla:

$$-\log L(\{y^{(n)}\}, \{\hat{y}^{(n)}\}) = \sum_n [-\sum_i y_i^{(n)} \log \hat{y}_i^{(n)}] = \sum_n H(y^{(n)}, \hat{y}^{(n)})$$

Eta, beraz, entropia gurutzatua minimizatzea egiantza-funtzioa maximizatzearen baliokidea da.

3.2.3 Optimizazio-algoritmoak

Optimizazio-algoritmo baten helburua funtzio baten minimo edo maximo globala topatzea da; sare neuronalen entrenamenduaren kasuan kostu-funtzioa minimizatzea izango da helburua. Beraz, minimizatzea helburu duen optimizazio-algoritmoa honela definitzen da:

Izan bedi f kostu-funtzioa, Θ ereduaren parametro posible guztien multzoa eta x sarreradatuak. Helburua ereduaren parametro hoberenak, $\theta^* \in \Theta$ aurkitzea da non

$$\forall \theta \in \Theta, f(x, \theta^*) \leq f(x, \theta)$$

Optimizazio-algoritmo desberdinak sortu dira sare neuronalak entrenatzeko, hemen erabilienetako batzuk aztertzen dira:

Gradientearen jaitsiera

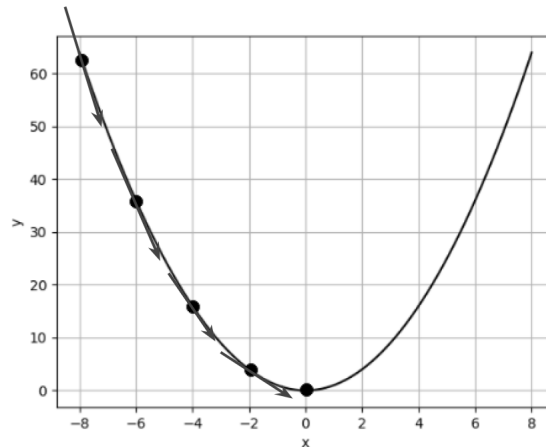
Gradientearen jaitsiera optimizazio-problema ebazteko gehien erabiltzen den algoritmoa da. Funtzioaren optimizazioa gradientearen kalkuluaren bidez gauzatzen da.

Aldagai anitzeko funtzio baten gradiente bere deribatu partzialek osatutako matrize bat da, matrize jacobitar izenekoa. Gradienteak puntu jakin batean funtzioaren hazkunde handieneko norabidea adierazten du.

Gradiente-jaitsieraren algoritmoan hazkunde handieneko norabide horren kontrako norantzan mugitzen dira ereduaren parametroak funtzioa minimizatzeke. Gradientearen jaitsieran, ereduaren θ parametroak, $\nabla f(x, \theta)$ kostu-funtzioaren gradiente eta η ikasketataza erabiliz, ondoko eran eguneratzen dira:

$$\theta = \theta - \eta \nabla f(x, \theta)$$

3.7 irudiak aldagai bakarreko funtzio batean gradientearen jaitsiera algoritmoak duen eboluzioa irudikatzen du.



3.7 Irudia: Gradientearen jaitsiera aldagai bakarreko funtzioan

Ikasketa-tasaren balioa garrantzi handikoa da. Balio handiegia aukeratzen bada, gerta daiteke kostu-funtzioa handitzea, txikitu ordez. Balio txikiegia aukeratzen bada, berriz, sareak oso mantso ikasiko du edo optimo lokal batean geratuko da trabatuta.

Gradientearen jaitsierak zenbait muga ditu; izan ere, algoritmoak ez du bermatzen minimo globalera iritsiko denik eta gerta daiteke minimo lokal batean gelditzea.

Gradiente jaitsieraren aldaerak

Gradientearen jaitsierak dituen arazoetako batzuei aurre egiteko, zenbait aldaera sortu dira: Gradientearen jaitsiera estokastikoa, mini-sorta bidezko gradientearen jaitsiera, etab.

Gradientearen jaitsieran gradientearen entrenamendurako datu guztiak kontuan hartuz kalkulatu da, eta, ondoren, ereduaren parametroak eguneratzen dira. Gradientearen jaitsiera estokastikoan, ingelesez *Stochastic Gradient Descent (SGD)*, adibide bakoitzeko gradientearen kalkulatu eta parametroak eguneratzen dira.

Mini-sorta bidezko gradientearen jaitsieran gradientearen N adibiderek, sorta edo ingelesez *batch* deitua, kalkulatu ondoren eguneratzen dira parametroak; era horretan algoritmoak hobeto konbergitzen du. Normalean, sare neuronalen arloan mini-sorta bidezko gradientearen jaitsierari gradientearen jaitsiera estokastikoa deitu ohi zaio.

Gradientearen jaitsieraren hobekuntzak

Esan bezala, gradientearen jaitsierak zenbait arazo izaten ditu minimo lokalekin edota datu-multzo sakabanatuekin. Hori dela eta, arazo horiei aurre egiten dioten zenbait algoritmo diseinatu dira, besteak beste, Momentum, Adagrad, Adadelta eta Adam. Hala ere, algoritmo horiek ere gradientearen kalkuluan oinarritzen dira, baina ez dira hauen detaileak aztertuko, aurrerago ikusiko denez gure kasuan mini-sorta bidezko gradientearen jaitsiera erabiltzea erabaki delako.

Backpropagation

Aipatutako optimizazio-algoritmo guztietan gradientearen kalkulua egin behar da era iteratibo batean. Hori dela eta, gradienteen kalkulua konputazionalki efizientea den algoritmo bat erabiliz egiten da, *backpropagation* algoritmoa hain zuzen ere.

Backpropagation [Rumelhart et al.-ek \(1986\)](#) proposatutako algoritmoa da, katearen erregela eta programazio dinamikoan oinarritzen dena gradientearen kalkulua era eraginkor batean egiteko.

Programazio dinamikoa behin baino gehiagotan egin behar diren kalkuluen errepikapena eragozten duen programazio-teknika bat bezala ikus daiteke. Katearen erregela bi funtzioen konposizioaren deribatua kalkulatzeko formula da:

Izan bedi f eta g funtzioak. f deribagarria bada x puntuan eta g deribagarria bada $f(x)$ puntuan, orduan $g \circ f$ deribagarria da x puntuan, eta hau da bere deribatua:

$$(g \circ f)'(x) = (g(f(x)))' = g'(f(x)) \cdot f'(x)$$

Leibniz-en notazioa erabiliz eta $z = g(y)$ eta $y = f(x)$ definituz:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

Backpropagation algoritmoaren funtzionamendua ulertzeko erarik errazena adibide bat aztertzea da. Demagun [3.8](#) irudiko sarea erabiltzen dela, erabiltzen den aktibazio-funtzioa funtzio sigmoidea dela:

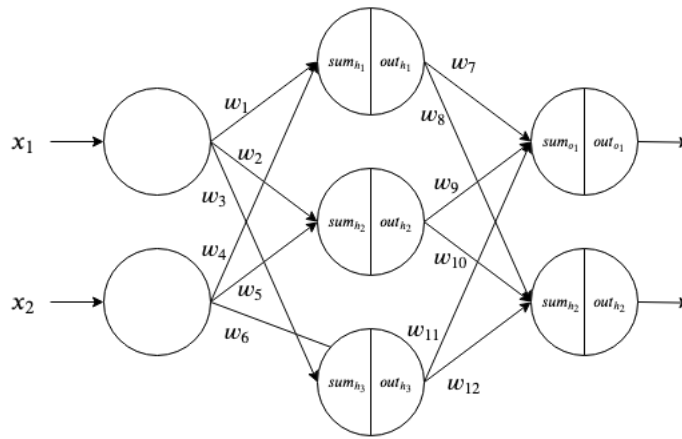
$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

beraz, bere deribatua ondokoa da:

$$\varphi'(x) = \varphi(x) \cdot (1 - \varphi(x))$$

eta kostu-funtzioa batezbesteko errore karratua dela. Kalkuluak errazteko, demagun entrenatzeko datu bakar bat dugula $(x, y) = ((x_1, x_2), (y_1, y_2))$:

$$\mathcal{L} = \frac{1}{2}(y_1 - out_{o_1})^2 + \frac{1}{2}(y_2 - out_{o_2})^2$$



$$\begin{aligned} sum_{h_1} &= w_1 \cdot x_1 + w_4 \cdot x_2, & out_{h_1} &= \varphi(sum_{h_1}) \\ sum_{h_2} &= w_2 \cdot x_1 + w_5 \cdot x_2, & out_{h_2} &= \varphi(sum_{h_2}) \\ sum_{h_3} &= w_3 \cdot x_1 + w_6 \cdot x_2, & out_{h_3} &= \varphi(sum_{h_3}) \end{aligned}$$

$$\begin{aligned} sum_{o_1} &= w_7 \cdot out_{h_1} + w_9 \cdot out_{h_2} + w_{11} \cdot out_{h_3}, & out_{o_1} &= \varphi(sum_{o_1}) \\ sum_{o_2} &= w_8 \cdot out_{h_1} + w_{10} \cdot out_{h_2} + w_{12} \cdot out_{h_3}, & out_{o_2} &= \varphi(sum_{o_2}) \end{aligned}$$

3.8 Irudia: Backpropagation adibiderako sare neuronalak eta ekuazioak

Beraz, optimizatu beharreko funtzioa hau litzateke:

$$\mathcal{L} = \frac{1}{2}(y_1 - (\varphi(w_7 \cdot \varphi(w_1 \cdot x_1 + w_4 \cdot x_2) + w_9 \cdot \varphi(w_2 \cdot x_1 + w_5 \cdot x_2) + w_{11} \cdot \varphi(w_3 \cdot x_1 + w_6 \cdot x_2))))^2 +$$

$$\frac{1}{2}(y_2 - (\varphi(w_8 \cdot \varphi(w_1 \cdot x_1 + w_4 \cdot x_2) + w_{10} \cdot \varphi(w_2 \cdot x_1 + w_5 \cdot x_2) + w_{12} \cdot \varphi(w_3 \cdot x_1 + w_6 \cdot x_2))))^2$$

Non $x = (x_1, x_2)$ eta $y = (y_1, y_2)$ balio konstanteak diren eta optimizatu beharreko parametroak $\theta = (w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}, w_{11}, w_{12})$ diren.

w_7 eta w_1 parametroen gradienteak honela kalkulatzen dira:

$$\frac{\partial \mathcal{L}}{\partial w_7} = \frac{\partial \mathcal{L}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial sum_{o_1}} \cdot \frac{\partial sum_{o_1}}{\partial w_7}$$

$$\frac{\partial \mathcal{L}}{\partial w_7} = (out_{o_1} - y_1) \cdot (\varphi(sum_{o_1}) \cdot (1 - \varphi(sum_{o_1}))) \cdot out_{h_1}$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \left(\frac{\partial \mathcal{L}_{o_1}}{\partial out_{o_1}} \cdot \frac{\partial out_{o_1}}{\partial sum_{o_1}} \cdot \frac{\partial sum_{o_1}}{\partial out_{h_1}} + \frac{\partial \mathcal{L}_{o_2}}{\partial out_{o_2}} \cdot \frac{\partial out_{o_2}}{\partial sum_{o_2}} \cdot \frac{\partial sum_{o_2}}{\partial out_{h_1}} \right) \cdot \frac{\partial out_{h_1}}{\partial sum_{h_1}} \cdot \frac{\partial sum_{h_1}}{\partial w_1}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_1} = & ((out_{o_1} - y_1) \cdot (\varphi(sum_{o_1}) \cdot (1 - \varphi(sum_{o_1})))) \cdot w_7 + \\ & + (out_{o_2} - y_2) \cdot (\varphi(sum_{o_2}) \cdot (1 - \varphi(sum_{o_2}))) \cdot w_8 \cdot \varphi'(sum_{h_1}) \cdot x_1 \end{aligned}$$

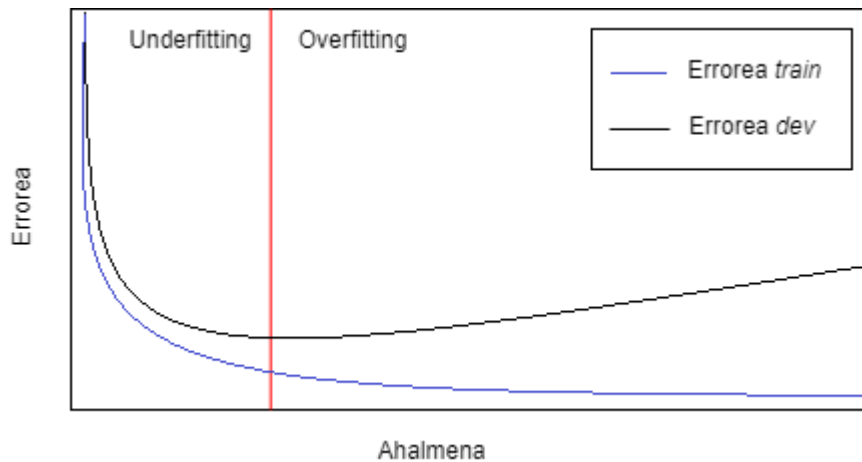
Bai w_1 bai w_7 parametroaren gradienteak kalkulatzeko $\frac{\partial out_{o_1}}{\partial sum_{o_1}} = \varphi(sum_{o_1}) \cdot (1 - \varphi(sum_{o_1}))$ kalkulatu behar da. Programazio dinamikoari esker kalkulu hori behin baino ez da egingo.

3.2.4 Ereduaren orokortzea

Sare neuronal baten helburua funtzio bat estimatzea da, eta, horretarako, optimizazio-algoritmoak erabiltzen dira. Beraz, sare neuronal bat entrenatzean optimizazio-algoritmoak erabili arren, helburua ez da entrenatzeko erabilitako datuekin ahalik eta kostu-funtzioaren baliorik baxuena lortzea, baizik eta entrenamenduan erabili ez diren eta banaketa berdina jarraitzen duten beste datuekin (balidazio datuak) ere errore baxua lortzea. Bai entrenatzeko erabilitako datuekin, bai banaketako beste datuekin errorea baxua denean, sareak orokortzea lortzen duela esaten da.

Entrenatzeko erabilitako datuekiko kostu-funtzioaren balioaren eta balidazio datuekiko balioaren arteko diferentzia handiegia bada, entrenamenduan *overfitting* gertatu dela esaten da. Entrenatzeko erabilitako datuekin lortutako kostu-funtzioaren balioa ez bada nahiko baxua, entrenamenduan *underfitting* gertatu dela esaten da (3.9 irudia).

Overfitting eta *underfitting* ez gertatzeko, sareak entrenamenduan erabili ez dituen datuekin zenbateko errorea daukan jakin behar da. Datu-multzo bakarra dagoenez, normalean, eskura dauden datuak hiru zatitan banatzen dira: *train*, *dev* eta *test*.



3.9 Irudia: *Overfitting* eta *underfitting* sarearen ikasketan

Datu-multzotik sortutako *train* zatia ereduaren parametroak eguneratzeko erabiltzen da optimizazio-algoritmoarekin batera; zati horretako adibide guztiak behin erabili direnean epoka bat pasa dela esaten da. Eredua hainbat epoka eginez entrenatzen da, eta epoka bakoitzaren ostean pisu eguneratuak dituen lortutako ereduaren ebaluazioa egiten da *dev* zatia erabiliz. Epoka guztiak amaitzean ebaluazio hoberena jaso duen epokako eredu aukeratu da. Ereduaren aukeraketan *dev* zatia zeharkako eragina dauka, hori dela eta, aukeraturako ereduaren benetako ebaluazioa lortzeko *test* zatia erabiltzen da. Ondoko algoritmoak azaldutako prozesua gauzatzen du:

Algoritmoa 1: Sare neuronalen entrenamendua

input : *n*: iterazio kopurua

data: datuak

proportzioa: datuak hiru zatitan banatzeko proportzioa

eredua: ereduak

train, dev, valid = datuak_banatu(*data*, *proportzioak*) ;

eredu_lista = [] ;

ereduen_balidazioak = [] ;

for *n* **do**

eredu_lista += pisuak_eguneratu(*eredua*, *train*);

ereduen_balidazioak += ebaluatu(*eredua*, *dev*) ;

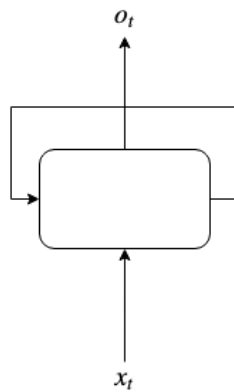
end

eredua = eredurik_hoberena(*eredu_lista*, *ereduen_balidazioak*);

ebaluatu(*eredua*, *valid*);

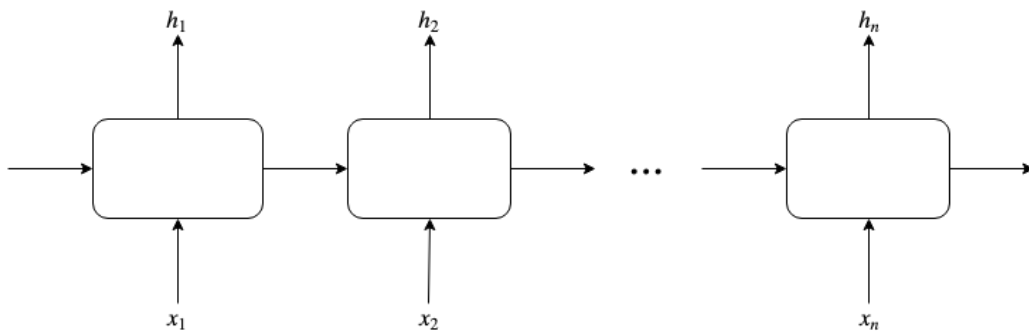
3.3 Sare neuronal errekurteak

Lehen aipatu den bezala, sare errekurteetan, ingelesez *Recurrent Neural Network (RNN)*, neuronen arteko loturak atzeranzkoak izan daitezke (3.10 irudia), era horretan sareak nolabaiteko memoria lortzen du eta horrek ahalbidetzen du sekuentziak prozesatzea.



3.10 Irudia: Sare errekurteak

Sare errekurte bat adierazteko modu erabili bat, sarea bera destolestuta ikustea da (3.11 irudia). Era horretan, sare errekurteak denboran zehar errepikatzen den sare berdinen kopien segida bezala ikus daiteke, non une oro irteera bat sortu eta sarearen egoera ezkutu eguneratua pasatzen zaion hurrengo sareari.



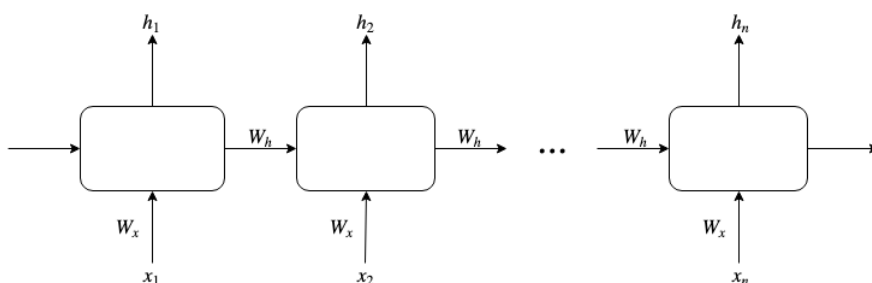
3.11 Irudia: Sare errekurte destolestua

3.3.1 Sare errekurrente sinplea

Sare errekurrenterik sinpleenean, $h_t \in \mathbb{R}^d$ uneko egoera ezkutua, $x_t \in \mathbb{R}^n$ uneko sarrera eta h_t aurreko egoera ezkutuaren transformazio linealen ($W_x \in \mathbb{R}^{d \times n}$, $W_h \in \mathbb{R}^{d \times d}$) baturari, g funtzio ez-lineal bat aplikatuz eguneratzen da eta y_t irteera egoera ezkutua bera da:

$$h_t = g(W_x \cdot x_t + W_h \cdot h_{t-1})$$

$$y_t = h_t$$



3.12 Irudia: Sare errekurrente sinplea

Backpropagation through time

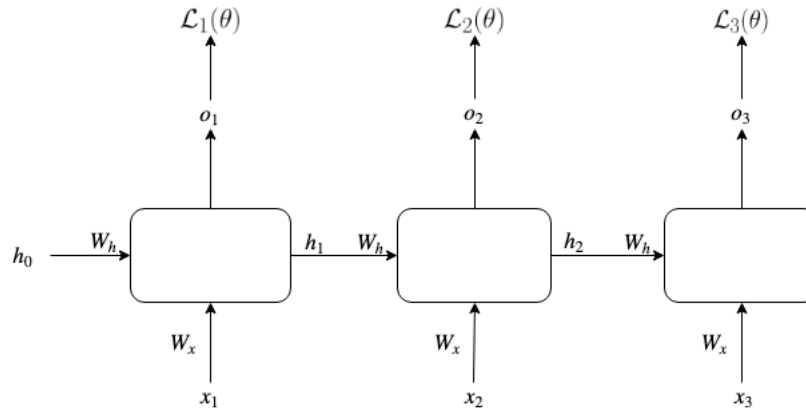
Sare errekurrenteen ikasketan *Backpropagation through time (BTT)* algoritmoa erabiltzen da eta *backpropagation* algoritmoaren ideia bera sare errekurrentetan aplikatzean datza algoritmoaren ideia.

Sare errekurrente batean une bakoitzean irteera bat dago, beraz, kostu-funtzioa irteera guztien gaineko errorearen batura bezala definitzen da:

$$\mathcal{L}(\theta) = \sum_{t=1}^T \mathcal{L}_t(\theta)$$

Sare errekurrentetan gradienteak kalkulatzeko dagoen arazoa $\mathcal{L}_t(\theta)$ uneko errorean parametro batek daukan eragina kalkulatzeko datza. Parametroa egoera ezkutuaren eguneraketan eragina badu, aurreko denbora-une guztietan parametroak daukan eragina ere kontuan hartu behar da.

Demagun 3.13 irudiko sarea erabiltzen dela, non hiru denbora-une burutzen diren eta $\frac{\partial \mathcal{L}_3(\theta)}{\partial W_h}$ kalkulatu nahi dela.



3.13 Irdia: *Backpropagation through time*

Katearen erregela aplikatuz:

$$\frac{\partial \mathcal{L}_3(\theta)}{\partial W_h} = \frac{\partial \mathcal{L}_3(\theta)}{\partial h_3} \cdot \frac{\partial h_3}{\partial W_h}$$

Baina $\frac{\partial h_3}{\partial W_h}$ ezin da zuzenean kalkulatu:

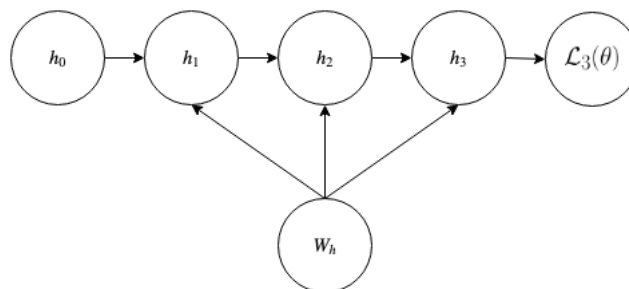
$$h_3 = g(W_x \cdot x_3 + W_h \cdot h_2)$$

$$h_2 = g(W_x \cdot x_2 + W_h \cdot h_1)$$

$$h_1 = g(W_x \cdot x_1 + W_h \cdot h_0)$$

$$h_3 = g(W_x \cdot x_3 + W_h \cdot (g(W_x \cdot x_2 + W_h \cdot (g(W_x \cdot x_1 + W_h \cdot h_0))))))$$

W_h -k egoera ezku bakoitzarengan du eragina, eta, era berean, egoera ezku baten gainean aurreko egoera guztiek daukate eragina, 3.14 irudiko grafo ordenatuan adierazten den bezala.



3.14 Irdia: W_h -ren eraginaren grafo ordenatua

$\frac{\partial h_3}{\partial W_h}$ kalkulatzeko W_h -k $\mathcal{L}_3(\theta)$ -n duen eragina denboran zehar atzerantz barreiatzen da, non $\frac{\partial^+ h_k}{\partial W_h}$, h_{k-1} konstante bezala hartuz kalkulatu den:

$$\frac{\partial h_3}{\partial W_h} = \frac{\partial^+ h_3}{\partial W_h} + \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial W_h}$$

$$\frac{\partial h_3}{\partial W_h} = \frac{\partial^+ h_3}{\partial W_h} + \frac{\partial h_3}{\partial h_2} \cdot \left[\frac{\partial^+ h_2}{\partial W_h} + \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_h} \right]$$

$$\frac{\partial h_3}{\partial W_h} = \frac{\partial^+ h_3}{\partial W_h} + \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial^+ h_2}{\partial W_h} + \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial^+ h_1}{\partial W_h} = \sum_{k=1}^3 \frac{\partial h_3}{\partial h_k} \frac{\partial^+ h_k}{\partial W_h}$$

Aurreko kalkulua erabiliz edozein uneko $\mathcal{L}_t(\theta)$ -ren egoera ezkutuan eragina duen parametro batekiko duen gradienteak kalkulatzeko formula orokorra lor daiteke:

$$\frac{\partial \mathcal{L}_t(\theta)}{\partial W_h} = \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \frac{\partial h_t}{\partial W_h} = \frac{\partial \mathcal{L}_t(\theta)}{\partial h_t} \sum_{k=1}^t \frac{\partial h_t}{\partial h_k} \frac{\partial^+ h_k}{\partial W_h}$$

3.3.2 LSTM

Long short-term memory (LSTM) motako sare neuronal errekurrenteak [Hochreiter and Schmidhuber](#)-ek (1997) proposatu zituzten lehen aldiz, ordura arteko sare errekurrenteek izandako *vanishing* eta *exploding gradient* arazoa konpontzeko.

Vanishing eta *exploding gradient* arazoa sarearen ikasketa-prozesuan gertatzen da gradiente oso txikia edo oso handia denean, hurrenez hurren. Arazo horrek sareko pisuak ia ez eguneratzea edo gehiegi aldatzea eragiten du eta, ondorioz, sarearen ikasketa egokia ez da gertatzen.

LSTMak *vanishing* eta *exploding gradient* arazoari aurre egiteko duen ezaugarri garrantzitsuenetako bat *gated architecture*koa da; hau da, zenbait ate erabiltzen ditu sareko informazioarekin zer egingo den erabakitzeko. Ateak 0 eta 1 arteko balioak dituzten bektoreak dira $[0, 1]^n$. Bektore horiek lortzeko funtzio sigmoidea erabiltzen da.

Arkitektura

LSTMaren arkitektura 3.15 irudian irudikatzen da, non ondorengo elementuak agertzen diren:

x_t : t uneko sarrera, $x = (x_1, \dots, x_n)$ izanik sarrera osoa.

c_t : t uneko zelula-egoera

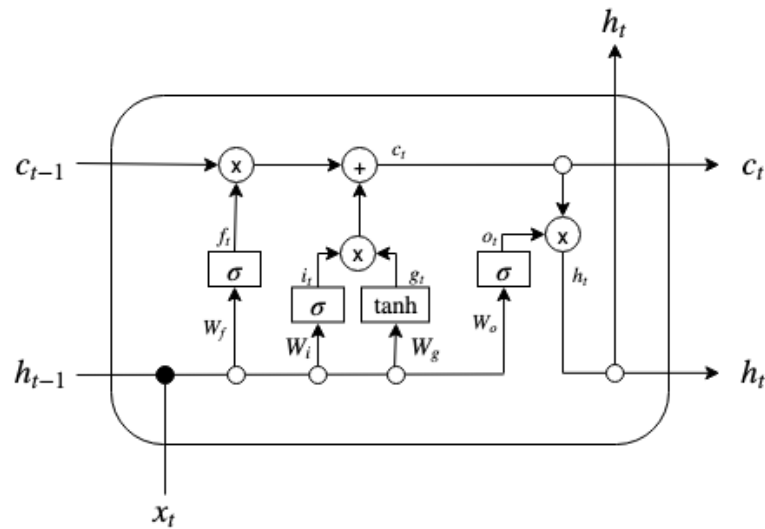
h_t : t uneko egoera ezkutua

f_t : t uneko ahazteko atea

i_t : t uneko sarrera-atea

o_t : t uneko irteera-atea

$W_f, W_i, W_o, W_g, b_f, b_i, b_o, b_g$: Sarearen parametroak. b_f, b_i, b_o, b_g ez dira 3.15 irudian agertzen irudia sinplifikatzearen, baina ondorengo formuletan ikusiko den bezala balio atalaseak sartzen dituzte kalkuluetan.



3.15 Irudia: LSTM

Lehenengo urratsa c_t zelula-egoeran zer informazio mantenduko den erabakitzea da. Horretarako, h_{t-1} aurreko egoera ezkutua eta x_t uneko sarrera kontuan hartuz f_t ahazteko atea kalkulatu da, $[x_t, h_{t-1}]$, x_t eta h_{t-1} bektoreen kateamendua izanik:

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

Hurrengo pausoa c_t zelula-egoera zer baliorekin eguneratuko den erabakitzea da. Horretarako, alde batetik, g_t balio hautagai batzuk kalkulatzen dira:

$$g_t = \tanh(W_g \cdot [x_t, h_{t-1}] + b_g)$$

eta i_t sarrera-atearekin batera:

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i)$$

c_t zelula-egoera eguneratzeko g_t balio hautagai horietatik zeintzuk erabiliko diren erabakitzen da, * elementu bidezko biderketa izanik:

$$c_t = f_t * c_{t-1} + i_t * g_t$$

Beraz, c_t zelula-egoera eguneratzen da c_{t-1} aurreko zelula-egoera, f_t horren atea, g_t balio hautagaiak eta i_t uneko sarrera-atea erabiliz.

Azkenik, h_t irteerako egoera ezkutua kalkulatu behar da. Horretarako, c_t zelula-egoera berrituari tangente hiperbolikoa aplikatu eta o_t irteera-atearekin biderkatuz; irteera-ateak zelula-egoeratik zer balio itzuliko diren erabakitzen du:

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

3.3.3 GRU

GRU (Gated Recurrent Unit), *LSTM*aren aldaera sinplifikatu bat da [Cho et al.-ek \(2014\)](#) aurkeztua. Ahazteko atea eta sarrera-atea konbinatzen ditu eguneratze-atean eta zelula-egoera eta egoera ezkutua ere konbinatzen ditu:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

3.4 Kodetzaile-dekodemetzaile arkitektura

Kodemetzaile-dekodemetzaile arkitektura, ingelesez *encoder-decoder* edo *seq2seq*, sekuentziak sekuentziekin mapatzen dituen arkitektura bat da [Sutskever et al.-ek \(2014\)](#) garatua.

Sarrera eta irteerako sekuentzien luzera aldez aurretik ezaguna denean sare errekurente bat gai da bi sekuentzien arteko mapatzea egiteko, baina bi sekuentziak luzera desberdinekoak badira, eta haien luzeren arteko erlazioa konplikatu denean, mapatzea egitea ez da hain tribiala.

Arazo horri aurre egiteko estrategiarik sinpleena sarrerako sekuentzia luzera finkoko bektore batera mapatzea da RNN bat erabiliz, eta, ondoren, bektorea irteerako sekuentziara mapatu beste RNN bat erabiliz. Hori da, hain zuzen, kodetzaile-dekodemetzaile arkitektura egiten duena. Arkitektura bi ataletan bana daiteke: kodetzailea, sarrerako sekuentzia luzera finkoko bektore batera mapatzen duen atala eta dekodezailea, bektorea helburu sekuentziarekin mapatzen duen atala.

3.4.1 Kodetzailea

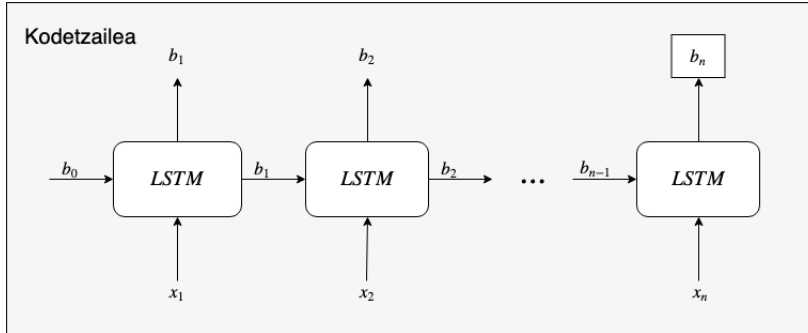
Kodemetzailea sare errekurente bat da, gehienetan LSTM bat, sekuentzien luzerako zein laburrerako dependentziak harrapatzen dituelako, eta sarrerako sekuentzia dimensio finituko bektore batera mapatzen du. Sarrerako $x = (x_1, \dots, x_n)$ sekuentziaren errepresentazioa sare errekurentearen azkenengo b_n egoera ezkutua izango da (3.16 irudia). Hasierako egoera ezkutua, b_0 , ausaz edo probabilitate-banaketa bat jarraituz hasieratu ohi da. Beraz, egoera ezkutua honela kalkulatu dira:

$$b_i = RNN(x_i, b_{i-1})$$

3.4.2 Dekodemetzailea

Dekodemetzailea beste sare errekurente bat da. Kodetzaileak lortutako b_n sarrerako sekuentzia-errepresentazioa erabiltzen da dekodezailearen egoera ezkutua hasieratzeko, eta, horren arabera eta sortzen doan irteeren arabera, irteerako sekuentzia sortzen du.

Dekodemetzailea hizkuntza-eredu bezala erabiltzen da. Hizkuntza-ereduak sekuentzien gaineko probabilitate-banaketak dira eta kodetzaile-dekodemetzaile arkitekturaren $p(y_1, \dots, y_m |$



3.16 Irudia: Kodetzailea

x_1, \dots, x_n) probabilitate baldintzatua estimatu ahal izateko erabiltzen da. Hori lortzeko horrela deskonposatzen da probabilitate baldintzatua:

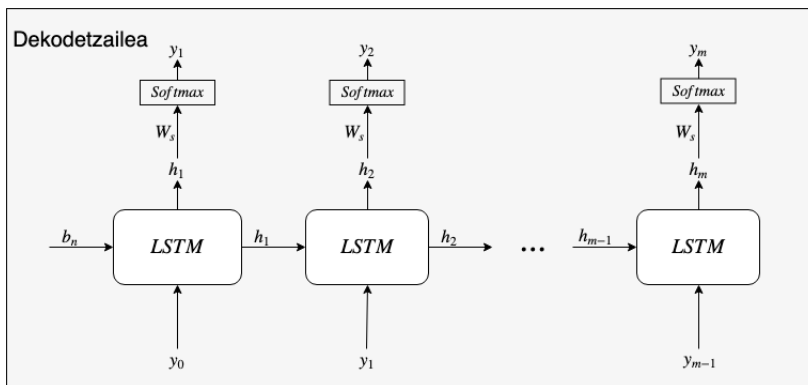
$$p(y_1, \dots, y_m \mid x_1, \dots, x_n) = \prod_{i=1}^m p(y_i \mid b_n, y_1, \dots, y_{i-1})$$

$p(y_i \mid b_n, y_1, \dots, y_{i-1})$ distribuzio bakoitza \mathcal{V} helburu hiztegiko hitz guztien gainean *softmax* funtzioa aplikatuz adierazten da (3.17 irudia). Ondokoak dekodetzailearen ekuazioak dira:

$$h_0 = b_n$$

$$h_i = RNN(y_{i-1}, h_{i-1})$$

$$p(y_i \mid b_n, y_1, \dots, y_{i-1}) = \text{softmax}(W_s \cdot h_i)$$



3.17 Irudia: Dekodetzailea

3.5 Atentzio-mekanismoa

Kodetzaile-dekodetzaile arkitekturak zenbait muga ditu: sekuentzia oso luzeetan zaila da kodetzaileak sarrerako sekuentziako beharrezko informazio guztia bektore batean islatzea. Hori dela eta, kodetzaile-dekodetzaile arkitekturari atentzioa izeneko mekanismo bat gehitzen zaio, zeinak sare neuronalak behartzen duen, irteerako sekuentziako y_t elementu bakoitza sortzean, (x_1, \dots, x_n) sarrerako sekuentziako atal garrantzitsuak identifikatzera.

Atentzio-mekanismoa Bahdanau et al.-ek (2014) proposatu zuten lehen aldiz eta Luong et al.-ek (2015) ideia berdina aplikatu zuten arkitektura sinpleago batean. Azken hauek atentzioan oinarritutako bi mekanismo proposatu zituzten, Global eta Local hain zuzen. Atal honetan bi mekanismo horiek aztertuko dira, batez ere Global izenekoa, azken hau baita geroago erabiltzea erabaki dena.

Global eta Local mekanismoen arteko desberdintasuna atentzioa sarrerako elementu guztietan edo gutxi batzuetan aplikatzean datza. Bi mekanismoen funtzionamendua kodetzailearen h_t uneko egoera ezkutua hartu eta sarrerako sekuentziako informazio garrantzitsua harrapatzen duen c_t testuinguru-bektore bat sortzean datza. Horretarako, mekanismo bakoitzean era desberdinean kalkulatu diren a_t alineamendu-bektore batzuk erabiliz. Testuinguru-bektore horrekin atentziodun egoera ezkutu bat sortzen da, eta horrekin dekodetzaileko irteera:

$$c_t = \sum_{i=1, \dots, m} a_{i,t} b_i$$

$$\bar{h}_t = \tanh(W_c \cdot [c_t; h_t])$$

$$p(y_i | b_n, y_1, \dots, y_{i-1}) = \text{softmax}(W_s \cdot \bar{h}_t)$$

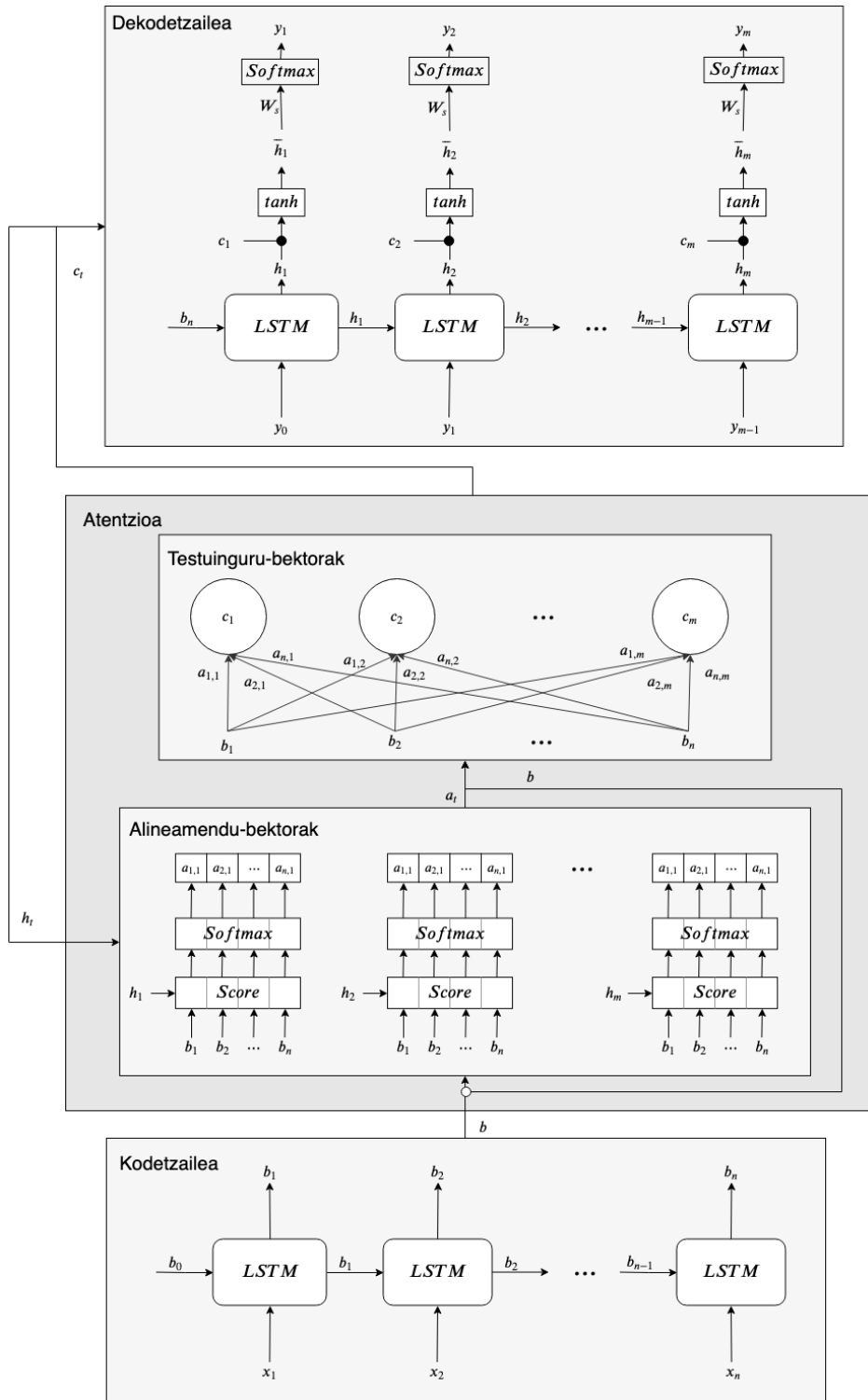
Global mekanismoak kodetzailearen aldeko egoera ezkutu guztiak hartzen ditu kontuan testuinguru-bektorea kalkulatzeko. Honela kalkulatu dira alineamendu-bektoreak:

$$a_t(s) = \text{align}(h_t, \bar{h}_s) = \text{softmax}(\text{score}(h_t, \bar{h}_s))$$

non *score* hiru funtzio hauetako bat izan daitekeen:

$$\text{score}(h_t, \bar{h}_s) = \begin{cases} h_t^\top \cdot \bar{h}_s & \text{dot} \\ h_t^\top \cdot W_a \cdot \bar{h}_s & \text{general} \\ v_a^\top \cdot \tanh(W_a \cdot [h_t; \bar{h}_s]) & \text{concat} \end{cases}$$

3.18 irudian atentzio-mekanismoa erabiltzen duen kodetzaile-dekodematzailearen arkitektura irudikatzen da.



3.18 Irudia: Kodetzaile-dekodematzaile arkitektura atentzio mekanismoarekin

3.6 Hitzen errepresentazioa

Testua sare neuronalekin prozesatu ahal izateko, bektore edo zenbaki eran adierazi behar da. Oro har, testua tokenetan banatzen da, eta, beraz, token horien errepresentazio bektorial edo numerikoa behar da. Atal honetan tokenak era numerikoan adierazteko era desberdinak aztertuko dira.

3.6.1 Indexatutako hiztegia

Indexatutako hiztegi errepresentazioa erabiliko den *corpuseko* tokenen hiztegi bat sortu eta token bakoitza hiztegian token horri dagokion indizearekin ordezkatzean datza. Oro har errepresentazio mota hau beste errepresentazio batzuekin batera erabiltzen da.

Hiztegia sortzeko askotan hiztegiaren tamaina mugatzen da. Tokenek *corpusean* daukaten agerpen kopuruaren arabera ordenatu eta hiztegian agerpen altuenekoak bakarrik sartzen dira. Behin hiztegia izanda token bakoitzari indize bat egokitzen zaio eta token bakoitza bere indizearengatik ordezkatzeko da.

Hiztegiari token berezi batzuk gehitu ohi zaizkio, besteak beste, *Out of Vocabulary (OoV)* hiztegian agertzen ez diren tokenak ordezkatu ahal izateko, *begining of sentence (BOS)* esaldi baten hasiera adierazteko, *end of sentence (EOS)* esaldi baten amaiera adierazteko, etab. 3.1 taulan *cat* eta *dog* tokenak eta *OoV*, *BOS* eta *EOS* token bereziak dituen hiztegia ikusten da.

id	word
1	OoV
2	BOS
3	EOS
4	cat
5	dog

3.1 Taula: Indexatutako hiztegi baten adibidea

3.6.2 One-hot kodeketa

One-hot kodeketa errepresentazioa indexatutako hiztegiaren errepresentazioarekin batera erabiltzen da. Token bakoitza hiztegiaren tamainako bektore bat bezala errepresentatzen da, non bektorea zeroz dagoen osatua elementu batean izan ezik; tokenari dagokion indizearen posizioan bat jartzen da. Adibidez, 3.1 taulako hiztegia hartuz *cat* tokenari 3.19 irudiko bektorea dagokio.

0	0	0	1	0
---	---	---	---	---

3.19 Irudia: *One-hot encoding*

Errepresentazio honen desabantairik handiena memoriaren erabilera handiegia egiten dela da eta errepresentazioak ez duela tokenen informazio semantikorik ematen.

3.6.3 Esanahi-bektoreak

Esanahi-bektoreetan tokenak dimentsio finituko zenbaki errealeko bektore bidez errepresentatzen dira. Ikasketarako datu nahiko badaude, tokenak errepresentatzen dituzten bektoreak ausaz hasiera daitezke eta entrenamenduan ereduak token bakoitzaren esanahia ikasiko du. Hala ere, orokorrean aurrez ikasitako esanahi-bektoreak erabiltzen dira.

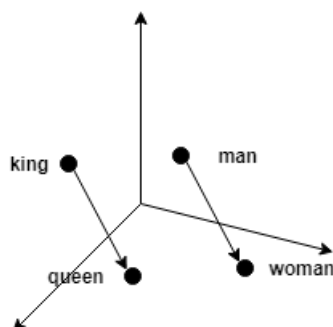
Aurre entrenatutako esanahi-bektoreak printzipio honen arabera ikasten dira: “Antzeko testuinguruetan agertzen diren tokenak antzeko esanahia izango dute”. Esanahi-bektoreak era horretan ikasita, tokenen arteko erlazio semantikoak islatzen dira bektoreetan.

Adibidez, ingeleseko *man*, *woman*, *king* eta *queen* hitzen errepresentazio bektoriala izanik, *woman* eta *man* bektoreen arteko kenketa *king* bektoreari gehitzen bazaio, *queen* da lortu den bektoretik hurbilen dagoen token baten errepresentazioa (3.20 irudia):

$$king + (woman - man) \sim queen$$

Beraz, bektoreek gizon-emakume erlazio semantikoa gordetzen dute.

Algoritmo desberdinak garatu dira esanahi-bektoreak ikasteko, ezagunenak: *word2vec* (Mikolov et al., 2013), GloVe (Pennington et al., 2014) eta fastText (Joulin et al., 2016). Esan bezala, denek printzipio berdinen arabera ikasten dituzte hitzen errepresentazioa baina ikuspegi desberdinetatik. Hemen GloVe aztertuko da sakonago, aurrerago erabiltzen dena delako.



3.20 Irudia: *man*, *woman*, *king*, *queen* hitzen arteko analogia

GloVe

GloVe [Pennington et al.-ek \(2014\)](#) hitzen errepresentazio banakatua lortzeko proposatutako eredu da. Sortutako momentuan existitzen ziren bi teknika nagusien abantailak konbinatzen ditu: matrize globalen faktORIZAZIOA eta *local context window* metodoak.

Matrize globalen faktORIZAZIOAN aljebra linealeko matrizeak faktORIZATZEKO teknikak erabiltzen dira *corpus*-aren gaineko terminoen maiztasun matrizean dimentsio erredukzioa gauzatzeko. Mota horretako teknikak ondo ustiatzen) dute *corpus*-eko informazio estatistikoa, baina analogiako atazetan, adibidez lehen aipatu den *man*, *woman*, *king*, *queen* erlazioa, gaizki moldatzen dira.

Local context window metodoetan hitzen arteko erlazio semantikoak ikasten dira *corpus*-ean lerroz-lerro leiho moduko bat pasaraziz. Bi modura ikas dezake ereduak, bai leihoko hitz bat emanik ingurukoak iragarritz (*skip-gram model*), bai testuingurua emanik hitz bat iragarritz (*continuous bag-of-words model*). Metodo horiek hobeto egiten dute analogia atazetan, baina ez dute informazio estatistikoa erabiltzen.

GloVe ereduak *corpus*-eko informazioa era estatistiko globalean eskuratu, eta *corpus*-eko edozein bi hitz baino ez hartuta, beraien arteko erlazio semantikoak adierazten duen funtzioa lortzea du helburu. Hori lortzeko funtzioak bete beharreko propietateak aztertu, eta hortik abiatuta erregresio eredu bilineal logaritmikoak helburua lortzeko egokiak direla ondorioztatzen dute [Pennington et al.-ek](#). Termino-termino agerpen kopuruaren gainean ikasten duen pisatutako karratu txikiaren eredu proposatzen dute.

4. KAPITULUA

Diseinua eta implementazioa

Ereduaren diseinua egiteko [Du et al.](#)-ek egindako lana hartu da oinarritzat, zehazki esaldi mailarako garatutako arkitektura. Diseinuan hainbat aspektu hartu dira kontuan: datu-multzoa, datu-multzoaren aurreprozesaketa, erreduaren arkitektura, erreduaren entrenamendua eta galderen sorkuntza. Azkenik, diseinatutakoaren implementazioa egin da.

4.1 Datu-multzoa

Proiektu hau garatzeko SQuAD datu-multzoaren lehenengo bertsioa erabili da; hori dela eta, horren xehetasunak aztertu dira.

[SQuAD](#), ingelesez *Stanford Question Answering Dataset*, [Rajpurkar et al.](#)-ek (2016) irakurizko ulermenerako sortutako datu-multzoa da. Bertan 100.000 galdera-erantzun bikoite baino gehiago jasotzen dira.

Datu-multzoa sortzeko ingeleseko wikipediako 536 artikulua erabili ziren eta *crowdworker*-en bitartez sortu ziren galderak. Hiru pausotan sortu zen datu-multzoa: pasartearen eskuraketa, pasartearen galdera-erantzun pareak eskuratu eta galderen kalitatearen azterketa.

Pasartearen eskuraketa

Kalitatezko artikulua lortzeko *Project Nayuki's-en Wikipedia's internal PageRanks* erabili zen sailkapeneko 10000 artikulua hobereak eskuratzeko; haietatik ausaz 536 aukeratu

ziren. Artikulu bakoitzeko 500 karaktere baina luzeagoak diren paragrafoak atera ziren, irudiak, taulak eta abarrekoak kenduta; guztira 23215 paragrafo lortuz.

Pasarteen galdera-erantzunak lortu

Pasarteen galderak sortzeko *crowdworker*-ak erabili ziren. Pertsona bakoitzak gutxienez 4 minutu pasa behar zituen paragrafo bakoitzaren gainean galderak sortzeko. Paragrafo bakoitzeko 5 galdera sortu eta erantzun behar zituzten. Erantzuna testuko zati bat izan behar zen, eta norberaren hitzak erabiliz galderak sortzea eskatzen zen.

Galderen kalitatearen azterketa

Sortutako galderen kalitatea bermatzeko *crowdworker* desberdinak erabili ziren. *Crowdworker*-ei paragrafoa eta sortutako galdera bat ematen zitzaien eta paragrafoan galderaren erantzuna zuen zatirik motzena adierazi behar zuten. Prozesu horren ostean galderen %2.6ak erantzuna ez zuela adierazi zen; hau da, sortutako galdera egokia ez zela.

4.2 Aurreprozesaketa

Eredua datu-multzoarekin entrenatu ahal izateko zenbait aurrepauso eman behar dira. Datu-multzoa galderen sorkuntzako atazara egokitu behar da; bai eta hiztegia sortu, eta hiztegiko hitzei dagozkien esanahi-bektoreak lortu.

4.2.1 Datu-multzoaren aurreprozesaketa

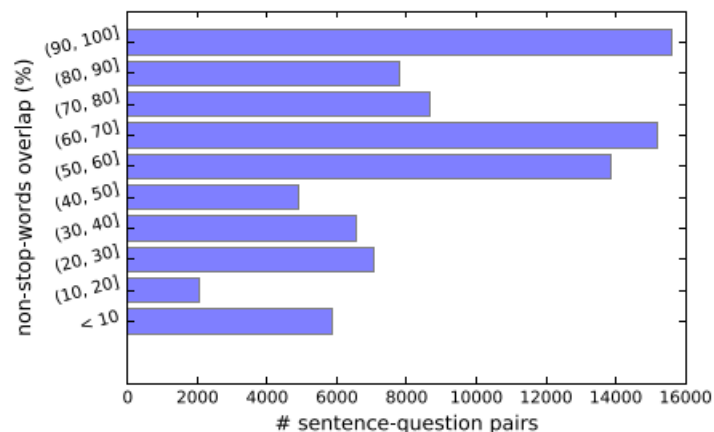
Aipatu da SQuAD galdera-erantzun sistemen sorkuntzarako datu-multzo bezala proposatu zela, non galdera bat emanda erreferentziak esaldi batean erantzuna non dagoen iragarri behar den. Hala ere, datu-multzoari buelta emanda; hau da, esaldiak sarrera bezala tratatuz eta galderak irteera bezala, galdera-sorkuntzaren entrenamendua egiteko datu-multzo bezala erabili daiteke. Datu-multzoari zenbait aldaketa egiten zaizkio: tokenizazioa, kimaketa, ikur bereziak gehitu, eta abar. Hurrengo lerroetan egindako aldaketak era zehatzagoan deskribatuko dira.

Datu-multzoaren aurreprozesaketa

Standford-eko Unibertsitatean garatutako hizkuntza naturala prozesatzeko CoreNLP (Manning et al., 2014) tresna erabiltzen da tokenizazioa eta esaldien banaketa egiteko. Datu-multzo osoa letra xehera pasatzen da baita ere. Galdera bakoitzari dagokion esaldia egokitzen zaio eta esaldi hori erabiliko da sarrera gisa.

Kimaketa

Datu-multzoak 4.1 irudian ikusten den esaldi-galdera pareen arteko *non-stop* hitzen gainezarpenaren distribuzioa ikusten da; *non-stop* garrantzi semantikoa duten hitzak dira. Galdera-erantzun pareen %6.67ak ez du *non-stop* hitzik partekatzen, gehienbat galdera bakoitzari dagokion erantzuna apuntatzean egindako errorearengatik. Hori dela eta, guxtienez *non-stop* hitz bat partekatzen duten esaldi-galdera pareak hartzen dira soilik.



4.1 Irudia: Esaldi-galdera pareen arteko *non-stop* hitzen arteko gainezarpen proportzioa. (Du et al., 2017)

Ikur bereziak

Galderen hasieran <s> eta amaieran </s> ikurrak gehitzen dira. Horrela dekodetzailerako lehenengo ikurra <s> da, galderaren hasiera. </s> ikurrak dekodetzailerak galdera sortzeaz amaitu duen jakiteko balio du.

Datu-multzoari aldaketa horiek egin ondoren guztira 70 mila esaldi-galdera pare lortzen

dira, non esaldiek 30 token inguru dituzten eta galderek 10 token inguru. Esaldi bakoitzak galdera bat baino gehiago izan dezake, beraz, esaldi bakoitzak batez beste 1.4 galdera ditu.

SQuAD datu-multzoaren sortzaileek galdera-erantzun sistemen sorkuntza sustatzeko lehia-keta moduko bat proposatu zuten; hori dela eta, datu-multzoko ebaluaziorako zatia ez dago ikusgai. Ikusgai dagoen zatia hartzen da datu-multzo oso bezala eta ausaz ondoko proportzioan banatzen da: *train %80, dev %10 eta test %10*.

4.2.2 Hiztegia

3.6 atalean ikusi den bezala, sare neuronalari ezin zaizkio tokenak zuzenean pasa; tokenen zenbakizko errepresentazioa behar da. Kasu honetan esanahi-bektoreak erabiltzen dira, baina lehenengo, indexatutako hiztegiak sortzen dira. Ondoren, esanahi-bektoreen matrize bat sortzen da, non token bakoitzari dagokion indizean token horri dagokion esanahi-bektorea dagoen.

Bi hiztegi desberdin egiten dira, bat sarrerako esaldientzat eta beste bat irteerako galderentzat. Hizkuntza berarekin lan egiten bada ere, sarrerako esaldietan eta irteerako galderetan agertzen diren maiztasun handieneko tokenak desberdinak direlako sortzen dira bi hiztegiak.

Sarrerako hiztegia datu-multzoko sarrerako esaldietan agertzen diren maiztasun handieneko 45 mila tokenek osatzen dute. Irteerako hiztegia, berriz, datu-multzoko irteerako galderetako maiztasun handieneko 28 mila tokenek osatzen dute. Bi hiztegietan lau ikur berezi gehitzen dira:

- `<s>`: galderaren hasiera adierazteko.
- `</s>`: galderaren amaiera adierazteko.
- `<unk>`: hiztegian ez dauden tokenak adierazteko.
- `<blank>`: betegarri bezala erabiltzen da.

Hiztegiak sortu ondoren datu-multzoko token guztiak dagokien hiztegiko indizeekin ordezkatzeko dira, hiztegian agertzen ez diren tokenak `<unk>` tokenari dagokion indizearekin ordezkatzeko.

4.2.3 Esanahi-bektoreak

4.2.3 azpiatalean azaldutako GloVe esanahi-bektoreak erabiltzen dira; zehazki [Common Crawl](#)-eko 840 mila milioi token eta 2.2 milioieko hiztegia duen datuekin entrenatutako 300 dimentsioko esanahi-bektoreak ([glove.840B.300d](#)). Sarearen ikasketan bektore hauek ez dira eguneratzen; hau da, tokenen errepresentazioak finkoak dira.

Hiztegi bakoitzeko tokenen GloVe bektoreak bilatzen dira, eta token baten bektorea ez badago ausaz sortutako bektore bat esleitzen zaio. Era erosoan erabili ahal izateko hiztegi token guztien esanahi-bektoreak dituen matrizea sortzen da, $hiztegi_tamaina \times 300$ dimentsiokoa. Token bati hiztegian dagokion indizea eta matrizean dagokion indizea berdinak dira.

4.3 Arkitektura

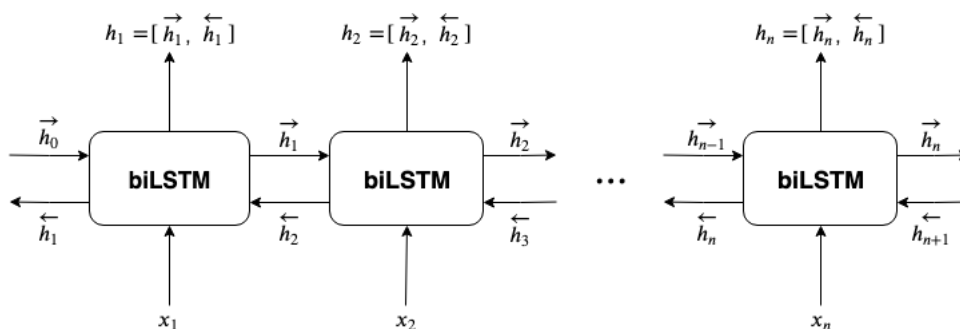
Ereduak 3.5 atalean azaldutako atentzio-mekanismoa duen kodetzaile-dekodemak arkitectura erabiltzen du. Hala ere, oinarritzat hartu den [Du et al.](#)-ek proposatutako arkitekturak zenbait berezitasun ditu. Alde batetik, erabilitako LSTMak ez dira 3.3.2 azpiatalean azaldutakoaren berdin berdinak, bai kodetzaile bai dekodezailean pilaratutako LSTMak erabiltzen dira, eta kodetzailean, gainera, pilaratutako LSTMa bi norantzakoa da.

4.3.1 Bi norantzako LSTMa

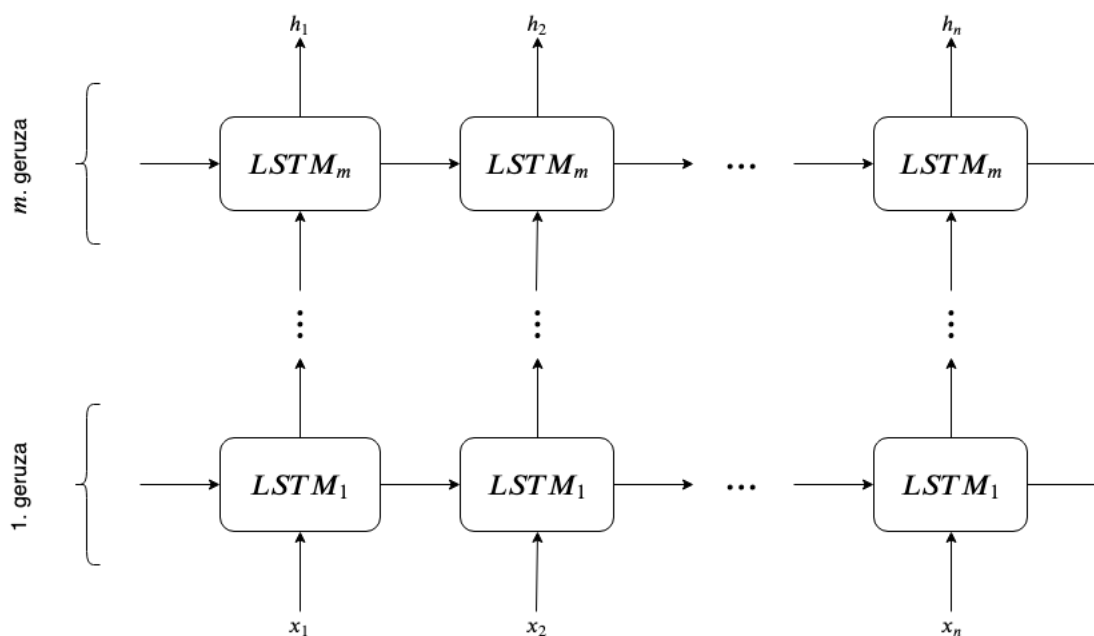
Bi norantzako LSTMan, ingelesez BiLSTM (*Bidirectional LSTM*), sarrerako sekuentzia aurretik atzera eta atzetik aurrera kodetzen da. Kodeketa bi norantzetan eginda BiLSTMak iraganetik zein etorkizunetik lortzen du aldi berean informazioa (4.2 irudia).

4.3.2 Pilaratutako LSTMa

Pilaratutako LSTMa, ingelesez *Stacked LSTM*, pilaratutako hainbat LSTMz dago osatua; hau da, LSTM bat jartzen da bata bestearen gainean, eta, beraz, baten irteera hurrengoaren sarrera da. LSTM bakoitzak geruza bat osatzen du (4.3 irudia).



4.2 Irudia: Bi norantzako LSTMa



4.3 Irudia: Pilaratutako LSTMa

4.3.3 Arkitekturaren xehetasunak

Inplementatutako arkitekturako kodetzailea bi geruzako pilaratutako BiLSTM bat da, eta bere egoera ezkutuaren tamaina 600 da, 300eko tamaina norantzko bakoitzean.

Dekodetzailea, berriz, bi geruzako pilaratutako LSTM bat da 600 tamainako egoera ezkutu dituela. Dekodetzaileko hasierako h_0 egoera ezkutu kodetzaileko norantza bakoitzeko, \vec{b}_n eta \overleftarrow{b}_1 , azkeneko egoerarekin hasieratzen da, $h_0 = [\vec{b}_n, \overleftarrow{b}_1]$.

4.4 Entrenamendua

Atal honetan ereduaren entrenamenduaren nondik norakoak aztertzen dira. Alde batetik, eredia entrenatzeko erabilitako kostu-funtzioa eta optimizazio-algoritmoa aztertzen dira. Bestetik, entrenamendua errazteko erabili diren estrategiak azaltzen dira. Azkenik, entrenamenduaren ostean sorkuntzarako erabiliko den ereduaren aukeraketa nola egiten den azaltzen da.

4.4.1 Kostu-funtzioa eta optimizazio-algoritmoa

Kostu-funtziotzat [3.2.2](#) azpiatalean azaldutako egiantza-funtzio negatiboa erabiltzen da. Izan ere, diseinatutako ereduaren helburua dekodetzaileak irteerako hiztegiaren gainean iragartzen duen probabilitate-banaketa eta benetako galderek jarraitzen dutenaren arteko diferentzia ahalik eta txikiena izatea delako.

Entrenamendurako erabiltzen den optimizazio-algoritmoa gradientearen jaitsiera estokastikoa (*SGD*) da, [3.2.3](#) azpiatalean aipatu zen bezala, mini-sorta bidezko gradientearen jaitsiera estokastikoari *SGD* deitzen zaio.

Sorten tamain 64koa izan da; hau da, kostu-funtzioaren gradientea 64 adibiderekin kalkulatu da, eta, ondoren, sareko pisuak eguneratzen dira. *train* zatiko adibideak 64 tamainako sortetan banatzen dira eta adibide guztiak erabili direnean epoka bat igaro dela esaten da. Sarea entrenatzeko 15 epoka egin dira, hortik aurrera *overfitting* gertatzen dela ikusi delako.

Optimizazio-algoritmoarekin erabiltzen den ikasketa-tasa 1.0 izan da lehenengo 8 epoketan; hortik aurrera epoka bakoitzean erdira zatitu da ikasketa-tasa. Epoka bakoitzeko sortak ausaz sortzen dira, adibideen ordenak sarearen ikasketan eraginik ez izateko.

Sorta osoa aldi berean ereduari sartu ahal izateko, sarrerako esaldi guztiak eta irteerako galdera guztiak luzera berekoak izan behar dira. Sortak ausaz sortzen direnez, sortako esaldi guztiek luzera berdina izateko `<blank>` ikurra gehitzen zaie esaldiaren hasieran, eta berdina egiten da galderekin baina `<blank>` ikurra galderaren amaieran gehituz ([4.4](#) irudia).



4.4 Irudia: *Padding*

4.4.2 Entrenamendua errazteko estrategiak

Entrenamenduan zehar zenbait estrategia erabiltzen dira entrenamendua errazteko: Parametroen hasieraketa, *dropout*, *teacher forcing* eta *gradient clipping*.

Parametroen hasieraketa

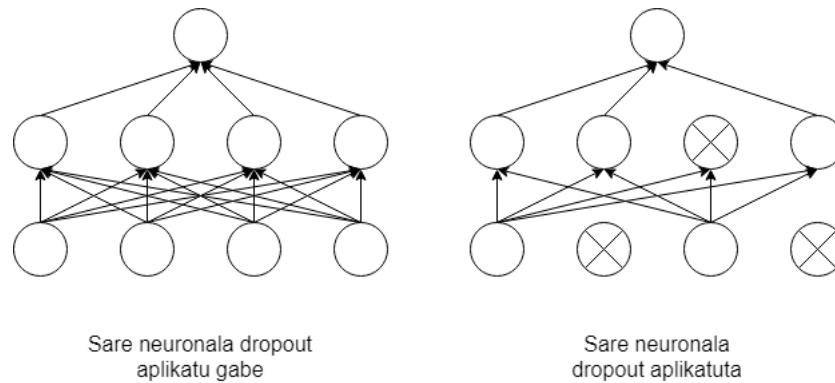
Sarearen parametroak probabilitate-banaketa zehatz bat jarraituz hasieratu ohi dira, sarearen irteerak oso balio txikiak edo oso handiak izatea ekiditeko. Kasu honetan $U(a, b)$ banaketa uniforme jarraitua erabiltzen da, $a = -0.1$ eta $b = 0.1$ izanik:

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \vee x > b \end{cases}$$

Dropout

Dropout erregularizazio-teknika bat da (Srivastava et al., 2014). Erregularizazioa *overfitting*-a ekiditeko ikasketa-algoritmo bati egiten zaion edozein aldaketa da. *Dropout* teknika entrenamenduan sare neuronal bateko ausaz aukeratutako zenbait unitate eta beraien konexioak kontuan ez hartzean datza (4.5 irudia). Era horretan neurona konketuak datu-multzoan espezializatzea ekiditen da.

Entrenamenduan *dropout*-a aplikatzen da bai kodetzaileko bai dekodetzaileko LSTMen geruzen artean 0.3ko probabilitatearekin.

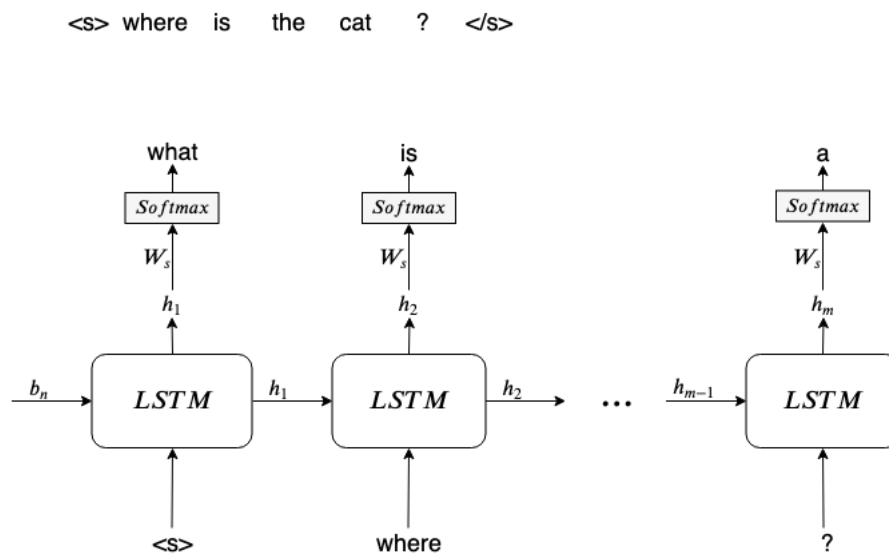


4.5 Irudia: Dropout

Teacher forcing

Teacher forcing-a sare errekurrentek entrenatzeko modu azkar eta eraginkorra da. Dekodetzaileko denbora-une bakoitzeko sarrerarako sareak aurreko denbora-unean sortutako irteera erabili ordez erreferentziazko irteera-sekuentzian dagokion elementua erabiltzean datza (4.6 irudia).

Sare errekurrentea era horretan entrenatuta askoz azkarrago ikasten du, ez duelako menpekotasunik aurreko pausoetan sortutako irteeren erroreekin.



Gradient clipping

Entrenamenduan zehar gerta daiteke gradienteak oso balio handiak hartzea, horrek sarearen parametroen eguneraketan aldaketa handiegiak egitea eragiten du, entrenamendua ezegonkorra eginez.

Hori ekiditeko era bat *gradient clipping* egitea da; hau da, gradientearen balio batetik aurrera bere balioa mugatzen da. Horrek ez du gradientearen norabidea aldatzen, bakarrik bere magnitudea mugatzen du. Entrenamenduan *gradient clipping* aplikatu da gradientearen normak 5 balioa gainditzen duenean.

4.4.3 Eredu-aukeraketa

Entrenamenduan zehar eta balidazioan epokaz epoka lortzen diren ereduak ebaluatzeko kostu-funtzioaren balioa zuzenean erabili ordez *perplexity*-a erabiltzen da, interpretagarriagoa delako. Zehazki *perplexity per word* erabiltzen da eta honela definitzen da:

$$\text{Perplexity per word} = \exp\left(\frac{-\sum_{i=1}^N p(x) \cdot \log \hat{p}(x)}{N}\right)$$

Beraz, *perplexity per word* N adibideren gaineko kostu-funtzioaren, egiantza-funtzio negatiboaren, balioaren batez bestekoaren esponentziala da.

Epoka bakoitzaren ostean datu-multzoko *valid* zatian kalkulatu da *perplexity*-a. *Perplexity* baxuena duen epokako ereduak aukeratzen da.

4.5 Sorkuntza

Behin entrenatutako ereduak izanik, helburua edozein testu emanda galdera onak sortzea da; hau da, probabilitate handieneko galderak lortzea:

$$\bar{y} = \arg \max_y \prod_{t=1}^n P(y_t | x, y_{<t})$$

Probabilitate handieneko galderak aurkitzeko, galdera posible guztiak aztertu beharko litrateke eta hori NP multzoko problema bat da. NP multzoa Turingen makina ez determinista batean denbora polinomikoan ebazgarriak diren problemen multzoa da, P multzoa,

berriz, Turingen makina determinista batean denbora polinomikoan ebazgarriak direnak. $P = NP$ denaren froga izan gabe ezinezkoa da jakitea problema era zehatzean ebazteko algoritmo eraginkorrik existitzen den; hori dela eta, bilaketa-heuristikoko algoritmoak erabiltzen dira. Bilaketa-heuristikoko algoritmoek ez dute bermatzen emaitzarik hobereana aurkituko denik, baina era eraginkor batean emaitza on bat lortzen dute.

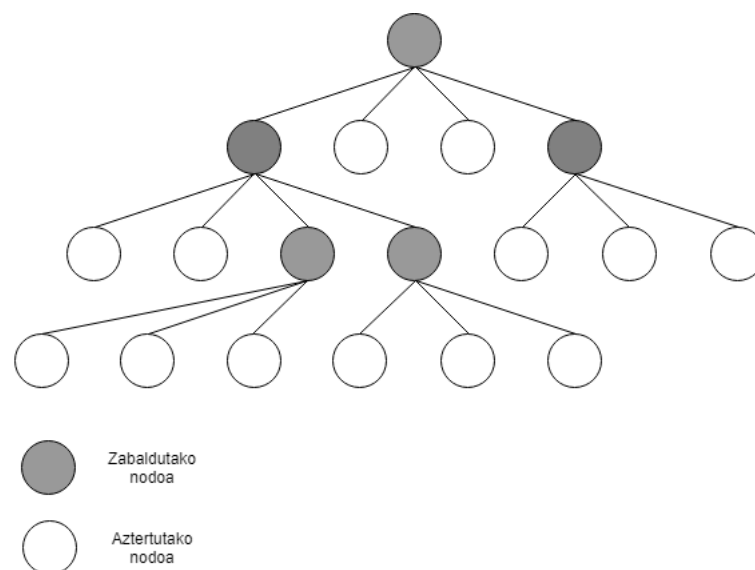
Algoritmorik sinpleena *best-first* litzateke; bertan momentu oro probabilitate handieneko tokena aukeratzen da. Konputazionalki oso eraginkorra da, ordena linealekoa delako, baina sortutako galdera probabilitate handienekotik oso urrun egon daiteke.

Beste bilaketa heuristiko bat *beam search* da. *Beam size* baxu bat mantenduta ez du denbora asko hartzen eta *best first* algoritmoak baino emaitza hobegoak lortzen ditu.

4.5.1 *Beam search*

Beam search bilaketa heuristiko algoritmo bat da, *best-first* algoritmoaren hobekuntza. Pauso bakoitzean bakarrik nodorik hobereana hedatu ordez, n nodorik hoberenak hedatzen ditu, non n *beam size* den.

Galderak sortzeko, hasierako nodoa hasierako ikurra duen galdera da. Pauso bakoitzean hedatzen diren nodoei token berri bat gehituz lortzen den puntuazio hobereana duten n nodo berriak jarraituko dira hedatzen. Puntuazioa gehitzen diren tokenen probabilitatearen logaritmoa gehituz lortzen da.



4.7 Irudia: *Beam search* adibidea (*beam size*= 2)

4.5.2 Kopiatzeko mekanismoa

Aipatu da bai sarrera bai irteerako hiztegian <unk> ikurra dagoela hiztegian ez dauden hitzak adierazteko. Galderak sortzean gerta daiteke sortutako galderan tokenen bat <unk> izatea. Kasu horietan *kopiatzeko mekanismoa* erabiltzen da.

<unk> ikurra sortu den momentuan sarrerako esaldian atentzio-pisu altuena zuen tokena kopiatzen da. Mekanismo honek ahalbidetzen du gutxi agertzen diren hitzekin galderak ondo sortzea.

4.6 Inplementazioa

Aipatu da diseinua egiteko [Du et al.](#)-ek egindako lana hartu dela oinarritzat, inplementaziorako, berriz, ez. [Du et al.](#)-ek Torch7-rekin egin dute inplementazioa OpenNMT sistemaren gainean. Torch ikasketa automatikorako kode-irekiko liburutegia da, Lua programaziolengoaian oinarritua.

OpenNMT [Klein et al.](#)-ek makina bidezko itzulpenerako garatutako kode-irekiko erreminta multzoa da. Makina bidezko itzulpenerako erabiltzen diren arkitekturak galderen sorkuntzarako erabiltzen direnen antzekoak dira; hori dela eta, [Du et al.](#)-ek ahalik eta aldaketa txikienak eginez OpenNMT moldatu dute haien eredia sortzeko.

Proiektu honen garapena Pytorch-en inplementatzea erabaki da, zerotik hasita. Pytorch Torch liburutegian oinarritzen den Python programaziolengoaierako liburutegia da, gehienbat Facebook-ek garatua. Pytorch erabiltzeko eta ulertzeko erraza da, gainera atzean komunitate handi bat dauka, proiektu bat garatzean oso gauza garrantzitsua dena.

OpenNMT-k Pytorch-erako bertsioa badauka ere, eta era antzeko beste *framework*-ak existitzen badira ere, inplementazio osoa egitea erabaki da; arkitektura bat zerotik inplementatzeak eskatzen duen ulermen maila lortzea premiazkoa ikusi delako.

Proiektua garatzeko sortutako kodea eskuragarri jarri da *bitbucket* plataforman ondorengo orrian: [galderensorkuntza](#).

5. KAPITULUA

Ebaluazioa eta emaitzak

Kapitulu honetan inplementatutako eredia ebaluatzeko erabilitako metriken funtzionamendua azaltzen da. Ondoren, eskuzko ebaluazioaren inguruan jarduten da. Azkenik, egindako esperimentazioa eta lortutako emaitzak komentatzen dira.

5.1 Ebaluazio automatikoa

Ebaluazio automatikoa egiteko [Chen et al.-ek \(2015\)](#) sortutako ebaluaziorako programa erabili da. Bertan zenbait metrika daude: BLEU, METEOR eta ROUGE. Metrika horiek erabili dira sortutako galderen ebaluazio automatikoa egiteko. Atal honetan metrika horiek zertan oinarritzen diren aztertuko da. Bestetik, erabili ez den arren, duela gutxi ateratako galderen erantzungarritasuna kontuan hartzen duen metrika berri bat ere aztertzen da.

5.1.1 BLEU

BLEU (*Bilingual Evaluation Understudy*) [Papineni et al.-ek \(2002\)](#) sortutako itzulpen automatikoan erabiltzen den metrika da, baina galderen sorkuntzarako, eta, oro har, testua sortzen den edozein atazarako erabili da ere. Metrika ondorengoan oinarritzen da: sortutako galdera bat kalitate handiagokoa izango da, erreferentziazko galderaren antzekoagoa

den heinean. Bi esaldien arteko antzekotasuna honela definitzen da BLEUn:

$$P_n = \frac{m}{w_t}$$

non m automatikoki sortutako galdera eta erreferentziako galderaren arteko n -grama komun kopurua den, n -grama komunak behin baina ez kontuan hartuta eta w_t automatikoki sortutako galderaren n -grama kopurua den. 1-gramak kontuan hartu badira $BLEU_1$ adierazi ohi da, 2-gramak kontuan hartu badira $BLEU_2$, 3-gramak $BLEU_3$, etabar.

Definitutako esaldien arteko antzekotasuna erabiliz, esaldi motzei ematen zaie lehentasuna; hori dela eta, esaldi motzentzako penalizazioa erabiltzen da. Automatikoki sortutako galdera eta benetakoaren arteko luzeraren diferentzia oso handia bada, ezin da esan esaldiak antzekoak direla. Hori zigortzeko hurrengo penalizazio erabiltzen da, non c automatikoki sortutako galderaren luzera den eta r erreferentziako galderarena:

$$PB = \begin{cases} 1 & c > r \text{ bada} \\ e^{1-\frac{r}{c}} & \text{bestela} \end{cases}$$

BLEUren balioa kalkulatzeko batezbesteko geometrikoa erabiltzen da, non $\sum_{n=1}^N w_n = 1$ den eta normalean $w_n = \frac{1}{N}$ erabiltzen den:

$$BLEU = PB \cdot \exp\left(\sum_{n=1}^N w_n \cdot \log P_n\right)$$

5.1.2 METEOR

METEOR (*Metric for Evaluation of Translation with Explicit ORDERing*) BLEUK dituen zenbait arazo konpontzeko [Banerjee and Lavie-ek \(2005\)](#) sortutako metrika da. METEOR metrika unigramen zehaztasun eta berreskuratzearen batezbesteko armonikoan oinarritzen da.

BLEUn bezala, lehenengo, automatikoki sortutako galdera eta erreferentziako galderaren tokenen arteko mapaketa egiten da; METEOREn printzipio honen arabera egiten da mapaketa: automatikoki sortutako galderako token guztiek erreferentziako galderaren token batera edo ezta batera dago mapatua, ahalik eta mapaketa gehien lortu nahi dira eta mapatze kopuru berdina duten bi mapaketaren artean gurutzaketa gutxienean daukana aukeratzen da.

Mapaketa izanik, doitasuna eta estaldura kalkulatzeko dira. Doitasuna honakoa da, m automatikoki sortutako galderaren eta erreferentziako galderaren arteko komuneko unigramak eta w_t automatikoki sortutako galderaren unigrama kopurua izanik:

$$P = \frac{m}{w_t}$$

Estaldura, berriz, honakoa, non w_r erreferentziako galderaren unigrama kopurua den:

$$R = \frac{m}{w_r}$$

Doitasuna eta estaldura batezbesteko armonikoa erabiliz konbinatzen dira:

$$F_{mean} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P}$$

Penalizazio faktore bat sartzen da baita ere. Automatikoki sortutako galdera eta erreferentziakoaren arteko mapaketan auzokideak ez diren zenbat eta mapaketa gehiago izan orduan eta handiagoa izango da penalizazioa. Penalizazioa honela kalkulatu da, non c erreferentziako galderan eta automatikoki sortutako galderan albokoak diren unigramen multzo kopurua den eta u_m mapatuak izan diren unigrama kopurua:

$$p = 0.5 \left(\frac{c}{u_m} \right)^3$$

Azkenik, p penalizazioa eta F_{mean} doitasun eta estalduraren batezbesteko armonikoa izanik, honela kalkulatu da METEOR balioa:

$$METEOR = F_{mean} \cdot (1 - p)$$

5.1.3 ROUGE

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) [Lin-k \(2004\)](#) sortutako metrika multzoa da. Hemen ROUGE-L aztertzen da, azpisekuentzia komunik luzeenean oinarritzen den metrika.

Bi esaldiren azpisekuentzia komunik luzeena horrela definitzen da: bi esaldiek ordena berean daukaten hitzen azpisekuentziarik luzeena. Hori erabilia ROUGE-L-ren intuizioa

ondokoa da: zenbat eta luzeagoa izan azpisekuentzia komunik luzeena orduan eta antzekoagoak izango dira bi galderak.

Izan bitez x erreferentziazko galdera m luzerakoa, y automatikoki sortutako galdera n luzerakoa eta $LCS(x, y)$ erreferentziazko galdera eta automatikoki sortutako galderaren arteko azpisekuentzia komunik luzeena. ROUGE-L doitasun eta estalduraren batezbesteko armonikoan oinarritzen da baita ere, beraz horrela kalkulatzen dira R_{lcs} estaldura eta P_{lcs} doitasuna:

$$R_{lcs} = \frac{LCS(x, y)}{m}$$

$$P_{lcs} = \frac{LCS(x, y)}{n}$$

Doitasuna eta estaldura batzeko batezbesteko armonikoa erabiltzen da, $\beta = P_{lcs}/R_{lcs}$ definituz:

$$\text{ROUGE-L} = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2P_{lcs}}$$

ROUGE-L erabiltzearen abantaila zera da: n-gramak ordez azpisegida komunik luzeena erabiltzen duenez, ez du eskatzen token komunak jarraian egotea baizik eta esaldi barruan ordena berean, eta, gainera, ez da aurretik zehaztu behar n-gramaren luzera. Desabantaila, berriz, bakarrik azpisekuentzia komunik luzeena kontuan hartzen duenez, beste azpisekuentzia motzagoak ez dira islatzen bukaerako puntuazioan.

5.1.4 Galderen erantzugarritasunean oinarritutako metrika

Zenbait kritika egin zaie azaldutako erako metrikeri, askotan metriken balioak eskuzko ebaluazioarekin bat ez datozelako (Callison-Burch et al., 2006; Bhattacharyya et al., 2007; Callison-Burch, 2009); izan ere, erreferentziazko galdera eta automatikoki sortutako galderaren arteko n-gramen parekotasunean edo komuneko azpisekuentzia luzeenean baino ez dira oinarritzen, eta ez dituzte kontuan hartzen sortutako galderek bete beharreko beste zenbait propietate garrantzitsu. Hori dela eta, Nema and Khapra-k (2018) galderaren erantzugarritasuna kontuan hartzen duen metrika bat proposatzen dute, beste metriekin batera erabili daitekeena. Metrika berri horren eta eskuzko ebaluazioaren arteko korrelazioa handiagoa da.

Galdera baten erantzugarritasuna lau elementuren presentziaren menpekoa da: hitz garrantzitsuak, entitate izenduak, galdera hitzak eta hitz funtzionalak. Metrika elementu horien agerpen kopuruan oinarritzen da.

Izan bedi, $c(S_r), c(S_n), c(S_q)$ eta $c(S_f)$ erreferentziazko galderan mapatzea duten automatikoki sortutako galderako hitz garrantzitsu, entitate izenduak, galdera hitz eta hitz funtzional kopurua, hurrenez hurren; $i \in \{r, n, q, f\}$, $\sum_i w_i = 1$ non w_i doitu daitezkeen pisuak diren eta $|l_i|, |r_i|$ i . kategoriako l automatikoki sortutako galderako edo r erreferentzia galderako hitz kopurua:

$$P_{avg} = \sum_i w_i \frac{c(S_i)}{|l_i|} \quad R_{avg} = \sum_i w_i \frac{c(S_i)}{|r_i|}$$

$$\text{Erantzungarritasuna} = 2 \frac{P_{avg} R_{avg}}{P_{avg} + R_{avg}}$$

Erantzungarritasuna edozein metrikarekin batera erabili daiteke ondoko eran, $\delta \in \{0, 1\}$ definituz:

$$\text{Q-Metrika} = \delta \cdot \text{Erantzungarritasuna} + (1 - \delta) \cdot \text{Metrika}$$

Hala ere, metrika hau ez da erabilia izan sistema bat ebaluatzeko, duela gutxi ateratakoa delako.

5.2 Eskuzko ebaluazioa

Ebaluazio-metrika automatikoen gabeziak direla eta, ereduak ebaluatzean gehienetan metrika automatikoak erabiltzeaz gain eskuzko ebaluazioa ere egiten da. Baina eskuzko ebaluazioa egiteko ez dago zehaztutako gida-lerrorik, eta, beraz, lortutako ebaluazioak, oro har, ez dira konparagarriak izaten eredu batetik bestera.

Eskuzko ebaluazioa *Crowdsourcing* bidez egiteko dago pentsatua. *Crowdworker* bakoitzari testu bat eta testu horretatik sortutako galderak erakusten zaizkio. *Crowdworker*-ak galdera horiek ebaluatu behar ditu gida-lerro batzuk jarraituta.

Galderak arlo pedagogikoaren ikuspegitik ebaluatzeko hurrengo gida-lerroak proposatu dira:

- Jarioa: galdera gramatikalki zuzena den.
- Anbiguetasuna: galderan anbiguetasunik duen.

- Erantzugarritasuna: galdera ereantzutea posible den.
- Adierazgarritasun pedagogikoa: galderak alde pedagogikotik garrantzia daukan.
- Ulermen mota: Galderak lantzen duen ulermen mota.

Eskuzko ebaluaziorako galderen kalitatea hezkuntza ikuspuntutik ebaluatzeko gida-lerroak ezartzeko asmoz, IXA taldeak parte hartzen duen [enetCollect](#) ikerketa-sarearen barruan 2019ko urtarrilean egin zen *crowdfest*-ean proposatutako gida-lerroak ebaluatu ziren. enetCollect hizkuntzen ikasketa eta Crowdsourcing-a konbinatzeko sare europar berria da, [COST Actions](#)-ek babestua. Gradu Amaierako Proiektu honen barruan aipatutako *crowdfest*-erako eskuzko ebaluaziorako gida-lerroak ebaluatzeko interfazearen lehenengo bertsioa garatu zen *Pybossa crowdsourcing*-erako *framework*-arekin (5.1 irudia).

5.3 Esperimentazioa eta emaitzak

Proiektu honen helburua [Du et al.](#)-ek garatutako eredua erreplikatzeko izan da, beraz, esperimentazioa egiteko beraiek azaldutako pausoak jarraitu dira. Helburua erreplikatzeko denez, [Du et al.](#)-ek erabilitako hiperparametro berdinak erabili dira, zeinak ereduaren garapenean aukeratuak izan diren:

- Iturriko hiztegi tamaina: 45 mila.
- Helburuko hiztegi tamaina: 28 mila.
- Esanahi-bektoreen dimentsioa: 300.
- Kodetzaileko LSTMaren egoera ezkutuaren dimentsioa: 600.
- Dekodetzaileko LSTMaren egoera ezkutuaren dimentsioa: 600.
- LSTMen geruza kopurua: 2.
- LSTMen geruzen arteko dropout probabilitatea: 0.3.
- Parametroen hasieraketa: $U(-0.1, 0.1)$.
- Gradient clipping norma maximoaren balioa: 5.
- Epoka kopurua: 15.

Cell Theory: Contribute

Text

A cell is the smallest structural and functional unit of an organism. Some organisms, like bacteria, consist of only one cell. Big organisms, like humans, consist of trillions of cells. Compare a human to a banana. On the outside, they look very different, but if you look close enough you'll see that their cells are actually very similar.

Most cells are so small that you cannot see them without the help of a microscope. It was not until 1665 that English scientist Robert Hooke invented a basic light microscope and observed cells for the first time, by looking at a piece of cork. You may use light microscopes in the classroom. You can use a light microscope to see cells (Figure below). But many structures in the cell are too small to see with a light microscope. So, what do you do if you want to see the tiny structures inside of cells?

In 1858, after using microscopes much better than Hooke's first microscope, Rudolf Virchow developed the hypothesis that cells only come from other cells. For example, bacteria, which are single-celled organisms, divide in half (after they grow some) to make new bacteria. In the same way, your body makes new cells by dividing the cells you already have. In all cases, cells only come from cells that have existed before. This idea led to the development of one of the most important theories in biology, the cell theory.

As with other scientific theories, many hundreds, if not thousands, of experiments support the cell theory. Since Virchow created the theory, no evidence has ever been identified to contradict it.

Question

what is the name of the division that For divide divide into half ?

Answer

Fluency	<input type="radio"/> The question is grammatically correct and idiomatic/natural	<input type="radio"/> The question is grammatically correct but does not read as fluently as we would like.	<input type="radio"/> There are some grammatical errors in the question	<input type="radio"/> The question is grammatically unacceptable.
Ambiguity	<input type="radio"/> The question itself is completely unambiguous	<input type="radio"/> The question could provide more information	<input type="radio"/> The question is totally ambiguous when asked out of the blue	
Answerability	<input type="radio"/> The question can be answered with the information from the source text.	<input type="radio"/> The question can be answered with the information from the source text plus some world knowledge*.	<input type="radio"/> The question can be answered using some world knowledge*, and without using information from the text	<input type="radio"/> The question can not be answered with the information from the source text, neither using some world knowledge*.
Pedagogical relevance	<input type="radio"/> The question is compulsory to be sure that the reader has understood the text.	<input type="radio"/> The question is relevant to evaluate the reading comprehension level but it is not compulsory.	<input type="radio"/> The question is related to the topic of the text but it does not measure its reading comprehension level.	<input type="radio"/> The question is not related to the topic of the text.
Comprehension type	<input type="radio"/> The question can be answered directly and explicitly from the text (facts, dates, times, locations, entities...)	<input type="radio"/> It is necessary to combine information from various parts of the text in order to answer the question.	<input type="radio"/> It is necessary to combine information from the literal understanding of the text with reader's knowledge and intuitions in order to answer the question.	<input type="radio"/> The question is about what might happen next or after a story ends.
			<input type="radio"/> The question asks about giving a global or comprehensive judgement about some aspect of the text.	<input type="radio"/> The question is about the reader's feelings for the text and the subject.

Submit

5.1 Irudia: Gida-lerroak ebaluatzeko *Pybossa*-rako lehenengo bertsioa

- Sorta tamaina: 64.
- Ikasketa-tasa: 1/8. epoka arte; hortik aurrera epoka bakoitzean zati bi.

Esan den bezala SQuAD datu-multzoa erabili da 4.2 azpiatalean azaldutako aurreprozesaketa eta *train*, *dev* eta *test*-erako banaketa erabiliz. Esperimentazioa *test* zatiaren gainean egiten da.

Alde batetik Du et al.-ek argitaratutako artikuluan jarritako emaitzak hartu dira, eta, bestetik, proiektu honetan inplementatutako ereduarenak (5.5 taula). Horretarako, entrenamenduan lortutako eredurik hobereena hartu da, *test* zatiko sarrerako esaldiak erabili dira ereduarekin galderak sortzeko, eta, ondoren, sortutako galderak eta erreferentziatzko galderekin metrika automatikoak kalkulatu dira Chen et al.-ek sortutako ebaluaziorako programa erabiliz.

Eredua	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
Du et al. esaldi	43.09	25.96	17.50	12.28	16.62	39.75
Gurea	39.74	22.72	14.80	10.03	14.21	27.96

5.1 Taula: Emaitzak

5.5 taulan ikusten denez, Du et al.-en metrikak, oro har, proiektu honetan garatutako ereduarekin lortutakoak baina zertxobait hobetoak dira beti. Diferentzia horren zergatia aztertuta da, inplementatutako ereduaren akatsen bat ote dagoen ikusteko. Bi ereduaren artean diferentzia non egon daitekeen aztertuta da ondoko elementuei erreparaturik: hiztegiak eta esanahi-bektoreak, parametroen hasieraketa, ikasketa eta ebaluazio metrikak.

Lehenik, gure egindako aurreprozesamenduan sortutako hiztegiak eta Du et al.-ek sortutako arteko diferentzia dagoen aztertuta da. Horretarako hiztegien arteko intersektzioa eta diferentzia aztertuz (5.2 eta 5.3 taulak). Zenbait tokenen diferentzia dagoela aztertuta da, bai iturriko hiztegiaren bai helburukoan. Desberdin diren token horien hiztegiaren indizeari erreparaturik, token horiek hiztegiaren azkenengo tokenak zirela aztertuta da. Horren zergatia ulertzeko, gogoratu behar da hiztegiak sortzean tokenak agerpen kopuruaren arabera ordenatzen direla eta token kopuru finitu bat baina ez dela hartzen. Beraz, mozketak egiten den tokenaren inguruan agerpen kopuru berdina duten zenbait token badaude, erabiltzen den ordenatzeko algoritmoaren arabera token batzuk edo beste batzuk geldituko dira kanpoan.

Hiztegiak desberdinak izanik gure inplementazioan eta Du et al.-ek egindakoan <unk> diren tokenak ez dira berdina izango, eta, beraz, sekuentzia horretan erabiltzeko esanahi-bektoreak ezta.

Du et al.-en sarrerako hiztegiko token kopurua:	45001
Gure sarrerako hiztegiko token kopurua:	45004
Intersekzioko token kopurua:	41154
Diferentziako token kopurua:	3850

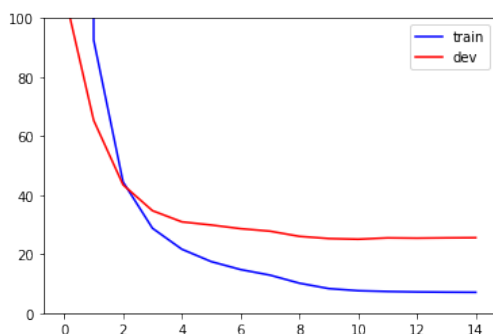
5.2 Taula: Sarrerako hiztegien konparaketa

Du et al.-en irteerako hiztegiko token kopurua:	28001
Gure irteerako hiztegiko token kopurua:	28004
Intersekzioko token kopurua:	21422
Diferentziako token kopurua:	6582

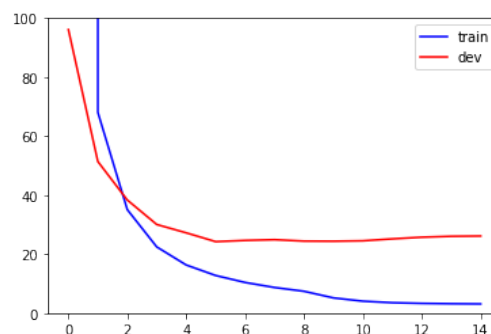
5.3 Taula: Irteerako hiztegien konparaketa

Bestetik, parametroak banaketa uniforme bat jarraituz hasieratzen dira, beraz entrenatzen den bakoitzean sarearen hasierako parametroen balioa desberdina da, estokastikotasuna sartuz sarearen entrenamenduan.

Ikasketa prozesuan *perplexity*-aren eboluzioa aztertu da Du et al. eta gurean, bai *train* zatian bai *dev* zatian (5.2 irudia eta 5.4 taula). Aurrerago aipatutako desberdintasunak kontuan hartuta, entrenamenduaren eboluzioa nahiko parekoa da. Azkenean antzeko *perplexity*-a duen eredua izanik.



(a) *Perplexity*-aren eboluzioa Du et al.-en ereduan



(b) *Perplexity*-aren eboluzioa gure ereduan

5.2 Irudia: *Perplexity*-aren konparaketa

Ebaluaziorako metriken kalkulua berdina da, bi kasuetan Chen et al.-ek sortutako ebaluaziorako programa erabiltzen delako.

Azkenik, Du et al.-ek beraien kodea eskuragarri daukate <https://github.com/xinyadu/nqg> horrian. Beraien kodea exekutatu da, eta ez dira artikulan argitaratutako emaitza berdinak lortu, oro har, metrika guztietan puntu pare bat baxuagoak dira. Hori aipatu den estokastikotasunarengatik da.

Epoka	Du et al.	Gurea
1	108.55	96.08
2	65.34	51.39
3	43.48	38.34
4	34.73	30.04
5	30.94	27.22
6	29.87	24.24
7	28.63	24.65
8	27.78	24.90
9	26.03	24.42
10	25.28	24.38
11	25.06	24.52
12	25.50	25.18
13	25.41	25.74
14	25.51	26.05
15	25.57	26.14

5.4 Taula: Du et al. eta gure ereduaren balidazioko *perplexity*-aren eboluzioen konparaketa

Gauzak horrela, inplementatutako ereduak lortutako emaitzak zertxobait baxuagoak izan harren, arrakastaz berrinplementatu dela ondorioztatu da. Kontuan hartuz etorkizunean ereduak erabili nahi izanez gero, hiperparametroen bilaketa bat egin beharko litzatekeela. 5.5 taulan *test* zatian esaldi bat emanik lortutako galdera eta erreferentziako galdera batzuk agertzen dira.

Sarrerako esaldia	Gure ereduak sortutako galdera	Erreferentziako galdera
All of the departments in the college of science offer phd programs, except for the department of pre-professional studies.	What kind of study does harvard offer?	Which department at notre dame is the only one to not offer a phd program?
Around 1899, professor jerome green became the first american to send a wireless message.	Who was the first american to send a computer message?	Which professor sent the first wireless message in the usa?
In 2014 the notre dame student body consisted of 12,179 students, with 8,448 undergraduates, 2,138 graduate and professional and 1,593 professional (law, m.div., business, m.ed.)	How many students were in washington university?	How many undergrads were attending notre dame in 2014?

5.5 Taula: Gure ereduarekin lortutako galdera batzuen adibidea

6. KAPITULUA

Hobekuntzak eta proposamenak

Kodetzaile-dekodetzaile arkitekturan oinarritzen diren galderen sorkuntza automatikoko sistemak eskuz egindako erregeletan oinarritu ordez zuzenean datuetan oinarritu, eta datu horien gainean orokortzeko gaitasuna badute ere, ez dituzte ulermena lantzen duten galderarik sortzen.

Hizkuntzaren Prozesamenduko beste atazetan ez bezala, non sarrerako sekuentzia eta irteerakoaren elementuen arteko batetik baterako mapatzea egin daitekeen, galderen sorkuntzan esaldi batean oinarrituz galdera bat baino gehiago sor daiteke, erantzuna den esaldi-zatiaren arabera. Gainera, galdera hori desberdina izango da galderaren intenzioa edota zailtasun maila kontuan hartzen badira.

Gauzak horrela, agerikoa da ereduaren arkitekturan hobekuntzak egin behar direla. Kapitulu honetan jadanik artearen egoeran egin diren zenbait hobekuntza aztertu, eta hobekuntza posible berri batzuk proposatzen dira.

6.1 Galderen sorkuntzaren artearen egoeraren hobekuntzak

Gaur egun Adimen Artifizialean dagoen ikerketa oso aktiboa da, gainera, sare neuronalen bidezko galderen sorkuntzak interes handia sortu du azken urteotan. Hori dela eta, Gradu Amaierako Lan hau hasi zenetik, galderen sorkuntza automatikoan ikerketa-bide berriak proposatzen dituzten zenbait artikulu argitaratu dira. Ikerketa-bide berri horien azterketa bat egin da, testuinguru pedagogikoan galdera sakon eta onak sortzen dituen ereduak

jarraitu beharreko lukeen norabidea aztertzeko. Aztertutako artikulu askoren helburua arlo pedagogikorako galderak sortzea ez bada ere, ideia asko erabilgarriak izan daitezke. Zehazki arlo pedagogikora bideratutako aurrera pausoak ere aztertzen dira.

6.1.1 Hobekuntza orokorrak

Du et al.-ek proposatutako, eta proiektu honetan implementatu den, atentziodun kodetzaile-dekodemak arkitekturak zenbait muga ditu. Alde batetik, normalean esaldi batetik galdera bat baino gehiago sor daitezke, galdetu nahi denaren arabera. Baina implementatutako ereduarekin esaldi batekin beti galdera berdina sortuko da, eta, oro har, galdera hori nahiko orokorra izatea eragiten du.

Bestetik, gerta daiteke galdera bat sortzeko esaldian behar den informazio guztia ez egotea; Du et al.-ek paragrafo mailan egiten badute ere, esaldi mailan baino emaitza okerragoak lortzen dituzte; izan ere, sarrera luzeagoa den heinean, sare errekurenteak atal esanguratsuak identifikatzeko zailtasun handiagoa du, informazio gehiago izateaz gain zarata gehiago ere sartzen duelako.

Argitaratutako artikuluek arazo hauek konpontzeko zenbait proposamen aurkezten dituzte, hala nola, erantzunaren kodeketa, galde-hitzaren sorkuntza bereizia, paragrafo mailako eredu hobetuak eta zer galdetu bilatzen duten ereduak. Aipatutako proposamenak kodetzaile-dekodemak arkitekturaren gaineko hobekuntza bezala aurkezten dira; hala ere, zeharo desberdinak diren adimen artifizialeko beste zenbait arkitektura erabiltzen dituzten ereduaren proposamena egon dira, bai eta ataza bat baino gehiago aldi berean entrenatzen dituztenak ere. Jarraian proposamen berrien gora beherak azaltzen dira. 6.1 taulan aipatuko diren zenbait artikuluren eta bertan erabiltzen dituzten tekniken laburpena dago.

Ezaugarri linguistiko gehigarriak

Ereduaren sarrerari tokenen esanahi-bektoreekin batera zenbait ezaugarri linguistiko gehitzen zaizkio, besteak beste, kategoria gramatikala, entitate izenduak, hitzaren kasua. Ezaugarri horiek era bektorialean adierazten dira, eta esanahi-bektoreekin batera kateatzen dira.

Kopiatzeko mekanismoa

[Du et al.](#)-ek hiztegitik kanpoko tokenak tratatzeko *<unk>* tokena erabiltzen dute. Token hori sortzen denean, momentu horretan atentzio gehien duen sarrerako esaldiko hitza kopiatzen dute.

Arazo horri aurre egiteko era desberdinak daude, gehien erabili dena [Gülçehre et al.](#)-ek (2016) proposatutako mekanismoa da. Bertan *<unk>* tokena sortu eta hori ordezkatu ordez, momentu oro token berri bat sortu edo sarrerako esalditik token bat kopiatzeko probabilitatea kalkulatu da. Token berri bat sortzea erabakitzen bada, irteerako hiztegiaren gaineko probabilitate-banaketa erabiliko da; aldiz, tokena kopiatzea erabakitzen bada, sarrerako esaldiaren tokenen gaineko probabilitate-banaketa erabiltzen da.

Erantzunaren kodeketa

Aipatu den bezala, esaldi batetik galdera bat baino gehiago sor daiteke; hori deta eta, zehazki erantzuna izango den esaldi-zatia ereduari esplizituki adieraztea proposatzen da ([Zhou et al., 2017](#); [Zhao et al., 2018](#); [Sun et al., 2018](#); [Song et al., 2018](#); [Kim et al., 2018](#)). Bi modu desberdin proposatu dira erantzunaren kodeketa egiteko.

Lehenengo moduan erantzunaren kokalekua adierazten da. Horretarako, [Zhou et al.](#)-ek BIO etiketatzea erabiltzen dute; hau da, token bakoitzeko erantzunaren hasiera (B), erantzuneko zatia (I) edo erantzunetik kanpoko (O) tokena den adierazten da. [Zhao et al.](#)-ek, berriz, bektore bitarra erabiltzen dute. Azkenik, [Sun et al.](#)-ek erantzunari garrantzi handiagoa emateaz gain erantzunaren inguruko tokenak garrantzi handiagokoak direla ere argudiatzen dute eta entrenagarriak diren posizio-bektoreak baliatuz egiten dute erantzunaren kodeketa.

Bigarren moduan erantzunaren kokalekua adierazi ordez, erantzun-zatia era banandu batean kodetzen da; izan ere, lehenengo erara eginda ereduak erantzuna den esaldi-zatiko tokenak sortzeko joera duelako, erabili ezin diren galderak sortuz. Erantzuna era bananduan kodetzeko beste sare errekurrente bat erabiltzen da. Kodetzeko era hau [Song et al.](#)-ek eta [Kim et al.](#)-ek proposatzen dute.

Galde-hitzaren sorkuntza bereizia

[Sun et al.](#)-ek aztertu zuten askotan sortutako galderako galde-hitza eta erantzunak eskatzen duen galde-hitza ez datozela bat, adibidez, galderak *nora* galde-hitza eskatzen duela

baina sortutako galderak *nondik* galde-hitza erabiltzen duela. Hori konpontzeko, galde-hitza era berezi batean sortzea proposatzen dute. Dekodetzean galde-hitza noiz sortu erabakitzen duen ikasten den aldagai bat erabiltzen dute, eta sortzea erabakitzen denean galde-hitzen hiztegi berezi baten gaineko probabilitate-banaketa erabiltzen dute. [Duan et al.](#)-ek (2017) ere galde-hitza era berezian sortzen dute, baina bi kodetzaile-dekodetzaile erabilia; bata testuiguruaren arabera galdera txantilo bat sortzeko eta bestea txantiloieko hutsuneak betetzeko.

Paragrafo mailako testuingurua

Paragrafoek eman dezaketen informazio gehigarria ondo ustiatzeko eta garrantzitsua ez den informazioa baztertzeko [Zhao et al.](#)-ek (2018) *gated self-attention* erabiltzen duen kodetzaileaz baliatzen dira bakarrik informazio garrantzitsua eskuratzeko.

Dokumentu hau idazterakoan BLEU, METEOR eta ROUGE metrika automatikoetan oinarrituta balio altuenak lortzen dituen eredu; hau da, artearen egoera dena, [Zhao et al.](#)-ek proposatutakoa da.

Erantzun posibleak bilatzen

Orain arte aipatutako hobekuntzek erantzuna sarrera bezala hartzen dute, baina erantzuna zein den alde aurretik jakiten dela suposatzen dute.

Hurrengo pauso bezala, erantzuna identifikatzea proposatzen da. Era horretara eginda, sare neuronaletan oinarritutako galderen sorkuntza, galderen sorkuntza tradizionalan egi-ten zen atazaren deskonposaketa hartzen du; hau da, alde batetik zer galdetuko den auke-
ratu eta ondoren hori nola galdetuko den. Bi artikulu argitaratu dira horren inguruan [Du and Cardie](#) (2017) eta [Subramanian et al.](#) (2017).

Bestelako proposamenak

Ataza anitzeko ikasketa automatikoko alor bat da, non aldi berean ataza bat baino gehiago ebaztea den helburua. Oro har, elkarrekin erlazionatutako atazak ikasten dira batera. Hori dela eta, galderen sorkuntza analisi semantikoarekin batera ikastea proposatzen dute [Guo et al.](#)-ek (2018). [Sachan and Xing](#)-ek (2018), berriz, galderen-erantzunerako atazarekin batera ikasten dute.

Testua erabiltzeaz gain beste erako sarrerak erabiltzea ere proposatu da. 2 kapituluaren aipatu zen sare neuronalen erabilera galderen sorkuntzarako erabiltzen lehenengoak [Serban et al.](#) izan zirela eta hirukoteen gainean egiten dutela. Horren ildora, [ElSahar et al.](#)-ek (2018) eredu horrek zituen zenbait arazo, batez ere gutxi ateratako entitateekin, konpontzeko kopiatzeko mekanismoa eta ezaugarri linguistikoak gehitzen dituzte.

Kontzeptu-mapak erabiltzeaz gain irudiak sarrera iturri bezala erabiltzea proposatu da baita ere ([Zhang et al.](#), 2016).

Models	Answer Encoding	Features					Performance		
		QW	PC	CP	LF	PG	BLEU-4	METEOR	ROUGE _L
Du et al. (2017)	not used						12.28	16.62	39.75
Duan et al. (2017)	not used	•					12.28	–	–
Zhou et al. (2017)	answer position			•	•		13.29	–	–
Yuan et al. (2017)	answer position			•		•	10.50	–	–
Wang et al. (2018)	answer position			•	•		13.86	18.38	44.37
Harrison et al. (2018)	answer position			•	•		14.39	19.54	43.00
Kumar et al. (2018b)	not used			•	•	•	16.17	19.85	43.90
Sun et al. (2018)	answer+context position	•		•			15.64	–	–
Zhao et al. (2018)	answer position		•	•			16.38	20.25	44.48
Du and Cardie (2018)	answer position		•	•			15.16	19.12	–
Song et al. (2018)	separate encoder			•			13.98	18.77	42.72
Kim et al. (2019)	separate encoder						16.20	19.92	43.96

6.1 Taula: Galderen sorkuntzaren artearen egoera ([Pan et al.](#), 2019). **QW:** galde-hitzaren sorkuntza, **PC:** paragrafo mailako testuingurua, **CP:** kopiatzeko mekanismoa, **LF:** ezaugarri linguistikoak, **PG:** *policy gradient*

6.1.2 Hobekuntzak arlo pedagogikoan

Galderen sorkuntza espreski alde pedagogikora bideratzen duten ikerketak ez dira hain ugariak, baina baten bat badago. Horren adibide bat [Wang et al.](#)-ek (2018) proposatutako eredia da. Bertan, proiektu honetan erabili den arkitekturaren ia berdina duen eredia erabili da; entrenamendua SQuAD datu-multzoan gauzatu dute, baina ebaluatzeko OpenStax ikas-liburuetatik ateratako datu-multzoaren gainean probatu dute emaitza onak lortuz. Beraz, ondorioztatzen dute galderak sortzeko SQuADen entrenatutako eredia ondo egokitzen dela OpenStax erako ikas-liburuetara.

LearningQ [Chen et al.](#)-ek (2018) alde pedagogikora bideratua dagoen sortutako datu-multzoa da. Online ikasteko plataforma nagusienetatik hartutako 230 mila dokumentu-galdera pare biltzen ditu; horietatik, 7 mila inktoreek sortutako galderak dira eta 223 mila ikasleek ikasketa-prozesua zehar sortutakoak. Era desberdineko ikasketa gaiak hartzen ditu: zientzia, matematika, humanitateak, eta abar. Galdera-erantzunerako beste datu-multzoekin alderatuta, galderak maila sakonagoko gaitasunak eskatzen dituzte. Gainera,

galderen erantzun gehienak ez dira zuzenean testuko zati bat eta dokumentuko zenbait esaldiren arteko arrazoiketa eskatzen dute.

[Chen et al.](#)-ek bai erregeletan oinarritutako bai kodetzaile-dekodemak arkitekturan oinarritutako ereduak egiten dute proba LearningQ datu-multzoaren gainean, eta beraien emaitzek ikustera ateratzen dute eredu horiek ez dituztela kalitate altuko galdera pedagogikoak sortzen.

6.2 Hobekuntza berrien proposamena

Hizkuntzaren Prozesamenduaren arloko beste ataza batzuetan artearen egoera lortzen ari diren zenbait teknika daude, eta momentuz ez da horietako batzuekin probarik egin galderen sorkuntzan; horregatik, hemen bi teknikaren azterketa egiten da.

6.2.1 Transformerra

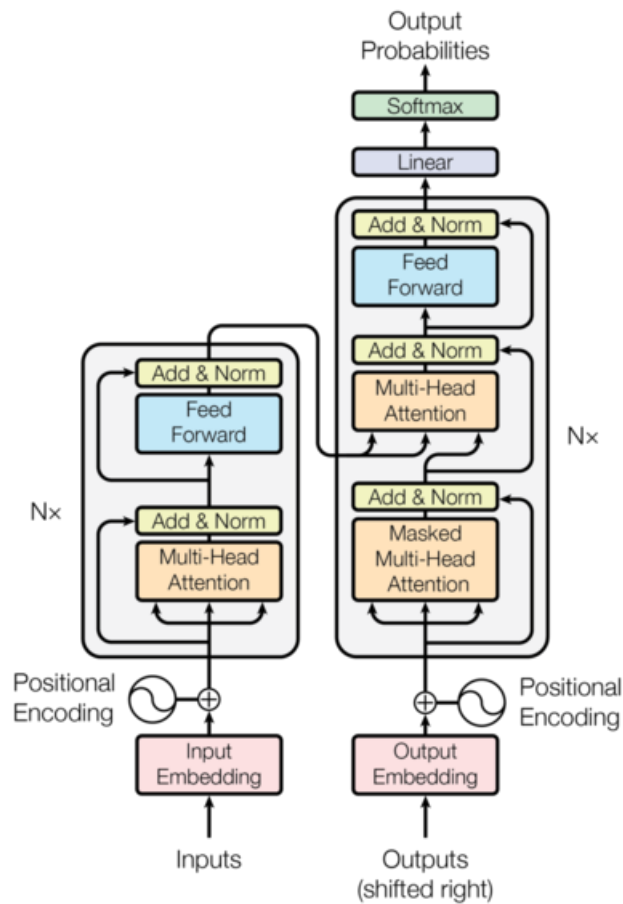
Transformerra [Vaswani et al.](#)-ek (2017) itzulpen automatikorako proposatutako eredia da. Ordura arte itzulpen automatikoko artearen egoera kodetzaile-dekodemak arkitektura erabiltzen zuen eredia zen. Sare errekurteek sekuentzia oso luzeekin arazoak izaten dituzte, eta LSTMek arazo hori zertxobait arintzen badute ere, arazoa ez da guztiz konpontzen.

Transformerra sare errekurteetan oinarritu eta sekuentziak elementuz elementu tratatu ordez, sekuentzia osoaren gainean egiten du lan. Paraleloan elementu guztiak tratatzeko sare konboluzionalak erabiltzen dituzte atentzioarekin batera, zehazki *self-attention* erabiltzen du. *Self-attention* 3.5 atalean aztertutako atentzio mekanismoaren ideia berdinean oinarritzen da, baina atentzioa uneko tokenean oinarrituz kalkulatu ordez sekuentziako elementu bakoitzak beste guztiekiko duen atentzioa kalkulatu da.

Transformerrak sei kodetzaile eta sei dekodetzaile ditu. Kodetzaile guztiek bi geruza dituzte, lehenengoan *self attention* gauzatzen da eta bigarrena *multilayer perceptroni* bat da. Dekodemak ere kodetzaileak dituen bi geruza horiek ditu, baina bien tartean beste atentzio geruza bat dauka sarrerako sekuentziako zati garrantzitsuetan zentratzeko (6.1 irudia).

Transformerra itzulpen automatikorako ez ezik, beste sekuentziatik sekuentziarako ataze-

tan erabili izan da arrakastaz. Hori dela eta, galderen sorkuntza egiteko probatzea interesgarria izango litzateke.



6.1 Irudia: Transformer (Vaswani et al., 2017)

6.2.2 Testuinguruaren araberako esanahi-bektoreak

Aztertu diren galderen sorkuntzako artikuluetan landu ez den, baina Hizkuntzaren Prozesamenduko beste ataza askotan artearen egoera hobetu duen beste gauza bat aurre entrenatutako hizkuntza-ereduen erabilera da.

Hizkuntza-ereduen helburua sekuentzia bateko hurrengo hitza iragartzea da. Aurre entrenatutako hizkuntza-ereduak helburu hori bilatzen duten datu-multzo erraldoietan entrenatutako ereduak dira. Aurre entrenatutako eredu horiek gauza desberdinetarako erabili daitezke, adibidez, testuinguruak kontuan hartzen duten esanahi-bektoreak lortzeko edota Hizkuntza Prozesamenduko beste ataza askotarako hasierako oinarri bezala erabili eta doitze fina egiteko.

4.2.3 azpiatalean aztertutako esanahi-bektoreetan token bakoitzari bektore bat dagokio, baina askotan token batek esanahi desberdinak ditu agertzen den testuinguruaren arabera. Aurre entrenatutako hizkuntza-ereduak erabiliz lortutako esanahi-bektoreek tokenaren testuinguru kontuan hartzen dute, esanahi-bektoreak dinamikoki moldatuz. Esanahi-bektore horiek proiektu honetan garatutako kodetzaile-dekodemak arkitekturan erabiltzea onuragarria litzatekeela pentsatzen da; era horretan, ereduak lortutako esaldien kodeketa kalitatea handiagoko izango delako.

Aurre entrenatutako hizkuntza-eredu ezagunenak ondokoak dira: ELMo (Peters et al., 2018), Flair (Akbik et al., 2018), ULMFiT (Howard and Ruder, 2018), BERT (Devlin et al., 2018) eta GPT-2 (Radford et al., 2019) dira. Jarraian ELMo eta BERT sakonago aztertzen dira, hizkuntza-ereduen funtzionamendua hobetu ulertzeko.

ELMo

ELMo Peters et al.-ek (2018) proposatutako aurre entrenatutako hizkuntza-eredua da, eta esanahi-bektoreek hitz mailako ezaugarriak izateaz gain testuinguru kontuan hartzea du helburu.

ELMo bi noranzko hizkuntza-eredu baten barneko egoeren gaineko funtzio lineal batean oinarritzen da. Bi noranzko hizkuntza-eredua bi hizkuntza-ereduz dago ostua: aurreranzko hizkuntza-eredua eta atzeranzko hizkuntza-eredua.

Aurreranzko hizkuntza-ereduak sekuentzia baten probabilitatea kalkulatzeko aurreko testuinguruaren, $(t_1, t_2, \dots, t_{k-1})$, arabera t_k tokenaren probabilitatea modelatuz egiten du:

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1})$$

Atzeranzko hizkuntza-ereduak antzeko gauza egiten du, baina t_k tokenaren probabilitatea etorkizuneko $(t_{k+1}, t_{k+2}, \dots, t_N)$ testuinguruan oinarrituz kalkulaten da:

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)$$

ELMok bi hizkuntza-ereduen egiantza-funtzio logaritmikoa maximizatuz ikasten du, eta hizkuntza-eredu bakoitza LSTM bat erabiliz inplementatzen da.

BERT

BERT [Devlin et al.-ek \(2018\)](#) garatutako aurre entrenatutako hizkuntza-eredua da, eta transformer arkitekturaren kodetzailean oinarritzen da. Beraz, ELMorekin konparatuz, ez da sare errekurteetan oinarritzen baizik eta *self attention* mekanismoan.

BERT transformerraren kodetzailean oinarritzen da, bi norabideko *self-attention*-arekin. Bi norabideetan kodetzen duenez bai iraganeko bai etorkizuneko informazioa dauka, eta, beraz, hurrengo tokena iragarri behar badu tranpa da, jadanik token horren informazioa jaso duelako. Hori dela eta, entrenamendu datuetako sekuentzien zenbait token maskara ikur batekin ordezkatzeko dira; ataza horri maskaradun hizkuntza-eredua deritza, eta BERTek ataza hori ebatziz ikasten ditu testuinguruaren araberrako esanahi-bektoreak.

7. KAPITULUA

Ondorioak eta etorkizunerako lana

Kapitulu honetan proiektuan zehar lortutako ondorioak aztertuko dira, baita ondorio pertsonalak ere. Azkenik, 6 kapituluan aztertutakoan oinarrituz etorkizunean proiektuari jarraipena emateko egin daitekeena zehazten da.

7.1 Ondorioak

7.1.1 Proiektuaren ondorioak

Proiektuan zehaztutako helburuak bete dira. Alde betetik, oinarritzat hartutako ereduaren berrinplementazioa egitea lortu da eta ereduaren funtzionamenduaren ulermen sendoa lortu da. Bestetik, artearen egoeran egin diren hobekuntzak aztertu dira eta beste hobekuntza berriak aztertu eta proposatu dira baita ere.

Inplementazioari dagokionez, zerotik hasita berrinplementatutako eredia atentzio mekanismoa erabiltzen duen kodetzaile-dekodemakina izan da, galderen sorkuntzarako *beam search* algoritmoa erabiltzen duena.

Inplementatutakoaren oinarri teorikoaren ulermen sakona lortzeko ondoko aspektuak landu dira: sare neuronalak, sare neuronalen ikasketa, sare neuronal errekurrenteak, kodetzaile-dekodemakina arkitektura, atentzio mekanismoa, hitzen errepresentazioa, ereduaren ebaluazioarako metrikak, ereduaren entrenamendua errazteko estrategiak eta sorkuntzarako *beam search* algoritmoa.

Artearen egoeran egon diren hobekuntzen azterketa egin da; alde batetik, proiektu honetan inplementatutako kodetzaile-dekodatzaile arkitekturaren gainean hobekuntzak egiten dituztenak aztertuz, eta, bestetik, beste erako proposamenak egiten dituztenak. Gainera, bi proposamen berri aztertu dira: transformer arkitektura probatzea galderen sorkuntza egiteko eta testuinguruaren arabera

7.1.2 Ondorio pertsonalak

Maila pertsonalean proiektuaren garapena oso esperientzia aberasgarria izan da hainbat aspekturengatik: eskuratutako ezagutza, ezagutza hori aplikatzea, eskala handiko proiektu bat kudeatu behar izatea eta proiektua garatzeak sortu dizkidan hausnarketa pertsonalak.

Eskuratutako ezagutzari dagokionez, proiektu honetan ikasketa sakoneko eredu bat inplementatu denez, eskuratutako ezagutza batez ere arlo horren ingurukoa da. Proiektua hastean ikasketa sakonari buruz ez neukan ia ezagutzarik, eta, beraz, tutoreek gomendatutako liburu eta zenbait artikuluren bidez gaia lantzen hasi nintzen. Aurrerago, konputazio adarreko *Deep Learning and Neural Networks* irakasgaiaren oinarri zabalagoa eskuratu nuen. Azkenik, dokumentu honen garapena egiteko are gehiago sakondu ahal izan dut kontzeptu askoren inguruan.

Graduko beste zenbait irakasgaitan landutako kontzeptu asko lagungarriak izan dira baita ere. Alde batetik, konputazio adarreko Algoritmoen Diseinua, Adimen Artifiziala eta Datu meatzaritza irakasgaiak. Bestetik, ikasketa sakonak daukan oinarri matematikoa dela eta Aljebra, Kalkulua eta Estatistika irakasgaietan landutako kontzeptu askoren aplikagarritasuna ikusi ahal izan dut. Azkenik, kodea garatzeko objektuetara orientatutako metodologia jarraituz egin da, horretarako, Programazio Modularra eta Objektu Orientazio irakasgai oinarri izanik.

Inplementazioa egiteko *Python* programazio-lengoaierako *Pytorch* liburutegia erabili da. *Python* programazio-lengoaia aurretik ezagutu arren bere jakintza hobetu da. Aldiz, *Pythorch* liburutegia ez nuen ezagutzen, eta inplementatutako arkitekturak duen konplexutasuna dela eta, liburutegiaren maneiua ona eskuratu da. Ikasketa sakoneko eredu bat inplementatzean erroreak aurkitzea nahiko zaila izaten dela ikusi dut baita ere.

Artearen egoeran egondako hobekuntzen azterketa eta memoria egiteko irakurritako artikuluko zientifiko kopurua oso handia izan da. Proiektua egin baino lehen ez nuen esperientzia handirik artikuluko zientifikoak irakurtzen, eta beste pertsonen egindako lana irakurtzea asko gustatzen zaidan jardura bat dela ikusi ahal izan dut.

Proiekturako memoria idazteak espero nuena baino lan eta buruhauste gehiago ekarri dizkit. Momentu askotan zaila egin zait zer idatzi nahi nuen aukeratzea eta idatzi nahi nuen hori ahalik eta era ulergarrienean egitea. Konturatu naiz ingeniari izan arren hizkuntzarekin erraztasuna izatea, bai idatziz bai hitz eginez, premiazkoa dela; izan ere, egindako lana gainontzekoei ondo transmititzea lana ondo egitea bezain garrantzitsua iruditzen zaidalako orain. Hori dela eta, etorkizunean aspektu horiek nire buruan gehiago lantzeko nahia sortu zait.

Proiektua IXA taldearen baitan garatu denez, ikerketa-talde baten funtzionamendua barrutik ikusi ahal izan dut, bai eta taldeko jendearekin lan egin. Taldean lan egiteak niretzat ezezaguna zen Hizkuntzaren Prozesamenduaren arloan murgiltzea ahalbidetu dit, horren inguruko jakin-mina sortuz, ez bakarrik alde konputazionaletik, alde linguistikotik ere.

Proiektuan zehar landutako gauzak oso interesgarriak egin zaizkit; izan ere, azken urteotan gizakion ikasketa-prozesua, makinaren ikasketa-prozesua eta bien arteko analogiak oso erakargarriak iruditu zaizkidan gaiak direlako. Proiektu honi esker gai horien inguruan gehiago sakondu ahal izan dut ikuspegi desberdinetatik. Alde batetik, galderek gizakion ikasketa-prozesuan duten eraginarengatik. Bestetik, sare neuronalek ikasteko erabiltzen duten prozesuarengatik. Azkenik, gizakion ikasketa-prozesuan eta makinaren ikasketa-prozesuan sortzen diren analogiengatik: *overfitting*, *underfitting*, orokortzea, helburu-funtzioa, ikasketa algoritmoa eta abar. Pertsonalki, analogia horietaz jabetzea nire ikasketa-prozesuaren eta izan dudana hezkuntza sistemaren inguruan hausnarketa sakonak egitera eramanez dit.

7.2 Etorkizunerako lana

Maila desberdineko gaitasun kognitiboak eskatzen dituzten galderak sortuko dituen ereduak lortzeko bide luzea dago oraindik. 6 kapituluaren hainbat hobekuntzen azterketa eta proposamena egin da. Atal honetan aztertu diren hobekuntzen laburpena gehi azter litezkeen zenbait proposamen zerrendatzen dira:

- **Galdera baldintzatu:** Implementatutako ereduak esaldi bat jasotzen du sarrera gisa, eta horretan oinarrituz galdera bat sortzen du, baina esaldi batetik galdera bat baino gehiago sor daiteke. Galdera baldintzatzeko zenbait aukera daude. Alde batetik, erantzuna gehitzea; horretarako, lehen pauso bezala erantzun hori alde aurretik ezaguna dela suposa daiteke, baina praktikan gehienetan erantzuna ez da ezaguna izango. Beraz, ereduak erantzun interesgarriak bilatzeko gai izan beharko luke. Bes-

te aldetik, erantzunarekin ez ezik, galderaren intentzioa edo zailtasuna mailarekin baldintzatzea interesgarria litzateke, batez ere alde pedagogikotik begira.

- **Galderaren irismena zabaldu:** Sortutako ereduaren galderen irismena esaldi mailakoa baino ez da, eta askotan esaldiko informazioa ez da nahikoa galdera on bat egiteko. Aukera oso interesgarri bat galderaren irismena zabaltzea litzateke. Paragrafo mailara zabaltzeko saiakerak egin badira ere, testu osoa kontuan hartzen duen eredia lortzea litzateke azken helburua.
- **Testuinguruan oinarritutako esanahi-bektoreak:** Inplementatutako ereduak erabiltzen dituen esanahi-bektoreak finkoak dira; hau da, hitz batek testuinguru batetik bestera duen esanahia aldatzen bada ere, esanahi-bektore berdina egokitzen zaio. Testuingurua kontuan hartzen duten esanahi-bektoreak erabilia esaldiaren kodeketa kalitate altuagokoa izango dela pentsatzen da.
- **Transformerra:** Transformerra arrakastaz erabili da sekuentziatik sekuentziarako beste ataza batzuetan, adibidez, itzulpen automatikoan. Interesgarria litzateke galderen sorkuntzarako nola moldatzen den aztertzea.
- **Ebaluazio metrika hobegoak:** Ikusi denez galderen sorkuntzarako, eta, oro har, Hizkuntzaren Prozesamenduko beste zenbait atazako sistemak ebaluatzeko erabiltzen diren metrikak ez dira guztiz egokiak, aspektu asko kontuan hartzen ez dituztelako. Alde pedagogikotik galderen kalitatea ebaluatzen duen metrika automatikoak aztertzea interesgarria litzateke.

Eranskinak

Proiektuaren Helburuen Dokumentua

Atal honetan proiektuaren inguruko xehetasunak emango dira, honen inguruko deskribapena, helburua, plangintza, lan metodologia, arrisku-plana eta kontingentzia-plana azalduz.

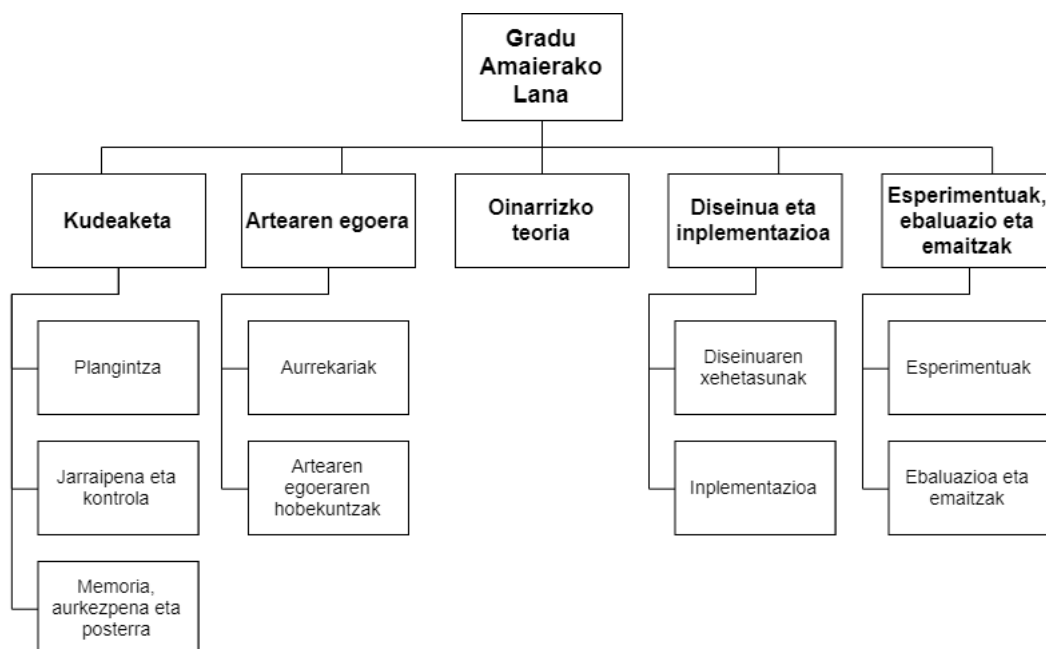
A.1 Proiektuaren deskribapena eta helburuak

Proiektu honen helburua galderen sorkuntzarako erabiltzen hasi den sare neuronaletan oinarritutako eredu bat berrinplementatzea da; horretarako, (Du et al., 2017)-ek (2017) egin-dako lana hartu da oinarritzat. Eredua berrinplementatzearen helburua sistemaren ulermen sakon bat lortu eta arlo pedagogikora begira etorkizunean egin daitezkeen hobekuntzen azterketa egitea da.

A.2 Proiektuaren plangintza

A.2.1 LDE diagrama

Proiektuan zehar burutu beharreko lanaren deskonposaketa egin da Lanaren Deskonposaketa Egitura (LDE) diagrama erabiliz (A.1 irudia).



A.1 Irudia: Lanaren Deskonposaketa Egitura

A.2.2 Lan-paketeak

LDE diagraman definitutako lan-paketeen deskribapen zehatza egin da. Ataza bakoitzari zehaztutako ordu kopurua [A.1](#) taulan agertzen dira. Ordu kopurua hurbilketa bat da eta proiektuan zehar aldatu daiteke. Ikusi daitekeen bezala inplementazioari esleitu zaio ordu kopuru gehien; hala ere, txostena eta oinarrizko teoria ordu kopuru handia behar izatea estimatzen da baita ere.

Plangintza

Ataza honetan proiektuaren planifikazioa garatu da. Garrantzitsua izango da proiektuaren helburuak zein atazak definitzea, emangarriak zehaztea eta mugarriak zein lan-metodologia aurrez definitzea. Gainera, proiektuaren bideragarritasuna eta arrisku-plana ere garatuko dira.

Jarraipena eta kontrola

Ataza honetan proiektuko helburu eta mugarri guztiak betetzen direla bermatuko da. Ustekabekorik gertatuz gero, proiektuak aurrera jarraitzen duela kontrolatuko da. Horretarako,

zenbait jarraipen bilera burutuko dira. Ataza hau proiektuaren bizitza-ziklo osoan zehar egongo da aktibo.

Memoria, aurkezpena eta posterra

Ataza honetan proiektuaren nondik norakoak azaltzen dituzten memoria, aurkezpena eta posterra egingo dira.

- **Memoria:** Proiektuaren inguruko xehetasunak biltzen dituen dokumentua garatuko da.
- **Aurkezpena:** Proiektuaren defentsarako lagungarri izango den aurkezpena garatuko da. Aurkezpenean memorian azalduko diren puntu garrantzitsuenak agertu beharko dira, proiektua azaltzeko modu argi eta antolatuan.
- **Posterra:** Proiektuari ikusgarritasuna emateko Informatika Fakultateko sarreran jarriko diren Gradu Amaierako Lanen posterra garatuko da. Bertan proiektuaren laburpen orokor bat egingo da era argi eta ikusterraz batean.

Aurrekariak

Ataza honetan inplementatuko den eredia baino lehen erabiltzen ziren metodoak aztertuko dira, erregeletan oinarritutako metodoak hain zuzen ere. Metodo horien bidezko galderen sorkuntzaren ikuspegi orokor bat izateko hiru hurbilpen desberdinen azterketa egingo da.

Artearen egoeraren hobekuntzak

Ataza honetan artearen egoeran egin berri diren hobekuntzen azterketa egingo da ikasketa sakonean oinarritutako galderen sorkuntzaren inguruko artikulu berriak landuz. Era horretako sistemek zer norabide hartzen ari diren ikusteko. Proposamen berriren bat egingo da baita ere.

Oinarrizko teoria

Ataza honetan ikasketa sakonaren bidezko galderen sorkuntza egiteko beharrezko oinarri teorikoaren azterketa sakona egingo da, berrinplementatuko den sistemaren ulermen sendoa izateko.

Diseinuaren xehetasunak

Ataza honetan berrinplementatuko den sistemaren diseinuaren xehetasun guztiak aztertuko dira.

Inplementazioa

Ataza honetan sistemaren berrinplementazioa egingo da. Inplementazioa *Python* lengoaiarako *Pytorch* liburutegia erabiliz egingo da, beraz liburutegi horren antolaketa eta funtzionamendua ere landu beharko dira.

Esperimentuak

Ataza honetan berrinplementatutako sistemaren esperimentazioa egingo da, berrinplementazioa arrakastatsua izan den edo ez ikusteko.

Ebaluazioa eta emaitzak

Ataza honetan esperimentazioan lortutako emaitzen azterketa egingo da, eta berrinplementazioa arrakastatsua izan den edo ez ondorioztatuko da.

A.2.3 Emangarriak

Atal honetan proiektuan garatu beharreko emangarriak identifikatzen dira:

- Kodea
- Memoria
- Aurkezpena
- Posterra

Lan-paketea	Iraupena (ordutan)
Kudeaketa	145
Plangintza	5
Jarraipen eta kontrola	40
Memoria, aurkezpena eta posterra	100
Artearen egoera	65
Aurrekariak	15
Artearen egoeraren hobekuntzak	50
Oinarrizko teoria	100
Diseinua eta implementazioa	145
Diseinuaren xehetasunak	5
Implementazioa	140
Esperimentuak, ebaluazio eta emaitzak	25
Esperimentuak	10
Ebaluazio eta emaitzak	15
GUZTIRA	480

A.1 Taula: Lan-pakete bakoitzari esleitutako denbora

A.2.4 Mugarriak

Atal honetan proiektuan zehar garatu beharreko emangarrien emate-datak identifikatzen dira [A.2](#) taulan.

Emangarria	Amaiera-data
Kodea	2019-06-23
Memoria	2019-06-23
Posterra	2019-06-23
Aurkezpena	2019-07-1/12

A.2 Taula: Mugarriak

A.2.5 Gantt diagrama

Atal honetan [A.2](#) irudian agertzen den *Gantt-diagrama* garatu da proiektuaren mugarriak eta atazak kontuan izanik.

1. **Baliabideen kostua.** Proiektuan erabiliko diren baliabide guztiak eskuragarri daudela eta doakoak direla bermatu da.
2. **Baliabideen funtzionamendu bermea.** Proiektuan erabiliko diren baliabideak proiektuaren garapenean prest eta atzigarri egongo direla bermatu da.
3. **Denbora.** Proiektua garatzeko beharrezkoa den denbora egongo dela bermatzen da. Planifikatuta dagoen denborak izan ditzakeen desbideraketak kontuan izanik.
4. **Komunikazioa.** Aurretik aipatu bezala ikaslearen eta tutoreen arteko komunikazioa egokia eta eraginkorra dela bermatzen da.

A.5 Arriskuak eta prebentzioa

Proiektuaren dimentsioa kontuan izanik, hainbat arrisku eta oztopo egon daitezke garapenaren zehar, zeinen proiektuaren arrakasta baldintzatu dezakete. Hori dela eta, arrisku horiek identifikatu eta horien eragina aurreikusteko saiakera egin da. Hala ere, zerrenda honetan agertzen ez diren arriskuak gerta daitezkeela kontuan hartu behar da.

A.5.1 Arriskuak

- Mota honetako proiektuetan normala izaten da atazaren bat osatzeko aurre planifikatutako baino denbora gehiago behar izatea.
- Prozesuan zehar garrantzitsuak diren daturen galerak gerta daitezke.
- Ikasketa sakoneko eredu bat sortzen denez, horrek ikasteko behar duen denbora handia izaten da.

A.5.2 Prebentzioa

Aurreko azpiatalean azaldutako arrisku-plana kontuan izanik ondorengo prebentzio-plana sortu da arriskuak murrizteko.

- Proiektuan zehar egon daitezken atzerapenak direla eta, plangintza malgu bat sortzea erabaki da. Gainera, arazo handirik izan ezkeru proiektua ekainean entregatu

ordez, irailean entregatzeko aukera dago, baina ez da hasiera batean beharrezkoa ikusten.

- Proiektuko datuak ez galtzeko, hainbat neurri hartuko dira. Alde batetik proiektuarekin zerikusia daukan edozer *Google Drive* plataforman gordeko da, non fitxategien bertsioen kontrola egiten den. Bestetik proiektuko kodea gorde eta bertsioen kontrola egiteko *bitbucket* plataforma erabiliko da.
- Exekuzio luzeak egin bitartean denbora ez galtzeko beste ataza batzuk aurreratzen hastea erabaki da.

Bibliografía

- Aizenberg, I.Ñ., Aizenberg, N.Ñ., and Vandewalle, J. P. (2000). *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Kluwer Academic Publishers, Norwell, MA, USA.
- Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.
- Aldabe, I., Gonzalez-Dios, I., Lopez-Gazpio, I., Madrazo, J., and Maritxalar, M. (2013). Two approaches to generate questions in basque. *Procesamiento del Lenguaje Natural*, 51, 101-108. Consultado en <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4877/2892> (ISSN edición impresa:1135-5948) (ISSN edición digital:1989-7553).
- Aldabe, I., Maritxalar, M., and Soraluze, A. (2011). Question generation based on numerical entities in basque. In *2011 AAAI Fall Symposium Series*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Bajaj, P., Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.
- Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments.
- Bhattacharyya, P., Mukundan, S., and Shah, R. (2007). Some issues in automatic evaluation of english-hindi mt: more blues for bleu.

- Bloom, B. S., Engelhart, M. B., Furst, E. J., Hill, W. H., and Krathwohl, D. R. (1956). *Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain*. Longmans Green, New York.
- Callison-Burch, C. (2009). Fast, cheap, and creative: Evaluating translation quality using amazon’s mechanical turk. pages 286–295.
- Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluation the role of bleu in machine translation research.
- Chen, G., Yang, J., Hauff, C., and Houben, G.-J. (2018). Learningq: A large-scale dataset for educational question generation.
- Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.
- Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Comput. Linguist.*, 31(1):25–70.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- Du, X. and Cardie, C. (2017). Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Copenhagen, Denmark. Association for Computational Linguistics.
- Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.
- Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

- Dunlosky, J., Rawson, K. A., Marsh, E., Nathan, M. J., and Willingham, D. T. (2013). Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological science in the public interest : a journal of the American Psychological Society*, 14 1:4–58.
- ElSahar, H., Gravier, C., and Laforest, F. (2018). Zero-shot question generation from knowledge graphs for unseen predicates and entity types. *CoRR*, abs/1802.06842.
- Graesser, A. C. and Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137.
- Gülçehre, Ç., Ahn, S., Nallapati, R., Zhou, B., and Bengio, Y. (2016). Pointing the unknown words. *CoRR*, abs/1603.08148.
- Guo, D., Sun, Y., Tang, D., Duan, N., Yin, J., Chi, H., Cao, J., Chen, P., and Zhou, M. (2018). Question generation from SQL queries improves neural semantic parsing. *CoRR*, abs/1808.06304.
- Hacker, D., Dunlosky, J., and Graesser, A. (1998). *Metacognition in Educational Theory and Practice*. Educational Psychology Series. Taylor & Francis.
- Heilman, M. and A. Smith, N. (2009). Question generation via overgenerating transformations and ranking.
- Heilman, M. and Smith, N. A. (2010). Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 609–617, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Hinze, S. R. and Rapp, D.Ñ. (2014). Retrieval (sometimes) enhances learning: Performance pressure reduces the benefits of retrieval practice. *Applied Cognitive Psychology*, 28(4):597–606.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Howard, J. and Ruder, S. (2018). Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.
- Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

- Karpicke, J. and J. Grimaldi, P. (2012). Retrieval-based learning: A perspective for enhancing meaningful learning. *Educational Psychology Review*, 24:401–418.
- Kim, Y., Lee, H., Shin, J., and Jung, K. (2018). Improving neural question generation using answer separation. *CoRR*, abs/1809.02393.
- Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810.
- Kociský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. (2018). The narrativeqa reading comprehension challenge. *CoRR*, abs/1712.07040.
- Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. H. (2017). RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683.
- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- López Gazpio, I. (2013). Seneko: galderak automatikoki sortuz testuak lantzeko aukera ematen duen aplikazioa.
- Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.
- LW, A., DR, K., PW, A., KA, C., Mayer, R., PR, P., D. Raths, J., and MC, W. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*.
- M. Olney, A., Graesser, A., and Person, N. (2012). Question generation from concept maps. *Dialogue & Discourse*, 3.
- Madrazo Azpiazu, J. (2013). Hizkuntzaren prozesamendurako teknikak irakaskuntza arloan: galdera sortzaile automatikoa.
- Mannem, P., Prasad, R., and Joshi, A. (2010). Question generation from paragraphs at upenn: Qgstec system description.

- Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Nema, P. and Khapra, M. M. (2018). Towards a better metric for evaluating question generation systems. *CoRR*, abs/1808.10192.
- Pan, L., Lei, W., Chua, T., and Kan, M. (2019). Recent advances in neural question generation. *CoRR*, abs/1905.08949.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-j. (2002). Bleu: a method for automatic evaluation of machine translation.
- Pashler, H., M. Bain, P., Bottge, B., Graesser, A., Koedinger, K., Mcdaniel, M., and Metcalfe, J. (2007). Organizing instruction and study to improve student learning. ies practice guide. ncer 2007-2004. *National Center for Education Research*.
- Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–.

- Rus, V. and Graesser, A. C. (2009). The question generation shared task and evaluation challenge. In *The University of Memphis. National Science Foundation*.
- Sachan, M. and Xing, E. (2018). Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.
- Serban, I. V., García-Durán, A., Gülçehre, Ç., Ahn, S., Chandar, S., Courville, A. C., and Bengio, Y. (2016). Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *CoRR*, abs/1603.06807.
- Song, L., Wang, Z., Hamza, W., Zhang, Y., and Gildea, D. (2018). Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- Subramanian, S., Wang, T., Yuan, X., and Trischler, A. (2017). Neural models for key phrase detection and question generation. *CoRR*, abs/1706.04560.
- Sun, X., Liu, J., Lyu, Y., He, W., Ma, Y., and Wang, S. (2018). Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, pages 3104–3112, Cambridge, MA, USA. MIT Press.
- Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2016). Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.Ñ., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

- Wang, Z., S. Lan, A., Nie, W., Waters, A., J. Grimaldi, P., and Baraniuk, R. (2018). Qg-net: a data-driven question generation model for educational content. pages 1–10.
- Weinstein, Y., Gilmore, A., Szpunar, K., and McDermott, K. (2014). The role of test expectancy in the build-up of proactive interference in long-term memory. *Journal of experimental psychology. Learning, memory, and cognition*, 40.
- Weinstein, Y., Madan, C. R., and Sumeracki, M. A. (2018). Teaching the science of learning. *Cognitive Research: Principles and Implications*, 3(1):2.
- Zhang, S., Qu, L., You, S., Yang, Z., and Zhang, J. (2016). Automatic generation of grounded visual questions. *CoRR*, abs/1612.06530.
- Zhao, Y., Ni, X., Ding, Y., and Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.
- Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., and Zhou, M. (2017). Neural question generation from text: A preliminary study. *CoRR*, abs/1704.01792.